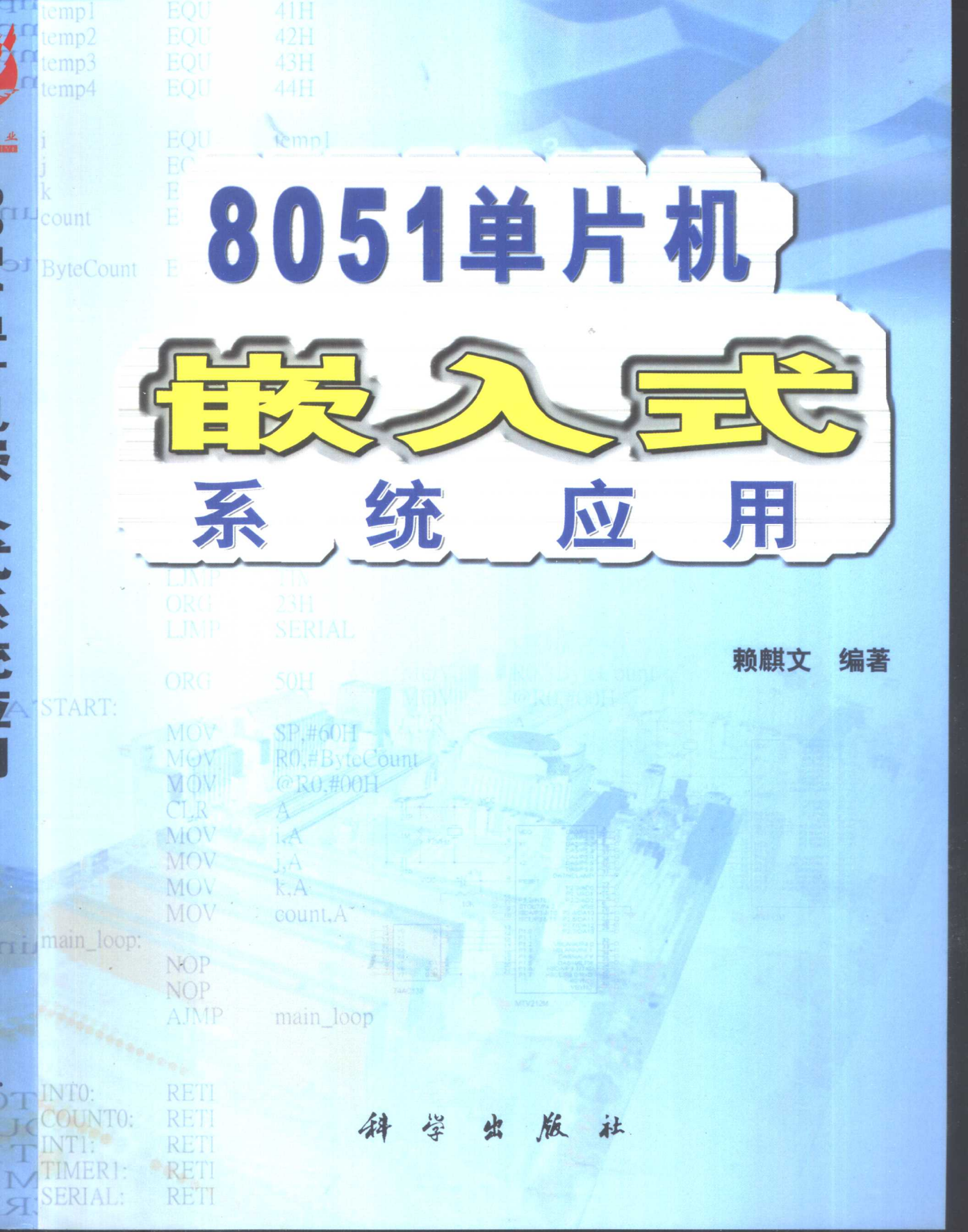


业

1

2

3

4

5

6

7

8

9

0

1

2

3

4

5

6

7

8

9

0

1

2

3

4

5

6

7

8

9

temp1 EQU 41H
temp2 EQU 42H
temp3 EQU 43H
temp4 EQU 44H

i EQU temp1
j EC
k E
count E

ByteCount E

8051单片机

嵌入式 系统应用

赖麒文 编著

START:

```
MOV SP,#60H  
MOV R0,#ByteCount  
MOV @R0,#00H  
CLR A  
MOV i,A  
MOV j,A  
MOV k,A  
MOV count,A
```

