计算机组成与结构专题实验

# 实验报告

# 第七次 51设计性实验

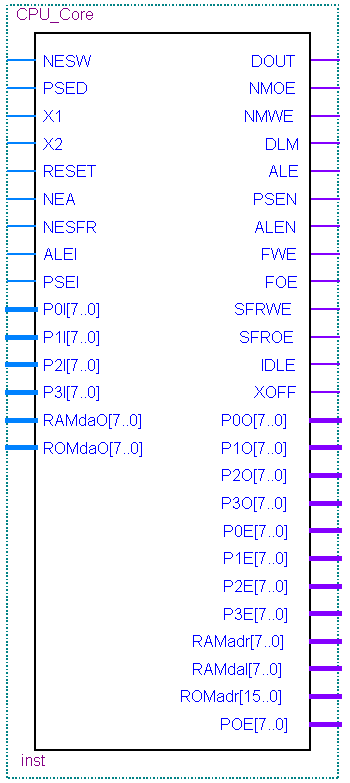
## 实验目的

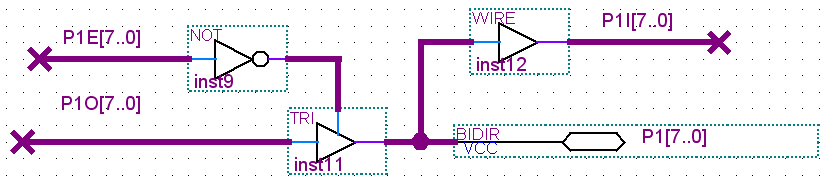
1．熟悉51单片机的原理和使用操作。2．利用51单片机，自主设计实验并完成相关功能。

## 实验原理

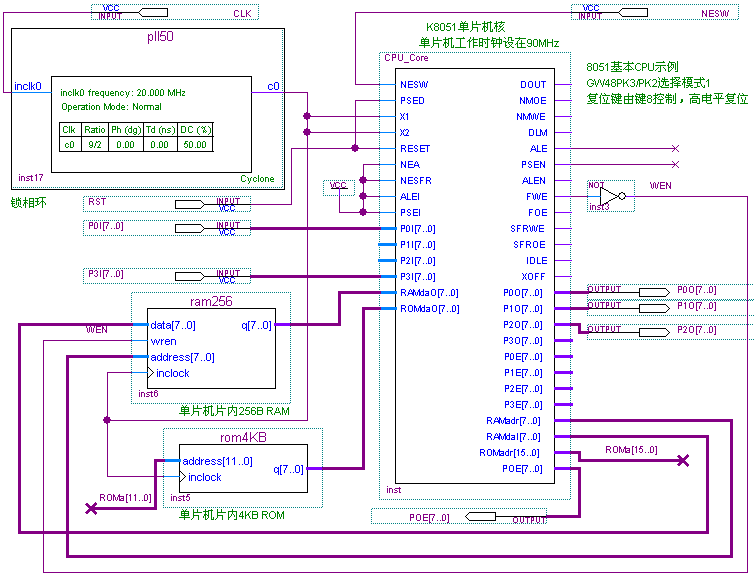
K8051单片机软核基本功能和结构如图。







K8051单片机I/O口设置成双向口的电路



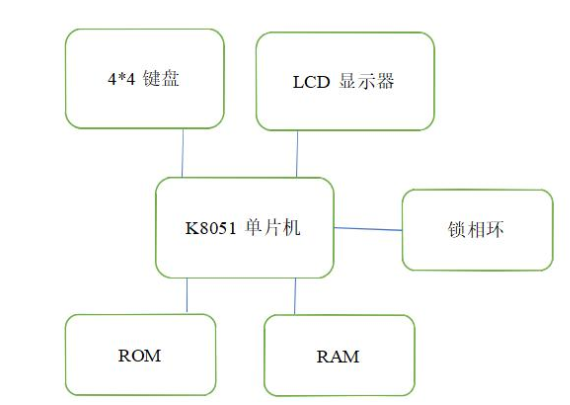
K8051基本实用电路

## 实验任务设计（具体应用描述）

经过组内商讨，我们决定使用 K8051 单片机实现一个二进制游戏， 该应用使用了 LCD 显示屏和、4\*4 键盘输入和扬声器。 游戏在一开始会出现等级划分（难， 中， 易） ， 在选择难度后， 可以进行二进制游戏。 答对一个加一分， 不同的难度有不同的个数和间隔。每次回答均会有扬声器提示，完成游戏时显示得分和总分，并有扬声器提示。

