**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： 硬件实验三

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10月24日

实验报告

## 【实验目的】

1.学习EDA软件的基本操作。

2.学习时序电路的设计方法。

3.掌握有限状态机的设计方法。

4.了解巴克码的发生与检测方法

## 【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】

1. 设计一个分频器，有1Hz或2Hz输出，要求：实验开发板上有一个50MHz的时钟脉冲（此频率过高，接到开发板的LED灯后，无法观察到LED灯一亮一灭的过程） ，设计一个分频器，使得分频后的时钟脉冲接到开发板上的LED灯后，肉眼可以观察到LED灯闪烁。

2.设计一个十进制加法计数器（第240页实验要求1），使用设计的分频器的输出信号作为计数器的时钟输入，再利用实验二中设计的七段显示译码器显示计数值。

3.巴克码发生器设计（第240-241页）和巴克码检测器设计（利用状态机来进行设计，第241-242页）。

## 【分频器实验】

## 【实验源代码】

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;

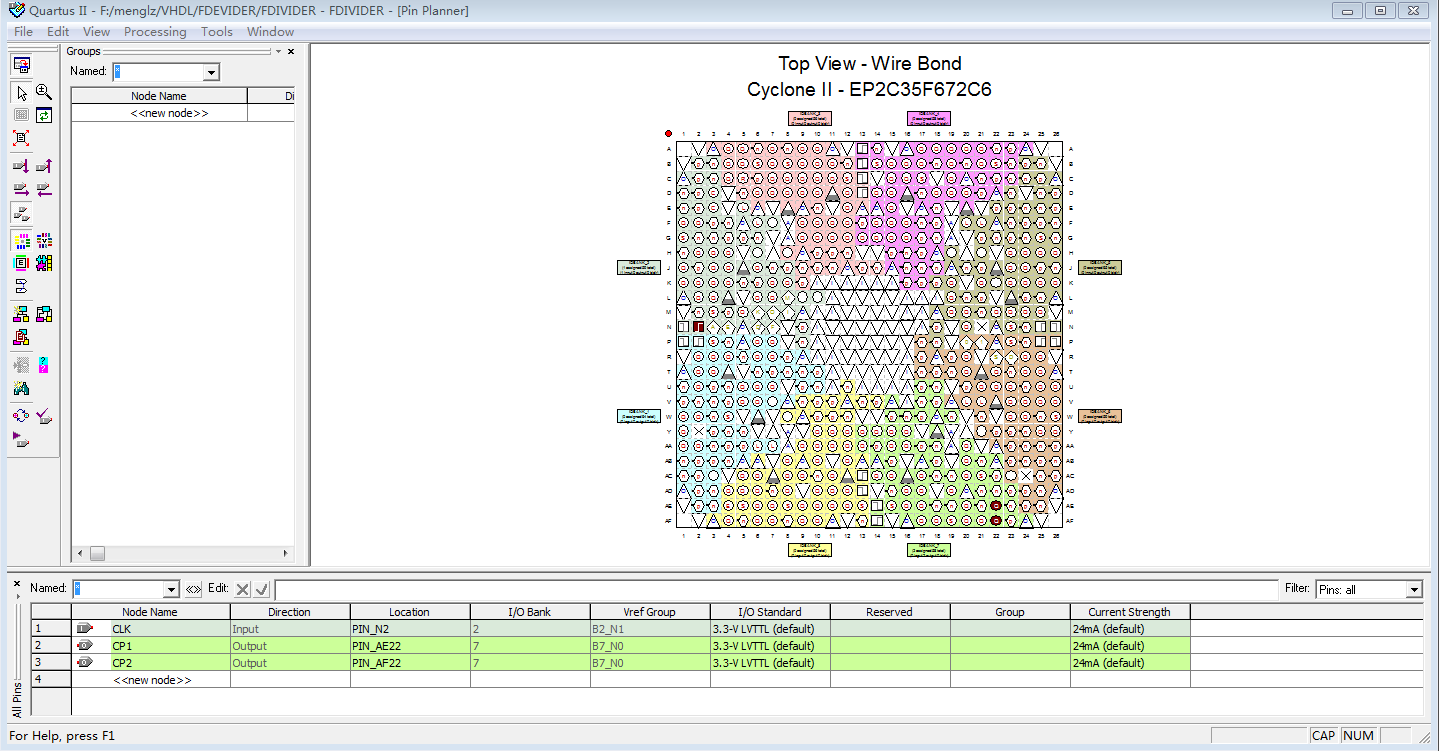
    END PROCESS;

