

# 目 录

一、实验要求.....	2
二、参考内容.....	3
1.与门逻辑的 VHDL 的源文件.....	3
2.8421BCD 转换成七段码译码器 VHDL 的源文件 .....	3
3. 带一个清零端，一个进位输出端十进制器计数器的 VHDL 源文件 .....	4
4. 50M 分频器的 VHDL 源文件.....	5
5. 由 VHDL 文件生成逻辑原理框图方法.....	6
三、DE0 外接引脚说明.....	14

## 一、实验要求

**要求 1：**学习并掌握硬件描述语言 VHDL；熟悉门电路的逻辑功能，并用硬件描述语言实现门电路的设计。参考“参考内容 1”中给出的与门源程序，编写一个异或门逻辑电路。1) 用 QuartusII 波形仿真验证；2) 下载到 DE0 开发板验证。

**要求 2：**熟悉中规模器件译码器的逻辑功能，用硬件描述语言实现其设计。参考“参考内容 2”中给出的将 8421BCD 码转换成 0-9 的七段码译码器源程序，编写一个将二进制码转换成 0-E 的七段码译码器。1) 用 QuartusII 波形仿真验证；2) 下载到 DE0 开发板，利用开发板上的数码管验证。

**要求 3：**熟悉时序电路计数器的逻辑功能，用硬件描述语言实现其设计。参考“参考内容 3”中给出的四位二进制计数器的源程序，编写一个计数器实现 0-E 计数。用 QuartusII 波形仿真验证；

**要求 4：**熟悉分频电路的逻辑功能，并用硬件描述语言实现其设计。参考“参考内容 4”中给出的 50M 分频器的源程序，编写一个能实现占空比 50% 的 5M 和 50M 分频器即两个输出，输出信号频率分别为 10Hz 和 1Hz。下载到 DE0 开发板验证。（提示：利用 DE0 板上已有的 50M 晶振作为输入信号，通过开发板上两个的 LED 灯观察输出信号）。电路框图如下：



**要求 5：**利用已经实现的 VHDL 模块文件，顶层文件采用原理图设计方法，实现 a,b,c,d,E,学号 5 位，E, d, c, b, a 计数自动循环显示，频率 1Hz 和 10Hz 可以切换。（提示：如何将 VHDL 模块文件在顶层原理图文件中引用，参考参考内容 5）

## 二、参考内容

### 1.与门逻辑的 VHDL 的源文件

```
LIBRARY IEEE;

USE IEEE.STD_LOGIC_1164.ALL;

ENTITY exa4_1 IS

PORT(A,B:IN STD_LOGIC;

C:OUT STD_LOGIC);

END exa4_1;

ARCHITECTURE fwm OF exa4_1 IS

BEGIN

C<=A AND B;

END;
```

### 2.8421BCD 转换成七段码译码器 VHDL 的源文件

```
LIBRARY IEEE;

USE IEEE.STD_LOGIC_1164.ALL;

ENTITY exa4_2 IS

PORT (data_in:IN STD_LOGIC_VECTOR(3 DOWNT0 0);

dis_out:OUT STD_LOGIC_VECTOR(6 DOWNT0 0));

END exa4_2;

ARCHITECTURE fwm OF exa4_2 IS

BEGIN

PROCESS(data_in)

BEGIN

CASE data_in IS
```

```

    WHEN "0000" => dis_out <= "1000000" ;--显示 0
    WHEN "0001" => dis_out <= "1111001" ;--显示 1
    WHEN "0010" => dis_out <= "0100100" ;--显示 2
    WHEN "0011" => dis_out <= "0110000" ;--显示 3
    WHEN "0100" => dis_out <= "0011001" ;--显示 4
    WHEN "0101" => dis_out <= "0010010" ;--显示 5
    WHEN "0110" => dis_out <= "0000010" ;--显示 6
    WHEN "0111" => dis_out <= "1111000" ;--显示 7
    WHEN "1000" => dis_out <= "0000000" ;--显示 8
    WHEN "1001" => dis_out <= "0010000" ;--显示 9

    WHEN OTHERS => dis_out <= "1111111" ;--灭灯，不显示

END CASE;

END PROCESS;

END fwm;

```

### 3. 带一个清零端，一个进位输出端十进制器计数器的 VHDL 源文件

```

LIBRARY IEEE;

USE IEEE.STD_LOGIC_1164.ALL;

USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY exa4_3 IS

    PORT ( clk,RST  : IN STD_LOGIC;

          DOUT : OUT STD_LOGIC_VECTOR (3 DOWNTO 0); -- 四位计数

          COUT : OUT STD_LOGIC);                    -- 进位位

END exa4_3;

ARCHITECTURE fwm OF exa4_3 IS

    SIGNAL Q1 : STD_LOGIC_VECTOR (3 DOWNTO 0);

```

```

BEGIN

    PROCESS(clk,RST)

    BEGIN

        IF RST = '0' THEN Q1<=(OTHERS => '0'); COUT<= '0';

        ELSIF  clk'  EVENT AND clk=' 1'  THEN

            Q1<=Q1+1;

            COUT<= '0';

            IF Q1 >= "1001" THEN Q1<=(OTHERS => '0'); COUT<= '1';

            END IF;

        END IF;

    END PROCESS;

    DOUT<=Q1 ;

END fwm;

```

#### 4. 50M 分频器的 VHDL 源文件

```

LIBRARY IEEE;

USE IEEE.STD_LOGIC_1164.ALL;

ENTITY exa4_4 IS

    PORT(clk:IN STD_LOGIC;

        clk_out:OUT STD_LOGIC);

END exa4_4;

ARCHITECTURE fwm OF exa4_4 IS

    CONSTANT m : INTEGER:= 25000000;    --50M 分频到 1Hz 时=25000000。

    SIGNAL tmp :STD_LOGIC;

BEGIN

    PROCESS(clk, tmp)

