

8 Input Bitonic Sorter Design

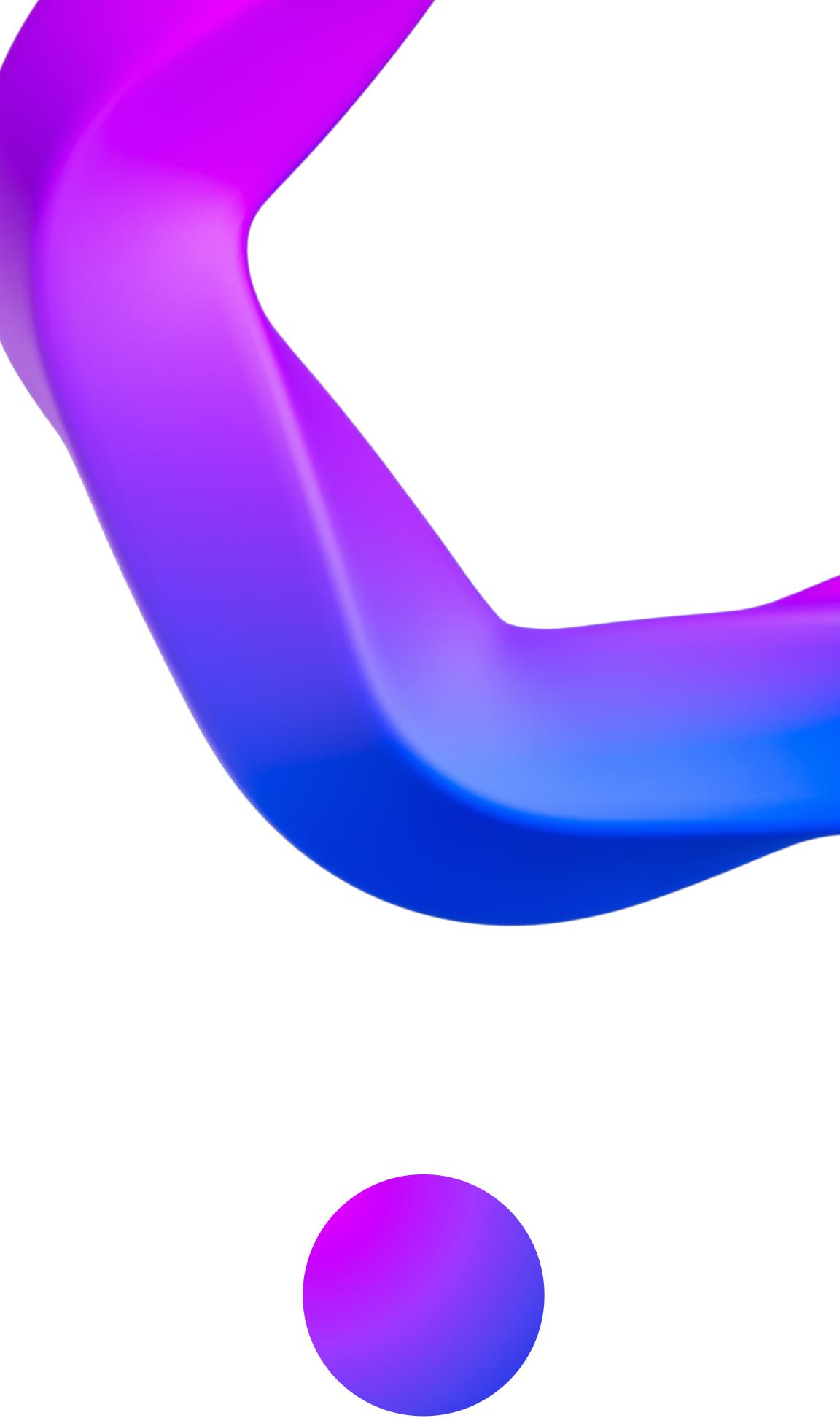
Presented By:

Akshay Sharma G - PES1UG21EC031

Akshay Anand - PES1UG21EC029

Akash Ravi Bhat - PES1UG21EC025

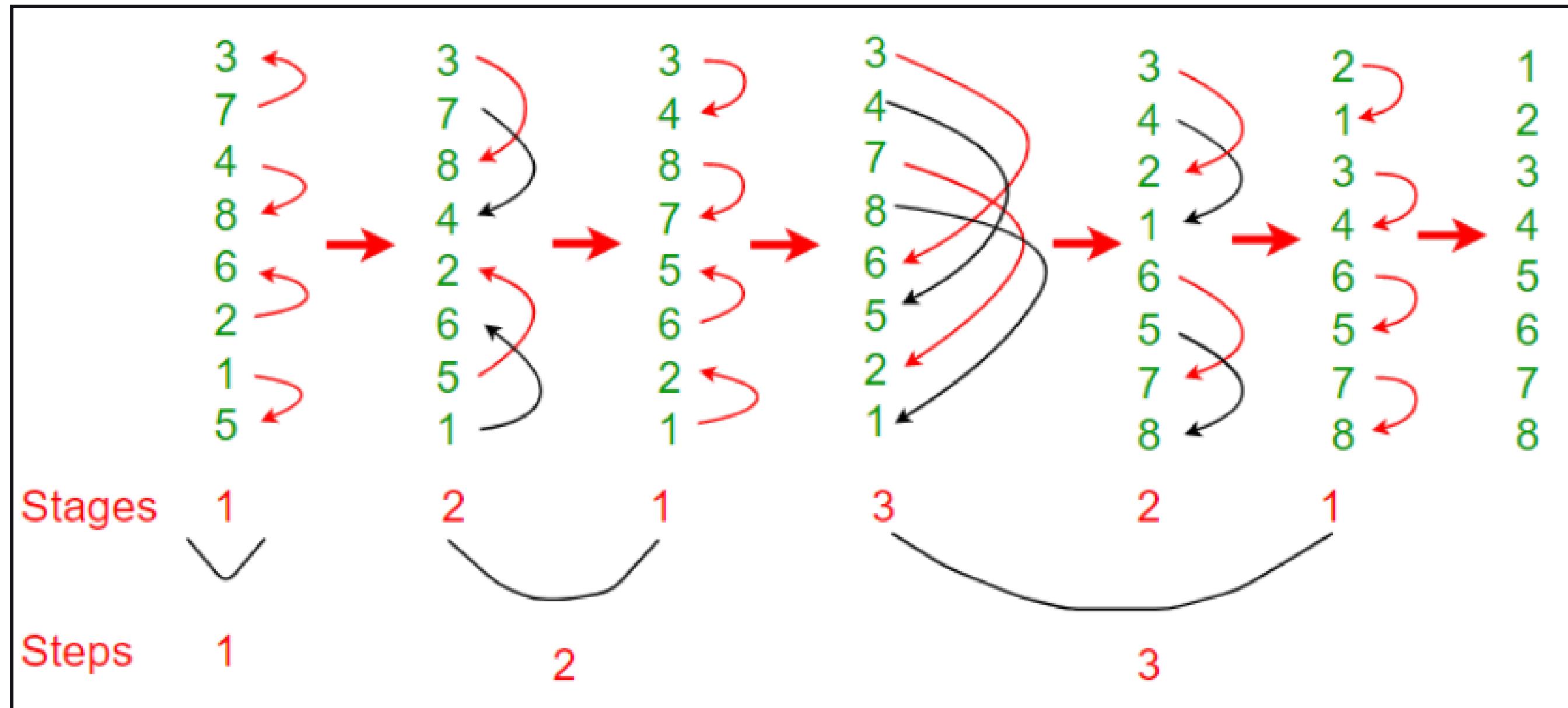
Akash Giridhar - PES1UG21EC023



Introduction

- The 8-bit bitonic sorter is a parallel sorting algorithm specifically tailored for hardware implementation.
- Its distinctive feature lies in the sorting network's pattern, beginning with a monotonic sequence and transitioning into a bitonic sequence.
- Operating on 8-bit data elements, the algorithm recursively constructs and sorts the bitonic sequence.
- This process is well-suited for efficient parallel processing, making it ideal for hardware architectures such as parallel processors or digital circuits.
- The 8-bit bitonic sorter finds practical applications in tasks requiring parallel sorting, with a notable use case in digital signal processing due to its effectiveness in handling 8-bit data elements.

Algorithm



Design File

1) Compare and Exchange block (CAE)

- Compare and exchange block
- Two inputs are compared and exchanged according to the direction(dir) signal.
- If dir = 0, the inputs are sorted in ascending order.
- If dir = 1, the inputs are sorted in descending order.

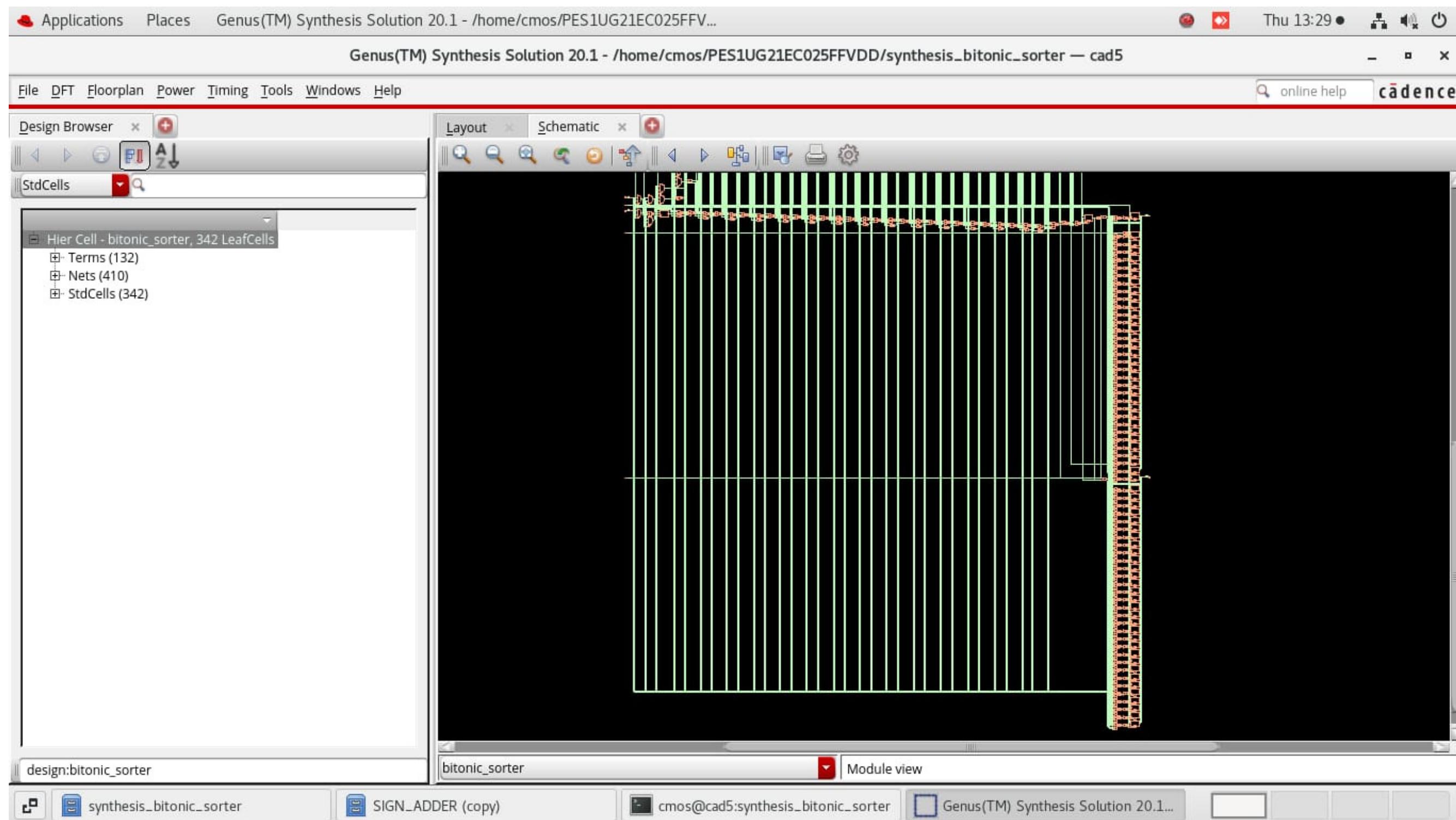
Design File

1) Compare and Exchange block (CAE)

```
1 `timescale 1ns/1ps
2 module bitonic_sorter(output wire [31:0] o1, o2, input wire [31:0] A, B, input wire clk, rst, dir, enable);
3   reg sel;
4   reg [31:0] out1, out2;
5   always @(posedge clk or posedge rst) begin
6     if (rst) begin
7       out1 <= 32'b0;
8       out2 <= 32'b0;
9     end
10    else if (enable) begin
11      if (dir == 1'b0) begin
12        sel <= (A > B);
13        if (sel==1) begin
14          out1<=B;
15          out2<=A;
16        end
17        else if (sel==0) begin
18          out1<=A;
19          out2<=B;
20        end
21      end
22      if (dir == 1'b1) begin
23        sel <= (A < B);
24        if (sel==1) begin
25          out1<=B;
26          out2<=A;
27        end
28        else if (sel==0) begin
29          out1<=A;
30          out2<=B;
31        end
32      end
33    end
34  end
35  assign o1 = out1;
36  assign o2 = out2;
37 endmodule
```