END BHV;

【仿真波形图】

该实验由于将50MHz分为1Hz或2Hz分频倍率太高，故已经难以仿真得到结果，但设计是能够实现相应功能的，其原理是通过一个计数变量，每当他达到一定值就将输出取反，实现了分频功能。

【引脚分配图】



## 【十进制加法计数器实验】

【实验源代码】

1.顶层设计CNT10\_DECODE源代码：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY CNT10\_DECODE IS

    PORT (

        CLK, RST, EN, LOAD : IN STD\_LOGIC;

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

        DOUT : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

        COUT : OUT STD\_LOGIC;

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

    );

END CNT10\_DECODE;

ARCHITECTURE BHV OF CNT10\_DECODE IS

    COMPONENT FDIVIDER IS

        PORT (

            CLK : IN STD\_LOGIC; --50MHz

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

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

    END COMPONENT FDIVIDER;

    COMPONENT CNT10 IS

        PORT (

            CLK, RST, EN, LOAD : IN STD\_LOGIC;

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

            DOUT : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

            COUT : OUT STD\_LOGIC);

    END COMPONENT CNT10;

    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;

    SIGNAL CLKIN : STD\_LOGIC;

    SIGNAL TOSG : STD\_LOGIC\_VECTOR(7 DOWNTO 0) := "00000000";

BEGIN

    GETCLK : FDIVIDER PORT MAP(CLK => CLK, CP1 => CLKIN);

    COUNTER : CNT10 PORT MAP(CLK => CLKIN, RST => RST, EN => EN, LOAD => LOAD, DATA => DATA, DOUT => TOSG(3 DOWNTO 0), COUT => TOSG(4));

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

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

    DOUT <= TOSG(3 DOWNTO 0);

    COUT <= TOSG(4);

END BHV;

2.计数器CNT10源代码：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY CNT10 IS

    PORT (

        CLK, RST, EN, LOAD : IN STD\_LOGIC;

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

        DOUT : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

        COUT : OUT STD\_LOGIC);

END CNT10;

ARCHITECTURE BHV OF CNT10 IS

BEGIN

    PROCESS (CLK, RST, EN, LOAD) --50MHz

        VARIABLE Q : STD\_LOGIC\_VECTOR(3 DOWNTO 0);

    BEGIN

        IF RST = '0' THEN

            Q := (OTHERS => '0');

        ELSIF CLK'EVENT AND CLK = '1' THEN

            IF EN = '1' THEN

                IF (LOAD = '0') THEN

                    Q := DATA;

                ELSE

                    IF Q < 9 THEN

                        Q := Q + 1;

                    ELSE

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

                    END IF;

                END IF;

            END IF;

        END IF;

        IF Q = "1001" THEN

            COUT <= '1';

        ELSE

            COUT <= '0';

        END IF;

        DOUT <= Q;

    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;

3.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;

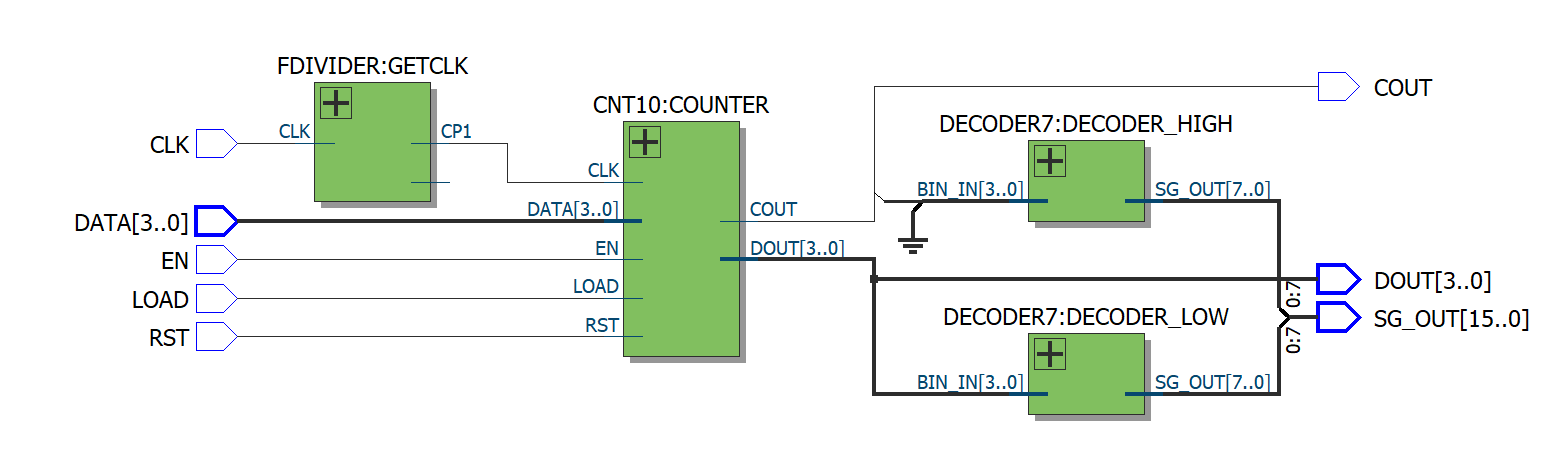
        END IF;

