

正确发挥ICU效能初探

林章省

(电子工程系)

摘 要

一位微处理器(ICU)主要用于逻辑运算和开关控制,此外还可以进行数值运算、比较和处理。但必须有适当的逻辑电路和软件作支持。本文着重介绍一位机用于数值比较和执行加法运算的程序设计方案,讨论了如何充分发挥一位机的简单、可靠、价廉等特点,以达到成本低、效益高之目的。

计算机用于生产过程自动控制,就其过程性质而论要面临三种主要任务:(1)连续过程自动控制;(2)继续过程自动控制;(3)连续与断续混合过程自动控制。

面对各种不同任务有各种不同复杂程度和规模的过程控制计算机,以获得尽可能高性能价格比,收到最好的效果。微处理器的生产也这样。由于工业控制领域里存在着量大面广的开关逻辑系统,美国莫托洛拉公司已经生产出一种面向工业上逻辑控制的一位微处理器MC14500,称为工业控制单元ICU(Industrial Control Unit)。它的出现为控制领域提供了一种简单、可靠、价廉的优良器件,具有极强的生命力。如同微处理器的出现给计算机领域产生革命作用一样,ICU的出现也在控制领域里激起了极大的波润,其应用发展迅猛异常。

然而它的主要优势在于开关逻辑控制,对于兼有数值处理的开关控制场合,它常常不得不让位于多位机。那么一位机是不是不能进行数值处理呢?本文将着重讨论如何发挥一位机现有功能,使它具有中速的数值处理能力的问题。

一、数值比较

假如ICU在控制过程中需要对两个四位二进制数值 $A_3A_2A_1A_0$ 和 $B_3B_2B_1B_0$ 进行比较,然后根据比较结果决定相应的处理,那么我们可以把这两个四位数据送到ICU系统输入的八个输入点。然后让ICU在适当的时候执行一段程序对它们进行比较。为了保存比较结果,我们需要三个暂存单元,设为EQU、AGT、BGT,分别表示 $A=B$ 、 $A>B$ 、 $A<B$,系统示意图如图1。

比较程序的设计思想是这样的:从最高位开始进行,若 $A_3>B_3$ 或 $A_3<B_3$,那么比较便可

本文1986年11月10日收到。

结束, 其低位不必再行比较, 若 $A_3 = B_3$, 则需对次高位 A_2, B_2 作比较。若 $A_2 > B_2$ 或 $A_2 < B_2$, 则比较也可告结束。否则继续对 A_1, B_1 进行比较, 直到最低位比较完毕为止。

ICU程序的处理方法可以有以下两种。

1. 采用允许结构程序

流程框图如图2, 其程序如下:

序号	源程序
1	ORC RR
2	IEN RR
3	OEN RR
4	LD A_3
5	ANDC B_3
6	STO AGT
7	LDC A_2
8	AND B_2
9	STO BGT
10	OR AGT
11	STOC EQU
12	OEN EQU
13	LD A_1
14	ANDC B_1
15	STO AGT
16	LDC A_0
17	AND B_0
18	STO BGT
19	OR AGT

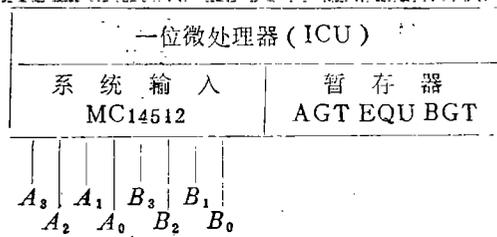


图1 两数值比较ICU系统示意图

序号	源程序
20	STOC EQU
21	OEN EQU
22	LD A_1
23	ANDC B_1
24	STO AGT
25	LDC A_1
26	AND B_1
27	STO BGT
28	OR AGT
29	STOC EQU
30	OEN EQU
31	LD A_0
32	ANDC B_0
33	STO AGT
34	LDC A_0
35	AND B_0
36	STO BGT
37	OR AGT
38	STOC EQU

以上程序包含四个OEN结构块, 分别对四对数值位进行比较。其结果存于AGT、BGT和EQU三个单元中。由于OEN EQU的控制作用, 只有当高一位比较结果是相等时才允许对次高位作比较。

2. 采用常规分支程序

此时系统必须有相应的转移逻辑, 下面的程序设计是假定在此基础上进行的, 其程序流程图如图3, 相应的ICU程序如下:

1	START: OSC RR	18	XNOR B_0
2	IEN RR	19	STOC EQU
3	OEN RR	20	AND A_0
4	STOC EQU	21	STO AGT
5	LDC A_3	22	LDC A_0
6	XNOR B_3	23	AND B_0
7	SKZ	24	STO BGT
8	JMP BRN1	25	JMP CNT
9	LDC A_2	26	BRN1: LD A_3
10	XNOR B_2	27	JMP SET
11	SKZ	28	BRN2: LD A_2
12	JMP BRN2	29	JMP SET
13	LDC A_1	30	BRN3: LD A_1