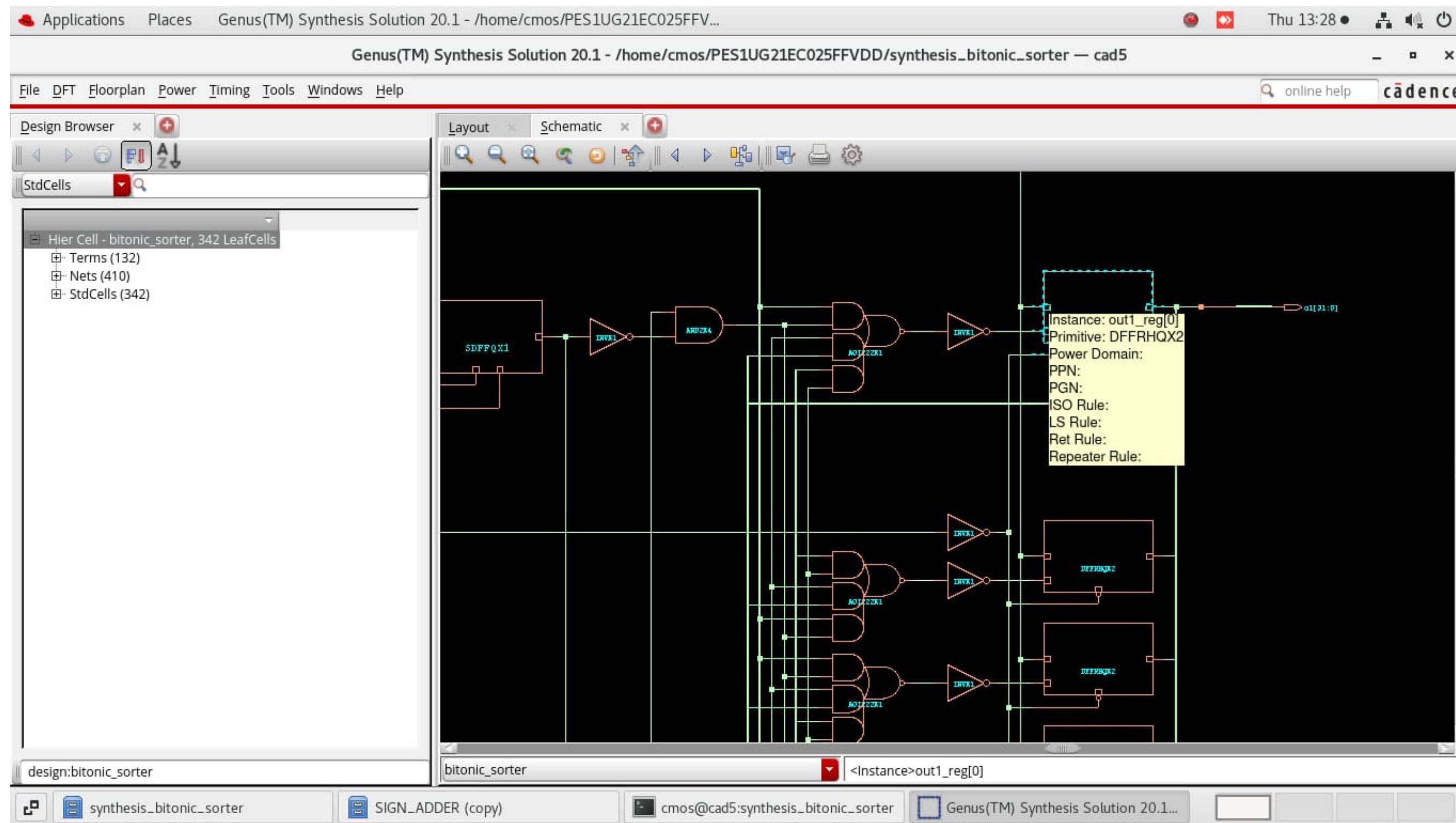
Block Diagram

1) Compare and Exchange block (CAE)



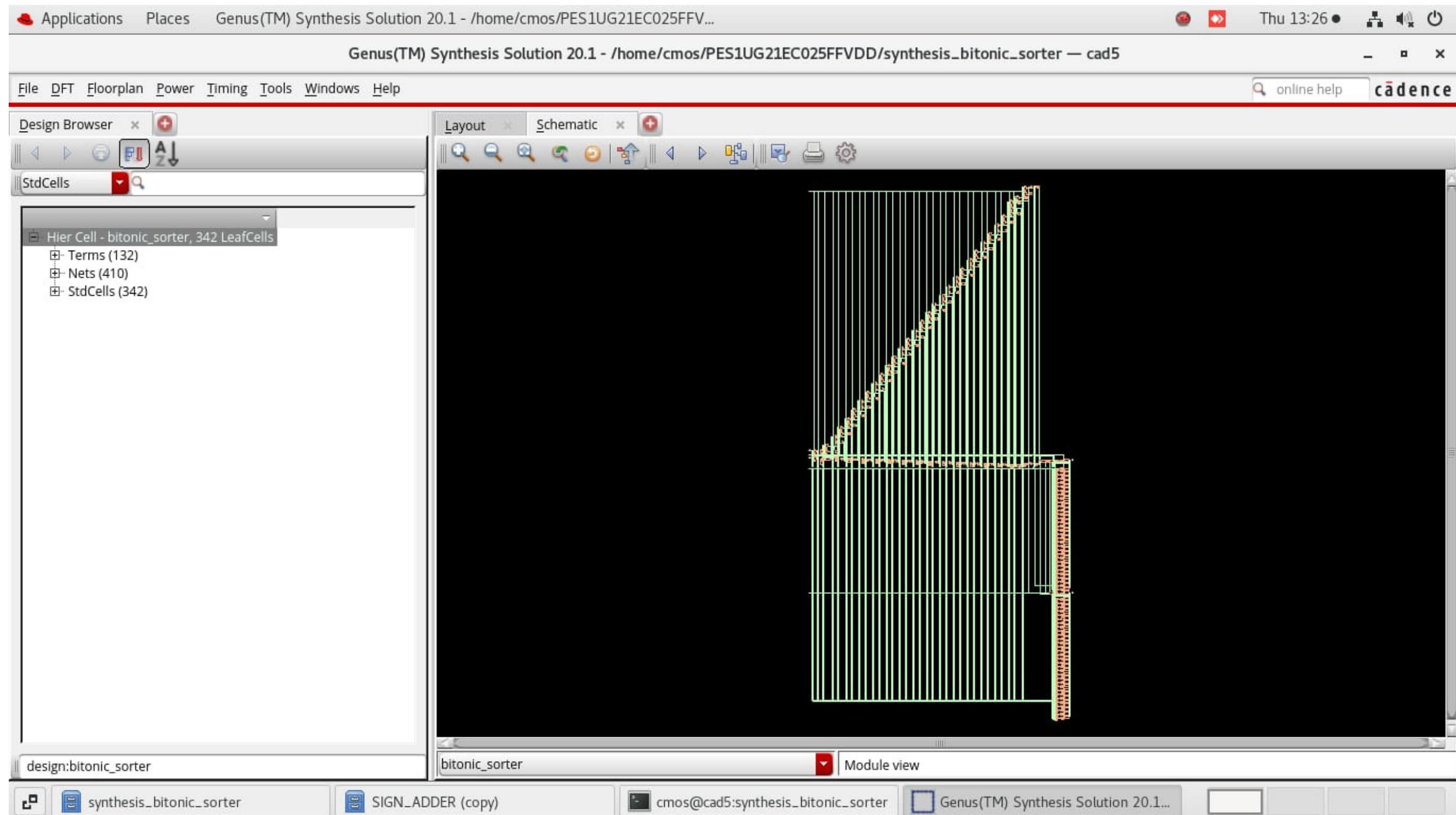
Block Diagram

1) Compare and Exchange block (CAE)



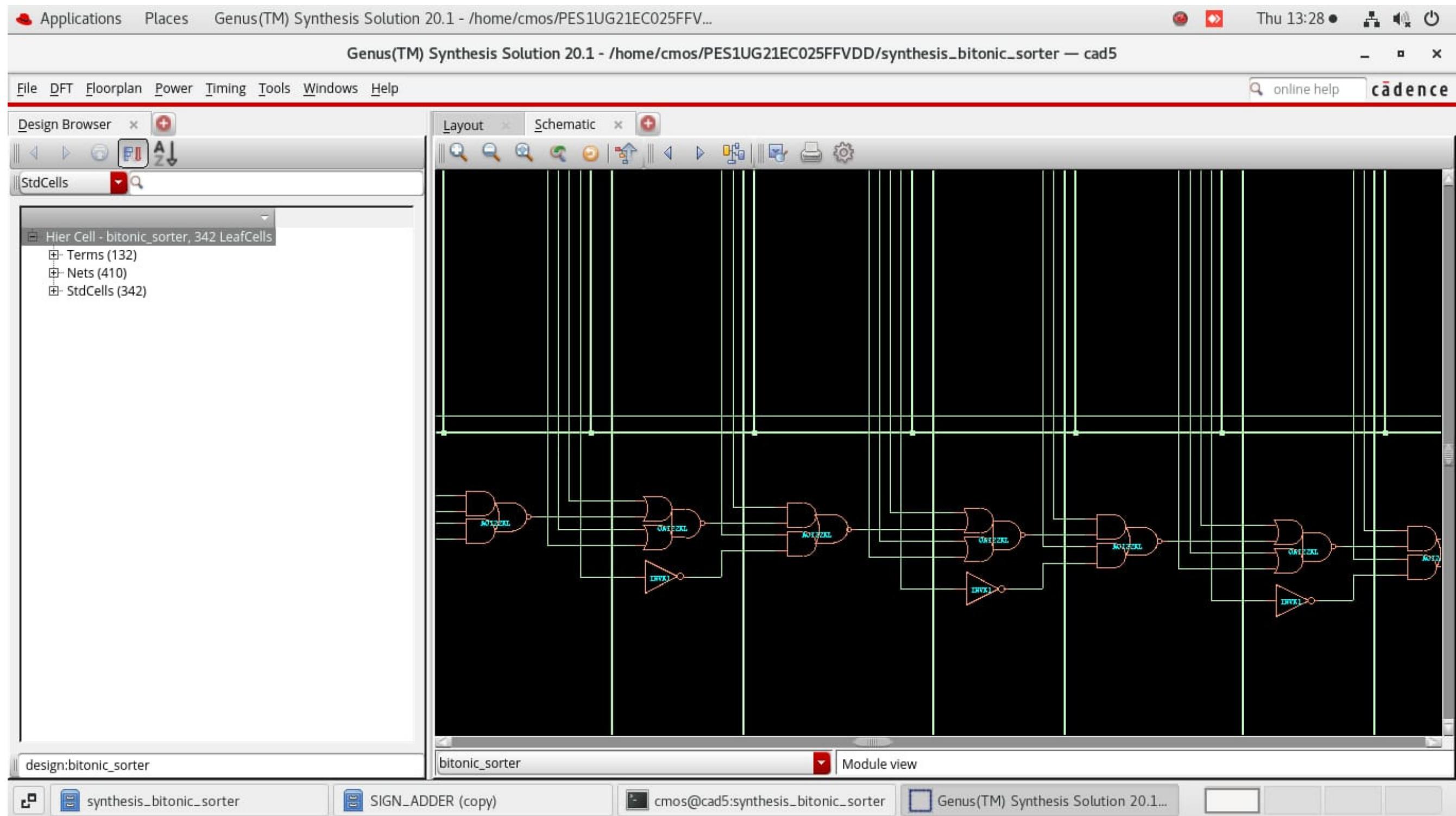
Block Diagram

1) Compare and Exchange block (CAE)