    END PROCESS;

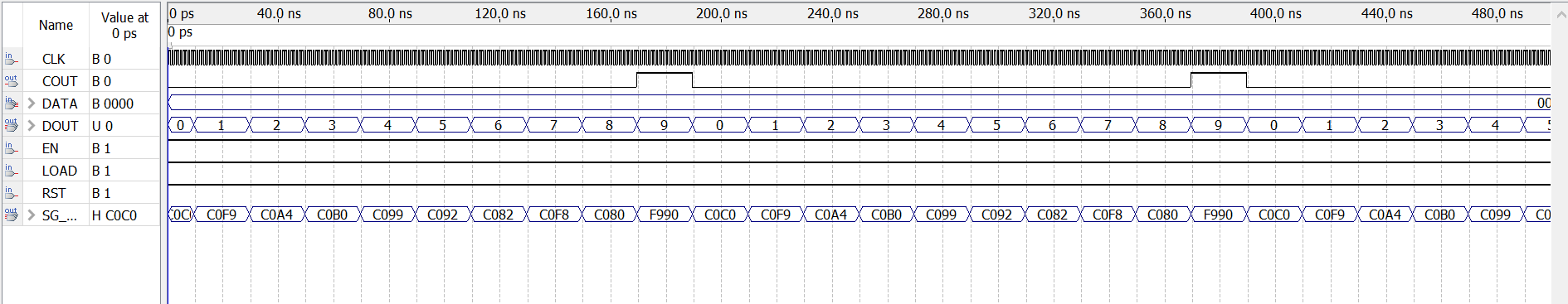
END BHV;

【仿真波形图】

本实验通过层次化设计方法设计了一个十进制加法计数器，从0开始计数，每到9则归0，且输出进位为1，带有清零端、使能端、置数端，其RTL电路图如下，可见其层次十分明确：

