

Họ tên: Dương Anh Khôi

MSSV: 22520696

Bài làm

Sử dụng Verilog HDL thiết kế mạch full adder 4 bits bằng cách:

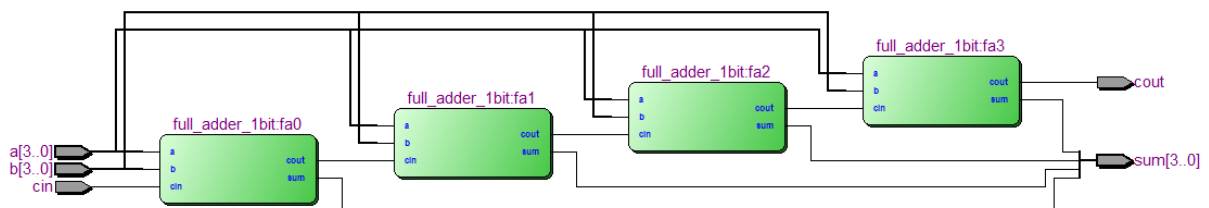
a. Sử dụng mô hình thiết kế cấu trúc

```
1  module full_adder_1bit(  
2      input a,  
3      input b,  
4      input cin,  
5      output sum,  
6      output cout  
7  );  
8      wire s1, c1, c2;  
9  
10     xor x1 (s1, a, b);  
11     xor x2 (sum, s1, cin);  
12  
13     and a1 (c1, a, b);  
14     and a2 (c2, s1, cin);  
15  
16     or o1 (cout, c1, c2);  
17 endmodule  
18  
19 module full_adder_4bit(  
20     input [3:0] a,  
21     input [3:0] b,  
22     input cin,  
23     output [3:0] sum,  
24     output cout  
25 );  
26     wire c1, c2, c3;  
27  
28     full_adder_1bit fa0(  
29         .a(a[0]),  
30         .b(b[0]),  
31         .cin(cin),
```

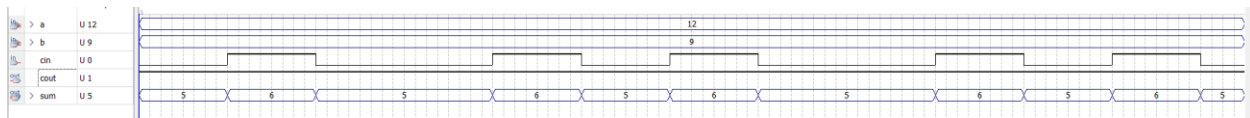
```

30         .b(b[0]),
31         .cin(cin),
32         .sum(sum[0]),
33         .cout(c1)
34     );
35
36     full_adder_1bit fa1(
37         .a(a[1]),
38         .b(b[1]),
39         .cin(c1),
40         .sum(sum[1]),
41         .cout(c2)
42     );
43
44     full_adder_1bit fa2(
45         .a(a[2]),
46         .b(b[2]),
47         .cin(c2),
48         .sum(sum[2]),
49         .cout(c3)
50     );
51
52     full_adder_1bit fa3(
53         .a(a[3]),
54         .b(b[3]),
55         .cin(c3),
56         .sum(sum[3]),
57         .cout(cout)
58     );
59 endmodule

```



Mô phỏng:

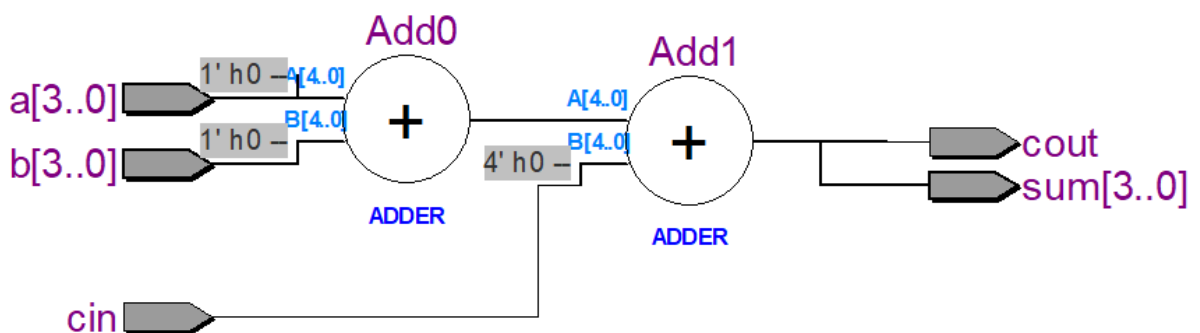


b. Sử dụng mô hình thiết kế hành vi với phép gán liên tục

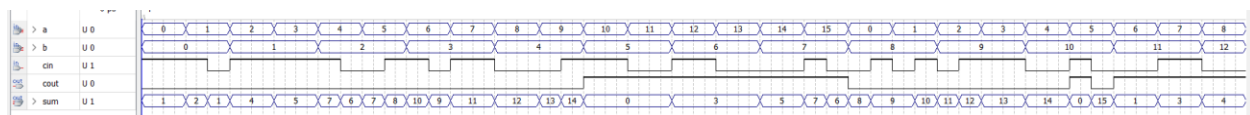
```

1  module full_adder_4bit_behavioral(
2      input [3:0] a,
3      input [3:0] b,
4      input cin,
5      output [3:0] sum,
6      output cout
7  );
8      wire [4:0] temp_sum;
9
10     assign temp_sum = a + b + cin;
11
12     assign sum = temp_sum[3:0];
13     assign cout = temp_sum[4];
14
15 endmodule

```



Mô phỏng:

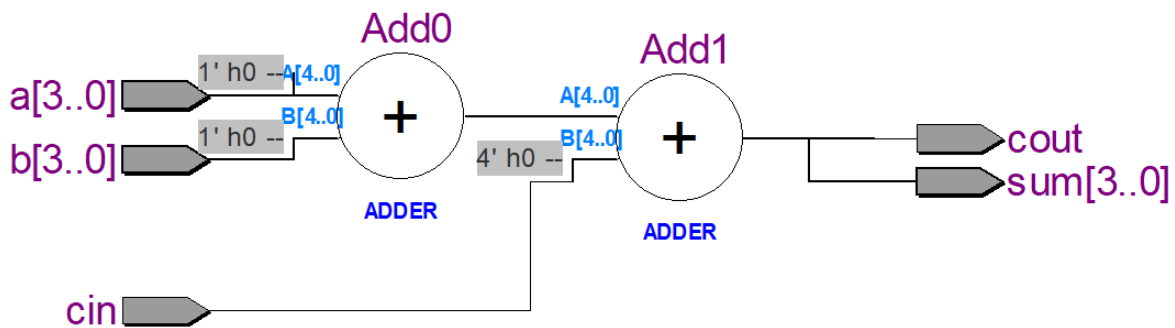


c. Sử dụng mô hình thiết kế hành vi với phép gán qui trình

```

1  module full_adder_4bit_procedural(
2      input [3:0] a,
3      input [3:0] b,
4      input cin,
5      output reg [3:0] sum,
6      output reg cout
7  );
8      always @(a or b or cin) begin
9          {cout, sum} = a + b + cin;
10     end
11 endmodule

```



Mô phỏng:

