**信息科学与工程学院**

**2020－2021学年第一学期**

实 验 报 告

课程名称： 电子设计自动化

实验名称： 硬件实验四

专 业 班 级 通信工程 二班

学 生 学 号 201800121050

学 生 姓 名 孟麟芝

实 验 时 间 2020年11月1日

实验报告

## 【实验目的】

本次实验为硬件实验考试。

## 【实验仪器与器材】

1.EDA开发软件一套

2.微机一台

3.实验开发系统一台

4.打印机一台

## 【实验要求】

设计一个二位的BCD码加一计时器，从十进制数00起递增到29止，显示用数码管HEX3和HEX2，当数字递增到29时，输出一个闪烁三次的绿色指示灯LEDG7，再输出一个常亮的红色指示灯LEDR3。

若有输入置位、复位、使能端等，使用拨动开关SW15，SW14，SW13。

层次设计用VHDL例化语言实现，不用图形输入方式。

## 【实验源代码】

1.顶层设计源代码：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY menglinzhi\_1050 IS

    PORT (

        CLK : IN STD\_LOGIC;

        EN : IN STD\_LOGIC := '0';

        RST : IN STD\_LOGIC := '0';

        SG\_OUT : OUT STD\_LOGIC\_VECTOR(15 DOWNTO 0);

        DATA\_OUT : BUFFER STD\_LOGIC\_VECTOR(7 DOWNTO 0);

        GREEN : BUFFER STD\_LOGIC := '0';

        RED : OUT STD\_LOGIC := '0'

    );

END menglinzhi\_1050;

ARCHITECTURE BHV OF menglinzhi\_1050 IS

    SIGNAL CLKIN : STD\_LOGIC;

    COMPONENT DECODER7 IS

        PORT (

            BIN\_IN : IN STD\_LOGIC\_VECTOR(3 DOWNTO 0);

            SG\_OUT : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0));

    END COMPONENT DECODER7;

    COMPONENT FDIVIDER IS

        PORT (

            CLK : IN STD\_LOGIC; --50MHz

            CP2 : BUFFER STD\_LOGIC := '0';

            CP1 : BUFFER STD\_LOGIC := '0');

    END COMPONENT FDIVIDER;

BEGIN

    PROCESS (CLKIN, RST, EN)

        VARIABLE Q : STD\_LOGIC\_VECTOR(7 DOWNTO 0) := "00000000";

        VARIABLE TWINK : INTEGER RANGE 0 TO 5;

    BEGIN

        IF EN = '1' THEN

            IF RST = '1' THEN

                Q := "00000000";

GREEN <= ‘0’;

RED <= ‘0’;

            ELSE

                IF CLKIN'EVENT AND CLKIN = '1' THEN

                    IF Q = X"29" THEN

                        IF TWINK <= 5 THEN

                            GREEN <= NOT GREEN;

                            TWINK := TWINK + 1;

                        ELSE

                            RED <= '1';

                        END IF;

                    ELSE

                        IF Q(3 DOWNTO 0) < 9 THEN

                            Q(3 DOWNTO 0) := Q(3 DOWNTO 0) + 1;

                        ELSE

                            IF Q(7 DOWNTO 4) < 2 THEN

                                Q(7 DOWNTO 4) := Q(7 DOWNTO 4) + 1;

                            END IF;

                            Q(3 DOWNTO 0) := (OTHERS => '0');

                        END IF;

                    END IF;

                END IF;

            END IF;

        END IF;

        DATA\_OUT <= Q;

    END PROCESS;

    GETCLK : FDIVIDER PORT MAP(CLK => CLK, CP2 => CLKIN);

    DECODER\_HIGH : DECODER7 PORT MAP(BIN\_IN => DATA\_OUT(7 DOWNTO 4), SG\_OUT => SG\_OUT(15 DOWNTO 8));

    DECODER\_LOW : DECODER7 PORT MAP(BIN\_IN => DATA\_OUT(3 DOWNTO 0), SG\_OUT => SG\_OUT(7 DOWNTO 0));

END BHV;

2.分频器FDIVIDER源代码：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY FDIVIDER IS

    PORT (

        CLK : IN STD\_LOGIC; --50MHz

        CP1 : BUFFER STD\_LOGIC := '0'; --1Hz

        CP2 : BUFFER STD\_LOGIC := '0');--2Hz

END FDIVIDER;

