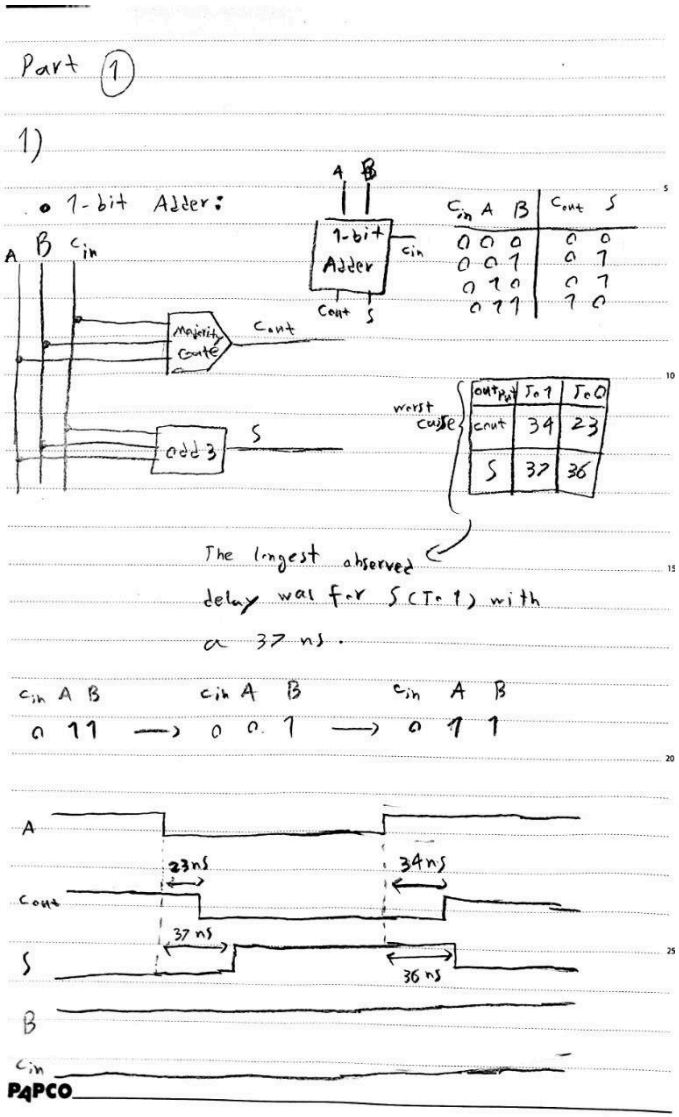
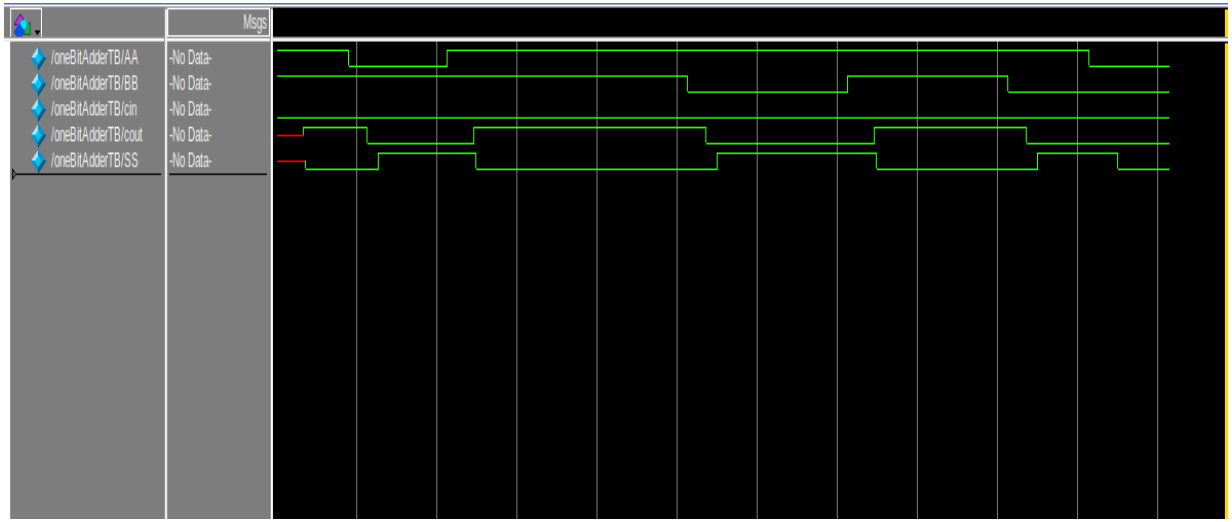
Eran Falahati 810102491 CA2-Fall403 course-id = 4031810189401

