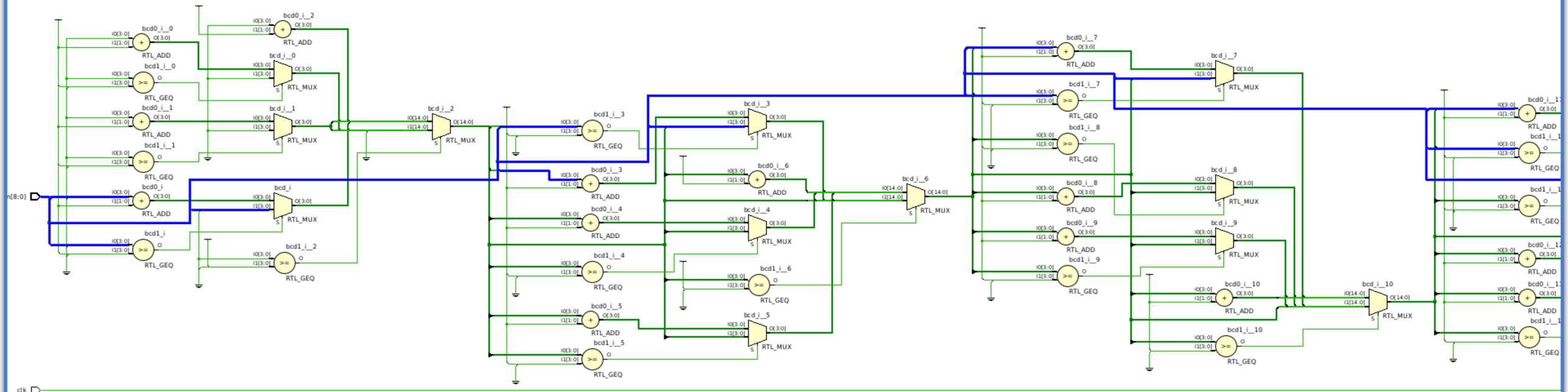
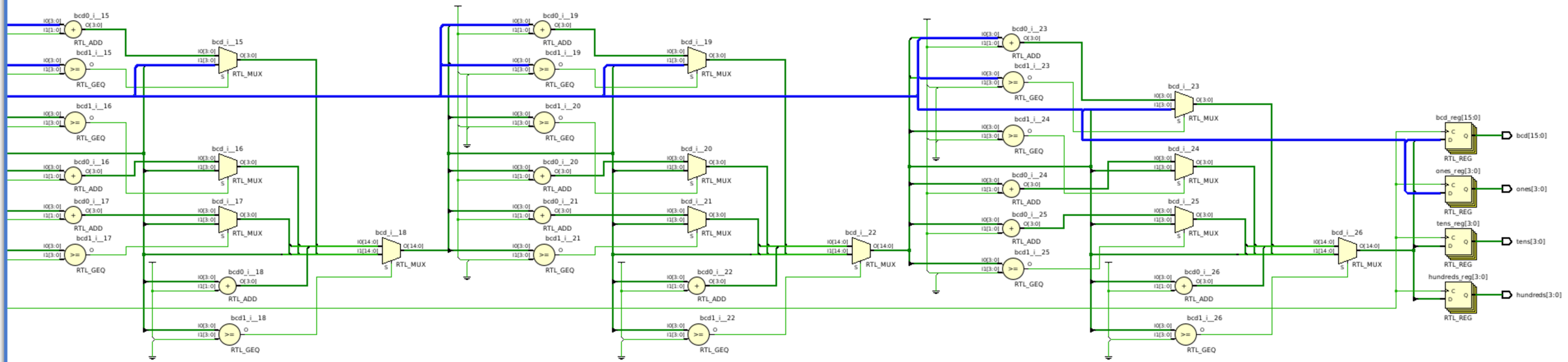
Schematic

