PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, BSK III Stage, Bengaluru-560 085

Department of Electronics and Communication Engineering

Course Title: Digital System Design
Course Code: UE21EC342AB1

Teacher: Rashmi Seethur

Project Title: 8 Input Bitonic Sorter Design

Done By:
Akshay Sharma G – PES1UG21EC031
Akshay Anand – PES1UG21EC029
Akash Ravi Bhat – PES1UG21EC025
Akash Giridhar – PES1UG21EC023

AIM

To write code and test bench for an 8 bit Bitonic Sorter and to test the sorter for 4 inputs and 8 inputs

INTRODUCTION

- **Definition**: A bitonic sorter is a parallel sorting algorithm designed for hardware implementation.

- **Bitonic Pattern**: The term "bitonic" refers to the sorting network's pattern, starting with a monotonic sequence (ascending or descending) followed by a bitonic sequence (alternating ascending and descending).

- **8-Bit Operation**: Specifically designed for 8-bit data elements.

Sorting Process:
1. Recursively build a bitonic sequence from the input data.
2. Sort the bitonic sequence until the entire sequence is sorted.

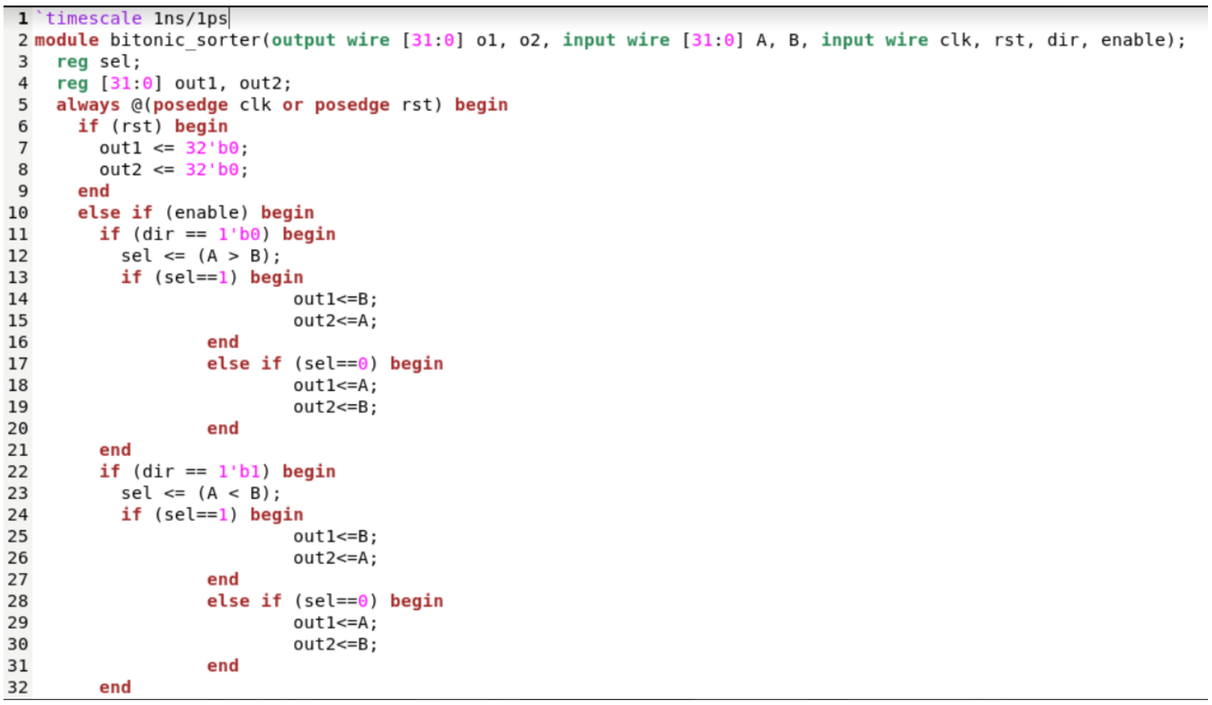
- **Parallel Processing**: The sorting network can be structured to enable parallel processing.

- **Hardware Implementation**: Well-suited for implementation in hardware architectures like parallel processors or digital circuits.

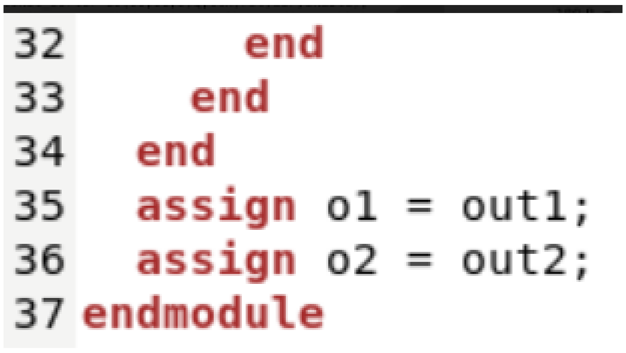
- **Application**: Particularly useful in scenarios where efficient parallel sorting of 8-bit data is needed, such as in digital signal processing.

DESIGN FILES :

1.Compare and Exchange block (CAE)

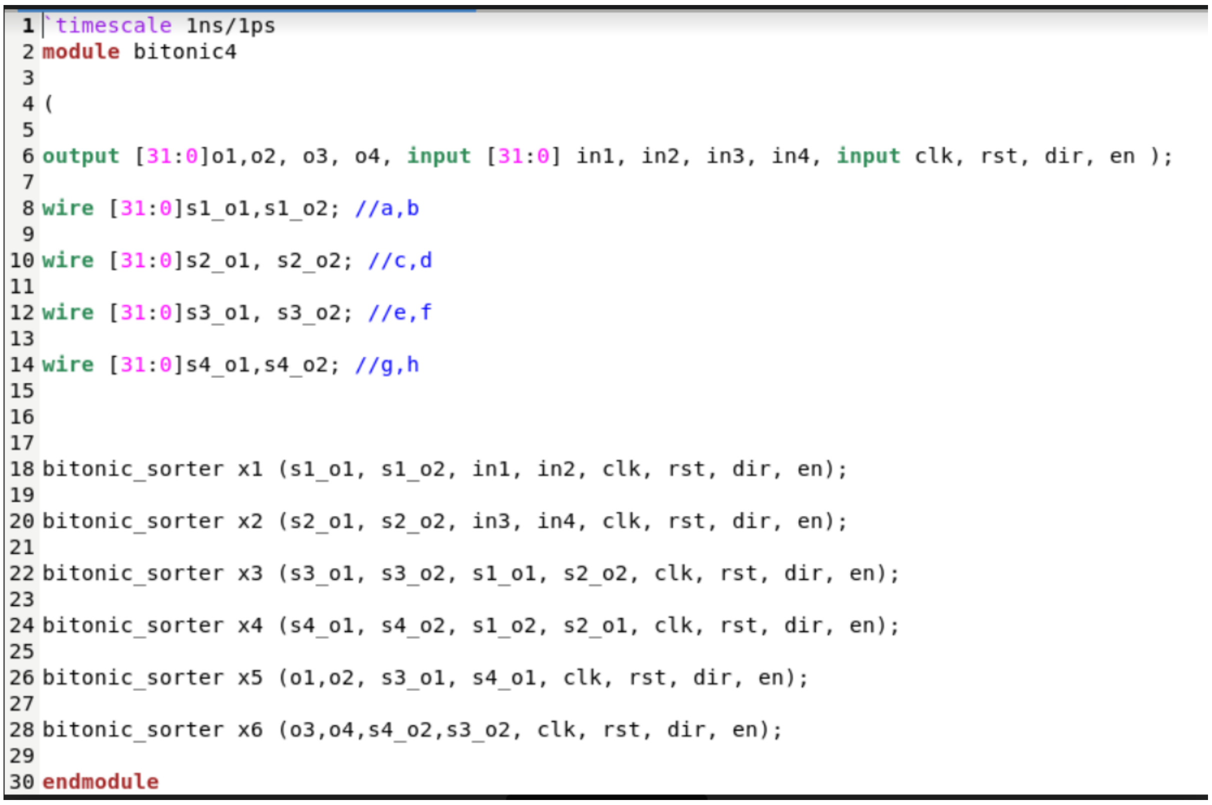


```verilog
`timescale 1ns/1ps
module bitonic_sorter(output wire [31:0] o1, o2, input wire [31:0] A, B, input wire clk, rst, dir, enable);
  reg sel;
  reg [31:0] out1, out2;
  always @(posedge clk or posedge rst) begin
    if (rst) begin
      out1 <= 32'b0;
      out2 <= 32'b0;
    end
    else if (enable) begin
      if (dir == 1'b0) begin
        sel <= (A > B);
        if (sel==1) begin
                        out1<=B;
                        out2<=A;
                end
                else if (sel==0) begin
                        out1<=A;
                        out2<=B;
                end
      end
      if (dir == 1'b1) begin
        sel <= (A < B);
        if (sel==1) begin
                        out1<=B;
                        out2<=A;
                end
                else if (sel==0) begin
                        out1<=A;
                        out2<=B;
                end
      end
```



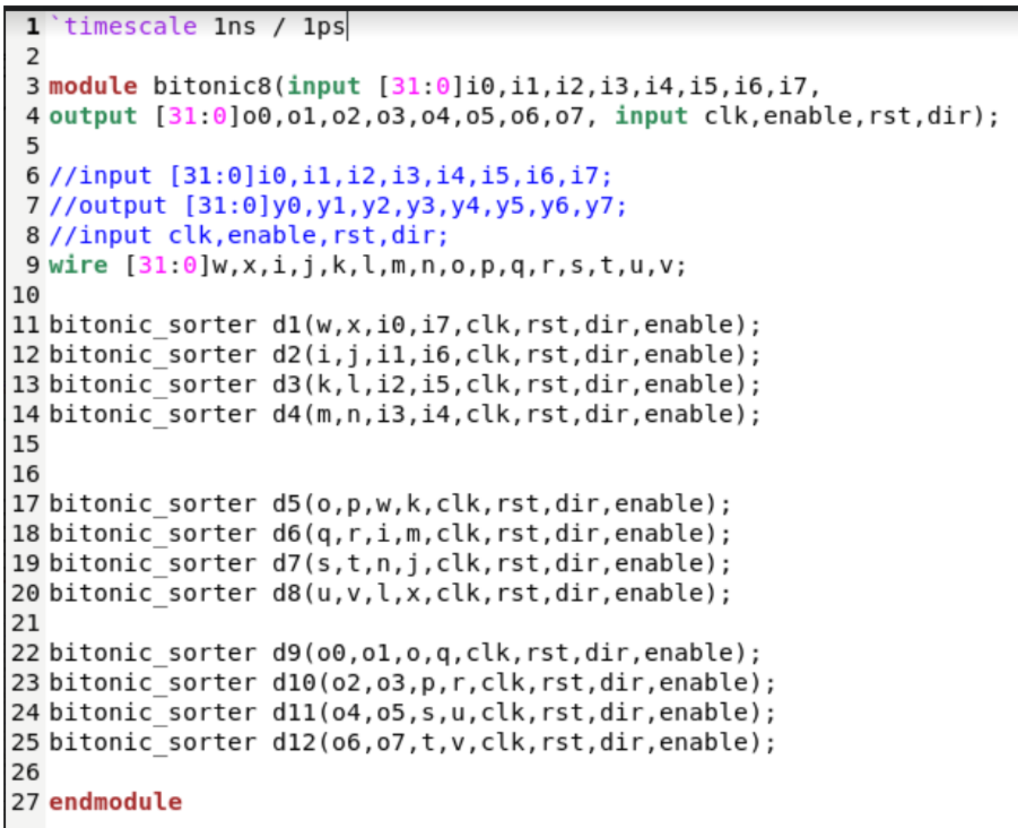
```verilog
      end
    end
  end
  assign o1 = out1;
  assign o2 = out2;
endmodule
```

2. 4 Input Bitonic Sorting



```verilog
`timescale 1ns/1ps
module bitonic4

(

output [31:0]o1,o2, o3, o4, input [31:0] in1, in2, in3, in4, input clk, rst, dir, en );

wire [31:0]s1_o1,s1_o2; //a,b

wire [31:0]s2_o1, s2_o2; //c,d

wire [31:0]s3_o1, s3_o2; //e,f

wire [31:0]s4_o1,s4_o2; //g,h



bitonic_sorter x1 (s1_o1, s1_o2, in1, in2, clk, rst, dir, en);

bitonic_sorter x2 (s2_o1, s2_o2, in3, in4, clk, rst, dir, en);

bitonic_sorter x3 (s3_o1, s3_o2, s1_o1, s2_o2, clk, rst, dir, en);

bitonic_sorter x4 (s4_o1, s4_o2, s1_o2, s2_o1, clk, rst, dir, en);

bitonic_sorter x5 (o1,o2, s3_o1, s4_o1, clk, rst, dir, en);

bitonic_sorter x6 (o3,o4,s4_o2,s3_o2, clk, rst, dir, en);

endmodule
```

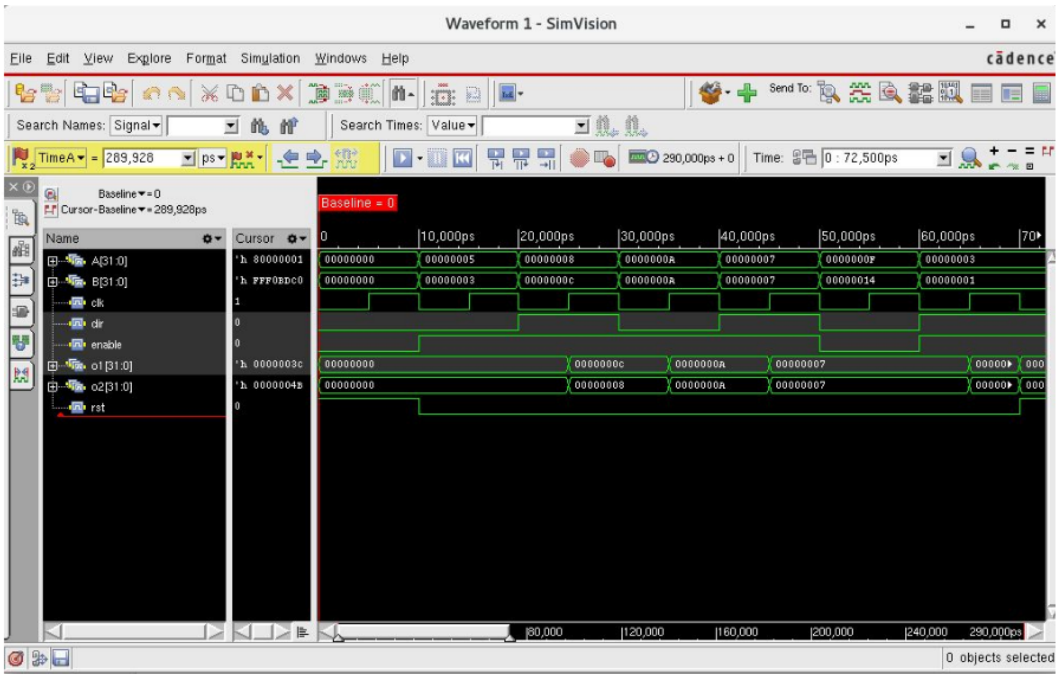
3. **8 Input Bitonic Sorting**



```verilog
`timescale 1ns / 1ps

module bitonic8(input [31:0]i0,i1,i2,i3,i4,i5,i6,i7,
output [31:0]o0,o1,o2,o3,o4,o5,o6,o7, input clk,enable,rst,dir);

//input [31:0]i0,i1,i2,i3,i4,i5,i6,i7;
//output [31:0]y0,y1,y2,y3,y4,y5,y6,y7;
//input clk,enable,rst,dir;
wire [31:0]w,x,i,j,k,l,m,n,o,p,q,r,s,t,u,v;

bitonic_sorter d1(w,x,i0,i7,clk,rst,dir,enable);
bitonic_sorter d2(i,j,i1,i6,clk,rst,dir,enable);
bitonic_sorter d3(k,l,i2,i5,clk,rst,dir,enable);
bitonic_sorter d4(m,n,i3,i4,clk,rst,dir,enable);


bitonic_sorter d5(o,p,w,k,clk,rst,dir,enable);
bitonic_sorter d6(q,r,i,m,clk,rst,dir,enable);
bitonic_sorter d7(s,t,n,j,clk,rst,dir,enable);
bitonic_sorter d8(u,v,l,x,clk,rst,dir,enable);

bitonic_sorter d9(o0,o1,o,q,clk,rst,dir,enable);
bitonic_sorter d10(o2,o3,p,r,clk,rst,dir,enable);
bitonic_sorter d11(o4,o5,s,u,clk,rst,dir,enable);
bitonic_sorter d12(o6,o7,t,v,clk,rst,dir,enable);

endmodule
```

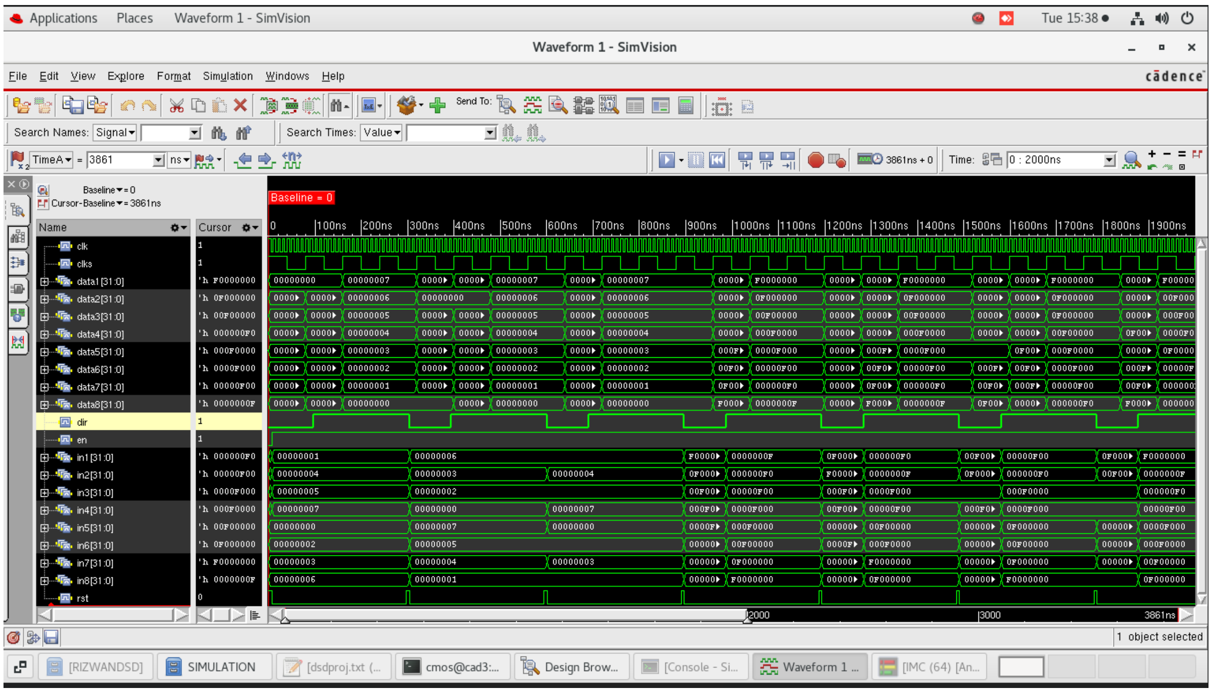
WAVEFORMS:

