# 微机原理与接口技术 实验报告一

来源：网络 作者：月落乌啼 更新时间：2025-01-21

*评阅微机原理与接口技术实验报告一姓名匡越学号1715211016时间地点实验题目一、实验目的1.熟悉Keil软件使用2.熟悉MCS-51指令3.学习简单程序的调试方法二、实验说明通过实验了解单片机内部存储器的结构和分配及读写存储器的方法，熟...*

评

阅

微机原理与接口技术

实验报告一

姓名

匡越

学号

1715211016

时间

地点

实验题目

一、实验目的1.熟悉Keil软件使用

2.熟悉MCS-51指令

3.学习简单程序的调试方法

二、实验说明

通过实验了解单片机内部存储器的结构和分配及读写存储器的方法，熟悉MCS-51指令同时，学习单片机程序编程、调试方法。

三、实验内容及步骤

1.启动PC机，打开Keil软件，软件设置为模拟调试状态。在所建的项目文件中输入源程序1，进行编译，如有错误按提示找到该行并纠错，重新编译直到通过。

2.编译无误后，打开CPU窗口，选择单步或跟踪执行方式运行程序，观察CPU窗口各寄存器的变化并将观察到的结果记录到预习报告。

3.新建另一个项目输入源文件2，打开CPU窗口，选择单步或跟踪执行方式运行程序，观察存储块数据变化情况记录到预习报告。点击复位按钮，改变存储块数据，点击全速执行快捷按钮，点击暂停按钮，观察存储块数据变化情况，记录到预习报告。点击复位按钮，改变存储块数据，分别LOOP、LOOP1设置断点，点击全速执行快捷按钮，在断点处观察寄存器及存储块数据变化情况。

WAVE软件使用方法参考其帮助文件。

四、实验程序流程框图、实验程序

1、源程序1

ORG

0000H

AJMP

MAIN

ORG

0030H

MAIN：

MOV

R0,#30H；

(R0)=

(00H)=

MOV

A,#40H；

(A)=

MOV

R6,A；

(A)=,(R6)=

MOV

A,@R0；

(R0)=

(A)=

MOV

40H,A；

(A)=

(40H)=

MOV

30H,40H；

(30H)=

(40H)=

MOV

R1,#40H；

(R1)=

MOV

@R1,#0AAH；(R1)=

(40H)=

MOV

SP,#60H；

(SP)=

PUSH

ACC；

(SP)=

(61H)=

PUSH

30H;

(SP)=

(62H)=

MOV

A,#0FFH；

(SP)=

(A)=

MOV

30H,#30H；

(SP)=

(30H)=

POP

ACC；

(SP)=

(A)=

POP

30H；

(SP)=

(30H)=

ADD

A,30H；

(30H)=

(A)=

Cy=

SUBB

A,#10；

(A)=

Cy=

MOV

R4,#00100100B；

(R4)=

H

MOV

A,#39H；

(A)

=

ADD

A,R4；

(A)

=

(R4=)

DA

A；

(A)

=

Cy=

MOV

28H,#55H；(28H)

=

Cy=

MOV

C,40H；

(PSW)

=

Cy=

MOV

26H,#00H；(26H)

=

Cy=

MOV

30H,C；

(30H)

=

(26H.1)

=

SJMP

$

j点击project，选择下拉式菜单中的New

project；

k选择所要的单片机，这里我们选择常用的Ateml

公司的AT89C51；

l新建一个File，输入源程序；

m将新建文件保存为text.asm的格式；

n鼠标在屏幕左边的Source

Group1

文件夹图标上右击弹出菜单，在这里可以做在项目中增加减少文件等操作。选“Add

File

to

Group

‘Source

Group

1’”弹出文件窗口，选择刚刚保存的文件；

o对程序进行编译运行；

使程序一得：

ORG

0000H

AJMP

MAIN

ORG

0030H

MAIN:

MOV

R0,#30H

;

(R0)=

0x30

(00H)=

0x0000

MOV

A,#40H

;

(A)=

0x40

MOV

R6,A

;

(A)=

0x40,(R6)=

0x40

MOV

A,@R0

;

(R0)=

0x30

(A)=

0x16

MOV

40H,A

;

(A)=0x16

(40H)=

0x0040

MOV

30H,40H

;

(30H)=

0x0030

(40H)=

0x0040

MOV

R1,#40H

;

(R1)=

0x40

MOV

@R1,#0AAH;(R1)=

0x40

(40H)=

0x0040

MOV

SP,#60H;

(SP)=

0x60

PUSH

ACC;

(SP)=

0x61

(61H)=

0x0061

PUSH

30H;

(SP)=

0x62

(62H)=

0x0062

MOV

A,#0FFH;

(SP)=

0x62

(A)=

0xff

MOV

30H,#30H;

(SP)=

0x62

(30H)=

0x0030

POP

ACC;

(SP)=

0x61

(A)=

0x16

POP

30H;

(SP)=

0x60

(30H)=

0x0030

ADD

A,30H;

(30H)=

0x0030

(A)=

0x2a

Cy=

SUBB

A,#10;

(A)=

0x20

Cy=

MOV

R4,#00100100B;

(R4)=

0x24

H

MOV

A,#39H;

(A)

=

0x39

ADD

A,R4;

(A)

=

0x5d

(R4=)

0x24

DA

A;

(A)

=

0x63

Cy=

MOV

28H,#55H;(28H)

=

0x0028

Cy=

MOV

C,40H;

(PSW)

=

0x80

Cy=

MOV

26H,#00H;(26H)

=

0x0026

Cy=

MOV

30H,C;

(30H)

=

0x0030

(26H.1)

=

SJMP

$

2、源程序2

设(30H)=4,(31H)=1,(32H)=3,(33H)=5,(34H)=2,(35H)=6

ORG

0000H

AJMP

MAIN

ORG

0030H

MAIN:

MOV

R0,#30H；30H→R0

MOV

R2,#6；6→R2

SORT:

MOV

A,R0;30H→A

MOV

R1,A；30H→R1

MOV

A,R2；6→A

MOV

R5,A；6→R5

CLR

F0；

状态标志位清零

DEC

R5；寄存器R5减一

MOV

A,@R1；R1→A

LOOP:

MOV

R3,A；A→R3

INC

R1

；寄存器R1增1

CLR

C

；进位标志位清零

MOV

A,@R1；31H→A

SUBB

A,R3；累加器内容减去寄存器内容

JNC

LOOP1；仅为标志位为1，则进行LOOP1

;以下代码完成数据交换

SETB

F0；状态标志位置1

MOV

A,@R1；31H→A

XCH

A,R3；将A于与R3的数据交换

MOV

@R1,A；将4赋值给寄存器R1(31H)

DEC

R1；寄存器减一

MOV

A,R3；1→A

MOV

@R1,A；将1赋值给寄存器R1(30H)

INC

R1;寄存器R1增一

LOOP1:

MOV

A,@R1；4→A

DJNZ

R5,LOOP；寄存器R5减一，不为零则回到LOOP

JB

F0,SORT；状态标志位为零，则回到SORT

SJMP

$

本DOCX文档由 www.zciku.com/中词库网 生成，海量范文文档任你选，，为你的工作锦上添花,祝你一臂之力！