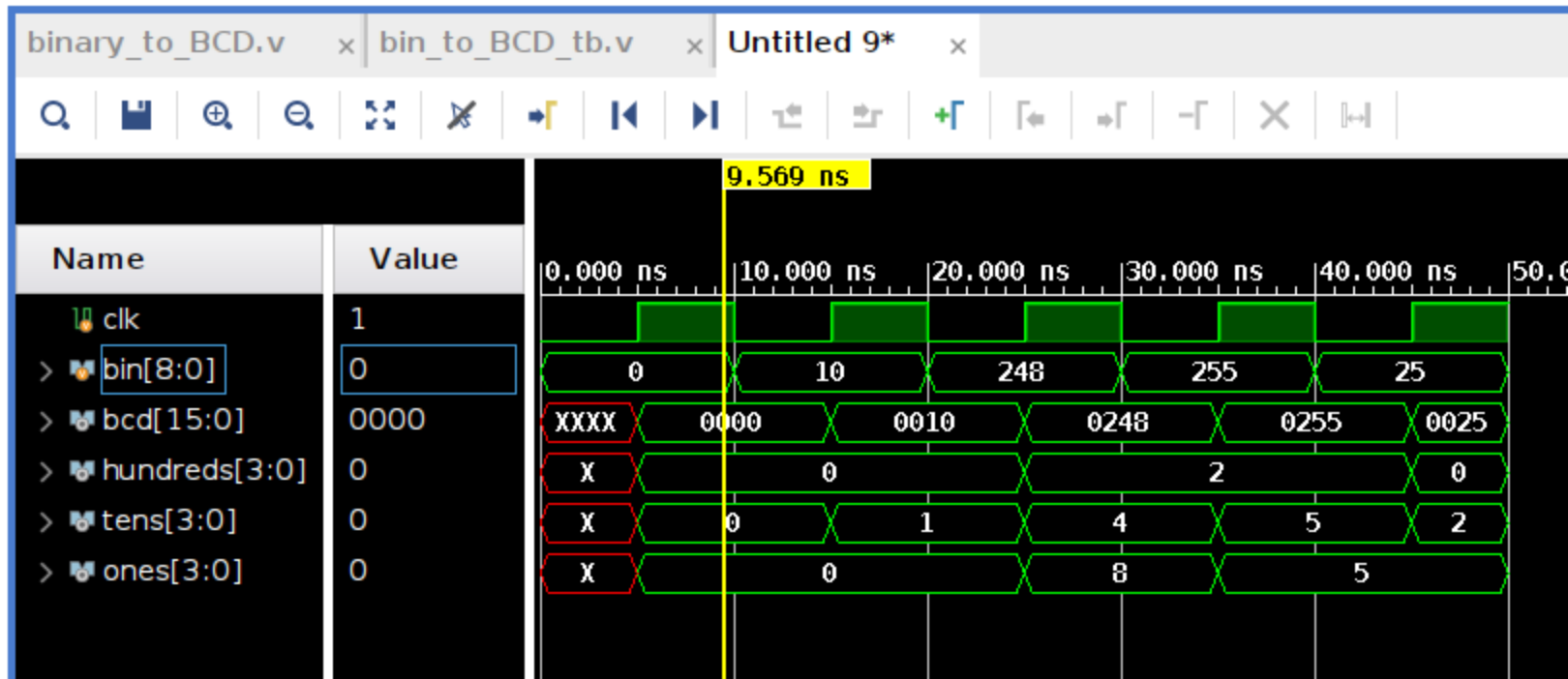
112 Cells 38 I/O Ports 362 Nets



Schematic

112 Cells 38 I/O Ports 362 Nets




binary_to_BCD.v
bin_to_BCD_tb.v
Untitled 9*
9.569 ns
Name
Value
0.000 ns
10.000 ns
20.000 ns
30.000 ns
40.000 ns
clk
1
bin[8:0]
0
bcd[15:0]
0000
hundreds[3:0]
0
tens[3:0]
0
ones[3:0]
0
0
10
248
255
25
XXXX
0000
0010
0248
0255
0025
X
0
2
0
X
0
1
4
5
2
X
0
8
5

binary_to_BCD.v | bin_to_BCD_tb.v | Untitled 9*

/home/itzzinfinity/Cozy Drive/100daysofRTL/day_044/Z bin2bcd/bin2bcd.srcs/sources_1/new/binary_to_BCD.v

```
1   `timescale 1ns / 1ps
2   //////////////////////////////////////////////////////////////////////////////////
3   // Engineer: Anjan Prasad
4   // Create Date: 11/04/2024 07:01:00 AM
5   // Module Name: binary_to_BCD
6   //////////////////////////////////////////////////////////////////////////////////
7
8   module binary_to_BCD(
9   input clk,
10   input [8:0] bin,
11     output reg [15:0] bcd,
12     output reg [3:0] ones,tens,hundreds
13     );
14
15  integer i;
16
17  always @(posedge clk) begin
18      bcd=0;
19      for (i=0;i<9;i=i+1) begin                    //Iterate once for each bit in input number
20          if (bcd[3:0] >= 5) bcd[3:0] = bcd[3:0] + 3;     //If any BCD digit is >= 5, add three
21      if (bcd[7:4] >= 5) bcd[7:4] = bcd[7:4] + 3;
22      if (bcd[11:8] >= 5) bcd[11:8] = bcd[11:8] + 3;
23      if (bcd[15:12] >= 5) bcd[15:12] = bcd[15:12] + 3;
24      bcd = {bcd[14:0],bin[8-i]};                  //Shift one bit, and shift in proper bit from input
25      ones = bcd[3:0];
26      tens = bcd[7:4];
27      hundreds = bcd[11:8];
28      end
29  end
30  endmodule
31
32
```

binary_to_BCD.v | bin_to_BCD_tb.v | Untitled 9*

/home/itzzinfinity/Cozy Drive/100daysofRTL/day_044/Z bin2bcd/bin2bcd.srcs/sim_1/new/bin_to_BCD_tb.v

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Engineer: Anjan Prasad
4  // Create Date: 11/04/2024 07:34:27 AM
5  // Module Name: bin_to_BCD_tb
6  //////////////////////////////////////////////////////////////////////////////////
7
8
9  module bin_to_BCD_tb;
10 reg clk=0;
11  reg [8:0] bin;
12    wire [15:0] bcd;
13   wire [3:0] ones,tens,hundreds;
14 binary_to_BCD DUT(clk,bin,bcd,ones,tens,hundreds);
15
16 always #5 clk = ~clk;
17 initial begin
18 bin = 9'd0;
19 #10 bin = 9'd10;
20 #10 bin = 9'd248;
21 #10 bin = 9'd255;
22 #10 bin = 9'd25;
23 #10 bin = 9'd28;
24 $finish;
25 end
26
27 endmodule
28
```