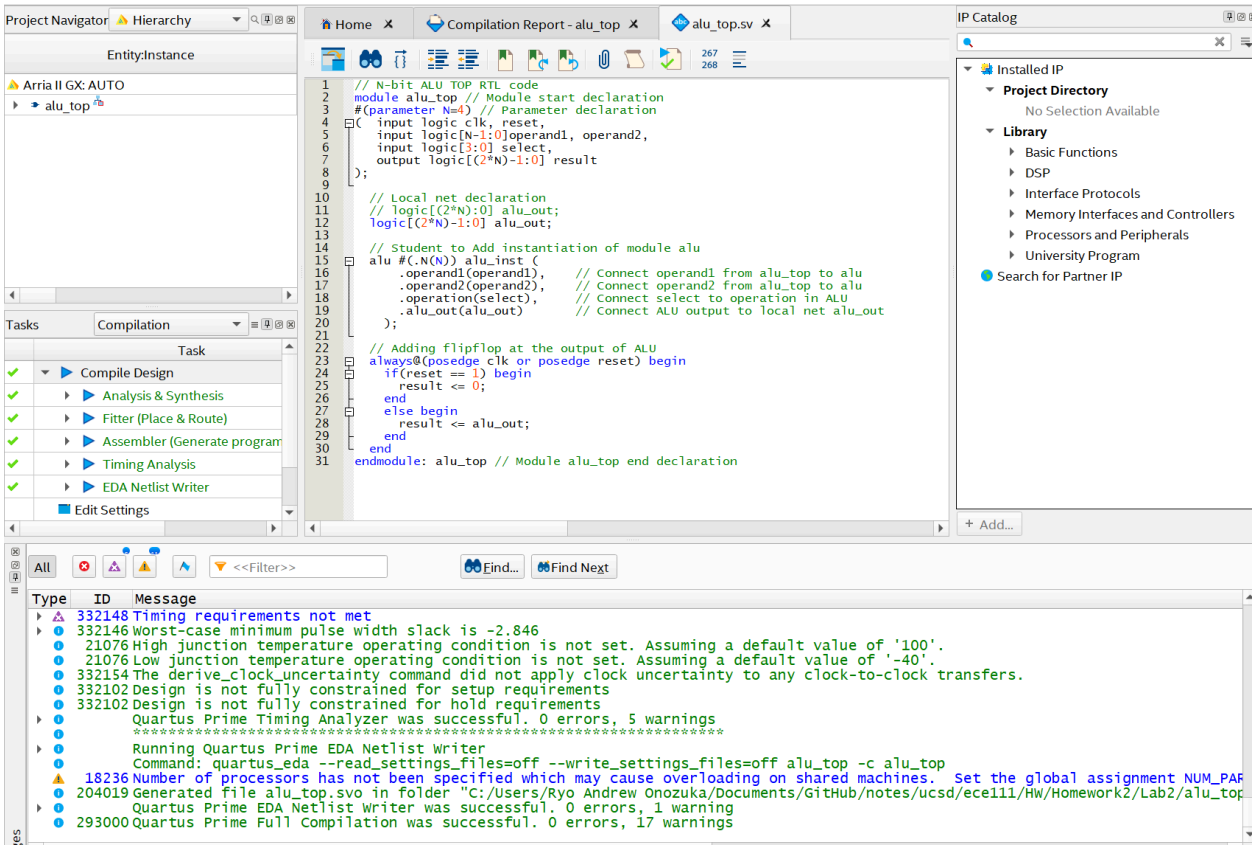
Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

alu_top.sv

ucsd > ece111 > HW > Homework2 > Lab2 > alu_top > alu_top.sv

```
1  // N-bit ALU TOP RTL code
2  module alu_top // Module start declaration
3  #(parameter N=4) // Parameter declaration
4  (  input logic clk, reset,
5     input logic[N-1:0]operand1, operand2,
6     input logic[3:0] select,
7     output logic[(2*N)-1:0] result
8  );
9
10    // Local net declaration
11    // logic[(2*N):0] alu_out;
12    logic[(2*N)-1:0] alu_out;
13
14    // Student to Add instantiation of module alu
15    alu #(.N(N)) alu_inst (
16        .operand1(operand1),     // Connect operand1 from alu_top to alu
17        .operand2(operand2),     // Connect operand2 from alu_top to alu
18        .operation(select),      // Connect select to operation in ALU
19        .alu_out(alu_out)        // Connect ALU output to local net alu_out
20      );
21
22    // Adding flipflop at the output of ALU
23    always@(posedge clk or posedge reset) begin
24      if(reset == 1) begin
25        result <= 0;
26      end
27      else begin
28        result <= alu_out;
29      end
30    end
31 endmodule: alu_top // Module alu_top end declaration
```



Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

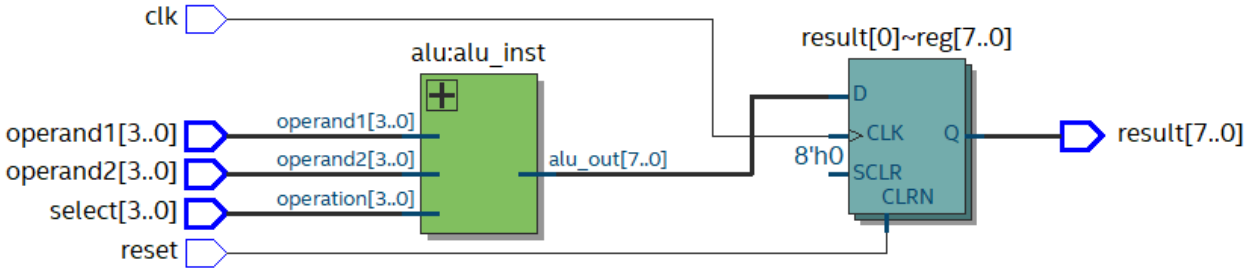
alu_top.sv Resource Usage Summary

ucsd > ece111 > HW > Homework2 > Lab2 > alu_top > alu_top-Resource Usage Summary.rpt

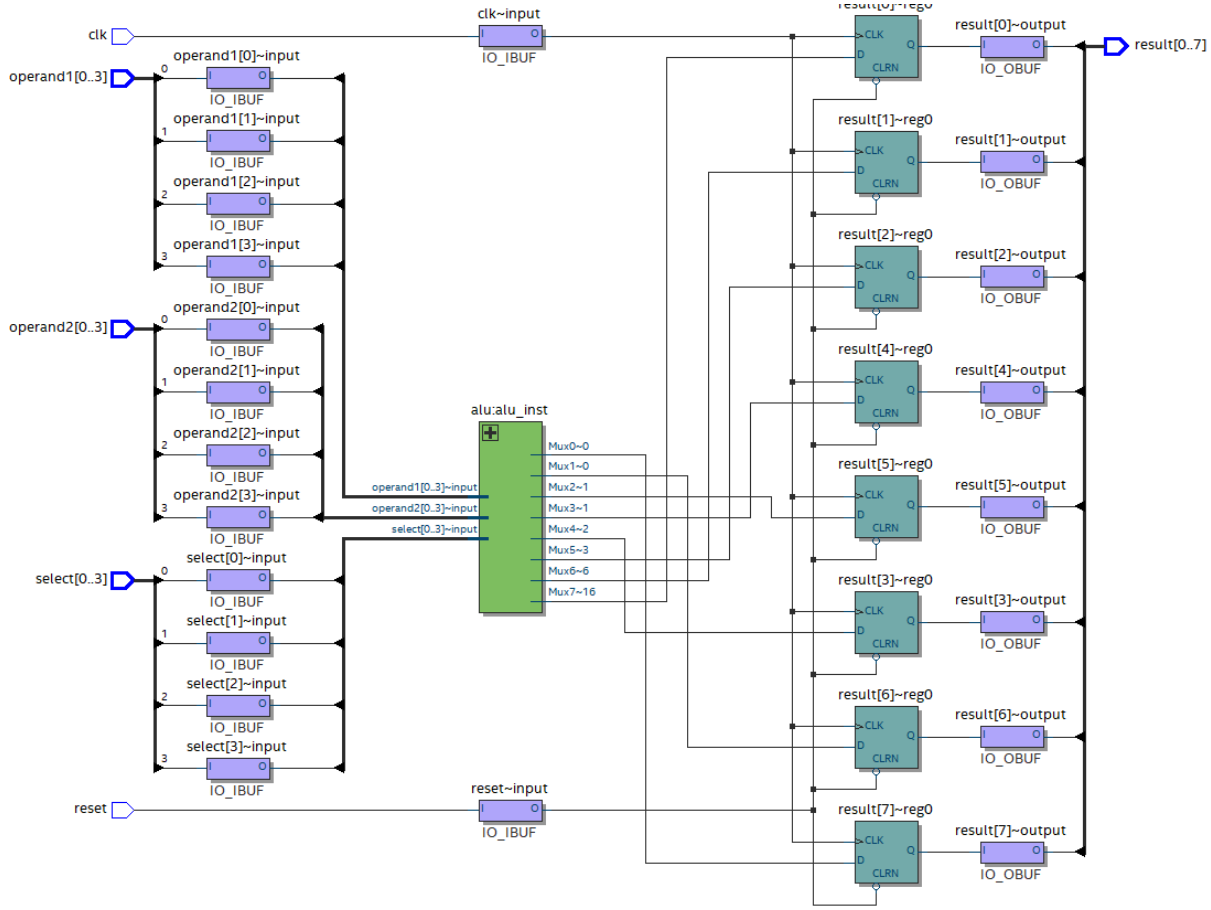
```
+-----------------------------------------------------------------------+
; Analysis & Synthesis Resource Usage Summary                           ;
+-----------------------------------------------+-----------------------+
; Resource                                      ; Usage                 ;
+-----------------------------------------------+-----------------------+
; Estimated ALUTs Used                          ; 113                   ;
;     -- Combinational ALUTs                    ; 113                   ;
;     -- Memory ALUTs                          ; 0                     ;
;     -- LUT_REGs                              ; 0                     ;
; Dedicated logic registers                     ; 8                     ;
;                                               ;                       ;
; Estimated ALUTs Unavailable                   ; 9                     ;
;     -- Due to unpartnered combinational logic ; 9                     ;
;     -- Due to Memory ALUTs                    ; 0                     ;
;                                               ;                       ;
; Total combinational functions                 ; 113                   ;
; Combinational ALUT usage by number of inputs  ;                       ;
;     -- 7 input functions                      ; 3                     ;
;     -- 6 input functions                      ; 6                     ;
;     -- 5 input functions                      ; 21                    ;
;     -- 4 input functions                      ; 31                    ;
;     -- <=3 input functions                    ; 52                    ;
;                                               ;                       ;
; Combinational ALUTs by mode                   ;                       ;
;     -- normal mode                           ; 54                    ;
;     -- extended LUT mode                     ; 3                     ;
;     -- arithmetic mode                       ; 33                    ;
;     -- shared arithmetic mode                ; 23                    ;
;                                               ;                       ;
; Estimated ALUT/register pairs used            ; 122                   ;
;                                               ;                       ;
; Total registers                               ; 8                     ;
;     -- Dedicated logic registers              ; 8                     ;
;     -- I/O registers                          ; 0                     ;
;     -- LUT_REGs                              ; 0                     ;
;                                               ;                       ;
;                                               ;                       ;
; I/O pins                                      ; 22                    ;
;                                               ;                       ;
; DSP block 18-bit elements                     ; 0                     ;
;                                               ;                       ;
; Maximum fan-out node                          ; operand2[3]~input     ;
; Maximum fan-out                               ; 31                    ;
; Total fan-out                                 ; 466                   ;
; Average fan-out                               ; 2.82                  ;
+-----------------------------------------------+-----------------------+
```

Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

alu_top.sv RTL viewer



alu_top.sv post mapping viewer

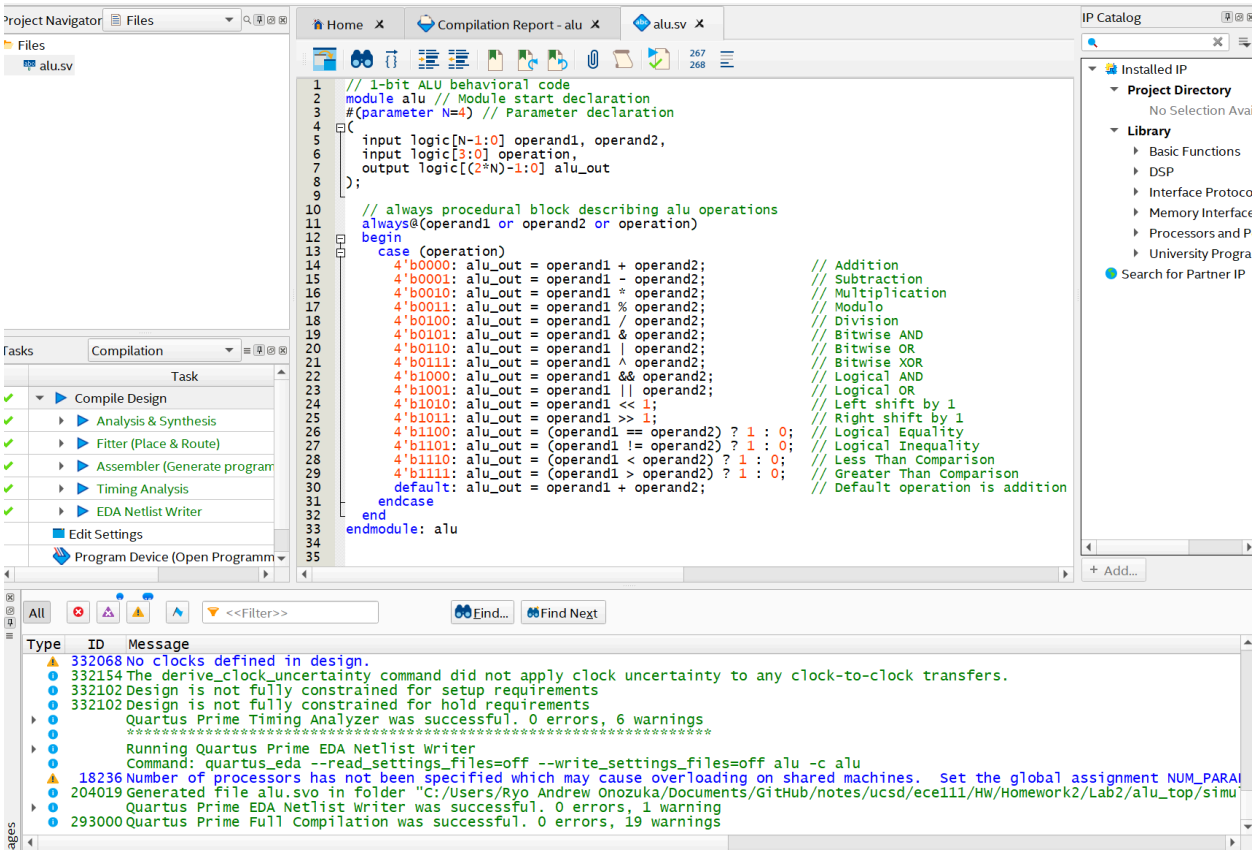


Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

alu.sv

ucsd > ece111 > HW > Homework2 > Lab2 > alu_top > alu.sv

```
// 1-bit ALU behavioral code
module alu // Module start declaration
#(parameter N=4) // Parameter declaration
(
  input logic[N-1:0] operand1, operand2,
  input logic[3:0] operation,
  output logic[(2*N)-1:0] alu_out
);

  // always procedural block describing alu operations
  always@(operand1 or operand2 or operation)
  begin
    case (operation)
      4'b0000: alu_out = operand1 + operand2;            // Addition
      4'b0001: alu_out = operand1 - operand2;            // Subtraction
      4'b0010: alu_out = operand1 * operand2;            // Multiplication
      4'b0011: alu_out = operand1 % operand2;            // Modulo
      4'b0100: alu_out = operand1 / operand2;            // Division
      4'b0101: alu_out = operand1 & operand2;            // Bitwise AND
      4'b0110: alu_out = operand1 | operand2;            // Bitwise OR
      4'b0111: alu_out = operand1 ^ operand2;            // Bitwise XOR
      4'b1000: alu_out = operand1 && operand2;           // Logical AND
      4'b1001: alu_out = operand1 || operand2;           // Logical OR
      4'b1010: alu_out = operand1 << 1;                  // Left shift by 1
      4'b1011: alu_out = operand1 >> 1;                  // Right shift by 1
      4'b1100: alu_out = (operand1 == operand2) ? 1 : 0; // Logical Equality
      4'b1101: alu_out = (operand1 != operand2) ? 1 : 0; // Logical Inequality
      4'b1110: alu_out = (operand1 < operand2) ? 1 : 0;  // Less Than Comparison
      4'b1111: alu_out = (operand1 > operand2) ? 1 : 0;  // Greater Than Comparison
      default: alu_out = operand1 + operand2;            // Default operation is addition
    endcase
  end
endmodule: alu
```



Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

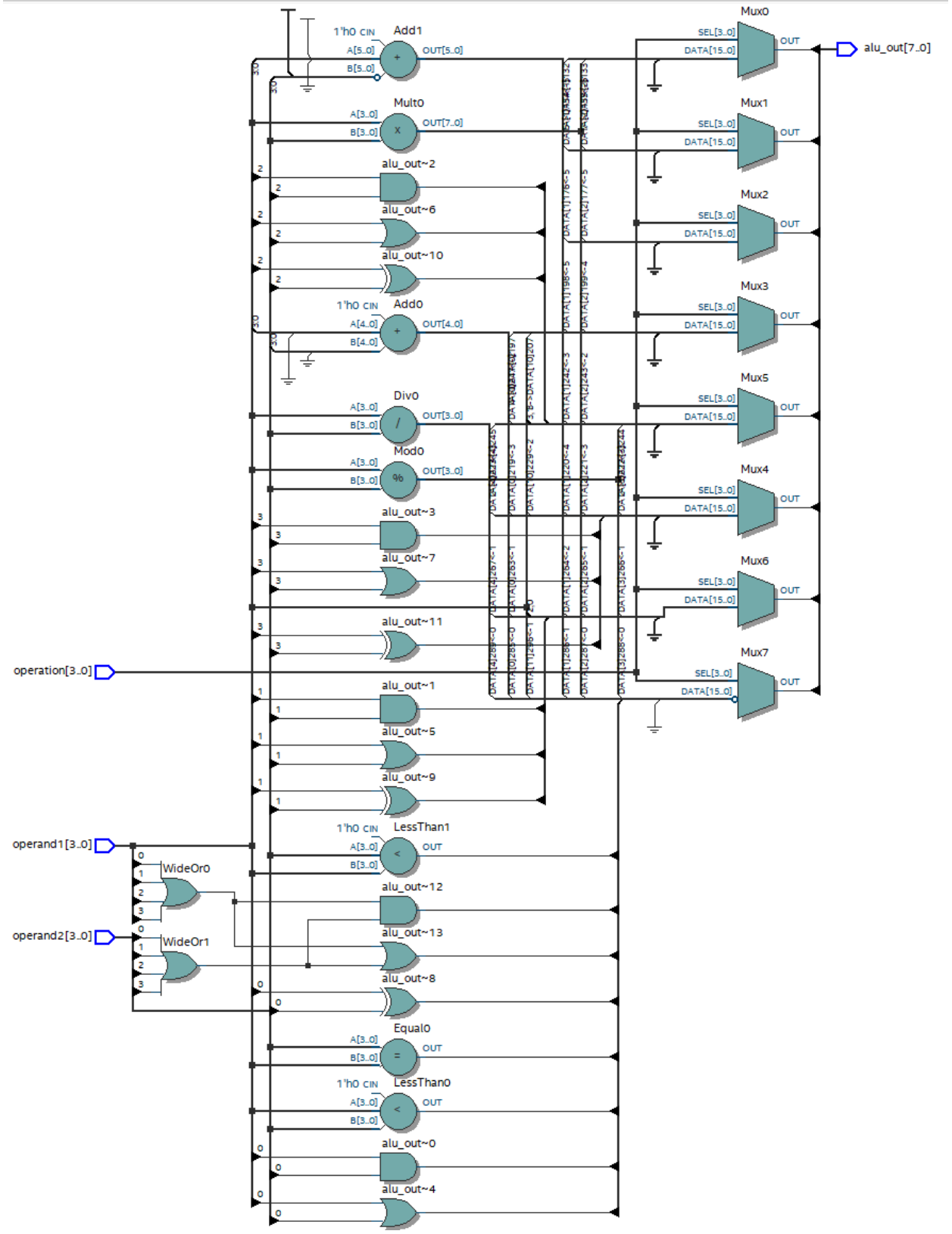
alu.sv Resource Usage Summary

ucsd > ece111 > HW > Homework2 > Lab2 > alu_top > alu-Resource Usage Summary.rpt

```
+-----------------------------------------------------------------------+
; Analysis & Synthesis Resource Usage Summary                          ;
+-----------------------------------------------+-----------------------+
; Resource                                      ; Usage                 ;
+-----------------------------------------------+-----------------------+
; Estimated ALUTs Used                          ; 113                   ;
;     -- Combinational ALUTs                    ; 113                   ;
;     -- Memory ALUTs                          ; 0                     ;
;     -- LUT_REGs                              ; 0                     ;
; Dedicated logic registers                     ; 0                     ;
;                                               ;                       ;
; Estimated ALUTs Unavailable                   ; 9                     ;
;     -- Due to unpartnered combinational logic ; 9                     ;
;     -- Due to Memory ALUTs                    ; 0                     ;
;                                               ;                       ;
; Total combinational functions                 ; 113                   ;
; Combinational ALUT usage by number of inputs  ;                       ;
;     -- 7 input functions                      ; 3                     ;
;     -- 6 input functions                      ; 6                     ;
;     -- 5 input functions                      ; 21                    ;
;     -- 4 input functions                      ; 31                    ;
;     -- <=3 input functions                    ; 52                    ;
;                                               ;                       ;
; Combinational ALUTs by mode                   ;                       ;
;     -- normal mode                           ; 54                    ;
;     -- extended LUT mode                     ; 3                     ;
;     -- arithmetic mode                       ; 33                    ;
;     -- shared arithmetic mode                ; 23                    ;
;                                               ;                       ;
; Estimated ALUT/register pairs used            ; 122                   ;
;                                               ;                       ;
; Total registers                               ; 0                     ;
;     -- Dedicated logic registers              ; 0                     ;
;     -- I/O registers                          ; 0                     ;
;     -- LUT_REGs                              ; 0                     ;
;                                               ;                       ;
;                                               ;                       ;
; I/O pins                                      ; 20                    ;
;                                               ;                       ;
; DSP block 18-bit elements                     ; 0                     ;
;                                               ;                       ;
; Maximum fan-out node                          ; operand2[3]~input     ;
; Maximum fan-out                               ; 31                    ;
; Total fan-out                                 ; 440                   ;
; Average fan-out                               ; 2.88                  ;
+-----------------------------------------------+-----------------------+
```

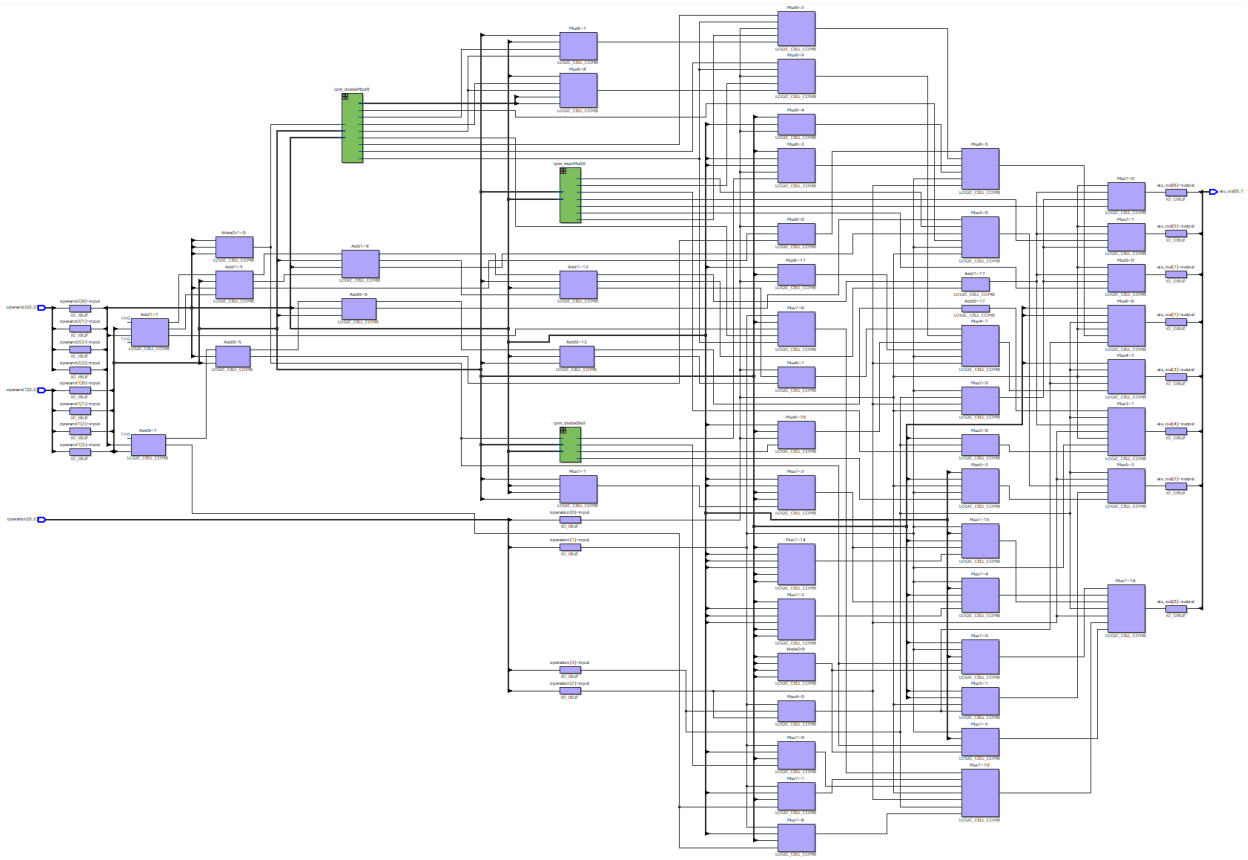
Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

