

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

---

实验名称: 双机对战五子棋程序

## 【摘 要】

利用汇编语言编写五子棋小游戏, 利用串行通信实现双机对战, 同时在屏幕和 16\*16 LED 点阵板显示棋盘。玩家的每步限时一分钟, 允许弃权, 获胜者播放音频以示庆祝。

## 【关键词】

五子棋 双机对战 串行通信 并行通信 分频 中断控制 数码管显示 点阵板显示 音频播放

## 一、 目的要求

1. 实现双机通信的五子棋对战程序;
2. 使用制表符在屏幕显示 15\*15 的五子棋连珠棋盘;
3. 在 16\*16 LED 点阵板上显示棋盘;
4. 输入坐标控制落子, 输入两位十六进制数字后自动确认, 输入一位数字时可按退格回退;
5. 程序启动时选择本机玩家先手执黑或后手执白;
6. 每步限时一分钟, 以数码管显示倒计时, 任意一方时间到或者按 ESC 键均为自动弃权;
7. 倒计时最后十秒, 每秒钟驱动微机扬声器响铃一次;
8. 玩家落子后自动判断是否产生输赢或和棋, 是则产生提示并退出;
9. 获胜方播放《欢乐颂》以示庆祝。

## 二、 实验器材

- |                   |                                     |
|-------------------|-------------------------------------|
| 1. 微机             | 3. 软件系统                             |
| 1) 内置定时/计数器 8253  | 1) MS-DOS 7.1 (Windows 98 集成)       |
| 2) 内置并行接口芯片 8255A | 2) Microsoft Macro Assembler 5.0    |
| 3) 内置扬声器          | 3) IBM Personal Computer Linker 2.0 |
| 2. 通用微机接口实验箱      | 4) Microsoft Windows 7              |
| 1) 串行接口芯片 8251A   | 5) UltraEdit 16.10                  |
| 2) 定时/计数器 8253    | 6) DOSBox v0.74                     |
| 3) 并行接口芯片 8255A   | 7) Borland Turbo C++ 3.0            |
| 4) 中断控制器 8259     |                                     |
| 5) LED 七段数码管      |                                     |
| 6) 16*16 LED 点阵板  |                                     |

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 三、 设计和实施方案

### 1. 设计方案的选择与论述

五子棋是起源于中国古代的传统黑白棋种之一。传统五子棋的棋具与围棋相同，棋子分为黑白两色，棋盘为 15×15，棋子放置于棋盘线交叉点上。两人对局，各执一色，轮流下一子，先将横、竖或斜线的 5 个或 5 个以上同色棋子连成不间断的一排者为胜<sup>[1]</sup>。

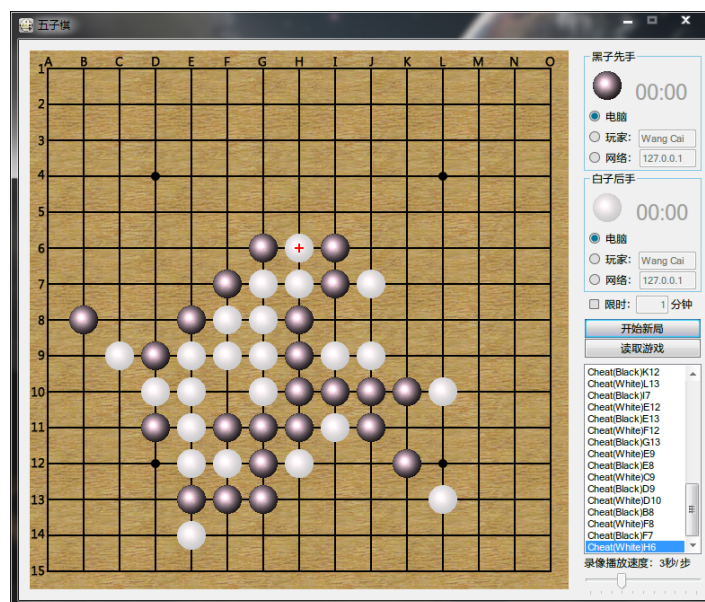


图 1 一款 Windows 平台上的五子棋游戏（白子胜）

本实验实现了五子棋游戏的双机通信对战功能，在屏幕上以制表符形式绘制棋盘，且在 16\*16LED 点阵板上显示棋盘，利用串行接口芯片 8251A 实现两台计算机之间的通讯功能。程序运行时提示选择先后手，双方轮流落子，每个玩家有一分钟思考时间，由定时/计数器 8253 产生 1Hz 的脉冲信号传至微机实验箱 IRQ 接口交给中断控制器 8259 处理，两块并行接口芯片 8255A 分别不断刷新 LED 七段数码管显示倒计时和点阵板显示棋盘。当倒计时最后十秒时，微机内置的 8253 和 8255A 会驱动内置扬声器蜂鸣报警，时间到、按 ESC 弃权、连成五子、棋盘满均会导致游戏结束，如果本机玩家获胜，程序将会驱动微机内置扬声器播放一遍《欢乐颂》。

在实验中主要用到的技术有串行通信、并行通信、中断处理、计数器分频、LED 七段数码管显示、LED 点阵板显示、微机内置扬声器发声等。

<sup>1</sup> 百度百科-五子棋 <http://baike.baidu.com/view/2697.htm>

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

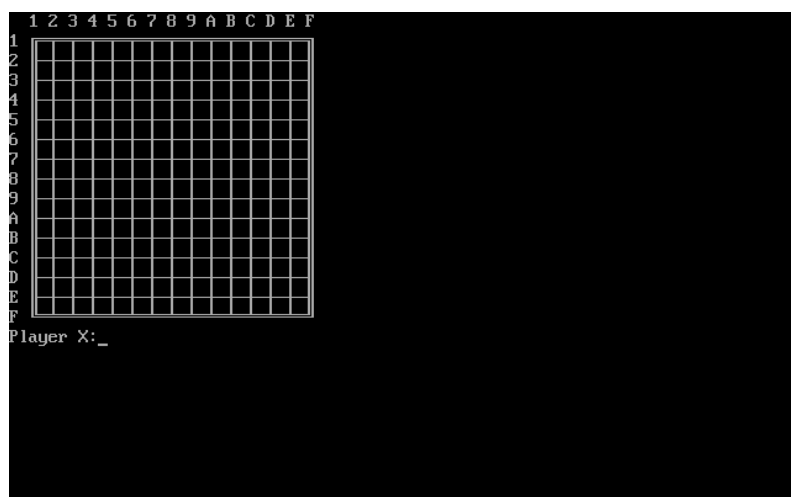


图 2 实验程序主界面 (DOSBox 下)

在最初提交自主设计实验的预习报告时,我还计划实现一定的人机智能,用纯图形界面的方式绘制棋盘,实现方向键控制光标落子。但是经过初步的调研和评估后,我认为用汇编语言实现五子棋游戏的 AI 算法过于复杂以至于难以实现;换用 C 语言等高级语言是可以实现这些算法,但是又与微机实验课程的初衷,通过汇编语言编程等方式熟悉微机工作原理和各接口芯片的使用关系不大,加之这三个星期我本人尚有四份大作业需要完成,期末考试临近,由于时间安排的原因不得不调整此次实验的最终目标,将原定的 AI 部分取消,图形界面的棋盘改成字符界面制表符“伪装图形”棋盘,控制方式也改为输坐标控制。这让我感到有一些遗憾,但是实验程序的其他主体功能均得到实现,并添加了双机通信对战以及在点阵板显示棋盘的功能。

实验中主要问题及处理方式如下:

- 1) 棋子表示:
  - a) 在屏幕上使用“X”,“O”分别代表黑子和白子;
  - b) 通过 LED 点阵板的并行接口芯片 8255A, 在 16\*16 点阵板上通过不同亮度红色点代表棋子。其中黑子颜色较亮白子较暗,这是通过点阵板不同的刷新频率刷新棋子位置 LED 实现的。
- 2) 棋盘绘制: 在绘制棋盘前先清除屏幕,之后输出列号,行号以及制表符绘制的棋盘;
- 3) 双机通信: 利用串行通信芯片 8251A, 采用查询方式发送/接收数据;
- 4) 倒计时:
  - a) 利用定时/计数器芯片 8253 产生 1Hz 脉冲信号,传至实验箱 IRQ 接口,由中断控制器 8259 接收产生中断改变剩余秒数;
  - b) 利用并行接口芯片 8255A 在七段数码管上显示倒计时。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

- 5) 判断胜负: 以落子位置为中心, 判断横/竖/斜/反斜四方向 9 个格子是否产生五子相连。
- 6) 音频播放: 使用微机内置 8255A、8253 驱动扬声器, 通过改变 8253 计数器 3 计数初值控制音频, 通过 8255A 的 B 口决定扬声器是否放音, 通过调用系统延迟中断 (AH = 86H INT15H) 控制每个音符延时长度。

## 2. 硬件资源分配及利用

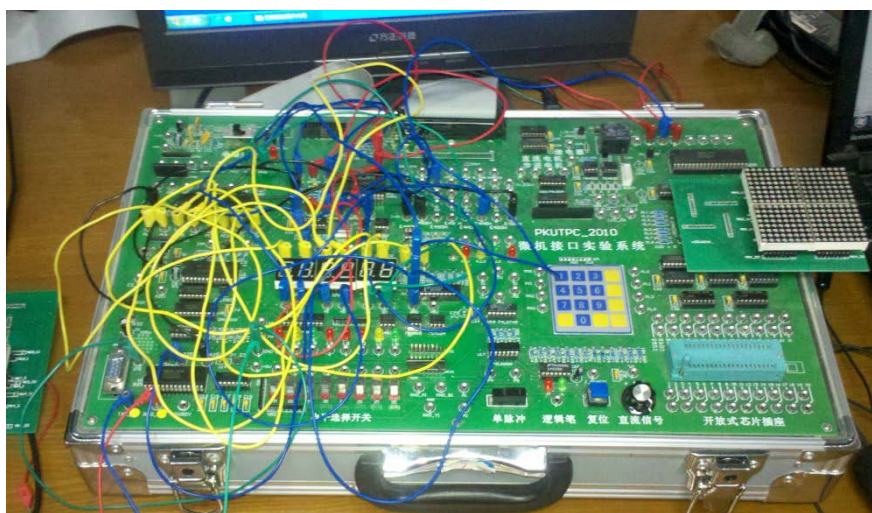


图 3 实验连线实拍

- 1) 串行接口芯片 8251A, 用于双机通信功能的实现, CS 端接 0E4B8H:

CLK 接 8253 芯片的 OUT0, CTS 接地并连接对方机器 CTS, TxD 接对方 RxD, RxD 接对方 TxD.

- a) 控制口: 地址 0E4B9H, 输出方式命令字, 操作命令字, 读取状态字:
  - i. 初始化时传入 3 个 00H
  - ii. 传入操作命令字 40H, 内部复位;
  - iii. 传入方式命令字 4EH, 1 位停止位, 无奇偶校验位, 数据字长 8 位, 异步方式, 波特率因子 16;
  - iv. 传入操作命令字 27H, 请求发送, 允许接收, 数据终端就绪, 允许发送;
  - v. 发送数据时不断读取状态字, 检查最低位是否为 1, 是 1 则可以发送;
  - vi. 接收数据时不断读取状态字, 检查次低位是否为 1, 是 1 则可以接收;
- b) 数据口: 地址 0E4B8H, 发送数据时将数据传入此端口, 读取数据时从此端口接收。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

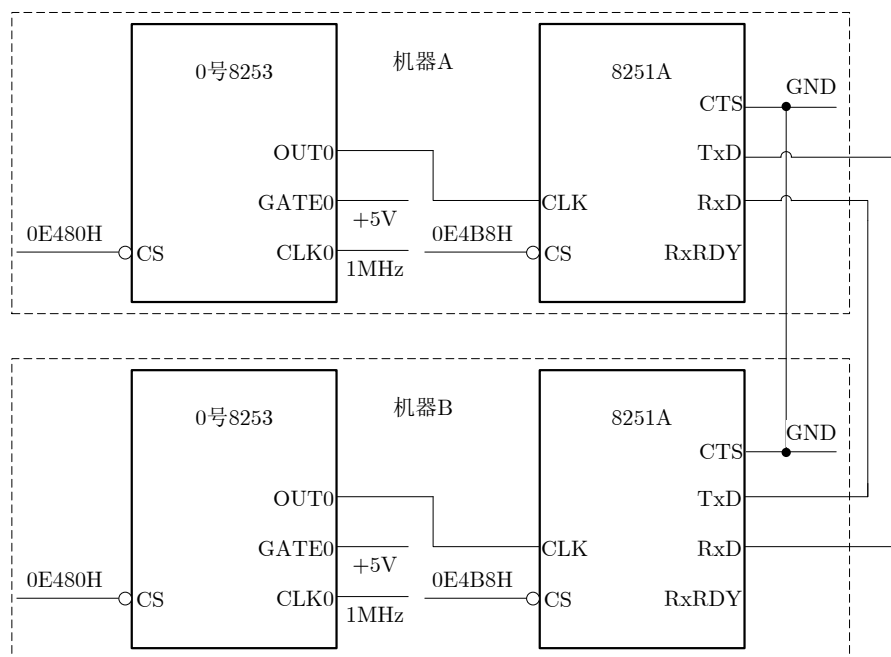


图 4 8251A 连线

## 2) 定时/计数器 8253:

a) 实验箱 8253 (0 号 8253), CS 端接 0E480H:

i. 控制口: 地址 0E483H, 输出控制字:

16H: 计数器 0, 方式 3, 二进制计数, 写低字节

76H: 计数器 1, 方式 3, 二进制计数, 先写低字节再写高字节;

B6H: 计数器 2, 方式 3, 二进制计数, 先写低字节再写高字节;

ii. 计数器 0, 为 8251A 芯片提供时钟信号:

CLK0 接 1MHz 脉冲信号, GATE0 接 5V, OUT0 接 8251A 时钟;

地址 0E480H, 写入计数初值 3AH(52D), 因为数据传输波特率 1200bps, 波特率因子 16, 频率 1MHz 的脉冲信号需要的分频倍数为:

$$n = \frac{10^6}{1200 \times 16} = 52$$

iii. 计数器 1, 将 1MHz 信号分频至 1KHz:

CLK1 接 1MHz 脉冲, GATE1 接 5V, OUT1 接计数器 2 的 CLK2, 产生 1KHz 脉冲;

地址 0E481H, 依次写入 0E8H 和 003H, 即 1000D (03E8H) 的低字节和高字节;

iv. 计数器 2, 将被分频至 1KHz 的信号再分频为 1Hz 以产生 1 秒 1 次的中断信号:

CLK2 接 OUT1, GATE2 接 5V, OUT2 接实验箱的 IRQ;

地址 0E482H, 依次写入 0E8H 和 003H, 即 1000D (03E8H) 的低字节和高字节。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

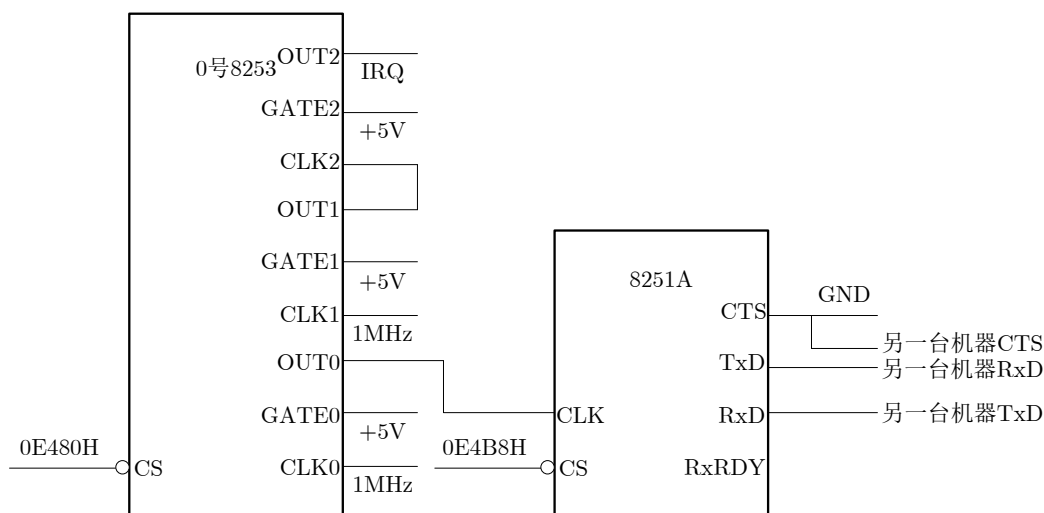


图 5 0 号 8253 连线

b) 微机内置 8253 (1 号 8253):

i. 控制口: 地址 043H, 输出控制字 B6H, 计数器 2, 方式 3, 二进制计数, 先写低字节再写高字节;

ii. 计数器 0, 1: 程序中没有用到

iii. 计数器 2: 地址 042H, 输出控制字:

倒计时最后十秒报警时先后写入 500 的低字节和高字节以产生蜂鸣报警声;

播放音频时, 对要播放的音调频率, 由 CLK2 输入频率 1.193181MHz(1234DDH)除以音调频率得到计数初值, 再先传入初值的低字节后传入高字节。

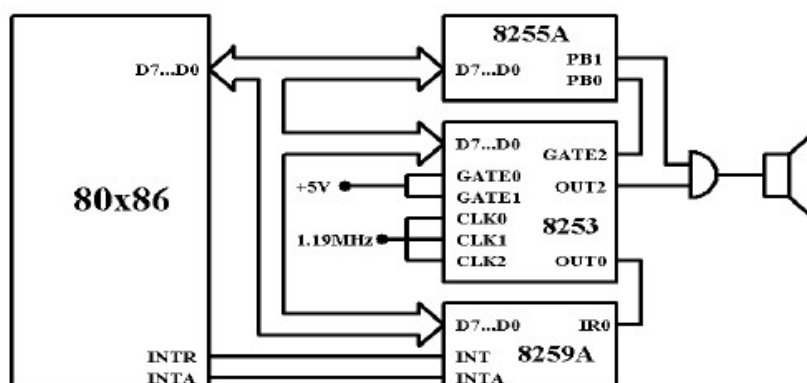


图 6 1 号 8253 与实验程序相关的硬件逻辑<sup>[1]</sup>

<sup>1</sup> 《微机原理与接口技术实验》第 66 页图 7.3, 微机内部已连好, 不需手工连线。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### 3) 并行接口芯片 8255A:

#### a) 控制数码管的 8255A (0 号 8255A), CS 端接 0E488H:

- i. 控制口: 地址 0E48BH, 输出控制字 80H, 即 A、B、C 口均输出;
- ii. A 口: 地址 0E488H, A0,A1 分别接数码管下 S0,S1 控制位码;
- iii. B 口: 地址 0E489H, 无用;
- iv. C 口: 地址 0E48AH, C0~C7 依次接数码管 a~dp, 控制段码;

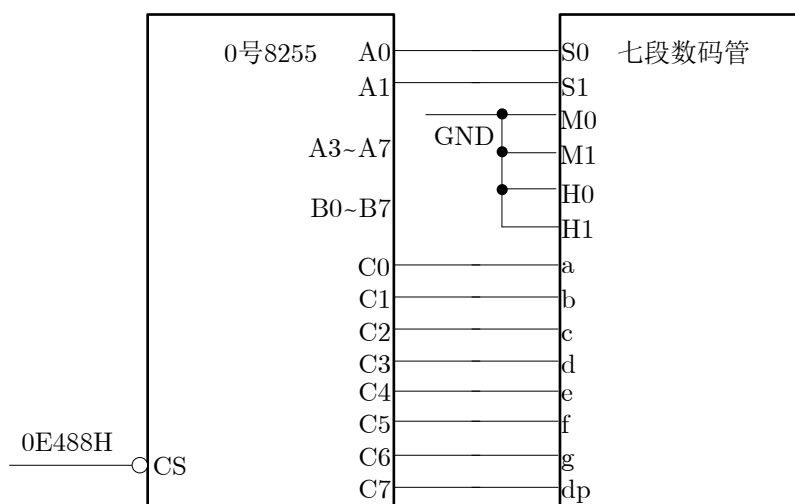


图 7 0 号 8255A 连线

#### b) 控制点阵板的 8255A (1 号 8255A), CS 端接 0E4A8H<sup>[1]</sup>:

- i. 控制口: 地址 0EABH, 输出控制字 80H, 即 A、B、C 口均输出;
- ii. A 口: 地址 0E4A8H, 控制点阵板下八行哪些行亮 (位为 1 的行亮);
- iii. B 口: 地址 0E4A9H, 控制点阵板上八行哪些行亮 (同上);
- iv. C 口: 地址 0E4AAH, 控制点阵板第几列亮 (00H~0FH);

#### c) 微机内置 8255A (2 号 8255A) <sup>[2]</sup>:

- i. 控制口: 63H, 程序中没有用到;
- ii. A 口: 60H, 程序中没有用到;
- iii. B 口: 61H, 最低位控制微机内置 8253 计数器 2 的 GATE2, 次低位与内置 8253 的 OUT2 相与连至内置扬声器, 控制是否放音;
- iv. C 口: 62H, 程序中没有用到;

<sup>1</sup> 连线图略, 将 CS 端接 0E4A8H, RESET 端接地即可。

<sup>2</sup> 连线见图 6, 同样不需手工连线。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

- 4) 中断控制器 8259, 接收 8253 产生的 1Hz 信号, 更新秒表计时:
  - a) I/O 端口地址 020H, 中断服务程序中写 OCW2,20H 为设置 IR0, 普通 EOI 方式;
  - b) I/O 端口地址 021H, 读写 OCW1:
    - i. 初始化时将次高位 (从高位开始的第二位) 置为 0 开放 IRQ6 中断;
    - ii. 退出时将次高位置为 1, 关闭 IRQ6 中断;
- 5) PCI9052:
  - a) 中断状态寄存器低 8 位, 地址 0EC4CH:
    - i. 初始化时输出为 43H, 即 PCI 中断控制允许中断, LINT1 极性为高电平有效, LINT1 有效;
    - ii. 退出时输出为 42H, 即 PCI 控制允许中断, LINT1 极性为高电平有效, LINT1 无效;
  - b) 中断状态寄存器高 8 位, 地址 0EC4DH:
    - i. 初始化时及中断服务程序均输出 1DH, 即 PCI9052 工作在 ISA 状态, 清除 LINT1, LINT2 边沿触发中断状态, LINT1 边沿触发。

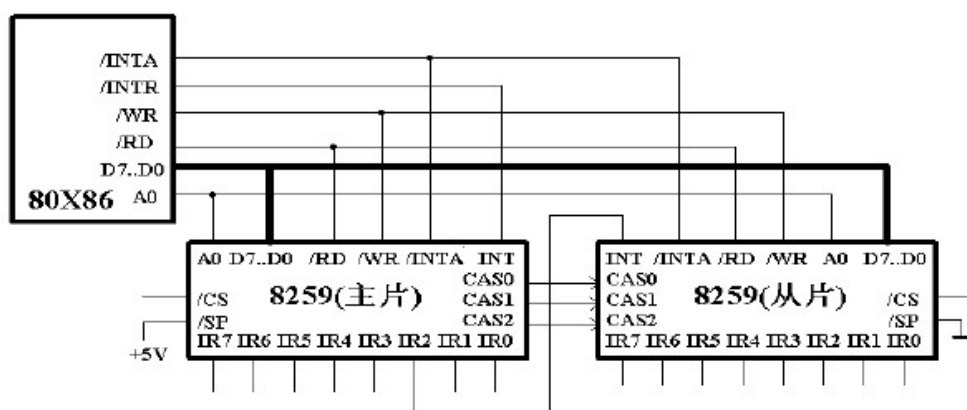


图 8 微机系统的中断结构<sup>[1]</sup>

- 6) LED 七段数码管: 接收 0 号 8255A 的输出显示倒计时, 位码端口 H0, H1, M0, M1, 接地, S0, S1 分别接 0 号 8255A 的 A0 和 A1, 段码端口 a, b, c, d, e, f, g, dp 分别接 0 号 8255A 的 C0~C7;
- 7) 16\*16 LED 点阵板: 接收 1 号 8255A 输出, 显示当前棋盘;
- 8) 微机内置扬声器: 在倒计时最后十秒钟蜂鸣报警, 在获胜后播放《欢乐颂》音频。

<sup>1</sup> 《微机原理与接口技术实验》第 60 页图 7.1



# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

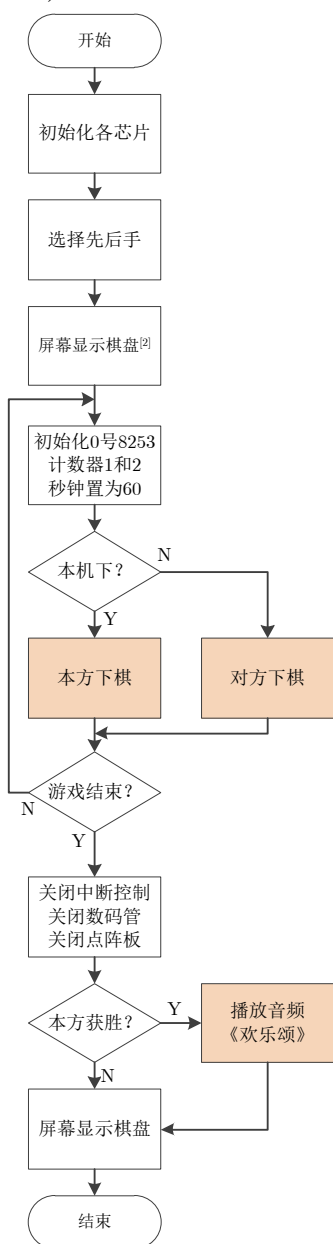
实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### 3. 主程序及各模块流程图

