微机原理及接口课程设计

实 验 报 告

|  |  |  |
| --- | --- | --- |
| 评 语： | 成绩 |  |
| 教 师：  年 月 日 | | |

班 级： 1513011

学 号：

姓 名：

地 点：

时 间：

实验一 51核发光二极管及数码管实验

一 实验题目

FPGA中8051核原理介绍以及51核发光二极管及数码管实验

二 实验环境

Quartus II

三 实验要求

1. 编写程序使led灯（LED0,LED6,LED7）每隔一段时间闪烁一次。

提示：编写延时程序 delay().（用for循环实现）

led灯的闪烁即是不停的亮和灭，即令通用IO口每隔一段时间的输出取反。如

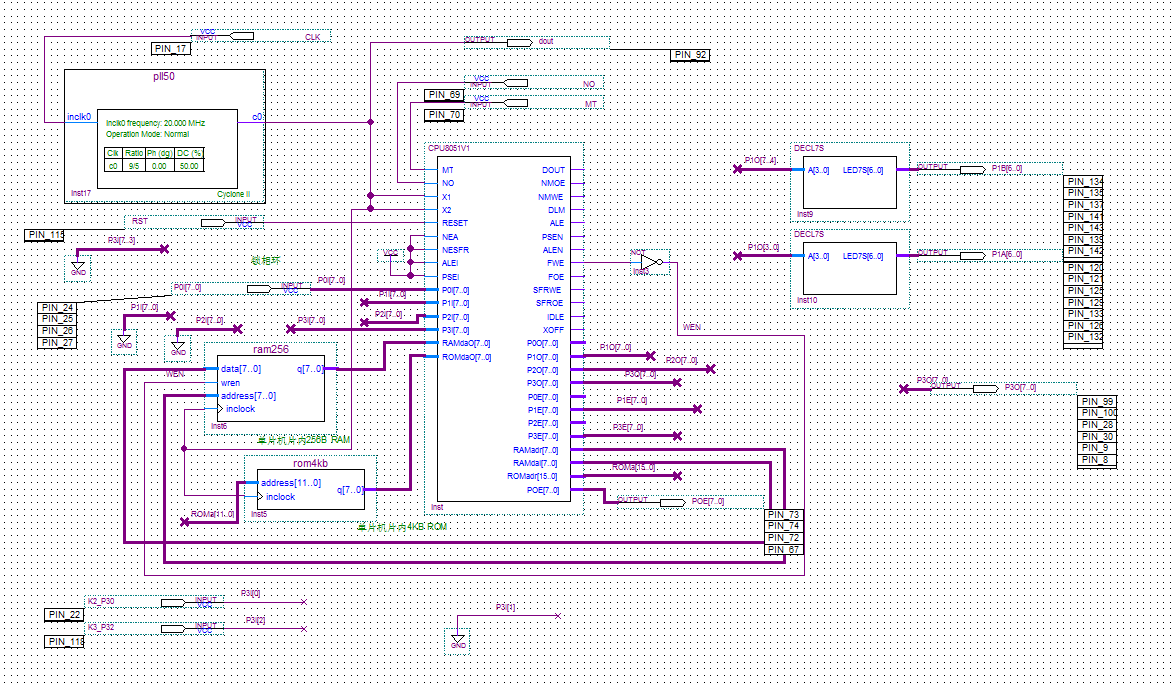
LED\_1=~LED\_1;

2.控制2个数码管显示数字。

提示：数码管是由P1口控制的。

四 实验设计

1.实验中原理图分析

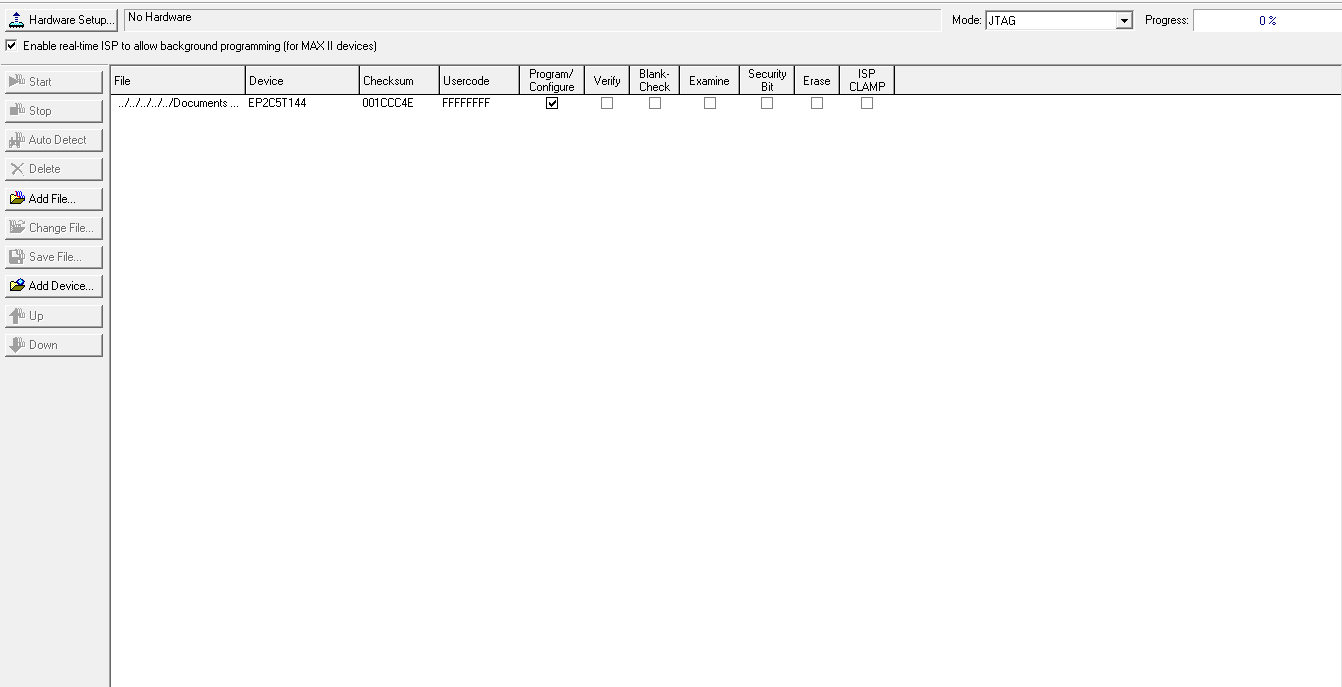


2. FPGA中51核的编绎及下载（Prog51Core工程）

（1）编绎工程。(做实验时不用重新编译，直接下载)

（2）连接下载JTAG口，进行下载，点击工具条上的图标，进行下载。

（3）点Hardware Setup选择下载器，只勾上program/configure，选start下载。



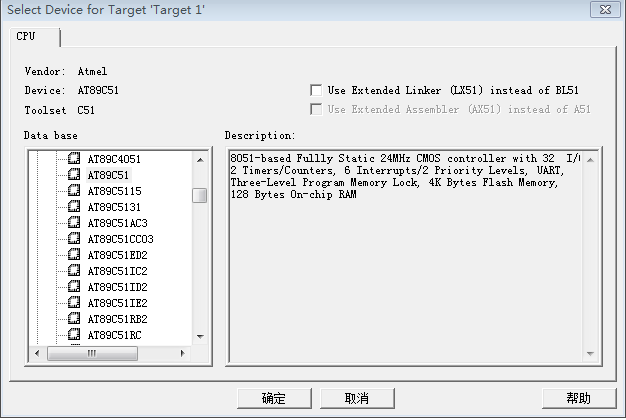
3.keil下51单片机编程

打开keil软件

点击project🡪uVersion project

在弹出的select device窗口中选择

Atmel🡪89C51



4.新建一个文档,输入对应的代码.保存到对应的工程目录中.

保存成.c文件.

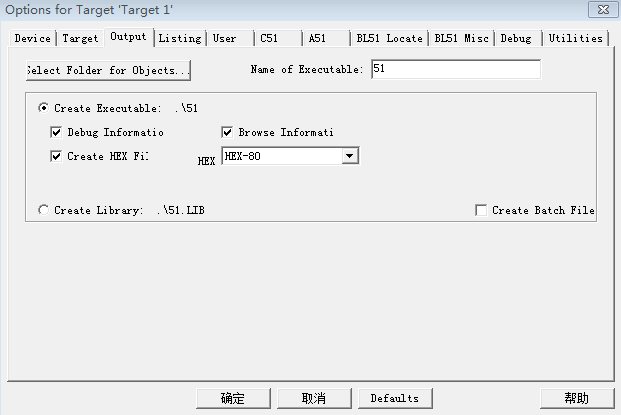
将新建的源文件添加到工程中.

