Project Report
Project Code : B2

Goal:- Deriving complete Control flow graphs (CFG) of processor modules

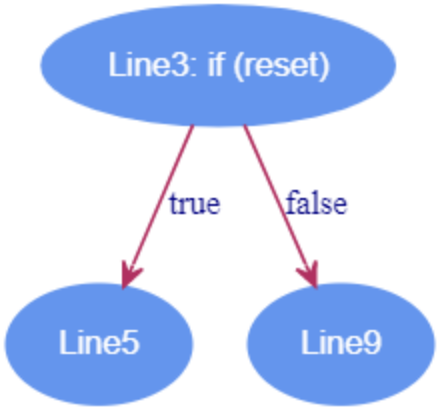
---

Group Members:
Shubh Doshi (B19EE080)
Shashi Prakash (B19EE076)

To solve this problem we have broken the problem into smaller parts. In verilog code, we have to handle many loops/control blocks.

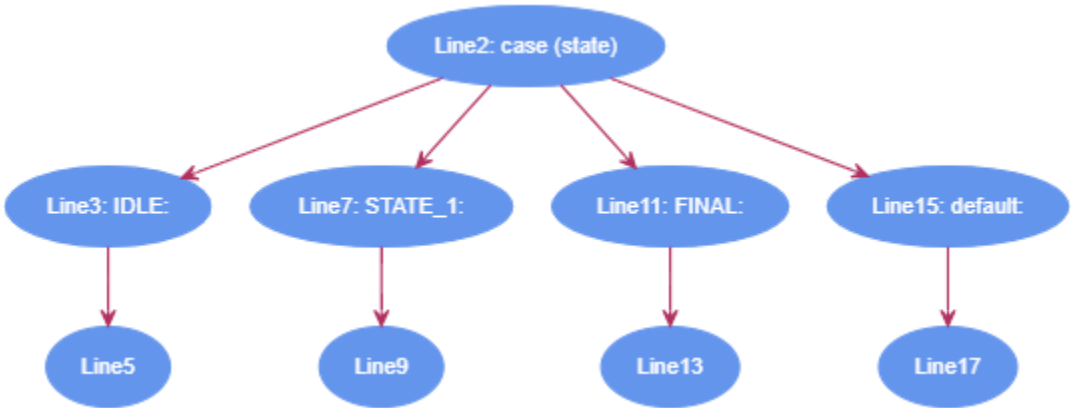
1. If , else if, else statement

```
always @(posedge clk or posedge reset)
 begin
   if (reset)
     begin
      q <= 1'b0;
     end
   else
     begin
      q <= d;
     end
 end
endmodule
```



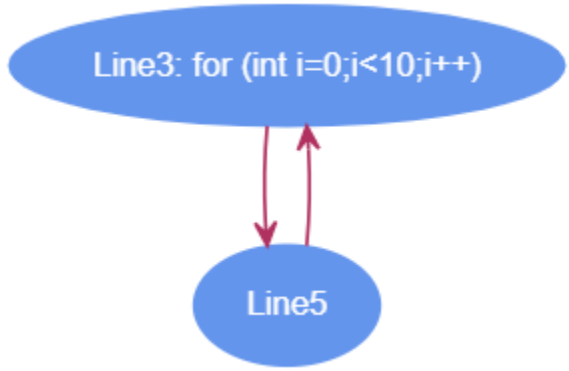
2. switch case statement

```
always @(posedge clk) begin
   case (state)
     IDLE:
     begin
       state <= IDLE;
     end
     STATE_1:
     begin
       state <= FINAL;
     end
     FINAL:
     begin
       state <= AcCESS;
     end
     default:
     begin
       state <= SETUP;
     end
   endcase
 end
```



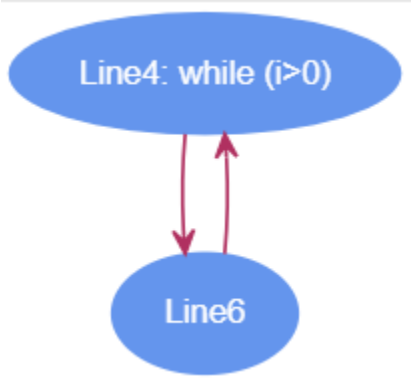
3. for loop

```
module tb;
  initial begin
    for (int i=0;i<10;i++)
    begin
      $display ("iteration [%0d]", i);
    end
  end
endmodule
```



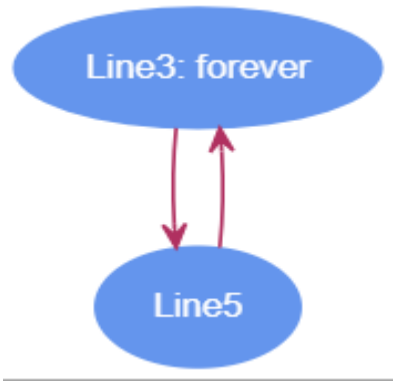
4. While loop

```
module mydesign;
  integer i = 5;
  initial begin
    while (i>0)
      begin
        $display ("Iteration#%0d", i);
        i = i-1;
      end
  end
endmodule
```