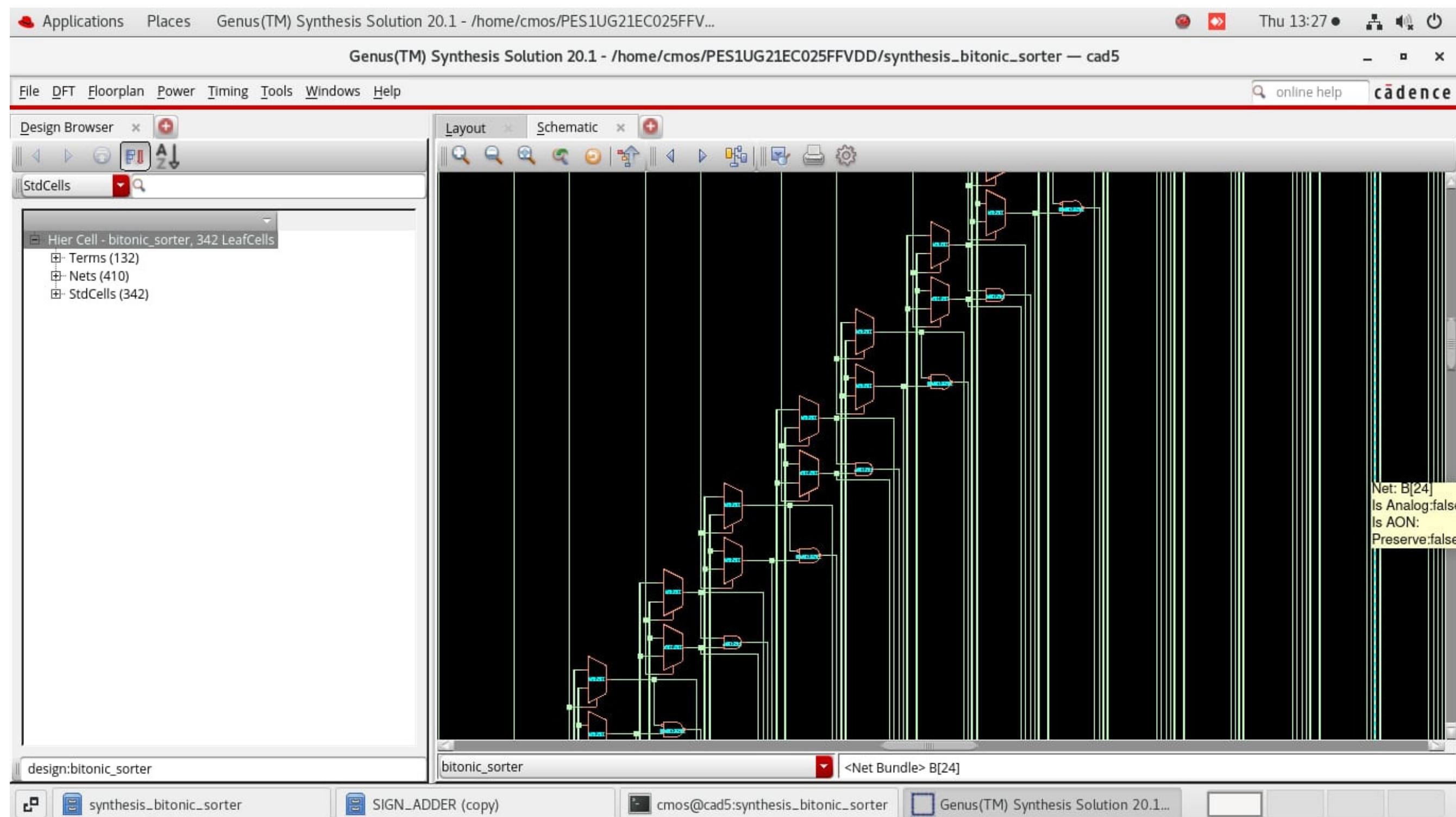
Block Diagram

1) Compare and Exchange block (CAE)



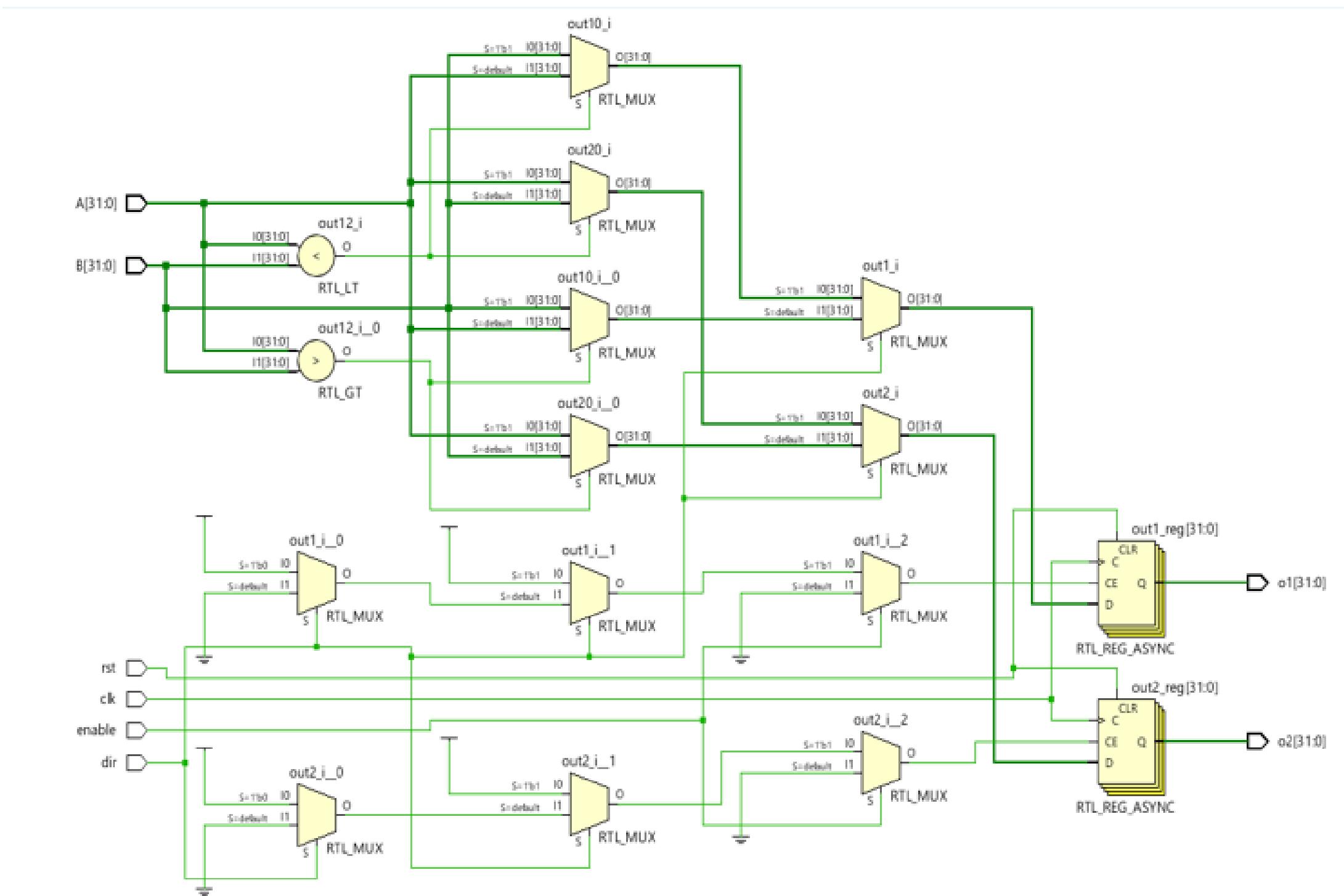
Block Diagram

1) Compare and Exchange block (CAE)



Block Diagram

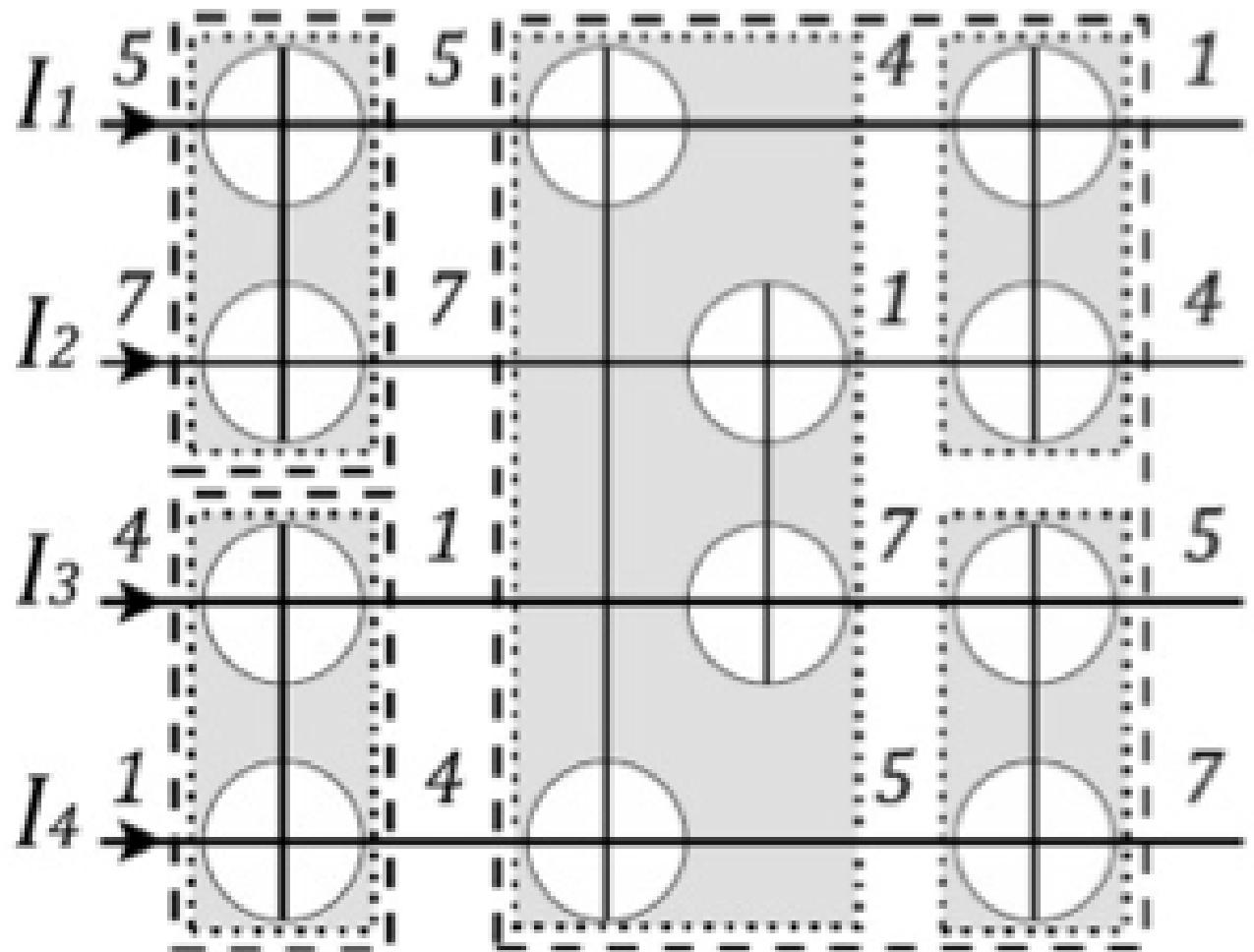
1) Compare and Exchange block (CAE)



