B023040050 顏義洋

Homework 1: 8-bit Adders Using Verilog in Structural, Dataflow, and Behavioral Levels

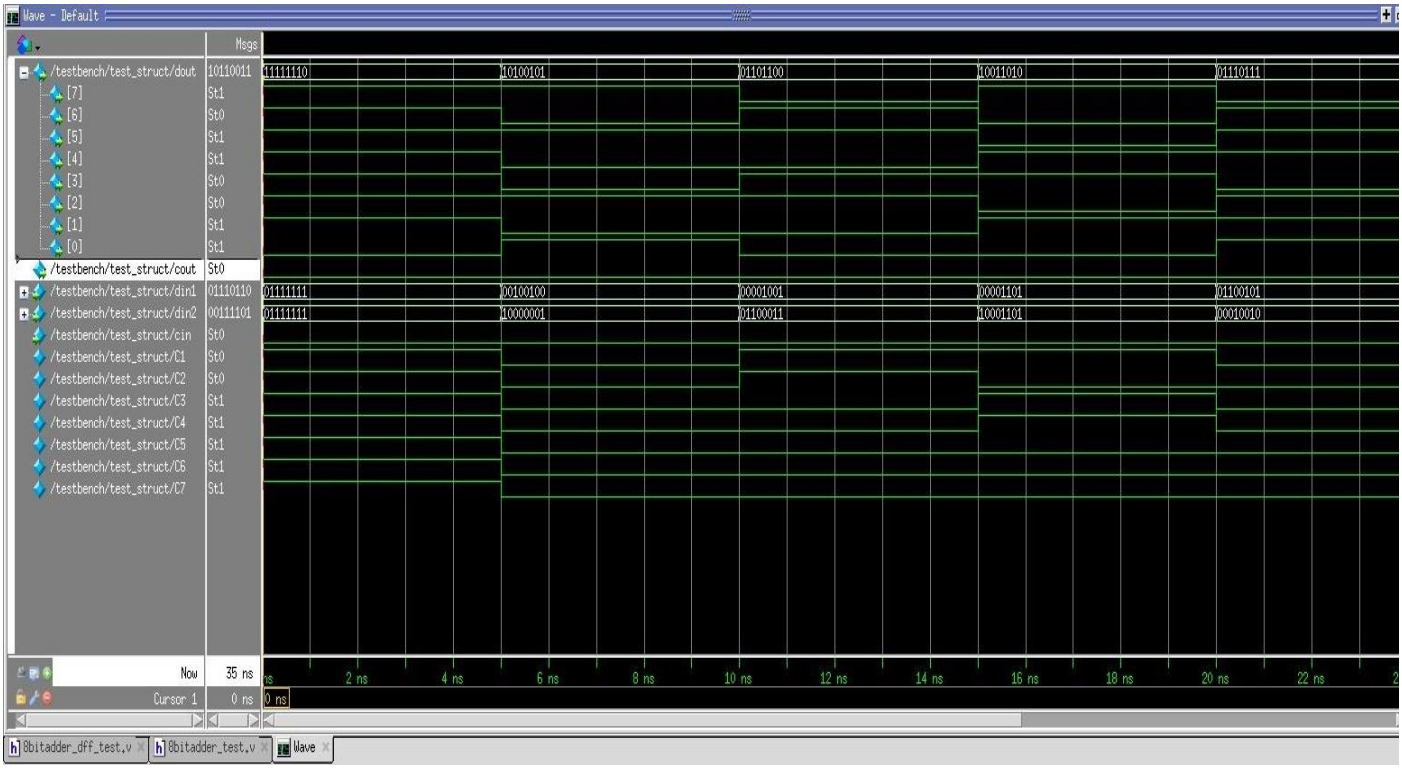
第一部分：Structure-Level Modeling

```
module half_adder (output S, C, input x, y);
xor (S, x, y);
and (C, x, y);
endmodule

module full_adder (output S, C, input x, y, z);
wire S1, C1, C2;
half_adder HA1 (S1, C1, x, y);
half_adder HA2 (S, C2, S1, z);
or G1 (C, C2, C1);
endmodule

module adder_struct ( output [7: 0] dout, output cout, input [7:0] din1, din2, input cin);
wire C1, C2, C3, C4, C5, C6, C7; // Intermediate carries
