

## به نام خدا

```
module BitonicSortHypercube(  
    input logic clk,  
    input logic [7:0] data_in [0:15],  
    output logic [7:0] data_out [0:15]  
);  
  
    // Internal registers to hold data  
    logic [7:0] data [0:15];  
  
    logic done1=1;  
    logic done2=0;  
  
    // Main code  
    always@(posedge clk) begin  
        $display("entered initialing");  
        if(done1) begin  
            // Copy input data to internal array  
            data[0] = data_in[0];  
            data[1] = data_in[1];  
            data[2] = data_in[2];  
            data[3] = data_in[3];  
            data[4] = data_in[4];  
            data[5] = data_in[5];  
            data[6] = data_in[6];  
            data[7] = data_in[7];  
            data[8] = data_in[8];  
            data[9] = data_in[9];  
            data[10] = data_in[10];  
            data[11] = data_in[11];  
            data[12] = data_in[12];  
            data[13] = data_in[13];  
            data[14] = data_in[14];  
            data[15] = data_in[15];  
  
            done1 = 0;  
            done2 = 1;  
        end  
    end  
end
```

ماژول ما یک ورودی کلاک و یک ورودی آرایه دارد و یک خروجی آرایه  
سورت شده دارد

در ابتدا همه آرایه ها assign می شوند و بعد بلاک always غیرفعال می شود و بلاک سورت فعال می شود.

از done1 و done2 جهت فعال کردن استفاده شده است.

```
for (integer b = 16; b > 1; b = b / 2) begin
  for (integer i = 0; i < 16; i = i + b) begin
    for (integer j = i; j < i + b / 2; j = j + 1) begin
      if (data[j] > data[j + b / 2]) begin
        temp = data[j];
        data[j] = data[j + b / 2];
        data[j + b / 2] = temp;
      end
    end
  end
end
```

الگوریتم سورت به اینگونه می باشد.

اما از آنجایی که حلقه سنتز نمی شود، کل نتایج با شروطش باز شده است و حدود 180 خط شرط داریم.

در هر شرط نیز با استفاده temp، عملیات swapping به سادگی انجام می شود.

```

23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241

```

```

if (data[0] > data[0]) begin
    temp = data[0];
    data[0] = data[0];
    data[0] = temp;
end
if (data[1] > data[0]) begin
    temp = data[1];
    data[1] = data[0];
    data[0] = temp;
end
if (data[2] > data[10]) begin
    temp = data[2];
    data[2] = data[10];
    data[10] = temp;
end
if (data[3] > data[11]) begin
    temp = data[3];
    data[3] = data[11];
    data[11] = temp;
end
if (data[4] > data[12]) begin
    temp = data[4];
    data[4] = data[12];
    data[12] = temp;
end
if (data[5] > data[13]) begin
    temp = data[5];
    data[5] = data[13];
    data[13] = temp;
end
if (data[6] > data[14]) begin
    temp = data[6];
    data[6] = data[14];
    data[14] = temp;
end
if (data[7] > data[15]) begin
    temp = data[7];
    data[7] = data[15];
    data[15] = temp;
end
end

////////////////////////////////////
if (data[0] > data[4]) begin
    temp = data[0];
    data[0] = data[4];
    data[4] = temp;
end
if (data[1] > data[5]) begin
    temp = data[1];
    data[1] = data[5];
    data[5] = temp;
end
if (data[2] > data[6]) begin
    temp = data[2];
    data[2] = data[6];
    data[6] = temp;
end
if (data[3] > data[7]) begin
    temp = data[3];
    data[3] = data[7];
    data[7] = temp;
end
end

////////////////////////////////////
if (data[0] > data[15]) begin
    temp = data[0];
    data[0] = data[15];
    data[15] = temp;
end
if (data[9] > data[11]) begin
    temp = data[9];
    data[9] = data[11];
    data[11] = temp;
end
if (data[10] > data[14]) begin
    temp = data[10];
    data[10] = data[14];
    data[14] = temp;
end
if (data[11] > data[13]) begin
    temp = data[11];
    data[11] = data[13];
    data[13] = temp;
end
end

////////////////////////////////////
if (data[0] > data[2]) begin
    temp = data[0];
    data[0] = data[2];
    data[2] = temp;
end
if (data[1] > data[3]) begin
    temp = data[1];
    data[1] = data[3];
    data[3] = temp;
end
end

////////////////////////////////////
if (data[4] > data[6]) begin
    temp = data[4];
    data[4] = data[6];
    data[6] = temp;
end
if (data[5] > data[7]) begin
    temp = data[5];
    data[5] = data[7];
    data[7] = temp;
end
end

////////////////////////////////////
if (data[0] > data[15]) begin
    temp = data[0];
    data[0] = data[15];
    data[15] = temp;
end
if (data[9] > data[11]) begin
    temp = data[9];
    data[9] = data[11];
    data[11] = temp;
end
if (data[10] > data[14]) begin
    temp = data[10];
    data[10] = data[14];
    data[14] = temp;
end
if (data[11] > data[13]) begin
    temp = data[11];
    data[11] = data[13];
    data[13] = temp;
end
end

////////////////////////////////////
if (data[0] > data[11]) begin
    temp = data[0];
    data[0] = data[11];
    data[11] = temp;
end
if (data[1] > data[2]) begin
    temp = data[1];
    data[1] = data[2];
    data[2] = temp;
end
end

////////////////////////////////////
if (data[2] > data[3]) begin
    temp = data[2];
    data[2] = data[3];
    data[3] = temp;
end
end

////////////////////////////////////
if (data[4] > data[5]) begin
    temp = data[4];
    data[4] = data[5];
    data[5] = temp;
end
end

////////////////////////////////////
if (data[6] > data[7]) begin
    temp = data[6];
    data[6] = data[7];
    data[7] = temp;
end
end

////////////////////////////////////
if (data[0] > data[5]) begin
    temp = data[0];
    data[0] = data[5];
    data[5] = temp;
end
end

////////////////////////////////////
if (data[10] > data[11]) begin
    temp = data[10];
    data[10] = data[11];
    data[11] = temp;
end
end

////////////////////////////////////
if (data[12] > data[13]) begin
    temp = data[12];
    data[12] = data[13];
    data[13] = temp;
end
end

////////////////////////////////////
if (data[14] > data[15]) begin
    temp = data[14];
    data[14] = data[15];
    data[15] = temp;
end
end

////////////////////////////////////

