// adding 10,20, 30 storing them in r1,r2 and r3 then storing sum in r4 and r5

module test\_mips32;

reg clk1,clk2;

integer k;

pipe\_MIPS32 mips (clk1,clk2);

initial

begin

clk1=0; clk2=0;

repeat (20) //generating two phase clock

begin

#5 clk1=1; #5 clk1=0;

#5 clk2=1; #5 clk2=0;

end

end

initial

begin

for (k=0; k<31; k++)

mips.Reg[k]=k;

mips.Mem[0] = 32'h2801000a; // ADDI R1,R0,10

mips.Mem[1] = 32'h28020014; // ADDI R2,R0,20

mips.Mem[2] = 32'h28030019; // ADDI R3,R0,25

mips.Mem[3] = 32'h0ce77800; // OR R7,R7,R7(DUMMY)

mips.Mem[4] = 32'h0ce77800;//OR R7,R7,R7 (DUMMY)

mips.Mem[5] = 32'h00222000;//ADD R4,R1,R2

mips.Mem[6] = 32'h0ce77800;//OR R7, R7,R7 (DUMMY)

mips.Mem[7] = 32'h00832800;//ADD R5, R4,R3

mips.Mem[8] = 32'hfc000000;//HLT

mips.HALTED =0;

mips.PC=0;

mips.TAKEN\_BRANCH = 0;

#280

for (k=0; k<6; k++)

$display ("R%1d - %2d", k, mips.Reg[k]);

end

initial

begin

$dumpfile ("mips.vcd");

$dumpvars (0, test\_mips32);

#300 $finish;

end

endmodule