```

```

VARIABLE cout : INTEGER:=0;

BEGIN

    IF clk'EVENT AND clk='1' THEN

        cout:=cout+1;          --计数器+1

        IF cout<=m THEN tmp<='0';    --计数小于等于 25000000 , 输出 0

        ELSIF cout<m*2 THEN tmp<='1';  --计数小于 50000000 , 输出 1

        ELSE cout:=0;              --计数器清零

        END IF;

    END IF;

END PROCESS;

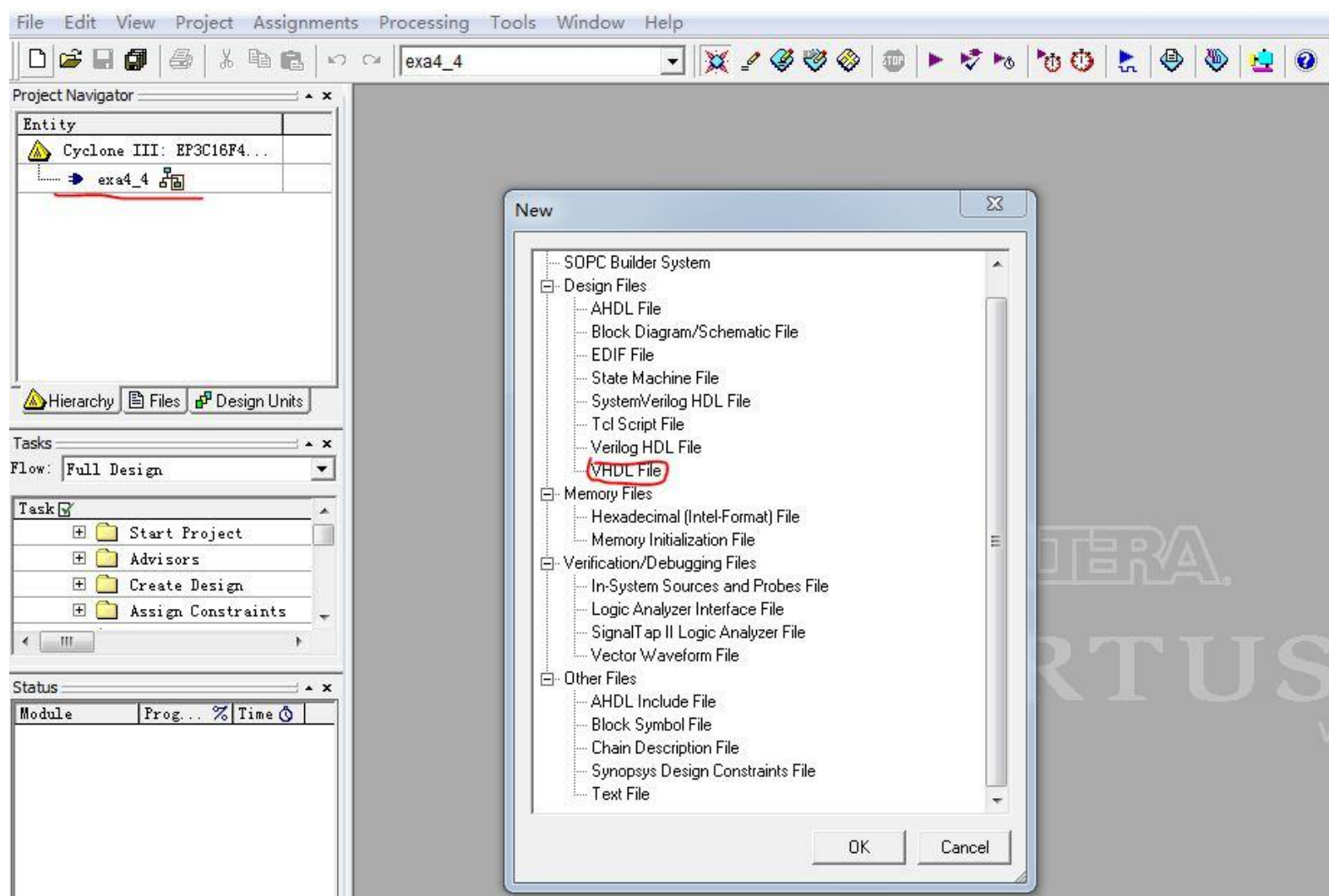
clk_out<=tmp;                  --分频器输出

END fwm;

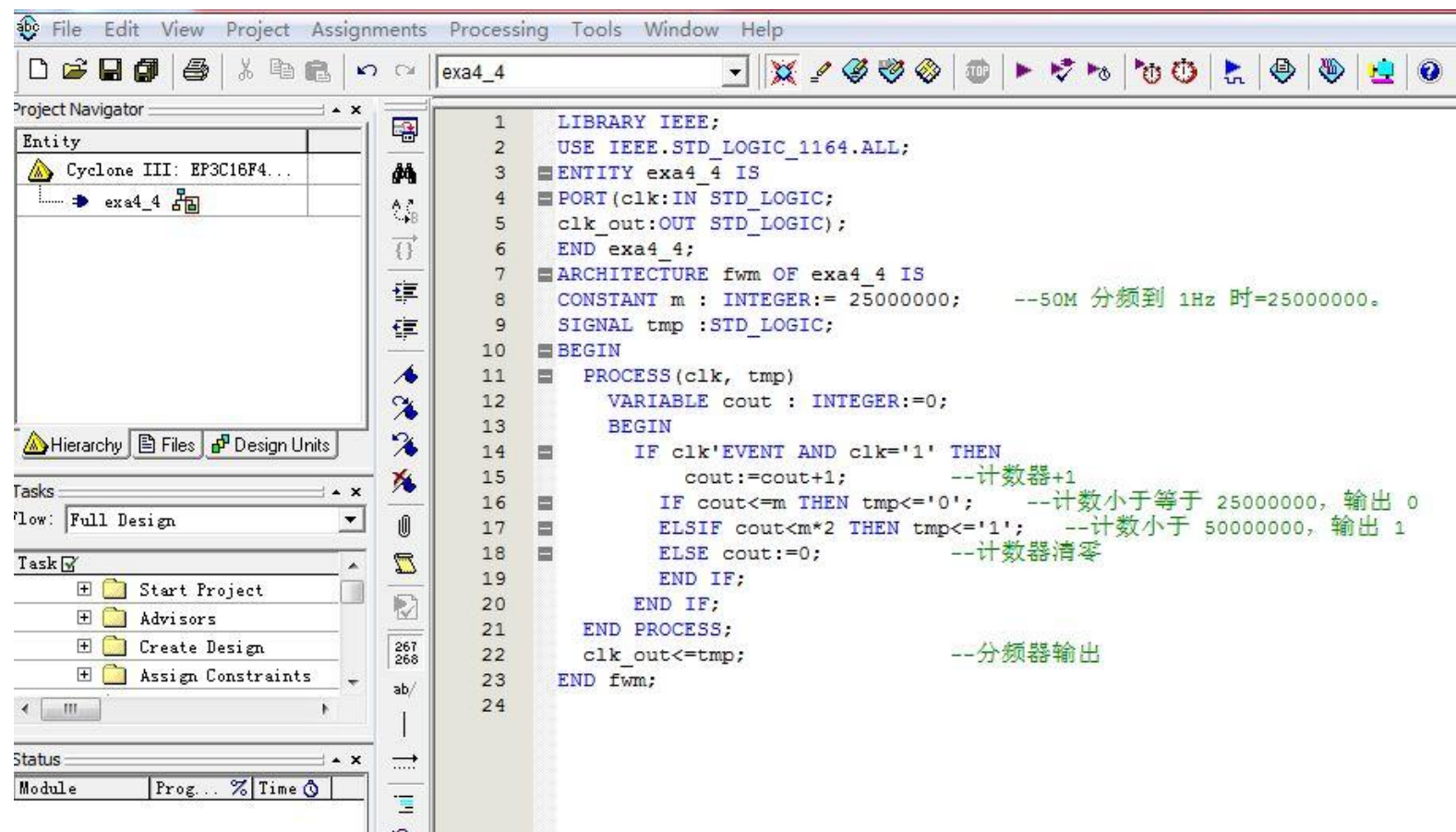
```

