

```

module TestEnvironment(

    input clock, clock2,
    input [7:0] dipSW_A, dipSW_B,
    input [3:0] rotSW_A, rotSW_B,
    input [4:0] pushSW_A, pushSW_B, pushSW_C, pushSW_D,

    output [7:0] LED7segA_D, LED7segE_H,
    output [3:0] slctA_D, slctE_H,
    output [7:0] LED,
    output BZ,

    output [7:0] LED7ep0, LED7ep1, LED7ep2, LED7ep3,
                                LED7ep4, LED7ep5, LED7ep6, LED7ep7,
    output [7:0] slctEp
);
//#####
// outputs

wire [7:0] LED_A, LED_B, LED_C, LED_D,
            LED_E, LED_F, LED_G, LED_H;

wire [7:0] LED_exA[0:7], LED_exB[0:7],
            LED_exC[0:7], LED_exD[0:7],
            LED_exE[0:7], LED_exF[0:7],
            LED_exG[0:7], LED_exH[0:7];

/*
output [7:0] LED,
output BZ
*/
////////////////////////////////////
// Usable wires

/*
input clock,clock2,
input [7:0] dipSW_A, dipSW_B,
input [3:0] rotSW_A, rotSW_B,
input [4:0] pushSW_A, pushSW_B, pushSW_C, pushSW_D,
*/

wire clock_x4,
        clock_x8,

```

```

        clock_x10,
        clock_x100,
        clock_x1000,
        clock_x2p16;

wire vdd;

////////////////////////////////////
//#####
###
//#####
###
// User Space

wire [15:0] pc,inst,mem_in,WBdata,result,LDresult,rs_ra,rd_rb,nextPC;
wire [5:0] control;
wire [2:0] WBadd;
wire RegWrite,branch;

// chatting
wire chattering_wire;
chattering c(.in(!pushSW_B[4]), .clock(clock), .out(chattering_wire));

MicroComputer_Block mc(.clock(clock2), .reset(pushSW_A[4]),
                        .exec(chattering_wire),
                        .PC_next(pc),.instruction(inst),
                        .control(control),.mem_in(mem_in)
                        ,.result(result),.Rs_Ra(rs_ra),.Rd_Rb(rd_rb),
                        .LDresult(LDresult),.WBdata(WBda
                        ta),.WBadd(WBadd),.RegWrite(RegWrite),
                        .IMnextPC(nextPC),.branch(branc
                        h));

assign LED[0] = control[0];
assign LED[1] = control[1];
assign LED[2] = control[2];
assign LED[3] = control[3];
assign LED[4] = control[4];
assign LED[5] = control[5];
assign LED[6] = 0;

```

```
assign LED[7] = RegWrite;
```

```
LED7segDecoder dA(.number(pc[15:12]),.led7seg(LED_A));  
LED7segDecoder dB(.number(pc[11:8]),.led7seg(LED_B));  
LED7segDecoder dC(.number(pc[7:4]),.led7seg(LED_C));  
LED7segDecoder dD(.number(pc[3:0]),.led7seg(LED_D));
```

```
LED7segDecoder dE(.number(inst[15:12]),.led7seg(LED_E));  
LED7segDecoder dF(.number(inst[11:8]),.led7seg(LED_F));  
LED7segDecoder dG(.number(inst[7:4]),.led7seg(LED_G));  
LED7segDecoder dH(.number(inst[3:0]),.led7seg(LED_H));
```

```
LED7segDecoder exA0(.number(rs_ra[15:12]),.led7seg(LED_exA[0]));  
LED7segDecoder exA1(.number(rs_ra[11:8]),.led7seg(LED_exA[1]));  
LED7segDecoder exA2(.number(rs_ra[7:4]),.led7seg(LED_exA[2]));  