.1_bit Adder

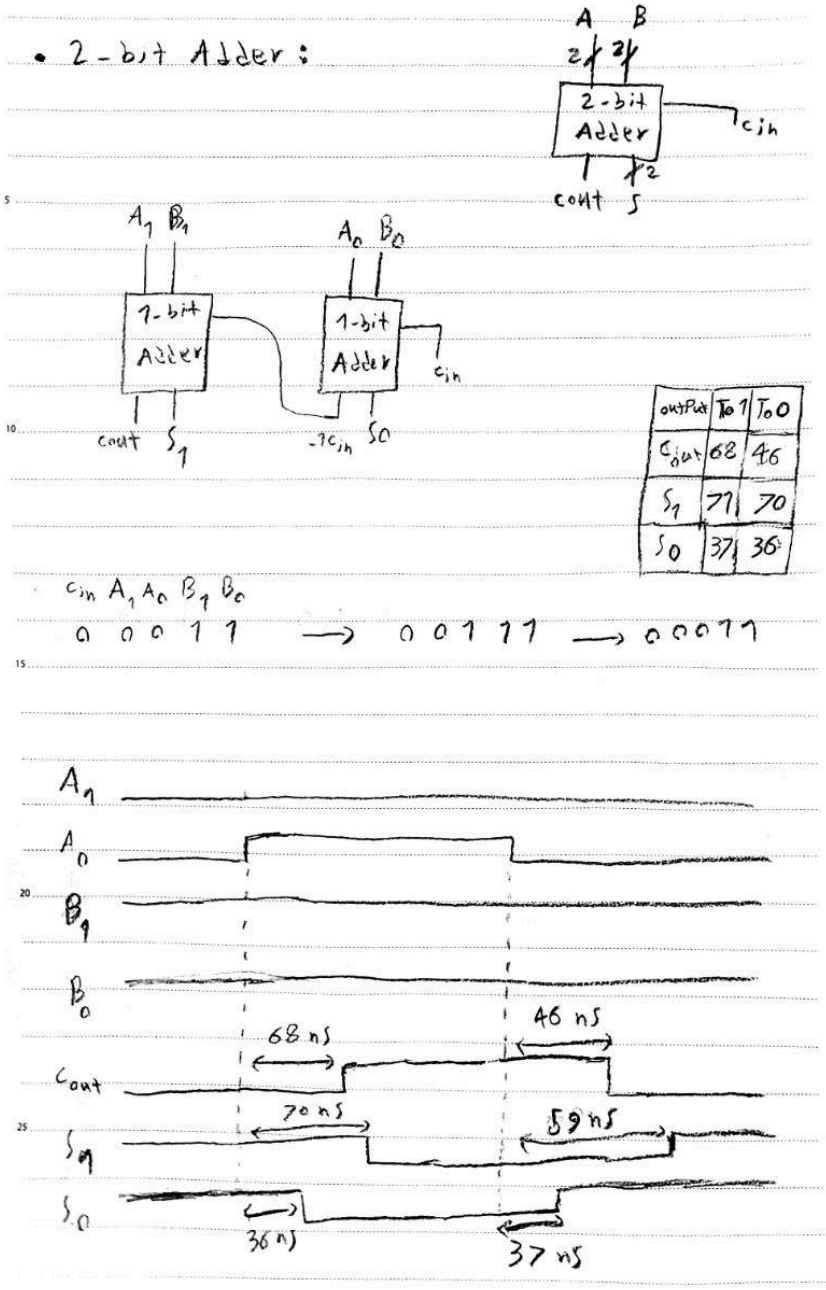




```
4   module _1BitAdder(input A,B,cin, output S,cout);
5
6       onesCounter onesCounter(A,B,cin, cout,S);
7
8   endmodule
```

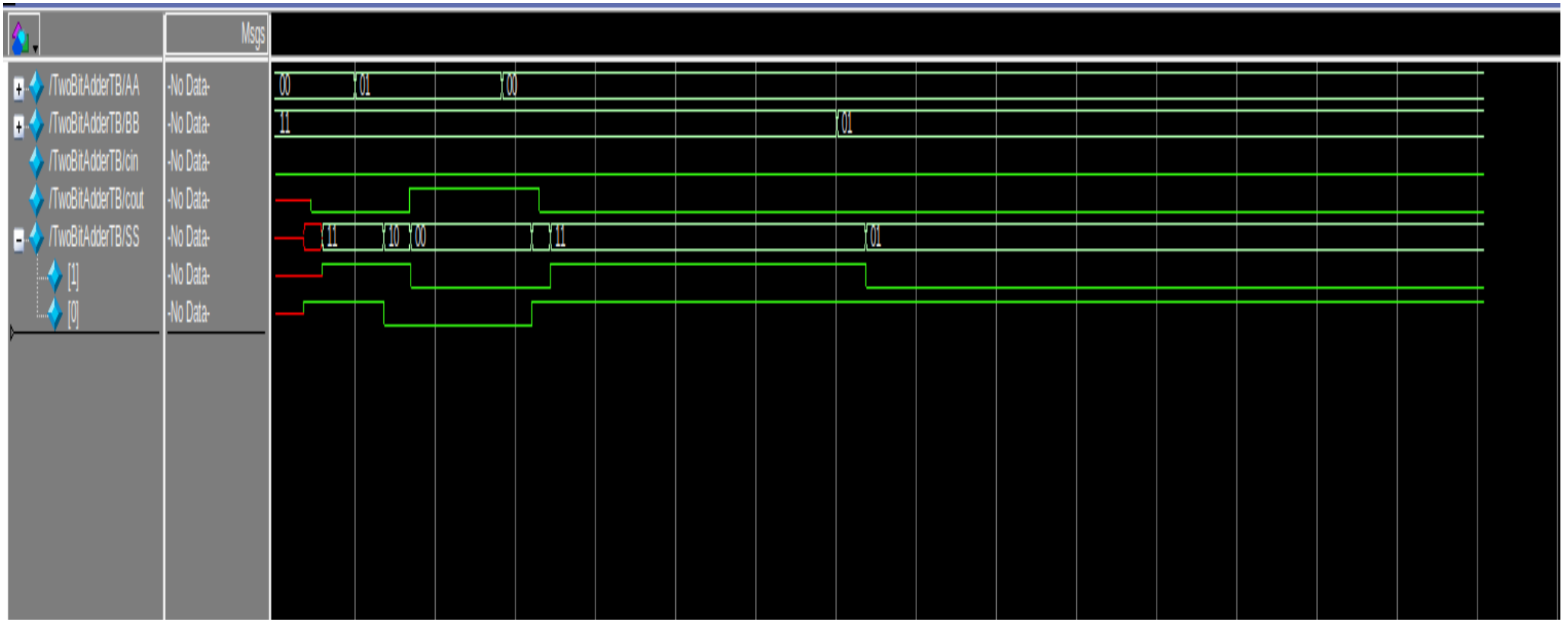
```
4   module oneBitAdderTB();
5
6       logic AA = 1'b1, BB = 1'b1, cin = 1'b0;
7       wire cout, SS;
8
9       _1BitAdder OBA(AA, BB, cin, SS, cout);
10
11      initial begin
12          #90 AA = 0;
13          #67
14          #56 AA = 1;
15          #100
16          repeat(3) #200 BB = ~BB;
17          #76
18          #26 AA = 0;
19          #100 $stop;
20      end
21
22  endmodule
```