Design File

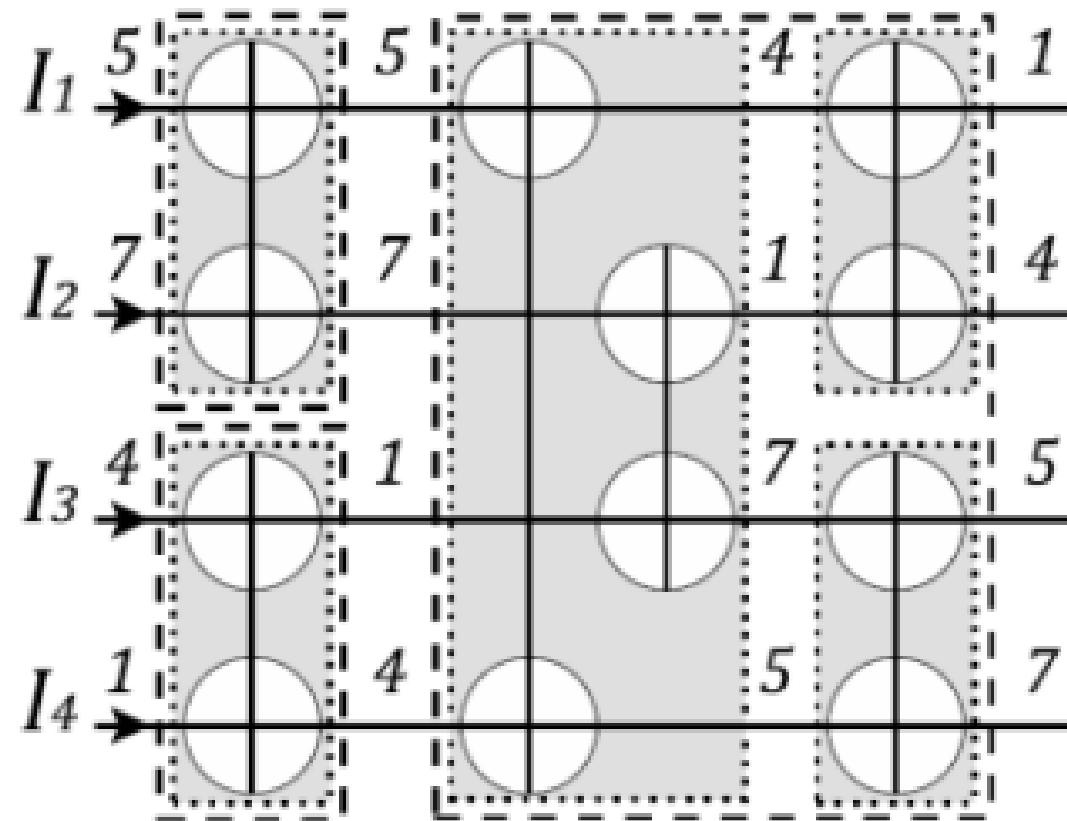
2) 4 Input Bitonic Sorting

- 4 unsorted inputs are given and 4 sorted outputs are obtained.
- This can be achieved by using 6 CAE blocks, computed in 3 clock cycles.
- Here, the previous module of CAE block is instantiated 6 times.



Design File

2) 4 Input Bitonic Sorting



```
1 `timescale 1ns/1ps
2 module bitonic4
3
4 (
5
6   output [31:0]o1,o2, o3, o4, input [31:0] in1, in2, in3, in4, input clk, rst, dir, en );
7
8   wire [31:0]s1_o1,s1_o2; //a,b
9
10  wire [31:0]s2_o1, s2_o2; //c,d
11
12  wire [31:0]s3_o1, s3_o2; //e,f
13
14  wire [31:0]s4_o1,s4_o2; //g,h
15
16
17
18  bitonic_sorter x1 (s1_o1, s1_o2, in1, in2, clk, rst, dir, en);
19
20  bitonic_sorter x2 (s2_o1, s2_o2, in3, in4, clk, rst, dir, en);
21
22  bitonic_sorter x3 (s3_o1, s3_o2, s1_o1, s2_o2, clk, rst, dir, en);
23
24  bitonic_sorter x4 (s4_o1, s4_o2, s1_o2, s2_o1, clk, rst, dir, en);
25
26  bitonic_sorter x5 (o1,o2, s3_o1, s4_o1, clk, rst, dir, en);
27
28  bitonic_sorter x6 (o3,o4,s4_o2,s3_o2, clk, rst, dir, en);
29
30 endmodule
```

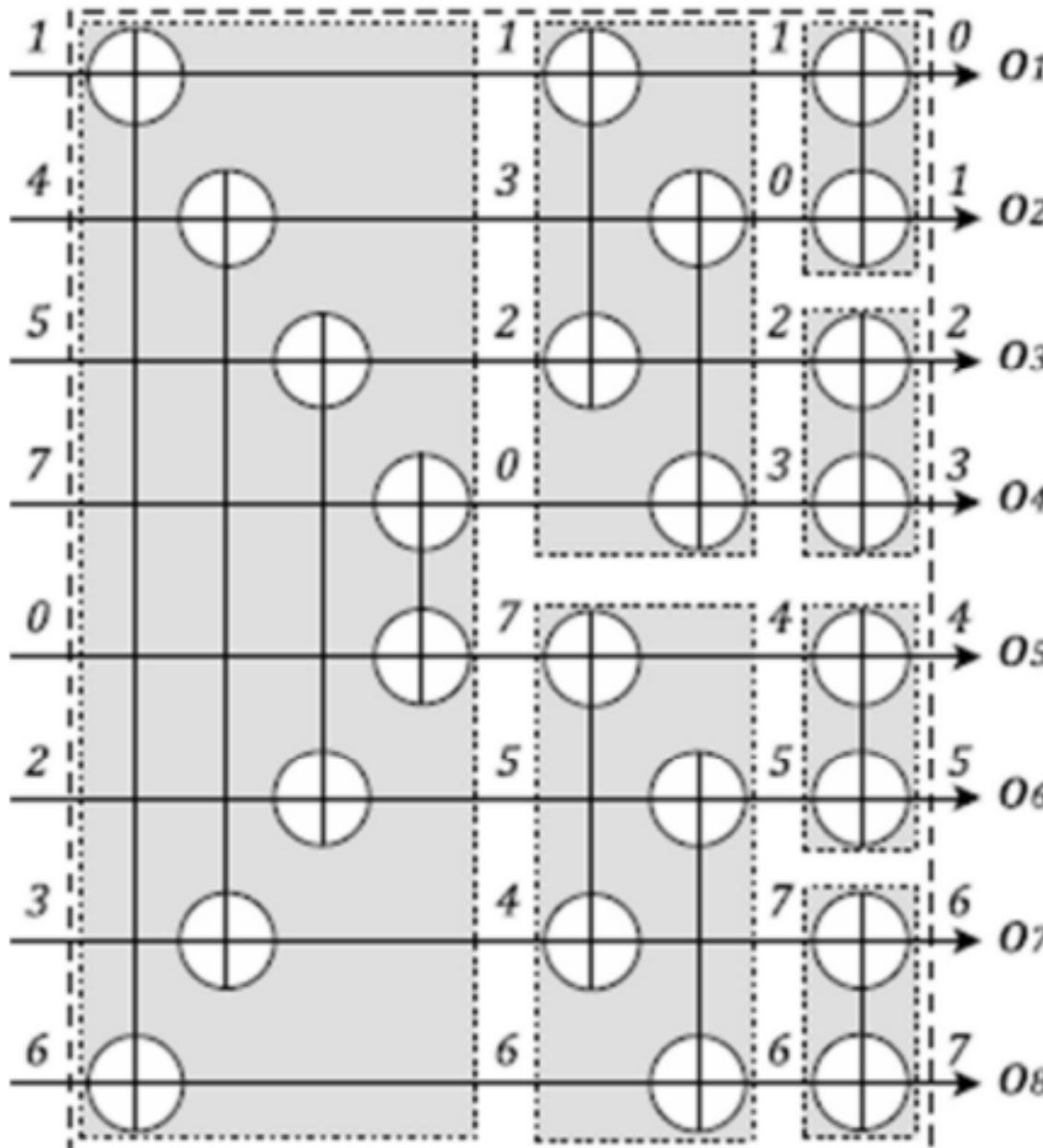
Design File

3) 8 Input Bitonic Sorting

- 2 set of 4 sorted inputs are given and 8 sorted outputs are obtained.
- This is also computed in 3 stages, using 12 CAE blocks.
- Here, CAE block is instantiated 12 times.

Design File

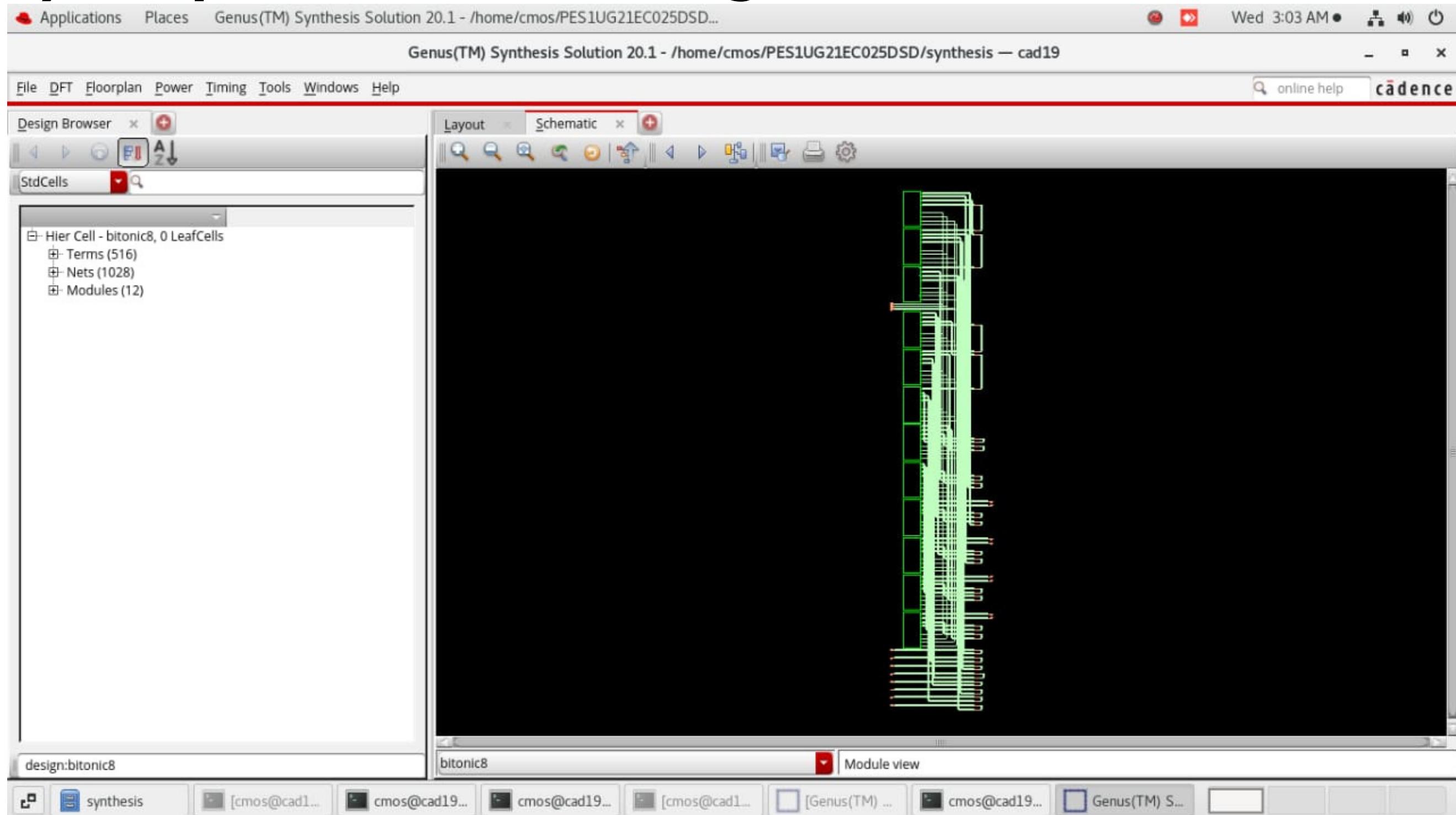
3) 8 Input Bitonic Sorting



```
1 `timescale 1ns / 1ps
2
3 module bitonic8(input [31:0]i0,i1,i2,i3,i4,i5,i6,i7,
4 output [31:0]o0,o1,o2,o3,o4,o5,o6,o7, input clk,enable,rst,dir);
5
6 //input [31:0]i0,i1,i2,i3,i4,i5,i6,i7;
7 //output [31:0]y0,y1,y2,y3,y4,y5,y6,y7;
8 //input clk,enable,rst,dir;
9 wire [31:0]w,x,i,j,k,l,m,n,o,p,q,r,s,t,u,v;
10
11 bitonic_sorter d1(w,x,i0,i7,clk,rst,dir,enable);
12 bitonic_sorter d2(i,j,i1,i6,clk,rst,dir,enable);
13 bitonic_sorter d3(k,l,i2,i5,clk,rst,dir,enable);
14 bitonic_sorter d4(m,n,i3,i4,clk,rst,dir,enable);
15
16
17 bitonic_sorter d5(o,p,w,k,clk,rst,dir,enable);
18 bitonic_sorter d6(q,r,i,m,clk,rst,dir,enable);
19 bitonic_sorter d7(s,t,n,j,clk,rst,dir,enable);
20 bitonic_sorter d8(u,v,l,x,clk,rst,dir,enable);
21
22 bitonic_sorter d9(o0,o1,o,q,clk,rst,dir,enable);
23 bitonic_sorter d10(o2,o3,p,r,clk,rst,dir,enable);
24 bitonic_sorter d11(o4,o5,s,u,clk,rst,dir,enable);
25 bitonic_sorter d12(o6,o7,t,v,clk,rst,dir,enable);
26
27 endmodule
```