main_loop:

```
NOP  
NOP  
AJMP main_loop
```

INT0: RETI
COUNT0: RETI
INT1: RETI
TIMER1: RETI
SERIAL: RETI

科学出版社

8051 单片机嵌入式系统应用

赖麒文 编著

科学出版社

2002

内 容 简 介

单片机的应用非常广泛,本书以应用最广的 8051 单片机为例,由浅入深地阐述程序的结构、设计技巧与解决方法。

本书提供给读者许多汇编程序设计的例程,列出了许多程序的设计方法,并阐述其设计结构与意义,希望读者通过它能融会贯通,奠定日后软件思想基础及提高软件设计能力。

本书对初学者有很好的指导作用,也适合现阶段从事嵌入式系统的设计人员参考之用。

本书繁体字版原书名为《8051 单片机嵌入式系统入门与实务应用》,由文魁资讯股份有限公司出版,版权属赖麒文所有。本书简体字中文版由文魁资讯股份有限公司授权科学出版社独家出版。未经本书原版出版者和本书出版者书面许可,任何单位和个人不得以任何形式或任何手段复制或传播本书的部分或全部。

版权所有,翻印必究。

图字:01-2002-2266 号

图书在版编目(CIP)数据

8051 单片机嵌入式系统应用/赖麒文编著. —北京:科学出版社,2002
ISBN 7-03-010659-8

I .8... II.赖... III.单片微型计算机, 8051—系统设计 IV.TP368.1

中国版本图书馆 CIP 数据核字(2002)第 050540 号

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2002 年 7 月第 一 版 开本: 710×1000 1/16

2002 年 7 月第一次印刷 印张: 34 3/4

印数: 1—5 000 字数: 724 000

定价:49.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

前 言

8051 单片机的应用非常广泛，如电话机、温度、流量控制、马达、冰箱、电视、洗衣机、电子表、微波炉、监视器、CD-RW、医疗器材、通讯设备、卫星导航等。利用微电脑所制造的电子商品一年比一年增加，而微电脑 CPU 的价格也越来越低，8051 微电脑已经流行十几年了，以后的应用会越来越广泛。

本书以市面上应用最广的 8051 单片机为例，由浅入深地阐述程序结构、设计技巧与解决方法，非常适合学生们制作实验之用，以作为日后提升嵌入式系统设计能力的基础，也适合现阶段从事嵌入式系统的设计人员参考之用。

本书提供给读者许多汇编程序设计的例程，介绍了许多的程序设计方法，并阐述其设计结构与意义，希望读者能融会贯通，奠定日后软件思维基础及提高软件设计能力。本书的宗旨是提供一些解决问题的方法来启发读者活用汇编语言，我们希望当读者想要设计某种功能的软件时，可以立即参考本书，从而能很快地了解其程序设计概念与技巧。

书中范例的程序结构、设计技巧与解决方法都以堆积木的方式，由浅入深地解说。请你仔细阅读本书，去了解到底有哪些例程，它们分别是做什么用的，其功能是什么，以及程序为何要这样设计等。

本书为 8051 单片机基础与进阶训练的实用教材，希望对初学者有很好的引导作用，也希望对编写嵌入式软件兴趣勃勃，但就是不知从何着手；或是具有工程背景，有基本的程序设计经验及基础的硬件知识者；或是精通嵌入式系统设计，但需要进一步参考资料的核心技术者；或是使用不同型号的微电脑设计者都能通过本书学习到有关硬件的知识及软件设计的思想与解决方法，从而成为这方面的佼佼者。