右键单击Source Group 1，选择Add files to Group

Project🡪build target编译工程。

KEIL默认情况下是不生成.hex下载文件的。

需要如下设置



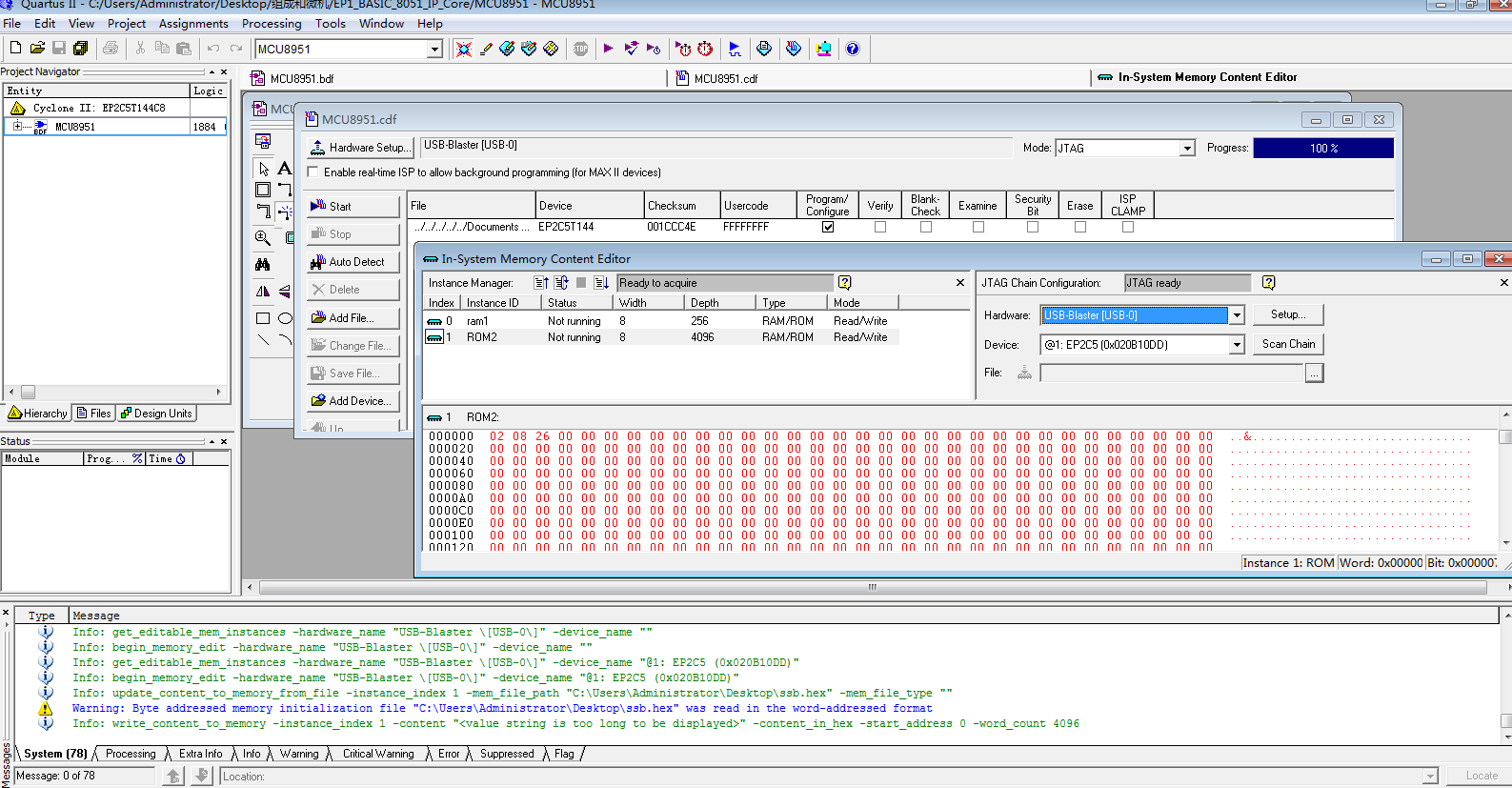
1. 编绎C程序，

生成目标代码hex文件,打开Quartus II 中工具下的系统内存数据编缉器,

按窗口右侧的setup进行JTAG口查找，查找后能找到一个RAM，一个ROM。

ROM内的数据既为程序的目标代码，右击ROM选import data from file

选择生成的.hex目标文件，继续右击ROM，选WriteData.按复位键KEY1运行



1. 实验代码如下：

#include<reg51.h>

sbit LED\_1=P3^6;

sbit LED\_2=P3^7;

void delay(int del)

{

int i,j;

for(i=0;i<del;i++)

for(j=0;j<5000;j++);

}

void LED\_LM()

{

LED\_1=~LED\_1;

LED\_2=~LED\_2;

}

void main()

{

LED\_1=1;

LED\_2=0;

while(1)

{

delay(100);

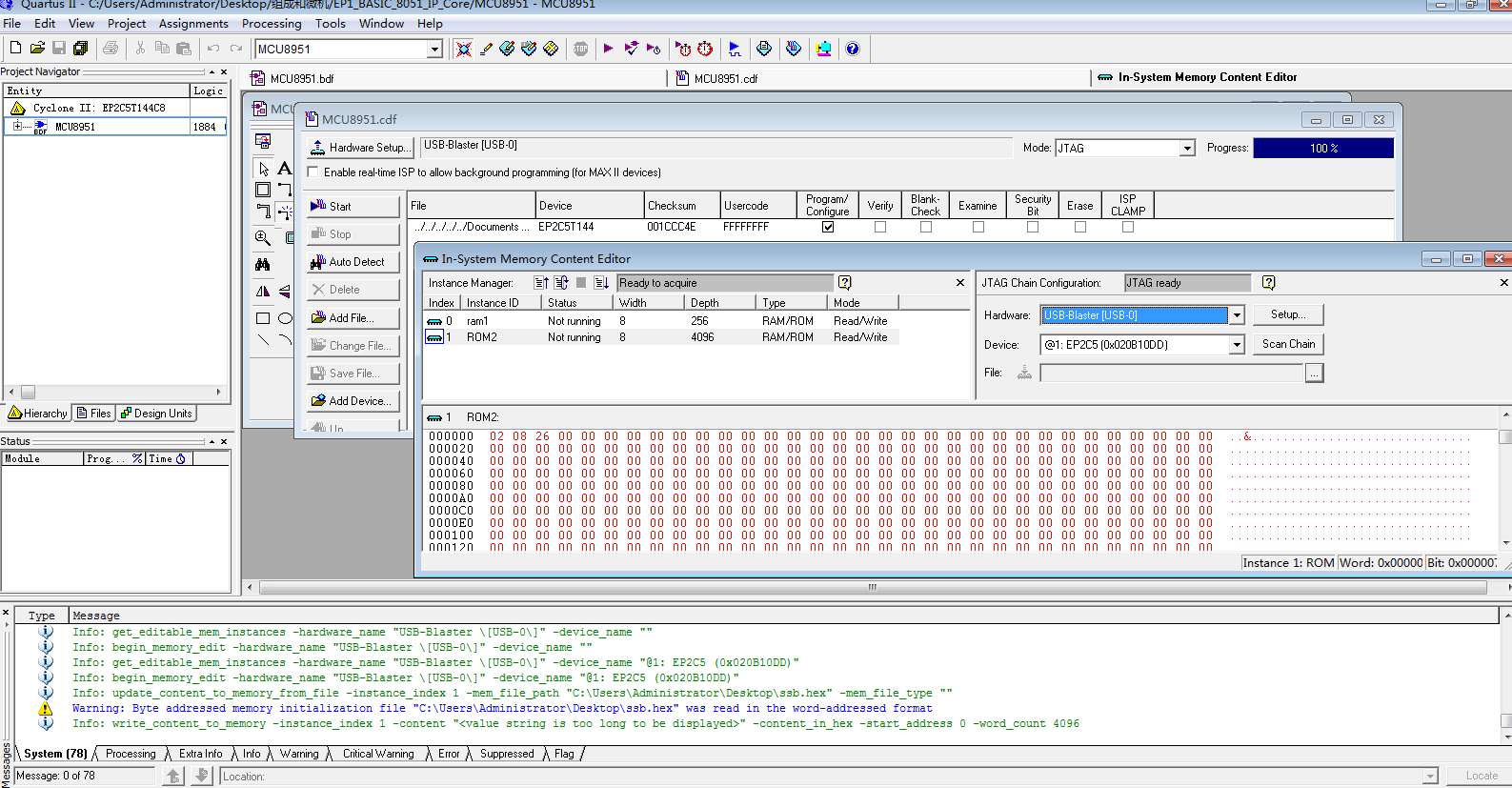
LED\_LM();