## 5. 由 VHDL 文件生成逻辑原理框图方法

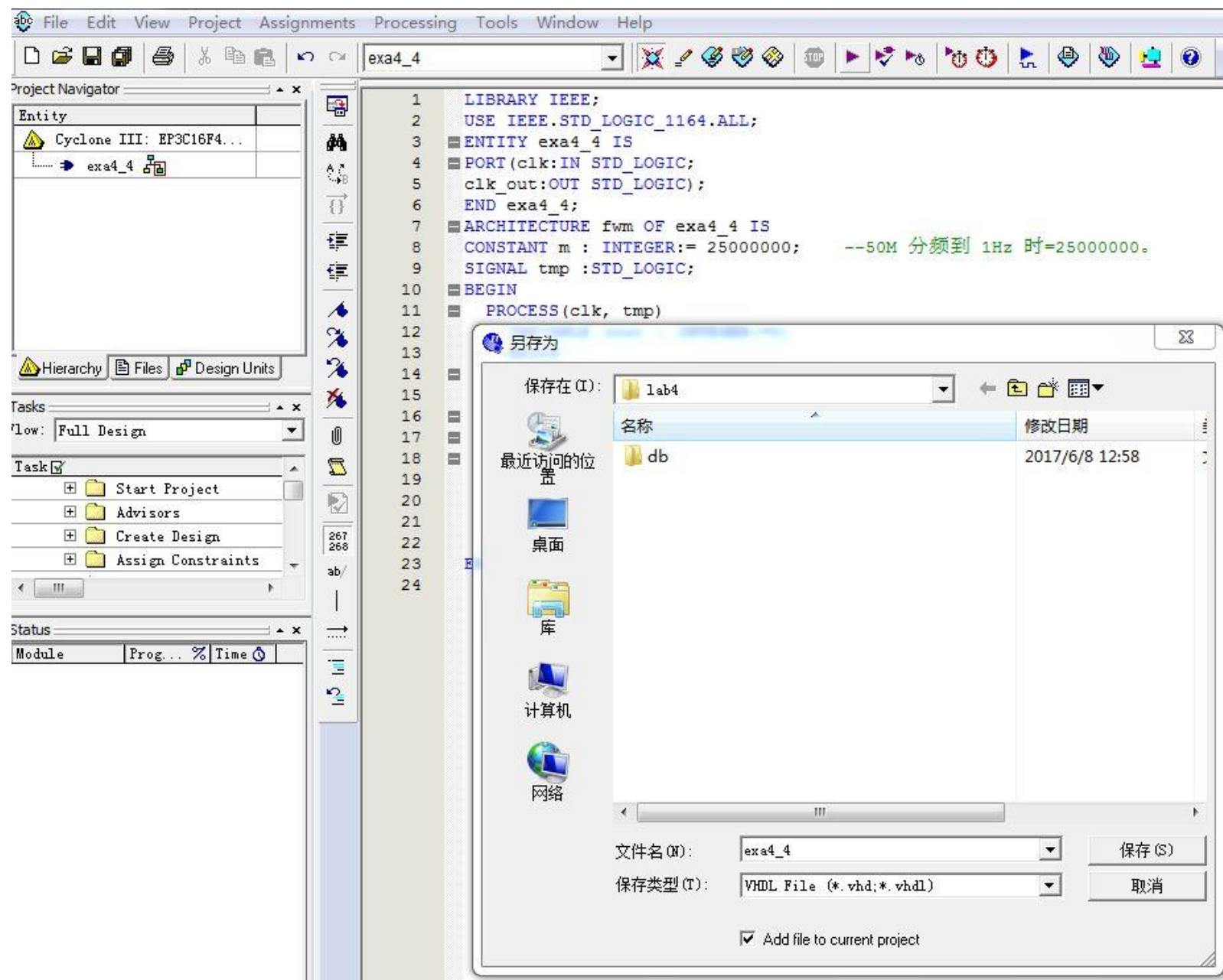
### 第一步：新建 VHDL 文件



## 第二步：输入 VHDL 文件

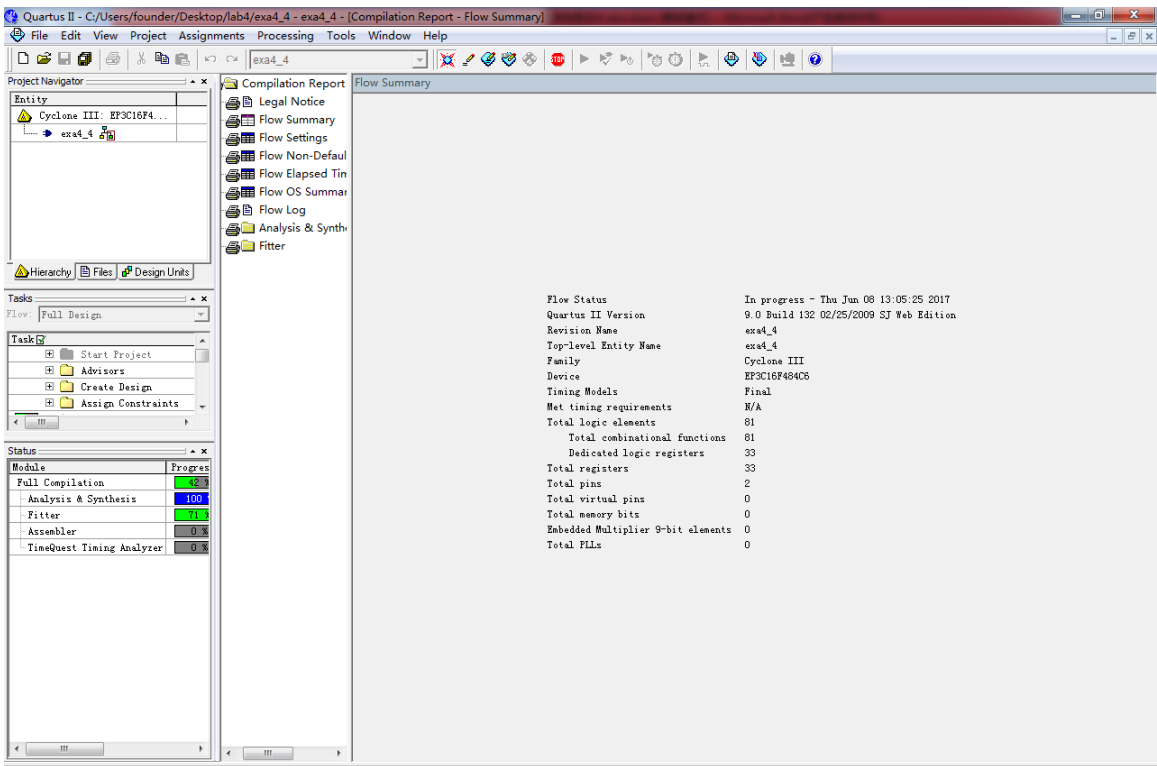


## 第三步：保存文件



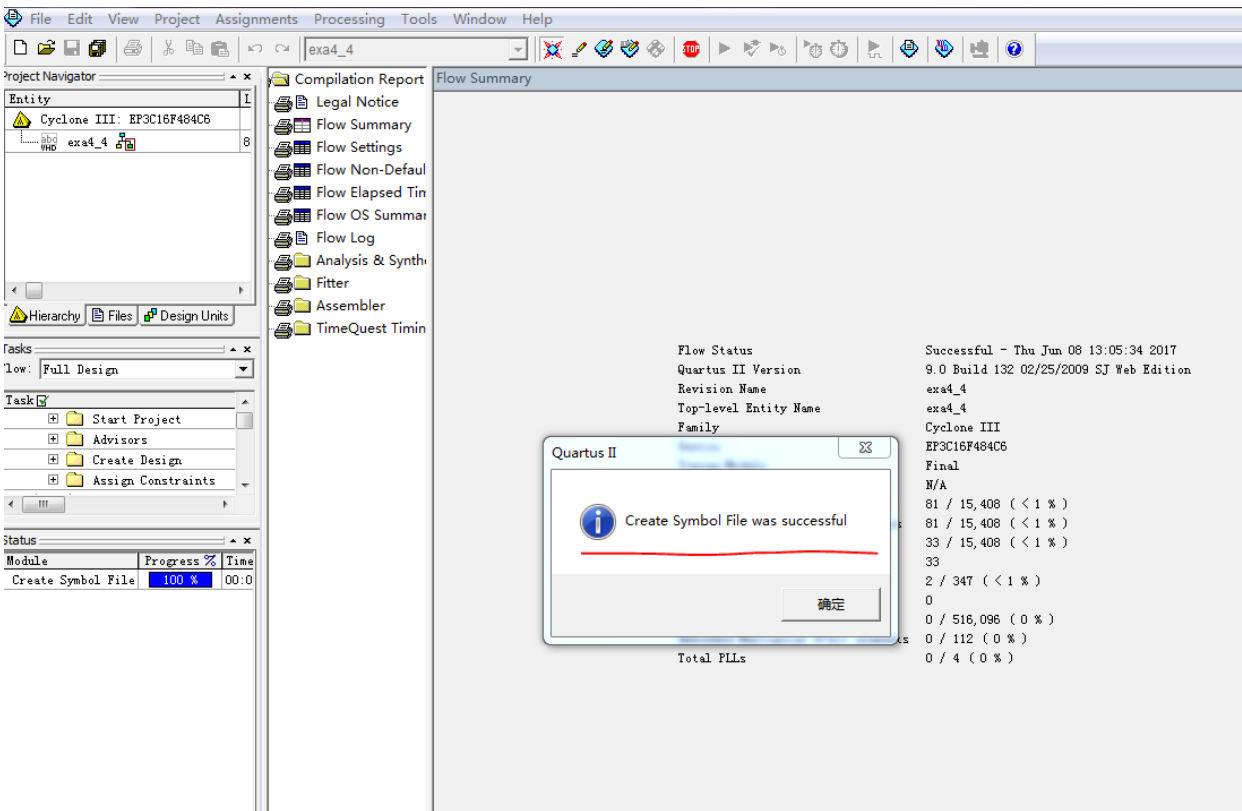
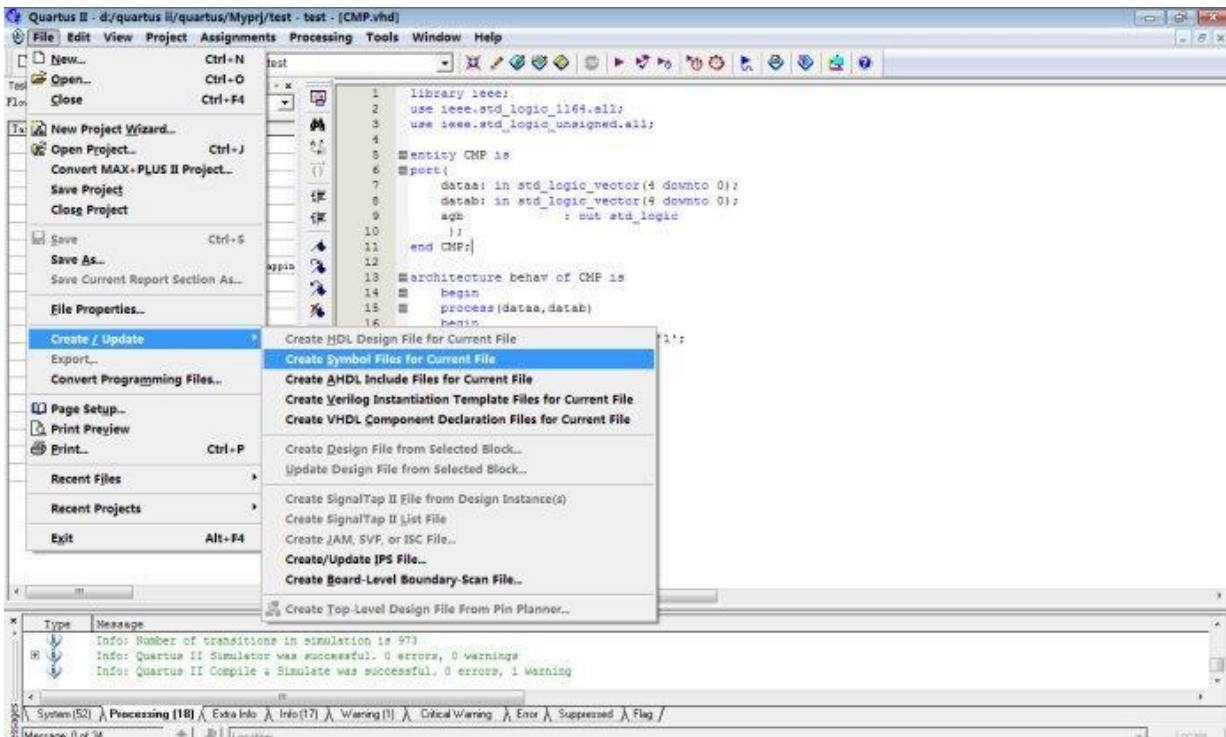
## 第四步：编译