```

## شبیه سازی/simulation

تست بنچ به اینگونه هست:

```
`include "BitonicSortHypercube.sv"
module tb_BitonicSortHypercube;

    // Inputs
    logic clk=0;
    logic [7:0] data_in[0:15];

    // Outputs
    logic [7:0] data_out[0:15];

    // Instantiate the DUT
    BitonicSortHypercube dut (
        .clk(clk),
        .data_in(data_in),
        .data_out(data_out)
    );

    // Clock generation
    always #5 clk = ~clk;

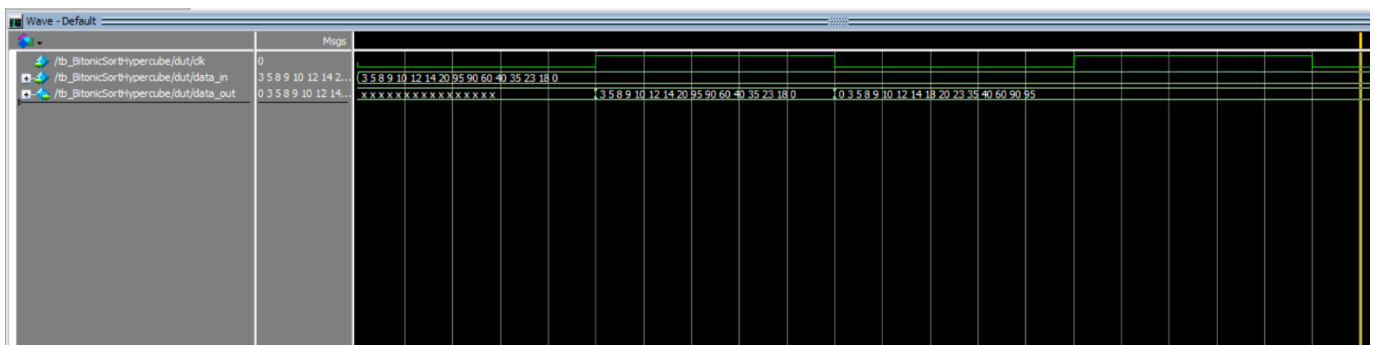
    // Stimulus
    initial begin
        // Initialize inputs
        data_in = {7'h03,7'h05, 7'h08, 7'h09,7'h0A,7'h0C, 7'h0E, 7'h14,7'h5F, 7'h5A,7'h3C,7'h28, 7'h23, 7'h17,7'h12, 7'h00};

        // Wait for sorting process to complete
        #100;

        // End simulation
        $finish;
    end
endmodule
```

طبق صورت سوال تمامی المان های آرایه assign شده اند.

موج:



به ترتیب از چپ به راست برای پایین ترین موج داریم:  
ابتدا آرایه لود می شود و از حالت x مجهول خارج می شود.  
در نهایت کاملاً صورت شده است

# سنتز/synthesis

The screenshot displays the Vivado 2019.2 IDE interface during the synthesis phase of a project named 'project\_1'. The top menu bar includes File, Edit, Flow, Tools, Reports, Window, Layout, View, and Help. The top status bar indicates 'Synthesis Complete' with a green checkmark.

The left sidebar, titled 'Flow Navigator', shows the project's workflow. The 'SYNTHESIS' section is expanded, highlighting 'Run Synthesis' and 'Open Synthesized Design'. Other sections include 'Settings', 'IP INTEGRATOR', 'SIMULATION', and 'RTL ANALYSIS'.

The main workspace is divided into several panes:

- Sources:** Shows a 'Netlist' tab with an '(empty)' netlist.
- Properties:** A pane for viewing object properties, currently showing 'Select an object to see properties'.
- Project Summary:** A pane for project details.
- Device:** A pane for device configuration.
- Tcl Console:** A pane for running Tcl commands and viewing logs. It displays the following output:

```
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00 . Memory (MB): peak = 1999.090 ; gain = 0.000
INFO: [Project 1-479] Netlist was created with Vivado 2019.2
INFO: [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00 . Memory (MB): peak = 1999.090 ; gain = 0.000
INFO: [Project 1-111] Unisim Transformation Summary:
No Unisim elements were transformed.
open_run: Time (s): cpu = 00:00:12 ; elapsed = 00:00:13 . Memory (MB): peak = 2482.742 ; gain = 1453.473
```

The bottom status bar shows 'Type a Tcl command here'.