简明计算机入门

〔日〕 马目洋一 著
潘福美 译
林国宁 校

内 容 简 介

本书根据载于日刊“電子技術”的连载讲座“やさしいコンピュータ入門”（马目洋一著，1976年2月号载完）译出。

本书以微程序计算机为重点，从整机概念出发，通过对电子计算机工作过程的剖析以及典型实例，深入浅出、简明扼要地介绍了现代电子计算机的工作原理和实现方法。书中插图较多，所以比较具体、形象，易于理解。

全书共分13章，第1～3章讲述通常的电子计算机结构和工作概况，第4～11章着重论述微程序计算机的原理和方法，第12章谈用户程序设计，第13章介绍输入/输出接口电路和中断处理。

本书可供从事计算机、自动控制、集成电路等方面工作的人员及高等院校学生和自学者参考。

やさしいコンピュータ入門

〔日〕 馬目洋一

日刊“電子技術”1976年

*

简明计算机入门

潘福美 译

林国宁 校

*

国防工业出版社 出版

北京市书刊出版业营业许可证出字第074号

新华书店北京发行所发行 各地新华书店经售

国防工业出版社印刷厂印装

*

850×1168 1/32 印张 6 5/16 158千字

1979年9月第一版 1979年9月第一次印刷 印数：000,001—240,005册

统一书号：15034·1836 定价：0.83元

目　录

第 1 章　计算机的基本结构

1.1　程　　序

为了理解计算机都通用的最基本的结构和操作原理，首先要列举一些简单的程序例子。

例如，1 加 2 怎样变成 3。

把这个问题用某种编译程序的语言来表示为

X＝1

Y＝2

Z＝X＋Y

STOP（停止）

通常，在编译程序的语言里，式中符号“＝”与数学上用的等号含义是不同的，在这里，把符号“＝”规定为将右边的东西放进左边去的记号。这样，上式被解释为

把 1 放入 X

把 2 放入 Y

把 X＋Y 放入 Z

为了直观起见，也可表示为

X←— 1

Y←— 2

Z←— X＋Y

STOP（停止）

可借助假想的箱子来理解：

把 1 装进名叫 X 的箱子

把 2 装进名叫 Y 的箱子

把箱 X 中的内容加上箱 Y 中的内容，装进名叫 Z 的箱子

停止

现用编译程序语言书写的程序来进行编译。所谓编译，就是

把为人们容易理解的上述符号的表现形式变为计算机能够执行的形式。

为了处理箱X和箱Y中的内容，计算机须有“算盘”那样的寄存器，叫做累加器。

编译的结果，把上述表达式变为下列A、B、C、D、X、Y、Z诸项。

A：将箱X中的内容送入累加器。

B：将箱Y中的内容与累加器中内容相加（相加的结果放入“算盘”那样的累加器）。

C：将累加器中的内容放入箱Z中●。

D：使计算机停机。

X：箱X中已放入1。

Y：箱Y中已放入2。

Z：箱Z中放什么都可以。这里，是放入零。

在这些项目中，A、B、C、D表示主动，通常叫做指令；X、Y、Z表示被动，通常叫做数据。

分解为上述诸项，就变成了计算机能够直接执行的程序形式。为了书写这种形式的程序，有汇编程序语言或机器语言。

因为现在的计算机大都采用在机器内部储存程序的方式，故上列诸项目中，不分指令和数据都各装入一个箱子。这个概念很重要。

这里仅考虑了1＋2的计算顺序。如果考虑把计算机本身的控制顺序也装入一个特殊箱子，这就被理解为微程序设计技术。以上诸项用汇编程序语言表示如下：

```
A   LDA    X
B   ADA    Y
C   STA    Z
```

● 原文中漏掉Z字。——译者注

```
D   HLT
X   DEC    1
Y   DEC    2
Z   DEC    0
```

对任意地址M，将这种汇编程序语言的意义规定如下（在计算机用语中，许多箱子的集合体叫做存储器，箱的名字叫做地址）：

LDA M 〔将（M）存入A〕〔Load（M）into A〕

即把M地址的内容放入叫做A寄存器的累加器。

ADA M 〔加（M）到A〕〔Add（M）to A〕

即把M地址的内容与累加器中的内容相加，并将结果送至A。

STA M 〔将（A）存入M〕〔Store（A）into M〕

即把累加器的内容存入M地址

HLT 〔停机〕 〔Halt〕

即计算机停机。

M DEC n〔定义十进制常数〕〔Defines decimal constants〕

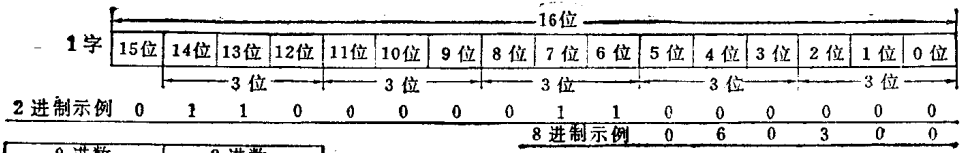
即将十进制数n放入M地址。

由以上规定可知，用汇编程序语言表示的上列表达形式，意味着X地址的内容1与Y地址的内容2相加，将相加结果置入Z后停止。为了用框图来理解汇编程序的具体操作过程，下面再用计算机语言来表示这个例子。