## 第五步：生成原理图框图

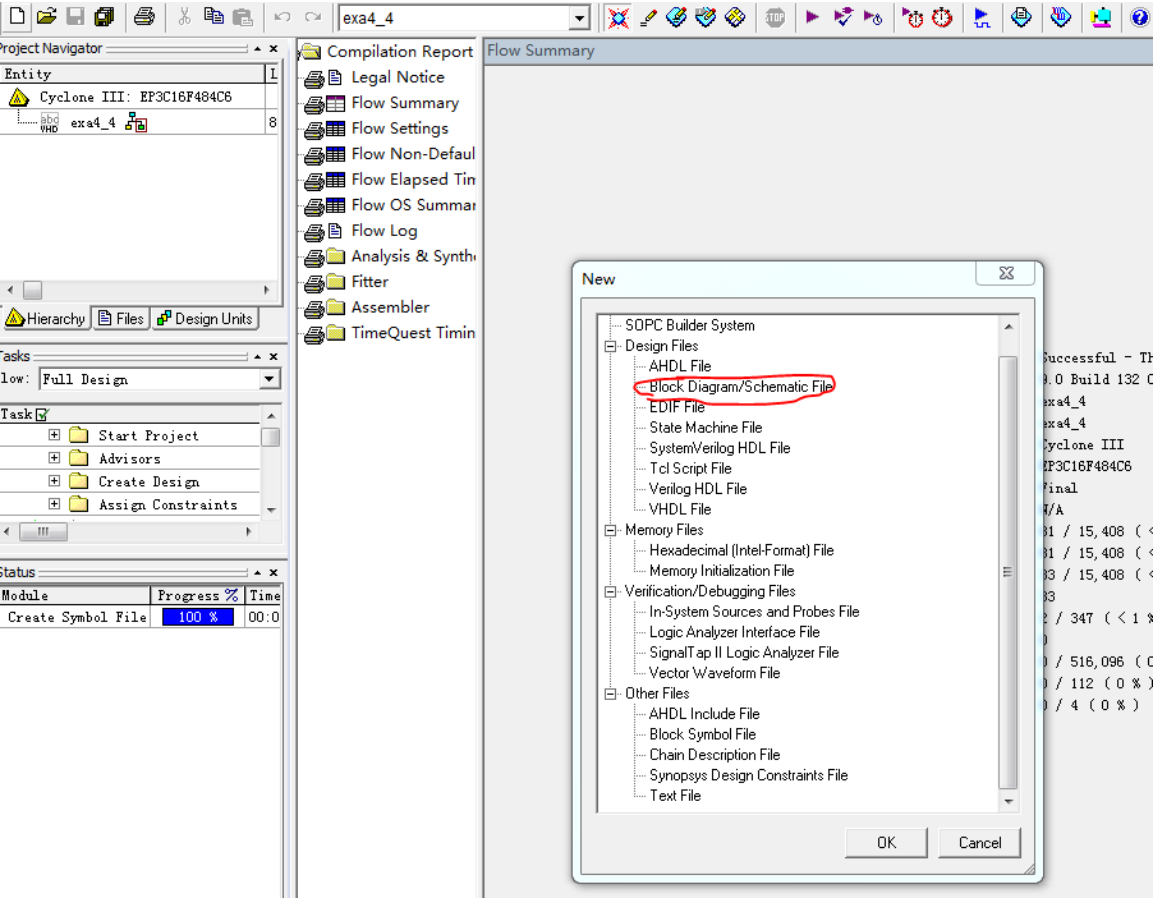
打开菜单栏，依次选择 File-> Create/Update -> Create Symbol Files for Current File。



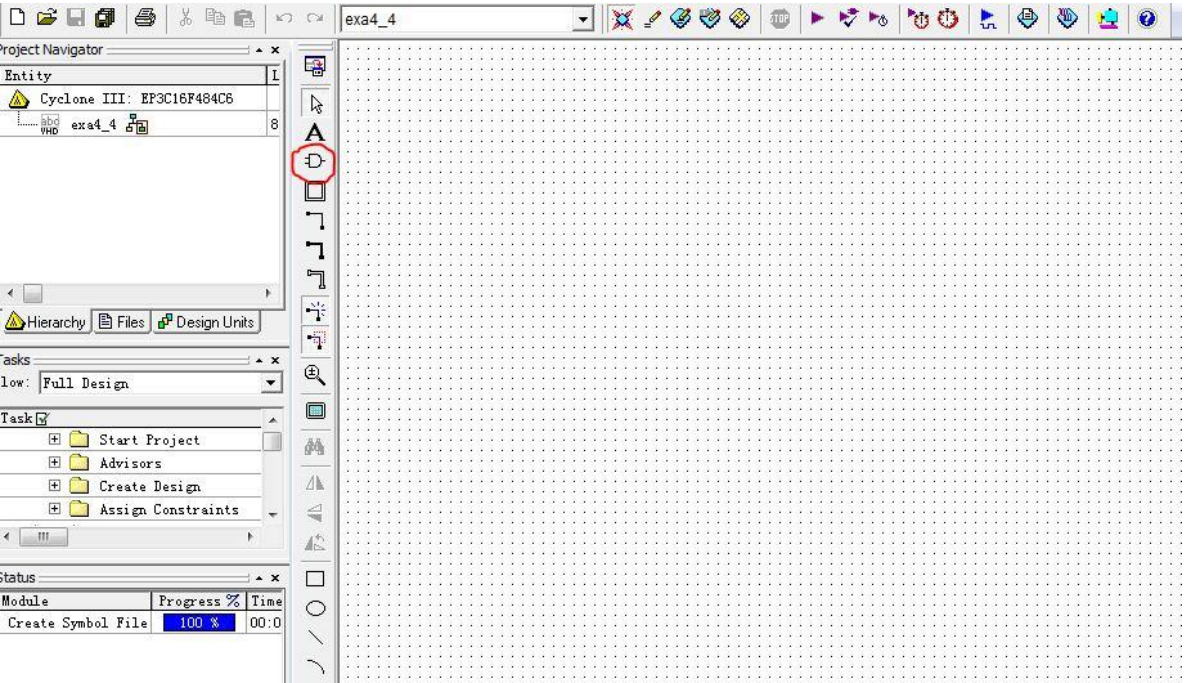


第六步：调用已经生成的原理框图

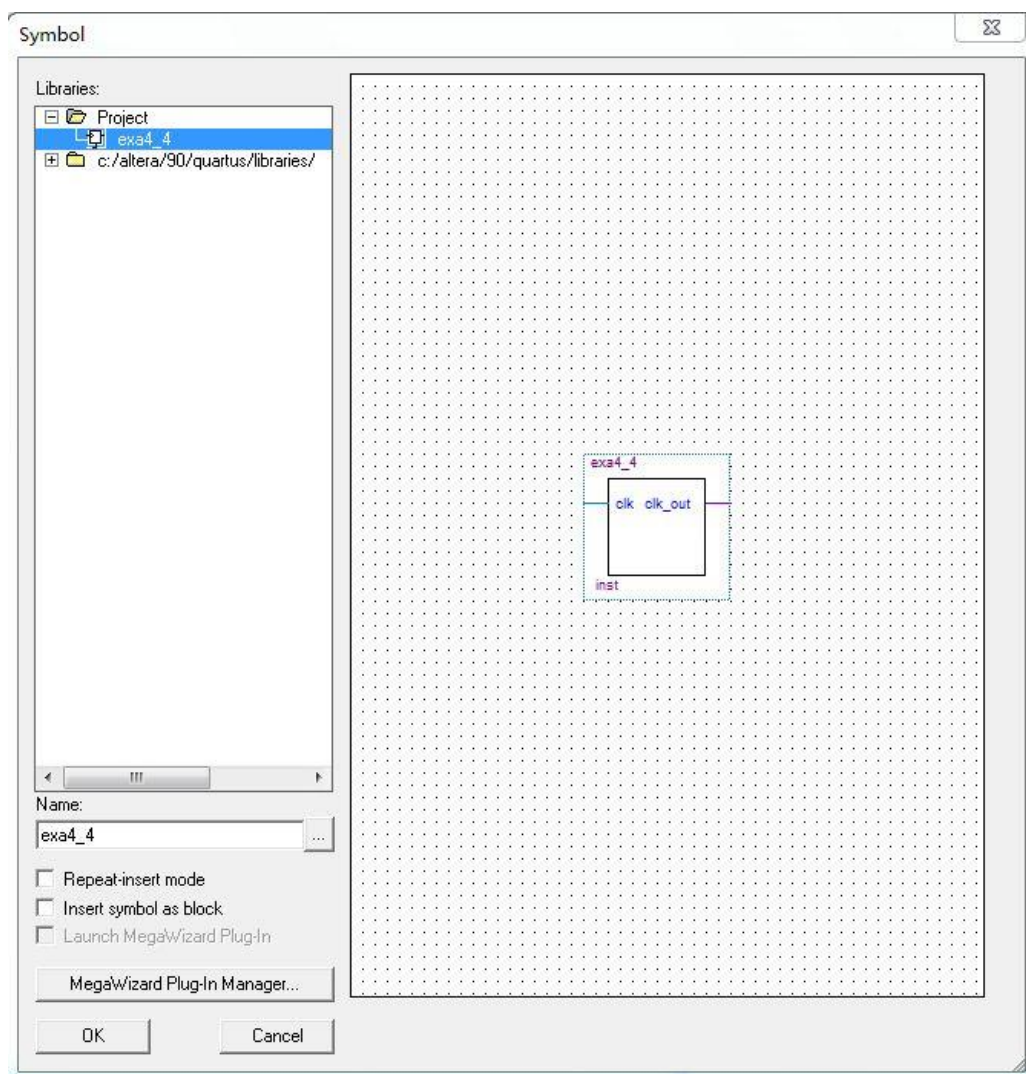
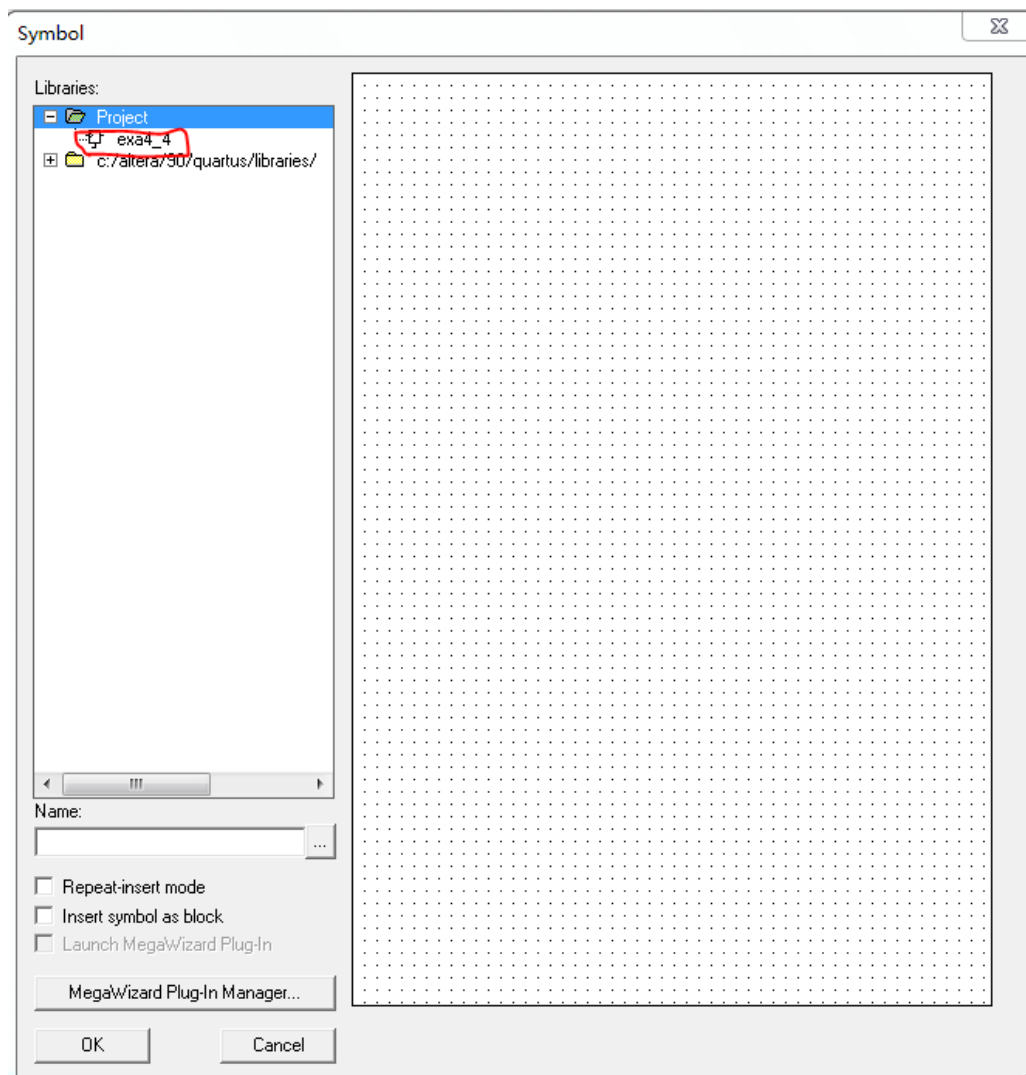
a ) 新建原理图文件



