EEE3050 Theory on Computer Architectures (Spring 2017)
Prof. Jinkyu Jeong

HW2_Helper

2017.05.19

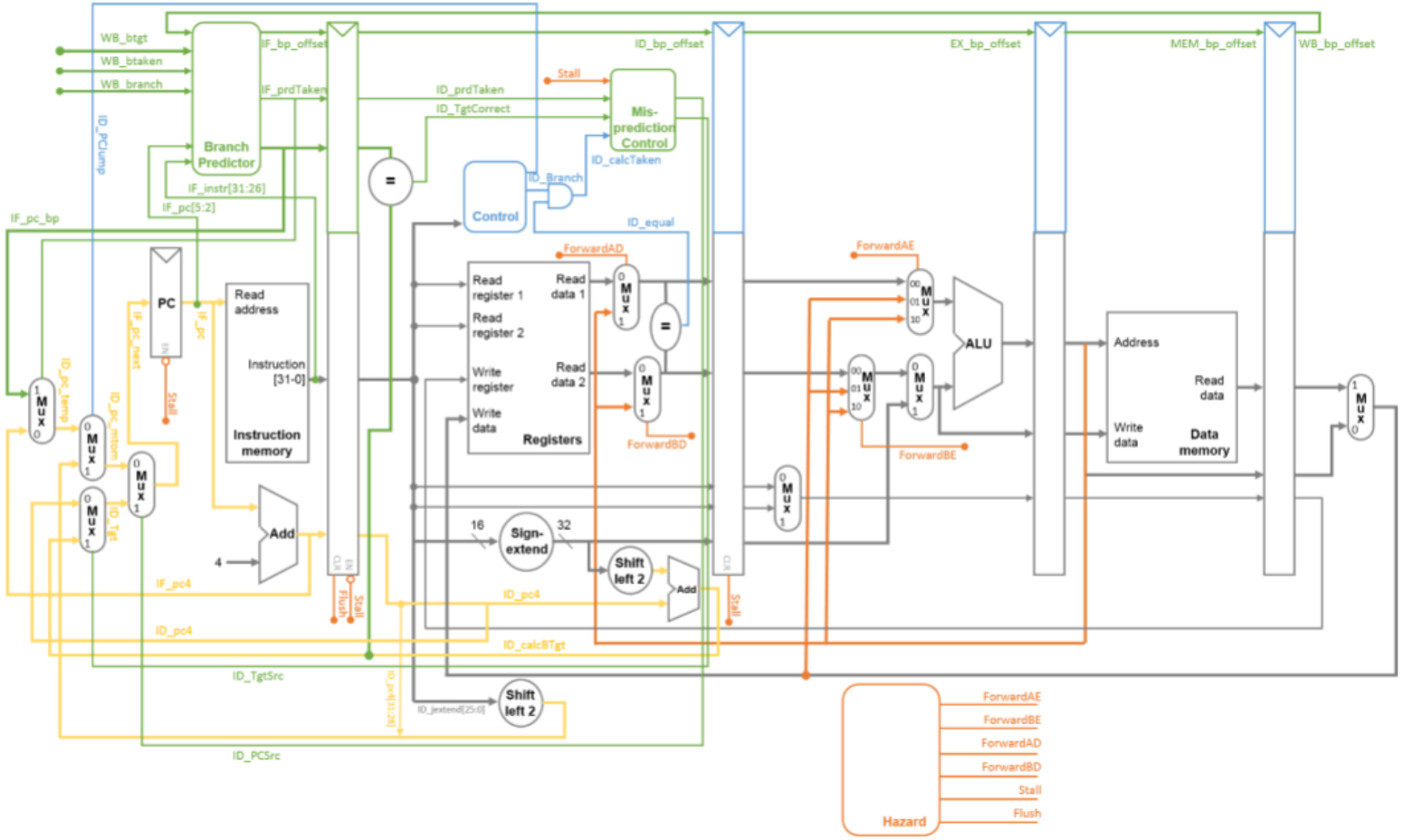
TA 이규선(GYUSUN LEE) / 안민우(MINWOO AHN)

Your questions all about…

What I have to do…?

How can I do…?

What you have to do!



What you have to do! (Part B)

```
module  hazard (H_ID_rs, H_ID_rt, H_EX_rt, H_EX_rs,
            H_MEM_RegWrite, H_WB_RegWrite,
            H_EX_RegRd, H_MEM_RegRd, H_WB_RegRd, H_EX_MemtoReg,
H_MEM_MemtoReg, H_ID_PCSrc,H_ID_PCJump,
            H_ID_Branch, H_EX_RegWrite, H_Stall,H_Flush,
            H_ForwardAE, H_ForwardBE, H_ForwardAD, H_ForwardBD);

    input [4:0] H_ID_rs, H_ID_rt, H_EX_rt, H_EX_rs, H_EX_RegRd, H_MEM_RegRd,
H_WB_RegRd;
    input H_MEM_RegWrite, H_WB_RegWrite, H_EX_RegWrite, H_ID_Branch,
H_EX_MemtoReg, H_MEM_MemtoReg, H_ID_PCSrc, H_ID_PCJump;
    output reg [1:0] H_ForwardAE, H_ForwardBE;
    output reg H_ForwardAD, H_ForwardBD;
    output reg H_Stall, H_Flush;
    initial
        begin
            H_ForwardAE  = 2'b00;
            H_ForwardBE  = 2'b00;
            H_ForwardAD  = 1'b0;
            H_ForwardBD  = 1'b0;
            H_Stall = 1'b0;
            H_Flush = 1'b0;
        end
endmodule
```