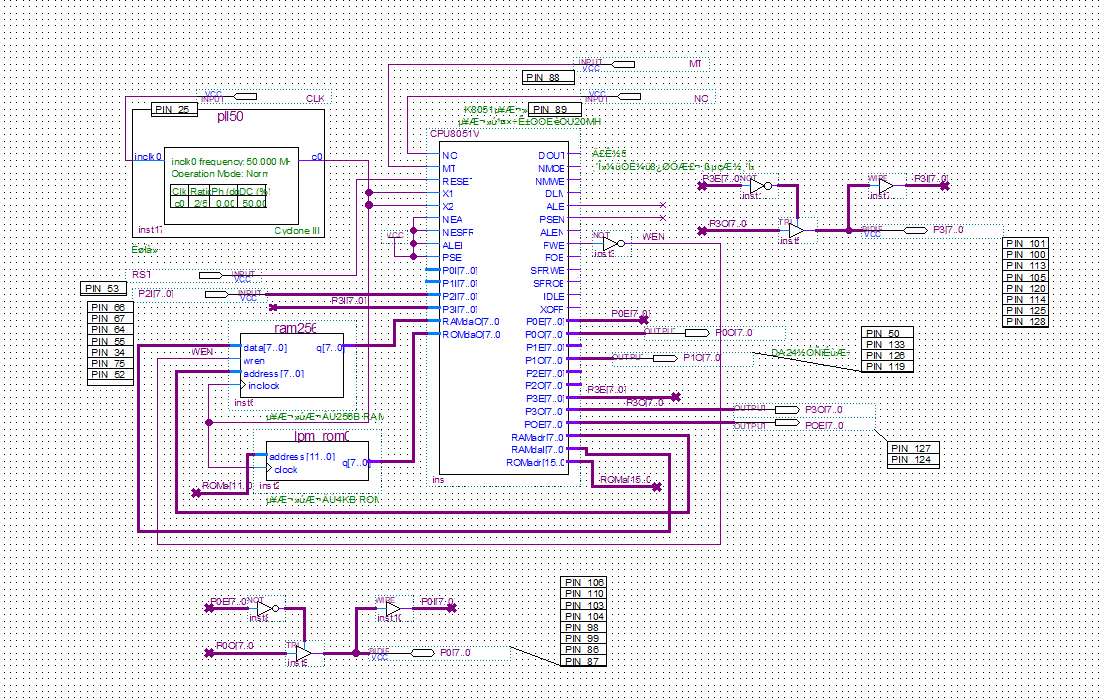
## 实验步骤及结果

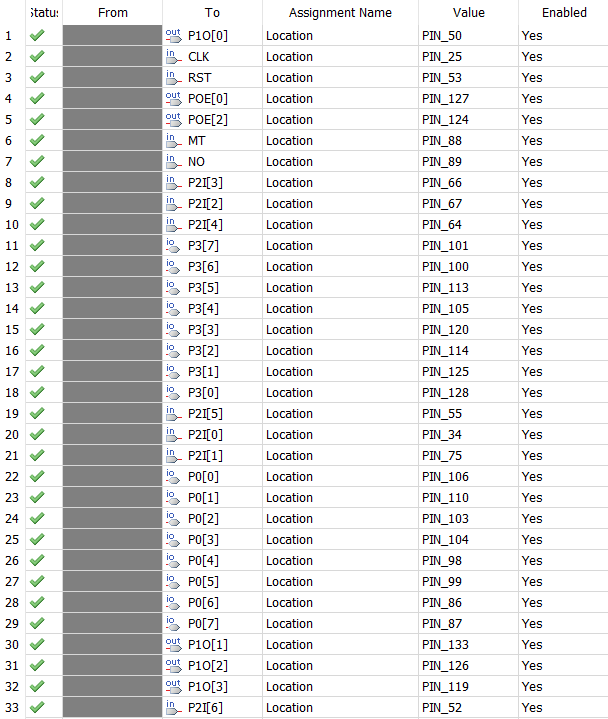
### 硬件设计框图及相关描述



ROM 内写有游戏程序的 hex 文件， 控制整个应用与元件的执行逻辑。 4x4的键盘对应 1-16输入， LCD 负责显示游戏画面以及所得分数、 难度等级。

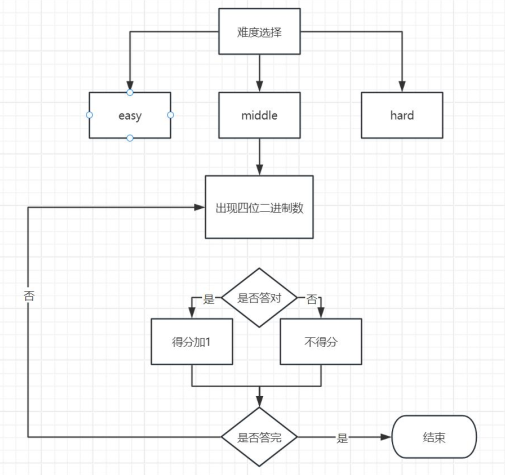
### Quartus下硬件设计原理图、模式及引脚说明





本次实验我们采取模式5，便于控制扬声器、LCD等相关元器件。在引脚绑定中，我们绑定了LCD、4\*4键盘和扬声器所需的引脚。

### 软件设计流程图及相关描述

****

软件设计上主要需要实现难度选择功能、LCD的刷新（显示）功能、按键输入判对错功能、结算页面功能和扬声器功能。

### 汇编或C语言源代码

#include <reg51.h>//51单片机编程需要的库

#include <intrins.h>//\_nop\_延时函数需要该库

#include <stdio.h>

#include <stdlib.h>

#include <string.h> //后面字符串函数中取得数组的个数中用到;调用strlen函数

#define uchar unsigned char

#define uint unsigned int

uchar code song\_tone[]= // 生日快乐歌的音符频率表，不同频率由不同的延时来决定

{

    212,212,190,212,159,169,212,212,190,212,142,159,212,212,106,126,129,169,190,119,119,126,159,142,159,0

};

uchar code song\_long[]= // 生日快乐歌节拍表，节拍决定每个音符的演奏长短

{

    9,3,12,12,12,24,9,3,12,12,12,24,9,3,12,12,12,12,12,9,3,12,12,12,24,0

};

uint code row11[22] = {0,1,1,1,0,1,1,0,1,0,0,0,0,1,1,1,1,0,1,0,0,1};

uint code row22[22] = {1,0,1,0,1,0,0,1,1,0,0,1,1,0,1,0,0,1,1,1,1,0};