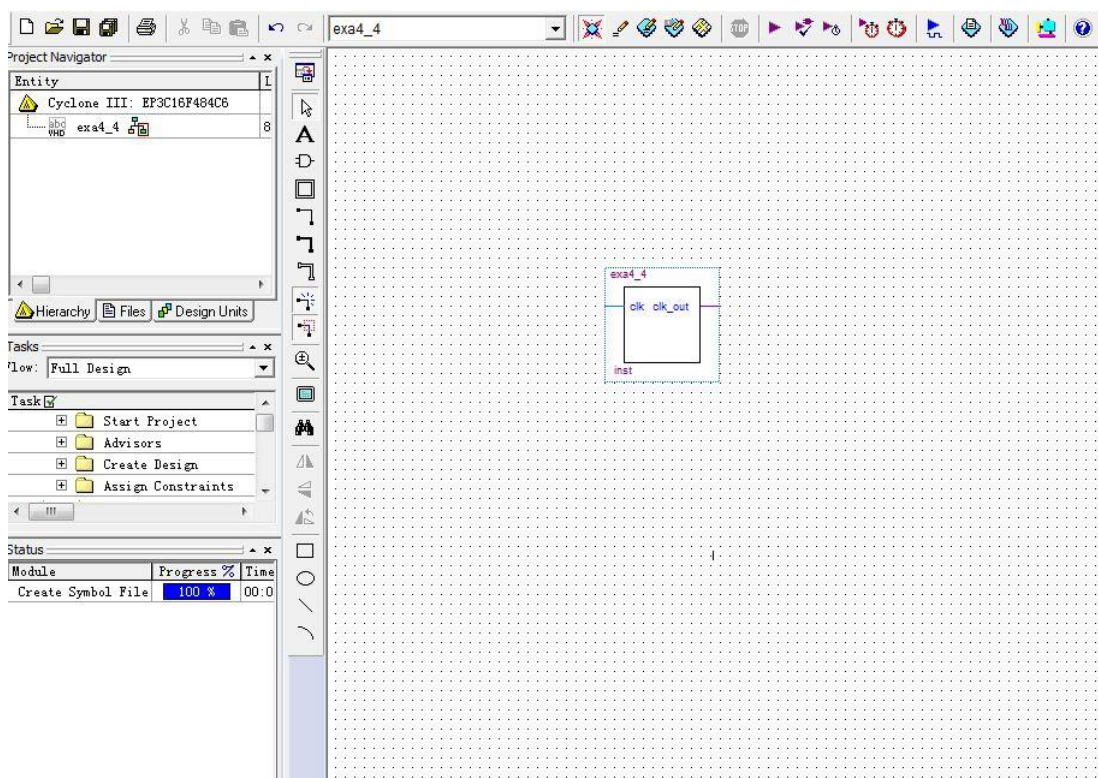
b)左键单击 symbol tool 按钮



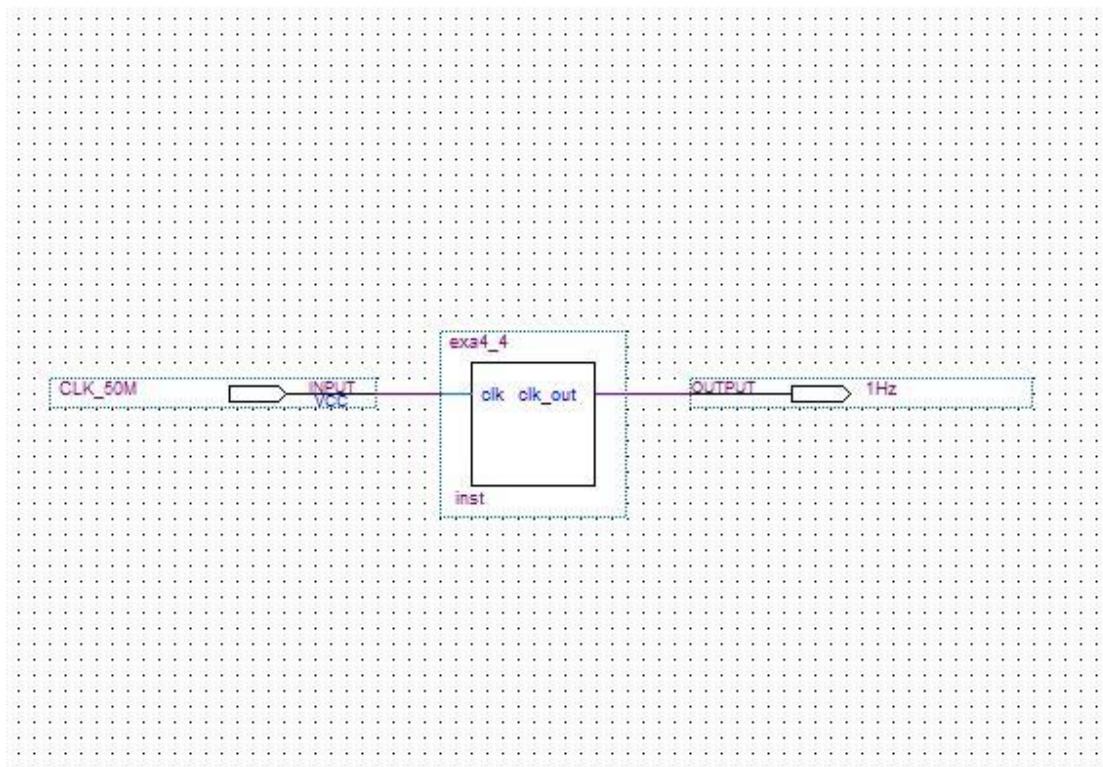
c ) 选择 project 下 exa4\_4



d) 将选择的 50M 分频模块放置到原理图文件中

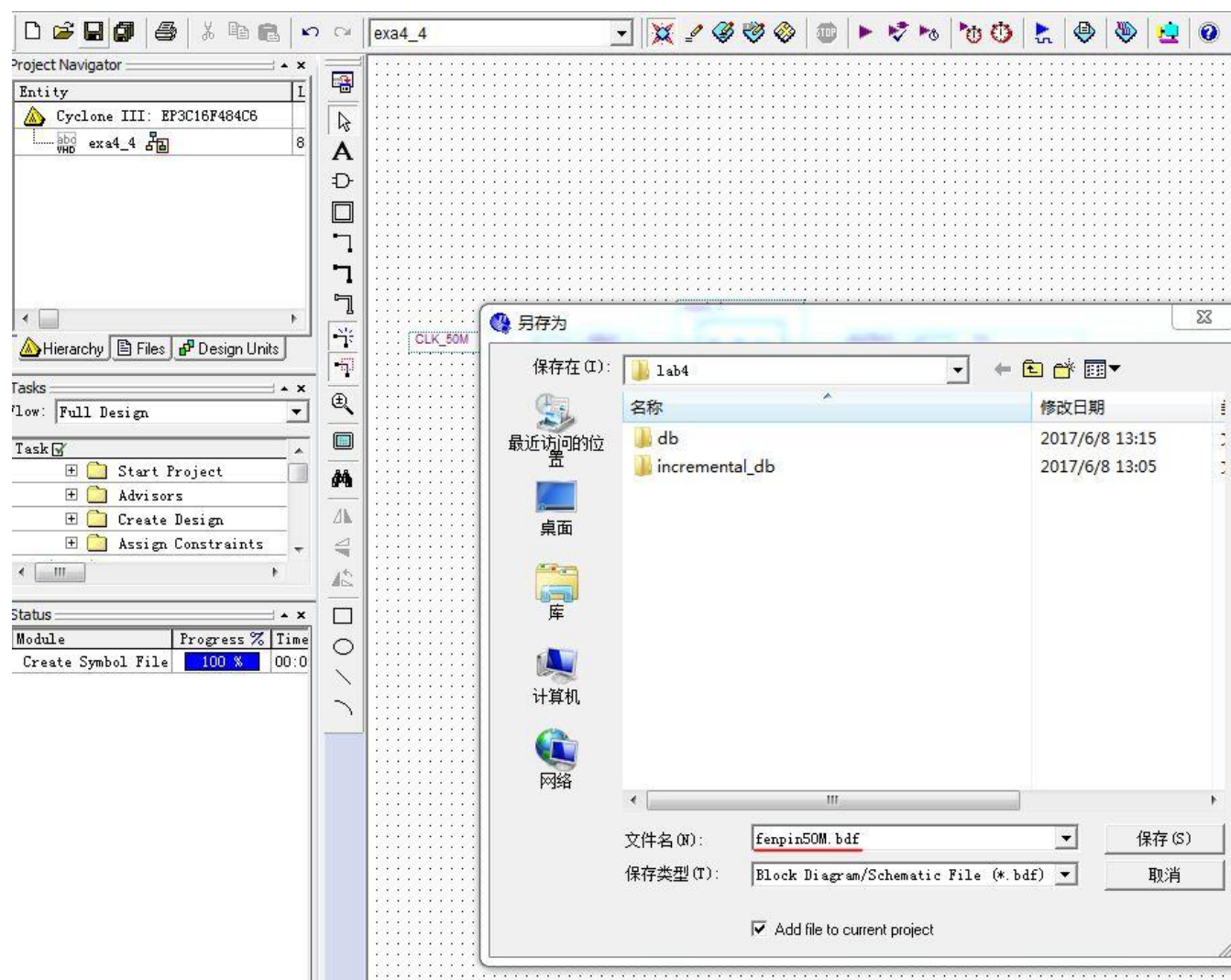


e) 利用生成的原理框图，可以实现 50M 分频，输出 1Hz 信号。输入的时钟信号连接实验板上的 PIN\_G21 引脚。



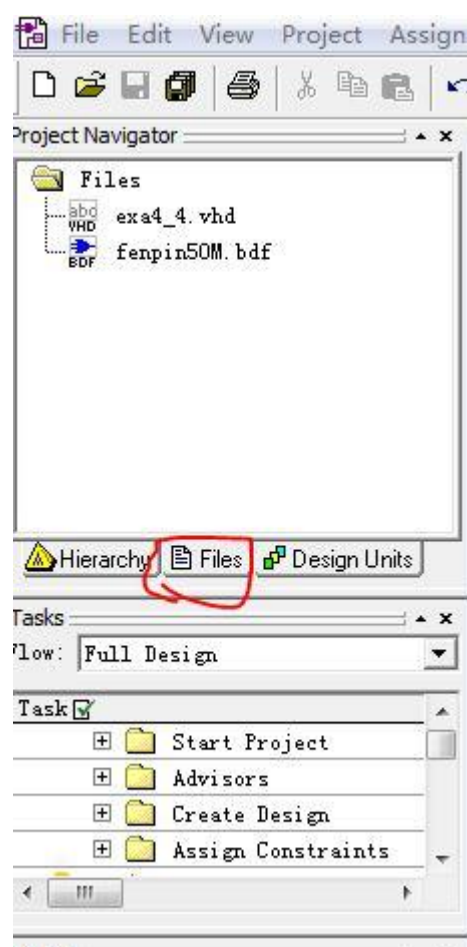
f) 保存新建的原理图文件

左键单击 save，弹出如下对话框，可以在文件名框中更改为合适的文件名，本例为 fenpin50M：