LED7segDecoder exA3(.number(rs_ra[3:0]),.led7seg(LED_exA[3]));
```

```
LED7segDecoder exA4(.number(rd_rb[15:12]),.led7seg(LED_exA[4]));  
LED7segDecoder exA5(.number(rd_rb[11:8]),.led7seg(LED_exA[5]));  
LED7segDecoder exA6(.number(rd_rb[7:4]),.led7seg(LED_exA[6]));  
LED7segDecoder exA7(.number(rd_rb[3:0]),.led7seg(LED_exA[7]));
```

```
LED7segDecoder exB0(.number(result[15:12]),.led7seg(LED_exB[0]));  
LED7segDecoder exB1(.number(result[11:8]),.led7seg(LED_exB[1]));  
LED7segDecoder exB2(.number(result[7:4]),.led7seg(LED_exB[2]));  
LED7segDecoder exB3(.number(result[3:0]),.led7seg(LED_exB[3]));
```

```
LED7segDecoder exB4(.number(mem_in[15:12]),.led7seg(LED_exB[4]));  
LED7segDecoder exB5(.number(mem_in[11:8]),.led7seg(LED_exB[5]));  
LED7segDecoder exB6(.number(mem_in[7:4]),.led7seg(LED_exB[6]));  
LED7segDecoder exB7(.number(mem_in[3:0]),.led7seg(LED_exB[7]));
```

```
LED7segDecoder exC0(.number(WBadd[2]),.led7seg(LED_exC[0]));  
LED7segDecoder exC1(.number(WBadd[1]),.led7seg(LED_exC[1]));  
LED7segDecoder exC2(.number(WBadd[0]),.led7seg(LED_exC[2]));  
//LED7segDecoder exC3(.number(),.led7seg(LED_exC[3]));  
LED7segDecoder exC4(.number(WBdata[15:12]),.led7seg(LED_exC[4]));  
LED7segDecoder exC5(.number(WBdata[11:8]),.led7seg(LED_exC[5]));  
LED7segDecoder exC6(.number(WBdata[7:4]),.led7seg(LED_exC[6]));  
LED7segDecoder exC7(.number(WBdata[3:0]),.led7seg(LED_exC[7]));
```

```
//LED7segDecoder exD0(.number(cd[3]),.led7seg(LED_exD[0]));
```

```
//LED7segDecoder exD1(.number(cd[2]),.led7seg(LED_exD[1]));
LED7segDecoder exD2(.number(branch),.led7seg(LED_exD[2]));
//LED7segDecoder exD3(.number(cd[0]),.led7seg(LED_exD[3]));

LED7segDecoder exD4(.number(nextPC[15:12]),.led7seg(LED_exD[4]));
LED7segDecoder exD5(.number(nextPC[11:8]),.led7seg(LED_exD[5]));
LED7segDecoder exD6(.number(nextPC[7:4]),.led7seg(LED_exD[6]));
LED7segDecoder exD7(.number(nextPC[3:0]),.led7seg(LED_exD[7]));
```

```
assign LED_exE[0] = 8'b0;
assign LED_exE[1] = 8'b0;
assign LED_exE[2] = 8'b0;
assign LED_exE[3] = 8'b0;
assign LED_exE[4] = 8'b0;
assign LED_exE[5] = 8'b0;
assign LED_exE[6] = 8'b0;
assign LED_exE[7] = 8'b0;
```

```
assign LED_exF[0] = 8'b0;
assign LED_exF[1] = 8'b0;
assign LED_exF[2] = 8'b0;
assign LED_exF[3] = 8'b0;
assign LED_exF[4] = 8'b0;
assign LED_exF[5] = 8'b0;
assign LED_exF[6] = 8'b0;
assign LED_exF[7] = 8'b0;
```

```
assign LED_exG[0] = 8'b0;
assign LED_exG[1] = 8'b0;
assign LED_exG[2] = 8'b0;
assign LED_exG[3] = 8'b0;
assign LED_exG[4] = 8'b0;
assign LED_exG[5] = 8'b0;
assign LED_exG[6] = 8'b0;
assign LED_exG[7] = 8'b0;
```

```
assign LED_exH[0] = 8'b0;
assign LED_exH[1] = 8'b0;
assign LED_exH[2] = 8'b0;
```

```

assign LED_exH[3] = 8'b0;
assign LED_exH[4] = 8'b0;
assign LED_exH[5] = 8'b0;