}

}

五 实验仿真与测试

写入后结果如下



六 实验结果分析

实验中LED灯交替闪烁，因为将两个输出一个为0， 一个为1，所以输出时两个是不同步的。

七 实验小结

本次实验开始了微机部分，经过本次实验，我学会了如何将程序加载到开发板中。学会了开发板的使用。

实验二 键盘扫描

一 实验题目

键盘扫描

二 实验环境

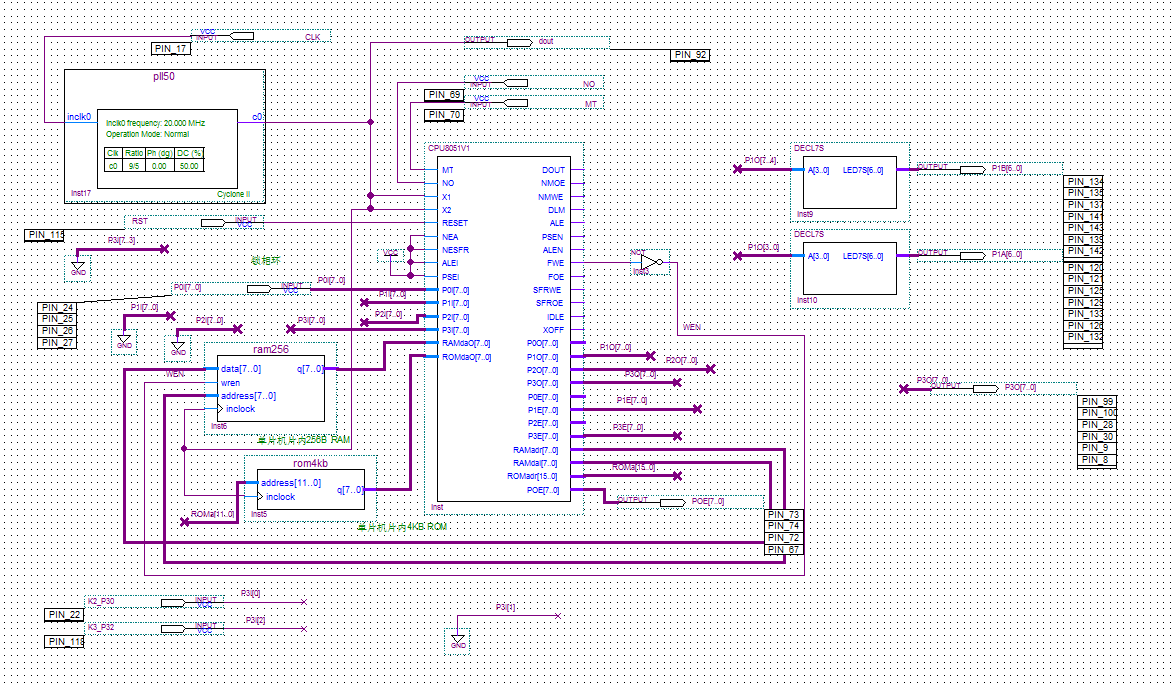
Quartus II

三 实验要求

使用查询方式进行键盘扫描码判断（数码管输出）。

1. 实验设计

1.实验中原理图分析

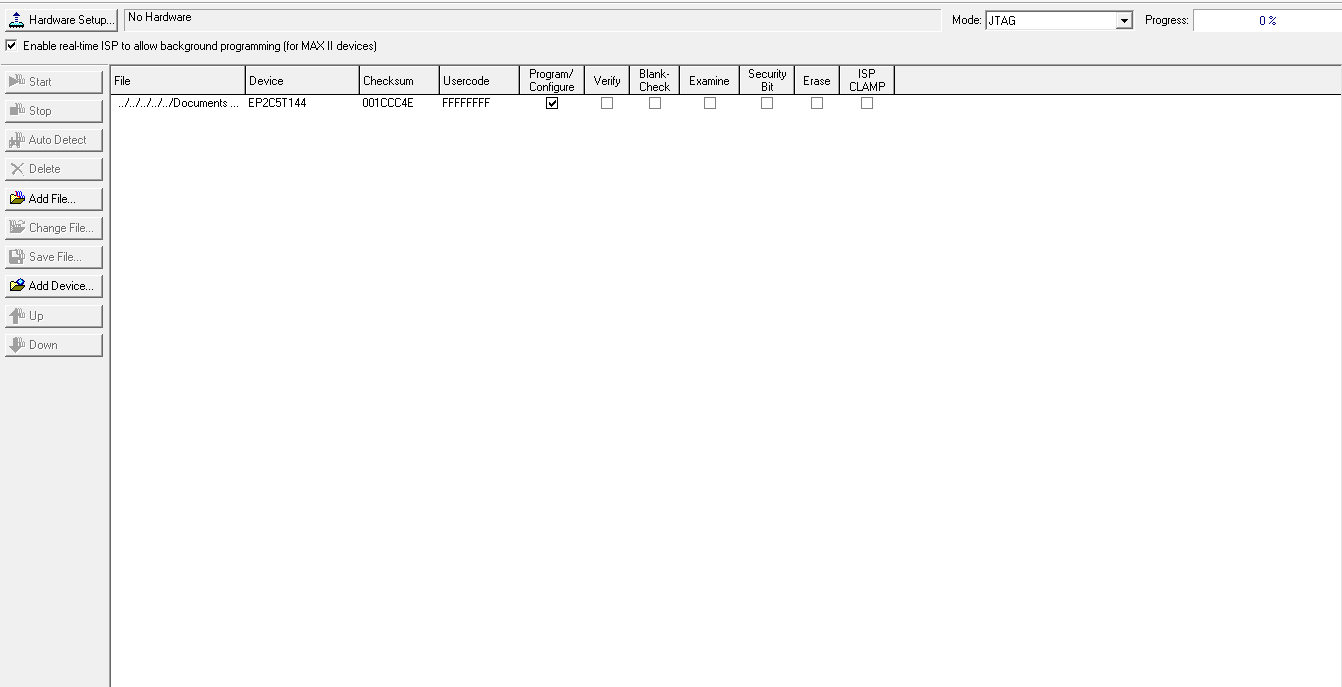


2. FPGA中编绎及下载

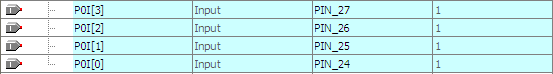
（1）编绎工程。(做实验时不用重新编译，直接下载)

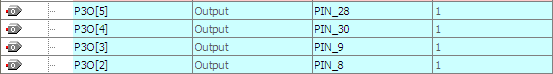
（2）连接下载JTAG口，进行下载，点击工具条上的图标，进行下载。

（3）点Hardware Setup选择下载器，只勾上program/configure，选start下载。



(4)FPGA中键盘扫描设置





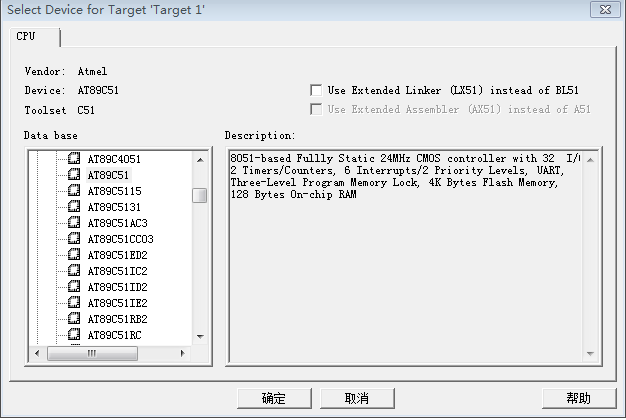
3.keil下51单片机编程

打开keil软件

点击project🡪uVersion project

在弹出的select device窗口中选择

Atmel🡪89C51



4.新建一个文档,输入对应的代码.保存到对应的工程目录中.

保存成.c文件.

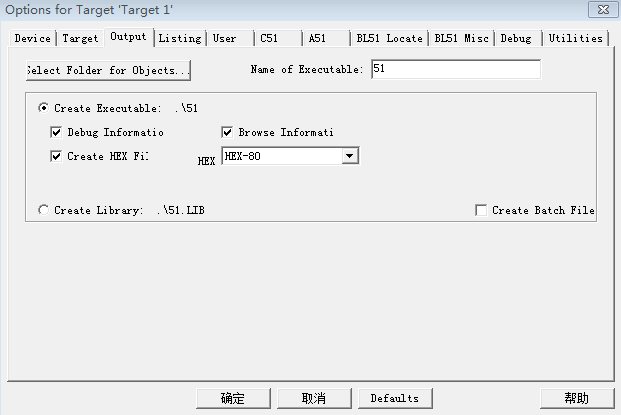
将新建的源文件添加到工程中.