uint code row33[22] = {0,0,1,1,1,1,1,1,1,1,1,0,1,1,0,1,0,0,0,0,1,1};

uint code row44[22] = {1,1,1,0,0,0,1,0,1,0,1,1,0,1,1,1,1,1,1,1,1,1};

uint code answers[22] = {5,9,15,10,6,10,11,6,15,2,3,5,6,11,13,11,9,5,13,5,7,11};

uint num;

uint interval;

uchar i;

//重新定义各个控制引脚的名称,  sbit 意思为<重定义>,,不要跟 bit 搞混,bit是汇编中 "位"

sbit lcdrs = P1^1;         //重定义,rs电平为1则传送数据,为0则转送指令

sbit lcdrw = P1^2;        //控制LCD读或者写;为1则读LCD,为0则写入LCD

sbit lcden = P1^3;        //LCD行动控制,EN为下降沿则交互执行,即EN = 1;跟着EN = 0;

//短暂延时函数

void delay\_ms(uint z)

{

  uchar x,y;

   for(x=z;x>0;x--)

    for(y=124;y>0;y--);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*读忙子程序\*\*\*\*\*\*\*\*\*\*\*\*\*

// 读忙字程序,用于判断LCD液晶是否忙状态.如果不进行判断可能会导致数据写入LCD失败.

void dbusy()                  //无返回值 函数名 (空)

{

     P0 = 0xff;                //把0xff发送给LCD的数据总线

     lcdrs = 0;               //选指令

     lcdrw = 1;               //选择读

     lcden = 1;               //使能端置1(高电平)

     while (P0 & 0x80);    //循环 (如果P0和0x80相等的话) P0 & 0x80相与,即P0总线的8位二进制数与10000000比较,

                               //全部相同则结果为 1<真>;忙状态成立,程序在此处死循环相与

     lcden = 0;               //使能端置0(低电平)