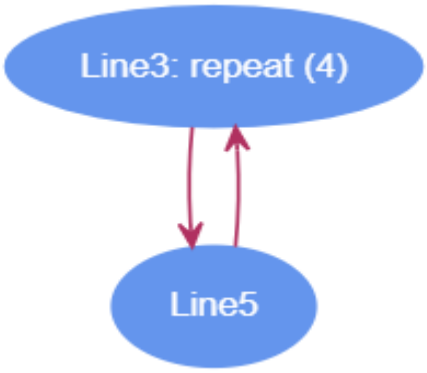
5. forever loop

```
module mydesign:
 initial begin
   forever
   begin
     $display("This will be printed")
   end
  end
endmodule
```



6. repeat loop

```
module mydesign:
  initial begin
    repeat (4)
      begin
        $display("there is a  system");
      end
  end
endmodule
```

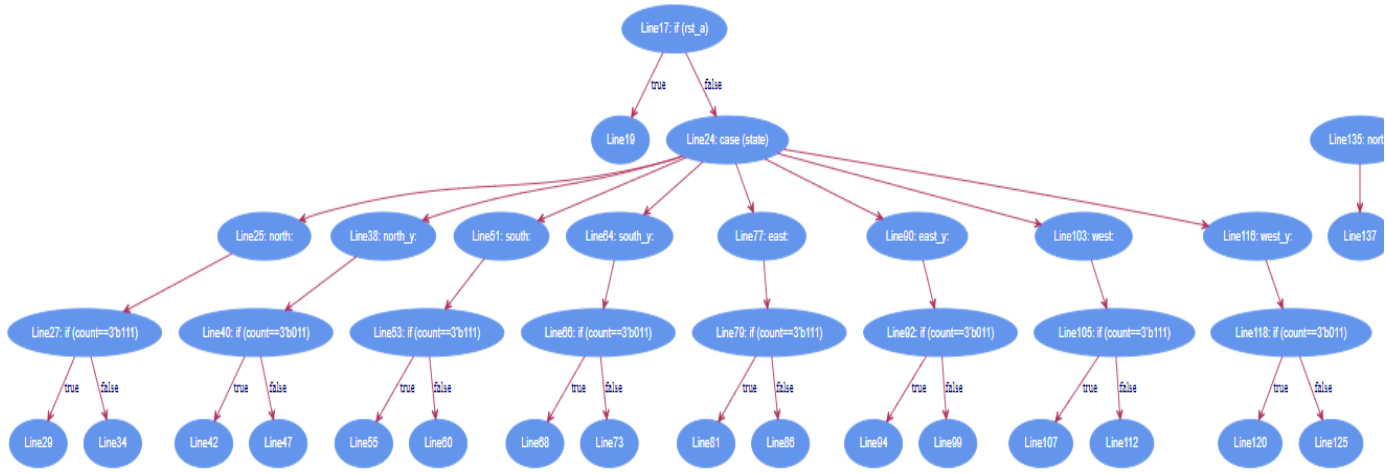


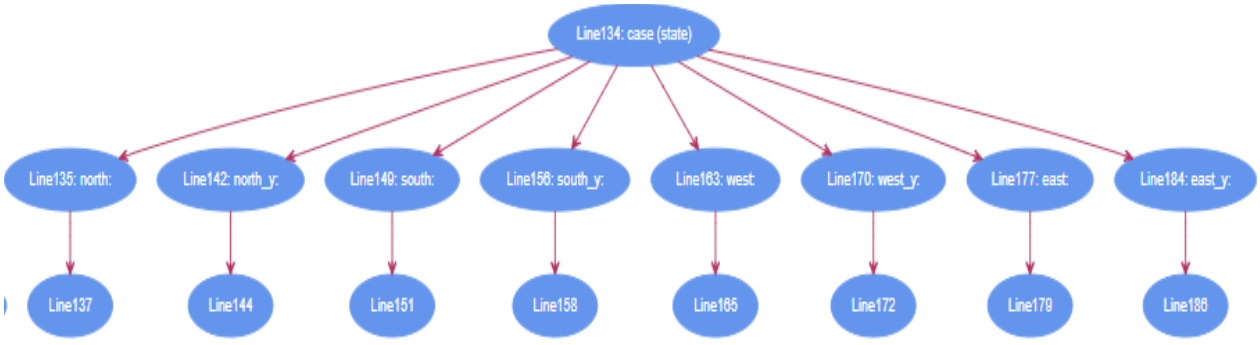
Control flow diagram for large input files

As the input file is large, we provide .txt files of verilog code in the zip files.

1. Traffic_light_controller verilog code

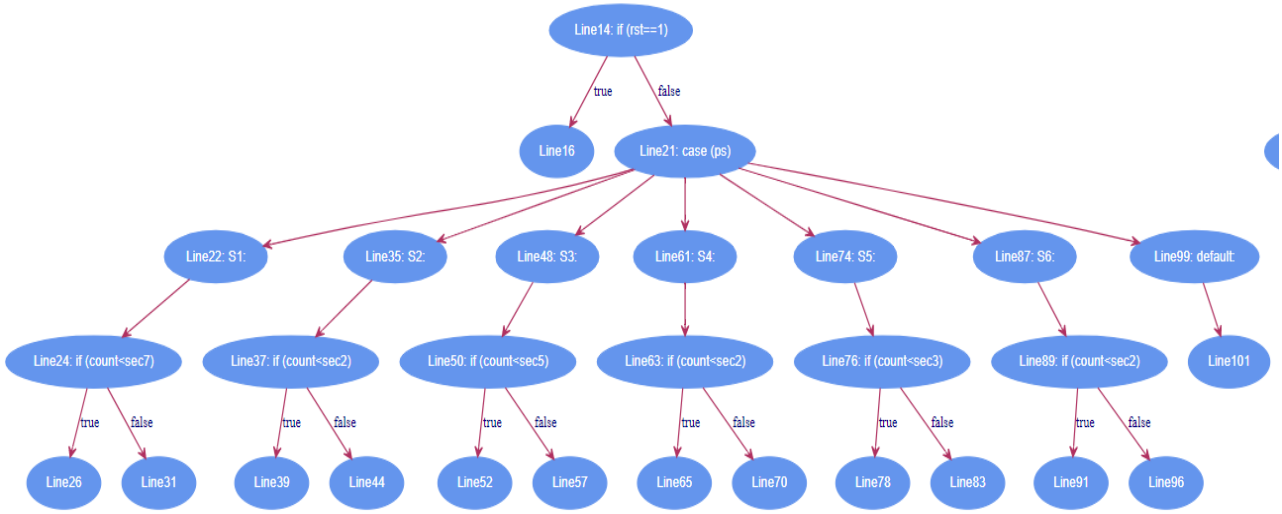
Control flow diagram:

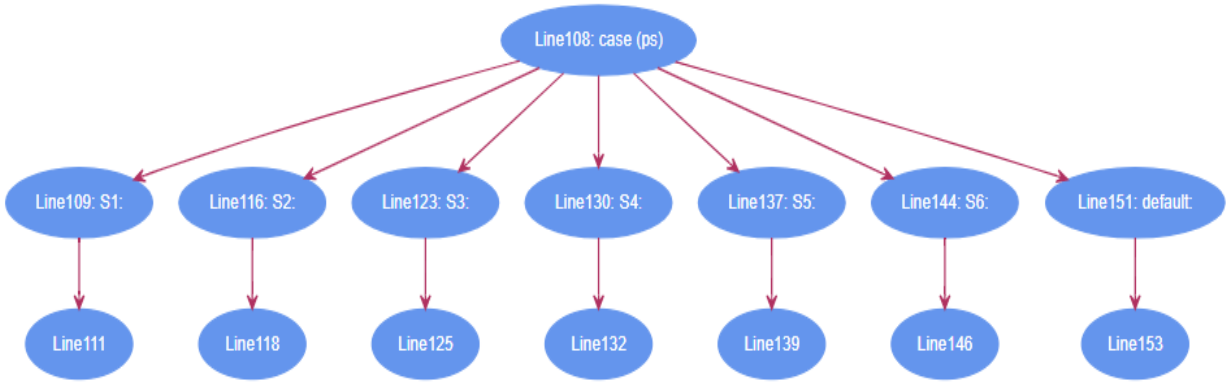




2. Case_1 verilog code

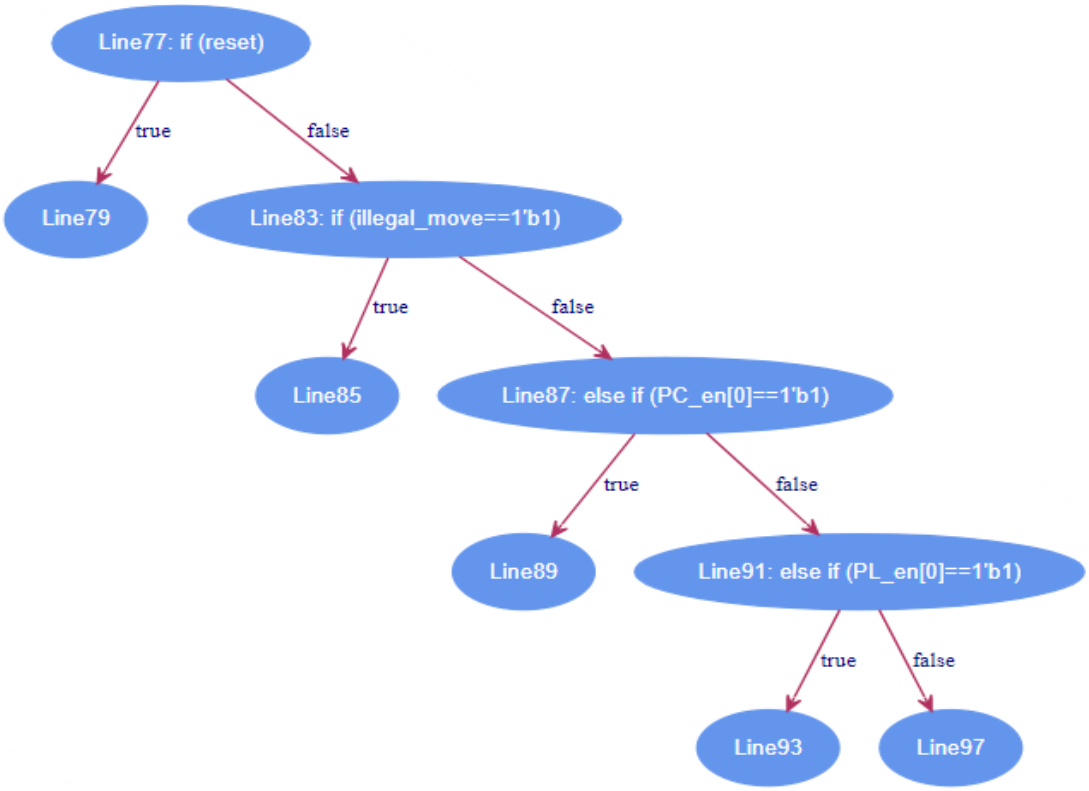
Control flow diagram:

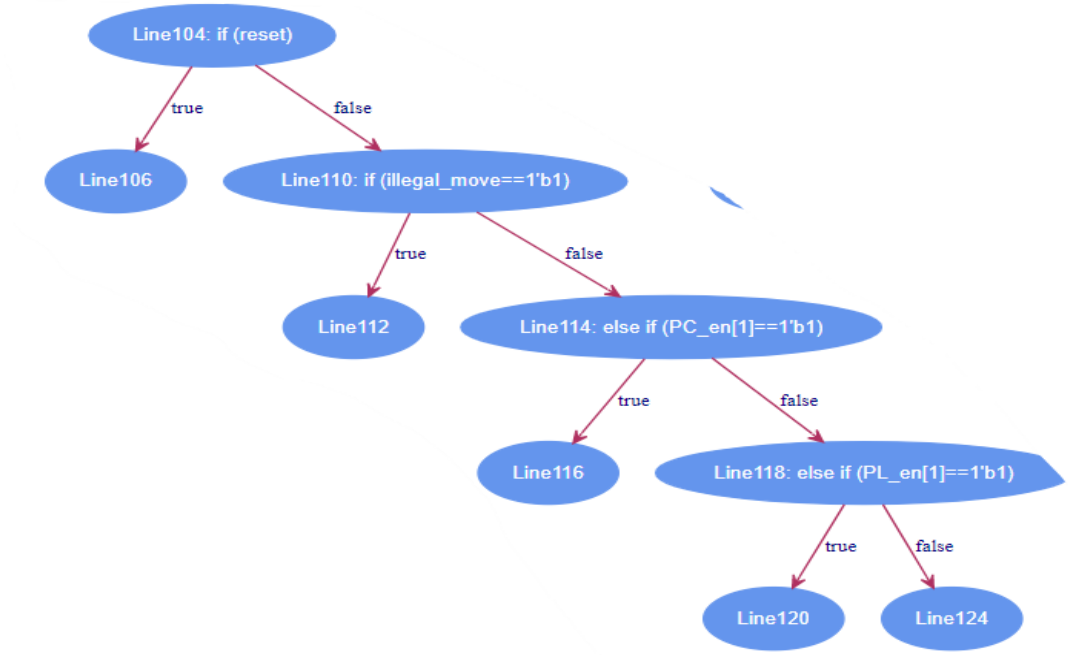


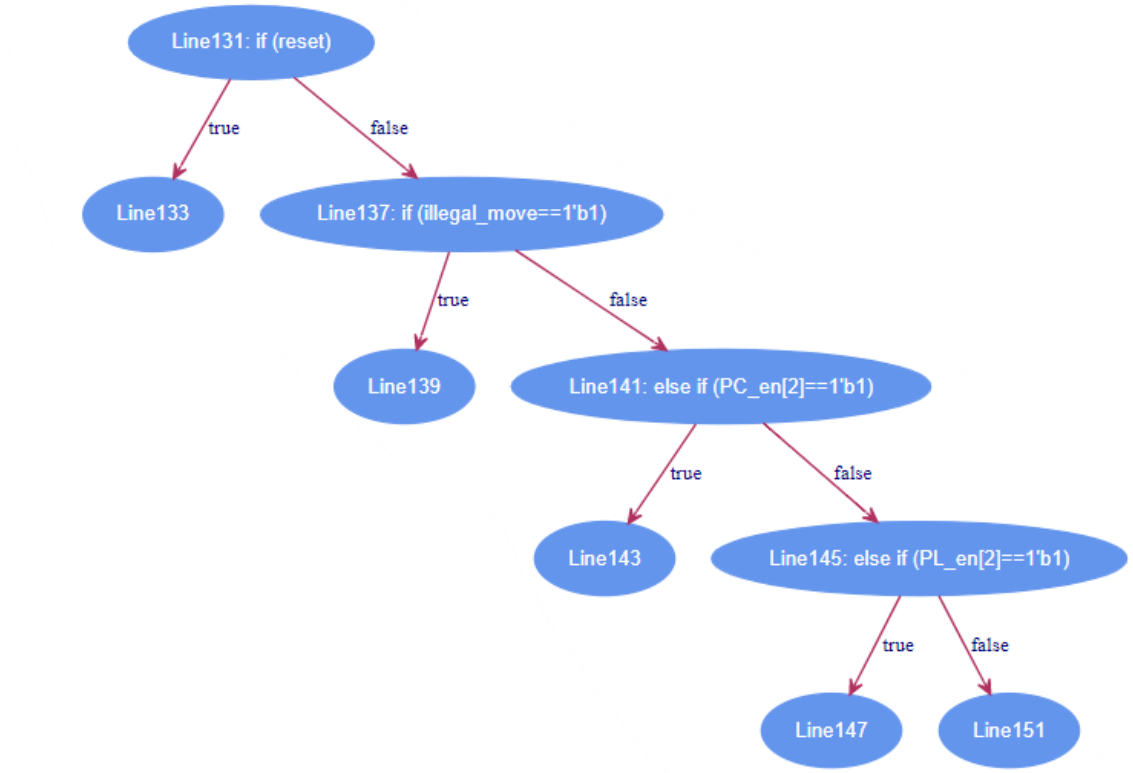


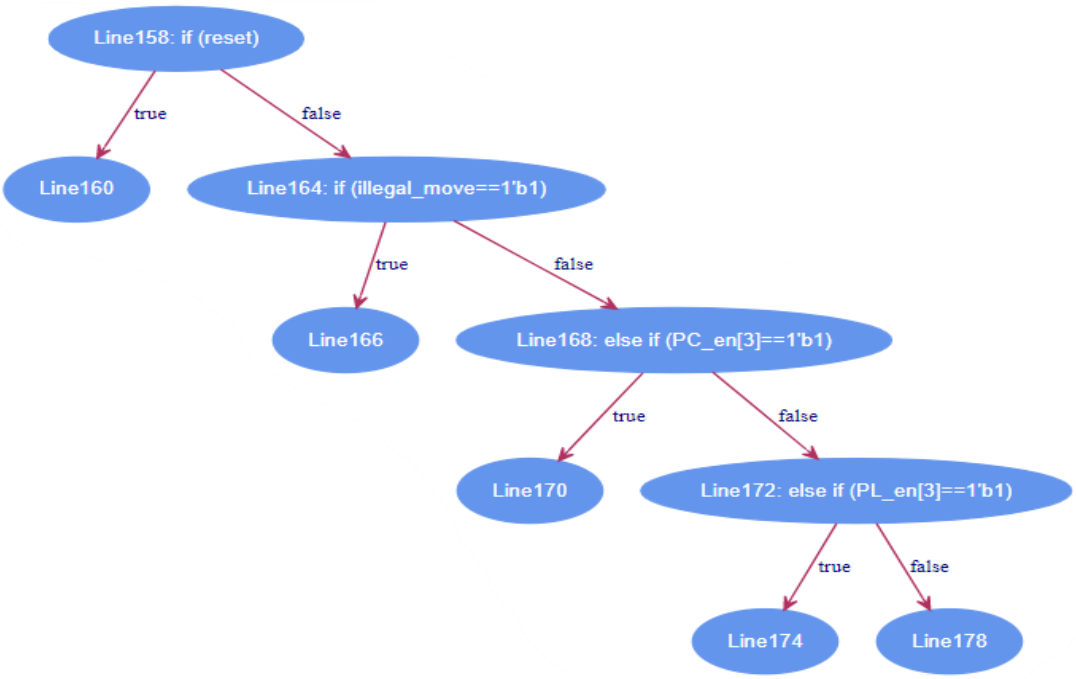