右键单击Source Group 1，选择Add files to Group

Project🡪build target编译工程。

KEIL默认情况下是不生成.hex下载文件的。

需要如下设置



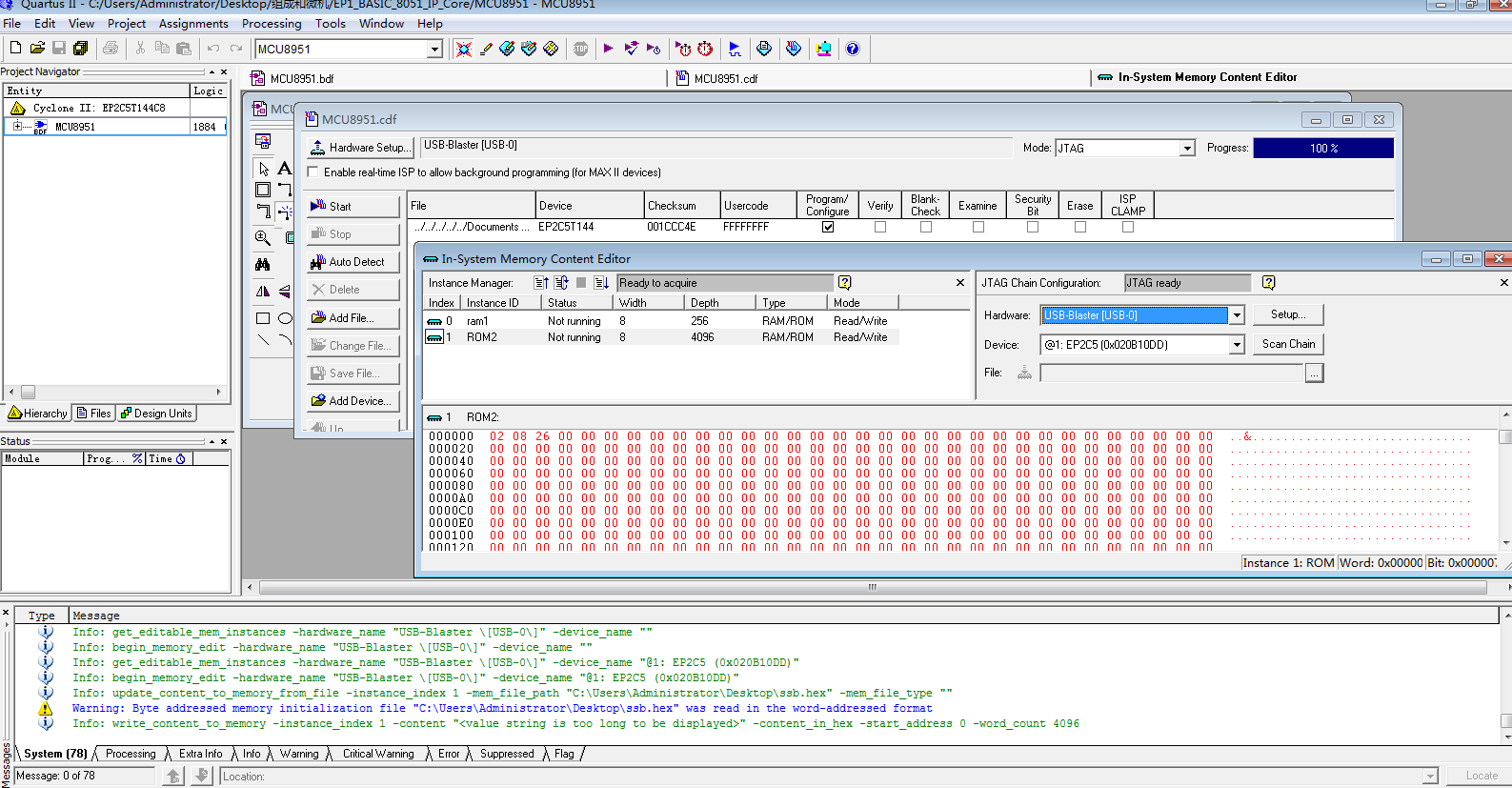
5.编绎C程序，

生成目标代码hex文件,打开Quartus II 中工具下的系统内存数据编缉器,

按窗口右侧的setup进行JTAG口查找，查找后能找到一个RAM，一个ROM。

ROM内的数据既为程序的目标代码，右击ROM选import data from file

选择生成的.hex目标文件，继续右击ROM，选WriteData.按复位键KEY1运行



6.实验代码如下：

#include<reg52.h>

unsigned char code DIG\_CODE[16]={0x3F, 0x06, 0x5B, 0x4F,

0x66,0x6D, 0x7D, 0x07,

0x7F, 0x6F, 0x77,0x7C,

0x39, 0x5E, 0x79, 0x71};

void Delay10ms(unsigned int c) //误差 0us

{

unsigned char a, b;

//c已经在传递过来的时候已经赋值了，所以在for语句第一句就不用赋值了

for (;c>0;c--)

{

for (b=38;b>0;b--)

{

for (a=130;a>0;a--);

}

}

}

void main()

{

char a=0;

P0=~DIG\_CODE[0];

while(1)

{

P2=0x0F;

if(0x0F != P2)//读取按键是否按下

{

Delay10ms(1);//延时10ms进行消抖

if(0x0F != P2)//再次检测键盘是否按下

{

//测试列

P2=0X0F;

switch(P2)

{

case(0X07):

a=0; break;

case(0X0b):

a=4; break;

case(0X0d):

a=8; break;

case(0X0e):

a=12; break;

}

//测试行

P2=0XF0;

switch(P2)

{

case(0X70):

break;

case(0Xb0):

a=a+1; break;

case(0Xd0):

a=a+2; break;

case(0Xe0):

a=a+3;break;

}

while(0xf0 != P2) //检测按键松手检测

{

Delay10ms(1);

P0=~DIG\_CODE[a];