#### 1) 主程序:

##### a) 流程图<sup>[1]</sup>:



##### b) 相关代码:

##### 主程序:

```

START:  CALL INIT_8259    ;初始化中断控制器
        CALL INIT_8253    ;初始化定时/计数器
        CALL INIT_8251    ;初始化串行通信接口
        CALL INIT_8255    ;初始化并行通信接口
  
```

```

        CALL SELECT      ;选择先后手
        CALL PRINT_TAB    ;打印棋盘
LO_MAIN: MOV CL, 'X'      ;黑子下
        CALL INPUT        ;输入棋子
        CMP STOP, 00H     ;是否结束?
        JNZ DOS           ;结束返回 DOS

        MOV CL, 'O'       ;白子下
        CALL INPUT        ;输入棋子
        CMP STOP, 00H     ;是否结束?
        JZ LO_MAIN        ;结束返回 DOS
  
```

```

DOS:     CALL CLOSE_LED; 关闭 LED
        CALL CLOSE_8259  ;关闭中断控制器
        CMP STOP, 02H    ;检测是否获胜
        JNZ NOT_WIN      ;未获胜继续
        CALL MUSIC       ;播放欢乐颂
NOT_WIN: MOV AH, 4CH      ;返回 DOS
        INT 21H
  
```

##### 输入过程:

```

INPUT PROC NEAR          ;输入
        CALL INIT_CLOCK; 初始化时钟

        CMP CL, ME        ;比较是否为本机下
        JZ IN_ME
        CALL INPUT_ENEMY  ;对手下
        RET
IN_ME:  CALL INPUT_ME      ;自己下
        RET
INPUT ENDP
  
```

图 9 主程序流程图

<sup>1</sup> 流程图中红色部分将在后面说明其内部流程, 下同。

<sup>2</sup> 每个玩家落子后屏幕棋盘更新的过程包含在本方/对方下棋过程中落子过程内, 不再单独列出。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### 2) 本方下棋模块:

#### a) 流程图:

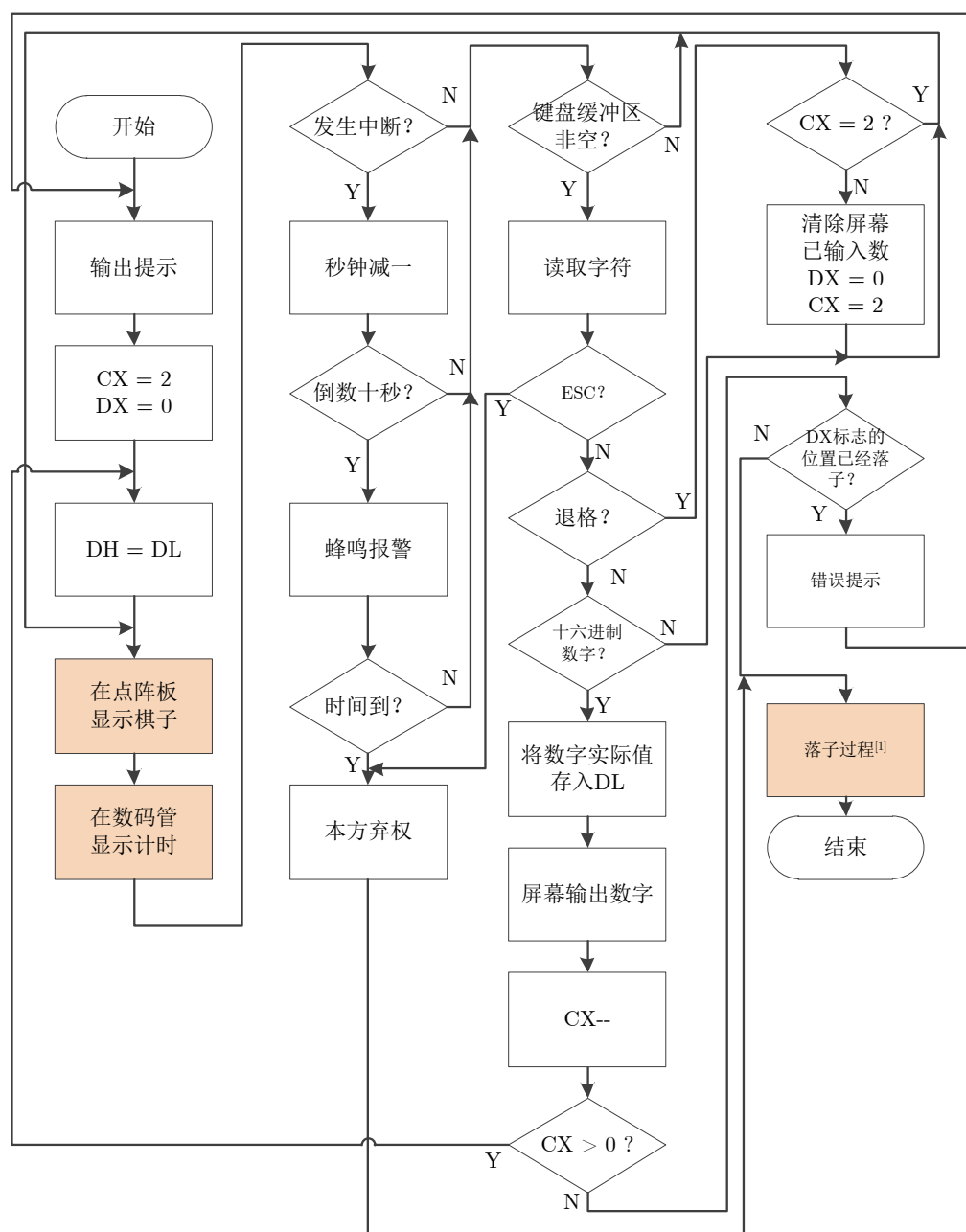


图 10 本方下棋流程图

<sup>1</sup> 落子过程包括本机玩家的发送过程、棋盘显示的更新过程、胜负及和局的计算过程，弃权视为虚拟落子 00，下同。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### b) 相关代码:

本方输入过程:

INPUT\_ME PROC NEAR ;自己下  
PUSH AX  
PUSH BX  
PUSH SI

LO\_INPUT\_ME1:  
PRINTS PLAYER\_TIP  
PRINTC CL;输出棋子  
PRINTC 3AH;输出冒号

PUSH CX  
MOV CX, 02H  
;接收两位十六进制数字  
MOV DX, 00H  
;输入的初始为 0

LO\_INPUT\_ME2:  
MOV DH, DL  
;将低 8 位移到高 8 位

LO\_INPUT\_ME3:  
CALL SHOW\_TAB  
;在点阵板显示棋子  
CALL DISP\_SEC  
;在数码管显示倒计时  
CMP DI, 01H  
;检测是否发生中断  
JNZ NOIRQ\_INPUT\_ME

MOV AL, SEC  
;秒钟减一, BCD 计数  
DEC AL  
DAS  
MOV SEC, AL

MOV DI, 00H;清除中断  
CMP SEC, 10H  
;是否是倒数 10 秒  
JA NOBEEP\_INPUT\_ME  
CALL BEEP;响铃报警

NOBEEP\_INPUT\_ME:  
CMP SEC, 00H;是否到时  
JNZ NOIRQ\_INPUT\_ME<sup>[1]</sup>  
JMP TIMEOUT\_INPUT\_ME;

超时

NOIRQ\_INPUT\_ME:  
MOV AH, 01H  
;键盘缓冲区是否有数据输入  
INT 16H

JZ LO\_INPUT\_ME3  
  
MOV AH, 07H;输入内容  
INT 21H  
  
CMP AL, 1BH;是否为 ESC  
JNZ NEXT\_INPUT\_ME0  
JMP GIVE\_UP\_INPUT\_ME  
NEXT\_INPUT\_ME0:

CMP AL, 08H;是否退格  
JZ BKSP\_INPUT\_ME  
  
CMP AL, 31H  
;是否小于 '1'  
JB LO\_INPUT\_ME3

MOV BL, AL  
  
SUB AL, 30H;减去 '0'  
CMP AL, 0AH;是否小于  
0AH  
JB NEXT\_INPUT\_ME1

SUB AL, 07H  
JB LO\_INPUT\_ME3

CMP AL, 0AH  
;是否大于 0AH  
JB LO\_INPUT\_ME3

CMP AL, 10H  
;是否小于 10H  
JB NEXT\_INPUT\_ME1

SUB AL, 20H;减去 20H  
JB LO\_INPUT\_ME3

CMP AL, 0AH  
;是否大于 0AH  
JB LO\_INPUT\_ME3

CMP AL, 0FH  
;是否小于 10H  
JA LO\_INPUT\_ME3  
JMP NEXT\_INPUT\_ME1

BKSP\_INPUT\_ME:;退格  
CMP CX, 02H;是否未输入  
JZ LO\_INPUT\_ME3

PRINTC AL  
PRINTC ' '  
PRINTC AL;覆盖掉原数

MOV DX, 00H;清除位置  
MOV CX, 02H;  
JMP LO\_INPUT\_ME3

NEXT\_INPUT\_ME1:  
MOV DL, AL;存输入位置  
PRINTC BL;打印字符

DEC CX ;计数减一  
JZ NEXT\_INPUT\_ME2  
JMP LO\_INPUT\_ME2  
NEXT\_INPUT\_ME2:  
POP CX  
PRINTLN ;换行

MOV AX, DX  
ADD AX, 0404H<sup>[2]</sup>  
CALL XLAT\_2D

CMP BYTE PTR [SI], 'X'  
;检查是否要下的地方已落子  
JZ NEXT\_INPUT\_ME3  
CMP BYTE PTR [SI], 'O'  
;检查是否要下的地方已落子  
JZ NEXT\_INPUT\_ME3

JMP RET\_INPUT\_ME  
NEXT\_INPUT\_ME3:  
PRINTS WARNING  
;已落子警告  
JMP LO\_INPUT\_ME1

TIMEOUT\_INPUT\_ME: ;超时  
PRINTS TIMEOUT  
GIVE\_UP\_INPUT\_ME: ;弃权  
POP CX  
MOV DX, 00H

RET\_INPUT\_ME:  
POP SI  
POP BX  
POP AX

CALL PUT\_CHESS;落子  
RET  
INPUT\_ME ENDP

<sup>1</sup> 最初这里只写 JZ TIMEOUT\_INPUT\_ME, 但是带条件转移不能超过 128 字节, 而 JMP 可以, 故改写为现在的形式, 下同;

<sup>2</sup> 这里加 0404H 的原因请见于其后“判断五子连模块”。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### 3) 对方下棋模块:

#### a) 流程图:

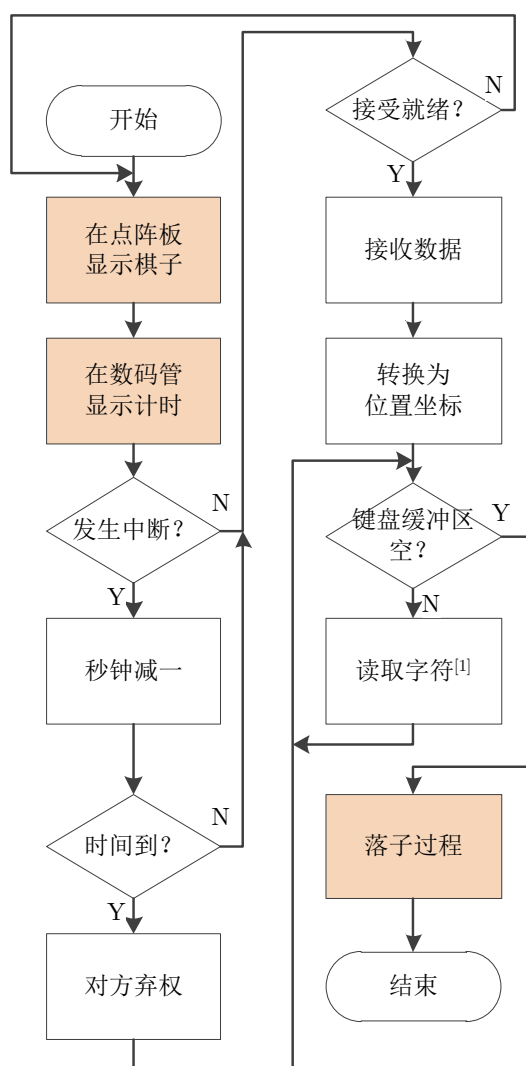


图 11 对方下棋流程图

#### b) 相关代码:

对方输入过程:

```

INPUT_ENEMY PROC NEAR    ;对手下
    CALL RECEIVE          ;调用接收过程
    CALL CLEAR_KEY        ;清键盘缓冲
    CALL PUT_CHESS        ;落子
    RET
INPUT_ENEMY ENDP
  
