

```

Ln#
1 module full_subtractor(a,b,bin,diff,bout);
2   input a;           // Minuend
3   input b;           // Subtrahend
4   input bin;         // Borrow-in
5   output diff;       // Difference
6   output bout;       // Borrow-out
7
8   assign diff = a ^ b ^ bin;           // Difference calculation using XOR
9   assign bout = (~a & b) | (bin & ~a) | (bin & b); // Borrow-out calculation
10
11 endmodule
12

```

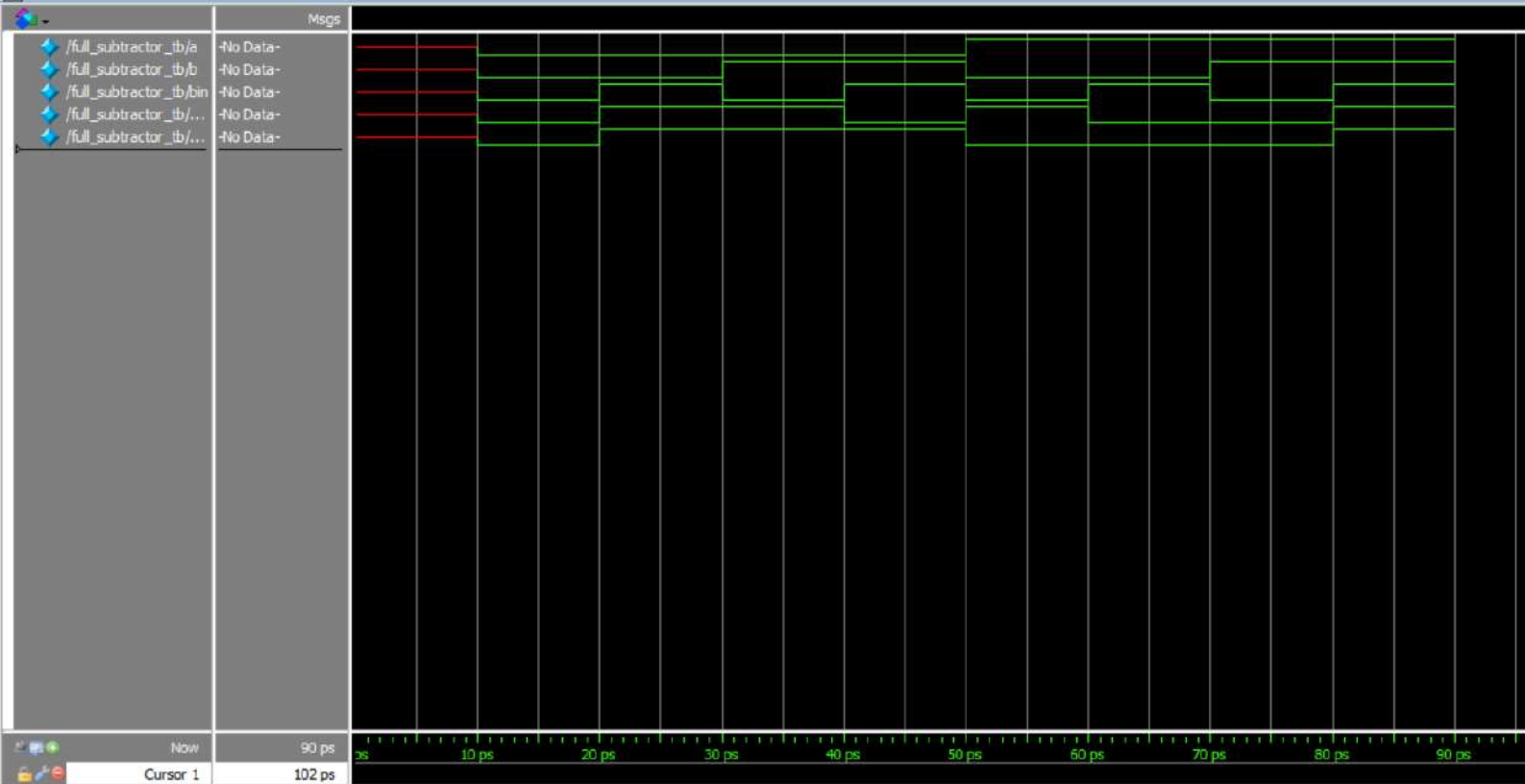
```
Ln#
1 module full_subtractor_tb;
2
3 // Testbench signals
4 reg a; // Testbench minuend input
5 reg b; // Testbench subtrahend input
6 reg bin; // Testbench borrow-in input
7 wire diff; // Testbench difference output
8 wire bout; // Testbench borrow-out output
9
10 // Instantiate the full subtractor
11 full_subtractor uut (
12     .a(a),
13     .b(b),
14     .bin(bin),
15     .diff(diff),
16     .bout(bout)
17 );
18
19 // Test sequence
20 initial
21 begin
22     $monitor("At time %t: a = %b, b = %b, bin = %b, diff = %b, bout = %b", $time, a, b, bin, diff, bout);
23     // Initialize inputs and apply test cases
24     #10 a = 0; b = 0; bin = 0; // Test case 1: a = 0, b = 0, bin = 0
25     #10 a = 0; b = 0; bin = 1; // Test case 2: a = 0, b = 0, bin = 1
26     #10 a = 0; b = 1; bin = 0; // Test case 3: a = 0, b = 1, bin = 0
27     #10 a = 0; b = 1; bin = 1; // Test case 4: a = 0, b = 1, bin = 1
28     #10 a = 1; b = 0; bin = 0; // Test case 5: a = 1, b = 0, bin = 0
29     #10 a = 1; b = 0; bin = 1; // Test case 6: a = 1, b = 0, bin = 1
30     #10 a = 1; b = 1; bin = 0; // Test case 7: a = 1, b = 1, bin = 0
31     #10 a = 1; b = 1; bin = 1; // Test case 8: a = 1, b = 1, bin = 1
32     #10 $finish;
33 end
34 endmodule
35
```



ColumnLayout Default



Wave - Default



Now 90 ps
Cursor 1 102 ps