1.CAE block



```
Name        Cursor
A[31:0]     'h 80000001
B[31:0]     'h FFF0EDC0
clk         1
dir         0
enable      0
o1[31:0]    'h 0000003C
o2[31:0]    'h 0000004B
rst         0
```

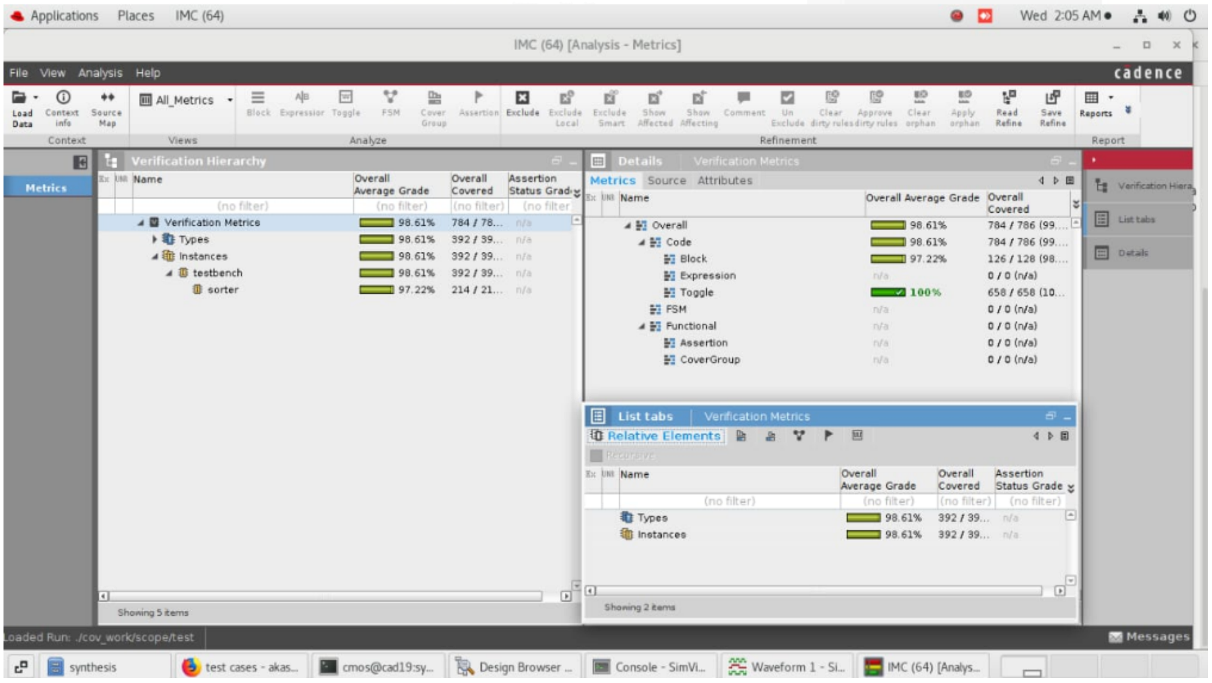
2. 8 input bitonic sorter:



```
Name          Cursor
clk           1
clks          1
data1[31:0]   'h F0000000
data2[31:0]   'h 0F000000
data3[31:0]   'h 00F00000
data4[31:0]   'h 000000F0
data5[31:0]   'h 000F0000
data6[31:0]   'h 0000F000
data7[31:0]   'h 00000F00
data8[31:0]   'h 0000000F
dir           1
en            1
in1[31:0]     'h 000000F0
in2[31:0]     'h 00000F00
in3[31:0]     'h 0000F000
in4[31:0]     'h 000F0000
in5[31:0]     'h 00F00000
in6[31:0]     'h 0F000000
in7[31:0]     'h F0000000
in8[31:0]     'h 0000000F
rst           0
```

COVERAGE REPORT:

1.CAE block



IMC (64) [Analysis - Metrics]

Loaded Run: ./cov_work/scope/test

Verification Hierarchy

Name	Overall Average Grade	Overall Covered	Assertion Status Grade
(no filter)	(no filter)	(no filter)	(no filter
Verification Metrice	98.61%	784 / 78...	n/a
Types	98.61%	392 / 39...	n/a
Instances	98.61%	392 / 39...	n/a
testbench	98.61%	392 / 39...	n/a
sorter	97.22%	214 / 21...	n/a

Showing 5 items

Details — Verification Metrics

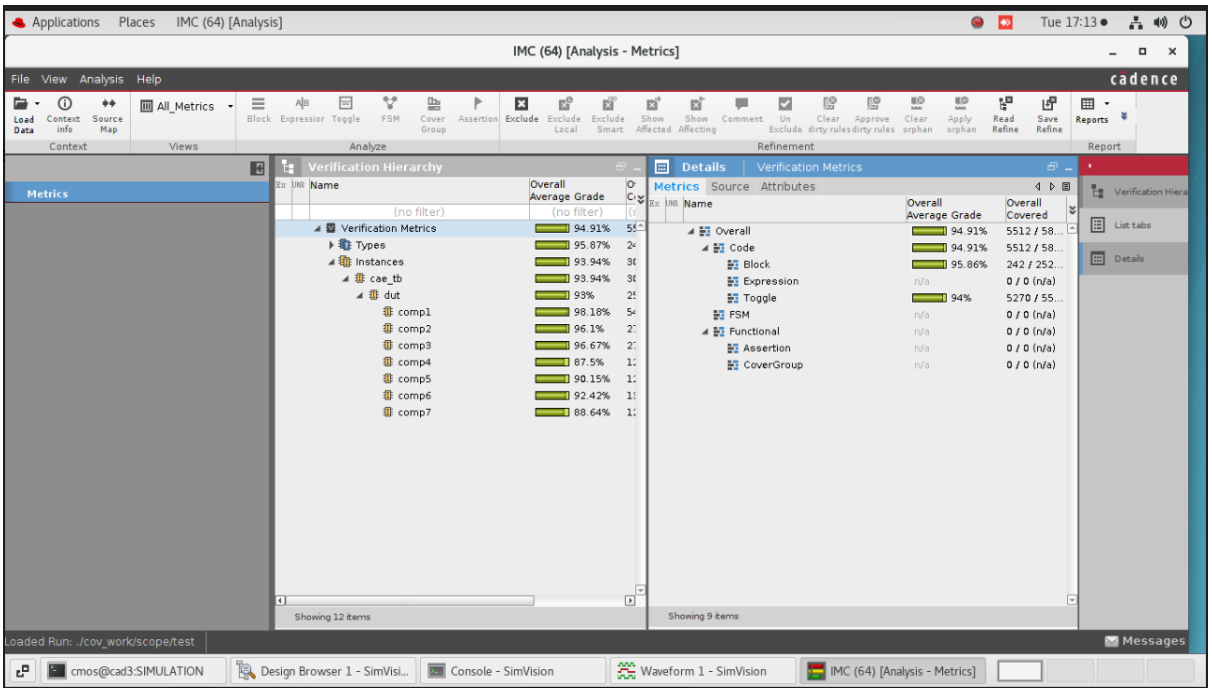
Name	Overall Average Grade	Overall Covered
Overall	98.61%	784 / 786 (99....
Code	98.61%	784 / 786 (99....
Block	97.22%	126 / 128 (98....
Expression	n/a	0 / 0 (n/a)
Toggle	100%	658 / 658 (10...
FSM	n/a	0 / 0 (n/a)
Functional	n/a	0 / 0 (n/a)
Assertion	n/a	0 / 0 (n/a)
CoverGroup	n/a	0 / 0 (n/a)

List tabs — Verification Metrics

Name	Overall Average Grade	Overall Covered	Assertion Status Grade
(no filter)	(no filter)	(no filter)	(no filter)
Types	98.61%	392 / 39...	n/a
Instances	98.61%	392 / 39...	n/a

Showing 2 items

2. 8 input bitonic sorter:



IMC (64) [Analysis - Metrics]