. 2-bit Adder

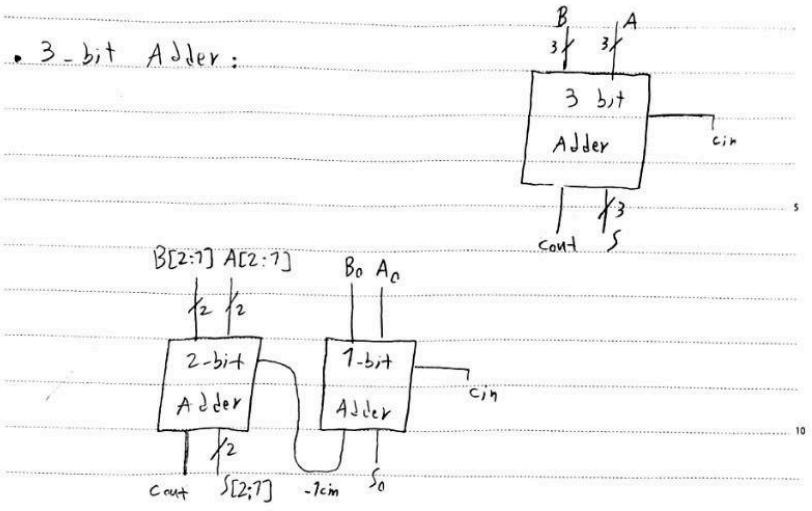

• 2-bit Adder :
A B
2 2
2-bit Adder
cin
cout S
2
A1 B1
A0 B0
1-bit Adder
1-bit Adder
cin
cout S1
-Tcin S0
output T01 T00
Cout 68 46
S1 71 70
S0 37 36
cin A1 A0 B1 B0
0 0 0 1 1 → 0 0 1 1 1 → 0 0 0 1 1
A1
A0
B1
B0
68 ns
46 ns
Cout
70 ns
59 ns
S1
S0
36 ns
37 ns

```
11  module _2BitAdder(input [1:0] A,B, input cin, output [1:0] S, output cout);
12
13      wire _1cin;
14      _1BitAdder OC1(A[0],B[0],cin,S[0],_1cin);
15      _1BitAdder OC2(A[1],B[1],_1cin,S[1],cout);
16
17  endmodule
```

```
25  module TwoBitAdderTB();
26
27      logic [1:0] AA = 2'b00;
28      logic [1:0] BB= 2'b11;
29      logic cin = 0;
30      wire [1:0]SS;
31      wire cout;
32
33      _2BitAdder TwoBA(AA,BB,cin,SS,cout);
34
35      initial begin
36          #100 AA[0] = 1;
37          #150
38          #34 AA[0] = 0;
39          #168
40          repeat(4) #250 BB[1] = ~BB[0];
41          #56 $stop;
42      end
43
44  endmodule
```


Msgs
/TwoBitAdderTB/AA
/TwoBitAdderTB/BB
/TwoBitAdderTB/cin
/TwoBitAdderTB/cout
/TwoBitAdderTB/SS
[1]
[0]
-No Data-
00
01
00
11
01
11
10
00
11
01