```

查询方式接收过程:

```

RECEIVE PROC NEAR        ;查询方式接收
    PUSH AX
    MOV DX, 0E4B9H
LO_RECEIVE:
    CALL SHOW_TAB        ;在点阵板显示棋子
    CALL DISP_SEC        ;在数码管显示计时
    CMP DI, 01H          ;是否中断
    JNZ NOIRQ_RECEIVE
    MOV AL, SEC           ;秒钟减一, BCD 计数
    DEC AL
    DAS
    MOV SEC, AL
    MOV DI, 00H           ;清除中断
    CMP SEC, 00H          ;是否超时
    JZ TIMEOUT_RECEIVE
  
```

```

NOIRQ_RECEIVE:
    IN AL, DX
    TEST AL, 02H
    JZ LO_RECEIVE         ;检查是否接受就绪
    MOV DX, 0E4B8H
    IN AL, DX             ;接收落子
    MOV DH, AL
    ;八位数字还原为十六位数字
    MOV DL, AL
    PUSH CX
    MOV CL, 04H
    SHR DH, CL
    POP CX
    AND DL, 0FH
    POP AX
    RET
  
```

```

TIMEOUT_RECEIVE:
    PRINTS TIMEOUT        ;打印超时
    MOV DX, 00H           ;落子置为弃权标志
    POP AX
    RET
RECEIVE ENDP
  
```

<sup>1</sup> 这里读入字符仅是为了清除键盘缓冲区，即在等待对方机器输入时自己误键入的字符，此字符无用。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### 4) 落子模块:

#### a) 流程图:

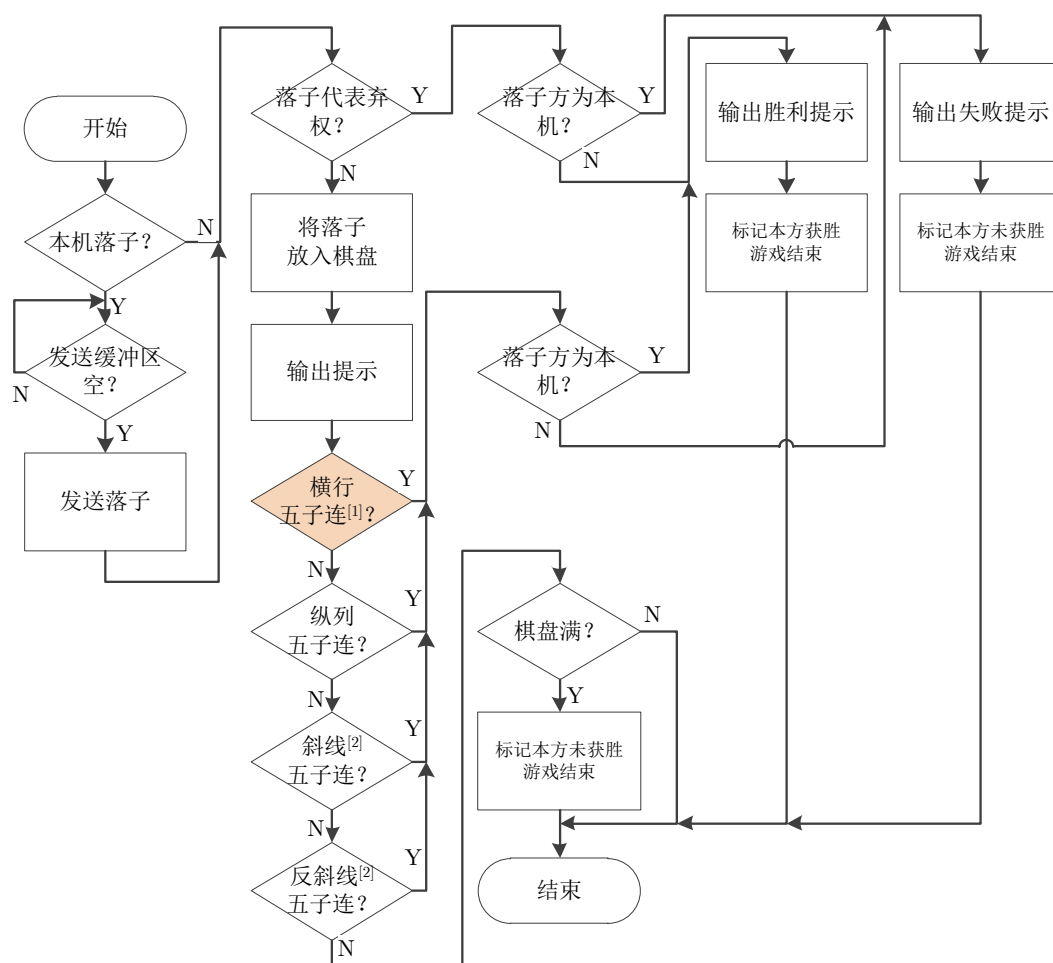


图 12 落子流程图

<sup>1</sup> 由于横行五子连、纵列五子连、斜线五子连、反斜线五子连判别方式近似，只介绍判别横行五子连；

<sup>2</sup> 本实验中斜线指平行于  $X = Y$  的直线，反斜线指平行于  $X = -Y$  的直线，下同。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### b) 相关代码:

<pre> 落子过程: PUT_CHESS PROC NEAR; 落子     CMP CL, ME     JNZ DO_NOT_SEND      CALL SEND     ;本方落子后要发送 DO_NOT_SEND:     CMP DX, 00H; 是否弃权     JZ GIVE_UP_PUT_CHESS      INC COUNT; 计数加一      PUSH AX     PUSH SI      MOV AX, DX     ADD AX, 0404H<sup>[1]</sup>     CALL XLAT_2D     MOV [SI], CL     ;将落的子放入指定位置     POP SI     POP AX      CALL PRINT_TAB     ;输出棋盘     PRINTS PLAYER_TIP     ;输出玩家提示     PRINTC CL     ;输出玩家棋子样式     PRINTC 3AH; 输出冒号     PRINTD DH     ;输出落子行号     PRINTD DL     ;输出落子列号     PRINTLN    ;输出换行      CALL CHECK_X; 检查横行     CMP LINE, 00H     JNZ ENDGAME </pre>	<pre>     CALL CHECK_Y; 检查纵列     CMP LINE, 00H     JNZ ENDGAME      CALL CHECK_XY     ;检查斜线     CMP LINE, 00H     JNZ ENDGAME      CALL CHECK_X_Y     ;检查反斜线     CMP LINE, 00H     JNZ ENDGAME     JMP NOT_ENDGAME  ENDGAME:     CMP ME, CL; 是否为自己     JZ WIN    ;是则胜     JMP LOSE  ;否则负  GIVE_UP_PUT_CHESS: ;弃权     PRINTLN     PRINTC CL     PRINTS GIVE_UP     ;输出提示     CMP ME, CL; 是否为自己     JNZ WIN    ;是则负     JMP LOSE  ;否则胜  WIN:     PRINTS YOU_WIN     MOV STOP, 02H; 设置结束     JMP RET_PUT_CHESS  LOSE:     PRINTS YOU_LOSE     MOV STOP, 01H; 设置结束     JMP RET_PUT_CHESS </pre>	<pre> NOT_ENDGAME:     CMP COUNT, 0E1H     ;是否下 225 个子     JNZ RET_PUT_CHESS      MOV STOP, 01H ; 结束     PRINTS DRAW_GAME  RET_PUT_CHESS:     RET PUT_CHESS ENDP  发送过程: SEND PROC NEAR     PUSH AX     PUSH DX      MOV DX, 0E4B9H     LO_SEND: IN AL, DX     TEST AL, 01H     ;发送缓冲区是否为空     JZ LO_SEND      POP DX      MOV AL, DH     ;十六位数字压缩为八位数字     PUSH CX     MOV CL, 04H     SHL AL, CL     POP CX      ADD AL, DL      PUSH DX     MOV DX, 0E4B8H     OUT DX, AL; 发送位置      POP DX     POP AX     RET SEND ENDP </pre>
--	---	---

<sup>1</sup> 这里加 0404H 的原因请见于其后“判断五子连模块”。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号            姓名 管 毓 清 (同组姓名           )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### 5) 判断五子连模块:

#### a) 基本思想:

以当前落子的位置为中心, 计算所在横行/纵列/斜线/反斜线四方向 9 个格子, 如果颜色同落子的颜色, 则记作 1, 颜色不同或未落子或越界则记作 0, 这样就得到了四个 9 位的数字。(实际中用 WORD 变量 LINE 保存)

将 LINE 与 1111B(1FH)相与, 如果得到结果仍是 1111B 则出现五子连, 否则 LINE 右移一位, 继续比较, 直到 LINE 为 0。

为了避免越界问题, 我将 15\*15 的连珠棋盘扩充至 23\*23。利用自己写的 XLAT\_2D 过程, 如传入 AH = 0101, 就得到扩充后棋盘的第一行第一列的地址存于 SI 中 (行列计数均以 1 开始)。所以在原棋盘中落子的位置需要在横、纵坐标上各加 4 位, 即前面的: **ADD AX, 0404H**。

#### b) 流程图, 以横行为例:

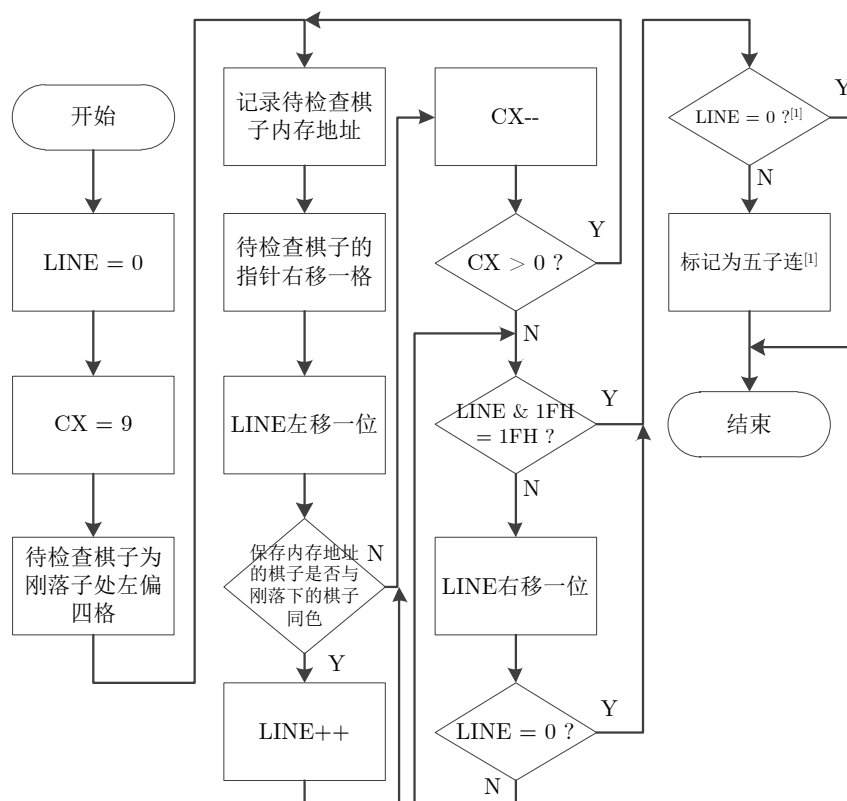


图 13 横行判断五子连流程图

<sup>1</sup> 检测 LINE = 0 并标记五子连部分在前面的 PUT\_CHESS 过程中。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### c) 相关代码:

横行检查:

```
CHECK_X PROC NEAR
    MOV LINE, 00H
    ;生成的行初设为 0
    PUSH AX
    PUSH BX
    PUSH CX
    PUSH SI

    MOV AX, DX
    ;初始列数落子位置左偏 4 格
    ADD AH, 04H;行数同一行

    MOV BX, CX;存落子颜色
    MOV CX, 09H;循环 9 次
LO_CHECK_X:
    CALL XLAT_2D;查表
    INC AL    ;右移一格
    SHL LINE, 01H
    ;生成的行左移一位
    CMP [SI], BL
    ;是否与落子同色
    JNZ NEXT_CHECK_X
    INC LINE
    ;相等则最低位加一
NEXT_CHECK_X:
    LOOP LO_CHECK_X

    POP SI
    POP CX
    POP BX
    POP AX

    CALL CHECK_LINE
    RET
CHECK_X ENDP

纵列检查:
CHECK_Y PROC NEAR
    MOV LINE, 00H

    PUSH AX
    PUSH BX
    PUSH CX
    PUSH SI

    MOV AX, DX
    ;初始行数为落子位置上 4 格
    ADD AL, 04H;列数同一列

    MOV BX, CX
    MOV CX, 09H
LO_CHECK_Y:
    CALL XLAT_2D
```

```
    INC AH
    ;检查位置下移一格,其他地方
    同横行检查
    SHL LINE, 01H
    CMP [SI], BL
    JNZ NEXT_CHECK_Y
    INC LINE
NEXT_CHECK_Y:
    LOOP LO_CHECK_Y

    POP SI
    POP CX
    POP BX
    POP AX

    CALL CHECK_LINE
    RET
CHECK_Y ENDP

斜线检查:
CHECK_XY PROC NEAR
    MOV LINE, 00H

    PUSH AX
    PUSH BX
    PUSH CX
    PUSH SI

    MOV AX, DX
    ;初始位置落子位置左上四格

    MOV BX, CX
    MOV CX, 09H
LO_CHECK_XY:
    CALL XLAT_2D
    ADD AX, 0101H ;检查位
    置右下移一格,其他地方同横
    行检查
    SHL LINE, 01H
    CMP [SI], BL
    JNZ NEXT_CHECK_XY
    INC LINE
NEXT_CHECK_XY:
    LOOP LO_CHECK_XY

    POP SI
    POP CX
    POP BX
    POP AX

    CALL CHECK_LINE
    RET
CHECK_XY ENDP

反斜线检查:
CHECK_XY PROC NEAR
```

```
    MOV LINE, 00H

    PUSH AX
    PUSH BX
    PUSH CX
    PUSH SI

    MOV AX, DX
    ;初始位置落子位置右上四格
    ADD AL, 08H

    MOV BX, CX
    MOV CX, 09H
LO_CHECK_X_Y:
    CALL XLAT_2D
    INC AH;检查位置右下移
    一格,其他地方同横行检查
    DEC AL
    SHL LINE, 01H
    CMP [SI], BL
    JNZ NEXT_CHECK_X_Y
    INC LINE
NEXT_CHECK_X_Y:
    LOOP LO_CHECK_X_Y

    POP SI
    POP CX
    POP BX
    POP AX

    CALL CHECK_LINE
    RET
CHECK_X_Y ENDP

检查 LINE 变量过程:
CHECK_LINE PROC NEAR
    PUSH AX

LO_CHECK_LINE:
    MOV AX, 1FH
    AND AX, LINE
    ;与 11111B 相与
    CMP AX, 1FH
    ;是否得到 11111B
    JZ NEXT_CHECK_LINE
    ;相等,有五子连,停止检查
    SHR LINE, 01H;右移一位
    JNZ LO_CHECK_LINE
    ;直到 LINE 为 0

NEXT_CHECK_LINE:
    POP AX
    RET
CHECK_LINE ENDP
```



# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### 6) LED 显示模块 (数码管显示倒计时模块, 点阵板显示棋盘模块):

#### a) 基本思想:

i. 数码管的显示由段码和位码共同决定, 在每一时刻所有位码为 1 的数码管均显示同样段码表示的数字, 所以需要利用视觉残留现象, 快速刷新 S0 和 S1 分别显示秒钟个位和十位。

需要注意的是, 在切换位码前需要先将段码置为 0 每一个位的所有数码管, 否则视觉残留会破坏秒钟显示效果;

ii. 16\*16 LED 点阵板可以看做一块 8255A(即上述 1 号 8255A), 每次 A 口输出的是点阵板下八行可以亮的行的集合 (对应的位为 1 则亮, 否则不亮), B 口输出的是点阵板上八行可以亮的行的集合。而 C 口决定哪一列亮, 所以同样需要利用视觉残留快速刷新点阵板。

此外, 由于点阵板只能显示红色, 为了区分黑白子, 我将白子的刷新频率设为黑子的八分之一, 这样白子相对黑子就会暗一些, 起到区分的作用。

iii. 数码管和点阵板显示的过程内部都没有循环, 因为它们在程序中已经被包含到其他过程的循环部分中了。

#### b) 流程图:

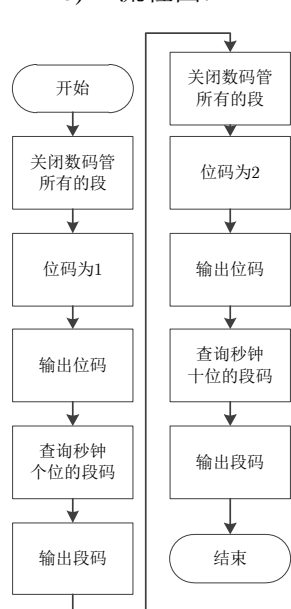


图 14 数码管模块流程图

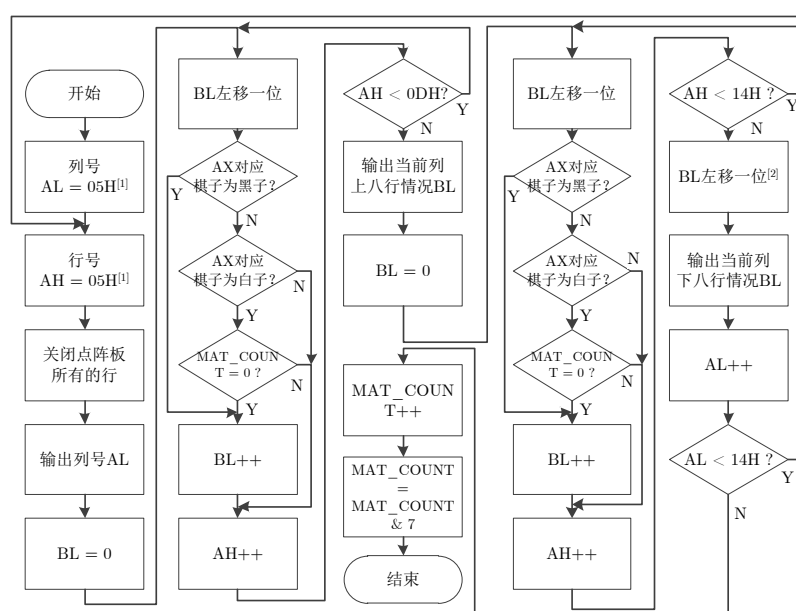


图 15 点阵板模块流程图

<sup>1</sup> 这里的 AL = 05H 即设列号实际棋盘第一列, AH = 05H 即设行号实际棋盘第一行, 依次类推;

<sup>2</sup> 由于五子棋 15 行, 下面七行比点阵板下半部八行少一行, 所以要多左移一次, 在程序中赋给 AL 后移位。

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### c) 相关代码:

数码管显示过程:

```
DISP_SEC PROC NEAR
    PUSH AX
    PUSH BX
    PUSH DX

    LEA BX, LED

    CALL CLOSE_LED
    ;关闭数码管所有段
    MOV DX, 0E488H
    MOV AL, 01H
    ;最右边数码管亮
    OUT DX, AL;输出位码

    MOV DX, 0E48AH
    MOV AL, SEC
    AND AL, 0FH
    XLAT
    ;获得秒钟个位段码
    OUT DX, AL;输出段码

    CALL CLOSE_LED
    ;关闭数码管
    MOV DX, 0E488H
    MOV AL, 02H
    ;次右边数码管亮
    OUT DX, AL
    ;输出位码
    MOV DX, 0E48AH
    MOV AL, SEC

    PUSH CX
    MOV CL, 04H
    SHR AL, CL
    POP CX

    XLAT
    ;获得秒钟十位段码
    OUT DX, AL;输出段码

    POP DX
    POP BX
    POP AX
    RET
DISP_SEC ENDP
```

点阵板显示过程:

```
SHOW_TAB PROC NEAR
    PUSH AX
    PUSH BX
    PUSH CX
    PUSH DX
    PUSH SI

    MOV AL, 05H
    ;从左上角开始落子
    LO_SHOW_TAB1:
    MOV AH, 05H
    CALL CLOSE_LED
    ;关点阵板所有行
    MOV DX, 0E4AAH
    SUB AL, 05H
    OUT DX, AL
    ;输出列号
    ADD AL, 05H

    MOV BX, 00H
    LO_SHOW_TAB2:
    SHL BX, 01H
    ;BX 左移一位
    CALL XLAT_2D

    CMP BYTE PTR [SI], 'X'
    JZ SHOW_CHESS1
    ;棋子为 X, 显示在点阵板上
    CMP BYTE PTR [SI], 'O'
    JNZ NOT_SHOW_CHESS1
    CMP MAT_COUNT, 00H
    ;棋子为 O 且 MATCOUNT 计数为
    0, 显示在点阵板上
    JNZ NOT_SHOW_CHESS1
    SHOW_CHESS1:
    INC BX
    NOT_SHOW_CHESS1:
    INC AH
    CMP AH, 0DH;前八行
    JB LO_SHOW_TAB2

    PUSH AX
    MOV DX, 0E4A9H
    ;控制点阵板上八行
    MOV AL, BL
    OUT DX, AL
    POP AX
```

```
MOV BX, 00H
LO_SHOW_TAB3:
    SHL BX, 01H
    ;BX 左移一位
    CALL XLAT_2D

    CMP BYTE PTR [SI], 'X'
    JZ SHOW_CHESS2
    ;棋子为 X, 显示在点阵板上
    CMP BYTE PTR [SI], 'O'
    JNZ NOT_SHOW_CHESS2
    CMP MAT_COUNT, 00H
    JNZ NOT_SHOW_CHESS2
    ;棋子为 O 且 MATCOUNT 计数为
    0, 显示在点阵板上
    SHOW_CHESS2:
    INC BX
    NOT_SHOW_CHESS2:
    INC AH
    CMP AH, 14H ;后七行
    JB LO_SHOW_TAB3

    PUSH AX
    MOV DX, 0E4A8H
    ;控制点阵板下八行
    MOV AL, BL
    SHL AL, 01H
    ;再左移一位
    OUT DX, AL
    POP AX

    INC AL
    CMP AL, 14H
    ;依次输出 15 个列
    JB LO_SHOW_TAB1

    INC MAT_COUNT
    AND MAT_COUNT, 07H
    ;MAT_COUNT 在 0~7 循环
    POP SI
    POP DX
    POP CX
    POP BX
    POP AX
    RET
SHOW_TAB ENDP
```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### 7) 音频播放模块:

#### a) 基本思想:

在“可编程定时/计数器与中断控制”实验时, 我们用到了微机内置的扬声器播放 1KHz 的蜂鸣声, 由于声音过于低沉, 我曾试图改变设定的发声频率。

微机内置扬声器是利用内置的 8253 计数器 2 控制其频率的, 由于计数器 2 的时钟 CLK2 接 1.19MHz 脉冲信号, 为得到 1KHz 输出信号, 必须将其分频 1190 倍。

在实验室我调了分频倍数为 500, 发声频率变为 2.38KHz, 听起来明显尖利很多, 后来我试着反汇编了 Turbo C++ 3.0 的 Sound 函数, 发现它也是利用调整微机 8253 计数器 2 分频改变发声的, 而 CLK2 的脉冲信号实际频率为  $1234DDH = 1193181Hz$ 。

通过这一发现, 我编写了利用微机扬声器播放《欢乐颂》来庆祝胜利的模块。每一个不同音符的延时是通过调用中断 AH = 86H INT 15H 实现的。

欢乐颂的频率数据:

```
FREQ DW 140H,140H,155H,180H,180H,155H,140H,120H,100H,100H,120H,140H,140H,120H,120H
      DW 140H,140H,155H,180H,180H,155H,140H,120H,100H,100H,120H,140H,120H,100H,100H
      DW 120H,120H,140H,100H,120H,140H,155H,140H,100H,120H,140H,155H,140H,120H,100H,120H,0C0H,140H
      DW 140H,140H,155H,180H,180H,155H,140H,155H,120H,100H,100H,120H,140H,120H,100H,100H
```

欢乐颂的延时数据:

```
TIME DW 04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,06H,02H,08H
      DW 04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,06H,02H,08H
      DW 04H,04H,04H,04H,04H,02H,02H,04H,04H,04H,02H,02H,04H,04H,04H,04H,04H,04H
      DW 04H,04H,04H,04H,04H,04H,04H,02H,02H,04H,04H,04H,06H,02H,08H
```