第七步：编译新生成的原理图文件

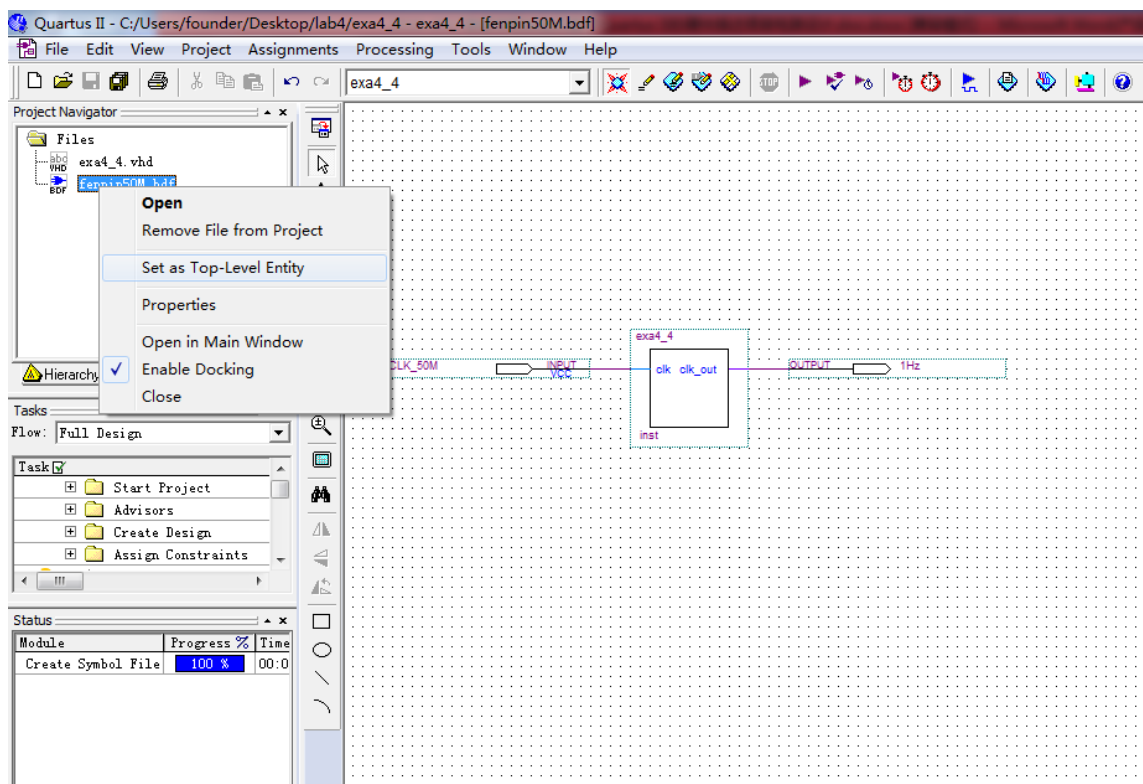
a) 在 Project Navigator 中选择 file 标签页



b) 设置原理图文件为顶层文件

右键单击原理图文件，在弹出的菜单栏中选择 **Set as Top Level Entity**





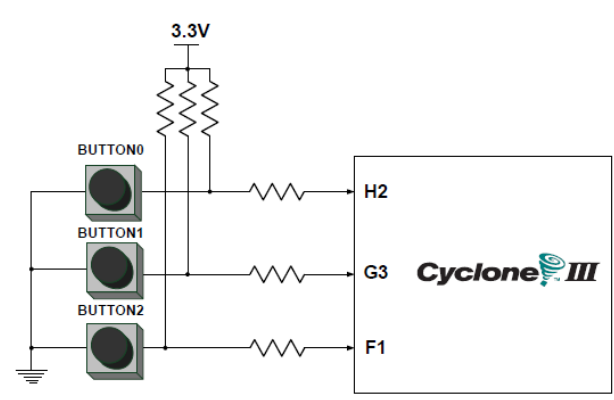
### c) 编译原理图文件

下载后的电路就可以将 50M 信号输入分频为 1Hz 信号输出。

### 总结

第四次实验需要按照类似步骤将编译过的 VHDL 文件生成原理图文件中可以调用的原理模块符号，实验中需要依次生成**分频模块**，**计数模块**，**译码模块**总计三个原理图模块，最后在顶层的原理图文件中完成系统连线。