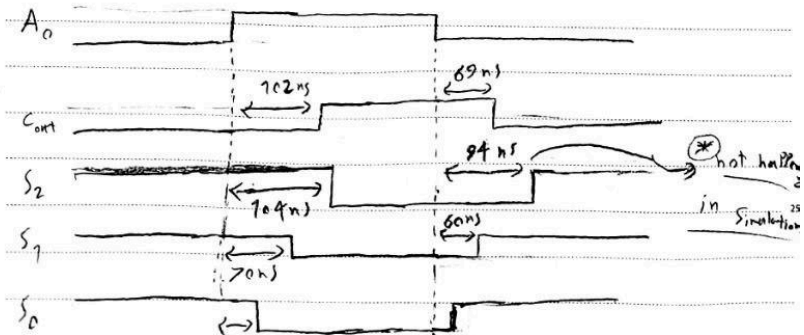
. 3-bit Adder



outBit	To 1	To 0
C_{out}	102	69
S_2	105	104
S_1	71	70
S_0	37	36

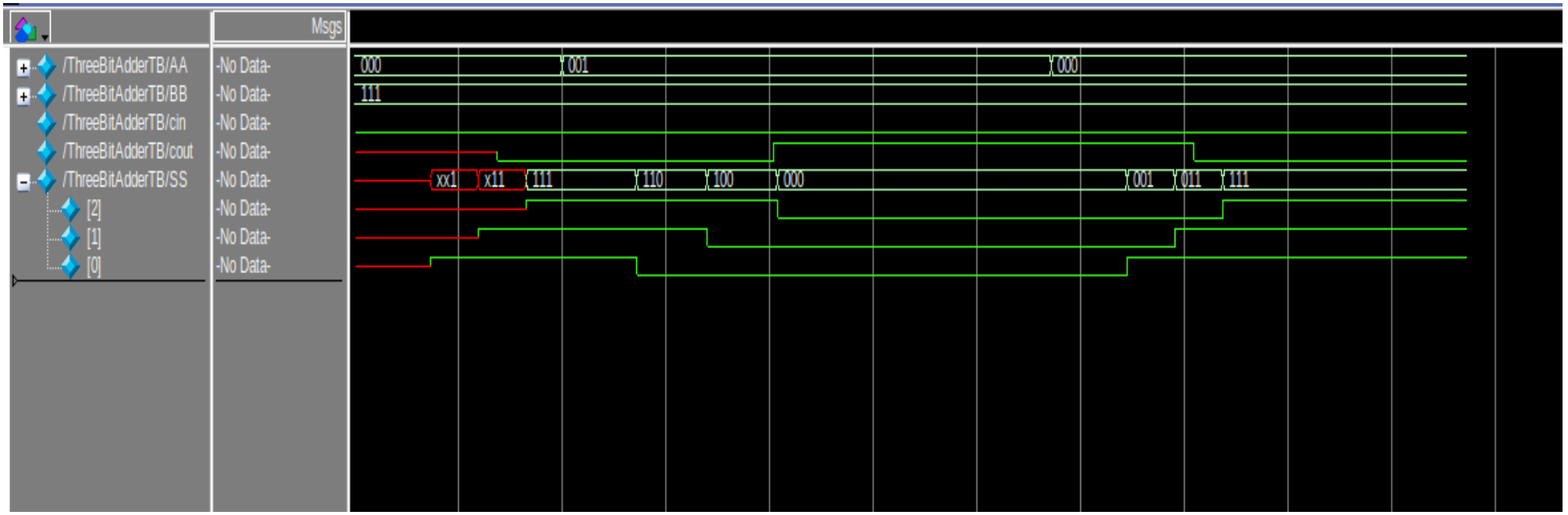
cin A B

0 000 1 11 → 00001111 → 00000111

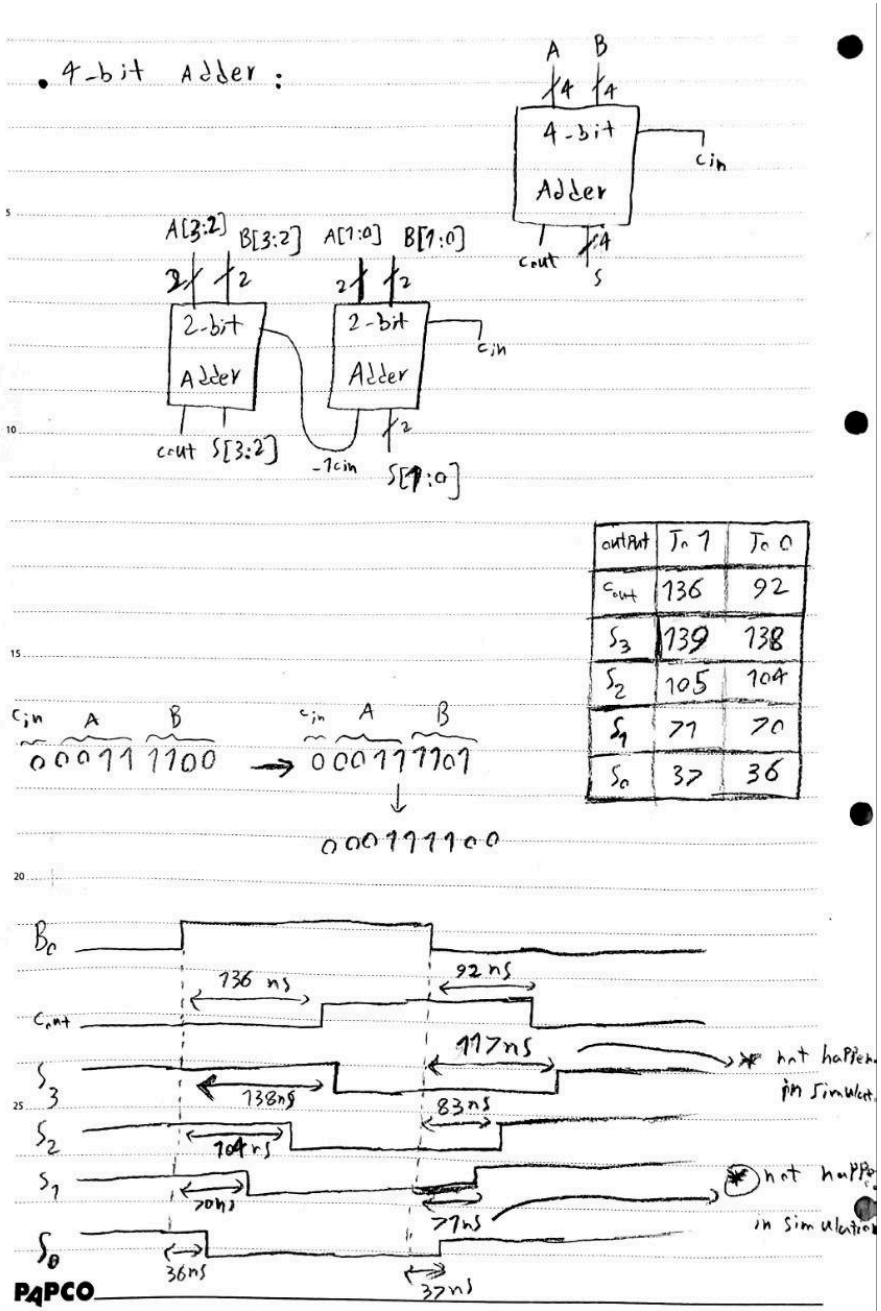


```
19  module _3BitAdder(input [2:0] A,B, input cin, output [2:0] S, output cout);
20
21      wire _1cin;
22      _1BitAdder OneBA(A[0],B[0],cin,S[0],_1cin);
23      _2BitAdder TwoBA(A[2:1],B[2:1],_1cin,S[2:1],cout);
24
25  endmodule
```

```
46  module ThreeBitAdderTB();
47
48      logic [2:0] AA = 3'b000;
49      logic [2:0] BB = 3'b111;
50      logic cin = 0;
51      wire [2:0] SS;
52      wire cout;
53
54      _3BitAdder ThreeBA(AA,BB,cin,SS,cout);
55
56      initial begin
57          #100 AA[0] = 1;
58          #180
59          #56 AA[0] = 0;
60          #200
61          $stop;
62      end
63
64  endmodule
```