Loaded Run: ./cov_work/scope/test

Verification Hierarchy

Name	Overall Average Grade	O C
(no filter)	(no filter)	(r
Verification Metrics	94.91%	55
Types	95.87%	24
Instances	93.94%	30
cae_tb	93.94%	30
dut	93%	25
comp1	98.18%	54
comp2	96.1%	27
comp3	96.67%	27
comp4	87.5%	12
comp5	90.15%	12
comp6	92.42%	13
comp7	88.64%	12

Showing 12 items

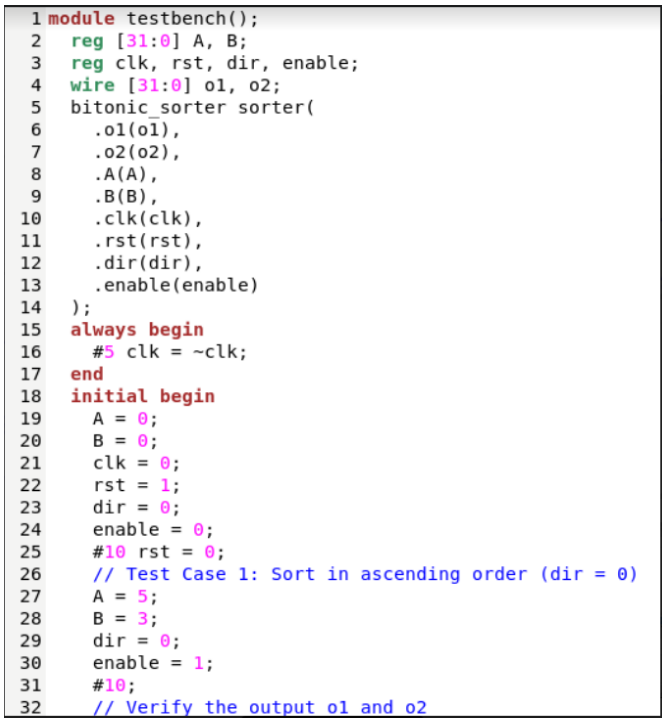
Details — Verification Metrics

Name	Overall Average Grade	Overall Covered
Overall	94.91%	5512 / 58...
Code	94.91%	5512 / 58...
Block	95.86%	242 / 252...
Expression	n/a	0 / 0 (n/a)
Toggle	94%	5270 / 55...
FSM	n/a	0 / 0 (n/a)
Functional	n/a	0 / 0 (n/a)
Assertion	n/a	0 / 0 (n/a)
CoverGroup	n/a	0 / 0 (n/a)

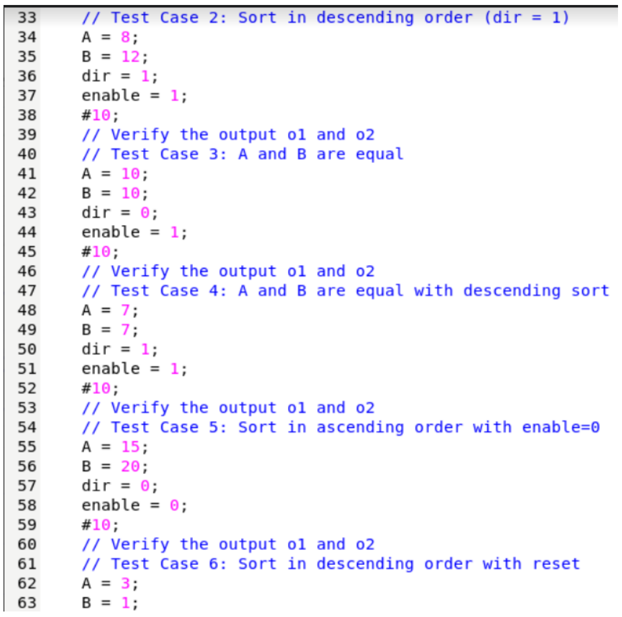
Showing 9 items

TESTBENCH:

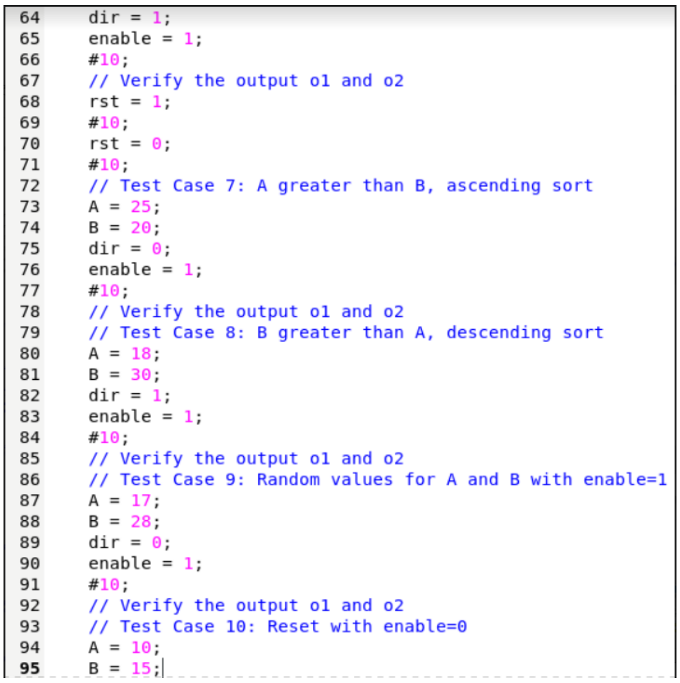
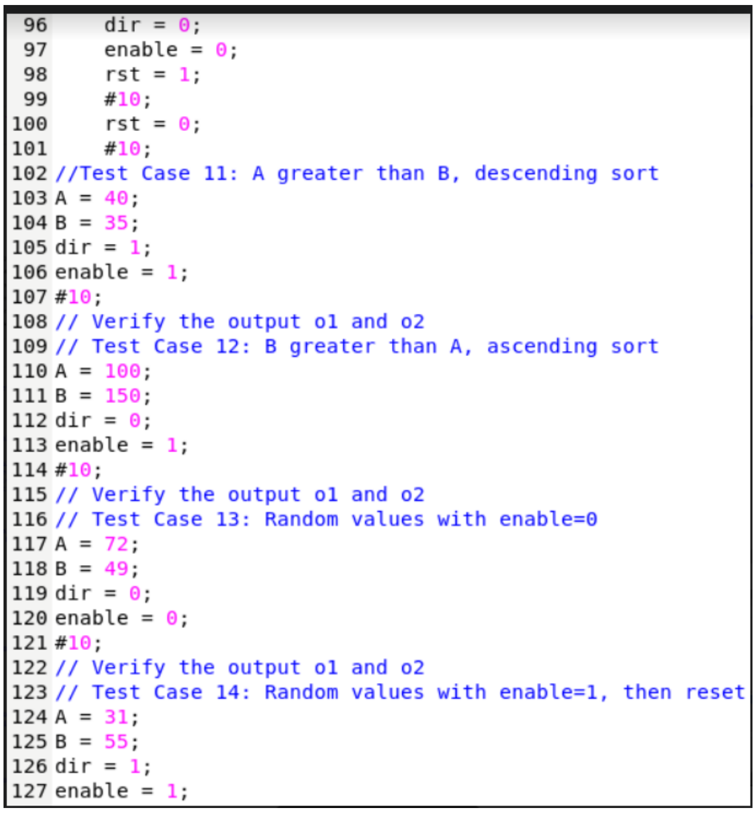
1. CAE block

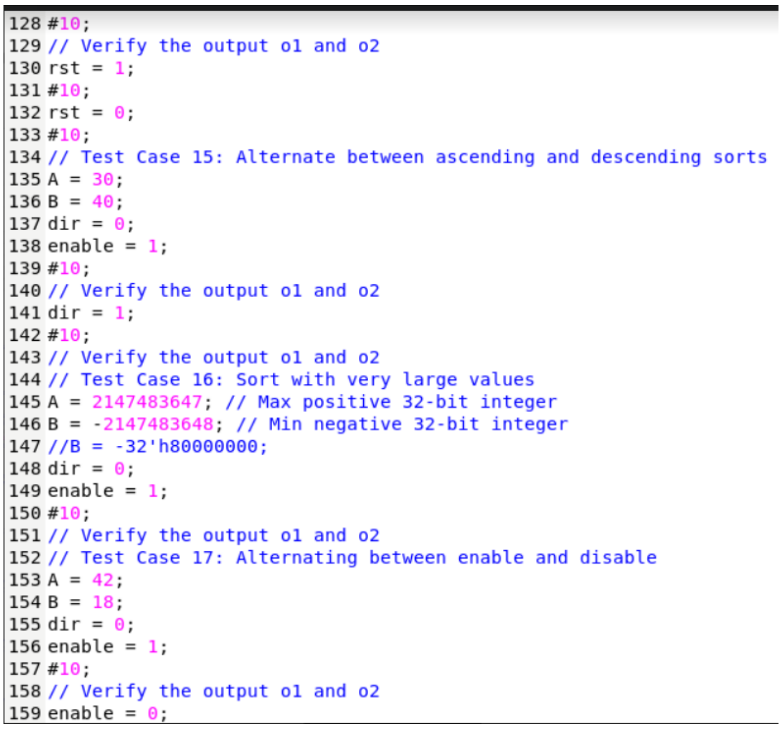


```verilog
module testbench();
  reg [31:0] A, B;
  reg clk, rst, dir, enable;
  wire [31:0] o1, o2;
  bitonic_sorter sorter(
    .o1(o1),
    .o2(o2),
    .A(A),
    .B(B),
    .clk(clk),
    .rst(rst),
    .dir(dir),
    .enable(enable)
  );
  always begin
    #5 clk = ~clk;
  end
  initial begin
    A = 0;
    B = 0;
    clk = 0;
    rst = 1;
    dir = 0;
    enable = 0;
    #10 rst = 0;
    // Test Case 1: Sort in ascending order (dir = 0)
    A = 5;
    B = 3;
    dir = 0;
    enable = 1;
    #10;
    // Verify the output o1 and o2
```

