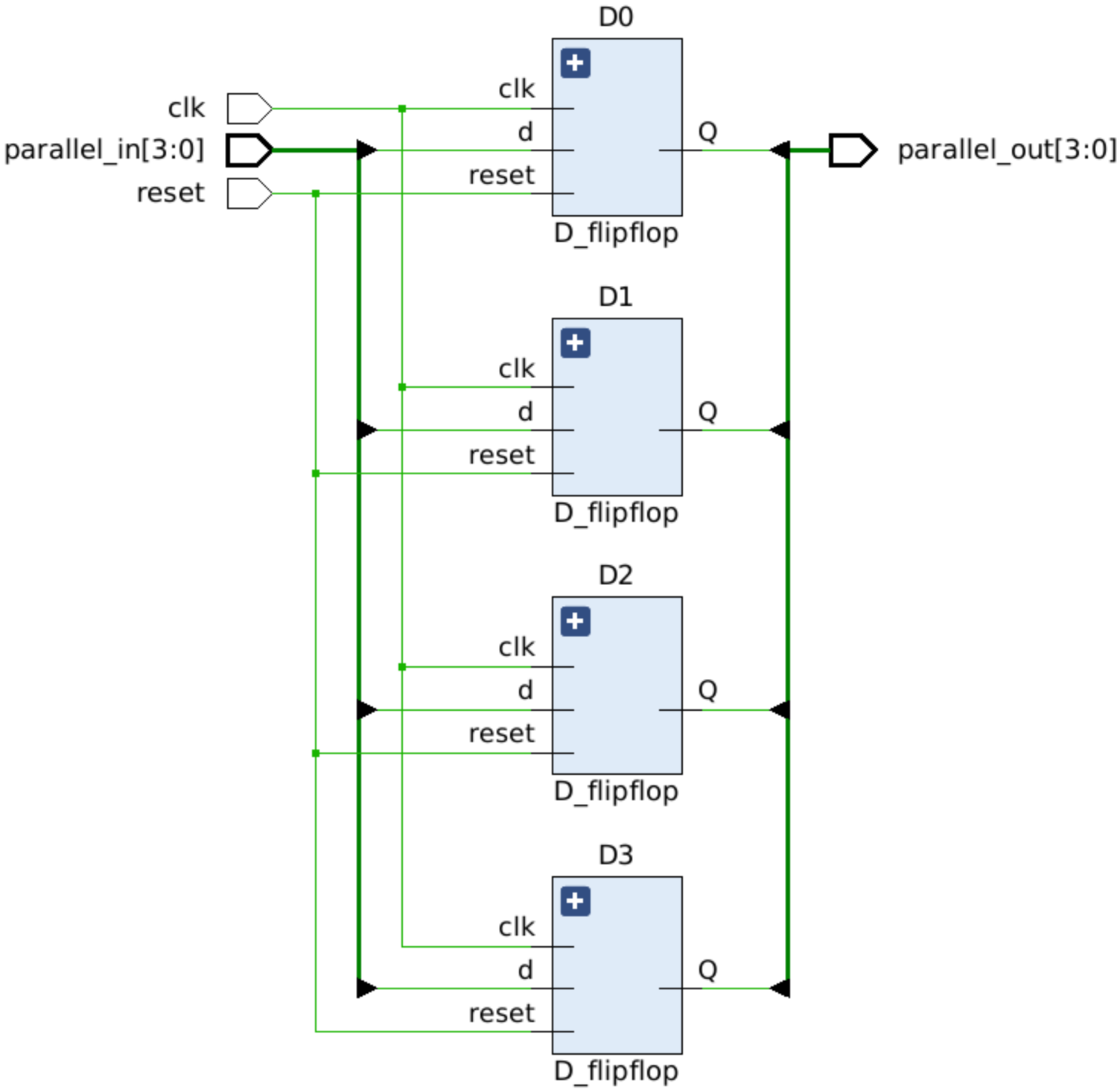
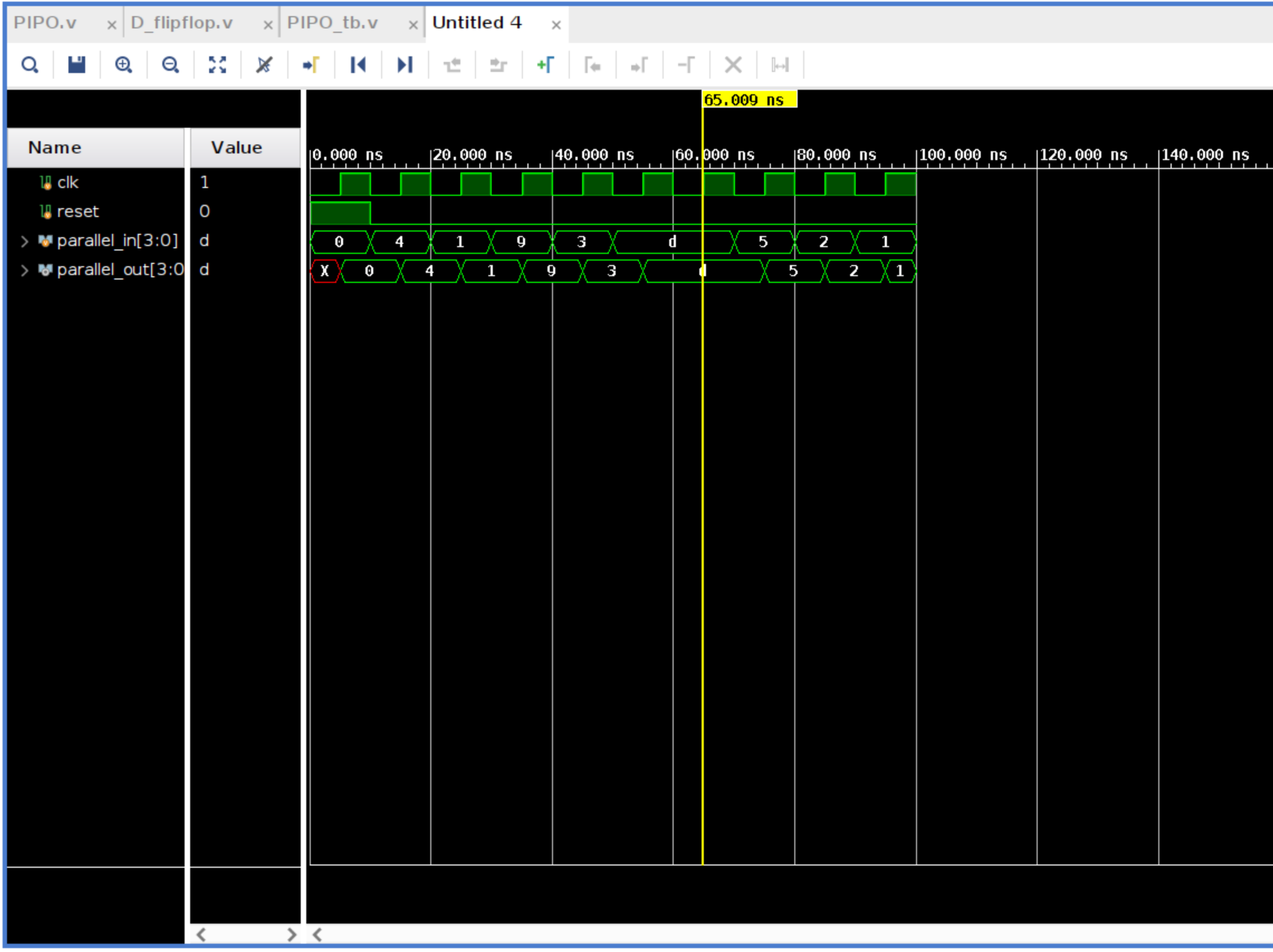
4 Cells 10 I/O Ports 10 Nets