由于作者水平有限，书中错漏之处在所难免，恳请读者批评指正。

作 者

目 录

第 1 章 指令介绍.....	1
1-1 指令符号的意义说明	1
1-2 8051 的指令集	1
第 2 章 汇编语言发展流程.....	18
2-1 如何进行软件开发	18
2-1-1 研究开发的流程	18
2-1-2 软件开发的步骤	19
2-2 变量寻址法	20
2-3 编译器和连接器的使用.....	30
2-4 烧录器的使用	36
第 3 章 基本程序设计的应用.....	37
第 4 章 开机动作流程的应用.....	98
4-1 PowerOnInitial.....	98
4-2 InitialCpu	100
4-3 InitialCpu1	101
4-4 InitialCpuIO.....	103
4-5 InitialEeprom	104
4-6 InitialVariable	107
第 5 章 延迟例程的应用.....	109
5-1 DelayX1ms	109
5-2 DelayX1ms1	110
5-3 DelayX1ms2	111
5-4 DelayX10ms	113
5-5 DelayX10ms1	114
5-6 Delay50us	116
5-7 ShortDelay	117
5-8 Timer40msDelay	118
第 6 章 输出/输入接口的应用	120
6-1 Led_1	120

6-2	LedOn	121
6-3	Input1	123
6-4	Input2	124
6-5	Input3	129
6-6	Input4	131
6-7	Input5	133
第 7 章	中断的应用	136
7-1	CountMain1	136
7-2	Timer0ISR_2	140
7-3	CountMain2	142
7-4	CountMain3	144
7-5	One_INT0ISR	145
7-6	More_INT0ISR	147
7-7	Timer1ISR_1	151
第 8 章	公用例程的应用	153
8-1	ByteVariableAdd	153
8-2	ByteVariableSub	154
8-3	ByteProcess	156
8-4	WordVariableAdd	158
8-5	WordVariableSub	160
8-6	WordProcess	161
8-7	Hex2Bcd1	164
8-8	Hex2Bcd2	165
8-9	Hex2Bcd3	167
8-10	Value255_100	169
8-11	Value100_128a	170
8-12	Value100_128b	171
8-13	RamClear	173
8-14	ZeroContinue	174
第 9 章	显示器的应用	176
9-1	LedFlash0	176
9-2	LedFlash1	177
9-3	LedFlash2	177
9-4	LedFlash3	179

9-5	LedFlash4	179
9-6	LedFlash5	180
9-7	LedFlash6	182
9-8	LedFlashGetkey	183
9-9	LedMain1	185
9-10	LedMain2	186
9-11	LedTimming	189
9-12	LedMain3	194
9-13	LedMain4	196
9-14	LedMain5	198
第 10 章	蜂鸣器的应用	207
10-1	Beep1	207
10-2	Beep2	210
10-3	Beep3	213
10-4	Beep4	215
10-5	BeepGetkey	219
10-6	Alarm1	222
10-7	Alarm2	224
10-8	AlarmGetkey	227
10-9	BeepLed	230
10-10	HardWareBeep1	234
10-11	HardWareBeep2	235
10-12	HardWareBeep3	236
第 11 章	演奏歌曲的应用	238
11-1	Sound	239
11-2	Music1	242
11-3	Music2	244
11-4	Music3	247
11-5	Music4	251
第 12 章	七段显示器的应用	257
12-1	BcdDisplay1	257
12-2	BcdDisplay2	259
12-3	BcdDisplay3	262
12-4	BcdDisplay4	265

第 13 章	点矩阵显示器的应用.....	267
13-1	Dot5x7_Display1.....	267
13-2	Dot5x7_Display2.....	269
13-3	Dot5x7_Display3.....	271
13-4	Dot5x7_Display4.....	281
13-5	Dot5x7_Display5.....	284
13-6	Dot5x7_Display6.....	287
第 14 章	译码器的应用.....	290
14-1	Output74138_1.....	290
14-2	Output74138_2.....	291
14-3	Output74138_3.....	293
14-4	Output74138_4.....	295
第 15 章	扩充输出端口的应用.....	299
15-1	Output4094_1.....	299
15-2	Output4094_2.....	302
15-3	Output4094_3.....	302
15-4	Output4094_5.....	304
第 16 章	脉冲的应用.....	307
16-1	OutPulse1.....	307
16-2	OutPulse2.....	308
16-3	OutPulse3.....	309
16-4	OutPulse4.....	311
16-5	PulseDetect1.....	312
16-6	PulseDetect2.....	314
16-7	PulseDetect3.....	315
16-8	PulseGenerator.....	317
16-9	PulseDuty1_Timer1ISR.....	319
16-10	PulseDuty2_Timer1ISR.....	321
16-11	CheckPulseCome.....	323
16-12	CheckPulseWidth.....	324
16-13	CheckPulseData.....	326
16-14	CheckPulseHiLow.....	329
16-15	PulseDecoder.....	332
16-16	EncoderProcess.....	335