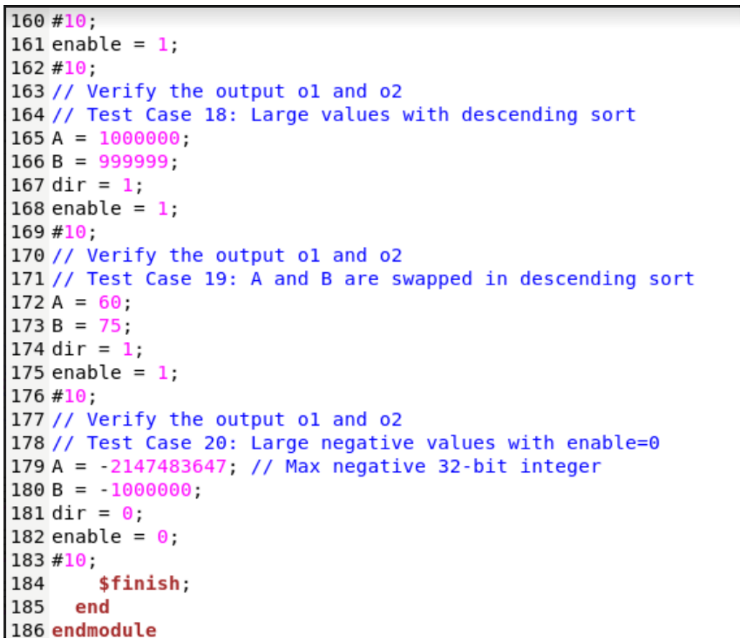


```verilog
    // Test Case 2: Sort in descending order (dir = 1)
    A = 8;
    B = 12;
    dir = 1;
    enable = 1;
    #10;
    // Verify the output o1 and o2
    // Test Case 3: A and B are equal
    A = 10;
    B = 10;
    dir = 0;
    enable = 1;
    #10;
    // Verify the output o1 and o2
    // Test Case 4: A and B are equal with descending sort
    A = 7;
    B = 7;
    dir = 1;
    enable = 1;
    #10;
    // Verify the output o1 and o2
    // Test Case 5: Sort in ascending order with enable=0
    A = 15;
    B = 20;
    dir = 0;
    enable = 0;
    #10;
    // Verify the output o1 and o2
    // Test Case 6: Sort in descending order with reset
    A = 3;
    B = 1;
```


```
    dir = 1;
    enable = 1;
    #10;
    // Verify the output o1 and o2
    rst = 1;
    #10;
    rst = 0;
    #10;
    // Test Case 7: A greater than B, ascending sort
    A = 25;
    B = 20;
    dir = 0;
    enable = 1;
    #10;
    // Verify the output o1 and o2
    // Test Case 8: B greater than A, descending sort
    A = 18;
    B = 30;
    dir = 1;
    enable = 1;
    #10;
    // Verify the output o1 and o2
    // Test Case 9: Random values for A and B with enable=1
    A = 17;
    B = 28;
    dir = 0;
    enable = 1;
    #10;
    // Verify the output o1 and o2
    // Test Case 10: Reset with enable=0
    A = 10;
    B = 15;
```

```
    dir = 0;
    enable = 0;
    rst = 1;
    #10;
    rst = 0;
    #10;
//Test Case 11: A greater than B, descending sort
A = 40;
B = 35;
dir = 1;
enable = 1;
#10;
// Verify the output o1 and o2
// Test Case 12: B greater than A, ascending sort
A = 100;
B = 150;
dir = 0;
enable = 1;
#10;
// Verify the output o1 and o2
// Test Case 13: Random values with enable=0
A = 72;
B = 49;
dir = 0;
enable = 0;
#10;
// Verify the output o1 and o2
// Test Case 14: Random values with enable=1, then reset
A = 31;
B = 55;
dir = 1;
enable = 1;
```



```
#10;
// Verify the output o1 and o2
rst = 1;
#10;
rst = 0;
#10;
// Test Case 15: Alternate between ascending and descending sorts
A = 30;
B = 40;
dir = 0;
enable = 1;
#10;
// Verify the output o1 and o2
dir = 1;
#10;
// Verify the output o1 and o2
// Test Case 16: Sort with very large values
A = 2147483647; // Max positive 32-bit integer
B = -2147483648; // Min negative 32-bit integer
//B = -32'h80000000;
dir = 0;
enable = 1;
#10;
// Verify the output o1 and o2
// Test Case 17: Alternating between enable and disable
A = 42;
B = 18;
dir = 0;
enable = 1;
#10;
// Verify the output o1 and o2
enable = 0;
```

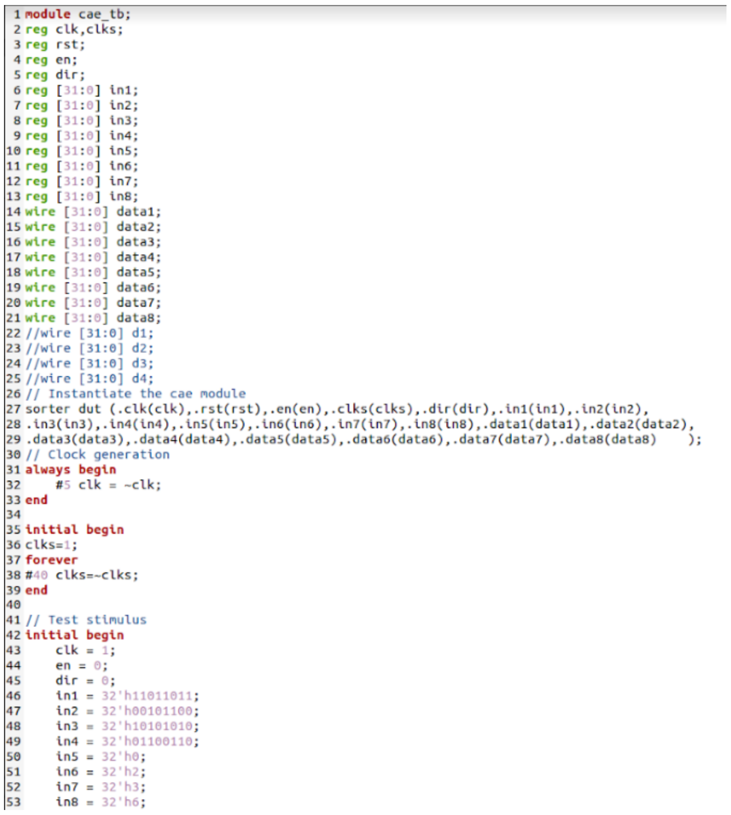


```
#10;
enable = 1;
#10;
// Verify the output o1 and o2
// Test Case 18: Large values with descending sort
A = 1000000;
B = 999999;
dir = 1;
enable = 1;
#10;
// Verify the output o1 and o2
// Test Case 19: A and B are swapped in descending sort
A = 60;
B = 75;
dir = 1;
enable = 1;
#10;
// Verify the output o1 and o2
// Test Case 20: Large negative values with enable=0
A = -2147483647; // Max negative 32-bit integer
B = -1000000;
dir = 0;
enable = 0;
#10;
    $finish;
  end
endmodule
```

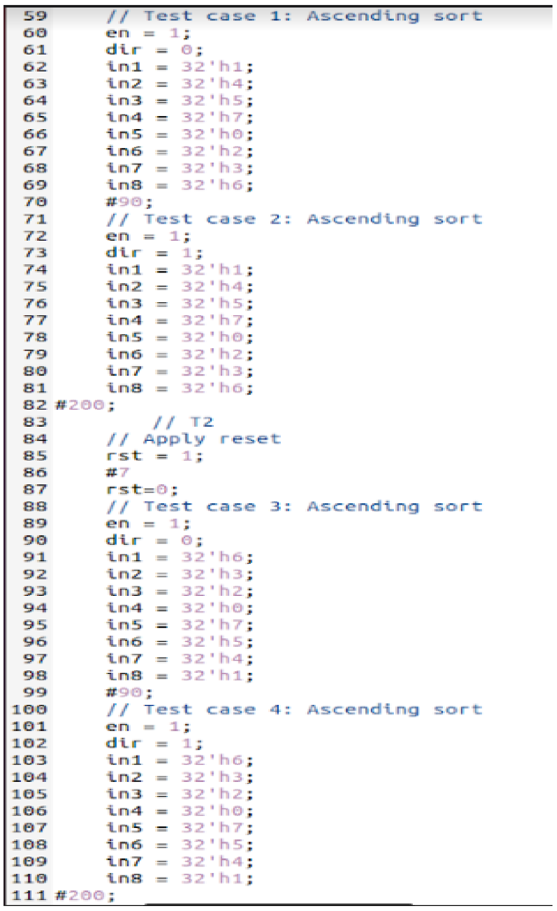
Test case 1 : The test verifies the output o1 and o2 after sorting the inputs 5 and 3 in ascending order.
Test case 2 : The test verifies the output o1 and o2 after sorting the inputs 8 and 12 in descending order.
Test case 3 : The test verifies the output o1 and o2 after sorting equal inputs in ascending order.
Test case 4 : The test verifies the output o1 and o2 after sorting equal inputs in descending order.
Test case 5 : The test verifies the output o1 and o2 without

performing sorting when enable is set to 0.
Similarly, many more test cases have been added to get a high coverage value