3. Tic_Tae_Toe_Game Verilog code

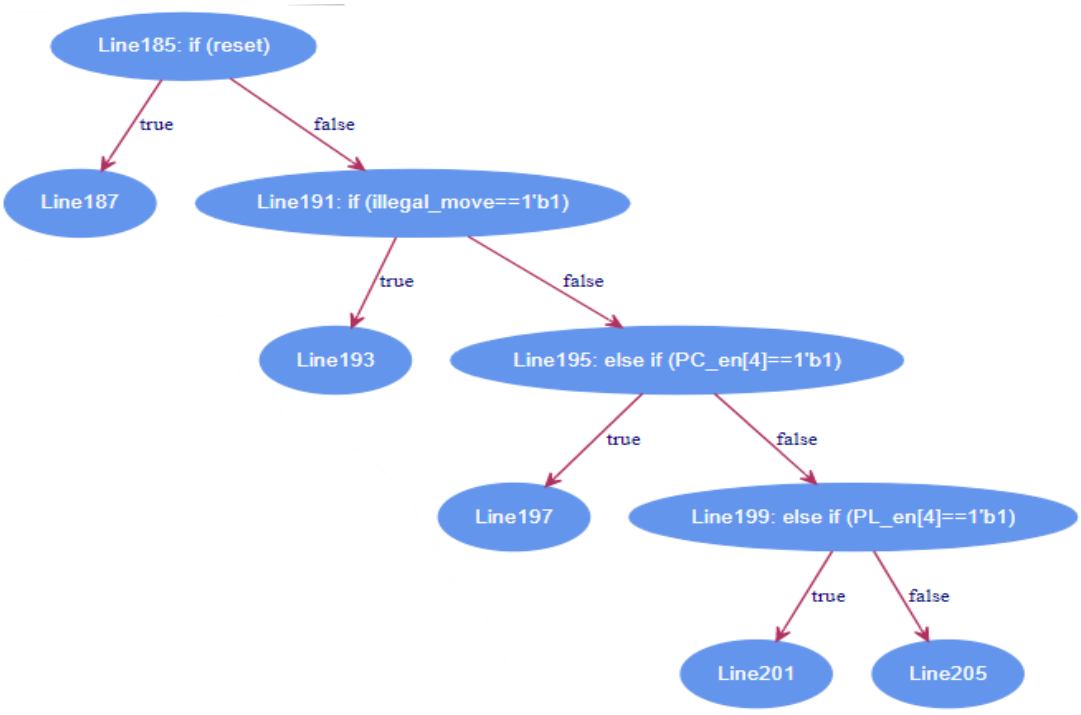
Control flow diagram:

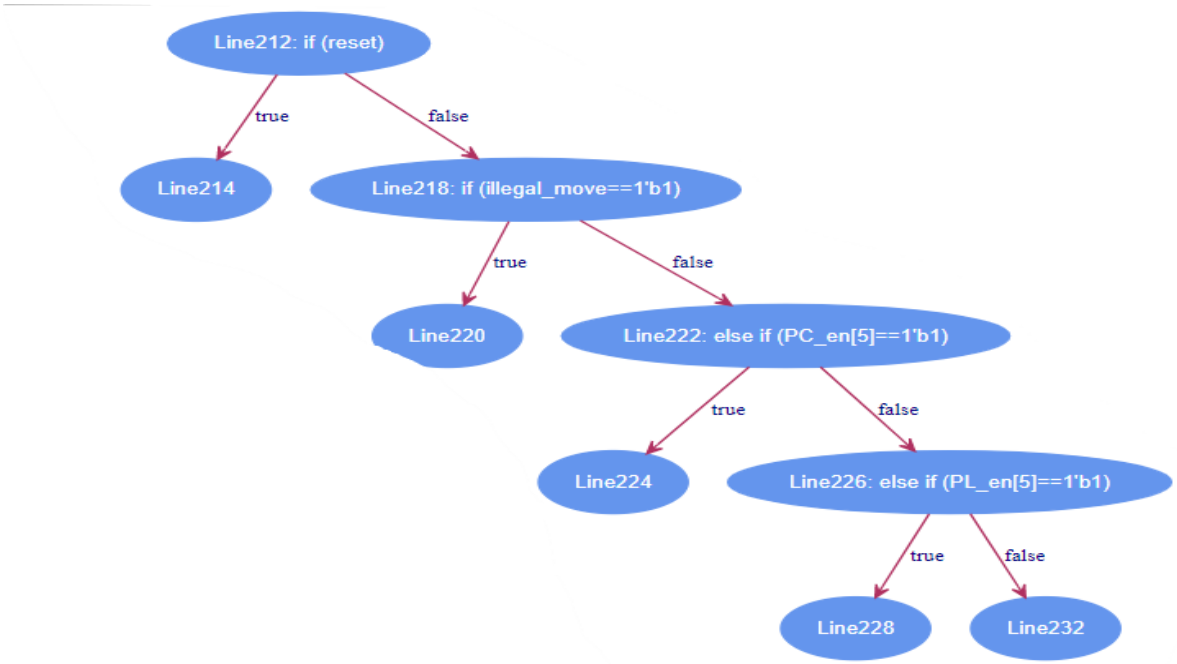


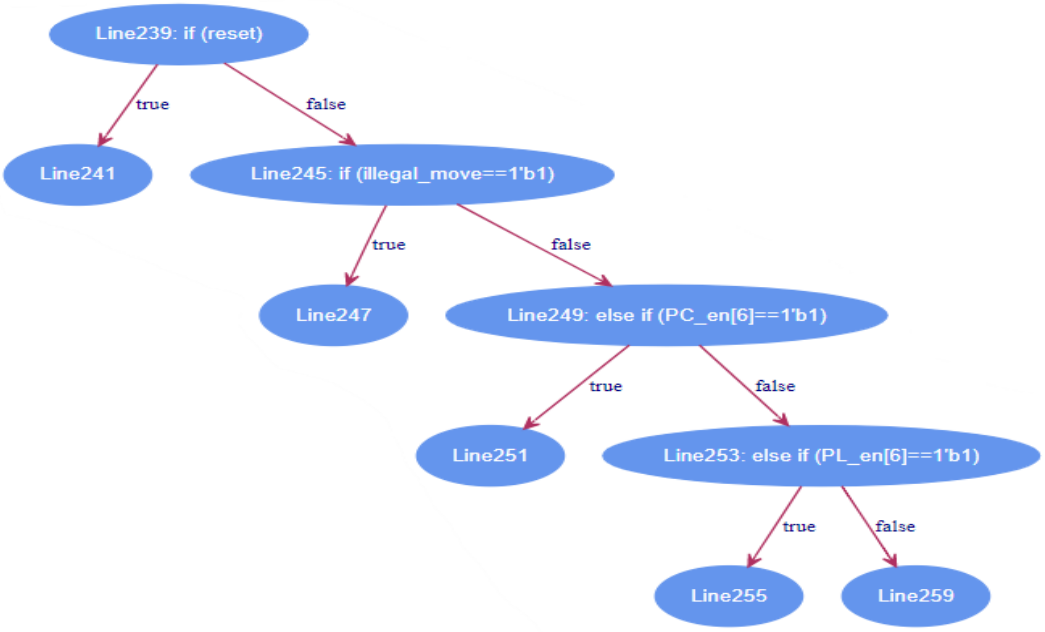

Line104: if (reset)
true
false
Line106
Line110: if (illegal_move==1'b1)
true
false
Line112
Line114: else if (PC_en[1]==1'b1)
true
false
Line116
Line118: else if (PL_en[1]==1'b1)
true
false
Line120
Line124