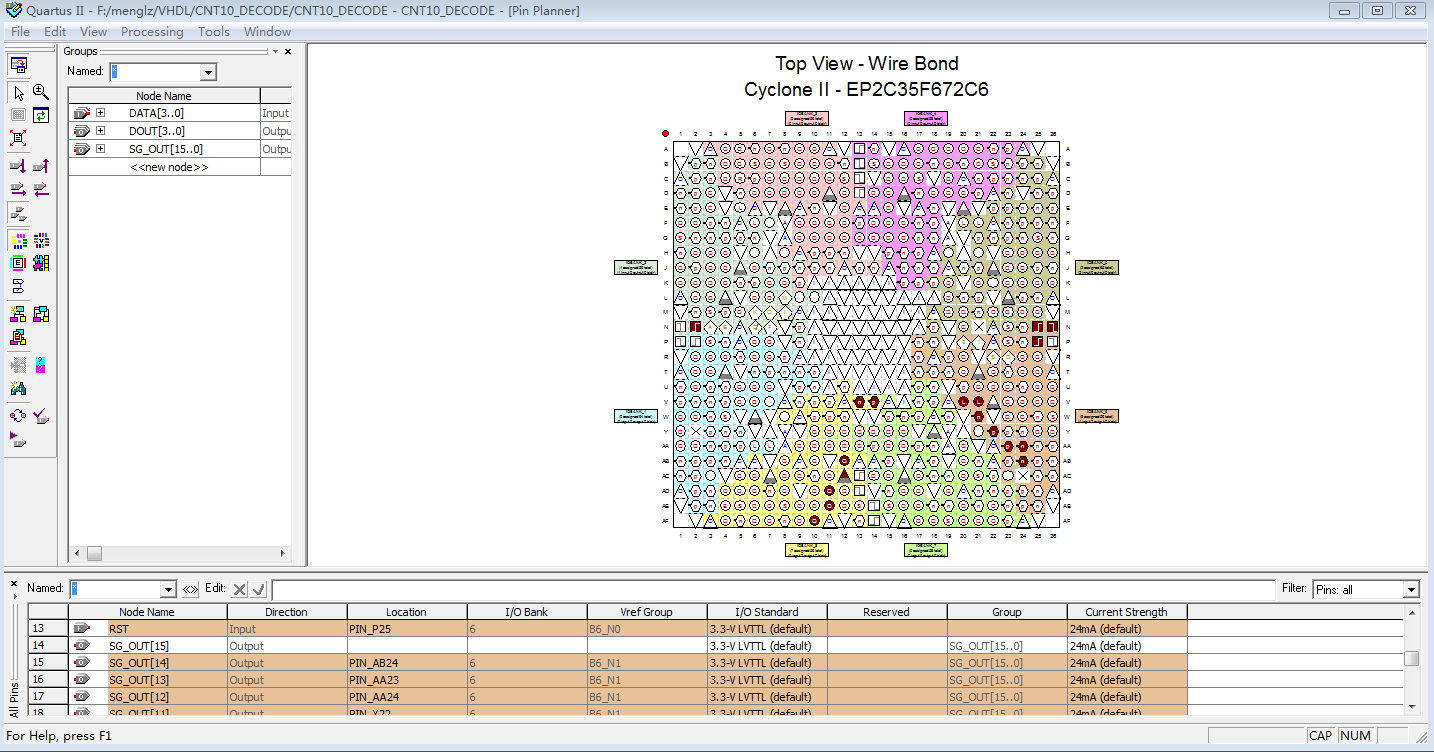


由于分频倍率过高，进行仿真得到如下结果，可见程序实现了功能



【引脚分配图】

需将输入的端的CLK连接50MHz的CLK发生源，其他均连接到拨动开关，译码输出SG\_OUT连接到两个数码管，COUT可以用一个发光二极管显示出来。



## 【巴克码发生与检测器实验】

## 【实验源代码】

1.巴克码发生器源代码：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY GENBUCKCODE IS

    PORT (

        CLK : IN STD\_LOGIC;

        BUCKOUT : OUT STD\_LOGIC;

        BEATOUT : BUFFER STD\_LOGIC := '0'

    );

END GENBUCKCODE;

ARCHITECTURE BHV OF GENBUCKCODE IS

BEGIN

    PROCESS (CLK)

        VARIABLE BUCK : STD\_LOGIC\_VECTOR(7 DOWNTO 0) := "01110010";

    BEGIN

        IF CLK'EVENT AND CLK = '1' THEN

            BUCKOUT <= BUCK(6);

            BUCK(7 DOWNTO 1) := BUCK(6 DOWNTO 0);

            BUCK(0) := BUCK(7);

        END IF;

    END PROCESS;

    BEATOUT <= CLK;

END BHV;

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY GENBUCKCODE IS

    PORT (

        CLK : IN STD\_LOGIC;

        BUCKOUT : OUT STD\_LOGIC;

        BEATOUT : BUFFER STD\_LOGIC := '0'

    );

END GENBUCKCODE;

ARCHITECTURE BHV OF GENBUCKCODE IS

BEGIN

    PROCESS (CLK)

        VARIABLE BUCK : STD\_LOGIC\_VECTOR(7 DOWNTO 0) := "01110010";

    BEGIN

        IF CLK'EVENT AND CLK = '1' THEN

            BUCKOUT <= BUCK(6);

            BUCK(7 DOWNTO 1) := BUCK(6 DOWNTO 0);

            BUCK(0) := BUCK(7);

        END IF;

    END PROCESS;

    BEATOUT <= CLK;

END BHV;

2.巴克码检测器源代码：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY DETECTBUCKCODE IS

    PORT (

        CLK : IN STD\_LOGIC;

