module jumppy(ret,clk,intialpc,instr,jump,pc,jal);

input clk;

input[31:0] instr;

input[31:0] intialpc;

input jump;

input ret;

input jal;

wire ab;

wire[31:0] pcinc;

wire[31:0] jumpadd;

wire[31:0] pcshift;

output[31:0] pc;

reg[31:0] pc;

reg[31:0] npc;

reg[31:0] retpc;

reg[31:0] h=32'h4;

reg[31:0] m=32'd31;

Adder32Bit ads(intialpc,h,pcinc,ab);

assign pcshift = ((pcinc>>28)<<28);

assign jumpadd = pcshift+(instr[25:0]<<2);

always@(jump or jumpadd or pcinc)

begin

if(jump==1)

npc=jumpadd;

else

npc=pcinc;

end

always@(jal or pcinc)

begin

if(jal==1)

retpc=pcinc;

end

endmodule

module Adder32Bit(input1, input2, out, overflowBit);

input [31:0] input1, input2;

output [31:0] out;

reg [31:0]out;

output overflowBit;

reg overflowBit;

always@(input1 or input2)

begin

{overflowBit , out } = input1 + input2;

end

endmodule