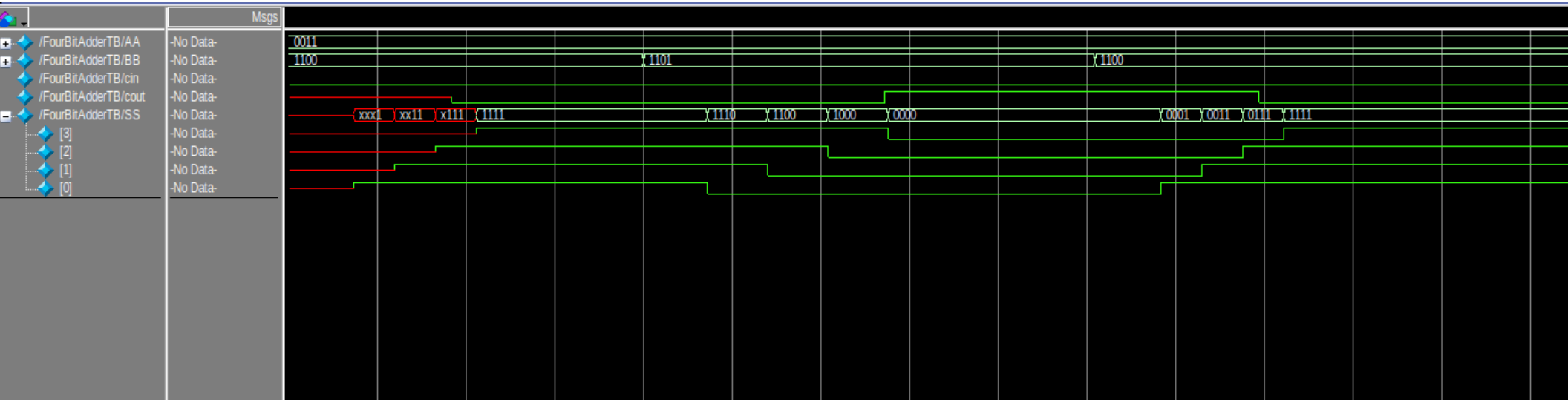

Msgs
/ThreeBitAdderTB/AA
/ThreeBitAdderTB/BB
/ThreeBitAdderTB/cin
/ThreeBitAdderTB/cout
/ThreeBitAdderTB/SS
[2]
[1]
[0]
-No Data-
000
001
000
111
xx1
x11
111
110
100
000
001
011
111