Block Diagram

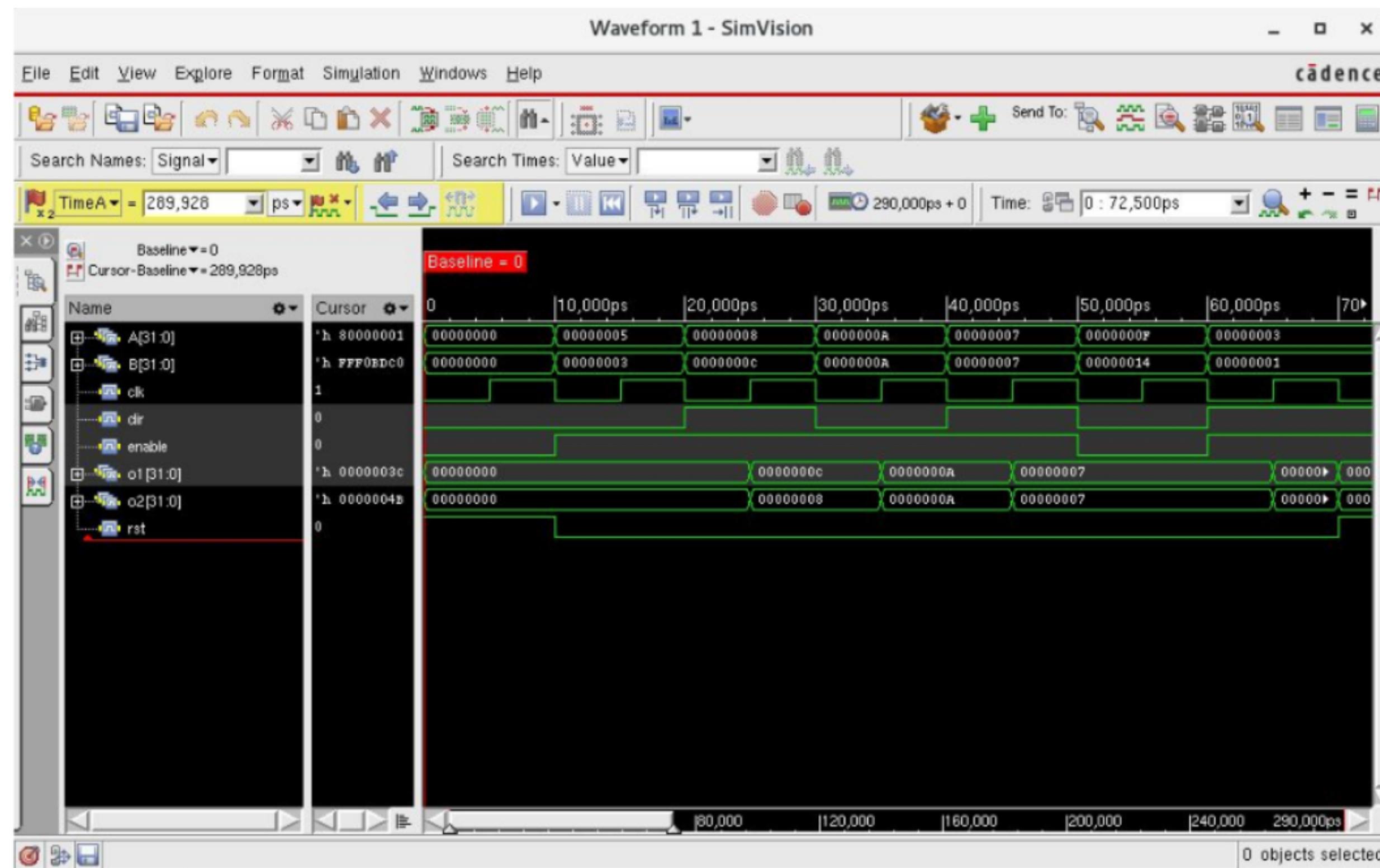
3) 8 Input Bitonic Sorting



**For 8 input sorting,
It takes 3 clock
cycles to give
outputs since the
design is executed
using 3 stages of
CAE blocks**

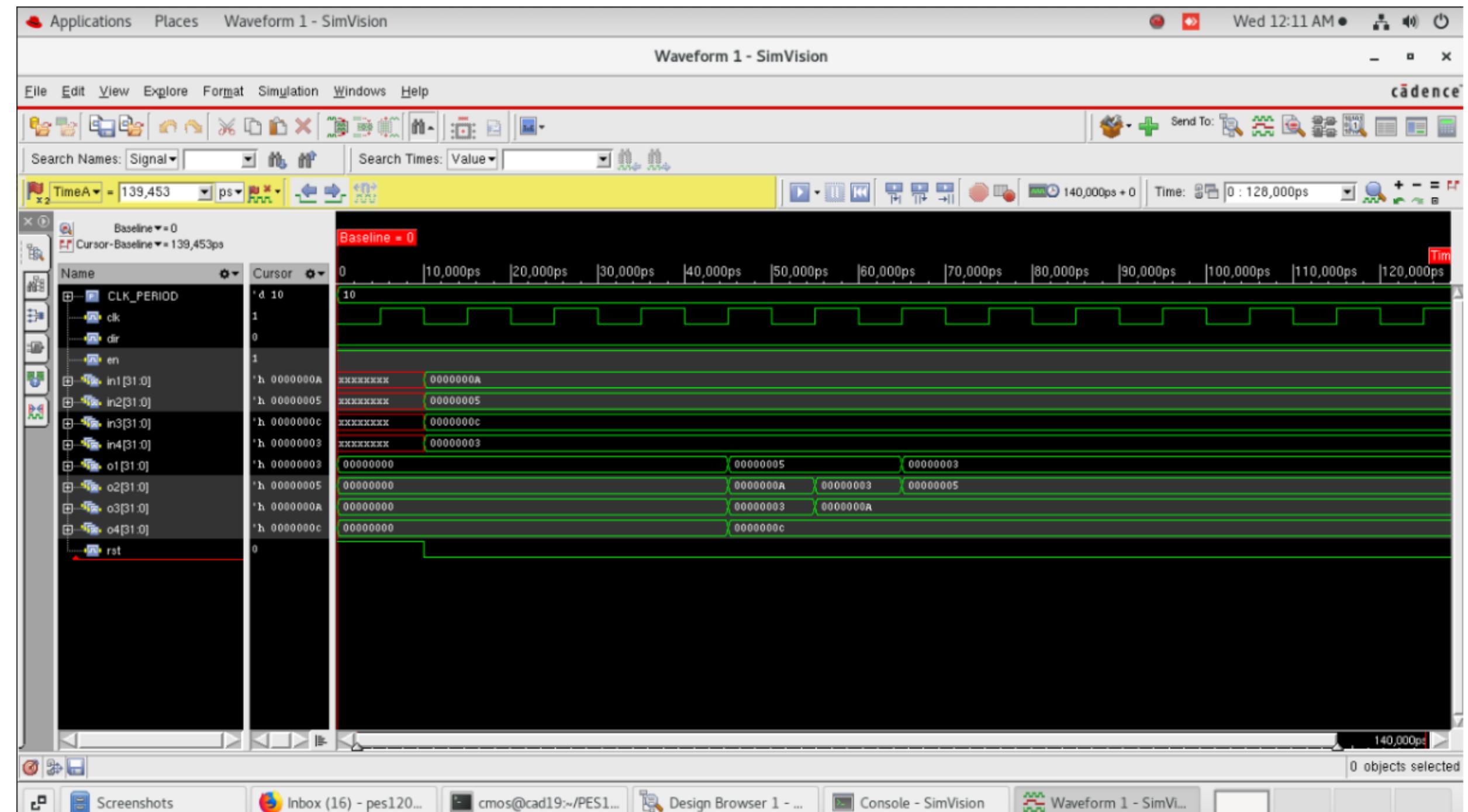
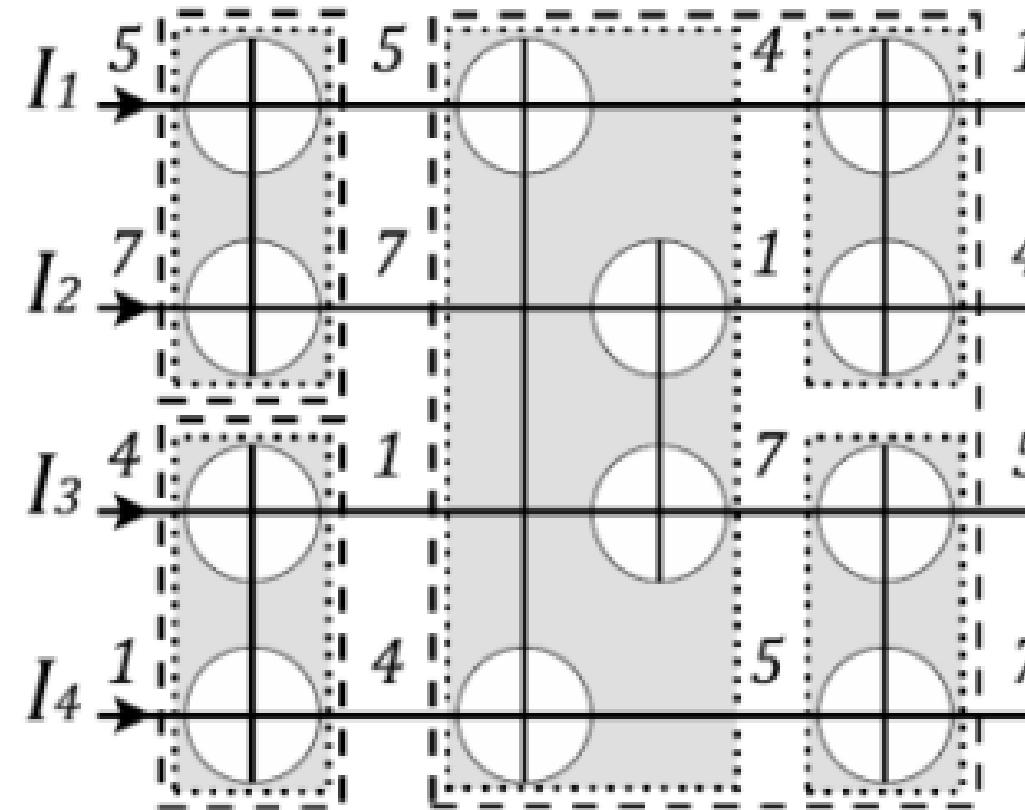
Waveforms