full_adder FA0 (dout[0], C1, din1[0], din2[0], cin),
FA1 (dout[1], C2, din1[1], din2[1], C1),
FA2 (dout[2], C3, din1[2], din2[2], C2),
FA3 (dout[3], C4, din1[3], din2[3], C3),
FA4 (dout[4], C5, din1[4], din2[4], C4),
FA5 (dout[5], C6, din1[5], din2[5], C5),
FA6 (dout[6], C7, din1[6], din2[6], C6),
FA7 (dout[7], cout, din1[7], din2[7], C7);
endmodule
```

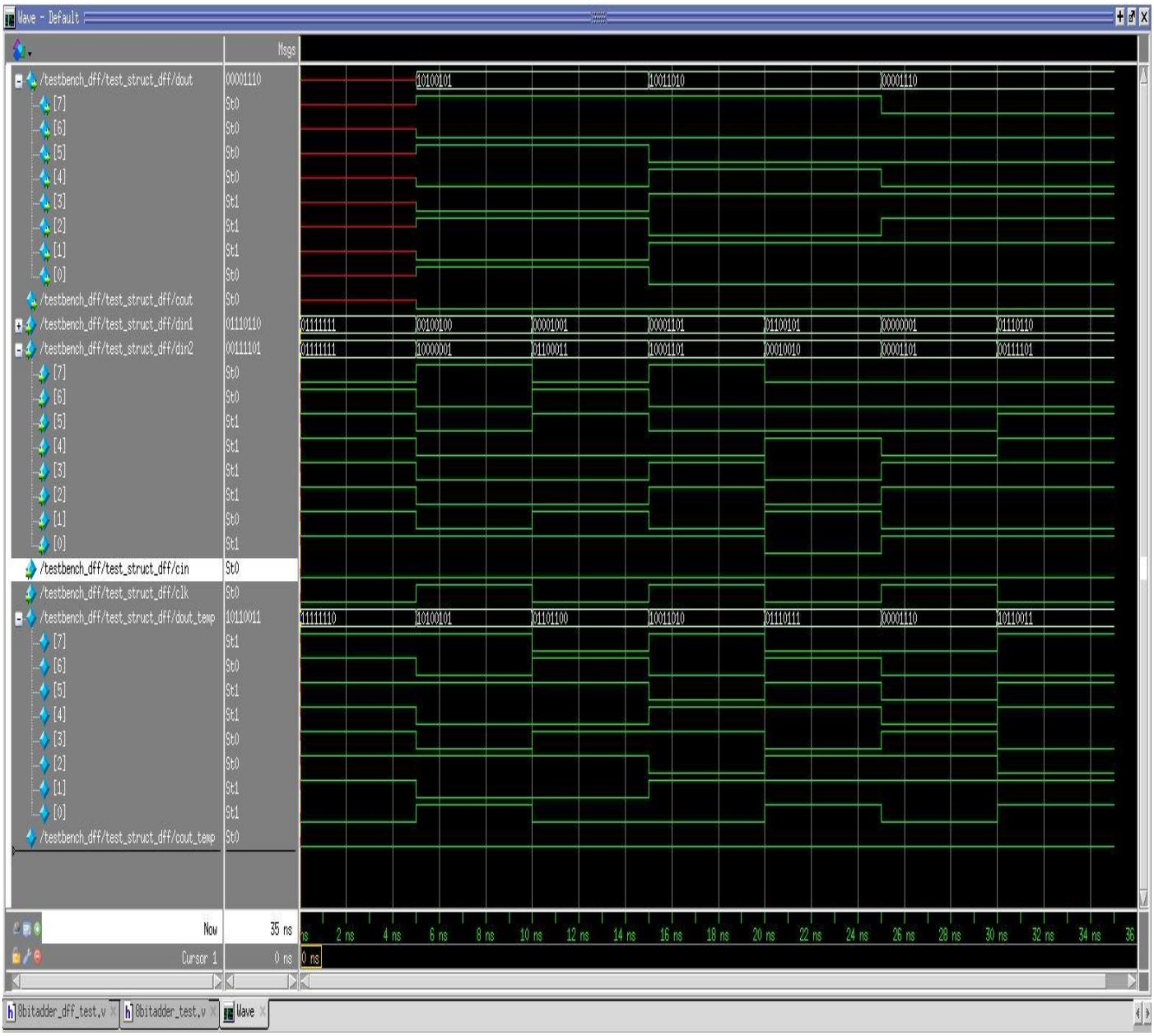
利用以前數位系統實驗課的寫的 half adder 跟 full adder，用八個 full adder 拼成 8-bit ripple carry adder。



配上 D-Flip-Flop：

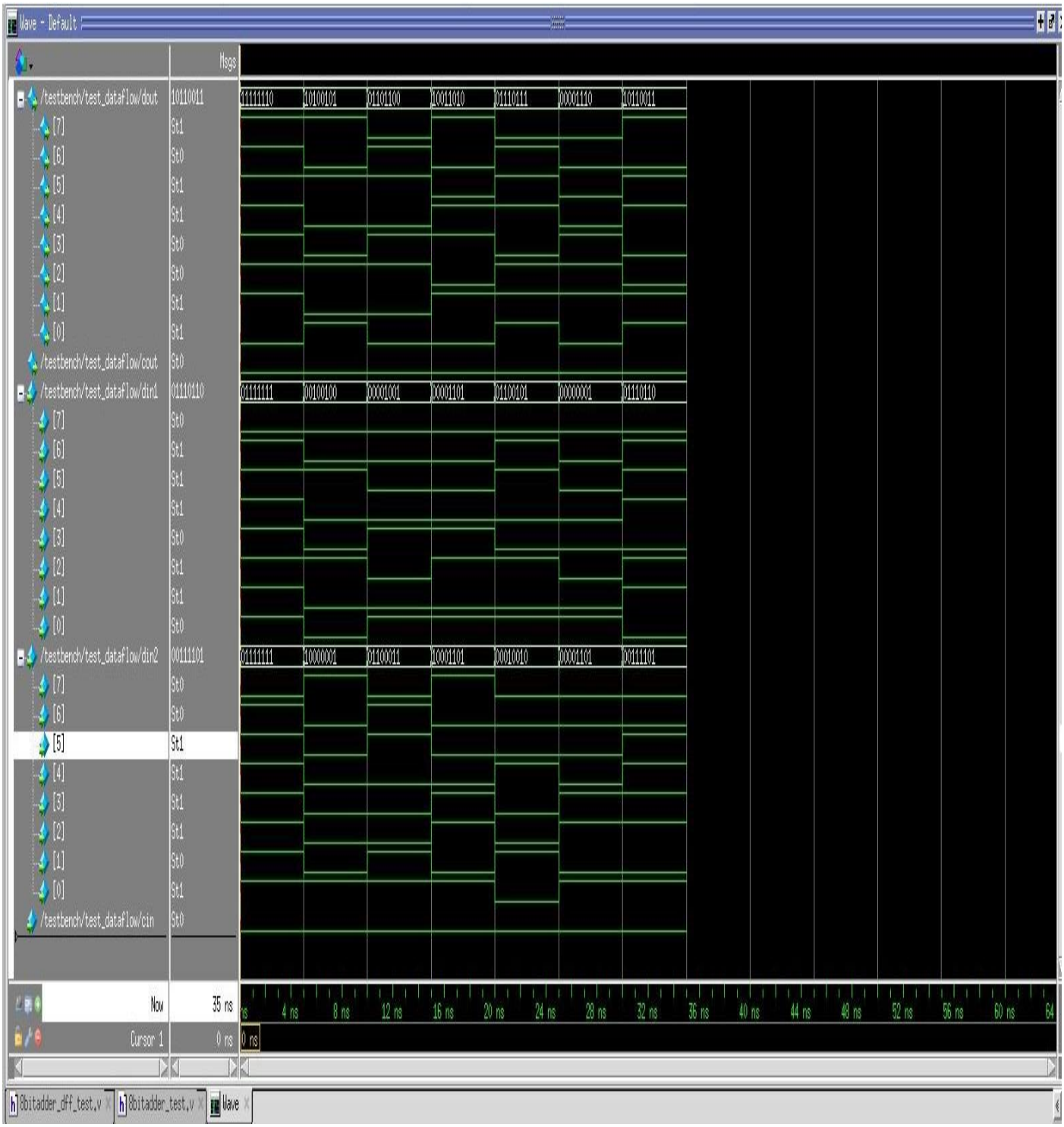
```
module adder_struct_dff