. 4-bit Adder

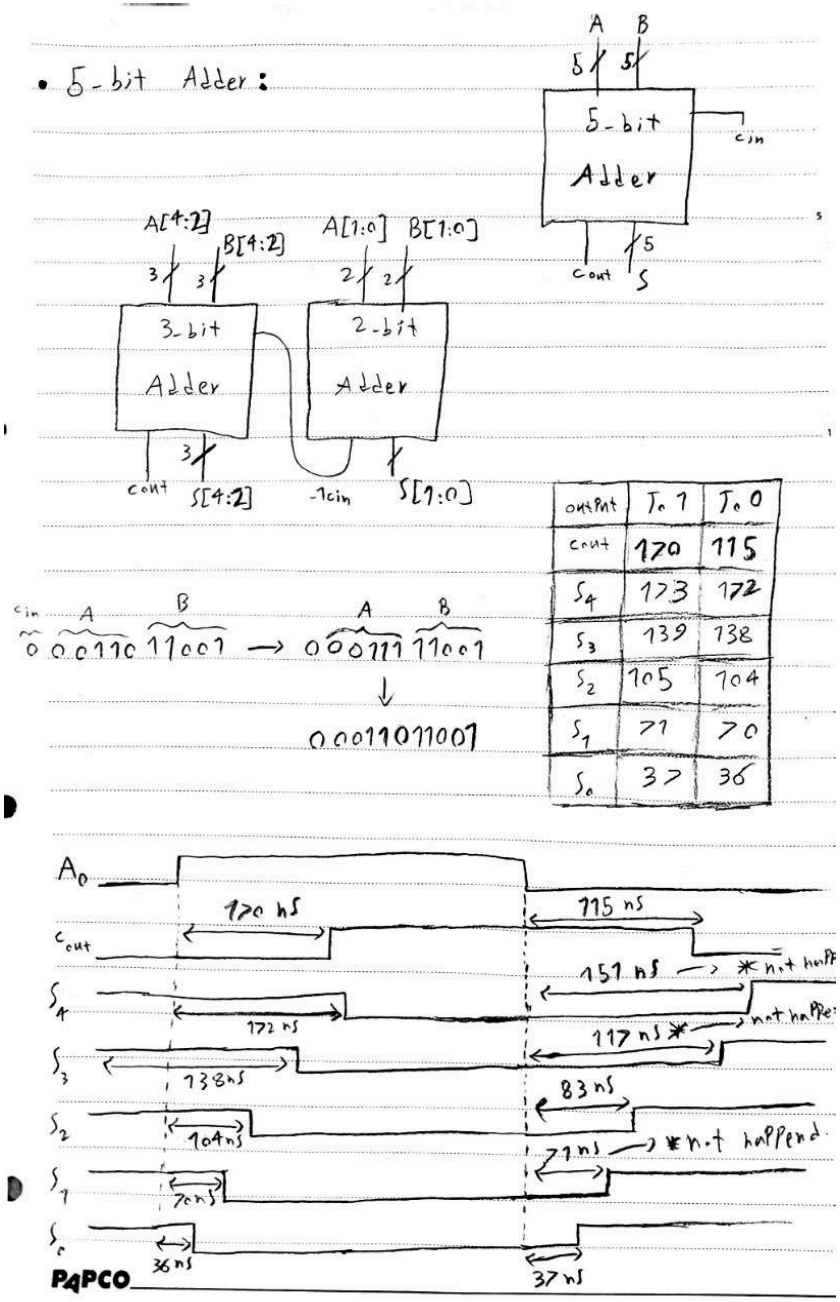

• 4-bit Adder :
A B
4 4
4-bit
Adder
cin
cout S
A[3:2] B[3:2] A[1:0] B[1:0]
2 2 2 2
2-bit
Adder
2-bit
Adder
cout S[3:2]
-1cin
S[1:0]
output | T=1 | T=0
Cout | 136 | 92
S3 | 139 | 138
S2 | 105 | 104
S1 | 71 | 70
S0 | 37 | 36
cin A B
00011 1100
cin A B
00011 1101
00011 1100
B0
Cout
S3
S2
S1
S0
136 ns
92 ns
117ns
138ns
83ns
104ns
70ns
71ns
36ns
37ns
not happen
in simulation
not happen
in simulation
P4PCO

```
28  module _4BitAdder(input [3:0] A,B, input cin, output [3:0] S, output cout);
29
30      wire _1cin;
31      _2BitAdder TwoBA1(A[1:0],B[1:0],cin,S[1:0],_1cin);
32      _2BitAdder TwoBA2(A[3:2],B[3:2],_1cin,S[3:2],cout);
33
34  endmodule
35
```

```
67   module FourBitAdderTB();
68
69      logic [3:0] AA = 4'b0011;
70      logic [3:0] BB = 4'b1100;
71      logic cin = 0;
72      wire [3:0] SS;
73      wire cout;
74
75      _4BitAdder ThreeBA(AA,BB,cin,SS,cout);
76
77      initial begin
78          #200 BB[0] = 1;
79          #196
80          #59 BB[0] = 0;
81          #267
82          $stop;
83      end
84
```

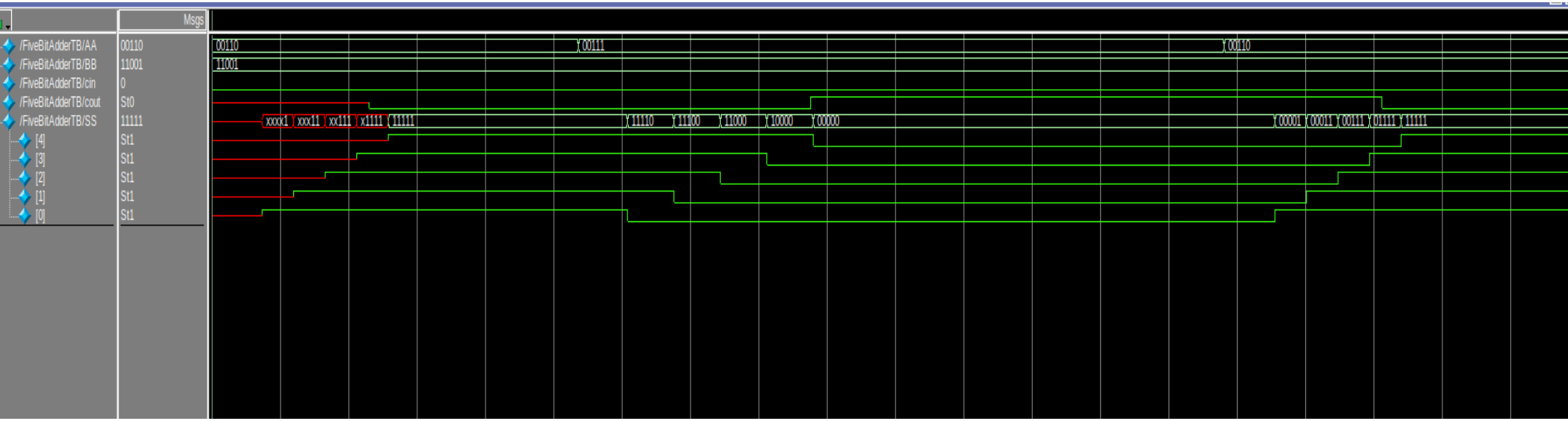

Msgs
/FourBitAdderTB/AA
/FourBitAdderTB/BB
/FourBitAdderTB/cin
/FourBitAdderTB/cout
/FourBitAdderTB/SS
[3]
[2]
[1]
[0]
-No Data-
0011
1100
1101
1100
xxx1
xx11
x111
1111
1110
1100
1000
0000
0001
0011
0111
1111