alu.sv RTL viewer



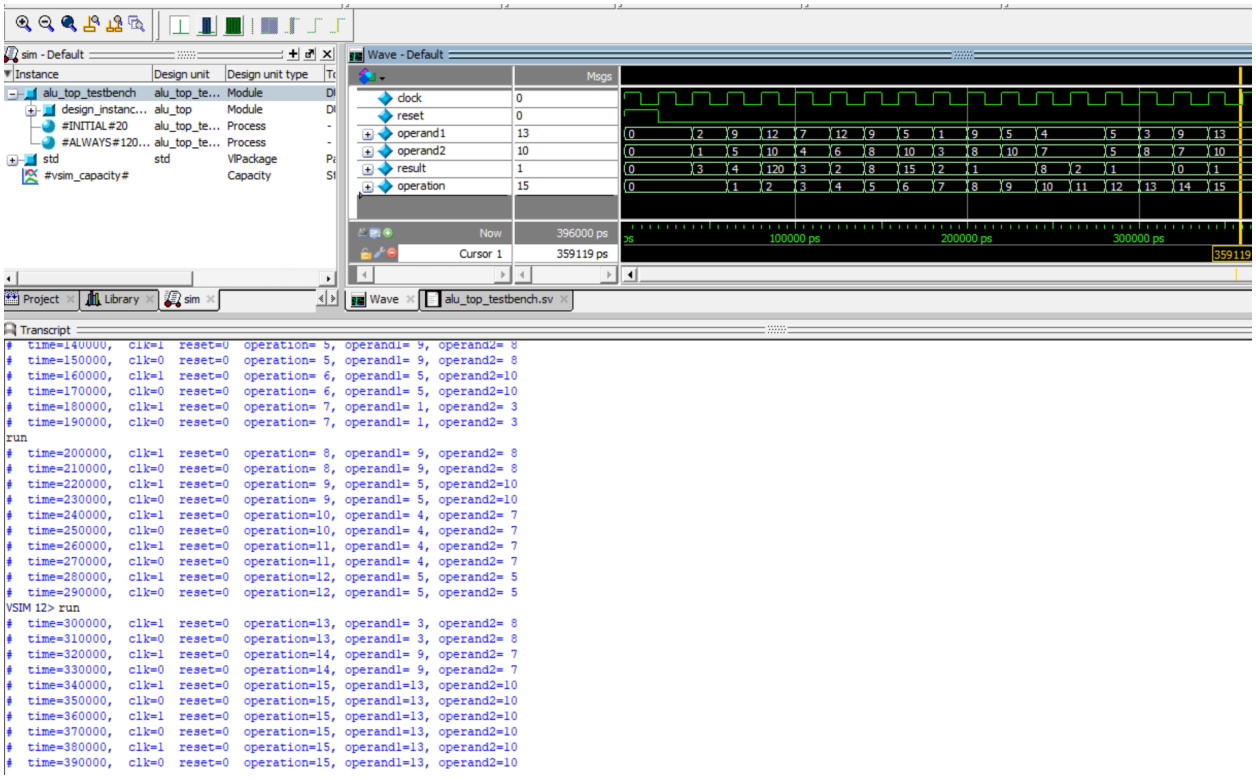
Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

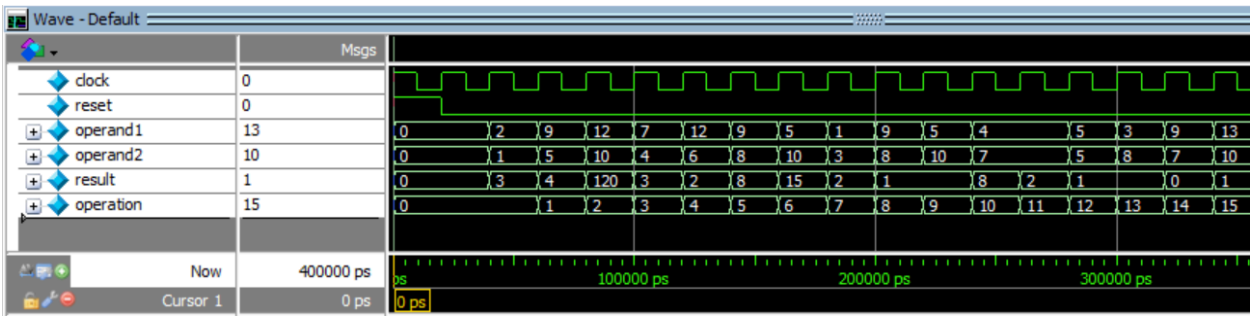
alu.sv post mapping viewer



Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

simulation waveform





simulation wavelength results explanation:

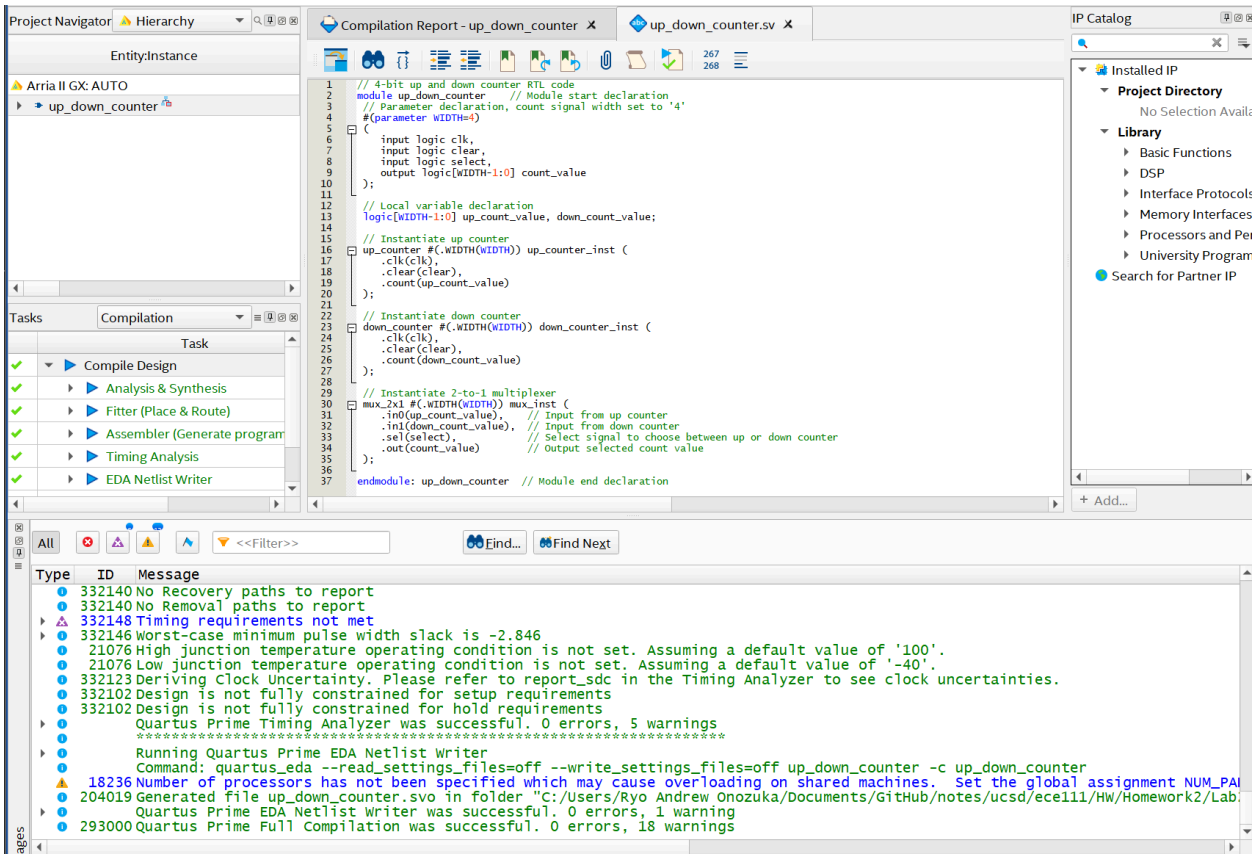
The testbench for the 4-bit ALU initializes the clock, reset, operand1, operand2, and operation signals, with the clock switching every 10ns and holding the reset high so the ALU's output can be reset every 20ns. The purpose of the testbench is to check if all of the outlined 16 operations are implemented correctly. The testbench sequentially applies different values to operand1, operand2, and the operation signal every 20ns, testing all 16 ALU operations (addition, subtraction, multiplication, etc.). A $monitor statement continuously prints the values of the clock, reset, operation, operands, and the resulting output, allowing us to observe how the ALU responds to each operation. This is why the operations go from 0 to 15 (16 total) and the results are the outcomes of each operation given the 2 numbers. For the results to be displayed correctly, I needed to make sure the wave displays were on leaf and the radix on unsigned.

Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

up_down_counter.sv

ucsd > ece111 > HW > Homework2 > Lab2 > up_down_counter > up_down_counter.sv

```sv
// 4-bit up and down counter RTL code
module up_down_counter    // Module start declaration
 // Parameter declaration, count signal width set to '4'
 #(parameter WIDTH=4)
 (
    input logic clk,
    input logic clear,
    input logic select,
    output logic[WIDTH-1:0] count_value
 );

 // Local variable declaration
 logic[WIDTH-1:0] up_count_value, down_count_value;

 // Instantiate up counter
 up_counter #(.WIDTH(WIDTH)) up_counter_inst (
    .clk(clk),
    .clear(clear),
    .count(up_count_value)
 );

 // Instantiate down counter
 down_counter #(.WIDTH(WIDTH)) down_counter_inst (
    .clk(clk),
    .clear(clear),
    .count(down_count_value)
 );

 // Instantiate 2-to-1 multiplexer
 mux_2x1 #(.WIDTH(WIDTH)) mux_inst (
    .in0(up_count_value),    // Input from up counter
    .in1(down_count_value),  // Input from down counter
    .sel(select),            // Select signal to choose between up or down counter
    .out(count_value)        // Output selected count value
 );

 endmodule: up_down_counter  // Module end declaration
```



Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

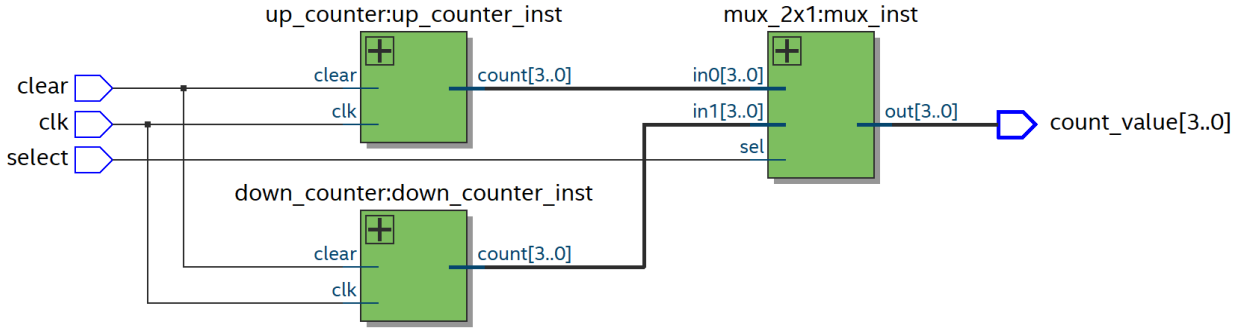
up_down_counter.sv Resource Usage Summary

ucsd > ece111 > HW > Homework2 > Lab2 > up_down_counter > up_down_counter-Resource Usage Summary.rpt

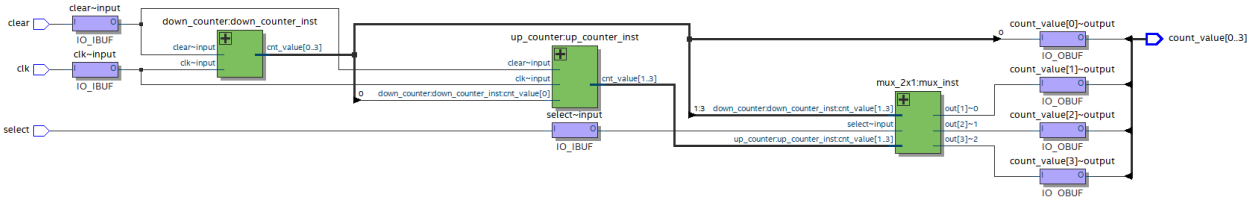
Resource	Usage
Estimated ALUTs Used	10
-- Combinational ALUTs	10
-- Memory ALUTs	0
-- LUT_REGs	0
Dedicated logic registers	7
Estimated ALUTs Unavailable	0
-- Due to unpartnered combinational logic	0
-- Due to Memory ALUTs	0
Total combinational functions	10
Combinational ALUT usage by number of inputs	
-- 7 input functions	0
-- 6 input functions	0
-- 5 input functions	0
-- 4 input functions	2
-- <=3 input functions	8
Combinational ALUTs by mode	
-- normal mode	10
-- extended LUT mode	0
-- arithmetic mode	0
-- shared arithmetic mode	0
Estimated ALUT/register pairs used	10
Total registers	7
-- Dedicated logic registers	7
-- I/O registers	0
-- LUT_REGs	0
I/O pins	7
DSP block 18-bit elements	0
Maximum fan-out node	down_counter:down_counter_inst|cnt_value[0]
Maximum fan-out	8
Total fan-out	60
Average fan-out	1.94

Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

up_down_counter.sv RTL viewer



up_down_counter.sv post mapping viewer

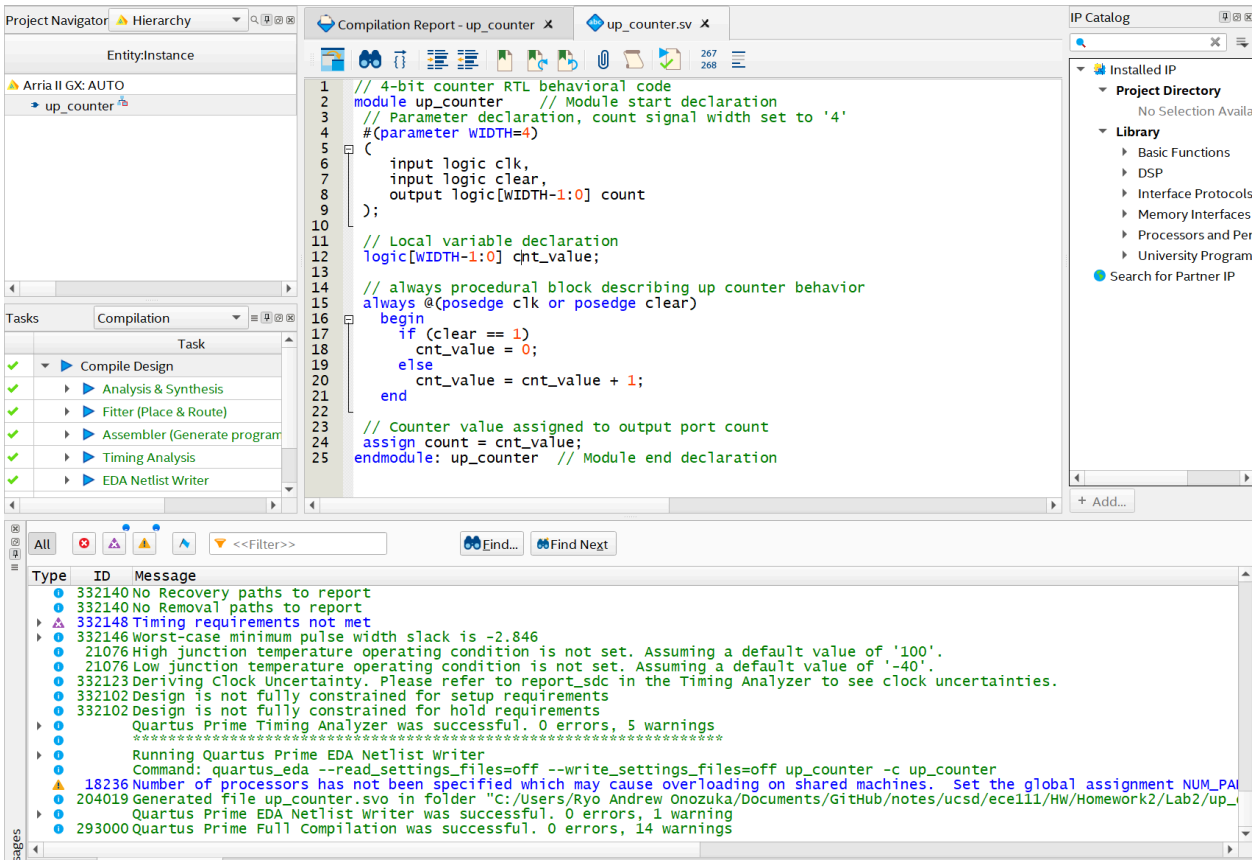


Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

up_counter.sv

ucsd > ece111 > HW > Homework2 > Lab2 > up_down_counter > up_counter.sv

```sv
// 4-bit counter RTL behavioral code
module up_counter    // Module start declaration
 // Parameter declaration, count signal width set to '4'
 #(parameter WIDTH=4)
 (
    input logic clk,
    input logic clear,
    output logic[WIDTH-1:0] count
 );

 // Local variable declaration
 logic[WIDTH-1:0] cnt_value;

 // always procedural block describing up counter behavior
 always @(posedge clk or posedge clear)
   begin
     if (clear == 1)
       cnt_value = 0;
     else
       cnt_value = cnt_value + 1;
   end

 // Counter value assigned to output port count
 assign count = cnt_value;
endmodule: up_counter  // Module end declaration
```



Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

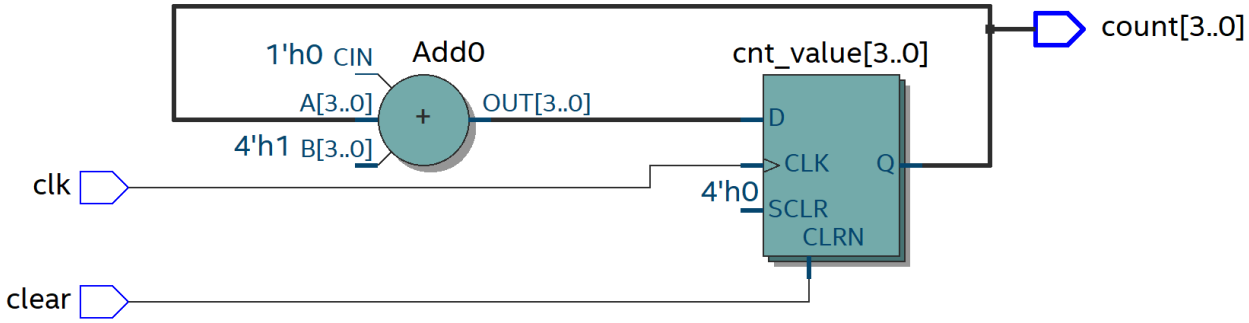
up_counter.sv Resource Usage Summary

ucsd > ece111 > HW > Homework2 > Lab2 > up_down_counter > up_counter-Resource Usage Summary.rpt

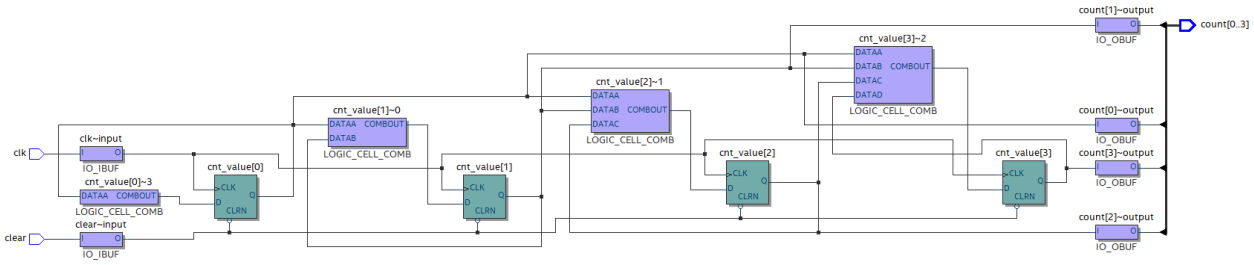
```
+---------------------------------------------+-------------+
; Resource                                    ; Usage       ;
+---------------------------------------------+-------------+
; Estimated ALUTs Used                        ; 4           ;
;     -- Combinational ALUTs                  ; 4           ;
;     -- Memory ALUTs                         ; 0           ;
;     -- LUT_REGs                             ; 0           ;
; Dedicated logic registers                   ; 4           ;
;                                             ;             ;
; Estimated ALUTs Unavailable                 ; 0           ;
;     -- Due to unpartnered combinational logic ; 0         ;
;     -- Due to Memory ALUTs                  ; 0           ;
;                                             ;             ;
; Total combinational functions               ; 4           ;
; Combinational ALUT usage by number of inputs  ;           ;
;     -- 7 input functions                    ; 0           ;
;     -- 6 input functions                    ; 0           ;
;     -- 5 input functions                    ; 0           ;
;     -- 4 input functions                    ; 1           ;
;     -- <=3 input functions                  ; 3           ;
;                                             ;             ;
; Combinational ALUTs by mode                 ;             ;
;     -- normal mode                          ; 4           ;
;     -- extended LUT mode                    ; 0           ;
;     -- arithmetic mode                      ; 0           ;
;     -- shared arithmetic mode               ; 0           ;
;                                             ;             ;
; Estimated ALUT/register pairs used          ; 4           ;
;                                             ;             ;
; Total registers                             ; 4           ;
;     -- Dedicated logic registers            ; 4           ;
;     -- I/O registers                        ; 0           ;
;     -- LUT_REGs                             ; 0           ;
;                                             ;             ;
;                                             ;             ;
; I/O pins                                    ; 6           ;
;                                             ;             ;
; DSP block 18-bit elements                   ; 0           ;
;                                             ;             ;
; Maximum fan-out node                        ; cnt_value[0] ;
; Maximum fan-out                             ; 5           ;
; Total fan-out                               ; 32          ;
; Average fan-out                             ; 1.60        ;
+---------------------------------------------+-------------+
```

Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

up_counter.sv RTL viewer



up_counter.sv post mapping viewer

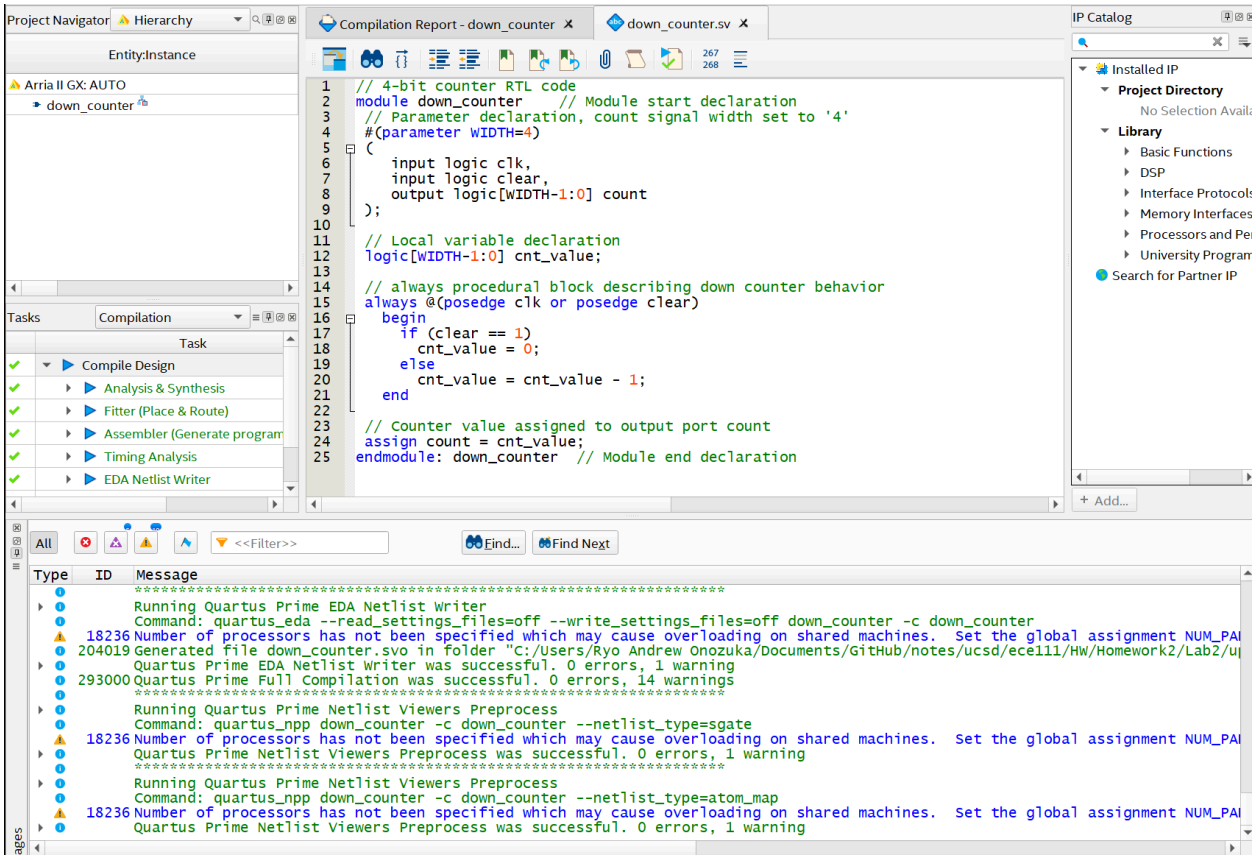


Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

down_counter.sv

ucsd > ece111 > HW > Homework2 > Lab2 > up_down_counter > down_counter.sv

```sv
// 4-bit counter RTL code
module down_counter    // Module start declaration
 // Parameter declaration, count signal width set to '4'
 #(parameter WIDTH=4)
 (
    input logic clk,
    input logic clear,
    output logic[WIDTH-1:0] count
 );

 // Local variable declaration
 logic[WIDTH-1:0] cnt_value;

 // always procedural block describing down counter behavior
 always @(posedge clk or posedge clear)
   begin
     if (clear == 1)
       cnt_value = 0;
     else
       cnt_value = cnt_value - 1;
   end

 // Counter value assigned to output port count
 assign count = cnt_value;
endmodule: down_counter  // Module end declaration
```



Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

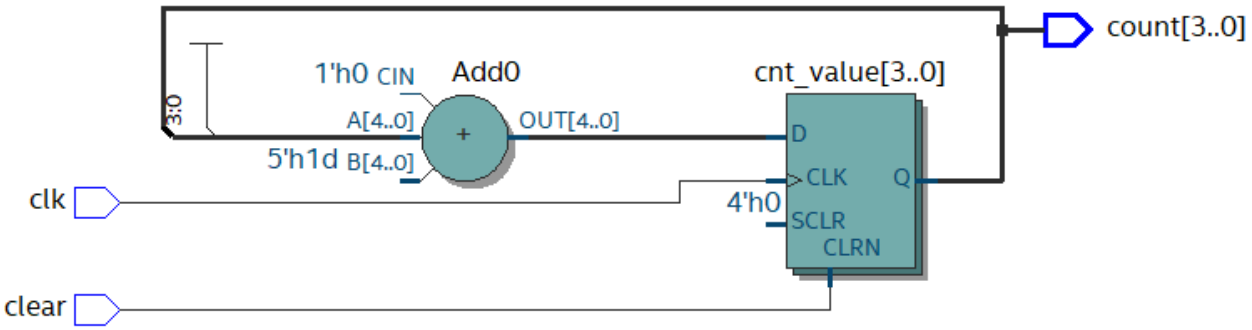
down_counter.sv Resource Usage Summary

ucsd > ece111 > HW > Homework2 > Lab2 > up_down_counter > down_counter-Resource Usage Summary.rpt

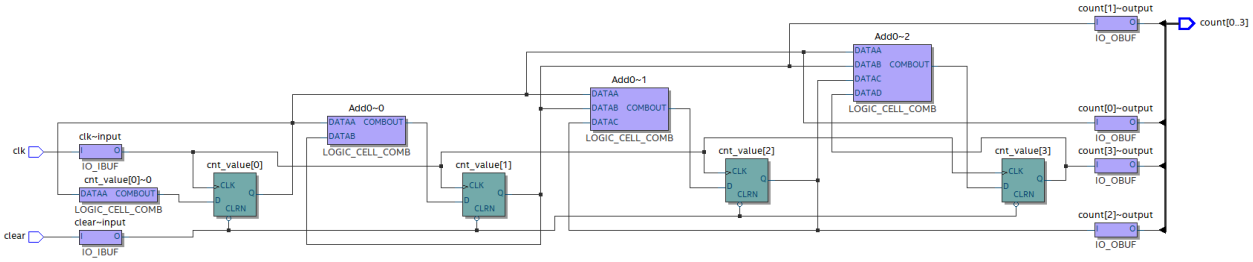
```
+-----------------------------------------------------------------------+
; Analysis & Synthesis Resource Usage Summary                           ;
+-----------------------------------------------+-----------------------+
; Resource                                      ; Usage                 ;
+-----------------------------------------------+-----------------------+
; Estimated ALUTs Used                          ; 4                     ;
;     -- Combinational ALUTs                    ; 4                     ;
;     -- Memory ALUTs                          ; 0                     ;
;     -- LUT_REGs                              ; 0                     ;
; Dedicated logic registers                     ; 4                     ;
;                                               ;                       ;
; Estimated ALUTs Unavailable                   ; 0                     ;
;     -- Due to unpartnered combinational logic ; 0                     ;
;     -- Due to Memory ALUTs                    ; 0                     ;
;                                               ;                       ;
; Total combinational functions                 ; 4                     ;
; Combinational ALUT usage by number of inputs  ;                       ;
;     -- 7 input functions                      ; 0                     ;
;     -- 6 input functions                      ; 0                     ;
;     -- 5 input functions                      ; 0                     ;
;     -- 4 input functions                      ; 1                     ;
;     -- <=3 input functions                    ; 3                     ;
;                                               ;                       ;
; Combinational ALUTs by mode                   ;                       ;
;     -- normal mode                            ; 4                     ;
;     -- extended LUT mode                      ; 0                     ;
;     -- arithmetic mode                        ; 0                     ;
;     -- shared arithmetic mode                 ; 0                     ;
;                                               ;                       ;
; Estimated ALUT/register pairs used            ; 4                     ;
;                                               ;                       ;
; Total registers                               ; 4                     ;
;     -- Dedicated logic registers              ; 4                     ;
;     -- I/O registers                          ; 0                     ;
;     -- LUT_REGs                              ; 0                     ;
;                                               ;                       ;
;                                               ;                       ;
; I/O pins                                      ; 6                     ;
;                                               ;                       ;
; DSP block 18-bit elements                     ; 0                     ;
;                                               ;                       ;
; Maximum fan-out node                          ; cnt_value[0]          ;
; Maximum fan-out                               ; 5                     ;
; Total fan-out                                 ; 32                    ;
; Average fan-out                               ; 1.60                  ;
+-----------------------------------------------+-----------------------+
```

Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

down_counter.sv RTL viewer



down_counter.sv post mapping viewer

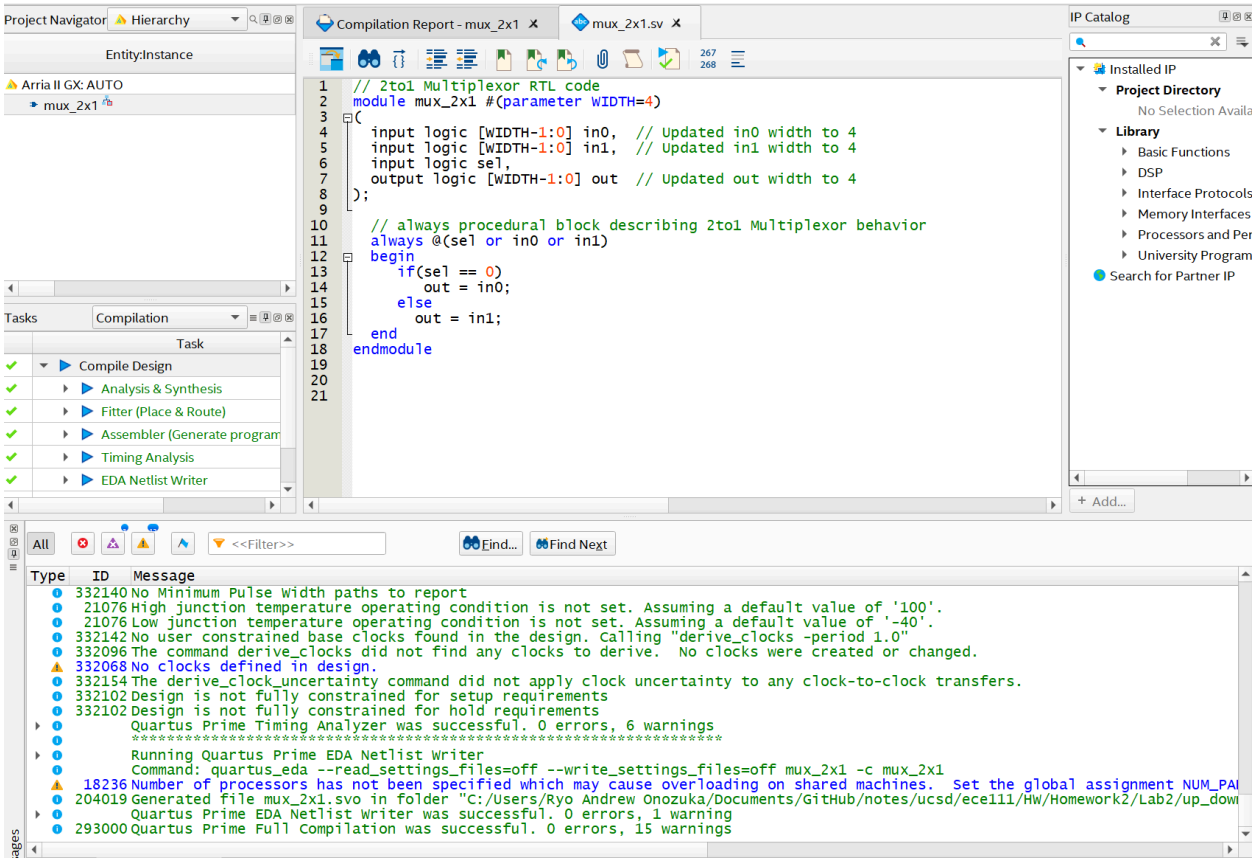


Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

mux_2x1.sv

ucsd > ece111 > HW > Homework2 > Lab2 > up_down_counter > mux_2x1.sv

```sv
// 2to1 Multiplexor RTL code
module mux_2x1 #(parameter WIDTH=4)
(
  input logic [WIDTH-1:0] in0,  // Updated in0 width to 4
  input logic [WIDTH-1:0] in1,  // Updated in1 width to 4
  input logic sel,
  output logic [WIDTH-1:0] out  // Updated out width to 4
);

  // always procedural block describing 2to1 Multiplexor behavior
  always @(sel or in0 or in1)
  begin
    if(sel == 0)
      out = in0;
    else
      out = in1;
  end
endmodule
```



Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

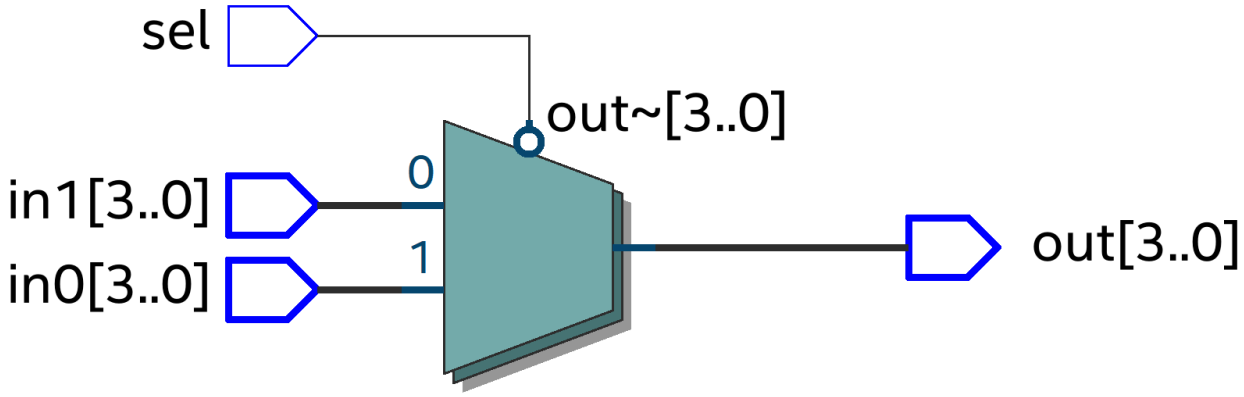
mux_2x1.sv Resource Usage Summary

ucsd > ece111 > HW > Homework2 > Lab2 > up_down_counter > mux_2x1-Resource Usage Summary.rpt

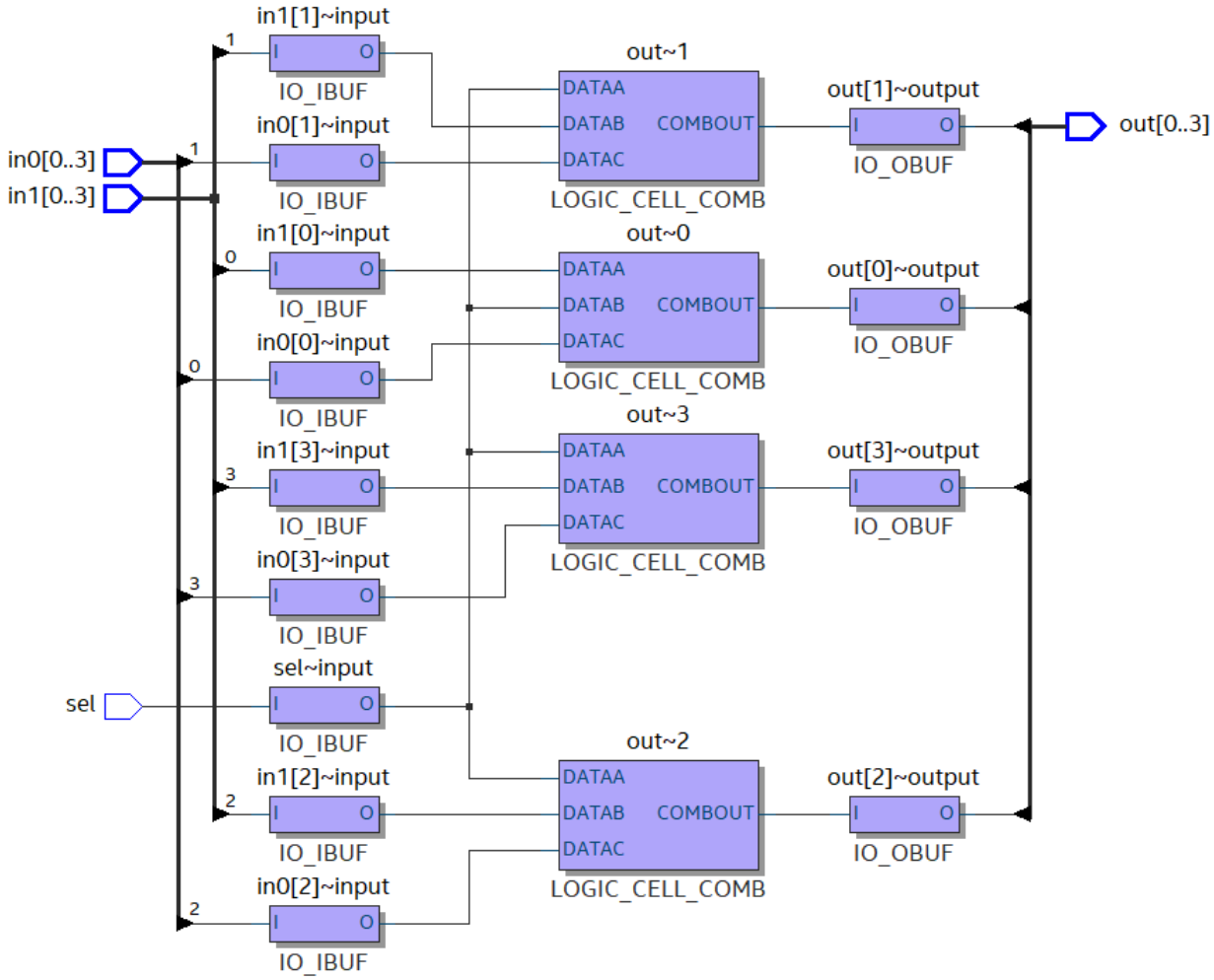
```
+-----------------------------------------------------------+
; Analysis & Synthesis Resource Usage Summary               ;
+-----------------------------------------------+-----------+
; Resource                                      ; Usage     ;
+-----------------------------------------------+-----------+
; Estimated ALUTs Used                          ; 4         ;
;     -- Combinational ALUTs                    ; 4         ;
;     -- Memory ALUTs                          ; 0         ;
;     -- LUT_REGs                              ; 0         ;
; Dedicated logic registers                     ; 0         ;
;                                               ;           ;
; Estimated ALUTs Unavailable                   ; 0         ;
;     -- Due to unpartnered combinational logic ; 0         ;
;     -- Due to Memory ALUTs                    ; 0         ;
;                                               ;           ;
; Total combinational functions                 ; 4         ;
; Combinational ALUT usage by number of inputs  ;           ;
;     -- 7 input functions                      ; 0         ;
;     -- 6 input functions                      ; 0         ;
;     -- 5 input functions                      ; 0         ;
;     -- 4 input functions                      ; 0         ;
;     -- <=3 input functions                    ; 4         ;
;                                               ;           ;
; Combinational ALUTs by mode                   ;           ;
;     -- normal mode                            ; 4         ;
;     -- extended LUT mode                      ; 0         ;
;     -- arithmetic mode                        ; 0         ;
;     -- shared arithmetic mode                 ; 0         ;
;                                               ;           ;
; Estimated ALUT/register pairs used            ; 4         ;
;                                               ;           ;
; Total registers                               ; 0         ;
;     -- Dedicated logic registers              ; 0         ;
;     -- I/O registers                          ; 0         ;
;     -- LUT_REGs                              ; 0         ;
;                                               ;           ;
;                                               ;           ;
; I/O pins                                      ; 13        ;
;                                               ;           ;
; DSP block 18-bit elements                     ; 0         ;
;                                               ;           ;
; Maximum fan-out node                          ; sel~input ;
; Maximum fan-out                               ; 4         ;
; Total fan-out                                 ; 29        ;
; Average fan-out                               ; 0.97      ;
+-----------------------------------------------+-----------+
```

Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

mux_2x1.sv RTL viewer

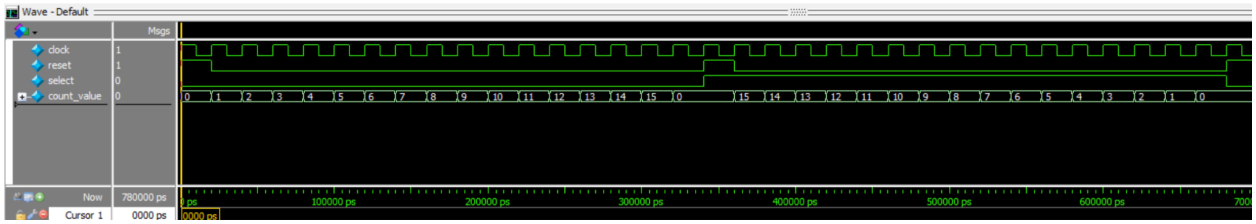


mux_2x1.sv post mapping viewer



Andrew Onozuka A16760043
ECE 111 HW2 | 10/16/2024

simulation waveform



simulation wavelength results explanation:

The testbench for the 4-bit up-down counter initializes the clock, reset, and select signals, with the clock toggling every 10ns. The reset is initially held high for 20ns to ensure the counter is reset. The purpose of the testbench is to check if the up-down counter correctly counts up or down based on the value of the select signal. The testbench starts with select = 0, causing the counter to count upwards from 0 to 15 after reset is deasserted at 20ns. After running for 320ns, the reset is asserted again, and the select signal is switched to 1, making the counter count down from 15 to 0. The simulation repeats this process to test both the up and down counting modes. A $monitor statement continuously prints the current time, the values of the reset, clock, and the count_value output, allowing us to observe how the counter responds to the changes in the select and reset signals.