}

}

}

}

}

五 实验仿真与测试

按下数字键，屏幕显示0，在4\*4键盘上每次按一个键依次出现0-F

六 实验结果分析

仿真结果符合情况，说明实验成功。

七 实验小结

通过本次实验，我发现FPGA板功能非常强大。通过对FPGA的学习，我更加深刻的理解了开发板的使用规则。

实验三 中断系统和定时/计数器

一 实验题目

中断系统和定时/计数器

二 实验环境

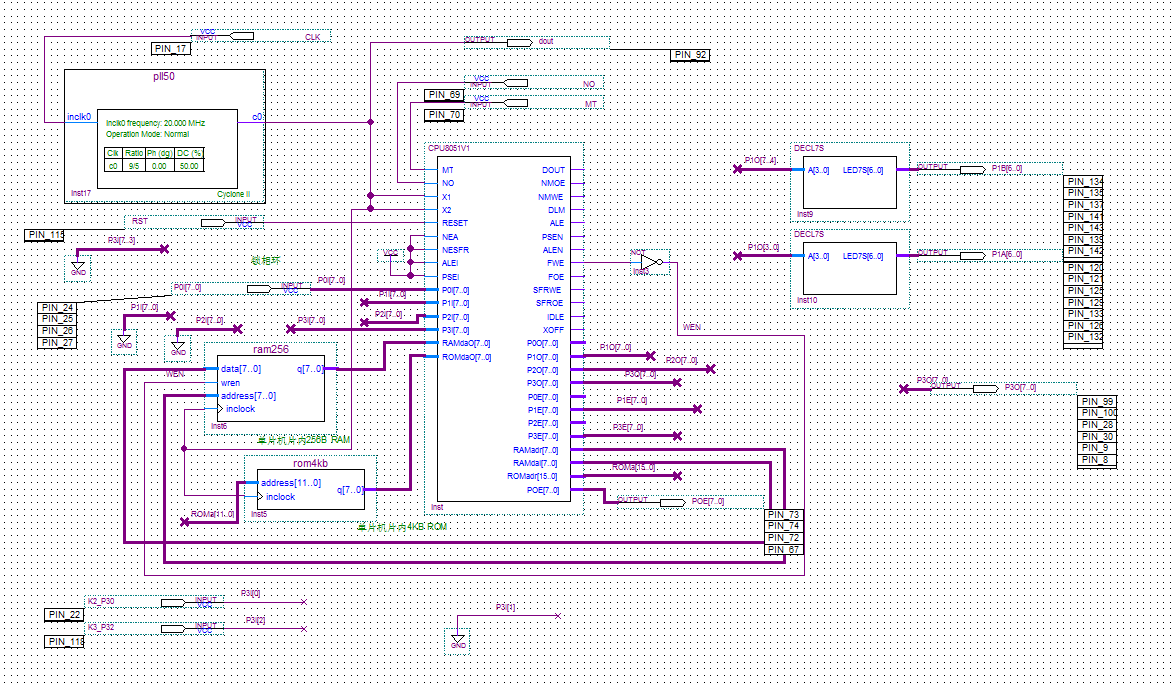
Quartus II

三 实验要求

使用定时器中断控制发光二极管每隔一段时间闪烁一次。

1. 实验设计

1.实验中原理图分析

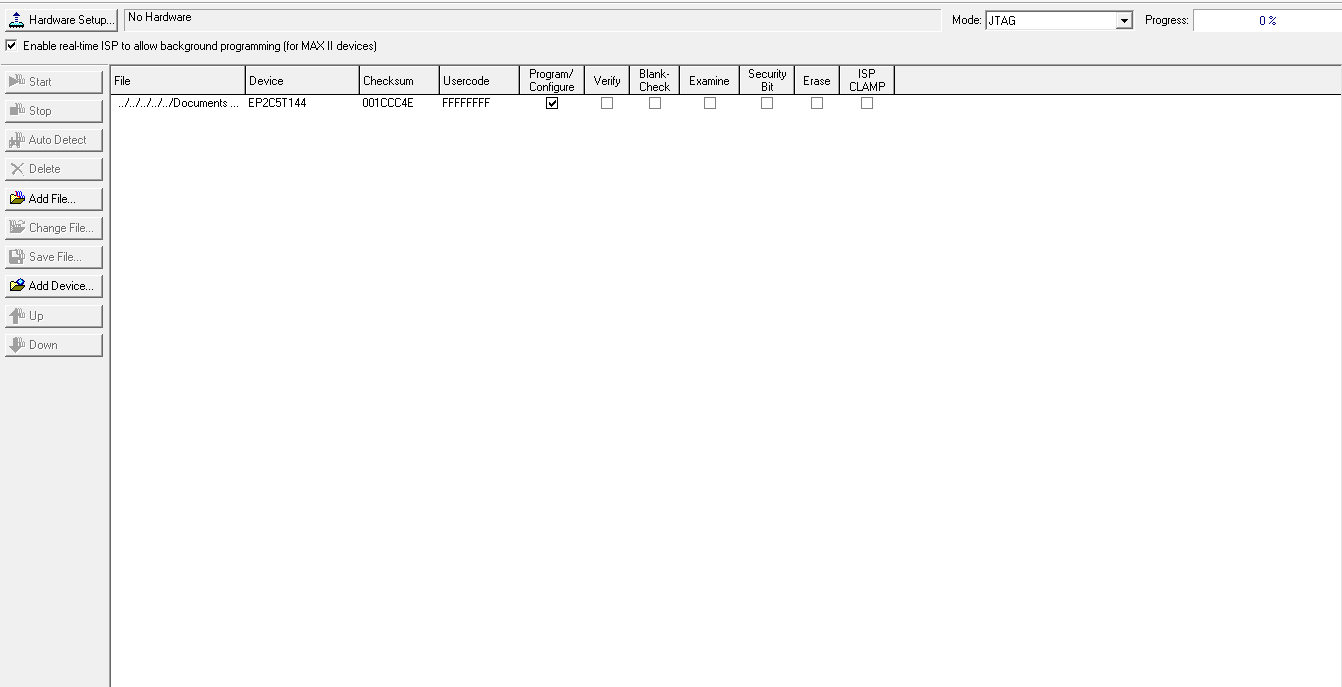


2. FPGA中编绎及下载

（1）编绎工程。(做实验时不用重新编译，直接下载)

（2）连接下载JTAG口，进行下载，点击工具条上的图标，进行下载。

（3）点Hardware Setup选择下载器，只勾上program/configure，选start下载。



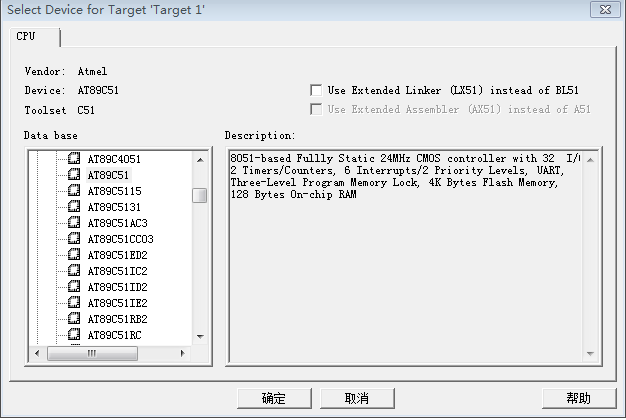
3.keil下51单片机编程

打开keil软件

点击project🡪uVersion project

在弹出的select device窗口中选择

Atmel🡪89C51



4.新建一个文档,输入对应的代码.保存到对应的工程目录中.

保存成.c文件.

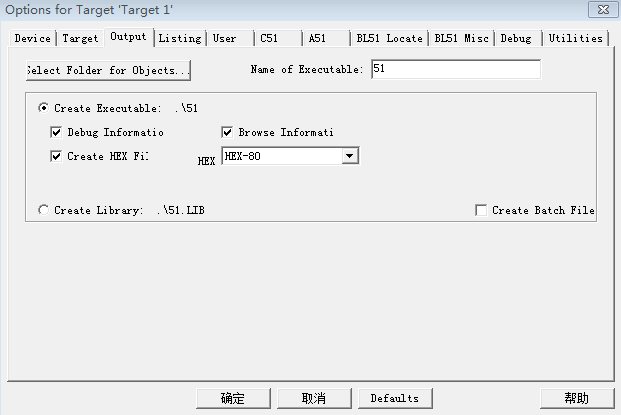
将新建的源文件添加到工程中.

右键单击Source Group 1，选择Add files to Group

Project🡪build target编译工程。

KEIL默认情况下是不生成.hex下载文件的。

需要如下设置



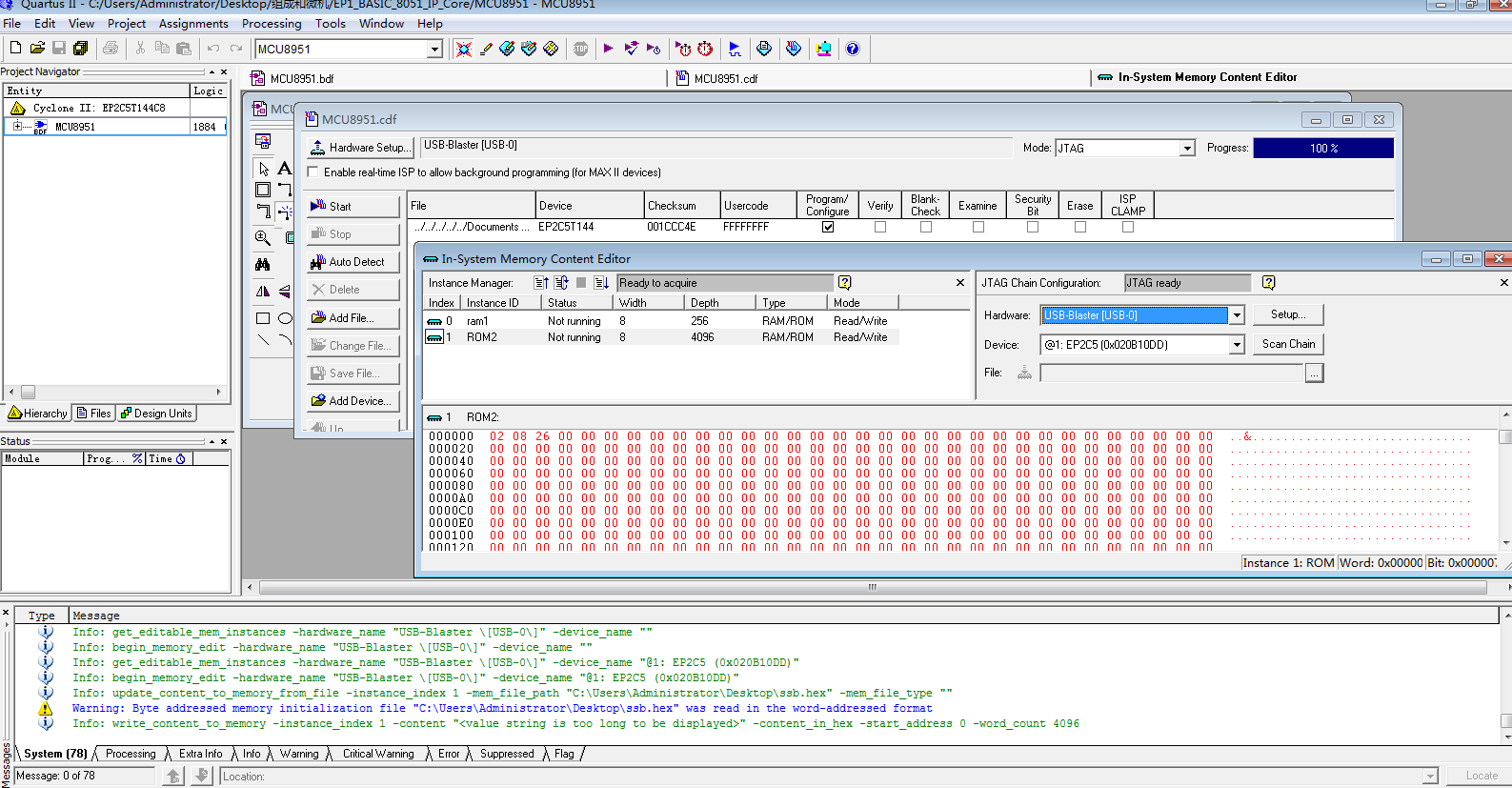
5.编绎C程序，

生成目标代码hex文件,打开Quartus II 中工具下的系统内存数据编缉器,

按窗口右侧的setup进行JTAG口查找，查找后能找到一个RAM，一个ROM。

ROM内的数据既为程序的目标代码，右击ROM选import data from file

选择生成的.hex目标文件，继续右击ROM，选WriteData.按复位键KEY1运行



6.实验代码如下：

#include<reg52.h>

int num=0;

void main()

{

P0=0XFF;

TMOD=0X01;

TH0=(65536-45872)/256;

TL0=(65536-45872)/256;

EA=1;

ET0=1;

TR0=1;

while(1)