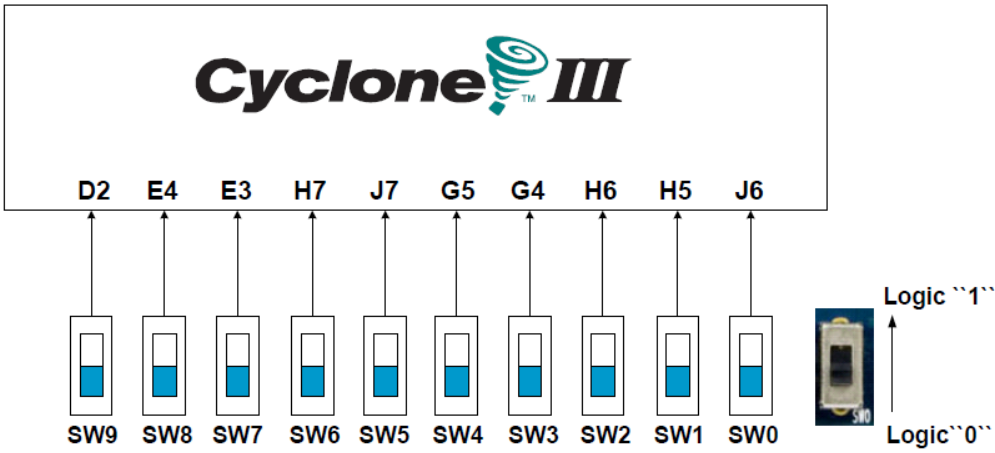
三、DE0 外接引脚说明



按钮开关与 Cyclone III FPGA 的连接

按钮开关的引脚分配

信号名	FPGA 引脚号	说明
BUTTON [0]	PIN_ H2	Pushbutton[0]
BUTTON [1]	PIN_ G3	Pushbutton[1]
BUTTON [2]	PIN_ F1	Pushbutton[2]

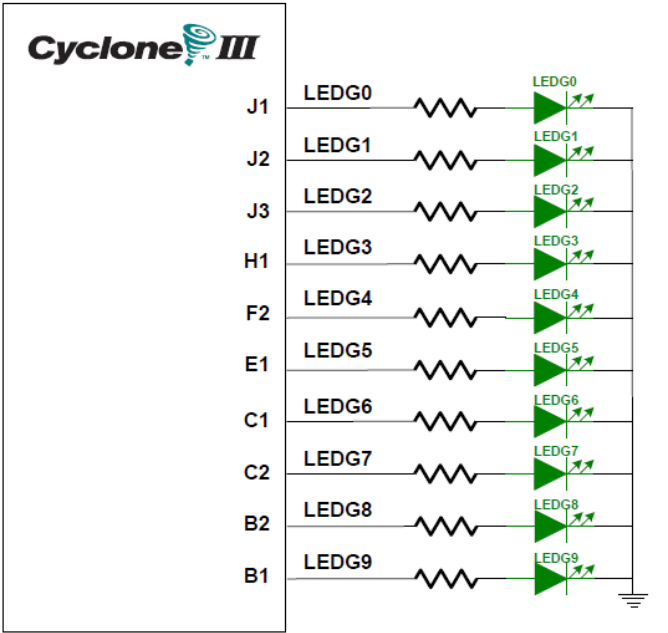


拨动开关与 CycloneIII FPGA 的连接

拨动开关的引脚分配

信号名	FPGA 引脚号.	说明
SW[0]	PIN_J6	Slide Switch[0]
SW[1]	PIN_H5	Slide Switch[1]
SW[2]	PIN_H6	Slide Switch[2]
SW[3]	PIN_G4	Slide Switch[3]

SW[4]	PIN_G5	Slide Switch[4]
SW[5]	PIN_J7	Slide Switch[5]
SW[6]	PIN_H7	Slide Switch[6]
SW[7]	PIN_E3	Slide Switch[7]
SW[8]	PIN_E4	Slide Switch[8]
SW[9]	PIN_D2	Slide Switch[9]



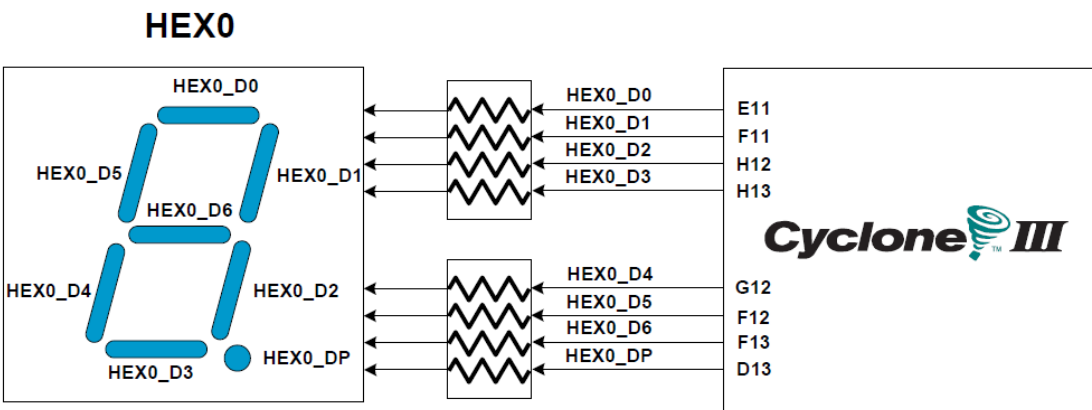
LED 灯与 Cyclone III FPGA 的连接

LED 灯的引脚分配

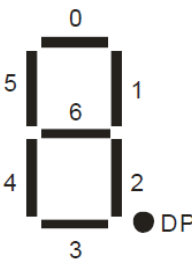
信号名	FPGA 引脚号	说明
LEDG[0]	PIN_J1	LED Green[0]
LEDG[1]	PIN_J2	LED Green[1]
LEDG[2]	PIN_J3	LED Green[2]
LEDG[3]	PIN_H1	LED Green[3]
LEDG[4]	PIN_F2	LED Green[4]
LEDG[5]	PIN_E1	LED Green[5]
LEDG[6]	PIN_C1	LED Green[6]



LEDG[7]	PIN_C2	LED Green[7]
LEDG[8]	PIN_B2	LED Green[8]
LEDG[9]	PIN_B1	LED Green[9]



七段数码管与 Cyclone III FPGA 芯片的连接示意图



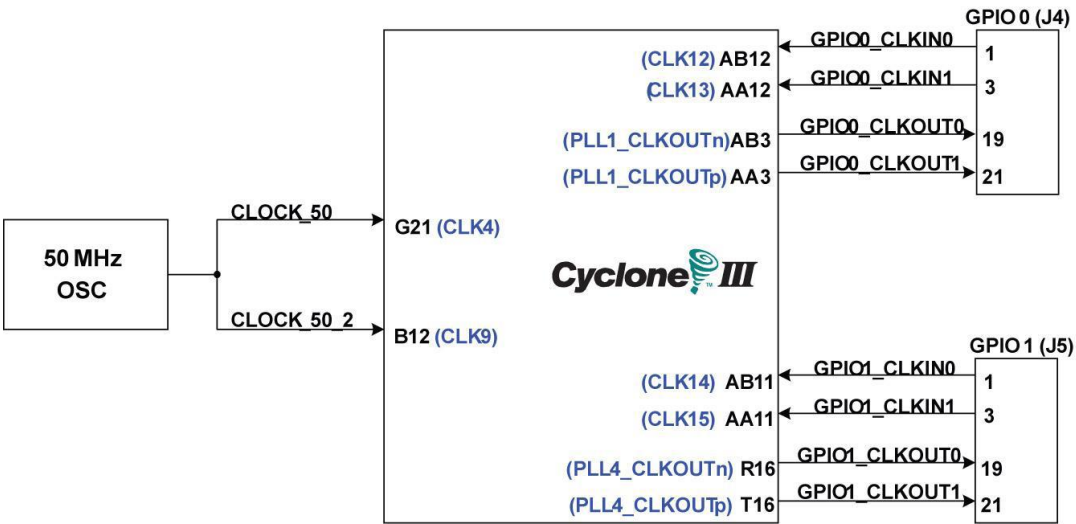
七段数码管每个字段的相应编号

七段数码管的引脚配置

信号名	FPGA 引脚号	说明
HEX0_D[0]	PIN_E11	Seven Segment Digit 0[0]
HEX0_D[1]	PIN_F11	Seven Segment Digit 0[1]
HEX0_D[2]	PIN_H12	Seven Segment Digit 0[2]
HEX0_D[3]	PIN_H13	Seven Segment Digit 0[3]
HEX0_D[4]	PIN_G12	Seven Segment Digit 0[4]
HEX0_D[5]	PIN_F12	Seven Segment Digit 0[5]