PIPO.v
D_flipflop.v
PIPO_tb.v
Untitled 4
Name
Value
clk
1
reset
0
parallel_in[3:0]
d
parallel_out[3:0
d
65.009 ns
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
140.000 ns
0 4 1 9 3 d 5 2 1
X 0 4 1 9 3 d 5 2 1

PIPO.v | D_flipflop.v | PIPO_tb.v | Untitled 4

/home/itzzinfinity/Cozy Drive/100daysofRTL/day_058/project_1/project_1.srcs/sources_1/new/PIPO.v

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Engineer: Anjan Prasad
4  // Create Date: 11/18/2024 05:23:28 AM
5  // Module Name: PIPO
6  //////////////////////////////////////////////////////////////////////////////////
7
8
9  module PIPO(
10     input clk, reset,
11     input [3:0] parallel_in,
12     output [3:0] parallel_out
13     );
14
15     D_flipflop D3(parallel_in[3], clk, reset, parallel_out[3]);
16     D_flipflop D2(parallel_in[2], clk, reset, parallel_out[2]);
17     D_flipflop D1(parallel_in[1], clk, reset, parallel_out[1]);
18     D_flipflop D0(parallel_in[0], clk, reset, parallel_out[0]);
19
20 endmodule
21
```

PIPO.v | D_flipflop.v | PIPO_tb.v | Untitled 4

/home/itzzinfinity/Cozy Drive/100daysofRTL/day_058/project_1/project_1.srcs/sim_1/new/PIPO_tb.v

```
1   `timescale 1ns / 1ps
2   //////////////////////////////////////////////////////////////////////////////////
3   // Engineer: Anjan Prasad
4   // Create Date: 11/18/2024 05:28:19 AM
5   // Module Name: PIPO_tb
6   //////////////////////////////////////////////////////////////////////////////////
7
8
9   module PIPO_tb;
10  reg clk, reset;
11  reg [3:0] parallel_in;
12  wire [3:0] parallel_out;
13
14  PIPO DUT(clk, reset, parallel_in, parallel_out);
15
16  initial begin
17      clk=1'b0;
18      forever #5 clk=~clk;
19      end
20
21   initial begin
22      reset= 1'b1;
23      parallel_in= 4'b0000;
24      #10 reset= 1'b0;
25      end
26
27      always #10 parallel_in= $random;
28
29   initial begin
30      $monitor("\t\t clk: %d  reset: %d  parallel_in: %b  parallel_out: %b", clk, reset, parallel_in, parallel_out);
31      #100 $finish;
32      end
33  endmodule
34
```