(   output [7:0]dout,output cout,input [7:0]din1,input [7:0]din2,input cin,input clk);
    wire [7:0] dout_temp;
    wire cout_temp;
    adder_struct addder(dout_temp,cout_temp,din1,din2,cin);
    D_FF    dff0(dout[0],dout_temp[0],clk),
            dff1(dout[1],dout_temp[1],clk),
            dff2(dout[2],dout_temp[2],clk),
            dff3(dout[3],dout_temp[3],clk),
            dff4(dout[4],dout_temp[4],clk),
            dff5(dout[5],dout_temp[5],clk),
            dff6(dout[6],dout_temp[6],clk),
            dff7(dout[7],dout_temp[7],clk),
            dff8(cout,cout_temp,clk);
endmodule
```

將相加的結果暫時存在 DFF 內，等 clock 正緣時讓 ouput 出現。



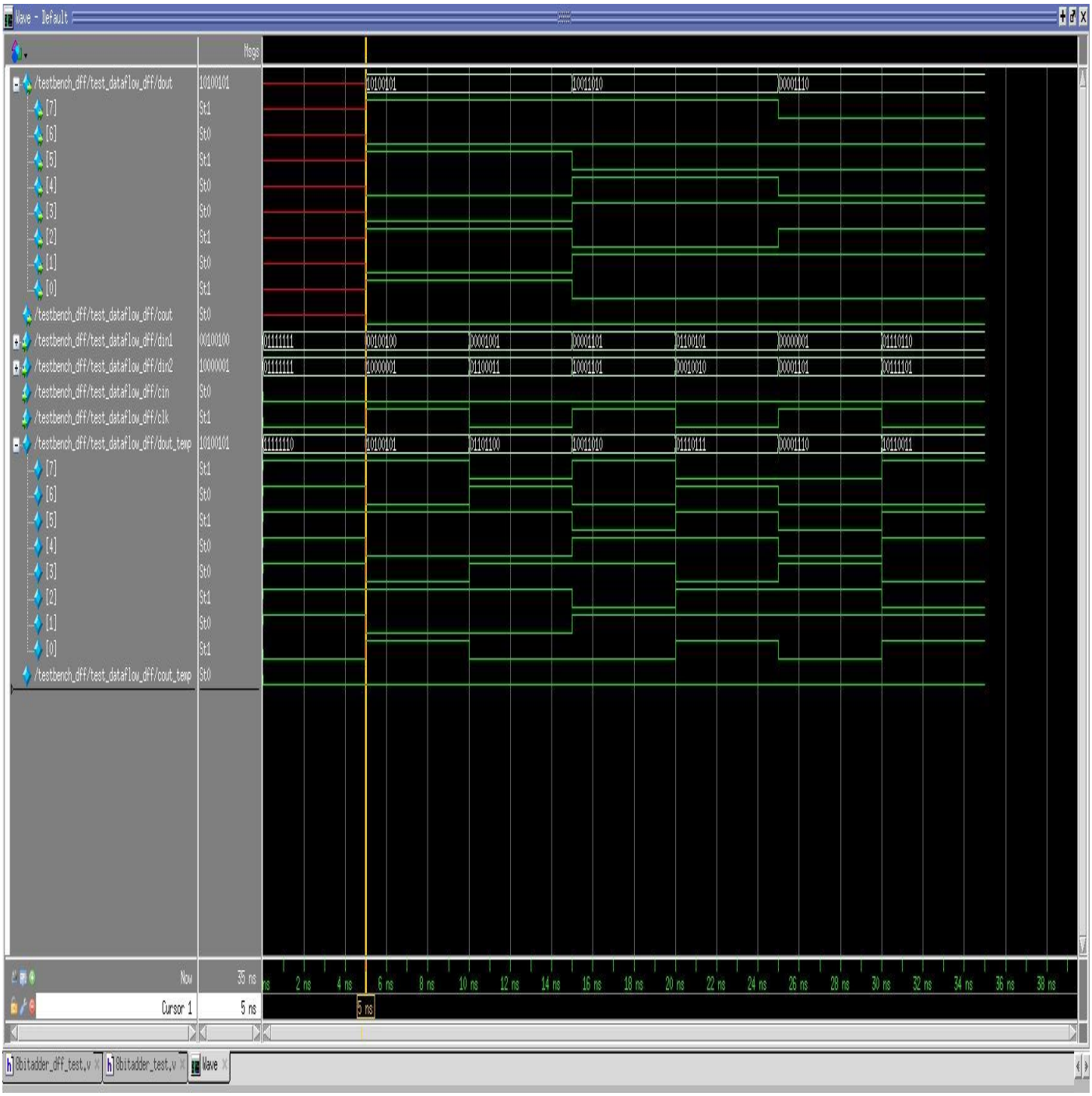
第二部分：Data Flow Modeling

```
module adder_dataflow ( output [7: 0] dout, output cout, input [7:0] din1, din2, input cin);
  assign {cout,dout}=din1+din2+cin;
endmodule
```



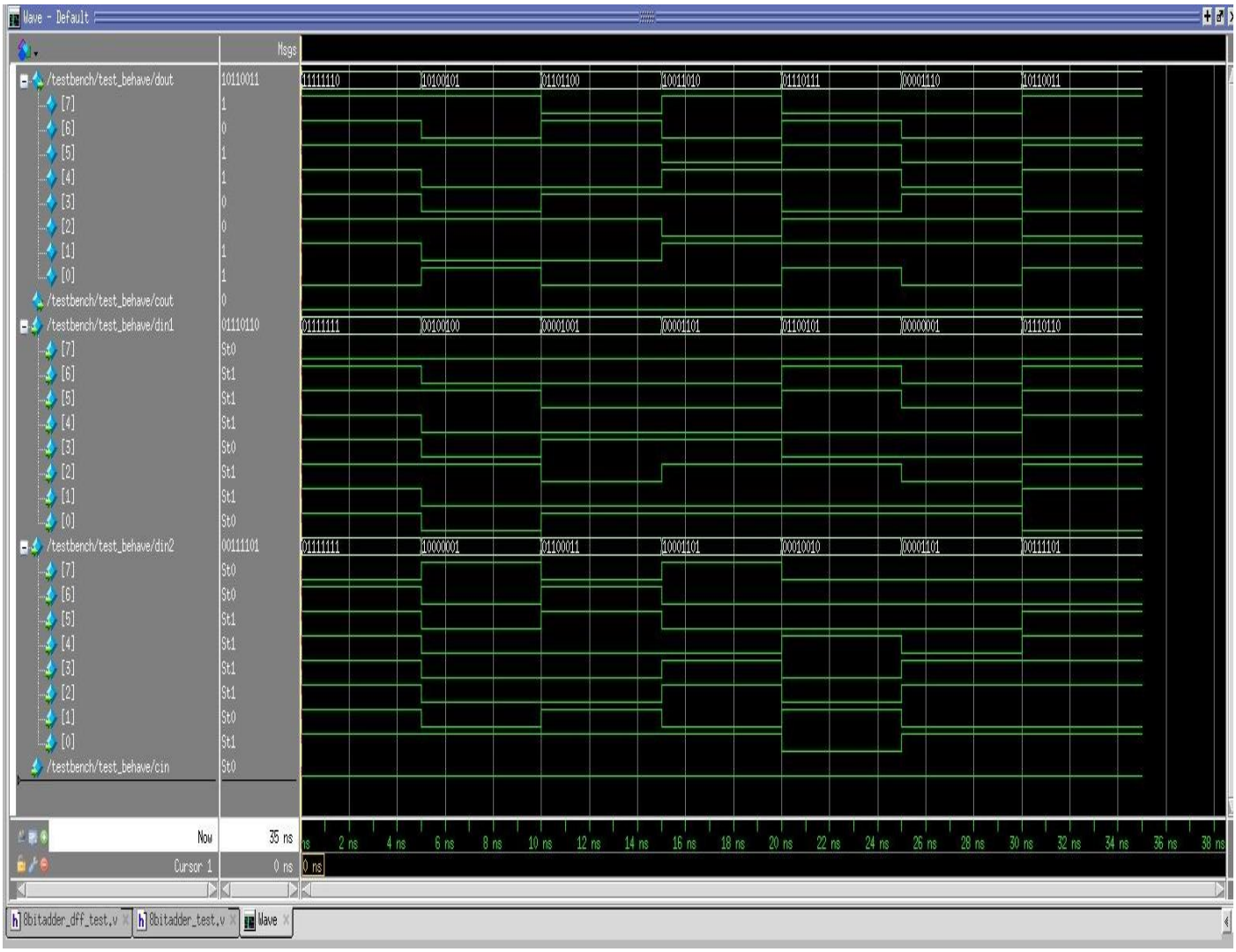
配上 D-Flip-Flop：

```
module adder_dataflow_dff
(    output [7:0]dout,output cout,input [7:0]din1,input [7:0]din2,input cin,input clk);
     wire [7:0] dout_temp;
     wire cout_temp;
     adder_dataflow addder(dout_temp,cout_temp,din1,din2,cin);