. 5-bit Adder

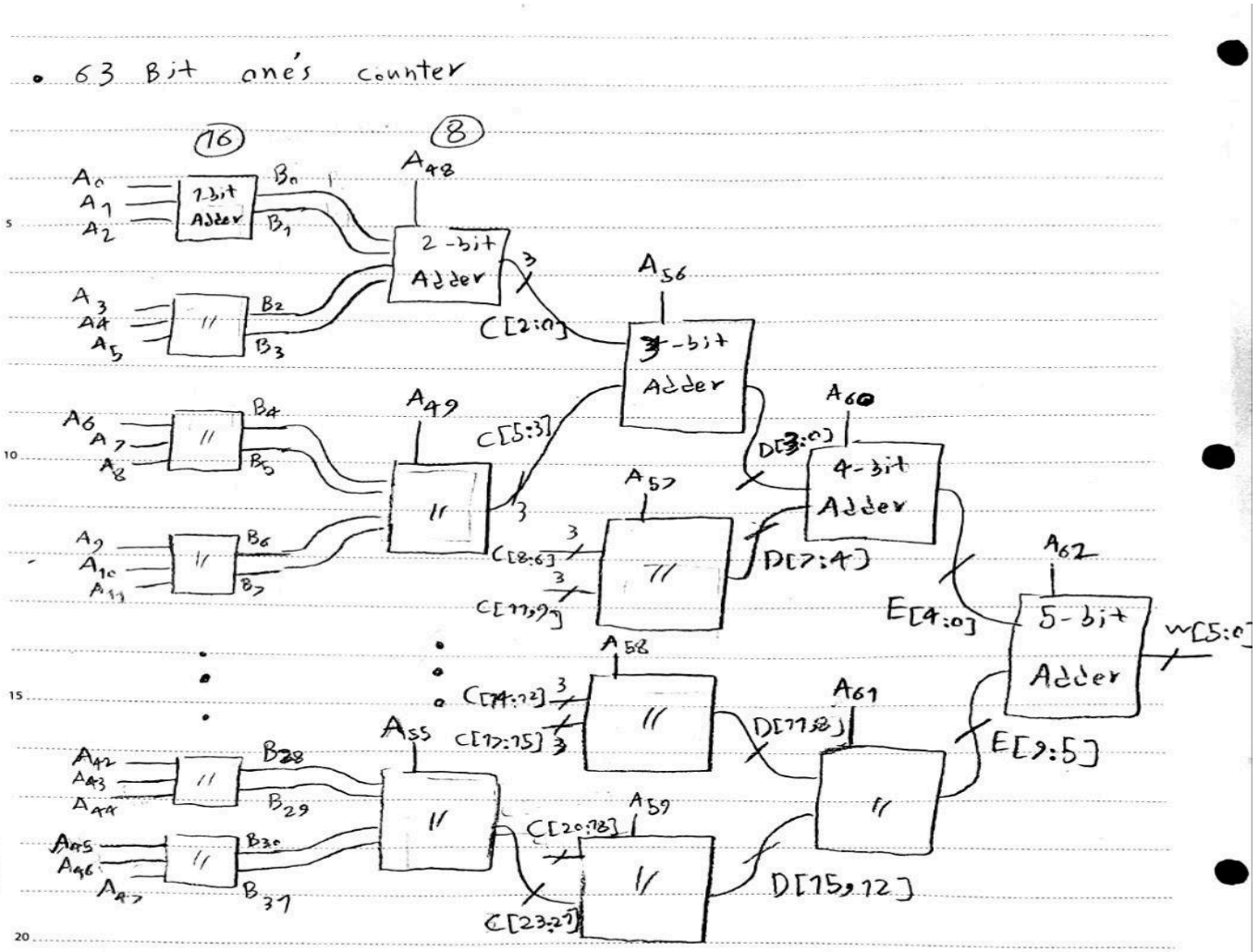

• 5-bit Adder :
A B
5 5
5-bit Adder
cin
cout S
5
A[4:2] B[4:2]
3 3
3-bit Adder
3
cout S[4:2]
A[1:0] B[1:0]
2 2
2-bit Adder
2
-1cin S[1:0]
cin A B
0 00110 11001 → 00111 11001
↓
000111011001
output T=1 T=0
cout 170 115
S4 173 172
S3 139 138
S2 105 104
S1 71 70
S0 37 36
A0
cout
S4
S3
S2
S1
S0
170 ns
115 ns
151 ns
* not happ
172 ns
117 ns
138ns
83 ns
104ns
71 ns
* not happend
70ns
36 ns
37 ns
P4PCO

```
37  module _5BitAdder(input [4:0] A,B, input cin, output [4:0] S, output cout);
38
39      wire _1cin;
40      _2BitAdder TwoBA1(A[1:0],B[1:0],cin,S[1:0],_1cin);
41      _3BitAdder ThreeBA1(A[4:2],B[4:2],_1cin,S[4:2],cout);
42
43  endmodule
```

```
87  module FiveBitAdderTB();
88
89      logic [4:0] AA = 5'b00110;
90      logic [4:0] BB = 5'b11001;
91      logic cin = 0;
92      wire [4:0] SS;
93      wire cout;
94
95      _5BitAdder ThreeBA(AA,BB,cin,SS,cout);
96
97      initial begin
98          #268 AA[0] = 1;
99          #386
100         #87 AA[0] = 0;
101         #324
102         $stop;
103     end
104
105 endmodule
```


Msgs
/FiveBitAdderTB/AA
/FiveBitAdderTB/BB
/FiveBitAdderTB/cin
/FiveBitAdderTB/cout
/FiveBitAdderTB/SS
[4]
[3]
[2]
[1]
[0]
00110
11001
0
St0
11111
St1
St1
St1
St1
St1
00110
11001
00111
00110
xxxx1
xxx11
xx111
x1111
11111
11110
11100
11000
10000
00000
00001
00011
00111
01111
11111

