```
1   `timescale 1ns / 1ps
2   /**********************************************************************************
3    *
4    * Author:   Jesus Luciano & Rosswell Tiongco
5    * Filename: IU.v
6    * Date:     3/5/2019
7    * Version:  1.1
8    *
9    * Notes: 1.0 Instruction unit which contains a Program Counter, Instruction
10   *            Memory, Instruction Register and a sign extended output for
11   *            immediate instructions.
12   *
13   *        1.1 Added PC_mux to select data type to be loaded into PC, controlled
14   *            by new pc_sel control signal
15   *
16   **********************************************************************************/
17  module IU(clk, reset, im_cs, im_wr, im_rd, pc_ld, pc_inc, ir_ld, PC_in,
18                PC_out, IR_out, SE_16, pc_sel);
19
20      input         clk, reset, im_cs, im_wr, im_rd, pc_ld, pc_inc, ir_ld;
21      input  [ 1:0] pc_sel;
22      input  [31:0] PC_in;
23
24      output [31:0] PC_out, IR_out, SE_16;
25
26      wire   [31:0] IM_out, PC_mux;
27
28      //PC
29      reg32_inc   PC(.clk(clk), .reset(reset), .ld(pc_ld), .inc(pc_inc),
30                     .D(PC_mux), .Q(PC_out) );
31
32      //instruction mem
33      Memory IM(.clk(clk), .cs(im_cs), .wr(im_wr), .rd(im_rd),
34                      .Address(PC_out), .D_In(32'h0), .D_Out(IM_out) );
35
36      //module reg32(clk, reset, ld, D, Q);
37      //ir
38      reg32       IR(.clk(clk), .reset(reset), .ld(ir_ld),
39                     .D(IM_out), .Q(IR_out) );
40
41      //sign externsion of IR for immediate values
42      assign SE_16 = {{ 16{IR_out[15]}}, IR_out[15:0]};
43
44      //PC_mux to select data to load into PC
45      assign PC_mux = (pc_sel == 2'b00) ? (PC_out + {SE_16[29:0], 2'b00})      :
46                      (pc_sel == 2'b01) ? {PC_out[31:28], IR_out[25:0], 2'b00} :
47                                          PC_in;
48
49  endmodule
50
```