// factorial of a number n

module test\_mips32;

reg clk1,clk2;

integer k;

pipe\_MIPS32 mips (clk1,clk2);

initial

begin

clk1=0; clk2=0;

repeat (40) //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'h280a00c8; // ADDI R10,R0,200

mips.Mem[1]=32'h28020001; // ADDI R2,R0,1

mips.Mem[2]=32'h0e94a000;// OR R20 ,R20,R20 DUMMY

mips.Mem[3]=32'h21430000;//LW R3,0(R10)

mips.Mem[4]=32'h0e94a000;//OR R20 ,R20,R20 DUMMY

mips.Mem[5]=32'h14431000;//LOOP :MUL R2,R2,R3

mips.Mem[6]=32'h2c630001;//R3,R3,1

mips.Mem[7]=32'h0e94a000;//OR R20 ,R20,R20 DUMMY

mips.Mem[8]=32'h3460fffc;//BNEQZ R3,LOOP (I.E. -4 OFFSET)

mips.Mem[9]=32'h2542fffe;// SW R2,-2(R10)

mips.Mem[10]=32'hfc000000; // HLT

mips.Mem[200]=7; //find the factorial of 'N';

mips.PC=0;

mips.HALTED=0;

mips.TAKEN\_BRANCH=0;

#2000 $display ("Mem[200] = %2d ,Mem[198] = %6d",mips.Mem[200],mips.Mem[198]);

end

initial

begin

$dumpfile ("mips.vcd");

$dumpvars (0,test\_mips32);

$monitor ("R2 : %4d",mips.Reg[2]);

#3000 $finish;

end

endmodule