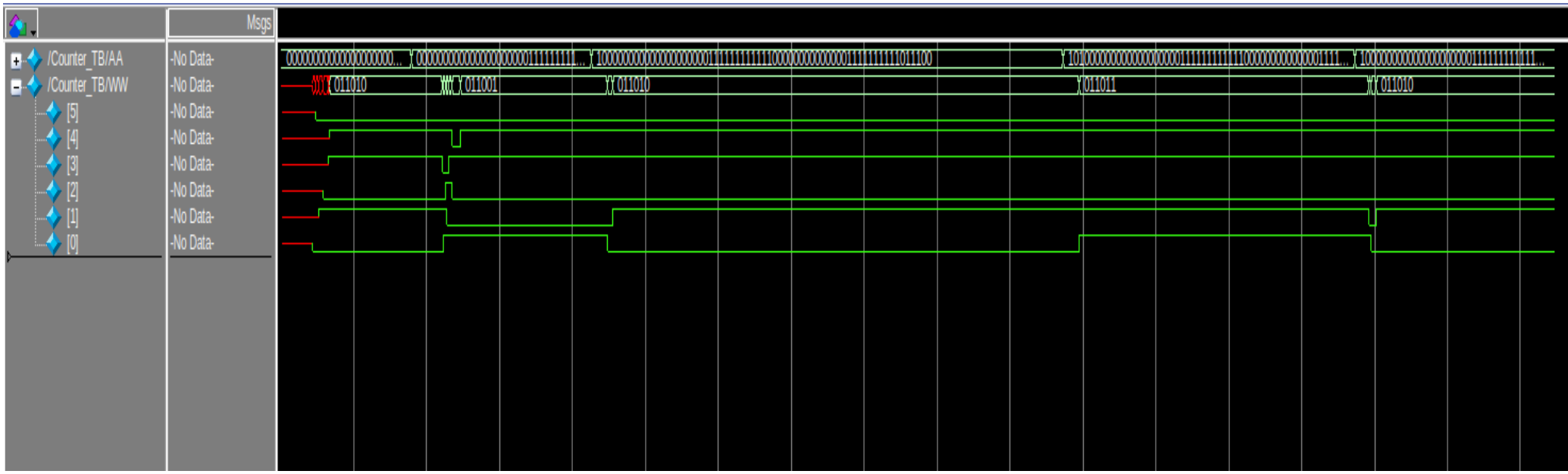
. 63-bit one's counter



Worst case delay : 37+ 71+ 105+ 139+ 173 = 525 ns

```
module _63OC(input [62:0] A, output [5:0] W);

    wire initial_cin = 0;
    wire cout, _cout;
    wire [31:0] B;
    wire [23:0] C;
    wire [15:0] D;
    wire [9:0] E;

    _1BitAdder OA1(A[0],A[1],A[2],B[0],B[1]);
    _1BitAdder OA2(A[3],A[4],A[5],B[2],B[3]);
    _1BitAdder OA3(A[6],A[7],A[8],B[4],B[5]);
    _1BitAdder OA4(A[9],A[10],A[11],B[6],B[7]);
    _1BitAdder OA5(A[12],A[13],A[14],B[8],B[9]);
    _1BitAdder OA6(A[15],A[16],A[17],B[10],B[11]);
    _1BitAdder OA7(A[18],A[19],A[20],B[12],B[13]);
    _1BitAdder OA8(A[21],A[22],A[23],B[14],B[15]);
    _1BitAdder OA9(A[24],A[25],A[26],B[16],B[17]);
    _1BitAdder OA10(A[27],A[28],A[29],B[18],B[19]);
    _1BitAdder OA11(A[30],A[31],A[32],B[20],B[21]);
    _1BitAdder OA12(A[33],A[34],A[35],B[22],B[23]);
    _1BitAdder OA13(A[36],A[37],A[38],B[24],B[25]);
    _1BitAdder OA14(A[39],A[40],A[41],B[26],B[27]);
    _1BitAdder OA15(A[42],A[43],A[44],B[28],B[29]);
    _1BitAdder OA16(A[45],A[46],A[47],B[30],B[31]);


    _2BitAdder TA1(B[3:2],B[1:0],A[48],C[1:0],C[2]);
    _2BitAdder TA2(B[7:6],B[5:4],A[49],C[4:3],C[5]);
    _2BitAdder TA3(B[11:10],B[9:8],A[50],C[7:6],C[8]);
    _2BitAdder TA4(B[15:14],B[13:12],A[51],C[10:9],C[11]);
    _2BitAdder TA5(B[19:18],B[17:16],A[52],C[13:12],C[14]);
    _2BitAdder TA6(B[23:22],B[21:20],A[53],C[16:15],C[17]);
    _2BitAdder TA7(B[27:26],B[25:24],A[54],C[19:18],C[20]);
    _2BitAdder TA8(B[31:30],B[29:28],A[55],C[22:21],C[23]);

    _3BitAdder THA1(C[5:3],C[2:0],A[56],D[2:0],D[3]);
    _3BitAdder THA2(C[11:9],C[8:6],A[57],D[6:4],D[7]);
    _3BitAdder THA3(C[17:15],C[14:12],A[58],D[10:8],D[11]);
    _3BitAdder THA4(C[23:21],C[20:18],A[59],D[14:12],D[15]);

    _4BitAdder FA1(D[7:4],D[3:0],A[60],E[3:0],E[4]);
    _4BitAdder FA2(D[15:12],D[11:8],A[61],E[8:5],E[9]);

    _5BitAdder FiA1(E[4:0],E[9:5],A[62],W[4:0],W[5]);

endmodule
```

```
1  `timescale 1ns/1ns
2  `include "63BitOne'sCounter.sv"
3
4  module Counter_TB();
5      logic [62:0] AA = 63'b000000000000000111111111111000000000000001111111111111100;
6      wire [5:0] WW;
7
8      _63BitOnesCounter counter(AA,WW);
9
10     initial begin
11         #896 AA[5] = 0;
12         #1000
13         #235 AA[62] = 1;
14         #1235
15         repeat(2) #2000  AA[60] = ~AA[60];
16         #1365
17         $stop;
18     end
19 endmodule
```


Msgs
/Counter_TB/AA
/Counter_TB/WW
[5]
[4]
[3]
[2]
[1]
[0]
-No Data-
-No Data-
-No Data-
-No Data-
-No Data-
-No Data-
-No Data-
011010
011001
011010
011011
011010