ARCHITECTURE BHV OF FDIVIDER IS

BEGIN

    PROCESS (CLK) --50MHz

        VARIABLE T0 : INTEGER RANGE 0 TO 24999999 := 0; --1Hz

        VARIABLE T1 : INTEGER RANGE 0 TO 12499999 := 0; --2Hz

    BEGIN

        IF (CLK'event AND CLK = '1') THEN

            IF (T0 = 24999999) THEN

                T0 := 0;

                CP1 <= NOT CP1;

            ELSE

                T0 := T0 + 1;

            END IF;

            IF (T1 = 12499999) THEN

                T1 := 0;

                CP2 <= NOT CP2;

            ELSE

                T1 := T1 + 1;

            END IF;

        END IF;

    END PROCESS;

END BHV;

3.七段译码器DECODER7源代码：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY DECODER7 IS

    PORT (

        BIN\_IN : IN STD\_LOGIC\_VECTOR(3 DOWNTO 0); --输入四位BCD码

        SG\_OUT : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)); --0~7分别对应共阳极七段数码管的a,b,c,d,e,f,g,db

END;

ARCHITECTURE BHV OF DECODER7 IS

BEGIN

    PROCESS (BIN\_IN)

    BEGIN

        CASE BIN\_IN IS

            WHEN "0000" => SG\_OUT <= X"C0";

            WHEN "0001" => SG\_OUT <= X"F9";

            WHEN "0010" => SG\_OUT <= X"A4";

            WHEN "0011" => SG\_OUT <= X"B0";

            WHEN "0100" => SG\_OUT <= X"99";

            WHEN "0101" => SG\_OUT <= X"92";

            WHEN "0110" => SG\_OUT <= X"82";

            WHEN "0111" => SG\_OUT <= X"F8";

            WHEN "1000" => SG\_OUT <= X"80";

            WHEN "1001" => SG\_OUT <= X"90";

            WHEN "1010" => SG\_OUT <= X"88";

            WHEN "1011" => SG\_OUT <= X"83";

            WHEN "1100" => SG\_OUT <= X"C6";

            WHEN "1101" => SG\_OUT <= X"A1";

            WHEN "1110" => SG\_OUT <= X"86";

            WHEN "1111" => SG\_OUT <= X"8E";

            WHEN OTHERS => NULL;

        END CASE;

    END PROCESS;

END;

## 【仿真波形图】

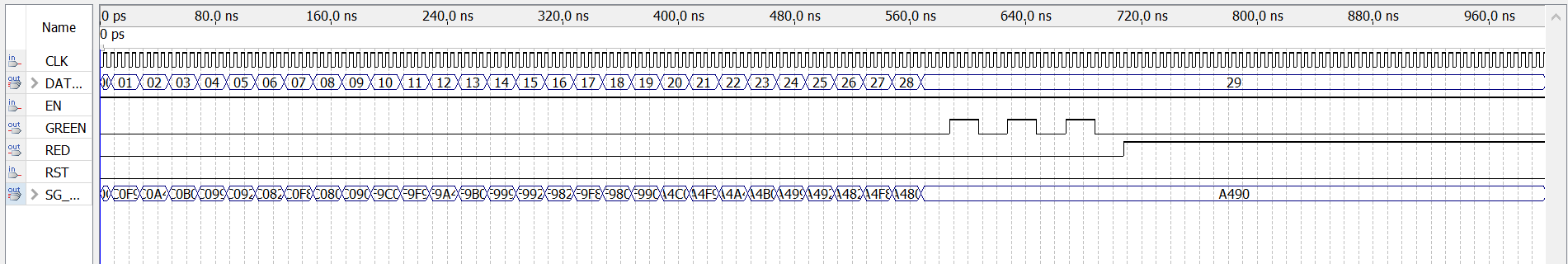
实验代码的思路如下：先对50MHz进行分频，若使能端为1则程序开始工作，若RST端为1则应置0重新开始计数，且要将输出清除。

用一个变量Q记录计数值，将其分为高四位和低四位，在第四位低于9时，每来一个时钟第四位都要加一，低四位计数到9时高四位要加1，且第四位要清零。

若总的变量Q高四位为0X2，第四位为0X9，则应停止计数，开始绿灯闪烁，绿灯闪烁是通过六次取反实现的，闪烁结束后红灯点亮。

将Q赋给输出，再将输出连接到一个译码器进行译码即可。

将分频倍率调小，进行仿真得到结果如下图：



## 【引脚分配图】