Format of inputs

[H]_[Stage where input used]_[Description of input]

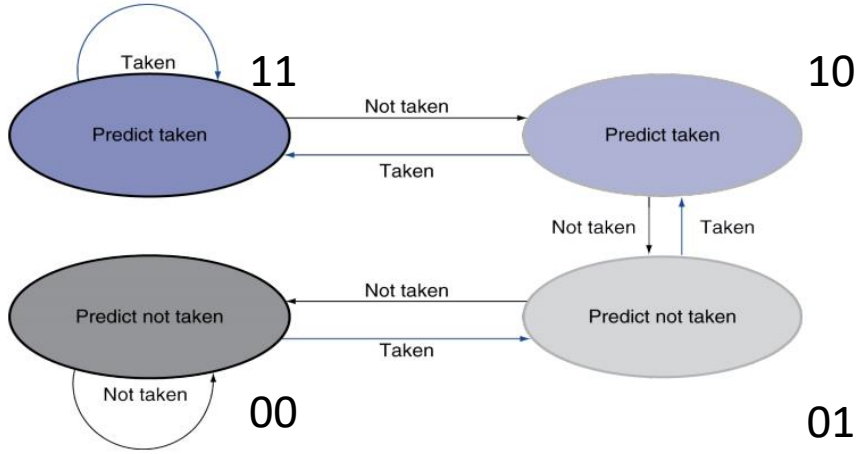
- ForwardAD, ForwardBD : Forward from Mem stage
- ForwardAE, ForwardBE : Forward from Mem or WB stage
- e.g. ForwardAE = 2’b00 => not hazard
  ForwardAE = 2’b01 => Forward from WB
  ForwardAE = 2’b10 => Forward from Mem

What you have to do! (Part C)

branch_predictor module(branch_prediction.v)

```
input   [31:0]  WB_btgt;
input   [5:0]  IF_opcode;
input   [3:0]  WB_bp_offset,IF_pc_offset;
input   clk,WB_branch,WB_btaken,halt;
output  reg  [31:0]  IF_pc_out;
output  reg[3:0]  IF_bp_offset;
output  reg  IF_prdTaken;
reg  [1:0]  prediction_buffer  [0:15];
reg  [31:0]  target_buffer  [0:15];
integer  fp,  i;
// Do not delete:  initialize  the predictor  and the target  buffer
initial
    begin
        for(i  = 0;i  <  16;  i  = i+1)
            begin
                prediction_buffer  [i]  = 2'b01;
                target_buffer  [i]  = 32'b0;
        end
    end
```

prediction_buffer



target_buffer

pc to jump(branch inst. 1)
pc to jump(branch inst. 2)
pc to jump(branch inst. 3)
pc to jump(branch inst. 4)
pc to jump(branch inst. 5)

What you have to do! (Part C) – cont.

```
   // Part C: output prediction  and target  address
   always @(IF_pc_offset  or IF_opcode)
      begin
 //
=============================================================
============================ //
 //                                                          //
 //    WRITE  YOUR CODES BELOW!!!!!!!!!!!
//
 //    Please write  your code to make branch  predictor  work  properly
//
 //                                                          //
 //
=============================================================
============================ //
         IF_pc_out  = 0;
         IF_prdTaken  = 0;
         IF_pc_out=32'bx;
      end
```

```
   // Part C: update the predictor  and target  buffer
   always @ (negedge  clk)
      begin
 //
=============================================================
============================ //
 //                                                          //
 //    WRITE  YOUR CODES BELOW!!!!!!!!!!!
//
 //    Please write  your code to make branch  predictor  work  properly
//
 //                                                          //
 //
=============================================================
============================ //

      end
endmodule
```

What you have to do! (Part C) – cont.

```
module  misprediction_control
(clk,prdTaken,calcTaken,tgtCorrect,PCsrc,TgtSrc,isBranch,stall);
   input   clk,prdTaken,calcTaken,tgtCorrect,isBranch,stall;
   output  reg  PCsrc,  TgtSrc;

    // Part C: misprediction  controller
    always @(prdTaken  or calcTaken  or tgtCorrect)
    begin
  //
============================================================
============================ //
  //                                                        //
  //     WRITE  YOUR CODES BELOW!!!!!!!!!!!
//
  //     Please write  your code to make misprediction  controller  work properly
//
  //                                                        //
  //
============================================================
============================ //
        PCsrc = calcTaken;
          TgtSrc  = isBranch;

    end

endmodule
```