        BUCKIN : IN STD\_LOGIC;

        RESULT : OUT STD\_LOGIC

    );

END DETECTBUCKCODE;

ARCHITECTURE BHV OF DETECTBUCKCODE IS

    TYPE STATES IS(ST0, ST1, ST2, ST3, ST4, ST5, ST6);

    SIGNAL CURRENT\_STATE, NEXT\_STATE:STATES;

BEGIN

    PROCESS (CLK)

    BEGIN

    IF CLK'EVENT AND CLK='1' THEN

        CURRENT\_STATE<=NEXT\_STATE;

        IF CURRENT\_STATE=ST6 AND BUCKIN='0' THEN

        RESULT<='1';

        ELSE

        RESULT<='0';

        END IF;

     END IF;

    END PROCESS;

    PROCESS (BUCKIN)

    BEGIN

        CASE CURRENT\_STATE IS

            WHEN ST0 => IF BUCKIN = '1' THEN NEXT\_STATE <= ST1; ELSE NEXT\_STATE <= ST0; END IF;

            WHEN ST1 => IF BUCKIN = '1' THEN NEXT\_STATE <= ST2; ELSE NEXT\_STATE <= ST0; END IF;

            WHEN ST2 => IF BUCKIN = '1' THEN NEXT\_STATE <= ST3; ELSE NEXT\_STATE <= ST0; END IF;

            WHEN ST3 => IF BUCKIN = '0' THEN NEXT\_STATE <= ST4; ELSE NEXT\_STATE <= ST0; END IF;

            WHEN ST4 => IF BUCKIN = '0' THEN NEXT\_STATE <= ST5; ELSE NEXT\_STATE <= ST0; END IF;

            WHEN ST5 => IF BUCKIN = '1' THEN NEXT\_STATE <= ST6; ELSE NEXT\_STATE <= ST0; END IF;

            WHEN ST6 => IF BUCKIN = '0' THEN NEXT\_STATE <= ST0; ELSE NEXT\_STATE <= ST0; END IF;

                WHEN OTHERS => NEXT\_STATE <= ST0;

        END CASE;

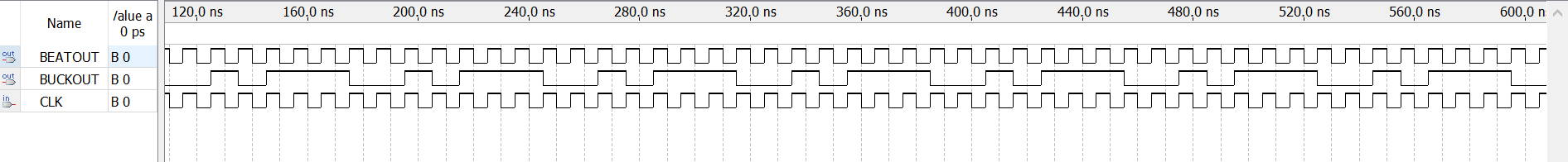
END PROCESS;

END BHV;

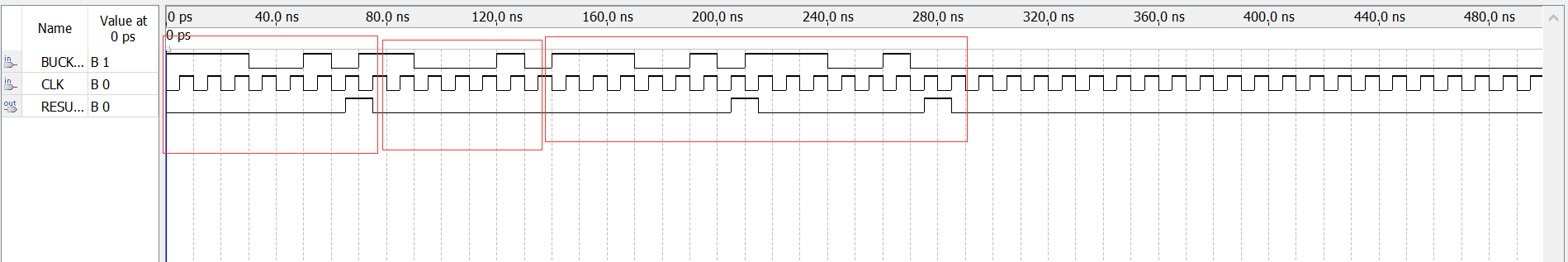
【仿真波形图】

巴克码发生器在数据通信、雷达和遥控领域有相当广泛的应用。它能自动产生周期性的序列码。本实验要求产生的序列码信号为”1110010”，可以用寄存器或同步时序电路实现。为了能够通过实验开发系统验证实验结果，可以使用两个输出端，其中一个输出端同时输出巴克码，另一个输出端输出节拍。