1) CAE block



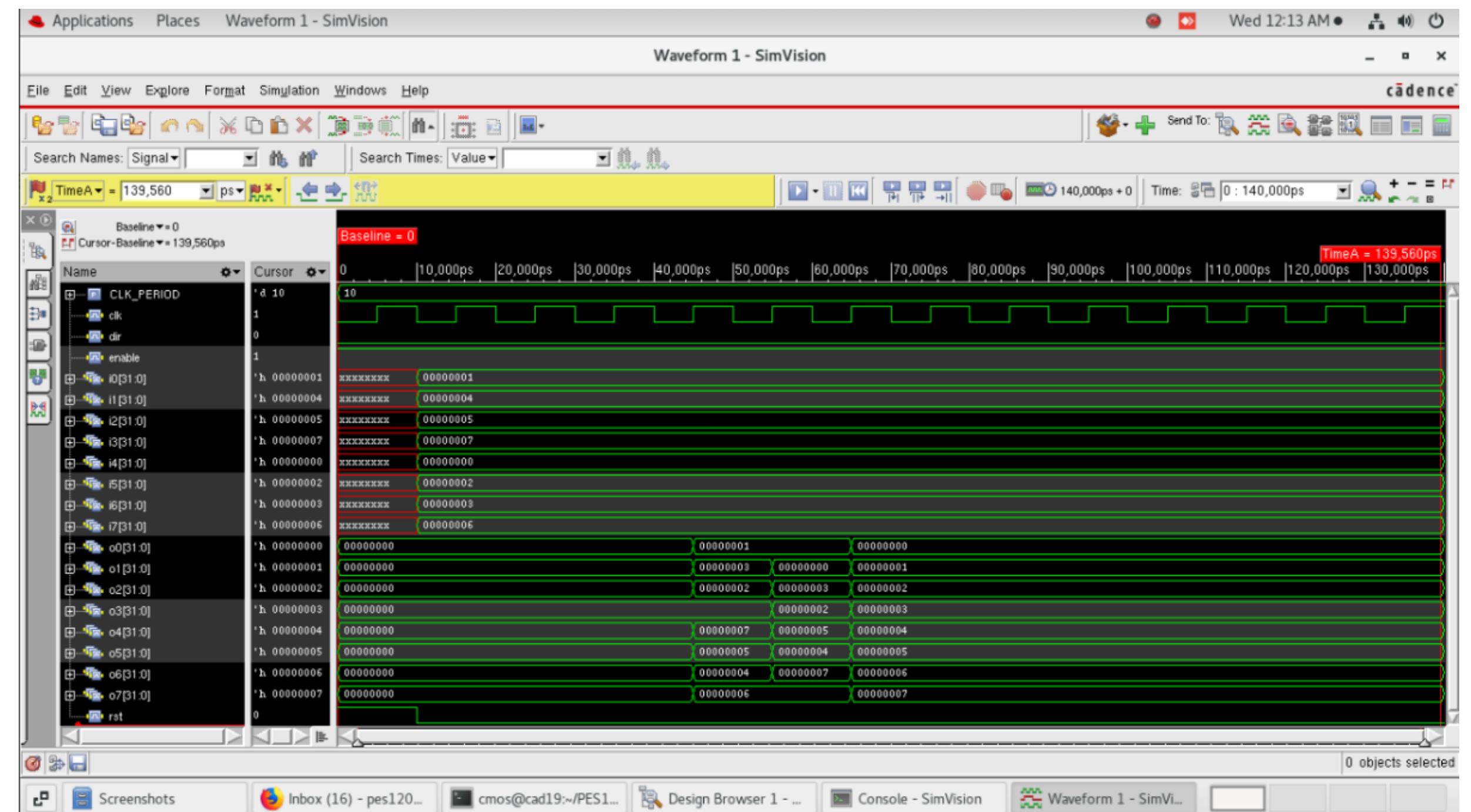
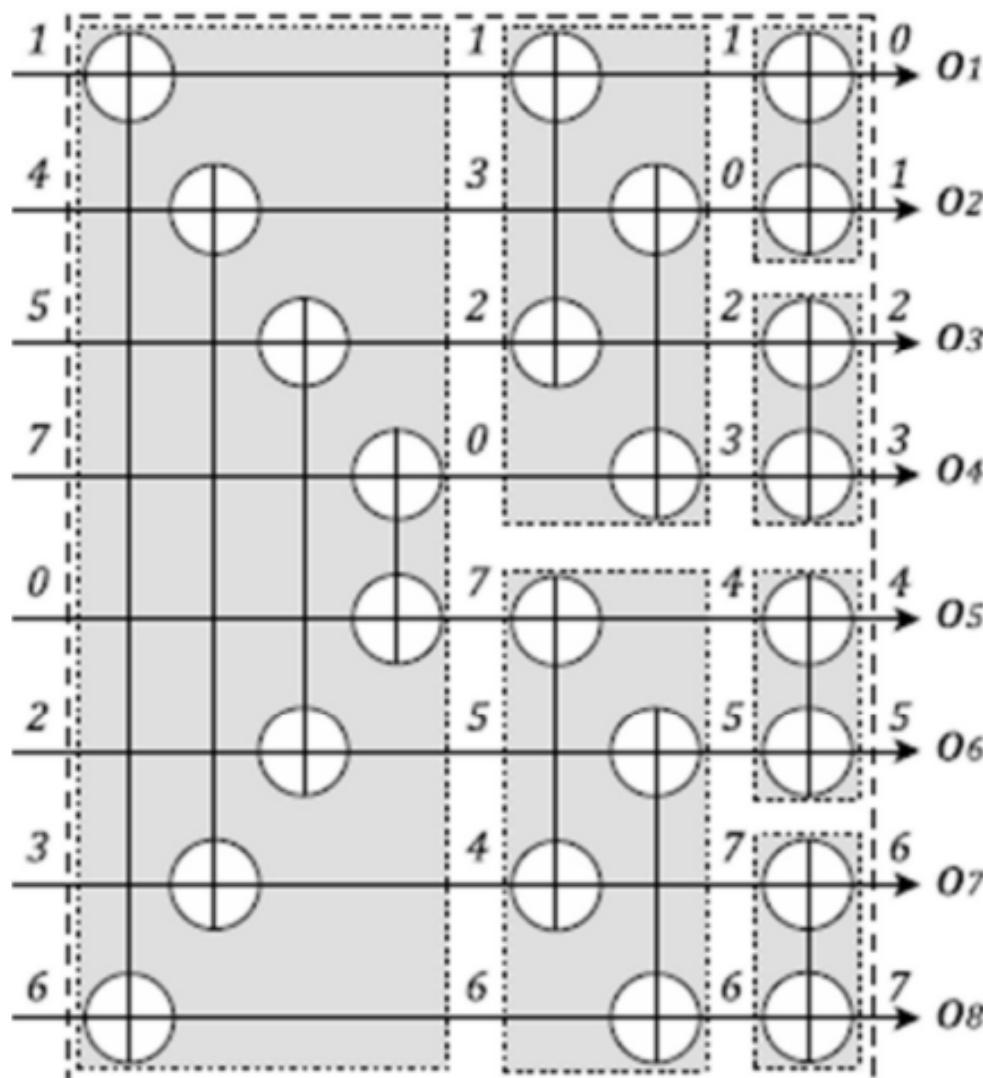
Waveforms

2) 4 input bitonic sorting



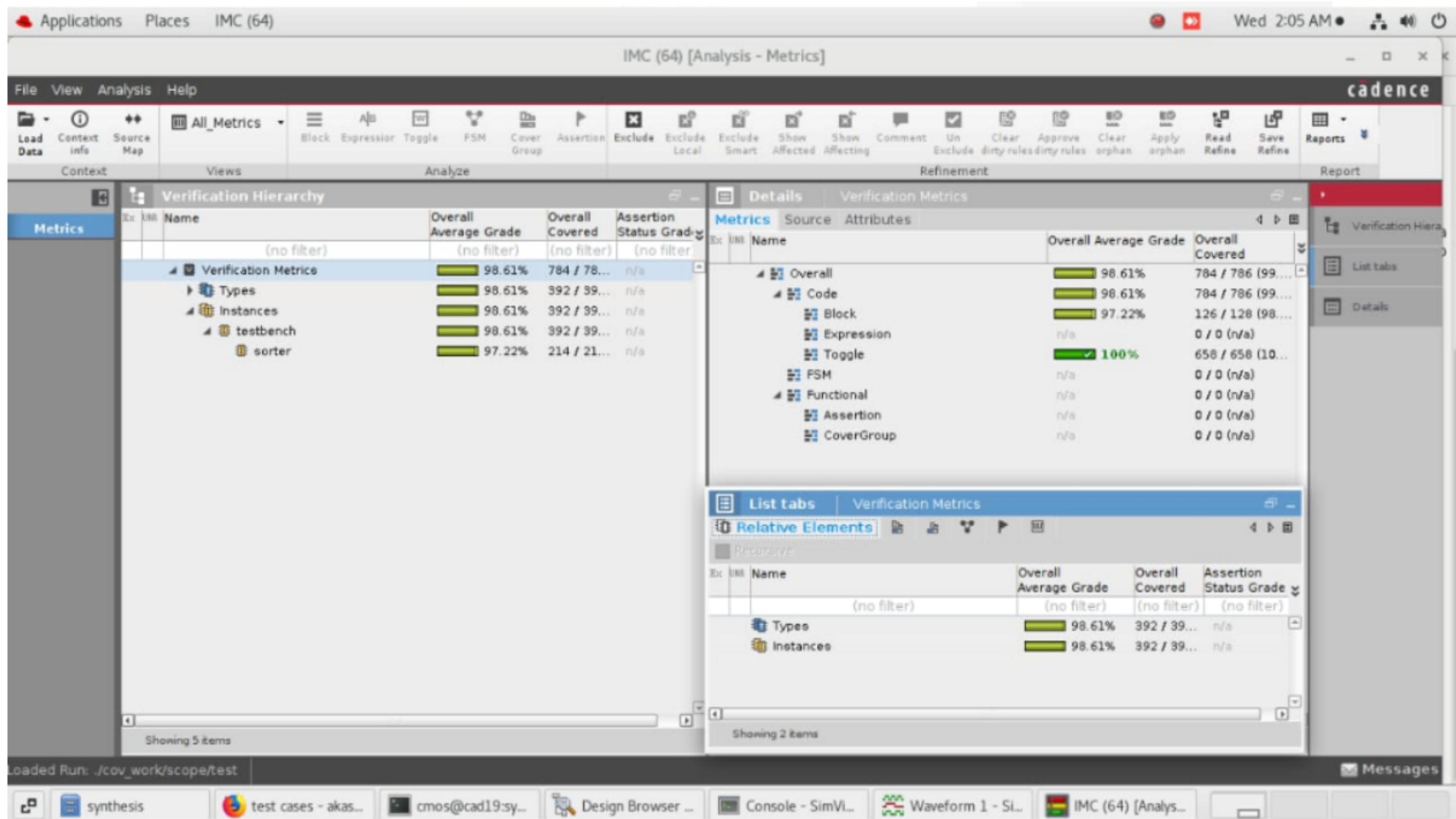
Waveforms

3) 8 input bitonic sorting



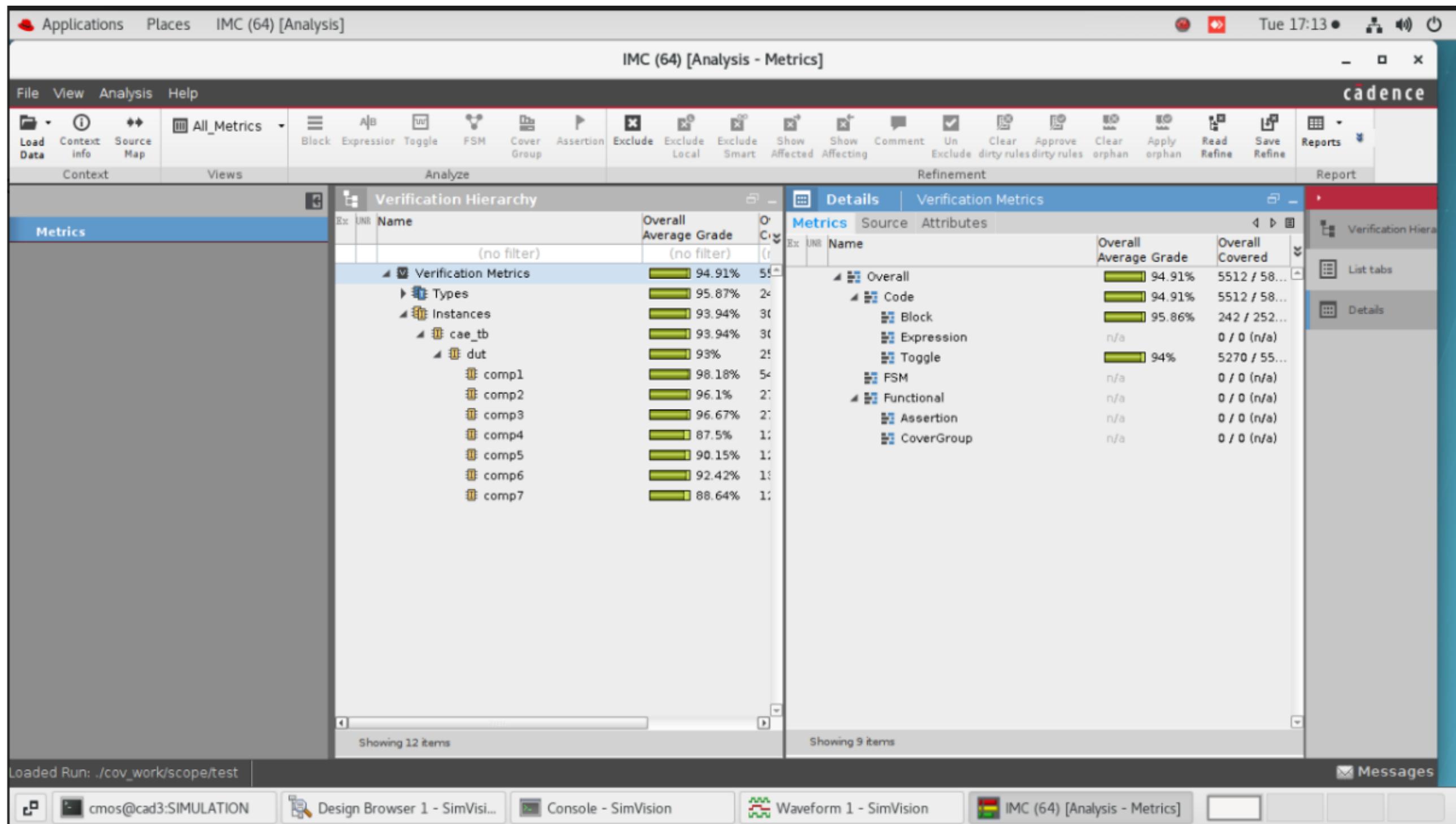
COVERAGE REPORT

1) CAE block



COVERAGE REPORT

2) 8 input bitonic sorter:



TESTBENCH

1) CAE block

```
1 module testbench();
2   reg [31:0] A, B;
3   reg clk, rst, dir, enable;
4   wire [31:0] o1, o2;
5   bitonic_sorter sorter(
6     .o1(o1),
7     .o2(o2),
8     .A(A),
9     .B(B),
10    .clk(clk),
11    .rst(rst),
12    .dir(dir),
13    .enable(enable)
14  );
15  always begin
16    #5 clk = ~clk;
17  end
18  initial begin
19    A = 0;
20    B = 0;
21    clk = 0;
22    rst = 1;
23    dir = 0;
24    enable = 0;
25    #10 rst = 0;
26    // Test Case 1: Sort in ascending order (dir = 0)
27    A = 5;
28    B = 3;
29    dir = 0;
30    enable = 1;
31    #10;
32    // Verify the output o1 and o2
33    // Test Case 2: Sort in descending order (dir = 1)
34    A = 8;
35    B = 12;
36    dir = 1;
37    enable = 1;
38    #10;
39    // Verify the output o1 and o2
40    // Test Case 3: A and B are equal
41    A = 10;
42    B = 10;
43    dir = 0;
44    enable = 1;
45    #10;
46    // Verify the output o1 and o2
47    // Test Case 4: A and B are equal with descending sort
48    A = 7;
49    B = 7;
50    dir = 1;
51    enable = 1;
52    #10;
53    // Verify the output o1 and o2
54    // Test Case 5: Sort in ascending order with enable=0
55    A = 15;
56    B = 20;
57    dir = 0;
58    enable = 0;
59    #10;
60    // Verify the output o1 and o2
61    // Test Case 6: Sort in descending order with reset
62    A = 3;
63    B = 1;
```

TESTBENCH

1) CAE block

```
64     dir = 1;
65     enable = 1;
66     #10;
67     // Verify the output o1 and o2
68     rst = 1;
69     #10;
70     rst = 0;
71     #10;
72     // Test Case 7: A greater than B, ascending sort
73     A = 25;
74     B = 20;
75     dir = 0;
76     enable = 1;
77     #10;
78     // Verify the output o1 and o2
79     // Test Case 8: B greater than A, descending sort
80     A = 18;
81     B = 30;
82     dir = 1;
83     enable = 1;
84     #10;
85     // Verify the output o1 and o2
86     // Test Case 9: Random values for A and B with enable=1
87     A = 17;
88     B = 28;
89     dir = 0;
90     enable = 1;
91     #10;
92     // Verify the output o1 and o2
93     // Test Case 10: Reset with enable=0
94     A = 10;
95     B = 15;
```

```
96     dir = 0;
97     enable = 0;
98     rst = 1;
99     #10;
100    rst = 0;
101    #10;
102   //Test Case 11: A greater than B, descending sort
103  A = 40;
104  B = 35;
105  dir = 1;
106  enable = 1;
107 #10;
108 // Verify the output o1 and o2
109 // Test Case 12: B greater than A, ascending sort
110 A = 100;
111 B = 150;
112 dir = 0;
113 enable = 1;
114 #10;
115 // Verify the output o1 and o2
116 // Test Case 13: Random values with enable=0
117 A = 72;
118 B = 49;
119 dir = 0;
120 enable = 0;
121 #10;
122 // Verify the output o1 and o2
123 // Test Case 14: Random values with enable=1, then reset
124 A = 31;
125 B = 55;
126 dir = 1;
127 enable = 1;
```

TESTBENCH

1) CAE block

```
128 #10;
129 // Verify the output o1 and o2
130 rst = 1;
131 #10;
132 rst = 0;
133 #10;
134 // Test Case 15: Alternate between ascending and descending sorts
135 A = 30;
136 B = 40;
137 dir = 0;
138 enable = 1;
139 #10;
140 // Verify the output o1 and o2
141 dir = 1;
142 #10;
143 // Verify the output o1 and o2
144 // Test Case 16: Sort with very large values
145 A = 2147483647; // Max positive 32-bit integer
146 B = -2147483648; // Min negative 32-bit integer
147 //B = -32'h80000000;
148 dir = 0;
149 enable = 1;
150 #10;
151 // Verify the output o1 and o2
152 // Test Case 17: Alternating between enable and disable
153 A = 42;
154 B = 18;
155 dir = 0;
156 enable = 1;
157 #10;
158 // Verify the output o1 and o2
159 enable = 0;
```

```
160 #10;
161 enable = 1;
162 #10;
163 // Verify the output o1 and o2
164 // Test Case 18: Large values with descending sort
165 A = 1000000;
166 B = 999999;
167 dir = 1;
168 enable = 1;
169 #10;
170 // Verify the output o1 and o2
171 // Test Case 19: A and B are swapped in descending sort
172 A = 60;
173 B = 75;
174 dir = 1;
175 enable = 1;
176 #10;
177 // Verify the output o1 and o2
178 // Test Case 20: Large negative values with enable=0
179 A = -2147483647; // Max negative 32-bit integer
180 B = -1000000;
181 dir = 0;
182 enable = 0;
183 #10;
184     $finish;
185 end
186 endmodule
```

TESTBENCH

2) 8 input Test Bench

```
1 module cae_tb;
2 reg clk,clks;
3 reg rst;
4 reg en;
5 reg dir;
6 reg [31:0] in1;
7 reg [31:0] in2;
8 reg [31:0] in3;
9 reg [31:0] in4;
10 reg [31:0] in5;
11 reg [31:0] in6;
12 reg [31:0] in7;
13 reg [31:0] in8;
14 wire [31:0] data1;
15 wire [31:0] data2;
16 wire [31:0] data3;
17 wire [31:0] data4;
18 wire [31:0] data5;
19 wire [31:0] data6;
20 wire [31:0] data7;
21 wire [31:0] data8;
22 //wire [31:0] d1;
23 //wire [31:0] d2;
24 //wire [31:0] d3;
25 //wire [31:0] d4;
26 // Instantiate the cae module
27 sorter dut (.clk(clk),.rst(rst),.en(en),.clks(clks),.dir(dir),.in1(in1),.in2(in2),
28 .in3(in3),.in4(in4),.in5(in5),.in6(in6),.in7(in7),.in8(in8),.data1(data1),.data2(data2),
29 .data3(data3),.data4(data4),.data5(data5),.data6(data6),.data7(data7),.data8(data8));
30 // Clock generation
31 always begin
32 #5 clk = ~clk;
33 end
34
35 initial begin
36 clks=1;
37 forever
38 #40 clks=~clks;
39 end
40
41 // Test stimulus
42 initial begin
43 clk = 1;
44 en = 0;
45 dir = 0;
46 in1 = 32'h11011011;
47 in2 = 32'h00101100;
48 in3 = 32'h10101010;
49 in4 = 32'h01100110;
50 in5 = 32'h0;
51 in6 = 32'h2;
52 in7 = 32'h3;
53 in8 = 32'h6;
```

```
59 // Test case 1: Ascending sort
60 en = 1;
61 dir = 0;
62 in1 = 32'h1;
63 in2 = 32'h4;
64 in3 = 32'h5;
65 in4 = 32'h7;
66 in5 = 32'h0;
67 in6 = 32'h2;
68 in7 = 32'h3;
69 in8 = 32'h6;
70 #90;
71 // Test case 2: Ascending sort
72 en = 1;
73 dir = 1;
74 in1 = 32'h1;
75 in2 = 32'h4;
76 in3 = 32'h5;
77 in4 = 32'h7;
78 in5 = 32'h0;
79 in6 = 32'h2;
80 in7 = 32'h3;
81 in8 = 32'h6;
82 #200;
83 // T2
84 // Apply reset
85 rst = 1;
86 #7
87 rst=0;
88 // Test case 3: Ascending sort
89 en = 1;
90 dir = 0;
91 in1 = 32'h6;
92 in2 = 32'h3;
93 in3 = 32'h2;
94 in4 = 32'h0;
95 in5 = 32'h7;
96 in6 = 32'h5;
97 in7 = 32'h4;
98 in8 = 32'h1;
99 #90;
100 // Test case 4: Ascending sort
101 en = 1;
102 dir = 1;
103 in1 = 32'h6;
104 in2 = 32'h3;
105 in3 = 32'h2;
106 in4 = 32'h0;
107 in5 = 32'h7;
108 in6 = 32'h5;
109 in7 = 32'h4;
110 in8 = 32'h1;
111 #200;
```

TESTBENCH

2) 8 input Test Bench

```
146    // Test case 5: Ascending sort
147    en = 1;
148    dir = 0;
149    in1 = 32'b11110000000000000000000000000000;
150    in2 = 32'b00001111000000000000000000000000;
151    in3 = 32'b00000001110000000000000000000000;
152    in4 = 32'b00000000000111000000000000000000;
153    in5 = 32'b00000000000001110000000000000000;
154    in6 = 32'b00000000000000000111000000000000;
155    in7 = 32'b00000000000000000000000001110000;
156    in8 = 32'b00000000000000000000000000001111;
157    #90;
```

```
419    // Test case 6: Descending sort
420    en = 1;
421    dir = 1;
422    in8 = 32'b00000000000000000000000000001111;
423    in7 = 32'b11110000000000000000000000000000;
424    in6 = 32'b00001111000000000000000000000000;
425    in5 = 32'b00000001110000000000000000000000;
426    in4 = 32'b00000000000111000000000000000000;
427    in3 = 32'b00000000000000011100000000000000;
428    in2 = 32'b00000000000000000000000000000000;
429    in1 = 32'b00000000000000000000000000000000;
```

Reports

1) CAE Block

Timing

```
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   (-,-)

#-----
```



Reports

1) CAE Block

Area - with constraints

Applications Places Terminal Thu 13:30 ●

cmos@cad5:synthesis_bitonic_sorter

File Edit View Search Terminal Help

@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>

Reports

1) CAE Block

Area - without constraints

```
=====
Generated by:          Genus(TM) Synthesis Solution 20.11-s111_1
Generated on:          Nov 21 2023 11:32:19 pm
Module:                bitonic_sorter
Operating conditions: _nominal_ (balanced_tree)
Wireload mode:         enclosed
Area mode:             timing library
=====
```