第 17 章	扩充输入端口的应用	340
17-1	Input4051_1	340
17-2	Input4051_2	341
17-3	Input4051_3	343
17-4	Input4051_4	344
17-5	Input4051_5	346
17-6	Input4051_6	349
17-7	Input4067_1	351
17-8	Input4067_2	355
17-9	Input4067_3	358
第 18 章	键盘操作的应用	361
18-1	InputKey1	361
18-2	InputKey2	363
18-3	InputKey3	365
18-4	ScanKey1	367
18-5	ScanKey2	372
18-6	GetKey1	376
18-7	GetKey2	381
18-8	KeyCheck	386
18-9	KeyCountCheck	388
18-10	KeyProcess	390
第 19 章	可控制电源电压的应用	398
19-1	LM7805	398
19-2	LM7812	398
19-3	LM317	399
19-4	Dac08	400
19-5	SawTooth	401
19-6	TriAngle	402
19-7	Square	404
第 20 章	内存 IC 93C66 的应用	406
20-1	PushEeprom93c66	406
20-2	EepWriteData	407
20-3	PopEeprom93c66	413
20-4	ReadROM	414

第 21 章	IIC BUS 的应用	416
21-1	IIC BUS 简介	416
21-2	IIC 总线协议	417
21-3	IIC BUS 时序	419
21-3-1	I2cStart	419
21-3-2	I2cStop	420
21-3-3	I2cWait	420
21-3-4	I2cSentByte	421
21-3-5	I2cSentByte1	423
21-3-6	I2cReceiveByte	424
21-3-7	SendAcknowledge	425
21-3-8	I2cByteWrite	426
21-3-9	I2cByteWrite1	427
21-3-10	I2cByteWrite2	429
21-3-11	I2cByteRead	431
21-3-12	I2cSentData	432
21-3-13	I2cReceiveData	434
21-3-14	DataSetBit	436
21-3-15	DataClearBit	439
第 22 章	PWM IC 的应用	443
22-1	PWM_Output	443
22-2	TEST_DacOut	444
第 23 章	内存 IC 24C08 的应用	448
第 24 章	记忆 IC 24C32 的应用	476
24-1	Eeprom24c32WriteByte_1	476
24-2	Eeprom24c32WriteByte_2	477
24-3	Eeprom24c32WriteMulti_1	479
24-4	Eeprom24c32WriteMulti_2	480
24-5	Eeprom24c32ReadByte_1	482
24-6	Eeprom24c32ReadByte_2	483
24-7	Eeprom24c32ReadWord_1	485
24-8	Eeprom24c32ReadWord_2	486
24-9	Eeprom24c32ReadMulti_1	488
24-10	Eeprom24c32ReadMulti_2	490

第 25 章	OSD IC 的应用.....	493
25-1	OsdStart	493
25-2	OsdStop.....	494
25-3	OsdSentByte	495
25-4	OsdReceiveByte	496
25-5	OsdFormatA_0	498
25-6	OsdFormatA	499
25-7	OsdFrameControl	500
25-8	OsdLocationSet	501
25-9	OsdRamClear.....	501
25-10	OsdEnable.....	502
25-11	OsdOpenUp	503
25-12	OsdNormal.....	503
25-13	OsdResetFont.....	504
25-14	OsdClearRow.....	506
25-15	OsdClearRow1	508
25-16	OsdPrintIcon	510
25-17	OsdStringAdr0.....	511
25-18	OsdStringAdr	513
25-19	OsdPrintString	516
25-20	OsdPrintString1	519
25-21	OsdDisableWindow1	521
25-22	OsdSetWindow	524
25-23	OsdBarHandle	526
25-24	OsdBarHandle1	531
25-25	OsdDisplayValue	535
25-26	OsdDisplayCount.....	542