     D_FF     dff0(dout[0],dout_temp[0],clk),
              dff1(dout[1],dout_temp[1],clk),
              dff2(dout[2],dout_temp[2],clk),
              dff3(dout[3],dout_temp[3],clk),
              dff4(dout[4],dout_temp[4],clk),
              dff5(dout[5],dout_temp[5],clk),
              dff6(dout[6],dout_temp[6],clk),
              dff7(dout[7],dout_temp[7],clk),
              dff8(cout,cout_temp,clk);
endmodule
```



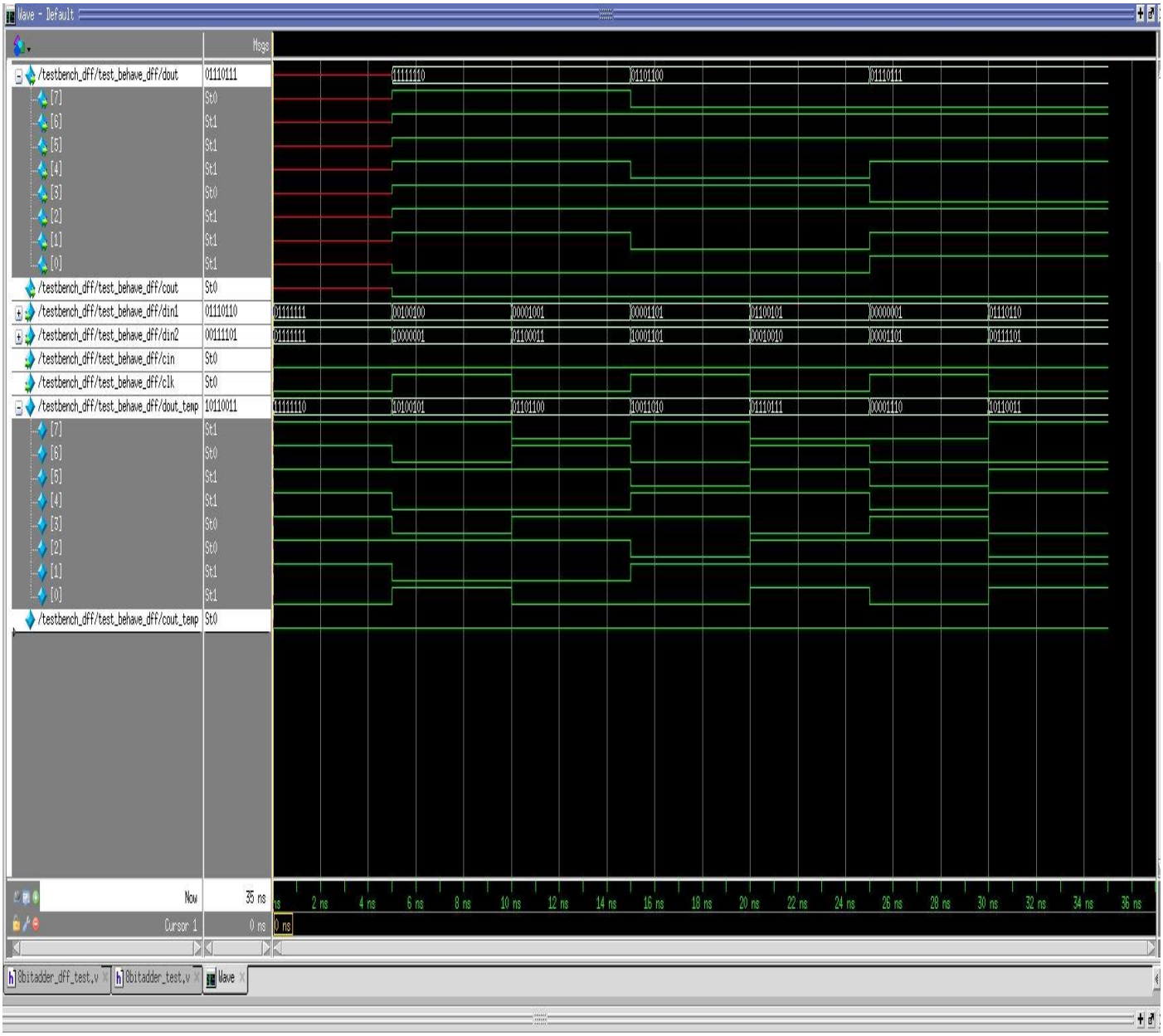
第三分：Behavioral Modeling

```
module adder_beh ( output reg [7: 0] dout, output reg cout, input [7:0] din1, din2, input cin);