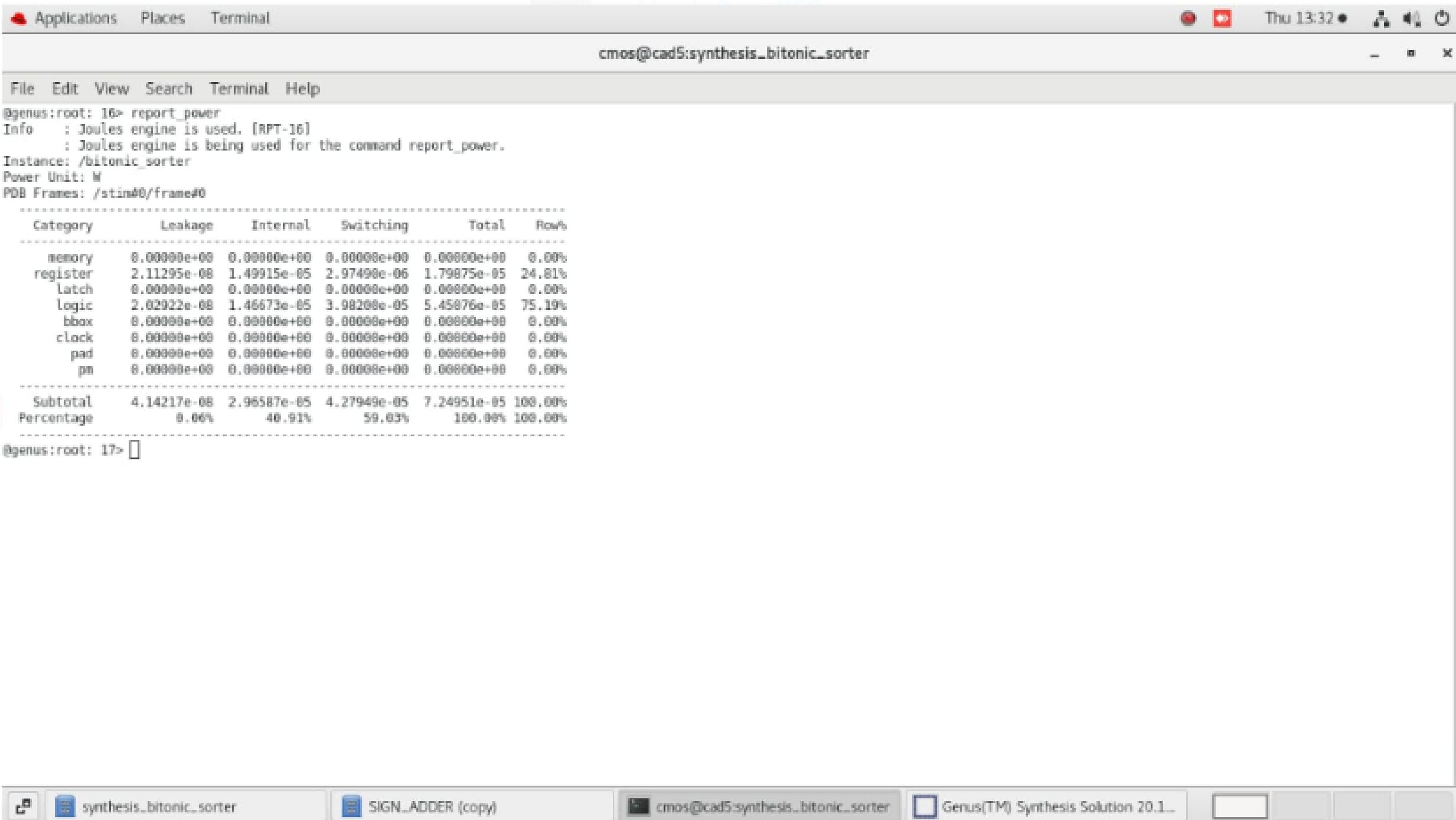
Instance	Module	Cell Count	Cell Area	Net Area	Total Area	Wireload
bitonic_sorter		292	4523.213	0.000	4523.213	<none> (D)

(D) = wireload is default in technology library

Reports

1) CAE Block

Power - with constraint



The screenshot shows a terminal window titled "cmos@cad5:synthesis_bitonic_sorter". The window displays a power consumption report from the Genus tool. The report starts with some informational text and then provides a detailed breakdown of power usage by category.

```
File Edit View Search Terminal Help
@genus:root: 16> report_power
Info   : Joules engine is used. [RPT-16]
        : Joules engine is being used for the command report_power.
Instances: /bitonic_sorter
Power Unit: W
POB Frames: /stim#0/frame#0

Category      Leakage    Internal    Switching     Total    Rows%
-----+-----+-----+-----+-----+-----+
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  5.45876e-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    0.00%      40.91%      59.03%      100.00%     100.00%

@genus:root: 17> 
```

The terminal window also shows the file menu, system status bar, and a toolbar at the bottom with icons for synthesis, simulation, and analysis.

Reports

1) CAE Block

Power - without constraint

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	8.63700e-06	-6.81458e-07	2.22768e-06	1.01832e-05	8.61%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	1.17476e-05	6.98478e-05	2.65588e-05	1.08154e-04	91.39%
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	2.03846e-05	6.91663e-05	2.87865e-05	1.18337e-04	100.00%
Percentage	17.23%	58.45%	24.33%	100.00%	100.00%

Reports

2) 8 Input Bitonic Sorter:

Timing

```

1 =====
2 Generated by: Genus(TM) Synthesis Solution 20.11-s111_1
3 Generated on: Nov 21 2023 12:49:42 pm
4 Module: sorter
5 Operating conditions: slow
6 Interconnect mode: global
7 Area mode: physical library
8 =====
9
10
11 Path 1: MET (183 ps) Setup Check with Pin comp1_m8_reg[20]/CK->D
12     Group: clk
13     Startpoint: (R) in1[0]
14     Clock: (R) clk
15     Endpoint: (R) comp1_m8_reg[20]/D
16     Clock: (R) clk
17
18             Capture           Launch
19     Clock Edge:+    10000            0
20     Drv Adjust:+      0            0
21     Src Latency:+      0            0
22     Net Latency:+      0 (I)        0 (I)
23     Arrival:=    10000            0
24
25     Setup:-        60
26     Uncertainty:-    200
27     Required Time:=  9740
28     Launch Clock:-      0
29     Input Delay:-    300
30     Data Path:-    9257
31     Slack:=        183
32
33 Exceptions/Constraints:
34     input_delay          300
35
36
37
38
39
30 constraintsvbw.sdc_line_26_31_1

```

36 #	37 #	Timing Point	Flags	Arc	Edge	Cell	Fanout	Load (fF)	Trans (ps)	Delay (ps)	Arrival (ps)	Instance Location
38 #	39 #											
40	in1[0]	-	-	R	(arrival)		4	13.0	0	0	300	(-, -)
41	comp1_gt_136_21_Y_lt_166_21_g1172/Y	-	AN->Y	R	NOR2BX1		2	6.9	231	160	460	(-, -)
42	comp1_gt_136_21_Y_lt_166_21_g1100/Y	-	A1N->Y	R	OAI2BB1X1		1	4.4	93	210	670	(-, -)
43	comp1_gt_136_21_Y_lt_166_21_g1099/Y	-	B0->Y	F	OAI12XL		1	4.7	324	210	880	(-, -)
44	comp1_gt_136_21_Y_lt_166_21_g1097/Y	-	A1->Y	R	AOI22X1		1	4.4	183	218	1098	(-, -)
45	comp1_gt_136_21_Y_lt_166_21_g1095/Y	-	A1->Y	F	OAI22XL		1	4.7	345	270	1368	(-, -)
46	comp1_gt_136_21_Y_lt_166_21_g1094/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	1592	(-, -)
47	comp1_gt_136_21_Y_lt_166_21_g1093/Y	-	A1->Y	F	OAI22XL		1	4.7	345	271	1863	(-, -)
48	comp1_gt_136_21_Y_lt_166_21_g1092/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	2087	(-, -)
49	comp1_gt_136_21_Y_lt_166_21_g1091/Y	-	A1->Y	F	OAI22XL		1	4.7	345	271	2358	(-, -)
50	comp1_gt_136_21_Y_lt_166_21_g1090/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	2582	(-, -)
51	comp1_gt_136_21_Y_lt_166_21_g1089/Y	-	A1->Y	F	OAI22XL		1	4.7	345	271	2853	(-, -)
52	comp1_gt_136_21_Y_lt_166_21_g1088/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	3077	(-, -)
53	comp1_gt_136_21_Y_lt_166_21_g1087/Y	-	A1->Y	F	OAI22XL		1	4.7	345	271	3348	(-, -)
54	comp1_gt_136_21_Y_lt_166_21_g1086/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	3572	(-, -)
55	comp1_gt_136_21_Y_lt_166_21_g1085/Y	-	A1->Y	F	OAI22XL		1	4.7	345	271	3843	(-, -)
56	comp1_gt_136_21_Y_lt_166_21_g1084/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	4067	(-, -)
57	comp1_gt_136_21_Y_lt_166_21_g1083/Y	-	A1->Y	F	OAI22XL		1	4.7	345	271	4338	(-, -)
58	comp1_gt_136_21_Y_lt_166_21_g1082/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	4562	(-, -)
59	comp1_gt_136_21_Y_lt_166_21_g1081/Y	-	A1->Y	F	OAI22XL		1	4.7	345	271	4833	(-, -)
60	comp1_gt_136_21_Y_lt_166_21_g1080/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	5057	(-, -)
61	comp1_gt_136_21_Y_lt_166_21_g1079/Y	-	A1->Y	F	OAI22XL		1	4.7	345	271	5328	(-, -)
62	comp1_gt_136_21_Y_lt_166_21_g1078/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	5552	(-, -)
63	comp1_gt_136_21_Y_lt_166_21_g1077/Y	-	A1->Y	F	OAI22XL		1	4.7	345	271	5823	(-, -)
64	comp1_gt_136_21_Y_lt_166_21_g1076/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	6047	(-, -)
65	comp1_gt_136_21_Y_lt_166_21_g1075/Y	-	A1->Y	F	OAI22XL		1	4.7	345	271	6318	(-, -)
66	comp1_gt_136_21_Y_lt_166_21_g1074/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	6542	(-, -)
67	comp1_gt_136_21_Y_lt_166_21_g1073/Y	-	A1->Y	F	OAI22XL		1	4.7	344	271	6813	(-, -)
68	comp1_gt_136_21_Y_lt_166_21_g1072/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	7037	(-, -)
69	comp1_gt_136_21_Y_lt_166_21_g1071/Y	-	A1->Y	F	OAI22XL		1	4.7	344	271	7308	(-, -)
70	comp1_gt_136_21_Y_lt_166_21_g1070/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	7532	(-, -)
71	comp1_gt_136_21_Y_lt_166_21_g1069/Y	-	A1->Y	F	OAI22XL		1	4.7	344	271	7802	(-, -)
72	comp1_gt_136_21_Y_lt_166_21_g1068/Y	-	A1->Y	R	AOI22X1		1	4.4	184	224	8027	(-, -)
73	comp1_gt_136_21_Y_lt_166_21_g1067/Y	-	A1->Y	R	OA21X1		3	10.2	170	246	8272	(-, -)
74	comp1_gt_136_21_Y_lt_166_21_g1066/Y	-	B0->Y	F	AOI2BB1X1		2	7.4	149	139	8412	(-, -)
75	g41226/Y	-	A->Y	R	INVX1		1	4.7	89	96	8507	(-, -)
76	g41168/Y	-	A1->Y	R	AO22X4		64	179.2	685	516	9024	(-, -)
77	g39758_2883/Y	-	A1->Y	F	AOI222X1		1	4.6	306	380	9404	(-, -)
78	g39502/Y	-	A->Y	R	INVX1		1	4.5	117	153	9557	(-, -)
79	comp1_m8_reg[20]/D	<<	-	R	DFFQXL		1	-	-	0	9557	(-, -)

Reports

2) 8 Input Bitonic Sorter:

Area

```
1 =====
2 Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
3 Generated on:           Nov 21 2023 12:49:44 pm
4 Module:                 sorter
5 Operating conditions:   slow
6 Interconnect mode:     global
7 Area mode:              physical library
8 =====
9
10 Instance Module  Cell Count  Cell Area  Net Area    Total Area
11 -----
12 sorter          4048        10399.382  4906.956    15306.338 |
```

Reports

2) 8 Input Bitonic Sorter: Power

2	Power Unit:	W				
3	PDB Frames:	/stim#0/frame#0				

5	Category	Leakage	Internal	Switching	Total	Row%

7	memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
8	register	1.42406e-07	5.60532e-04	1.80549e-04	7.41223e-04	49.45%
9	latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
10	logic	2.48426e-07	2.20435e-04	5.01087e-04	7.21770e-04	48.16%
11	bbox	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
12	clock	0.00000e+00	0.00000e+00	3.58318e-05	3.58318e-05	2.39%
13	pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
14	pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%

16	Subtotal	3.90832e-07	7.80967e-04	7.17468e-04	1.49883e-03	100.00%
17	Percentage	0.03%	52.11%	47.87%	100.00%	100.00%
18	-----					

Advantages

01

Parallel Efficiency: Enables simultaneous processing for faster sorting.

02

Hardware Suitability: Well-adapted for implementation in hardware architectures.

03

Structured Network: Utilizes a structured sorting network, simplifying design.

04

Predictable Performance: Offers consistent performance for real-time applications.

Disadvantages

01

Limited Precision: Optimized for 8-bit data precision.

02

Hardware Complexity: Implementation can be intricate in hardware.

03

Scalability Challenges: Less efficient for significantly larger data sizes.

04

Adaptability Complexity: Modifications for different data sizes may pose challenges.

Applications

- **Digital Signal Processing (DSP):** Efficient for parallel sorting in DSP tasks.
- **Image Processing:** Useful in sorting pixel values in images with 8-bit precision.
- **Sensor Data Processing:** Sorting sensor data with limited precision in real-time.
- **Telecommunications:** Sorting tasks in communication protocols with 8-bit data.



Thank you

References

- 🌐 www.reallygreatsite.com