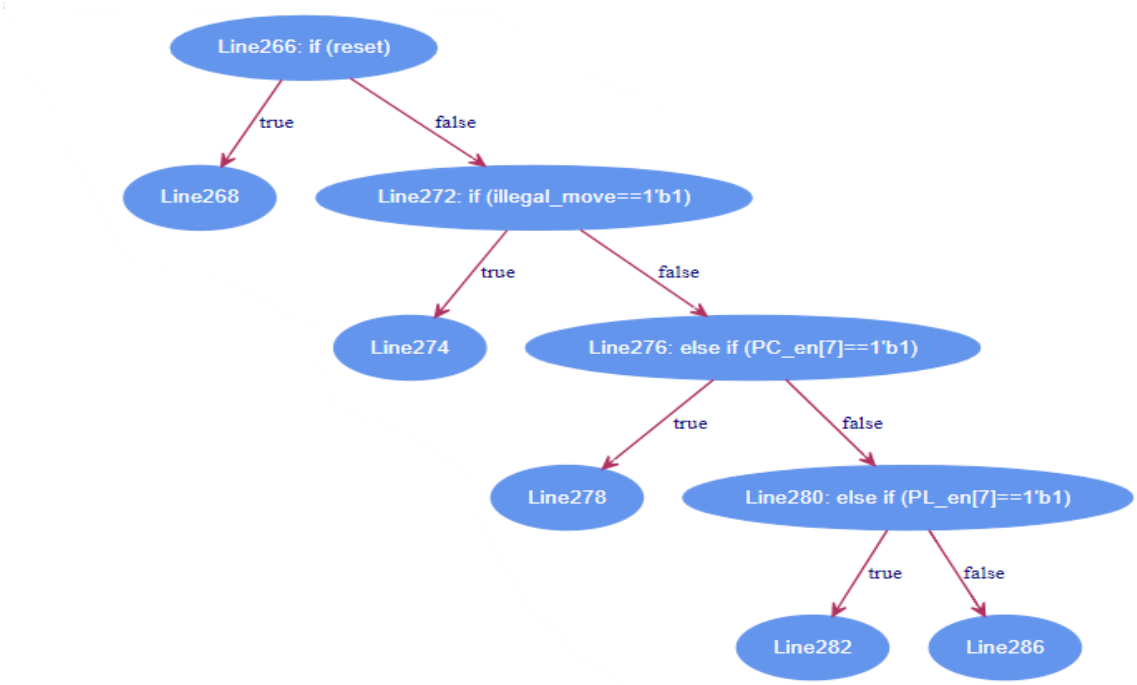

Line131: if (reset)
true
false
Line133
Line137: if (illegal_move==1'b1)
true
false
Line139
Line141: else if (PC_en[2]==1'b1)
true
false
Line143
Line145: else if (PL_en[2]==1'b1)
true
false
Line147
Line151

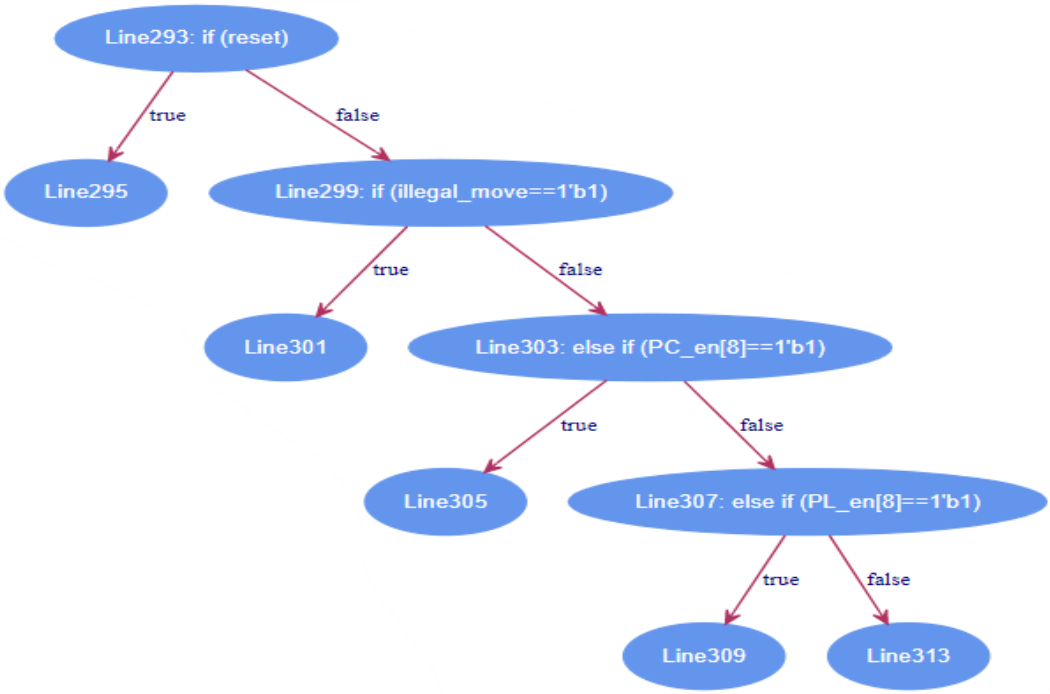

Line158: if (reset)
true
false
Line160
Line164: if (illegal_move==1'b1)
true
false
Line166
Line168: else if (PC_en[3]==1'b1)
true
false
Line170
Line172: else if (PL_en[3]==1'b1)
true
false
Line174
Line178