}

void T0\_timer() interrupt 1

{

TH0=(65536-45872)/256;

TL0=(65536-45872)/256;

num++;

if(num==20)

{

P0=~P0;

num=0;

}

}

五 实验仿真与测试

每一秒灯闪烁一次。

六 实验结果分析

程序开始计时，每50毫秒中断一次，20次中断是可以使得灯闪烁一次。

七 实验小结

通过本次实验，我了解了中断系统的定时与计时功能。理解了中断响应的条件:

1.中断源有中断请求；

2.此中断源的中断允许位为1；

3.CPU开中断（即EA=1）。

以上三条同时满足时，CPU才有可能响应中断。

理清了定时/计数器的工作原理以及掌握了用FPGA设计有关中断系统的实验。

实验四 FPGA中串口设计

一 实验题目

FPGA中串口设计

二 实验环境

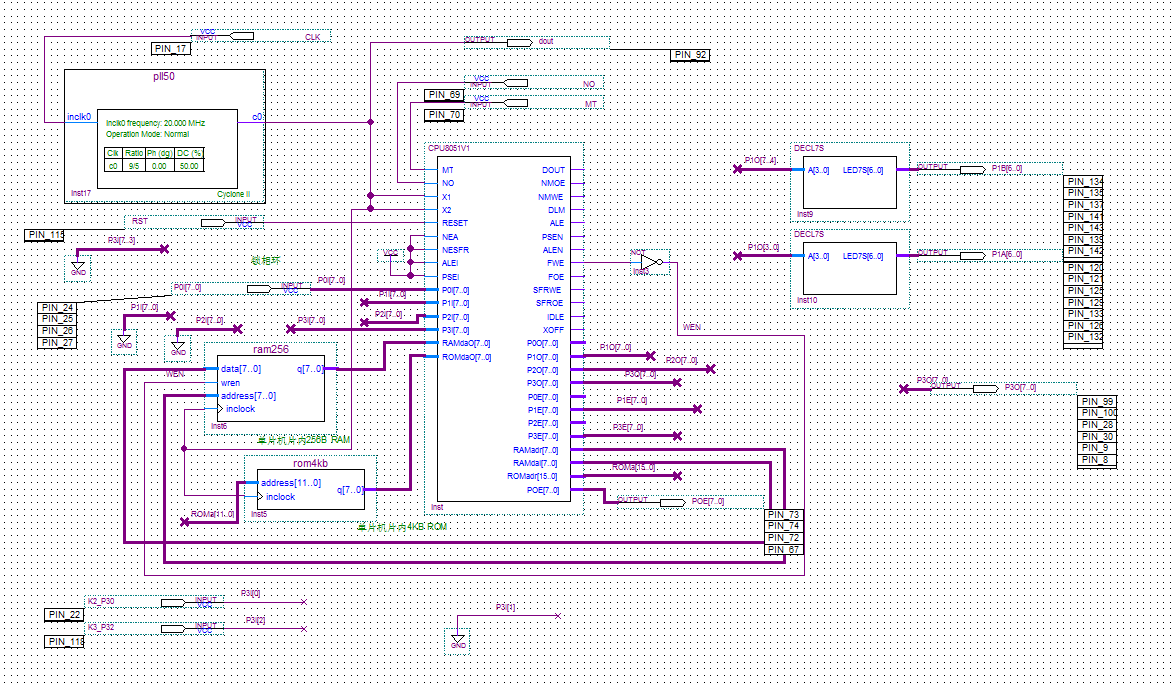
Quartus II

三 实验要求

实验要求完成较为复杂的串口通信工作

1. 实验设计

1.实验中原理图分析

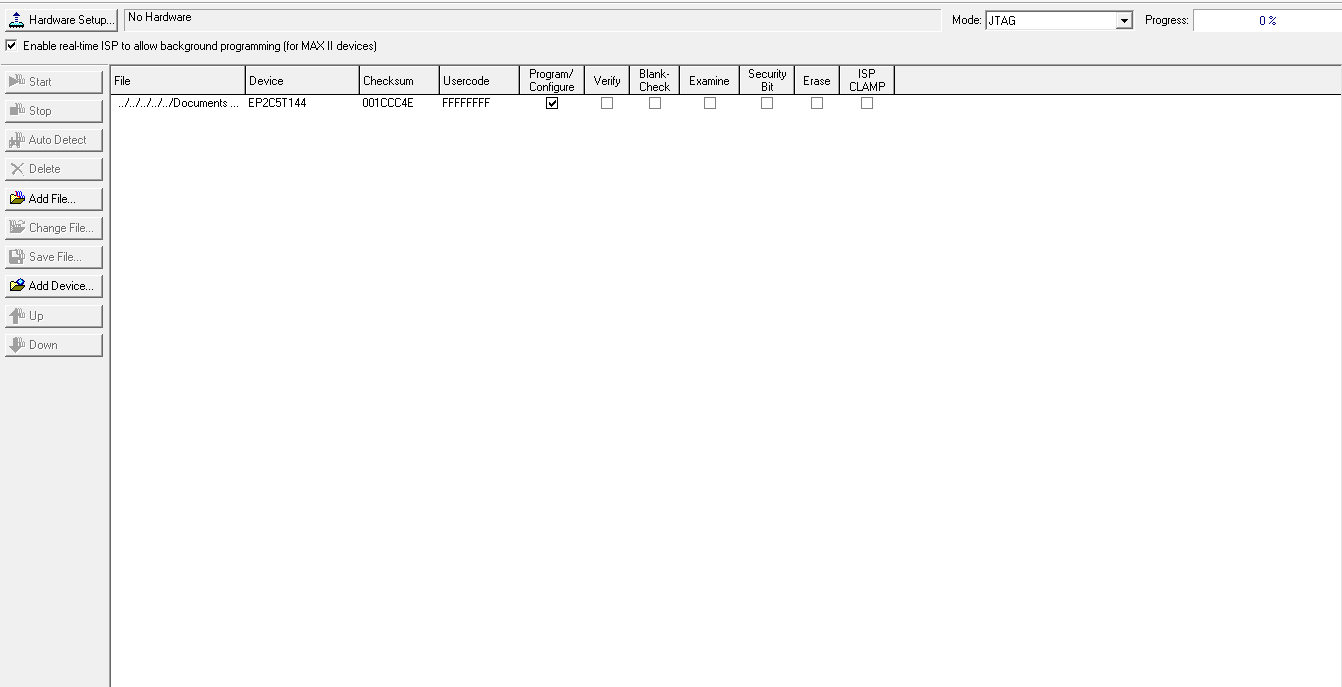


2. FPGA中编绎及下载

（1）编绎工程。(做实验时不用重新编译，直接下载)

（2）连接下载JTAG口，进行下载，点击工具条上的图标，进行下载。

（3）点Hardware Setup选择下载器，只勾上program/configure，选start下载。



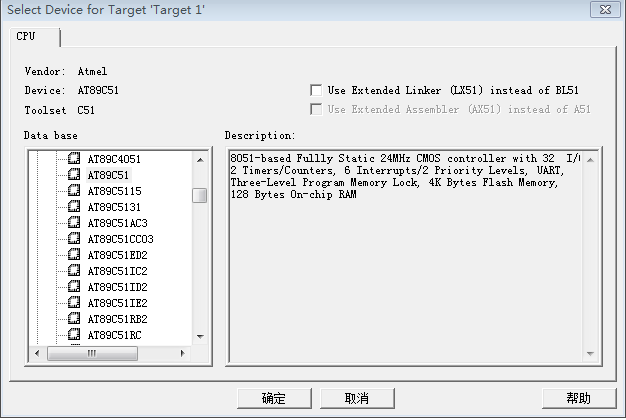
3.keil下51单片机编程

打开keil软件

点击project🡪uVersion project

在弹出的select device窗口中选择

Atmel🡪89C51



4.新建一个文档,输入对应的代码.保存到对应的工程目录中.

保存成.c文件.

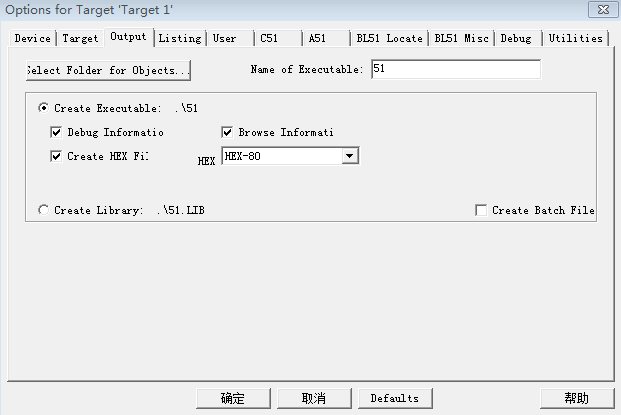
将新建的源文件添加到工程中.