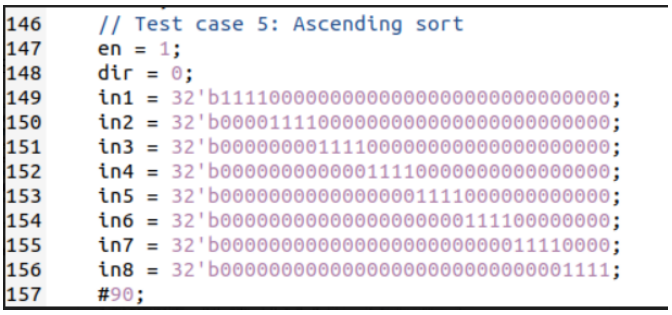
2. 8 input Test Bench



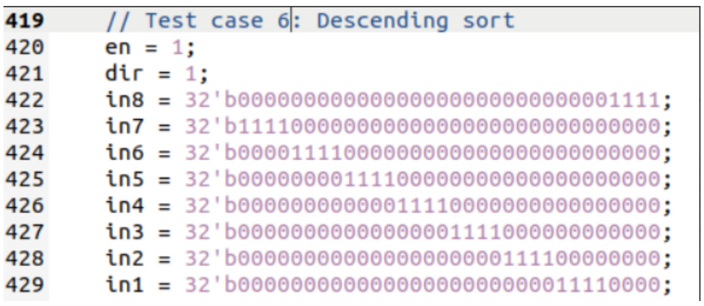
```verilog
module cae_tb;
reg clk,clks;
reg rst;
reg en;
reg dir;
reg [31:0] in1;
reg [31:0] in2;
reg [31:0] in3;
reg [31:0] in4;
reg [31:0] in5;
reg [31:0] in6;
reg [31:0] in7;
reg [31:0] in8;
wire [31:0] data1;
wire [31:0] data2;
wire [31:0] data3;
wire [31:0] data4;
wire [31:0] data5;
wire [31:0] data6;
wire [31:0] data7;
wire [31:0] data8;
//wire [31:0] d1;
//wire [31:0] d2;
//wire [31:0] d3;
//wire [31:0] d4;
// Instantiate the cae module
sorter dut (.clk(clk),.rst(rst),.en(en),.clks(clks),.dir(dir),.in1(in1),.in2(in2),
.in3(in3),.in4(in4),.in5(in5),.in6(in6),.in7(in7),.in8(in8),.data1(data1),.data2(data2),
.data3(data3),.data4(data4),.data5(data5),.data6(data6),.data7(data7),.data8(data8)    );
// Clock generation
always begin
    #5 clk = ~clk;
end

initial begin
clks=1;
forever
#40 clks=~clks;
end

// Test stimulus
initial begin
    clk = 1;
    en = 0;
    dir = 0;
    in1 = 32'h11011011;
    in2 = 32'h00101100;
    in3 = 32'h10101010;
    in4 = 32'h01100110;
    in5 = 32'h0;
    in6 = 32'h2;
    in7 = 32'h3;
    in8 = 32'h6;
```



```
    // Test case 1: Ascending sort
    en = 1;
    dir = 0;
    in1 = 32'h1;
    in2 = 32'h4;
    in3 = 32'h5;
    in4 = 32'h7;
    in5 = 32'h0;
    in6 = 32'h2;
    in7 = 32'h3;
    in8 = 32'h6;
    #90;
    // Test case 2: Ascending sort
    en = 1;
    dir = 1;
    in1 = 32'h1;
    in2 = 32'h4;
    in3 = 32'h5;
    in4 = 32'h7;
    in5 = 32'h0;
    in6 = 32'h2;
    in7 = 32'h3;
    in8 = 32'h6;
#200;
        // T2
    // Apply reset
    rst = 1;
    #7
    rst=0;
    // Test case 3: Ascending sort
    en = 1;
    dir = 0;
    in1 = 32'h6;
    in2 = 32'h3;
    in3 = 32'h2;
    in4 = 32'h0;
    in5 = 32'h7;
    in6 = 32'h5;
    in7 = 32'h4;
    in8 = 32'h1;
    #90;
    // Test case 4: Ascending sort
    en = 1;
    dir = 1;
    in1 = 32'h6;
    in2 = 32'h3;
    in3 = 32'h2;
    in4 = 32'h0;
    in5 = 32'h7;
    in6 = 32'h5;
    in7 = 32'h4;
    in8 = 32'h1;
#200;
```



```
    // Test case 5: Ascending sort
    en = 1;
    dir = 0;
    in1 = 32'b11110000000000000000000000000000;
    in2 = 32'b00001111000000000000000000000000;
    in3 = 32'b00000000111100000000000000000000;
    in4 = 32'b00000000000011110000000000000000;
    in5 = 32'b00000000000000001111000000000000;
    in6 = 32'b00000000000000000000111100000000;
    in7 = 32'b00000000000000000000000011110000;
    in8 = 32'b00000000000000000000000000001111;
    #90;
```



```
    // Test case 6: Descending sort
    en = 1;
    dir = 1;
    in8 = 32'b00000000000000000000000000001111;
    in7 = 32'b11110000000000000000000000000000;
    in6 = 32'b00001111000000000000000000000000;
    in5 = 32'b00000000111100000000000000000000;
    in4 = 32'b00000000000011110000000000000000;
    in3 = 32'b00000000000000001111000000000000;
    in2 = 32'b00000000000000000000111100000000;
    in1 = 32'b00000000000000000000000011110000;
```

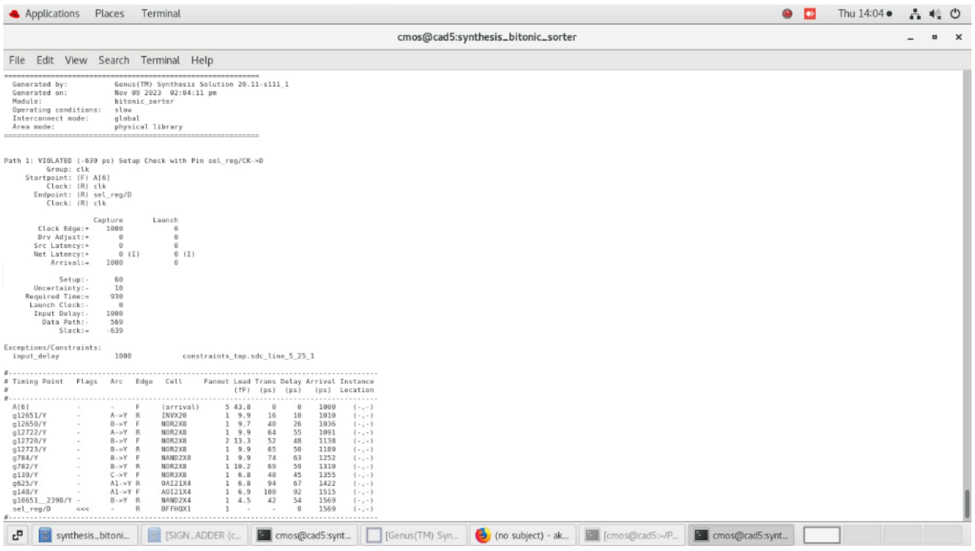
Test case 1 : The test verifies the output after applying ascending sorting.
Test case 6 : This test is to verify the outputs after applying descending sort

Similarly, many more cases have been added to get a high coverage value

REPORTS:

