CSEE4824 Computer Architecture Project1

Zhongheng Cheng (zc2737)

4-bit Combinational Priority Selector

ps4-assign

To use "assign" for constructing ps4, assign every digit of gnt seperately.

project1 > source > ps4-assign.sv

```
1
2   module ps4 (
3       input        [3:0] req,
4       input              en,
5       output logic [3:0] gnt
6   );
7
8       assign gnt[3] = en & req[3];
9       assign gnt[2] = en & req[2] & ~req[3];
10      assign gnt[1] = en & req[1] & ~req[2] & ~req[3];
11      assign gnt[0] = en & req[0] & ~req[1] & ~req[2] & ~req[3];
12
13
14  endmodule
```

ps4-if_else

To use "if_else" for constructing ps4, assign every value by conditions inside "always_comb".

project1 > source > ps4-if_else.sv

```
1
2   module ps4 (
3       input        [3:0] req,
4       input              en,
5       output logic [3:0] gnt
6   );
7
8       always_comb begin
9           gnt = 4'b0000;
10          if (en) begin
11              if (req[3])
12                  gnt = 4'b1000;
13              else if (req[2] & ~req[3])
14                  gnt = 4'b0100;
15              else if (req[1] & ~req[2] & ~req[3])
16                  gnt = 4'b0010;
17              else if (req[0] & ~req[1] & ~req[2] & ~req[3])
18                  gnt = 4'b0001;
19          end
20      end
21
22
23  endmodule
```

ps4_testing

The testbench is passed for all cases.

```
Chronologic VCS simulator copyright 1991-2019
Contains Synopsys proprietary information.
Compiler version P-2019.06_Full64; Runtime version P-2019.06_Full64;  Feb  6 13:20 2024
Time:   0 req:0000 en:1 gnt:0000
Time:   5 req:1000 en:1 gnt:1000
Time:  10 req:0100 en:1 gnt:0100
Time:  15 req:0010 en:1 gnt:0010
Time:  20 req:0001 en:1 gnt:0001
Time:  25 req:0101 en:1 gnt:0100
Time:  30 req:0110 en:1 gnt:0100
Time:  35 req:1110 en:1 gnt:1000
Time:  40 req:1111 en:1 gnt:1000
Time:  45 req:1111 en:0 gnt:0000
Time:  50 req:0110 en:0 gnt:0000
@@@ Passed
```

Hierarchical Priority Selectors

ps8

To build a ps8, we can build a ps2 firstly.

```
 2    module ps2 (
 3        input        [1:0] req,
 4        input              en,
 5        output logic [1:0] gnt,
 6        output logic       req_up
 7    );
 8
 9        assign req_up = |req;
10        assign gnt[1] = en & req[1];
11        assign gnt[0] = en & req[0] & ~req[1];
12
13    endmodule
```

Then use the ps2 to build a ps4, which is made of two ps2s.

```
16  module ps4 (
17      input         [3:0] req,
18      input               en,
19      output logic [3:0] gnt,
20      output logic       req_up
21  );
22
23      logic req_high;
24      logic req_low;
25
26      ps2 high(
27          .req(req[3:2]),
28          .en(en),
29          .gnt(gnt[3:2]),
30          .req_up(req_high)
31      );
32
33      ps2 low(
34          .req(req[1:0]),
35          .en(en & ~req_high),
36          .gnt(gnt[1:0]),
37          .req_up(req_low)
38      );
39
40      assign req_up = req_high | req_low;
```

Finally, use the ps4 to build a ps8, similiar to the previous step.

```
46  module ps8 (
47      input         [7:0] req,
48      input               en,
49      output logic [7:0] gnt,
50      output logic       req_up
51
52  );
53
54      logic req_high;
55      logic req_low;
56
57      ps4 high(
58          .req(req[7:4]),
59          .en(en),
60          .gnt(gnt[7:4]),
61          .req_up(req_high)
62      );
63
64      ps4 low(
65          .req(req[3:0]),
66          .en(en & ~req_high),
67          .gnt(gnt[3:0]),
68          .req_up(req_low)
69      );
70
71      assign req_up = req_high | req_low;
72
```

```
73
74      endmodule
```