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*写数据或指令子程序\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void tcmddata (int x, unsigned char  DATA)//无返回值 函数名 (定义整形变量 x, 定义字符型变量 DATA)

{

   dbusy();

   delay\_ms(50);              //读忙子程序

   P0 = DATA;      //如果不忙,则把参数 DDATA 获取到的变量赋值给 总线<P0>

   lcdrw = 0;      //读写端选择为 写<0>

   lcdrs = x;      //数据或指令端选择为参数 <x>的值, x的值为0或者1;为0表示总线传输的是控制LCD指令,为1表示要显示的数据

   lcden = 1;      //使能端置1

   lcden = 0;      //使能端置0; 接上一条指令形成一个下降沿,LCD识别到下降沿信号则读取总线内容

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LCD初始化函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void lcd\_init()            //无返回值 函数名 (空)，这里2004的初始化和1602的初始化指令完全相同

{

   lcden=0;           //使能端EN清零，因为上电是默认高电平，所以先清零

   //显示模式设置

  tcmddata(0, 0x38);  //函数名 (指令, 指令内容)  //  0发送给tcmddata函数中的x,再经tcmddata函数发送给rs,即rs=0代表此时设置为命令方式;0x38为命令内容

                      //0x38  对1602设置16\*2，对2004就代表16\*4显示；5\*7点阵；8位数据接口

   //显示开关及光标设置

  tcmddata(0, 0x08);  //0设置命令方式，0x08  表示只开显示

   //数据控制设置

  tcmddata(0, 0x06); //0设置命令方式，0x06  表示当写入一个字符的数据后，地址指针自动加1，且光标右移，字符不动

  tcmddata(0, 0x0c);  //0设置命令方式，0x0c  表示开显示，不显示光标

    tcmddata(0, 0x01);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*定义显示位置\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void set\_xy(uchar x,uchar y)

{

   switch(x)

   {

    case 1: y = y + 0x80;break;

    case 2: y = y + 0xc0;break;

    case 3: y = y + 0x94;break;

    case 4: y = y + 0xd4;break;

    default: y = y + 0x80;//

   }

   tcmddata(0,y);//0设置命令方式，规定显示起始的行列，x行，y列

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*显示函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void display(uchar x,uchar y,uchar \*s)

{

    set\_xy(x,y);

    while(\*s)   //对于一个不定长的字符串进行显示

    {

     P0=\*s;

     tcmddata(1,\*s);

     s++;

    }

}

sbit beep  = P1^0;//蜂鸣器接P1O[0]

unsigned int FTemp;//存储当前按下电子琴键对应频率的变量

unsigned char Timer0\_H,Timer0\_L,Time;//Timer0\_H与Timer0\_L共同存储当前播放音符对应的频率,Time存储当前音符的音长

unsigned int code tab[] = {//4\*4键盘对应的音的预置数(16位) 吉他1234弦空弦+前三品

    64684,64898,65030,65157,//D, G, B, E

    64732,64934,65058,65178,//bE, #G, C, F

    64777,64968,65085,65198,//E, A, #C, #F

    64820,64994,65110,65217 //F, bB, D, G

};

unsigned char Keyscan(void){//读取当前键盘被按下的键,返回值为键的号码(0~15),没有键按下返回16

    unsigned char i, j, temp, Buffer[4] = {0xfe, 0xfd, 0xfb, 0xf7};

        for(j = 0; j < 4; j++){

            P3 = Buffer[j];

            \_nop\_();

            temp = 0x80;

            for(i = 0; i < 4; i++){

                if(!(P3 & temp)){

                    return (i + j \* 4);//i是行号,j是列号,对比按键的行、列地址，确定按下键的行和列

                }

                temp >>= 1;

            }

        }

    return 16;//如果没有键按下，返回16

}

// 播放函数

void playmusic()

{

    uint i =0,j,k;

    while(song\_long[i]!=0||song\_tone[i]!=0) //判断是否时长或者音符长度为0，为0歌曲即结束

    {

        for(j=0;j<song\_long[i]\*20;j++) //时长及音符长度不为0时，逐一播放各个音符，song\_long 为拍子长度，这里20为延时倍数，修改这个值可加快或减缓音乐的播放速度