assign LED_exH[6] = 8'b0;
assign LED_exH[7] = 8'b0;
////////////////////////////////////
////////////////////////////////////
//dust
/*
EX ex(.clock(clock),.PC(16'b0),
      .WDaddress(dipSW_B[2:0]),.control({~pushSW_A[0],~pushSW_A[1],4'b1
001}),
      .ALUcontrol(dipSW_A[3:0]),//.immediat({8'b0,dipSW_B}),
      .Rs_Ra({rotSW_A,12'b0000_0000_0000}),.Rd_Rb({rotSW_B,12'b0000_0
000_0000}),
      .IMnextPC_(pc),.branch_(LED[7]),//.WBaddress_(),.control_(cont),
      .result_(result),//.Rb_(),.ConditionCode_(cd)
      );

wire [15:0] pc;
wire [15:0] result;
wire [3:0] cd;
wire [3:0] cont;
wire branch;
*/

////////////////////////////////////
//#####
##
//#####
##
//##### DON'T TOUCH
#####
//#####
##
// assign
assign vdd = 1;
counter4 x4(.clock(clock),.plus(vdd),.overflow(clock_x4));
counter8 x8(.clock(clock),.plus(vdd),.overflow(clock_x8));
counter10 x10(.clock(clock),.plus(vdd),.overflow(clock_x10));
counter10 x100(.clock(clock_x10),.plus(vdd),.overflow(clock_x100));
counter10 x1000(.clock(clock_x100),.plus(vdd),.overflow(clock_x1000));

```

```

counter2p16 x2p16(.clock(clock),.plus(vdd),.overflow(clock_x2p16));
LED4set LEDa_d(.CK(clock_x4),
               .LED3(LED_A),.LED2(LED_B),.LED1(LED_C),.LED0(
LED_D),
               .LED(LED7segA_D),.selecters(slctA_D));
LED4set LEDe_h(.CK(clock_x4),
               .LED3(LED_E),.LED2(LED_F),.LED1(LED_G),.LED0(
LED_H),
               .LED(LED7segE_H),.selecters(slctE_H));

LED8set LEDEx0(.CK(clock_x100),
               .LEDa(LED_exA[0]),.LEDb(LED_exB[0]),.LEDc(LED_
exC[0]),.LEDd(LED_exD[0]),
               .LEDe(LED_exE[0]),.LEDf(LED_exF[0]),.LEDg(LED_e
xG[0]),.LEDh(LED_exH[0]),
               .LED(LED7ep0),.selecters(slctEp));
LED8set LEDEx1(.CK(clock_x100),
               .LEDa(LED_exA[1]),.LEDb(LED_exB[1]),.LEDc(LED_
exC[1]),.LEDd(LED_exD[1]),
               .LEDe(LED_exE[1]),.LEDf(LED_exF[1]),.LEDg(LED_e
xG[1]),.LEDh(LED_exH[1]),
               .LED(LED7ep1));
LED8set LEDEx2(.CK(clock_x100),
               .LEDa(LED_exA[2]),.LEDb(LED_exB[2]),.LEDc(LED_
exC[2]),.LEDd(LED_exD[2]),
               .LEDe(LED_exE[2]),.LEDf(LED_exF[2]),.LEDg(LED_e
xG[2]),.LEDh(LED_exH[2]),
               .LED(LED7ep2));
LED8set LEDEx3(.CK(clock_x100),
               .LEDa(LED_exA[3]),.LEDb(LED_exB[3]),.LEDc(LED_
exC[3]),.LEDd(LED_exD[3]),
               .LEDe(LED_exE[3]),.LEDf(LED_exF[3]),.LEDg(LED_e
xG[3]),.LEDh(LED_exH[3]),
               .LED(LED7ep3));
LED8set LEDEx4(.CK(clock_x100),
               .LEDa(LED_exA[4]),.LEDb(LED_exB[4]),.LEDc(LED_
exC[4]),.LEDd(LED_exD[4]),
               .LEDe(LED_exE[4]),.LEDf(LED_exF[4]),.LEDg(LED_e
xG[4]),.LEDh(LED_exH[4]),
               .LED(LED7ep4));
LED8set LEDEx5(.CK(clock_x100),
               .LEDa(LED_exA[5]),.LEDb(LED_exB[5]),.LEDc(LED_

```

