

```

1  `timescale 1ns / 1ns // `timescale time_unit/time_precision
2
3  // Full adder for 2 4-bit numbers
4  module fullAdder4bit(input [9:0] SW, output [9:0] LEDR);
5      wire [3:0] A,B,S; // I/O
6      wire Cin,Cout; // I/O wires
7      wire C1,C2,C3; // Intermediary wires
8
9      // Input assignments
10     assign A = SW[3:0];
11     assign B = SW[7:4];
12     assign Cin = SW[8];
13
14     adder bit0(.a(A[0]),.b(B[0]),.s(S[0]),.cin(Cin),.cout(C1));
15     adder bit1(.a(A[1]),.b(B[1]),.s(S[1]),.cin(C1),.cout(C2));
16     adder bit2(.a(A[2]),.b(B[2]),.s(S[2]),.cin(C2),.cout(C3));
17     adder bit3(.a(A[3]),.b(B[3]),.s(S[3]),.cin(C3),.cout(Cout));
18
19     // Outputs
20     assign LEDR[3:0] = S;
21     assign LEDR[9] = Cout;
22 endmodule
23
24 // Adder bit slice
25 // digit = a XOR b XOR cin, carry = a*b+a*cin+b*cin
26 module adder(input a,b,cin, output s,cout);
27     assign s = a ^ b ^ cin;
28     assign cout = (a&b) | (a&cin) | (b&cin);
29 endmodule
30

```