HEX0_D[6]	PIN_F13	Seven Segment Digit 0[6]
HEX0_DP	PIN_D13	Seven Segment Decimal Point 0
HEX1_D[0]	PIN_A13	Seven Segment Digit 1[0]
HEX1_D[1]	PIN_B13	Seven Segment Digit 1[1]
HEX1_D[2]	PIN_C13	Seven Segment Digit 1[2]
HEX1_D[3]	PIN_A14	Seven Segment Digit 1[3]
HEX1_D[4]	PIN_B14	Seven Segment Digit 1[4]
HEX1_D[5]	PIN_E14	Seven Segment Digit 1[5]
HEX1_D[6]	PIN_A15	Seven Segment Digit 1[6]
HEX1_DP	PIN_B15	Seven Segment Decimal Point 1
HEX2_D[0]	PIN_D15	Seven Segment Digit 2[0]
HEX2_D[1]	PIN_A16	Seven Segment Digit 2[1]
HEX2_D[2]	PIN_B16	Seven Segment Digit 2[2]
HEX2_D[3]	PIN_E15	Seven Segment Digit 2[3]
HEX2_D[4]	PIN_A17	Seven Segment Digit 2[4]
HEX2_D[5]	PIN_B17	Seven Segment Digit 2[5]
HEX2_D[6]	PIN_F14	Seven Segment Digit 2[6]
HEX2_DP	PIN_A18	Seven Segment Decimal Point 2
HEX3_D[0]	PIN_B18	Seven Segment Digit 3[0]
HEX3_D[1]	PIN_F15	Seven Segment Digit 3[1]

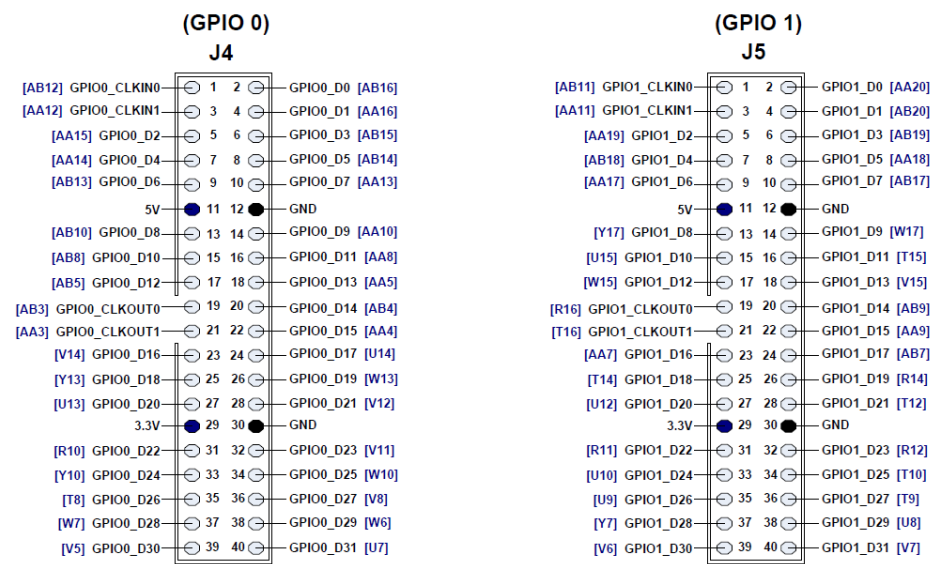
HEX3_D[2]	PIN_A19	Seven Segment Digit 3[2]
HEX3_D[3]	PIN_B19	Seven Segment Digit 3[3]
HEX3_D[4]	PIN_C19	Seven Segment Digit 3[4]
HEX3_D[5]	PIN_D19	Seven Segment Digit 3[5]
HEX3_D[6]	PIN_G15	Seven Segment Digit 3[6]
HEX3_DP	PIN_G16	Seven Segment Decimal Point 3



时钟分配电路的方块图

时钟输入的引脚分配

信号名	FPGA 引脚号.	说明
CLOCK_50	PIN_G21	50 MHz clock input



GPIO0_D[12]	PIN_AB5	GPIO Connection 0 IO[12]
GPIO0_D[13]	PIN_AA5	GPIO Connection 0 IO[13]
GPIO0_D[14]	PIN_AB4	GPIO Connection 0 IO[14]
GPIO0_D[15]	PIN_AA4	GPIO Connection 0 IO[15]
GPIO0_D[16]	PIN_V14	GPIO Connection 0 IO[16]
GPIO0_D[17]	PIN_U14	GPIO Connection 0 IO[17]
GPIO0_D[18]	PIN_Y13	GPIO Connection 0 IO[18]
GPIO0_D[19]	PIN_W13	GPIO Connection 0 IO[19]
GPIO0_D[20]	PIN_U13	GPIO Connection 0 IO[20]
GPIO0_D[21]	PIN_V12	GPIO Connection 0 IO[21]
GPIO0_D[22]	PIN_R10	GPIO Connection 0 IO[22]
GPIO0_D[23]	PIN_V11	GPIO Connection 0 IO[23]
GPIO0_D[24]	PIN_Y10	GPIO Connection 0 IO[24]
GPIO0_D[25]	PIN_W10	GPIO Connection 0 IO[25]
GPIO0_D[26]	PIN_T8	GPIO Connection 0 IO[26]
GPIO0_D[27]	PIN_V8	GPIO Connection 0 IO[27]
GPIO0_D[28]	PIN_W7	GPIO Connection 0 IO[28]
GPIO0_D[29]	PIN_W6	GPIO Connection 0 IO[29]
GPIO0_D[30]	PIN_V5	GPIO Connection 0 IO[30]
GPIO0_D[31]	PIN_U7	GPIO Connection 0 IO[31]

GPIO0_CLKIN[0]	PIN_AB12	GPIO Connection 0 PLL In
GPIO0_CLKIN[1]	PIN_AA12	GPIO Connection 0 PLL In
GPIO0_CLKOUT[0]	PIN_AB3	GPIO Connection 0 PLL Out
GPIO0_CLKOUT[1]	PIN_AA3	GPIO Connection 0 PLL Out
GPIO1_D[0]	PIN_AA20	GPIO Connection 1 IO[0]
GPIO1_D[1]	PIN_AB20	GPIO Connection 1 IO[1]
GPIO1_D[2]	PIN_AA19	GPIO Connection 1 IO[2]
GPIO1_D[3]	PIN_AB19	GPIO Connection 1 IO[3]
GPIO1_D[4]	PIN_AB18	GPIO Connection 1 IO[4]
GPIO1_D[5]	PIN_AA18	GPIO Connection 1 IO[5]
GPIO1_D[6]	PIN_AA17	GPIO Connection 1 IO[6]
GPIO1_D[7]	PIN_AB17	GPIO Connection 1 IO[7]
GPIO1_D[8]	PIN_Y17	GPIO Connection 1 IO[8]
GPIO1_D[9]	PIN_W17	GPIO Connection 1 IO[9]
GPIO1_D[10]	PIN_U15	GPIO Connection 1 IO[10]
GPIO1_D[11]	PIN_T15	GPIO Connection 1 IO[11]
GPIO1_D[12]	PIN_W15	GPIO Connection 1 IO[12]
GPIO1_D[13]	PIN_V15	GPIO Connection 1 IO[13]
GPIO1_D[14]	PIN_AB9	GPIO Connection 1 IO[14]
GPIO1_D[15]	PIN_AA9	GPIO Connection 1 IO[15]

GPIO1_D[16]	PIN_AA7	GPIO Connection 1 IO[16]
GPIO1_D[17]	PIN_AB7	GPIO Connection 1 IO[17]
GPIO1_D[18]	PIN_T14	GPIO Connection 1 IO[18]
GPIO1_D[19]	PIN_R14	GPIO Connection 1 IO[19]
GPIO1_D[20]	PIN_U12	GPIO Connection 1 IO[20]
GPIO1_D[21]	PIN_T12	GPIO Connection 1 IO[21]
GPIO1_D[22]	PIN_R11	GPIO Connection 1 IO[22]
GPIO1_D[23]	PIN_R12	GPIO Connection 1 IO[23]
GPIO1_D[24]	PIN_U10	GPIO Connection 1 IO[24]
GPIO1_D[25]	PIN_T10	GPIO Connection 1 IO[25]
GPIO1_D[26]	PIN_U9	GPIO Connection 1 IO[26]
GPIO1_D[27]	PIN_T9	GPIO Connection 1 IO[27]
GPIO1_D[28]	PIN_Y7	GPIO Connection 1 IO[28]
GPIO1_D[29]	PIN_U8	GPIO Connection 1 IO[29]
GPIO1_D[30]	PIN_V6	GPIO Connection 1 IO[30]
GPIO1_D[31]	PIN_V7	GPIO Connection 1 IO[31]
GPIO1_CLKIN[0]	PIN_AB11	GPIO Connection 1 PLL In
GPIO1_CLKIN[1]	PIN_AA11	GPIO Connection 1 PLL In
GPIO1_CLKOUT[0]	PIN_R16	GPIO Connection 1 PLL Out