右键单击Source Group 1，选择Add files to Group

Project🡪build target编译工程。

KEIL默认情况下是不生成.hex下载文件的。

需要如下设置



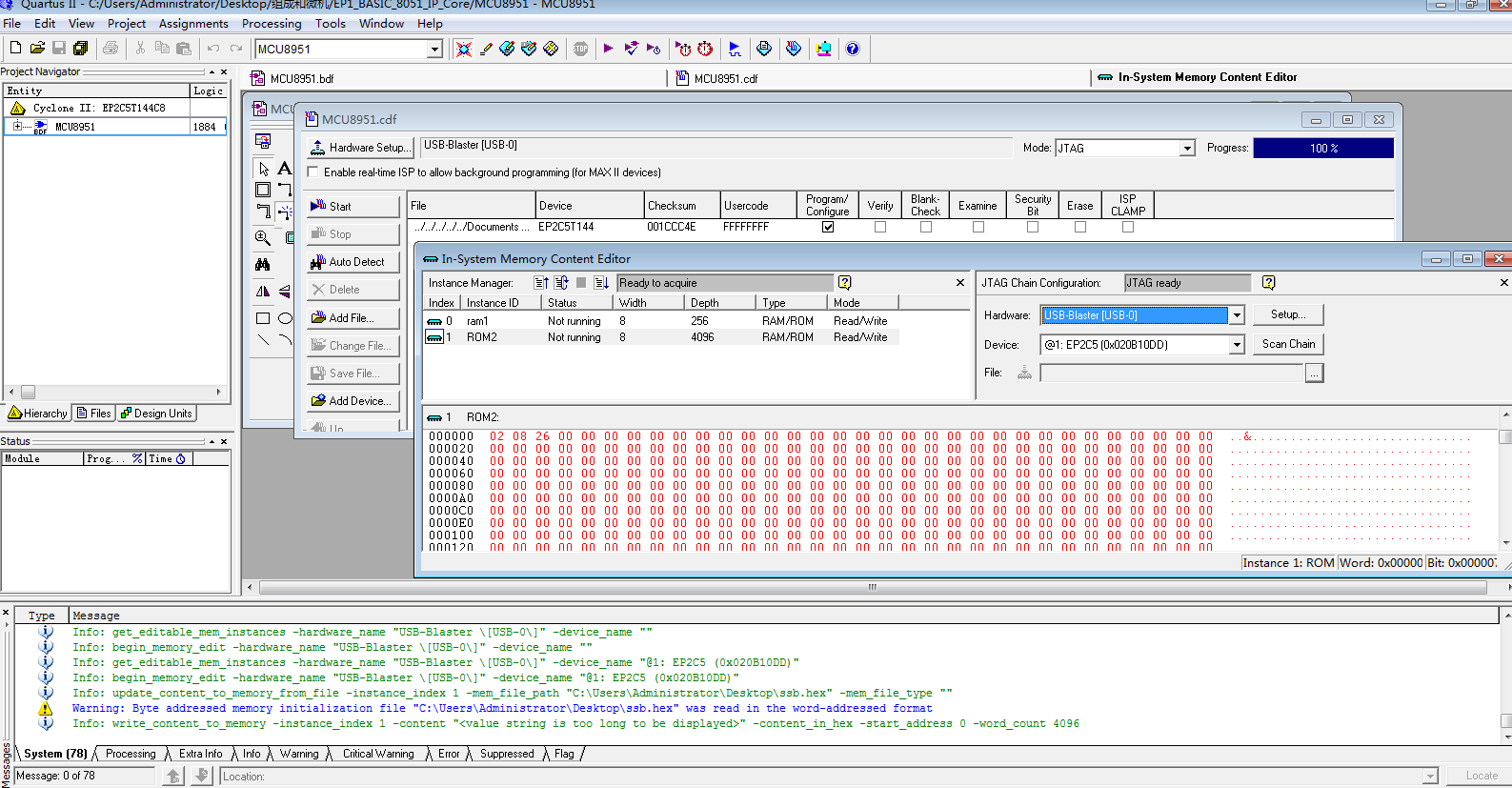
5.编绎C程序，

生成目标代码hex文件,打开Quartus II 中工具下的系统内存数据编缉器,

按窗口右侧的setup进行JTAG口查找，查找后能找到一个RAM，一个ROM。

ROM内的数据既为程序的目标代码，右击ROM选import data from file

选择生成的.hex目标文件，继续右击ROM，选WriteData.按复位键KEY1运行



6.实验代码如下：

#include<reg51.h>

sbit LED\_1=P3^6;

sbit LED\_2=P3^7;

unsigned char buffer;

void receive() interrupt 4

{

ES=0;

RI=0; //将接收中断标志清零

buffer=SBUF;

SBUF=buffer+1;

while(!TI); //将数据buffer发送出去

TI=0;

ES=1;

}

void main()

{

TMOD=0x20;//设置定时器1为工作方式2

TH1=0xd9; //设置波特率为2400

TR1=1; //启用定时器1

REN=1; //开启接收

SM0=0; //工作方式1

SM1=1;

EA=1; //CPU中断允许位

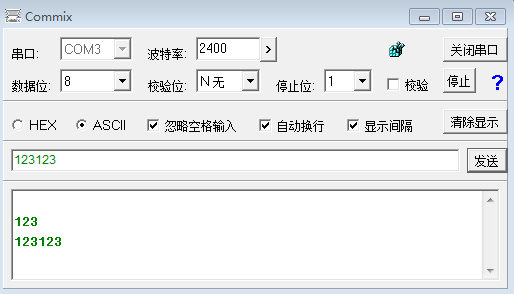
ES=1; //串行口中断允许位

}

五 实验仿真与测试

实现串口通信，输入1，显示为3，输入a，显示为63

六 实验结果分析



七 实验小结

本次实验因为相对比较复杂，所以在实验过程中表现的很是吃力，最终还是完成了实验。通过本次实验，我更加深刻的理解了串口通信的原理，学会了串口通信的实现机制。

实验五 程序计数器PC与地址寄存器AR实验

一 实验题目

FPGA中串口设计

二 实验环境

Windows7 Quartus II

三 实验要求

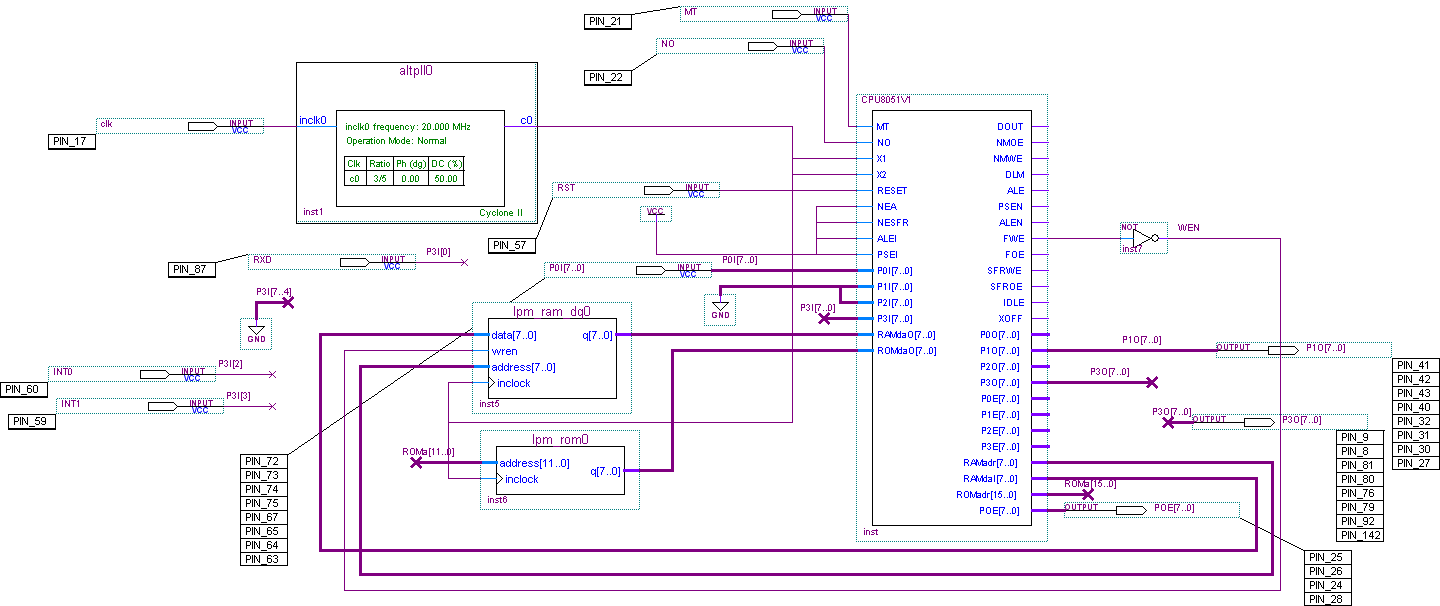
1．掌握地址单元的工作原理。

2．掌握的两种工作方式，加1计数和重装计数器初值的实现方法；

3．掌握地址寄存器从程序计数器获得数据和从内部总线获得数据 的实现方法。

1. 实验设计

1.实验原理图分析



2.地址单元主要由三部分组成：地址寄存器和多路开关。程序计数器PC用以指出下一条指令在主存中的存放地址，CPU正是根据PC的内容去存取指令的。因程序中指令是顺序执行的，所以PC有自增功能。程序计数器提供下一条程序指令的地址，如电路图4-2-1所示，在T4时钟脉冲的作用下具有自动加1的功能；在LDPC信号的作用下可以预置计数器的初值（如子程序调用或中断响应等）。当LDPC为高电平时，计数器装入data[ ]端输入的数据。aclr是计数器的清0端，高电平有效（高电平清零）；aclr为低电平时，允许计数器正常计数。