#### b) 流程图:

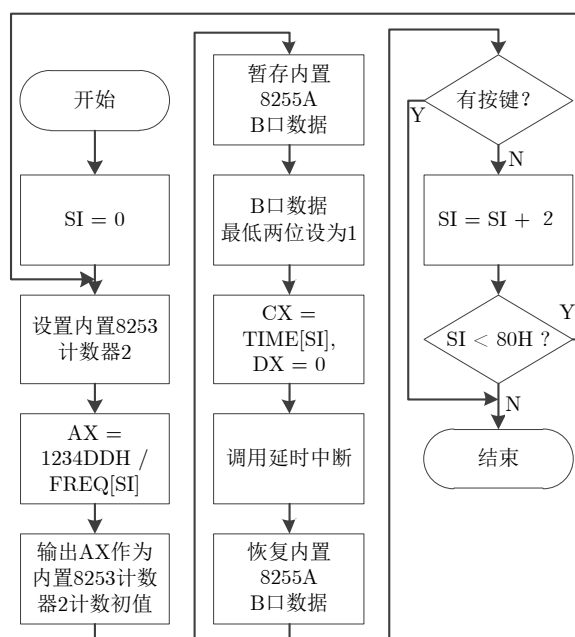


图 16 音频播放模块流程图

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### c) 相关代码:

播放整个《欢乐颂》音频过程:

```
MUSIC PROC NEAR
    PUSH SI
    MOV SI, 00H           ;第一个音符开始
LO_MUSIC:
    CALL PLAY             ;播放
    MOV AH, 01H           ;如果有按键输入结束播放
    INT 16H
    JNZ RET_MUSIC
    ADD SI, 02H           ;偏移地址加 2
    CMP SI, 80H           ;直到结束
    JB LO_MUSIC
RET_MUSIC:
    POP SI
    RET
MUSIC ENDP
```

播放一个音符过程:

```
PLAY PROC NEAR           ;播放一个音符
    PUSH AX
    PUSH BX
    PUSH CX
    PUSH DX

    MOV AL, 0B6H
    OUT 43H, AL           ;设置内置 8253 计数器 2, 方式 3, 二进制位计数, 先写低字节再写高字节
    MOV AX, 34DDH
    MOV BX, FREQ[SI]      ;获取音符频率
    MOV DX, 12H
    DIV BX                ;用 1193181Hz 除以音符频率
    OUT 42H, AL           ;输出低字节
    MOV AL, AH
    OUT 42H, AL           ;输出高字节

    IN AL, 61H            ;暂存微机 8255A 的 B 口数据
    MOV BL, AL
    OR AL, 03H            ;设置 8255A 的 B0,B1 口为 1
    OUT 61H, AL

    MOV AH, 86H           ;延时指定时间
    MOV CX, TIME[SI]
    MOV DX, 00H
    INT 15H

    MOV AL, BL            ;将暂存的原微机 8255A 之 B 口数据恢复
    OUT 61H, AL

    POP DX
    POP CX
    POP BX
    POP AX
    RET
PLAY ENDP
```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 4. 其他较重要的程序部分简介

1) 未提到的较重要的过程:

名称	功能	传入参数	返回结果	备注
BEEP	响铃			原理同 PLAY 函数
CLOSE_8259	退出时设置中断控制器			
CLOSE_LED	关闭数码管和点阵板			数码管各段均设为不亮 点阵板各行均设为不亮
CLS	清屏			
INIT_8251	初始化 8251A			
INIT_8253	初始化 0 号 8253 计数器 0			
INIT_8255	初始化各非内置 8255			
INIT_8259	初始时设置中断			
INIT_CLOCK	初始化 0 号 8253 计数器 1 和计数器 2			
PRINT_TAB	在屏幕打印棋盘			
SELECT	选择先后手		ME 设为 本方棋子	
XLAT_2D	根据坐标查找棋盘内容	AH 行号 AL 列号	SI 所要查 坐标位置对 应内存地址	行号列号均从 1 开始 查询的是扩展棋盘 (23*23)
IRQ	中断服务程序		DI = 1	需要先置中断向量

2) 较重要的数据:

名称	类型	功能
COUNT	DB	已下步数
FREQ	DW 串	《欢乐颂》各音符频率
LED	DB 串	不同数字对于七段数码管显示的样式
LINE	DW	统计当前落子为中心某方向 9 格是否出现五子连
MAT_COUNT	DB	循环计数, 使白子在点阵板刷新频率为黑子 1/8
ME	DB	本方棋子
SEC	DB	倒计时秒
STOP	DB	0 = 未停止, 1 = 停止但本方未胜, 2 = 本方胜
TAB	DB 串	棋盘 (包含制表符显示)
TIME	DW 串	《欢乐颂》各音符持续时间

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

---

实验名称: 双机对战五子棋程序

## 四、 功能及结果分析

### 1. 使用方法说明

- 1) 进入程序时提示是否先手, 输入 Y/y 为先手, 输入 N/n 为后手;
- 2) 输入两个十六进制数字构成的坐标落子, 输入一个数字时可按退格撤销, 本方输入时可按 ESC 弃权;
- 3) 七段数码管显示倒计时, 倒数十秒响铃报警, 时间到自动弃权;
- 4) 16\*16 LED 点阵板将以较亮红色显示黑子, 较暗红色显示白字;
- 5) 任意一方五子连或弃权或棋盘满均会导致游戏结束;
- 6) 若本方获胜, 播放《欢乐颂》。

### 2. 测试结果

程序可以正常运行并实现所有设计的功能。

### 3. 不足与改进

程序存在的不足和可以改进的地方有:

- 1) 双方倒计时不同步, 可能一方还差 1~2 秒另一方已显示对方弃权;
- 2) 采用文本字符界面, 用 X 和 O 代表黑子白子, 可以改为图形界面;
- 3) 输坐标交互方式不够友好, 可以改为方向键控制光标或鼠标点击(需要配置鼠标驱动);
- 4) 可以增加单机双人对战和人机对战;
- 5) 可改用 DAC0832 数模转换器驱动外置扬声器播放更逼真的《欢乐颂》(似乎有些偏题)。

## 五、 结束语

此次自主设计实验安排了三个星期供我们完成, 但是由于我个人期末 Deadline 过多, 以及其他科目考试的原因, 直到第二周周日才开始着手进行实验的调研工作。鉴于时间和实现复杂度的问题, 原有预习报告的一些实验目标被更改, 有一些我很想实现的功能(如 AI 人工智能)没有实现, 但是我最终完成了本实验报告的所有功能, 即原预习报告中的屏幕显示棋子棋盘、判断胜负、数码管显示倒计时、倒数十秒响铃报警、手动弃权、超时弃权、判断五子连、胜利者播放音频以及新增加的联机对战、点阵板显示等内容。

本次实验, 以及本学期微机实验的所有实验均由本人独立完成。在本实验项目, 以及整个微机实验课程中我学到了很多关于汇编语言、微型计算机工作原理的内容, 也锻炼了自己的动手能力。在这里我想感谢任课老师和各位同学本学期对我的帮助, 祝老师和同学们期末和假期愉快。

1000010284 管毓清

2012 年 5 月 31 日

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（1）

```

MULTIPLY MACRO X, Y                                ; 乘法宏
    PUSH AX
    PUSH BX
    PUSH DX

    MOV AX, X                                        ; 将待乘两数分别存入 AX, BX
    MOV BX, Y
    MOV DX, 00H
    MUL BX

    POP DX
    POP BX

    MOV X, AX
    POP AX
ENDM MULTIPLY

PRINTC MACRO X                                       ; 打印字符宏
    PUSH DX
    MOV DL, X                                       ; 将待打印字符的 ASCII 码存入 DL
    CALL PRINT_CHAR                                ; 调用字符打印过程
    POP DX
ENDM PRINTC

PRINTD MACRO X                                       ; 打印十六进制数字宏
    PUSH AX
    MOV AL, X                                       ; 将待打印十六进制数字存入 AL
    CALL PRINT_DIGIT                                ; 调用十六进制数字打印过程
    POP AX
ENDM PRINTD

PRINTLN MACRO X                                      ; 换行宏
    CALL PRINT_ENDL                                ; 调用换行过程
ENDM PRINTLN

PRINTS MACRO X                                       ; 打印字符串宏
    PUSH DX
    LEA DX, X                                       ; 将待打印字符串数组的地址存入 DX
    CALL PRINT_STR                                  ; 调用字符串打印过程
    POP DX
ENDM PRINTS

DATA SEGMENT                                         ; 数据段
    LED DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H ; 控制各十六进制数数码管显示
        DB 7FH, 67H, 77H, 7CH, 39H, 5EH, 79H, 71H
    SEC DB 0                                         ; 当前秒
    MAT_COUNT DB 0                                   ; 点阵计数, 决定是否显示白子

    TOP DB ' 1 2 3 4 5 6 7 8 9 A B C D E F', 0DH, 0AH, '$'; 棋盘的列号
    TAB DB 05CH DUP(00H)                            ; 棋盘
        DB 004H DUP(00H), 0C9H, 00DH DUP(0D1H), 0BBH, 004H DUP(00H)
        DB 00DH DUP(004H DUP(00H)), 0C7H, 00DH DUP(0C5H), 0B6H, 004H DUP(00H))

```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

## 实验名称: 双机对战五子棋程序

### 附录: 实验程序源代码 (2)

```

        DB 004H DUP(00H), 0C8H, 00DH DUP(0CFH), 0BCH, 004H DUP(00H)
        DB 05CH DUP(00H)
LINE DW 0
;统计当前落子处所在横/纵/斜/反斜的 9 个格子是否出现五子连

ME DB 0                                ;本方棋子
COUNT DB 0                           ;一共下的步数
STOP DB 0                              ;是否停止, 0 代表不停止, 1 代表平局或负, 2 代表获胜

;一些提示信息
YOU_WIN DB 0DH, 0AH, 'You win!', 0DH, 0AH, '$'
YOU_LOSE DB 0DH, 0AH, 'You lose!', 0DH, 0AH, '$'
DRAW_GAME DB 0DH, 0AH, 'Draw game!', 0DH, 0AH, '$'
GIVE_UP DB ' Give up.$'

WELCOME DB 'Welcome! Are you the first player?(Y/N)', 0DH, 0AH, '$'
WARNING DB 'Invalid input! Please try again.', 0DH, 0AH, '$'
TIMEOUT DB 0DH, 0AH, 'Timeout.$'

PLAYER_TIP DB 'Player $'

;欢乐颂每个音符的频率
FREQ DW 140H,140H,155H,180H,180H,155H,140H,120H,100H,100H,120H,140H,140H,120H,120H
      DW 140H,140H,155H,180H,180H,155H,140H,120H,100H,100H,120H,140H,120H,100H,100H
      DW 120H,120H,140H,100H,120H,140H,155H,140H,100H,120H,140H,155H,140H,120H,100H,120H,0C0H,140H
      DW 140H,140H,155H,180H,180H,155H,140H,155H,120H,100H,100H,120H,140H,120H,100H,100H

;欢乐颂每个音符持续时间
TIME DW 04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,06H,02H,08H
      DW 04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,06H,02H,08H
      DW 04H,04H,04H,04H,04H,02H,02H,04H,04H,04H,02H,02H,04H,04H,04H,04H,04H,04H
      DW 04H,04H,04H,04H,04H,04H,04H,02H,02H,04H,04H,04H,04H,06H,02H,08H
DATA ENDS

STACK SEGMENT STACK
STACK ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:
    CALL INIT_8259                ;初始化中断控制器
    CALL INIT_8253                ;初始化定时/计数器
    CALL INIT_8251                ;初始化串行通信接口
    CALL INIT_8255                ;初始化并行通信接口

    CALL SELECT                   ;选择先后手
    CALL PRINT_TAB               ;打印棋盘
LO_MAIN:
    MOV CL, 'X'                  ;黑子下
    CALL INPUT                   ;输入棋子
    CMP STOP, 00H               ;是否结束?
    JNZ DOS                      ;结束返回 DOS

```



# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（3）

```

MOV CL, 'O'                ;白子下
CALL INPUT                 ;输入棋子
CMP STOP, 00H              ;是否结束?
JZ LO_MAIN                 ;结束返回 DOS

DOS: CALL CLOSE_LED        ;关闭 LED
CALL CLOSE_8259            ;关闭中断控制器
CMP STOP, 02H              ;检测是否获胜
JNZ NOT_WIN                ;未获胜继续
CALL MUSIC                 ;播放欢乐颂

NOT_WIN:
MOV AH, 4CH                ;返回 DOS
INT 21H

BEEP PROC NEAR             ;响铃，调用微机内置 8253、8255A
PUSH AX
PUSH BX
PUSH CX

MOV BL, 0FFH               ;循环次数
MOV AL, 0B6H
;设置内置 8253 计数器 2，方式 3，二进制位计数，先写低字节再写高字节
OUT 43H, AL

MOV AX, 500
OUT 42H, AL                ;输出计数初值低字节
MOV AL, AH
OUT 42H, AL                ;输出计数初值高字节

IN AL, 61H                 ;暂存微机 8255A 的 B 口数据
MOV AH, AL
OR AL, 03H                 ;设置 8255A 的 B0,B1 口为 1
OUT 61H, AL

MOV CX, 00H
LO_BEEP:
LOOP LO_BEEP               ;循环 256 次
DEC BL
JNZ LO_BEEP                ;循环 255*256 次

MOV AL, AH                 ;将暂存的原微机 8255A 之 B 口数据恢复
OUT 61H, AL

POP CX
POP BX
POP AX
RET
BEEP ENDP

CHECK_LINE PROC NEAR      ;检查五子棋生成的一行是否有五子连