        {                              //song\_long为拍子长度,一个节拍大概为 400ms-500ms，这里的节拍又受下面一个音符的影响，所以只能根据大概的来取值

            if(Keyscan()==0)return;

            beep = ~beep; //电平翻转的频率会产生出不同的音调的播放效果

            for(k=0;k<song\_tone[i]/5;k++); //这里的3为频率增减调节，修改该值会整体调高或者降低音调，但降低该值时，应适当加大节拍延时，反之应适当将节拍延时调小

        }

        delay\_ms(10);

        i++;

    }

}

void DelayUs2x(unsigned char t){//延时2tus

    while(--t);

}

void DelayMs(unsigned char t){//延时tms

    while(t--){

        DelayUs2x(245);

        DelayUs2x(245);

    }

}

void delay(unsigned char t){//延时0.25ts

    unsigned char i;

    for(i=0;i<t;i++){

    DelayMs(250);

    }

    TR0=0;

}

void Song(){//音乐播放

    TH0=Timer0\_H;

    TL0=Timer0\_L;

    TR0=1;

    delay(Time);//播放一个Time倍八分音符时长的音

}

void main(void){

    unsigned char Key\_Value = 16, Key\_Temp1 = 16, Key\_Temp2 = 16;//默认Key\_Value为16(键盘未按下),Key\_Temp1存放4\*4键盘按键的位置,Key\_Temp2作校验用

    int persent\_beat = -2;

    int score = 0;

    int ii;

    int present\_row;

    int correct;

    char printstr[5];

    char printstr1[10];

    char printstr2[10];

    int is\_playing = 0;

    TMOD |= 0x11;//工作模式赋值为00010001,前4位表示定时器1工作,后四位表示定时器0工作

    EA  = 1;//允许中断

    ET0 = 1;//定时器0允许中断

    IT0 = 1;//高电平触发

    ET1 = 1;//定时器1允许中断

    lcd\_init();//初始化

    while(1){//循环执行

        lcd\_init();

        if (persent\_beat > 0 && persent\_beat % interval == 0){

            if (Key\_Value == answers[persent\_beat/interval-1]){

                                    correct = 1;

                        }

                    if (correct == 0){

                            display(1,0,"F");

                        }

                        else if (correct == 1 && Key\_Value != 16){

                            display(1,0,"T");

                            score += 1;

                        }

        }

            if (is\_playing == 0) {

                display(1,0,"Binary Game");

                display(2,0,"1 for easy");

                display(3,0,"2 for medium");

                display(4,0,"3 for difficult");

                while(Key\_Temp1 == 16){//如果有键按下(不是16)

                    Key\_Temp1 = Keyscan();//读取按键的编号

                    Key\_Temp2 = Keyscan();//再读一遍进行校验

                    if (Key\_Temp1 == 1){//如果校验无误

                        is\_playing = 1;

                        num = 6;

                        interval = 3;

                    }

                    else if(Key\_Temp1 == 2){

                        is\_playing = 1;

                        num = 14;

                        interval = 2;

                    }

                    else if(Key\_Temp1 == 3){

                        is\_playing = 1;

                        num = 22;

                        interval = 1;

                    }

                    else{

                        Key\_Temp1 = 16;

                    }

                }

            }

            else {

                if (persent\_beat == -2) {

                    display(2,8,"Ready");

                    persent\_beat++;

                    delay(10);

                }

                else if (persent\_beat == -1) {

                    display(2,8,"Go");

                    persent\_beat++;

                    delay(10);

                }

                else {

                    unsigned char persent\_delay = 245;

                    unsigned char delay\_times = 15;

                    Key\_Temp1 = 16;

                    for (ii = 1; ii <num+1 ; ii++){

                        present\_row = ii\*interval-persent\_beat;

                        if (present\_row >= 1 && present\_row <= 19){

                            sprintf(printstr,"%d",row11[ii-1]);

                            display(1,present\_row,printstr);

                            sprintf(printstr,"%d",row22[ii-1]);

                            display(2,present\_row,printstr);

                            sprintf(printstr,"%d",row33[ii-1]);

                            display(3,present\_row,printstr);

                            sprintf(printstr,"%d",row44[ii-1]);

                            display(4,present\_row,printstr);

                        }

                    }

                    persent\_beat++;

                    while (Key\_Temp1 == 16 && persent\_delay) {//当没有按键被按下或没到时间

                        //Key\_Temp3 = Keyscan();//读取按键的编号

                        Key\_Value = 16;

