```
1   `timescale 1ns/100ps
2
3
4   module Program_Counter
5    #( parameter word_size = 8,OP0=2,OP1=3,OP2=4,OP3=6 )
6     (
7              output reg     [word_size-1: 0] Reg_PC ,
8              input  [1:0] Op_type,
9              input  [word_size/2-1:0]       data_in,
10             input  Load_PC_H,Load_PC_L, Inc_PC , Dec_PC, Reset_Vector,
11             input  clk, rst);
12
13  reg Lower_Byte ;
14  reg [word_size/2-1:0] temp ;
15
16  always @ (posedge clk)   // or negedge rst)
17    begin
18                  if (!rst)
19                    begin
20                              Reg_PC <= 16'hfffc;   //16'h0020;; // Reset Address Vector => PC_L=m[fffd],
    PC_H=m[fffc];
21           Lower_Byte<=1'b1;
22         end
23
24  /*
25     else if (Load_PC & !Lower_Byte)
26        begin
27        if(Reset_Vector)
28         Reg_PC[word_size-1:0] <= temp;
29        Reg_PC[word_size*2-1:word_size] = data_in;
30                            Lower_Byte<=1'b1;
31        end
32       else if (Load_PC & Lower_Byte)
33        begin
34         if(Reset_Vector)
35          temp <= data_in;
36         else
37          begin
38          Reg_PC[word_size-1:0] <= data_in;
39               Lower_Byte<=1'b0;
40          end
41       end
42        */
43                              /*
44       else if (Load_PC)
45        begin
46         if(Inc_PC & Lower_Byte)
47          begin
48           temp = data_in;
49           Lower_Byte=1'b0;
50           Reg_PC = Reg_PC + 1'b1;
51          end
52
53         else if (Inc_PC & !Lower_Byte)
54                    begin
55                    Reg_PC[word_size-1:0] = {data_in,temp};
56                    Lower_Byte =1'b1;
57                    end
58
```

```
59                 else if (Lower_Byte)
60          begin
61           Reg_PC[word_size/2-1:0] = data_in;
62          // Lower_Byte<=1'b0;
63          end
64                                      */
65               /*
66        else if (!Lower_Byte)
67                  begin
68                  Reg_PC[word_size-1:0] = {data_in,temp};
69                  Lower_Byte =1'b1;
70                  end */
71                  else if (Load_PC_L)
72                       begin
73                        if(Reset_Vector)
74              begin
75                temp = data_in;
76                Reg_PC = Reg_PC + 1'b1;
77              end
78                       else
79                         Reg_PC[word_size/2-1:0] = data_in;
80                 end
81
82           else if (Load_PC_H)
83                       begin
84                               if(Reset_Vector)
85                       Reg_PC[word_size-1:0] = {data_in,temp};
86                               else
87                                 Reg_PC[word_size-1:word_size/2] = data_in;
88                       end
89                   else if (Inc_PC)
90                           Reg_PC = Reg_PC + 1'b1;   // Op_type + 2;  // OP_TYPE
91                   else if (Dec_PC)
92                           Reg_PC = Reg_PC - 1'b1;
93           end
94
95
96  endmodule
97
98
```