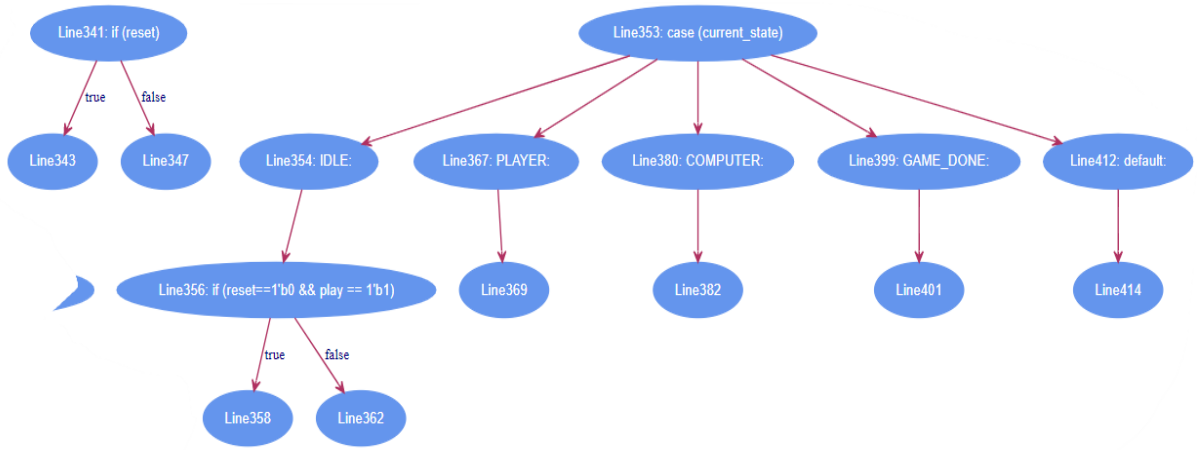

Line185: if (reset)
true
false
Line187
Line191: if (illegal_move==1'b1)
true
false
Line193
Line195: else if (PC_en[4]==1'b1)
true
false
Line197
Line199: else if (PL_en[4]==1'b1)
true
false
Line201
Line205

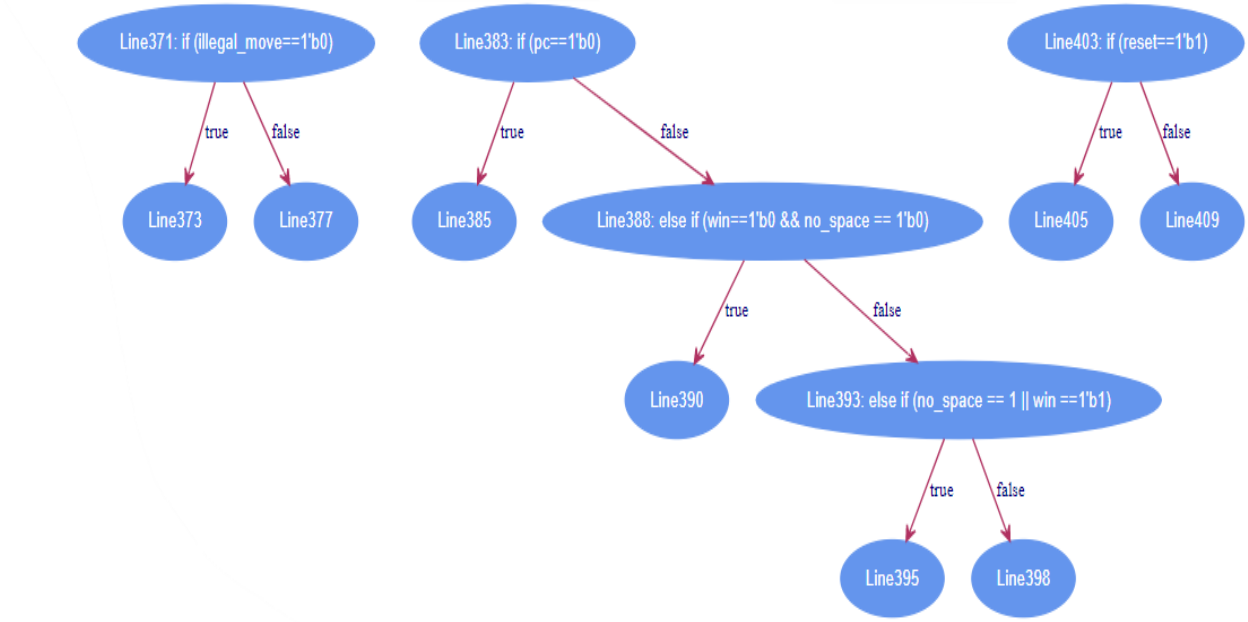

Line212: if (reset)
true
false
Line214
Line218: if (illegal_move==1'b1)
true
false
Line220
Line222: else if (PC_en[5]==1'b1)
true
false
Line224
Line226: else if (PL_en[5]==1'b1)
true
false
Line228
Line232


Line239: if (reset)
true
false
Line241
Line245: if (illegal_move==1'b1)
true
false
Line247
Line249: else if (PC_en[6]==1'b1)
true
false
Line251
Line253: else if (PL_en[6]==1'b1)
true
false
Line255
Line259


Line266: if (reset)
true
false
Line268
Line272: if (illegal_move==1'b1)
true
false
Line274
Line276: else if (PC_en[7]==1'b1)
true
false
Line278
Line280: else if (PL_en[7]==1'b1)
true
false
Line282
Line286


Line293: if (reset)
true
false
Line295
Line299: if (illegal_move==1'b1)
true
false
Line301
Line303: else if (PC_en[8]==1'b1)
true
false
Line305
Line307: else if (PL_en[8]==1'b1)
true
false
Line309
Line313


Line341: if (reset)
true
false
Line343
Line347
Line353: case (current_state)
Line354: IDLE:
Line367: PLAYER:
Line380: COMPUTER:
Line399: GAME_DONE:
Line412: default:
Line356: if (reset==1'b0 && play == 1'b1)
Line369
Line382
Line401
Line414
true
false
Line358
Line362


Line371: if (illegal_move==1'b0)
true
false
Line373
Line377
Line383: if (pc==1'b0)
true
false
Line385
Line388: else if (win==1'b0 && no_space == 1'b0)
true
false
Line390
Line393: else if (no_space == 1 || win ==1'b1)
true
false
Line395
Line398
Line403: if (reset==1'b1)
true
false
Line405
Line409