                        correct = 0;

                        --persent\_delay;

                        if (delay\_times > 0 && persent\_delay == 1) {

                            persent\_delay = 245;

                            delay\_times--;

                        }

                        TR0=0;

                        TR1 = 0;//停止计数器TR1

                        Key\_Temp1 = Keyscan();//读取按键的编号

                        if(Key\_Temp1 != 16){//如果有键按下(不是16)

                            Key\_Temp2 = Keyscan();//再读一遍进行校验

                            if (Key\_Temp1 == Key\_Temp2){//如果校验无误

                                Key\_Value = Key\_Temp1;//按键编号赋给Key\_Value

                                FTemp = tab[Key\_Value];//对应预置数的值赋给FTemp

                                TR1 = 1;//启动计数器TR1

                                while (Keyscan() < 16);//直到松开按键

                                TR1 = 0;

                                beep = 1;//SPK置为1

                            }

                        }

                        if (persent\_beat > num\*interval+1) {

                            sprintf(printstr1,"score %d",score);

                            sprintf(printstr2,"total %d",num);

                            display(1,2,"Finish!");

                            display(2,2,printstr1);

                            display(3,2,printstr2);

                            display(4,2,"0 to restart");

                            playmusic();

                            score = 0;

                            persent\_beat = -2;

                            is\_playing = 0;

                            delay(15);

                            break;

                        }

                    }

                }

            }

    }

}

void TIM0\_ISR() interrupt 1//定时器T0中断

{

    TR0 = 0;//停止定时器T0

    beep = !beep;//变反

    TH0 = Timer0\_H;//计数器预置数高八位

    TL0 = Timer0\_L;//计数器预置数低八位

    TR0 = 1;//启动定时器T0

}

void T1\_INT(void) interrupt 3//定时器T1中断

{

    TR1 = 0;//停止定时器T1

    TL1 = FTemp;//计数器预置数低八位

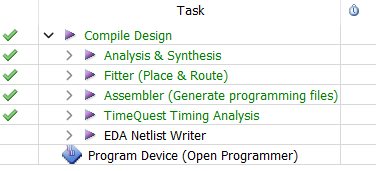
    TH1 = FTemp >> 8;//计数器预置数高八位

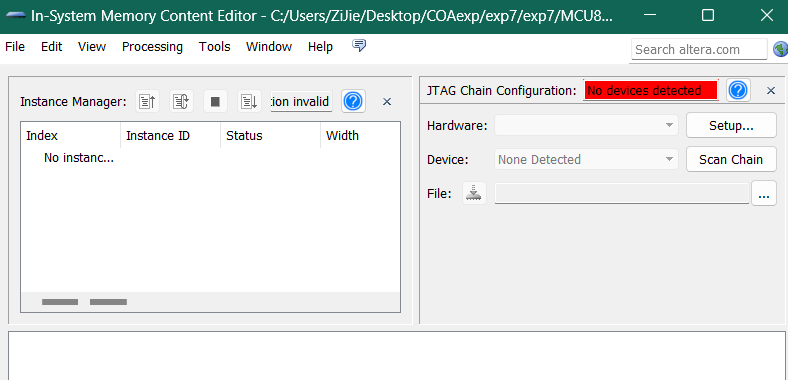
    beep = !beep;//变反

    TR1 = 1;//启动定时器T1

}

源代码主要在一个main函数中依次序实现了所需的功能。同时，函数display实现了LCD的显示功能、函数playmusic实现了音乐播放功能、Song实现了提示音功能，供main函数中调用。代码编写完成后，使用Keil软件编译为hex文件导入单片机的RAM中。





### 调试总结

经调试，可以在单片机上实现所需的二进制游戏功能。

## 实验总结及问题分析

实验基本实现了所需的功能，但是在实现过程中遇到了各种各样的问题，包括引脚绑定不对、C语言程序编写后无法正常运行等问题，求助了其他参与实验的同学，在此感谢为我们提供了帮助的所有同学。经过实验，我的软件使用技能得到提升，对计算机组成、51单片机的使用的理解得到加深，感谢李晨老师、张玉龙老师在计算机组成实验过程中提供的讲解和指导，此次实验为计算机组成实验划上了一个完美的句号。