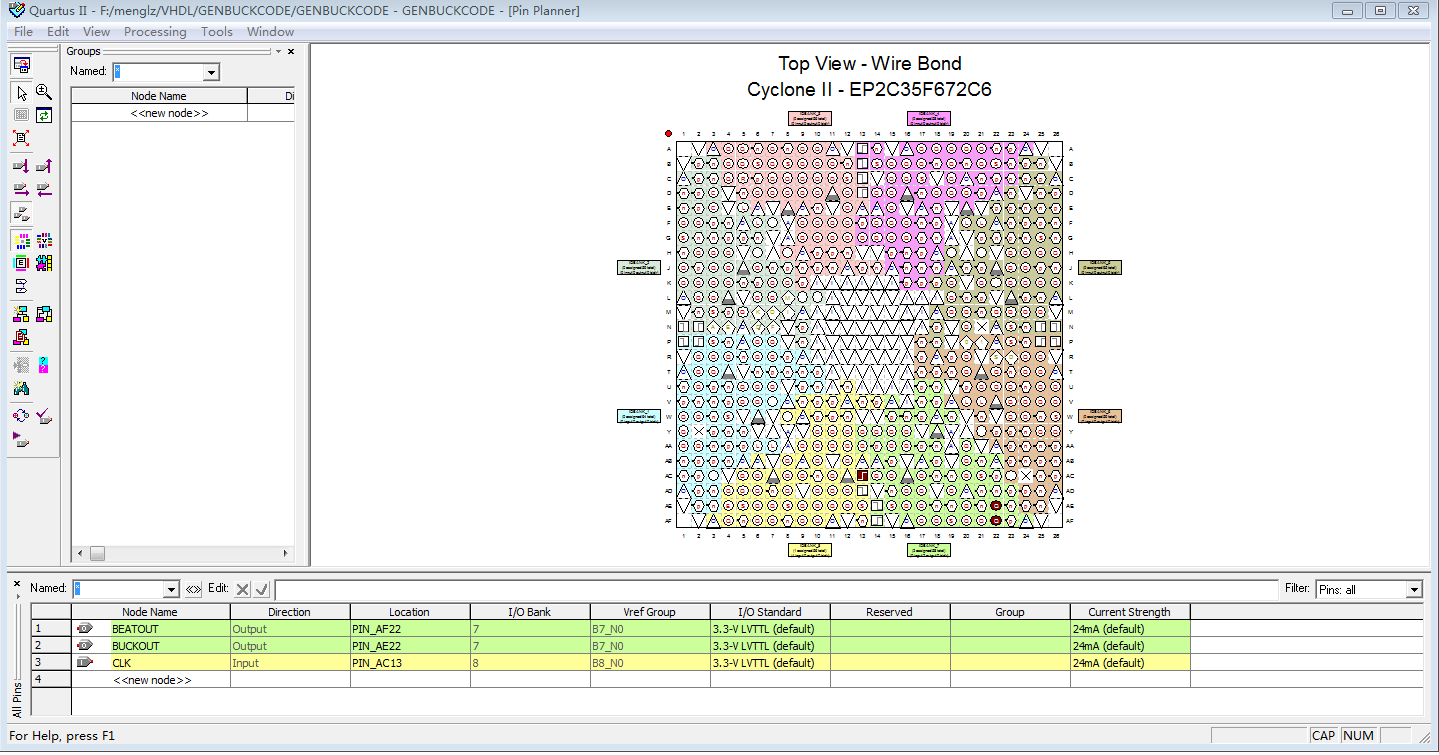
对巴克码发生器进行测试，CLK为输入，BEATOUT为节拍输出，BUCKOUT为巴克码输出，可见输出了周期性的序列。



对巴克码检测器进行测试，可见如下结果，输入序列非巴克码时，并没有结果输出，输入巴克码序列，则会有正确的检测结果’1’输出



【巴克码发生器引脚分配图】



【巴克码检测器引脚分配图】