ps8_testing

I selected some of the cases and wrote testcases for ps8. All the cases are passed.

```
Chronologic VCS simulator copyright 1991-2019
Contains Synopsys proprietary information.
Compiler version P-2019.06_Full64; Runtime version P-2019.06_Full64;  Feb  6 13:06 2024
Time:   0 req:00000000 en:1 gnt:00000000
Time:   5 req:10000000 en:1 gnt:10000000
Time:  10 req:01000000 en:1 gnt:01000000
Time:  15 req:00100000 en:1 gnt:00100000
Time:  20 req:00010000 en:1 gnt:00010000
Time:  25 req:00001000 en:1 gnt:00001000
Time:  30 req:00000100 en:1 gnt:00000100
Time:  35 req:00000010 en:1 gnt:00000010
Time:  40 req:00000001 en:1 gnt:00000001
Time:  45 req:11010000 en:1 gnt:10000000
Time:  50 req:01101000 en:1 gnt:01000000
Time:  55 req:00110100 en:1 gnt:00100000
Time:  60 req:00011010 en:1 gnt:00010000
Time:  65 req:00001101 en:1 gnt:00001000
Time:  70 req:11111110 en:1 gnt:10000000
Time:  75 req:11111111 en:1 gnt:10000000
Time:  80 req:11111111 en:0 gnt:00000000
Time:  85 req:00011010 en:0 gnt:00000000
@@@ Passed
$finish called from file "ps8_test.sv", line 62.
$finish at simulation time                   90
          V C S   S i m u l a t i o n   R e p o r t
Time: 90
CPU Time:      0.410 seconds;       Data structure size:   0.0Mb
Tue Feb  6 13:06:18 2024
output saved to program.out
```

Hierarchical Rotating Priority Selectors

rps4

To build a rps4, build a rps2 first.

```
module rps2 (
    input               sel,
    input         [1:0] req,
    input               en,

    output logic [1:0] gnt,
    output logic       req_up
);

    always_comb begin
        req_up = |req;
        if (~en) gnt = 2'b00;
        else begin
            gnt = req;
            if (req == 2'b11)
                gnt = (sel == 1) ? 2'b10 : 2'b01;
        end
    end

endmodule
```

Then use three rps2 modules according to the model shown in the instructions.

```
34      always_ff @(posedge clock) begin
35          if (reset) count <= 2'b00;
36          else begin
37              if (count == 2'b11) count <= 2'b00;
38              else count <= count + 1'b1;
39          end
40      end
41
42      logic [1:0] req_up_temp;
43      logic [1:0] gnt_temp;
44      logic req_up;
45
46      rps2 left(
47          .req(req[3:2]),
48          .en(gnt_temp[1]),
49          .sel(count[0]),
50          .gnt(gnt[3:2]),
51          .req_up(req_up_temp[1])
52      );
53
54      rps2 right(
55          .req(req[1:0]),
56          .en(gnt_temp[0]),
57          .sel(count[0]),
58          .gnt(gnt[1:0]),
59          .req_up(req_up_temp[0])
60      );
61
62      rps2 top(
63          .req(req_up_temp[1:0]),
64          .en(en),
```

```
65          .sel(count[1]),
66          .gnt(gnt_temp[1:0]),
67          .req_up(req_up)
68      );
```

rps4_testing

All tests are passed.

```
Chronologic VCS simulator copyright 1991-2019
Contains Synopsys proprietary information.
Compiler version P-2019.06_Full64; Runtime version P-2019.06_Full64;  Feb  5 20:17 2024
Time:   0 req:xxxx en:x gnt:xxxx, cnt:xx
Time:   5 req:xxxx en:x gnt:xxxx, cnt:00
Time:   6 req:0001 en:1 gnt:0001, cnt:00
Time:  15 req:0001 en:1 gnt:0001, cnt:01
Time:  16 req:0010 en:1 gnt:0010, cnt:01
Time:  25 req:0010 en:1 gnt:0010, cnt:10
Time:  26 req:0101 en:1 gnt:0100, cnt:10
Time:  35 req:0101 en:1 gnt:0100, cnt:11
Time:  36 req:0011 en:1 gnt:0010, cnt:11
Time:  45 req:0011 en:1 gnt:0001, cnt:00
Time:  46 req:1111 en:1 gnt:0001, cnt:00
Time:  55 req:1111 en:1 gnt:0010, cnt:01
Time:  65 req:1111 en:1 gnt:0100, cnt:10
Time:  75 req:1111 en:1 gnt:1000, cnt:11
Time:  85 req:1111 en:1 gnt:0001, cnt:00
Time:  86 req:1111 en:0 gnt:0000, cnt:00
Time:  95 req:1111 en:0 gnt:0000, cnt:01
Time: 105 req:1111 en:0 gnt:0000, cnt:10
@@@ Passed
$finish called from file "rps4_test.sv", line 95.
$finish at simulation time                  106
          V C S   S i m u l a t i o n   R e p o r t
Time: 106
CPU Time:      0.430 seconds;       Data structure size:   0.0Mb
Mon Feb  5 20:17:05 2024
output saved to program.out
```