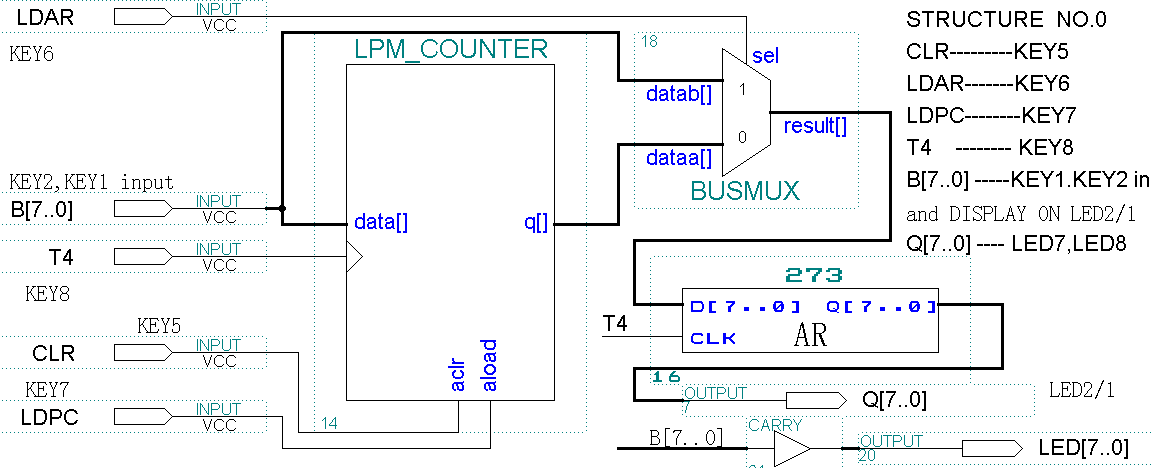


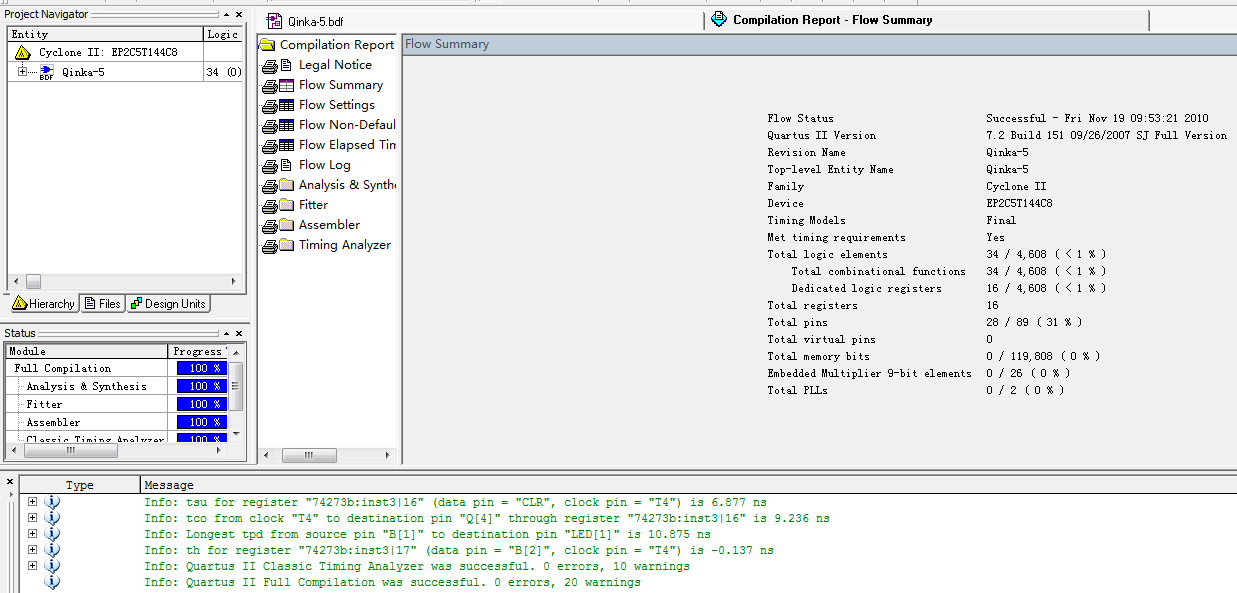
图4-2-1程序计数器原理图

3.地址寄存器AR（74273）锁存访问内存SRAM的地址。273中的地址来自两个渠道。一是程序计数器PC的输出，通常是下一条指令的地址；二是来自于内部数据总线的数据，通常是被访问操作数的地址。

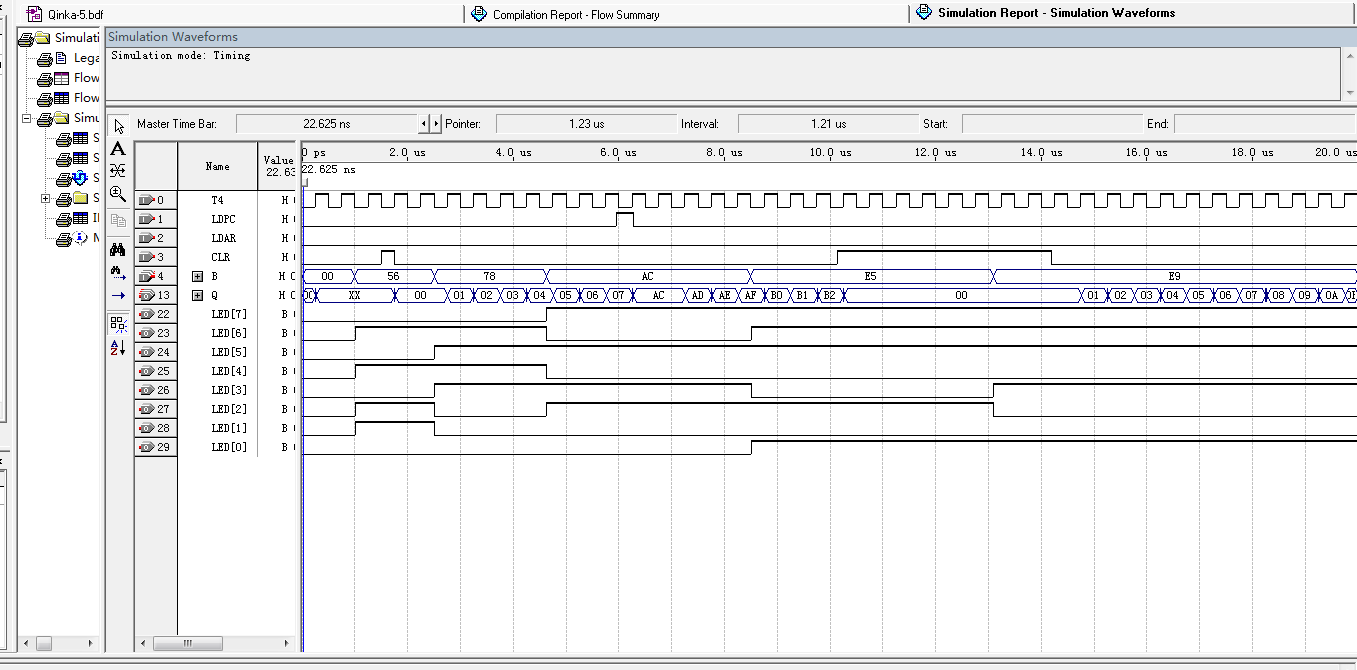
4.为了实现对两路输入数据的切换，在FPGA的内部通过总线多路开关BUSMUX进行选择。LDAR与多路选择器的sel相连，当LDAR为低电平，选择程序计数器的输出；当LDAR为高电平时，选择内部数据总线的数据。

五 实验仿真与测试

1.在成功绘制bdf实验图后，点击编译，得到如下实验结果：



2.通过对刚才编译的结果进行仿真，得实验截图如下：



1. 实验结果分析

实现对两路输入数据的切换，获得如上图的实验结果截图分析，可以看到ldpc层只有一个高电平，当LDPC=0时，观察程序计数器自动加1的功能；当LDPC=1时，观察程序计数器加载输出情况。因程序中指令是顺序执行的，所以PC有自增功能。程序计数器提供下一条程序指令的地址。

七 实验小结

通过这次实验，我初步掌握了地址单元的工作原理，了解到地址单元主要由三部分组成，地址寄存器和多路开关掌握了两种工作方式，加1计数和重装计数器初值的实现方法；掌握地址寄存器从程序计数器获得数据和从内部总线获得数据的实现方法。