  always@(din1 or din2 or cin)
  begin
    {cout,dout}=din1+din2+cin;
  end
endmodule
```



配上 D-Flip-Flop：

```
module adder_beh_dff
(   output [7:0]dout,output cout,input [7:0]din1,input [7:0]din2,input cin,input clk);
    wire [7:0] dout_temp;
    wire cout_temp;
    adder_beh addder(dout_temp,cout_temp,din1,din2,cin);
    D_FF    dff0(dout[0],dout_temp[0],clk),
            dff1(dout[1],dout_temp[1],clk),
            dff2(dout[2],dout_temp[2],clk),
            dff3(dout[3],dout_temp[3],clk),
            dff4(dout[4],dout_temp[4],clk),
            dff5(dout[5],dout_temp[5],clk),
            dff6(dout[6],dout_temp[6],clk),
            dff7(dout[7],dout_temp[7],clk),
            dff8(cout,cout_temp,clk);
endmodule
```


Wave - Default
Msgs
/testbench_dff/test_behave_dff/dout
/testbench_dff/test_behave_dff/cout
/testbench_dff/test_behave_dff/din1
/testbench_dff/test_behave_dff/din2
/testbench_dff/test_behave_dff/cin
/testbench_dff/test_behave_dff/clk
/testbench_dff/test_behave_dff/dout_temp
/testbench_dff/test_behave_dff/cout_temp
Now
Cursor 1
8bitadder_dff_test.v
8bitadder_test.v
Wave