```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（4）

```

    PUSH AX
LO_CHECK_LINE:
    MOV AX, 1FH
    AND AX, LINE                ;与 11111B 相与
    CMP AX, 1FH                 ;是否得到 11111B
    JZ NEXT_CHECK_LINE         ;相等, 有五子连, 停止检查
    SHR LINE, 01H              ;LINE 右移一位
    JNZ LO_CHECK_LINE;         直到 LINE 为 0
NEXT_CHECK_LINE:
    POP AX
    RET
CHECK_LINE ENDP

CHECK_X PROC NEAR                ;检查横行
    MOV LINE, 00H                ;生成的行初设为 0

    PUSH AX
    PUSH BX
    PUSH CX
    PUSH SI

    MOV AX, DX                    ;初始列数为落子位置左偏 4 格
    ADD AH, 04H                  ;行数为同一行

    MOV BX, CX                    ;暂存落子颜色
    MOV CX, 09H                  ;循环 9 次
LO_CHECK_X:
    CALL XLAT_2D                  ;查表
    INC AL                        ;右移一格
    SHL LINE, 01H                ;生成的行左移一位
    CMP [SI], BL                  ;是否与落子同色
    JNZ NEXT_CHECK_X
    INC LINE                      ;相等则最低位加一
NEXT_CHECK_X:
    LOOP LO_CHECK_X

    POP SI
    POP CX
    POP BX
    POP AX

    CALL CHECK_LINE
    RET
CHECK_X ENDP

CHECK_Y PROC NEAR                ;检查纵列
    MOV LINE, 00H                ;生成的行初设为 0

    PUSH AX
    PUSH BX
    PUSH CX
    PUSH SI

```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（5）

```

MOV AX, DX                                ;初始行数为落子位置上 4 格
ADD AL, 04H                                ;列数为落子同一列

MOV BX, CX
MOV CX, 09H
LO_CHECK_Y:
CALL XLAT_2D
INC AH                                      ;检查位置下移一格，其他地方同横行检查
SHL LINE, 01H
CMP [SI], BL
JNZ NEXT_CHECK_Y
INC LINE
NEXT_CHECK_Y:
LOOP LO_CHECK_Y

POP SI
POP CX
POP BX
POP AX

CALL CHECK_LINE
RET
CHECK_Y ENDP

CHECK_XY PROC NEAR                        ;检查斜线
MOV LINE, 00H                            ;生成的行初设为 0

PUSH AX
PUSH BX
PUSH CX
PUSH SI

MOV AX, DX                                ;初始位置在落子位置左上四格

MOV BX, CX
MOV CX, 09H
LO_CHECK_XY:
CALL XLAT_2D
ADD AX, 0101H                            ;检查位置右下移一格，其他地方同横行检查
SHL LINE, 01H
CMP [SI], BL
JNZ NEXT_CHECK_XY
INC LINE
NEXT_CHECK_XY:
LOOP LO_CHECK_XY

POP SI
POP CX
POP BX
POP AX

CALL CHECK_LINE
RET
CHECK_XY ENDP

CHECK_X_Y PROC NEAR                      ;检查反斜线

```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（6）

```

MOV LINE, 00H                                ;生成的行初设为 0

PUSH AX
PUSH BX
PUSH CX
PUSH SI

MOV AX, DX                                    ;初始位置在落子位置右上四格
ADD AL, 08H

MOV BX, CX
MOV CX, 09H
LO_CHECK_X_Y:
CALL XLAT_2D
INC AH                                        ;检查位置右下移一格，其他地方同横行检查
DEC AL
SHL LINE, 01H
CMP [SI], BL
JNZ NEXT_CHECK_X_Y
INC LINE
NEXT_CHECK_X_Y:
LOOP LO_CHECK_X_Y

POP SI
POP CX
POP BX
POP AX

CALL CHECK_LINE
RET
CHECK_X_Y ENDP

CLEAR_KEY PROC NEAR                          ;清空键盘缓冲
PUSH AX
LO_CLEAR_KEY:
MOV AH, 01H
INT 16H
JZ RET_CLEAR_KEY
MOV AH, 00H
INT 16H
JMP LO_CLEAR_KEY
RET_CLEAR_KEY:
POP AX
RET
CLEAR_KEY ENDP

CLOSE_8259 PROC NEAR                          ;退出 8259 中断控制器
PUSH AX
PUSH DX

IN AL, 21H
OR AL, 40H
OUT 21H, AL
;关闭 IRQ6

MOV DX, 0EC4CH
MOV AL, 42H
;关闭 PCI9052 中断

```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（7）

```

    OUT DX, AL
    STI

    POP DX
    POP AX
    RET
CLOSE_8259 ENDP

CLOSE_LED PROC NEAR                                ; 关闭 LED(数码管和点阵)
    PUSH AX
    PUSH DX

    MOV DX, 0E4A8H                                  ; 关闭点阵下八行显示
    MOV AL, 00H
    OUT DX, AL

    MOV DX, 0E4A9H                                  ; 关闭点阵上八行显示
    MOV AL, 00H
    OUT DX, AL

    MOV DX, 0E48AH                                  ; 关闭所有数码管
    MOV AL, 00H
    OUT DX, AL

    POP DX
    POP AX
    RET
CLOSE_LED ENDP

CLS PROC NEAR                                       ; 清屏
    PUSH AX
    PUSH BX
    PUSH CX
    PUSH DX

    MOV AH, 02H
    MOV BH, 00H
    MOV DX, 00H
    INT 10H                                         ; 改变光标位置至左上角

    MOV CX, 07D0H
LO_CLS:
    PRINTC ' '
    LOOP LO_CLS

    MOV AH, 02H
    MOV BH, 00H
    MOV DX, 00H
    INT 10H                                         ; 改变光标位置至左上角

    POP DX
    POP CX
    POP BX
    POP AX
    RET
CLS ENDP

```



# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（9）

```

MOV AL, 4EH
;方式字：异步方式，1 位停止位，8 位数据位，无奇偶校验位，波特率因子 16
OUT DX, AL
MOV AL, 27H                      ;命令字，请求发送，允许发送和接收，数据终端准备好
OUT DX, AL

POP DX
POP AX
RET
INIT_8251 ENDP

INIT_8253 PROC NEAR              ;初始化 8253
PUSH AX
PUSH DX

MOV DX, 0E483H                  ;计数器 0，方式 3，二进制计数，写低字节
MOV AL, 16H
OUT DX, AL

MOV DX, 0E480H
MOV AL, 3AH                      ;计数初值 52
OUT DX, AL

POP DX
POP AX
RET
INIT_8253 ENDP

INIT_8255 PROC NEAR              ;初始化 8255A
PUSH AX
PUSH DX

MOV DX, 0E48BH                  ;8255A 芯片各口设置输出
MOV AL, 80H
OUT DX, AL

MOV DX, 0E4ABH                  ;LED 点阵板 8255A 各口设置输出
MOV AL, 80H
OUT DX, AL

POP DX
POP AX
RET
INIT_8255 ENDP

INIT_8259 PROC NEAR              ;初始化 8259
PUSH AX
PUSH DX

MOV AX, CS
MOV DS, AX
MOV DX, OFFSET IRQ
MOV AX, 250EH
INT 21H                          ;置中断向量
CLI

```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（10）

```

MOV DX, 0EC4CH
MOV AL, 43H                ;开中断
OUT DX, AL

INC DX
MOV AL, 1DH                ;清除中断标志
OUT DX, AL

IN AL, 21H
AND AL, 0BFH               ;开 IRQ6 中断
OUT 21H, AL

MOV AX, DATA
MOV DS, AX
STI

POP DX
POP AX
RET
INIT_8259 ENDP

INIT_CLOCK PROC NEAR      ;初始化时钟的 8253
    PUSH AX
    PUSH DX

    MOV SEC, 60H           ;时间 60 秒

    MOV DX, 0E483H         ;计数器 1，方式 3，二进制计数，先写低字节再写高字节
    MOV AL, 76H
    OUT DX, AL

    MOV DX, 0E481H
    MOV AL, 0E8H           ;1000 的十六进制低字节
    OUT DX, AL

    MOV DX, 0E481H
    MOV AL, 03H           ;1000 的十六进制高字节
    OUT DX, AL

    MOV DX, 0E483H         ;计数器 2，方式 3，二进制计数，先写低字节再写高字节
    MOV AL, 0B6H
    OUT DX, AL

    MOV DX, 0E482H
    MOV AL, 0E8H
    OUT DX, AL

    MOV DX, 0E482H
    MOV AL, 03H
    OUT DX, AL

    MOV DI, 00H           ;未发生中断

    POP DX
    POP AX
    RET

```



# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号            姓名 管毓清 (同组姓名           )

实验日期 2012 年 5 月 18 日至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（11）

[illegible]

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（12）

```

CALL BEEP                                ;响铃报警

NOBEEP_INPUT_ME:
    CMP SEC, 00H                          ;是否到时
    JNZ NOIRQ_INPUT_ME
    JMP TIMEOUT_INPUT_ME                  ;跳至超时
NOIRQ_INPUT_ME:
    MOV AH, 01H                          ;检测键盘缓冲区是否有数据输入
    INT 16H
    JZ LO_INPUT_ME3

    MOV AH, 07H                          ;输入内容
    INT 21H

    CMP AL, 1BH                          ;是否为 ESC
    JNZ NEXT_INPUT_ME0
    JMP GIVE_UP_INPUT_ME                  ;ESC 弃权
NEXT_INPUT_ME0:
    CMP AL, 08H                          ;是否为退格
    JZ BKSP_INPUT_ME

    CMP AL, 31H                          ;是否小于 '1'
    JB LO_INPUT_ME3

    MOV BL, AL

    SUB AL, 30H                          ;减去 '0'
    CMP AL, 0AH                          ;是否小于 0AH
    JB NEXT_INPUT_ME1

    SUB AL, 07H
    JB LO_INPUT_ME3

    CMP AL, 0AH                          ;是否大于 0AH
    JB LO_INPUT_ME3

    CMP AL, 10H                          ;是否小于 10H (大写字母)
    JB NEXT_INPUT_ME1

    SUB AL, 20H                          ;减去 20H
    JB LO_INPUT_ME3

    CMP AL, 0AH                          ;是否大于 0AH
    JB LO_INPUT_ME3

    CMP AL, 0FH                          ;是否小于 10H (小写字母)
    JA LO_INPUT_ME3
    JMP NEXT_INPUT_ME1

BKSP_INPUT_ME:
    CMP CX, 02H                          ;是否未输入
    JZ LO_INPUT_ME3

```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（13）

```

PRINTC AL
PRINTC ' '
PRINTC AL                                ;退格覆盖掉原数

MOV DX, 00H                             ;清除位置
MOV CX, 02H                             ;已输入数重为 0
JMP LO_INPUT_ME3

NEXT_INPUT_ME1:
MOV DL, AL                              ;存入输入的位置
PRINTC BL                               ;打印字符

DEC CX                                  ;计数减一
JZ NEXT_INPUT_ME2
JMP LO_INPUT_ME2

NEXT_INPUT_ME2:
POP CX
PRINTLN                                ;换行

MOV AX, DX
ADD AX, 0404H
CALL XLAT_2D

CMP BYTE PTR [SI], 'X'                  ;检查是否要下的地方已落子
JZ NEXT_INPUT_ME3
CMP BYTE PTR [SI], 'O'                  ;检查是否要下的地方已落子
JZ NEXT_INPUT_ME3

JMP RET_INPUT_ME
NEXT_INPUT_ME3:
PRINTS WARNING                          ;已落子则警告
JMP LO_INPUT_ME1

TIMEOUT_INPUT_ME:                       ;超时
PRINTS TIMEOUT

GIVE_UP_INPUT_ME:                       ;弃权
POP CX
MOV DX, 00H

RET_INPUT_ME:
POP SI
POP BX
POP AX

CALL PUT_CHESS                          ;落子
RET
INPUT_ME ENDP

MUSIC PROC NEAR                         ;播放《欢乐颂》
PUSH SI
MOV SI, 00H                             ;第一个音符开始
LO_MUSIC:

```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（14）

```

CALL PLAY                                ;播放

MOV AH, 01H                              ;如果有按键输入结束播放
INT 16H
JNZ RET_MUSIC

ADD SI, 02H                              ;偏移地址加 2

CMP SI, 80H                              ;直到结束
JB LO_MUSIC
RET_MUSIC:
POP SI
RET
MUSIC ENDP

PLAY PROC NEAR                            ;播放一个音符
PUSH AX
PUSH BX
PUSH CX
PUSH DX

MOV AL, 0B6H
;设置内置 8253 计数器 2, 方式 3, 二进制位计数, 先写低字节再写高字节
OUT 43H, AL

MOV AX, 34DDH
MOV BX, FREQ[SI]                          ;获取音符频率
MOV DX, 12H
DIV BX                                    ;用 1193181Hz 除以音符频率

OUT 42H, AL                              ;输出低字节
MOV AL, AH
OUT 42H, AL                              ;输出高字节

IN AL, 61H                              ;暂存微机 8255A 的 B 口数据
MOV BL, AL
OR AL, 03H                              ;设置 8255A 的 B0,B1 口为 1
OUT 61H, AL

MOV AH, 86H                              ;延时指定时间
MOV CX, TIME[SI]
MOV DX, 00H
INT 15H

MOV AL, BL                              ;将暂存的原微机 8255A 之 B 口数据恢复
OUT 61H, AL

POP DX
POP CX
POP BX
POP AX
RET
PLAY ENDP

```

## 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录: 实验程序源代码 (15)

```
PRINT_CHAR PROC NEAR                                ;打印字符
    PUSH AX
    MOV AH, 02H
    INT 21H                                           ;输出 DL 存储字符
    POP AX
    RET
PRINT_CHAR ENDP

PRINT_DIGIT PROC NEAR                                ;打印十六进制数字
    PUSH DX
    MOV DL, AL
    CALL TO_ASC                                       ;将 DL 中数字转为 ASCII 码
    CALL PRINT_CHAR                                   ;打印字符
    POP DX
    RET
PRINT_DIGIT ENDP

PRINT_ENDL PROC NEAR                                 ;换行
    PUSH AX
    PUSH DX
    MOV AH, 02H

    MOV DL, 0DH                                       ;回车符
    INT 21H

    MOV DL, 0AH                                       ;换行符
    INT 21H

    POP DX
    POP AX
    RET
PRINT_ENDL ENDP

PRINT_STR PROC NEAR                                  ;打印字符串
    PUSH AX
    MOV AH, 09H                                       ;打印 DX 中的字符串
    INT 21H
    POP AX
    RET
PRINT_STR ENDP

PRINT_TAB PROC NEAR                                  ;打印棋盘
    PUSH AX
    PUSH CX
    PUSH DX
    PUSH SI

    CALL CLS                                           ;清屏
    PRINTS TOP                                         ;打印列号

    MOV AL, 00H
    LEA SI, TAB
    ADD SI, 57H
LO_TAB0:
    INC AL
```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（16）

```

        ADD SI, 08H
        CALL PRINT_DIGIT                ;打印行号

        MOV CX, 0FH
LO_TAB1:
        INC SI

        CMP CX, 0FH
        JNZ NEXT_TAB0

        PRINTC ' '
        JMP NEXT_TAB2
NEXT_TAB0:
        CMP AL, 01H
        JZ NEXT_TAB1
        CMP AL, 0FH
        JZ NEXT_TAB1

        PRINTC 0C4H
        JMP NEXT_TAB2
NEXT_TAB1:
        PRINTC 0CDH
NEXT_TAB2:
        PRINTC [SI]                    ;打印棋盘内容

        LOOP LO_TAB1

        PRINTLN                        ;换行
        CMP AL, 0FH                    ;一共十五行
        JNZ LO_TAB0

        POP SI
        POP DX
        POP CX
        POP AX
        RET
PRINT_TAB ENDP

PUT_CHESS PROC NEAR                  ;落子
        CMP CL, ME
        JNZ DO_NOT_SEND

        CALL SEND                    ;本方落子后要发送
DO_NOT_SEND:
        CMP DX, 00H                    ;是否弃权
        JZ GIVE_UP_PUT_CHESS

        INC COUNT                    ;计数加一

        PUSH AX
        PUSH SI

        MOV AX, DX
        ADD AX, 0404H
        CALL XLAT_2D
        MOV [SI], CL                    ;将落的子放入指定位置

```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（17）

```

POP SI
POP AX

CALL PRINT_TAB                ;输出棋盘

PRINTS PLAYER_TIP            ;输出玩家提示
PRINTC CL                    ;输出玩家棋子样式
PRINTC 3AH                   ;输出冒号
PRINTD DH                    ;输出落子行号
PRINTD DL                    ;输出落子列号
PRINTLN                      ;输出换行

CALL CHECK_X                 ;检查横行
CMP LINE, 00H
JNZ ENDGAME

CALL CHECK_Y                 ;检查纵列
CMP LINE, 00H
JNZ ENDGAME

CALL CHECK_XY                ;检查斜线
CMP LINE, 00H
JNZ ENDGAME

CALL CHECK_X_Y               ;检查反斜线
CMP LINE, 00H
JNZ ENDGAME

JMP NOT_ENDGAME
ENDGAME:
CMP ME, CL                   ;是否为自己
JZ WIN                       ;是则胜
JMP LOSE                     ;否则负

GIVE_UP_PUT_CHESS:          ;弃权
PRINTLN
PRINTC CL
PRINTS GIVE_UP              ;输出提示

CMP ME, CL                   ;是否为自己
JNZ WIN                      ;是则负
JMP LOSE                     ;否则胜

WIN: PRINTS YOU_WIN
MOV STOP, 02H                ;设置结束
JMP RET_PUT_CHESS

LOSE: PRINTS YOU_LOSE
MOV STOP, 01H                ;设置结束
JMP RET_PUT_CHESS

```

## 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录: 实验程序源代码 (18)

```
NOT_ENDGAME:
    CMP COUNT, 0E1H                ;是否下 225 个子
    JNZ RET_PUT_CHESS

    MOV STOP, 01H                  ;结束
    PRINTS DRAW_GAME

RET_PUT_CHESS:
    RET
PUT_CHESS ENDP

READ_CHAR PROC NEAR                ;读取按键, 存于 AL
    PUSH BX
    MOV BH, AH

    MOV AH, 01H
    INT 21H

    MOV AH, BH
    POP BX
    RET
READ_CHAR ENDP

RECEIVE PROC NEAR                  ;查询方式接收
    PUSH AX
    MOV DX, 0E4B9H
LO_RECEIVE:
    CALL SHOW_TAB                  ;在点阵板显示棋子
    CALL DISP_SEC                  ;在数码管显示倒计时
    CMP DI, 01H                    ;是否中断
    JNZ NOIRQ_RECEIVE

    MOV AL, SEC                     ;秒钟减一, BCD 计数
    DEC AL
    DAS
    MOV SEC, AL

    MOV DI, 00H                     ;清除中断

    CMP SEC, 00H                    ;是否超时
    JZ TIMEOUT_RECEIVE
NOIRQ_RECEIVE:
    IN AL, DX
    TEST AL, 02H
    JZ LO_RECEIVE                  ;检查是否接受就绪

    MOV DX, 0E4B8H
    IN AL, DX                       ;接收落子

    MOV DH, AL
    MOV DL, AL                       ;将八位数字还原为十六位数字

    PUSH CX
    MOV CL, 04H
    SHR DH, CL
```



# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（19）

```

    POP CX

    AND DL, 0FH

    POP AX
    RET
TIMEOUT_RECEIVE:
    PRINTS TIMEOUT                ;打印超时
    MOV DX, 00H                  ;落子置为弃权标志

    POP AX
    RET
RECEIVE ENDP

SELECT PROC NEAR                ;选择先后手
    PUSH AX

    PRINTS WELCOME
LO_SELECT:
    CALL READ_CHAR
    PRINTLN

    CMP AL, 'Y'
    JZ FIRST
    CMP AL, 'y'
    JZ FIRST                    ;先手

    CMP AL, 'N'
    JZ SECOND
    CMP AL, 'n'
    JZ SECOND                  ;后手

    PRINTS WARNING                ;警告后重试
    JMP LO_SELECT
FIRST:
    MOV ME, 'X'                  ;先手棋子为'X'
    POP AX
    RET
SECOND:
    MOV ME, 'O'                  ;后手棋子为'O'
    POP AX
    RET
SELECT ENDP

SEND PROC NEAR
    PUSH AX
    PUSH DX

    MOV DX, 0E4B9H
LO_SEND: IN AL, DX
    TEST AL, 01H                ;发送缓冲区是否为空
    JZ LO_SEND

    POP DX

    MOV AL, DH                  ;将十六位数字压缩为八位数字

```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录：实验程序源代码（20）

```

    PUSH CX
    MOV CL, 04H
    SHL AL, CL
    POP CX

    ADD AL, DL

    PUSH DX
    MOV DX, 0E4B8H
    OUT DX, AL                ;发送位置

    POP DX
    POP AX
    RET
SEND ENDP

SHOW_TAB PROC NEAR                ;点阵板显示棋盘
    PUSH AX
    PUSH BX
    PUSH CX
    PUSH DX
    PUSH SI

    MOV AL, 05H                ;从左上角开始落子
LO_SHOW_TAB1:
    MOV AH, 05H
    CALL CLOSE_LED            ;关 LED

    MOV DX, 0E4AAH
    SUB AL, 05H
    OUT DX, AL                ;输出列号
    ADD AL, 05H

    MOV BX, 00H
LO_SHOW_TAB2:
    SHL BX, 01H                ;BX 左移一位
    CALL XLAT_2D

    CMP BYTE PTR [SI], 'X'
    JZ SHOW_CHESS1            ;棋子为 X, 显示在点阵板上
    CMP BYTE PTR [SI], 'O'
    JNZ NOT_SHOW_CHESS1
    CMP MAT_COUNT, 00H        ;棋子为 O 且 MATCOUNT 计数为 0, 显示在点阵板上
    JNZ NOT_SHOW_CHESS1
SHOW_CHESS1:
    INC BX
NOT_SHOW_CHESS1:
    INC AH
    CMP AH, 0DH                ;前八行
    JB LO_SHOW_TAB2

    PUSH AX
    MOV DX, 0E4A9H            ;控制点阵板上八行
    MOV AL, BL
    OUT DX, AL
    POP AX

```

# 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )

实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录: 实验程序源代码 (21)

```

MOV BX, 00H
LO_SHOW_TAB3:
    SHL BX, 01H                ;BX 左移一位
    CALL XLAT_2D

    CMP BYTE PTR [SI], 'X'
    JZ SHOW_CHESS2            ;棋子为 X, 显示在点阵板上
    CMP BYTE PTR [SI], 'O'
    JNZ NOT_SHOW_CHESS2
    CMP MAT_COUNT, 00H
    JNZ NOT_SHOW_CHESS2        ;棋子为 0 且 MATCOUNT 计数为 0, 显示在点阵板上

SHOW_CHESS2:
    INC BX
NOT_SHOW_CHESS2:
    INC AH
    CMP AH, 14H                ;后七行
    JB LO_SHOW_TAB3

    PUSH AX
    MOV DX, 0E4A8H             ;控制点阵板下八行
    MOV AL, BL
    SHL AL, 01H                ;再左移一次
    OUT DX, AL
    POP AX

    INC AL
    CMP AL, 14H                ;依次输出 15 个列
    JB LO_SHOW_TAB1

    INC MAT_COUNT
    AND MAT_COUNT, 07H         ;MAT_COUNT 在 0~7 循环

    POP SI
    POP DX
    POP CX
    POP BX
    POP AX
    RET
SHOW_TAB ENDP

TO_ASC PROC NEAR              ;转换 ASCII 码
    ADD DL, 30H                ;加 '0'
    CMP DL, 39H                ;超过 '9'?
    JBE RET_ASC
    ADD DL, 07H                ;加 07H
RET_ASC:
    RET
TO_ASC ENDP

XLAT_2D PROC NEAR             ;查二维数组, AH 为行号, AL 为列号, 左上角为 0101H
    PUSH AX
    PUSH DX

    MOV DH, 00H

```

## 北 京 大 学 实 验 报 告

系别 计算机科学技术系 班号                      姓名 管 毓 清 (同组姓名                     )实验日期 2012 年 5 月 18 日 至 2012 年 6 月 1 日

实验名称: 双机对战五子棋程序

## 附录: 实验程序源代码 (22)

```
MOV DL, AH                ;行号减一
DEC DL
MULTIPLY DX, 17H          ;乘以 23

MOV AH, 00H
ADD DX, AX                ;加列号
DEC DX                    ;减一

LEA SI, TAB                ;查表
ADD SI, DX                ;加偏移

POP DX
POP AX
RET
XLAT_2D ENDP

IRQ PROC FAR               ;中断服务程序
PUSH AX
PUSH DX

MOV DI, 01H                ;提示有中断

MOV AL, 20H
OUT 20H, AL

MOV DX, 0EC4DH
MOV AL, 1DH                ;清除 PCI9052 的中断标志
OUT DX, AL

POP DX
POP AX
IRET
IRQ ENDP

CODE ENDS
END START
```