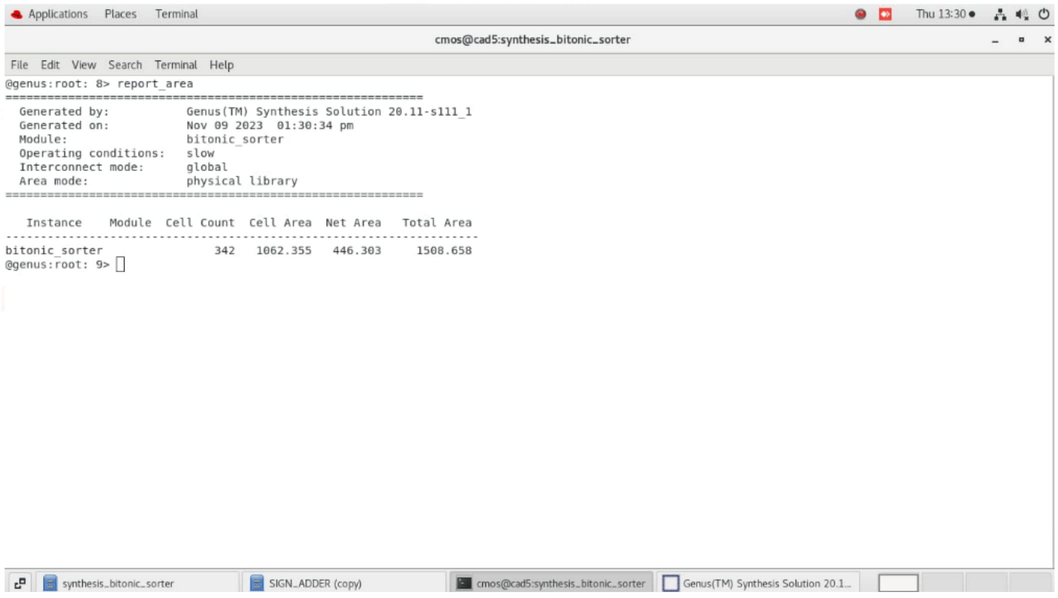
1.CAE Block

a) Timing



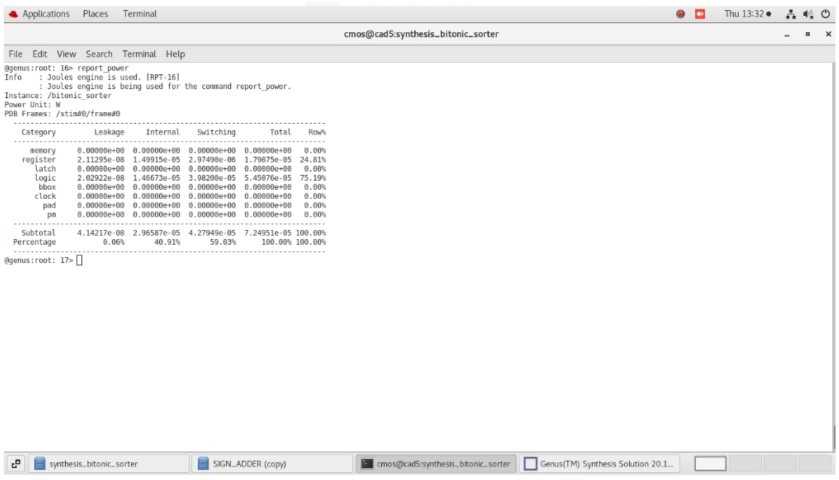
```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:           Nov 09 2023  02:04:11 pm
  Module:                 bitonic_sorter
  Operating conditions:   slow
  Interconnect mode:      global
  Area mode:              physical library
============================================================


Path 1: VIOLATED (-639 ps) Setup Check with Pin sel_reg/CK->D
          Group: clk
     Startpoint: (F) A[6]
          Clock: (R) clk
       Endpoint: (R) sel_reg/D
          Clock: (R) clk

                     Capture       Launch
        Clock Edge:+    1000            0
        Drv Adjust:+       0            0
       Src Latency:+       0            0
       Net Latency:+       0 (I)        0 (I)
           Arrival:=    1000            0

             Setup:-      60
       Uncertainty:-      10
     Required Time:=     930
      Launch Clock:-       0
       Input Delay:-    1000
         Data Path:-     569
             Slack:=    -639

Exceptions/Constraints:
  input_delay             1000            constraints_top.sdc_line_5_25_1

#-----------------------------------------------------------------------------------------
# Timing Point     Flags    Arc    Edge    Cell       Fanout Load Trans Delay Arrival Instance
#                                                            (fF)  (ps)  (ps)    (ps) Location
#-----------------------------------------------------------------------------------------
  A[6]             -        -      F       (arrival)       5 43.8     0     0    1000    (-,-)
  g12651/Y         -        A->Y   R       INVX20          1  9.9    16    10    1010    (-,-)
  g12650/Y         -        B->Y   F       NOR2X8          1  9.7    40    26    1036    (-,-)
  g12722/Y         -        A->Y   R       NOR2X8          1  9.9    64    55    1091    (-,-)
  g12720/Y         -        B->Y   F       NOR2X8          2 13.3    52    48    1138    (-,-)
  g12723/Y         -        B->Y   R       NOR2X8          1  9.9    65    50    1189    (-,-)
  g784/Y           -        B->Y   F       NAND2X8         1  9.9    74    63    1252    (-,-)
  g782/Y           -        B->Y   R       NOR2X8          1 10.2    69    59    1310    (-,-)
  g139/Y           -        C->Y   F       NOR3X8          1  6.8    48    45    1355    (-,-)
  g625/Y           -        A1->Y  R       OAI21X4         1  6.8    94    67    1422    (-,-)
  g148/Y           -        A1->Y  F       AOI21X4         1  6.9   100    92    1515    (-,-)
  g10651__2398/Y   -        B->Y   R       NAND2X4         1  4.5    42    54    1569    (-,-)
  sel_reg/D        <<<      -      R       DFFHQX1         1    -     -     0    1569    (-,-)
#-----------------------------------------------------------------------------------------
```

b) Area



```
@genus:root: 8> report_area
============================================================
  Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:           Nov 09 2023  01:30:34 pm
  Module:                 bitonic_sorter
  Operating conditions:   slow
  Interconnect mode:      global
  Area mode:              physical library
============================================================

   Instance    Module  Cell Count  Cell Area  Net Area   Total Area
---------------------------------------------------------------------
bitonic_sorter                342   1062.355   446.303     1508.658
@genus:root: 9>
```

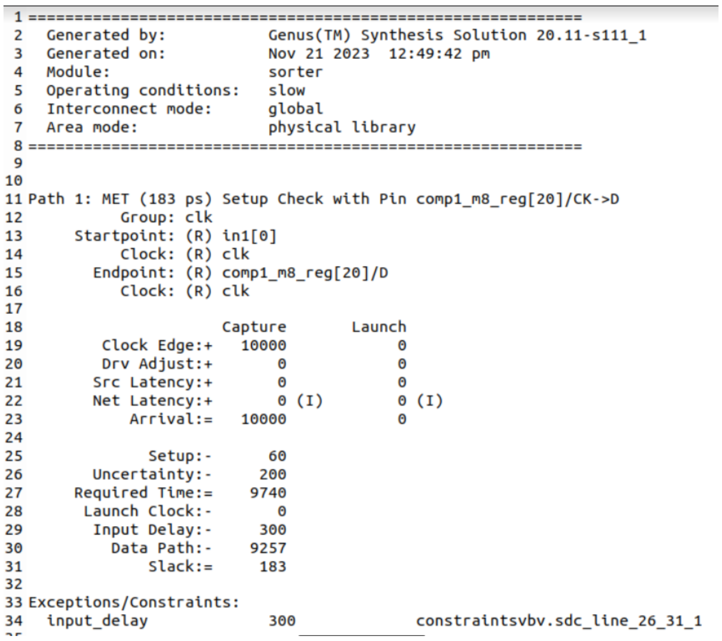
c)Power



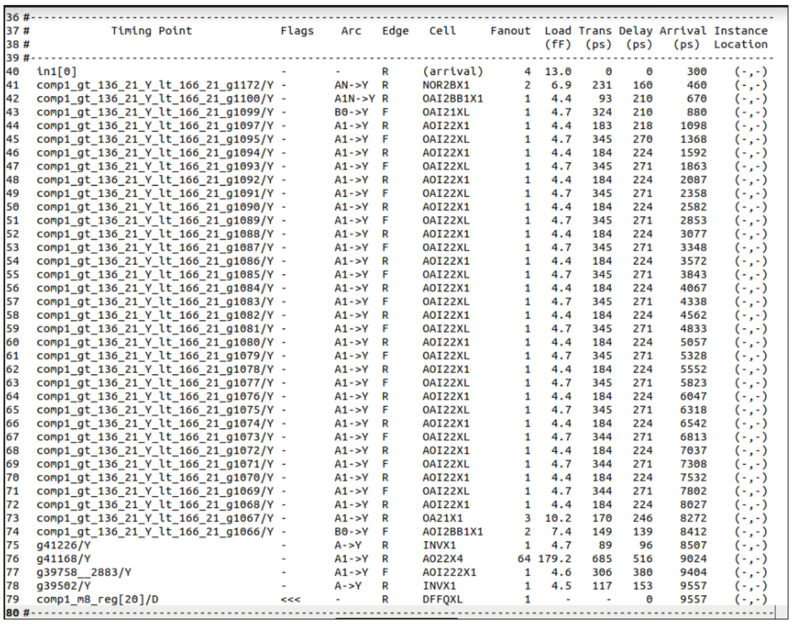
```
@genus:root: 16> report_power
Info    : Joules engine is used. [RPT-16]
        : Joules engine is being used for the command report_power.
Instance: /bitonic_sorter
Power Unit: W
PDB Frames: /stim#0/frame#0
  ---------------------------------------------------------------------
    Category       Leakage     Internal    Switching        Total    Row%
  ---------------------------------------------------------------------
      memory   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
    register   2.11295e-08  1.49915e-05  2.97490e-06  1.79875e-05  24.81%
       latch   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       logic   2.02922e-08  1.46673e-05  3.98200e-05  5.45076e-05  75.19%
        bbox   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       clock   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
         pad   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
          pm   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ---------------------------------------------------------------------
    Subtotal   4.14217e-08  2.96587e-05  4.27949e-05  7.24951e-05 100.00%
  Percentage         0.06%       40.91%       59.03%      100.00% 100.00%
  ---------------------------------------------------------------------
@genus:root: 17> 
```