14	XNOR	B ₁	31	SET,	STO	AGT
15	SKZ		32		STOC	RGT
16	JMP	BRN3	33	CNT,		
17	LDC	A ₀				

上述程序在三个地方使用 SKZ (判结果为 0 跳过下一步) 和 JMP (无条件转移) 两条指令相配合实现条件分支, 对应于框图中的三个判别框, 这是一位机较特殊的地方。

二、加法运算

设外界有一四位数据 $B = B_3B_2B_1B_0$

(B_3 为最高位), 接到 ICU 系统的四个输入点, 要与系统内部的另一数据 (存在四个暂存单元) $A = A_3A_2A_1A_0$ 相加, 结果仍存于 A , 即 $A + B \rightarrow A$ 。

要解这样的问题可以先设计一个一位加法子程序, 然后设计一个主程序, 并在其相应点调用子程序。同样, 它也要有适当的附加逻辑作保证。系统与外部设备间通过一个状态触发器以联络信号实现应答式的控制。当外设准备好一数据时就送一个信号把触发器置位, 表示准备就绪。ICU 则以程序询问方式检查该标志的状态 (图 4 中输入点 RDY), 决定可否执行加法程序, 而每次执行加法程序后又通过输出线 ADE 发出一个脉冲去把触发器复位, 作为应答信号。系统硬件简图见图 4。

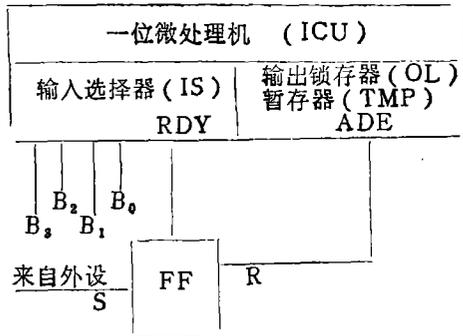


图4 有联络信号的ICU系统

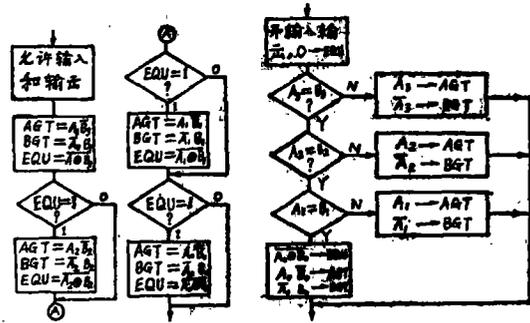


图2 两个四位数值比较的允许程序流程图

图3 两个四位数值比较的常规分支程序流程图

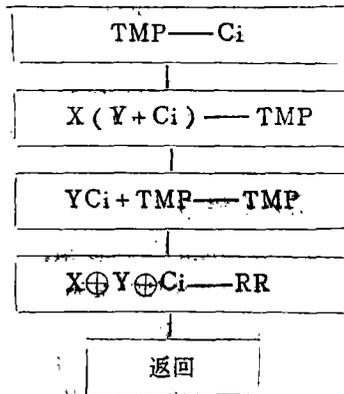


图5 一位加法子程序框图

1. 子程序设计

这里用一个暂存单元来存放中间结果和进位, 而在执行步骤上是先求进位输出后求和。并且和就保留在结果寄存器 RR 中。这既可少用一个暂存单元又可缩短程序。

子程序名字: ADD

入口条件: 被加数在 X, 加数在 Y, 进位入 (C_i) 在 TMP。

出口条件: 进位出 (C₀) 在 TMP, 和在 RR 中。

ADD 子程序流程图见图 5。

2. 主程序的设计

它的设计思想是从最低位开始依次向高位按位加。在每次调用ADD之前应把各该位的被加数送X，加数送Y；而在每次执行完子程序后，保留在RR中的和应送入A中的相应位，作为累加数的一位。但在TMP的进位出即为高一位的进位入，故无需另外操作。在执行本程序的开始先要检查一下RDY，且仅当RDY = 1时才可能执行加法程序，并向外发一个回答信号，图6是其流程图。

3. 源程序的程序单

(A) 子程序

```

1  ADD:  LD    TMP ; 原存在TMP中的进位输入
2        STO  Ci  ; 送Ci保存
3        OR   Y
4        AND  X
5        STO  TMP ; X (Y + Ci) 暂存在TMP中
6        LD   Y
7        AND  Ci
8        OR   TMP
9        STO  TMP , 进位输出X (Y + Ci) + YCi → TMP.
10       LD   Ci
11       XNOR Y
12       XNOR X ; RR = X ⊕ Y ⊕ Ci = X ⊕ Y ⊕ Ci = SUM
13       RTN
    
```

(B) 主程序

1	LDC	RDY	17	LD	A ₂
2	SKZ		18	STO	X
3	JMP	OTHER	19	LD	B ₂
4	STO	TMP	20	STO	Y
5	LD	A ₀	21	JSR	ADD
6	STO	X	22	STO	A ₂
7	LD	B ₀	23	LD	A ₃
8	STO	Y	24	STO	X
9	JSR	ADD	25	LD	B ₃
10	STO	A ₀	26	STO	Y
11	LD	A ₁	27	JSR	ADD
12	STO	X	28	STO	A ₃
13	LD	B ₁	29	ORC	RR
14	STO	Y	30	STO	ADE
15	JSR	ADD	31	STOC	ADE
16	STO	A ₁	32	OTHER:	

