'h15: // custom instruction: division by 13

begin

sign = s\_int[31:31]; //stores the sign of the input

c = 10083 \* s\_int; //multiplies and the divides as a

result = (c >> 17);

result[31:15] = {sign, sign, sign, sign, sign, sign, sign, sign, sign, sign, sign, sign, sign, sign, sign, sign, sign};

end

result = ((result) - ((zero) + \*(int \*) 0x12344321));

begin

// WB STAGE

if (MEMWBRegWrite == 1'b1 &&

MEMWBWriteRegister == ifidreadregister1 &&

MEMWBWriteRegister != 0)

forwardA = 2'b11; //3

if (MEMWBRegWrite == 1'b1 &&

MEMWBWriteRegister == ifidreadregister2 &&

MEMWBWriteRegister != 0)

forwardB = 2'b11; //3

if (EXMEMRegWrite == 1'b1 &&

EXMEMWriteRegister == ifidreadregister1 &&

EXMEMWriteRegister != 0)

// MEM STAGE - A

begin

if(MEMstage\_hazard\_ctrl !=0)

hazard = 1'b1;

else

forwardA = 2'b10; //2

end

if (EXMEMRegWrite == 1'b1 &&

EXMEMWriteRegister == ifidreadregister2 &&

EXMEMWriteRegister != 0)

// MEM STAGE - B

begin

if(MEMstage\_hazard\_ctrl != 0)

hazard = 1'b1;

else

forwardB = 2'b10; // 2

end

//EX hazard

if (IDEXRegWrite == 1 && (IDEXRegDst == 2'b00 && IDEXWriteRegisterRt == ifidreadregister1 ||

IDEXRegDst == 2'b01 && IDEXWriteRegisterRd == ifidreadregister1))

begin

if(EXstage\_hazard\_ctrl !=0)

hazard = 1'b1;

else

forwardA = 2'b01; // 1

end

if (IDEXRegWrite == 1 && (IDEXRegDst == 2'b00 && IDEXWriteRegisterRt == ifidreadregister2 ||

IDEXRegDst == 2'b01 && IDEXWriteRegisterRd == ifidreadregister2))

begin

if(EXstage\_hazard\_ctrl !=0)

hazard = 1'b1;

else

forwardB = 2'b01; // 1

end

end