第 1 章 指令介绍

1-1 指令符号的意义说明

Rn: 表示缓存器 R0~R7, 并受缓存器库选择位 RS0, RS1 控制。

Ri: 表示为 R0 或 R1, 以作为间接寻址的指针。

@Ri: 缓存器间接寻址, 即通过缓存器 R0 或 R1 作间接寻址指针, 可寻址到内部数据存储器 0~255 的地址。

#data: 立即寻址, 表示为八位的数据, 内容为 0~255。

bit: 表示在内部数据存储器可直接寻址的位, 或特殊功能缓存器 (SFR) 内可直接寻址的位。

direct: 直接寻址, 表示八位内部数据的地址码, 可寻址到内部数据存储器 0~127 的地址, 或者是 SFR 中的 I/O 端口、控制缓存器、状态缓存器等地址(80H~FFH)。

标名: 表示地址的符号名称。

(): 小括号表示缓存器或变量值。

例如:

(A): ACC 累积器的数值或内容。

(Rn): R0~R7 缓存器的值。

((Ri)): 由 R0 或 R1 所指到的地址的值。

(direct): 直接寻址变量的值。

(C): 进位标志的值。

(bit): 可以位寻址变量的值。

1-2 8051 的指令集

1. ACALL 标名 (地址码)

功能: 绝对无条件调用。

说明: 本指令 ACALL 无条件调用位于所标示的地址的子程序, 而子程序调用的起始地址必须位于程序内存与 ACALL 下一个指令的第一个字节的 2KB 字节范围内。在执行子程序之后, 程序计数器又会指到 ACALL 的下一个指令, 即程序会继续从 ACALL 的下一个指令开始执行。

2. ADD A,<来源字节>

功能：加法。

说明：将指令寻址的值与累积器相加，结果再存回累积器中，如果相加后的结果大于 255，表示有进位，则 C 进位标志会设定为“1”。

ADD A, Rn (Rn:R0~R7)

运算 $(A) \leftarrow (A) + (Rn)$

ADD A, direct （直接寻址，内存地址 00H~7FH）

运算 $(A) \leftarrow (A) + (\text{direct})$

ADD A, @Ri （缓存器间接寻址，i=0~1，内存地址 80H~FFH）

运算 $(A) \leftarrow (A) + ((Ri))$

ADD A, #data （立即数据寻址，data=0~255）

运算 $(A) \leftarrow (A) + \#data$

3. ADDC A,<来源字节>

功能：带进位的加法。

说明：将累积器的数值与来源字节的数值及目前进位标志三者相加后其结果再存回累积器 A 中。如果进行加法运算后的结果大于 255，表示有进位 C，进位标志会设定为“1”。

ADDC A, Rn

运算 $(A) \leftarrow (A) + (Rn) + (C)$

ADDC A, direct

运算 $(A) \leftarrow (A) + (\text{direct}) + (C)$

ADDC A, @Ri

运算 $(A) \leftarrow (A) + ((Ri)) + (C)$

ADDC A, #data

运算 $(A) \leftarrow (A) + \#data + (C)$

4. AJMP 标名

功能：绝对跳跃。

说明：AJMP 转移程序执行至标示的地址，其目的地址必须位于程序内存与 AJMP 的下一个指令的第一个字节 2KB 字节的范围之内。

5. ANL <目的字节>, <来源字节>

功能：两个字节作 AND 运算。

说明：将目的字节与来源字节作 AND 运算，其结果再存回目的字节内，常作为屏蔽位用，即将 bit0 ~ bit7 强迫清除为“0”

ANL A, Rn

运算 $(A) \leftarrow (A) .AND. (Rn)$

ANL A, direct

运算 $(A) \leftarrow (A) .AND. (direct)$

ANL A, @Ri

运算 $(A) \leftarrow (A) .AND. ((Ri))$

ANL A, #data

运算 $(A) \leftarrow (A) .AND. \#data$

ANL direct, A

运算 $(direct) \leftarrow (direct) .AND. (A)$

ANL direct, #data

运算 $(direct) \leftarrow (direct) .AND. \#data$

6. ANL C, <来源位>

功能：位变量（标志变量）之间的 AND 运算。

说明：如果来源位的值为“0”，则进位标志将被清除为“0”，否则进位标志保持目前状态值，而来源位仅允许直接位寻址模式。

ANL C, 位

运算 $(C) \leftarrow (C) .AND. (bit)$

ANL C, /位

运算 $(C) \leftarrow (C) .AND. /(bit)$

即将 C 和位的补救作 AND 运算

7. CJNE <目的字节>, <来源字节>, 标名

功能: 若二者不相等则跳跃。

说明: 将目的字节与来源字节作比较, 若二者不相等, 则程序跳往标名处执行, 如果目的字节的数值小于来源字节的数值, 则执行本指令之后, 进位标志 C= “1”, 否则 C= “0”。