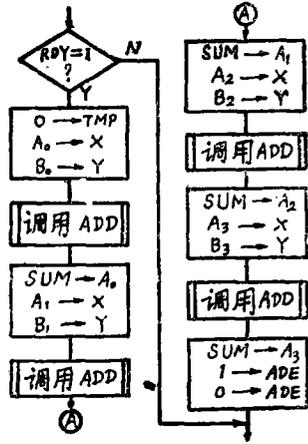


图6 四位加法主程序框图

注：借助于附加逻辑，调子指令JSR可以是NOPO或NOPF。

三、 几 点 讨 论

(1) 上面只讨论了四位数值比较和加法程序的设计方法,实际上对于任何长度的数值比较和相加都可以照这样处理,只是需要相应地加长程序。此外,一位机还可以执行减法、乘法和除法以及增量、减量、软件定时等运算,只要程序存储器、暂存器的容量和执行速度能满足具体应用要求,使用者具有一般数学逻辑能力和程序设计基础知识,便可设计出相应的程序来。

(2) 为提高运算速度、缩短程序,可以采用查表法。这是一种比较理想的方法。它使用一个ROM作为运算函数表,其地址接到ICU系统的某些输出点,数据输出接到系统的某些输入点(图7)。ROM的地址是参加某种运算的操作数有关位输入。其内容对应于所有可能输入状态的已知结果。这样,ICU便可简单地把要参加运算的有关数值位直接通过输出部件送到ROM的输入,而通过输入部件直接读取ROM的输出,作为运算结果;既方便又迅速。

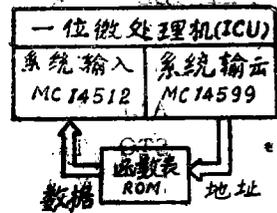


图 7 用查表法实现的ICU系统

例如,假定ROM作为四位加法,那么要对 $X = X_3X_2X_1X_0$ 和 $Y = Y_3Y_2Y_1Y_0$ 相加,并将结果仍放在X中,进位输出暂存在TMP单元,就可以编制如下一段程序来实现。这里的ROM应有8个地址端 A_7-A_0 和5个数据端即四位的和 S_3-S_0 及一位进位出 C_0 ,它们均与系统的输出、输入符号地址相对应。

1	LD	X_0	10	STO	A_4	19	LD	S_1
2	STO	A_0	11	LD	Y_1	20	STO	X_1
3	LD	X_1	12	STO	A_5	21	LD	S_2
4	STO	A_1	13	LD	Y_2	22	STO	X_2
5	LD	X_2	14	STO	A_6	23	LD	S_3
6	STO	A_2	15	LD	Y_3	24	STO	X_3
7	LD	X_3	16	STO	A_7	25	LD	C_0
8	STO	A_3	17	LD	S_0	26	STO	TMP
9	LD	Y_0	18	STO	X_0			

同前面介绍过的全部由ICU指令执行的加法程序比较,长度可缩短 $1/2$,执行速度可提高约2倍。

(3) 一位机的突出优点是简单、可靠、便宜,在开关逻辑控制领域里它优于多位机。由于它原理简单,很容易为一般电气技术人员及电器维修工人所掌握,因而便于推广应用;由于它保持了程序可编、灵活通用、适应性强,兼有设计灵活,扩展容易,器件制造工艺先进,有很强的抗干扰能力等特点,这一切就使得它具有独特的优越性和极广阔的发展前景。在逻辑运算与控制领域里有它的特殊位置,因此应该在此领域普遍推广,优先应用。此外,在对速度要求不是主要的地方,它还可兼一定的数据采集、数值运算、比较和处理等任务。

但对于复杂的计算和数据处理任务,一位机是难以胜任的。这时应该优先采用8位机或16位机,在某些场合下,为减轻多位机的负担,可以组合起来运用。使一位机和多位机各自发挥其特点,以便收到更好的效果。

参 考 文 摘

- (1) MC14500B Industrial control Unit Handbook, Theory and Application Motorola Corp, (1978).
- (2) 黄汉祥、朱介炎, 5 G14500系列CMOS一位微处理器, 晶峰器件应用, 1(1981).
- (3) 张为民译, 一位CMOS微处理器, 机械工业自动化, 1—2(1979).

An Investigation on Properly Bringing the Capabilities of Industrial Control Unit into Play

Liu Zhangsheng

Abstract

The industrial control oriented one-bit microprocessor known as Industrial Control Unit(ICU) is mainly used in logic operation and control of switching. with the support of appropriate logic circuit and software, it can also perform arithmetic operation, quantities comparison and data processing.

This paper introduces primarily programming schemes by which one-bit microcomputer is used to perform comparison and addition operations. It discusses also the problem of how to bring these features such as simplicity, reliability and cheapness of one-bit microcomputer into full play, so as to obtain a greater benefit inexpensively.