2. 8 Input Bitonic Sorter:

a) Timing

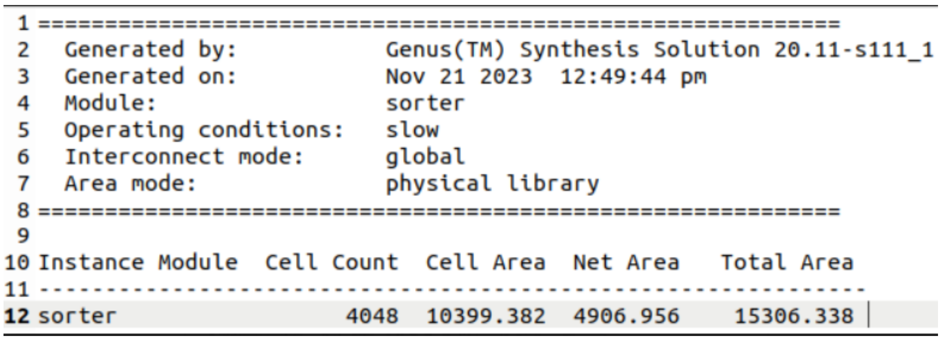


```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:           Nov 21 2023  12:49:42 pm
  Module:                 sorter
  Operating conditions:   slow
  Interconnect mode:      global
  Area mode:              physical library
============================================================


Path 1: MET (183 ps) Setup Check with Pin comp1_m8_reg[20]/CK->D
          Group: clk
     Startpoint: (R) in1[0]
          Clock: (R) clk
       Endpoint: (R) comp1_m8_reg[20]/D
          Clock: (R) clk

                     Capture       Launch
        Clock Edge:+   10000            0
        Drv Adjust:+       0            0
       Src Latency:+       0            0
       Net Latency:+       0 (I)        0 (I)
           Arrival:=   10000            0

             Setup:-      60
       Uncertainty:-     200
     Required Time:=    9740
      Launch Clock:-       0
       Input Delay:-     300
         Data Path:-    9257
             Slack:=     183

Exceptions/Constraints:
  input_delay             300             constraintsvbv.sdc_line_26_31_1
```



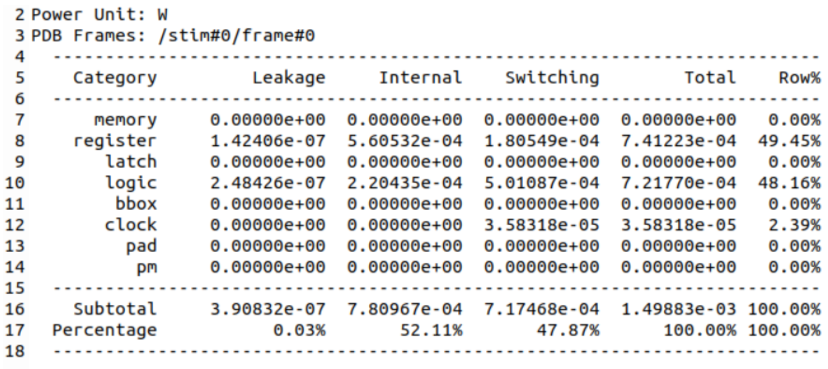
```
#---------------------------------------------------------------------------------------------------------------
#                Timing Point                Flags    Arc    Edge   Cell       Fanout  Load  Trans Delay Arrival Instance
#                                                                                      (fF)  (ps)  (ps)   (ps)   Location
#---------------------------------------------------------------------------------------------------------------
  in1[0]                                      -       -       R     (arrival)       4  13.0     0     0     300   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1172/Y -              AN->Y    R     NOR2BX1         2   6.9   231   160     460   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1100/Y -              A1N->Y   R     OAI2BB1X1       1   4.4    93   210     670   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1099/Y -              B0->Y    F     OAI21XL         1   4.7   324   210     880   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1097/Y -              A1->Y    R     AOI22X1         1   4.4   183   218    1098   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1095/Y -              A1->Y    F     OAI22XL         1   4.7   345   270    1368   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1094/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    1592   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1093/Y -              A1->Y    F     OAI22XL         1   4.7   345   271    1863   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1092/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    2087   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1091/Y -              A1->Y    F     OAI22XL         1   4.7   345   271    2358   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1090/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    2582   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1089/Y -              A1->Y    F     OAI22XL         1   4.7   345   271    2853   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1088/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    3077   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1087/Y -              A1->Y    F     OAI22XL         1   4.7   345   271    3348   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1086/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    3572   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1085/Y -              A1->Y    F     OAI22XL         1   4.7   345   271    3843   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1084/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    4067   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1083/Y -              A1->Y    F     OAI22XL         1   4.7   345   271    4338   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1082/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    4562   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1081/Y -              A1->Y    F     OAI22XL         1   4.7   345   271    4833   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1080/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    5057   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1079/Y -              A1->Y    F     OAI22XL         1   4.7   345   271    5328   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1078/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    5552   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1077/Y -              A1->Y    F     OAI22XL         1   4.7   345   271    5823   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1076/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    6047   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1075/Y -              A1->Y    F     OAI22XL         1   4.7   345   271    6318   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1074/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    6542   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1073/Y -              A1->Y    F     OAI22XL         1   4.7   344   271    6813   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1072/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    7037   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1071/Y -              A1->Y    F     OAI22XL         1   4.7   344   271    7308   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1070/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    7532   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1069/Y -              A1->Y    F     OAI22XL         1   4.7   344   271    7802   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1068/Y -              A1->Y    R     AOI22X1         1   4.4   184   224    8027   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1067/Y -              A1->Y    R     OA21X1          3  10.2   170   246    8272   (-,-)
  comp1_gt_136_21_Y_lt_166_21_g1066/Y -              B0->Y    F     AOI2BB1X1       2   7.4   149   139    8412   (-,-)
  g41226/Y                                    -      A->Y     R     INVX1           1   4.7    89    96    8507   (-,-)
  g41168/Y                                    -      A1->Y    R     AO22X4         64 179.2   685   516    9024   (-,-)
  g39758__2883/Y                              -      A1->Y    F     AOI222X1        1   4.6   306   380    9404   (-,-)
  g39502/Y                                    -      A->Y     R     INVX1           1   4.5   117   153    9557   (-,-)
  comp1_m8_reg[20]/D                          <<<    -        R     DFFQXL          1     -     -     0    9557   (-,-)
#---------------------------------------------------------------------------------------------------------------
```

b) Area



```
============================================================
  Generated by:            Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:            Nov 21 2023  12:49:44 pm
  Module:                  sorter
  Operating conditions:    slow
  Interconnect mode:       global
  Area mode:               physical library
============================================================

Instance Module  Cell Count  Cell Area  Net Area   Total Area
--------------------------------------------------------------
sorter                 4048  10399.382  4906.956    15306.338
```

c) Power



```
Power Unit: W
PDB Frames: /stim#0/frame#0
  -------------------------------------------------------------------------
    Category         Leakage       Internal      Switching        Total    Row%
  -------------------------------------------------------------------------
      memory     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00    0.00%
    register     1.42406e-07   5.60532e-04   1.80549e-04   7.41223e-04   49.45%
       latch     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00    0.00%
       logic     2.48426e-07   2.20435e-04   5.01087e-04   7.21770e-04   48.16%
        bbox     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00    0.00%
       clock     0.00000e+00   0.00000e+00   3.58318e-05   3.58318e-05    2.39%
         pad     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00    0.00%
          pm     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00    0.00%
  -------------------------------------------------------------------------
    Subtotal     3.90832e-07   7.80967e-04   7.17468e-04   1.49883e-03  100.00%
  Percentage           0.03%        52.11%        47.87%       100.00%  100.00%
  -------------------------------------------------------------------------
```

FINAL RESULTS:

The compare and exchange block design consumes 1.508 mm sq. area. The compactness of the design, as reflected by the 1.51 mm² area, suggests an efficient use of resources, contributing to a balanced trade-off between functionality and spatial efficiency.
The 8 input bitonic sorter design consumes 15.306 mm sq.area
The compare and exchange block design consumes 72.5 uW of total power and 42.8 uW switching power
The 8 input bitonic sorter design consumes 1.5 mW of total power and 0.72 mW switching power.

CONCLUSION:

In this project, we have successfully written the CAE block for bitonic sorter and have verified the sorting procedure with 4 and 8 inputs. Additionally, we have obtained quite high code coverage.

REFERENCE:

1. "Digital Design and Computer Architecture" by David Money Harris and Sarah L. Harris.

2. "Verilog HDL: A Guide to Digital Design and Synthesis" by Samir Palnitkar.

3. "Digital System Design with VHDL" by Mark Zwolinski.

4. https://en.wikipedia.org/wiki/Bitonic_sorter