CJNE A, #data, 标名

运算: 若(A)和#data 不相等, 则跳至标名的地址去执行。

CJNE A, direct, 标名

运算: 若(A)和(direct)不相等, 则跳至标名的地址去执行。

CJNE Rn, #data, 标名

运算: 若(Rn)和#data 不相等则跳至标名的地址去执行。

CJNE @Ri, #data, 标名

运算: 若((Ri))和#data 不相等则跳至标名的地址去执行。

8. CLR A

功能: 将累积器的内容清除为 00H。

说明: 本指令使得累积器内的所有位清除为 0。

9. CLR 位

功能: 清除可位寻址的位。

说明: 被指定到的位将被清除为 0, 可使用在进位标志或任何可直接寻址的位上运算。

CLR P1.3

运算: 清除 P1.3= “0”, 即 P1 端口 bit3 清除为 “0”。

CLR C

运算: $(C) \leftarrow 0$, 清除 C 进位标志。

CLR bit

运算: $(\text{位}) \leftarrow 0$

10. CPL A

功能: 取累积器内容的补码

说明: 将累积器内的每一个位取其补码, 即“1”→“0”, “0”→“1”

11. CPL 位

功能: 将寻址的位取补码。

说明: 所标示的位变量取其补码后再存回原处, 可在进位标志或可直接寻址的地址上运算。

CPL P1.2

运算: $(P1.2) \leftarrow \neg(P1.2)$

CPL C

运算: $(C) \leftarrow \neg(C)$

CPL bit

运算: $(\text{bit}) \leftarrow \neg(\text{bit})$

12. DA A

功能: 将累积器 ACC 的内容调整为二进制编码的十进制(BCD)的形式, 并将其结果再存回累积器中, 以作加法之用。

说明: 若累积器的 bit0 ~ bit3 的数值大于 9 或辅助进位为“1”, 则累积器之内含值加 6, 若有进位则高四位再加 1, 然后再检查 bit4 ~ bit7 的数值大于九或进位标志为“1”, 则高四位也要加 6, 结果若发生溢位则进位标志设定为“1”, 否则为“0”。

13. DEC 字节

功能: 将指定位组的内容减 1。

说明: 将所标示的变量减 1, 若原来值为 00H, 则执行本指令之后, 将会产生

借位而成为 FFH，本指令并不影响任何标志，若原来值为 01H，则减后为 00H，也不会设定 Z=“1”，请不要使用 JZ 的指令。

DEC A

运算： $(A) \leftarrow (A) - 1$

DEC Rn

运算： $(Rn) \leftarrow (Rn) - 1$

DEC direct

运算： $(direct) \leftarrow (direct) - 1$

DEC @Ri

运算： $((Ri)) \leftarrow ((Ri)) - 1$

14. DIV AB

功能：A 除以 B。

说明：将累积器 ACC 数值除以缓存器 B 的数值，其结果“商”存入累积器中，“余数”则放在 B 缓存器中，即 $A/B=A...B$ 。

15. DJNZ <字节>, 标名

功能：所指定的字节的内容先减 1，再判断其结果若不为 00H，则跳至标名的地址去执行。

说明：字节可以是一个缓存器或可直接寻址的字节，经减 1 后仍会存入原来的字节内，若原来值为 00H，经减 1 后会成为 FFH 并存入至原来的字节内。

DJNZ Rn, 标名

运算： $(Rn) \leftarrow (Rn) - 1$ ，如果结果(Rn)不等于 00H，则跳至标名的地址去执行。

DJNZ direct, 标名

运算： $(direct) \leftarrow (direct) - 1$ ，若结果(direct)不等于 00H，则跳至标名的地址去执行。

16. INC <字节>

功能：将被寻址的字节的内容加 1，并存回原来的字节。

说明：若起始值为 FFH，则加 1 后的内容变为 00H，此指令并不会影响任何标志。