规定计算机语言中，一个字由16位构成，如图1-1所示，用由最末位起每三位划界的八进制数来表示。正确地说，在计算机用语中，一个箱子叫做一个字（1Word），而计算箱子容量的单位叫做位（Bit）。

```
地址   内容
100    060300（程序从 100₈ 地址开始，且 LDA、ADA、
              STA 分别对应于 06₈、04₈、07₈）●
```

● 脚标8表示该数是8进制数。下同。——译者注



8进数	2进数
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

图1-1　由16位构成的一个字用8进制数来表示

101　040301

102　070302

103　102000

300　000001（X地址、Y地址、Z地址分别对应于300_8、301_8、302_8）

301　000002

302　000000

上列计算机语言程序中的各地址100_8、101_8、102_8、103_8、300_8、301_8、302_8是按照编译程序对应于箱子名称A、B、C、D、X、Y、Z适当分配的。

1.2　计算机的操作

把上节的计算机语言指令存入图1-2所示的主存储器。图1-2中诸寄存器各具有如下的作用。

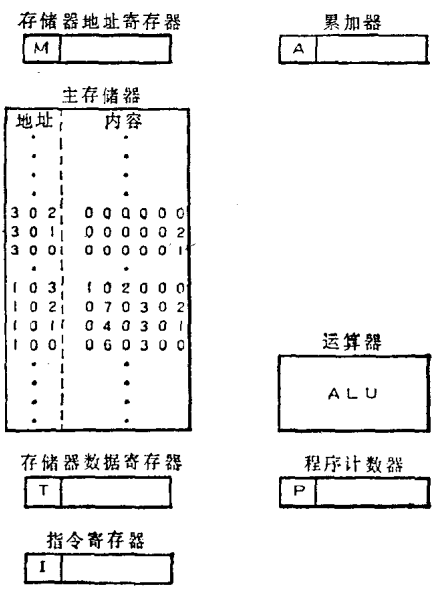


图1-2　计算机的基本部件

P寄存器：程序计数器。指定程序的顺序。

M寄存器：存储器地址寄存器。指定主存储器的地址。

T寄存器：存储器数据寄存器。存放主存储器的内容。但究竟存放哪一个

地址的内容，由M寄存器指定。

I寄存器：指令寄存器。存放实际执行的指令。

A寄存器：累加器。即上节所说的寄存器。

再无其它寄存器。ALU(Arithmetic and. Logic Unit)是运算器，是进行算术运算和逻辑运算的部件。

由以上各部分的作用简单地说明 1+2 变成 3 的过程。

i）取对应于 LDA X 的指令（图 1-3）

① 置 100_8 到P寄存器，使它工作。
M寄存器接受P寄存器的内容。

② 在主存储器中，按M寄存器的内容指定 100_8 地址。

③ T寄存器接受 100_8 地址的内容 060300_8。

④ I寄存器接受T寄存器内容作为执行指令。

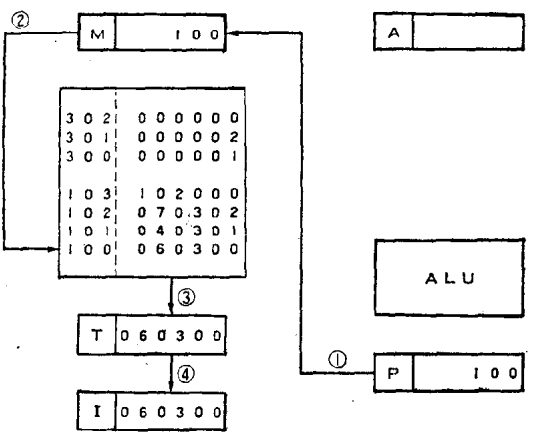


图1-3 LDA X 的取指令阶段

图 1-3 表示上述的操作过程。这是一个为了取指令的定义符操作，通常叫做取指令阶段（Fetch Phase）或第 I 阶段。

ii）执行对应于取来的LDA X 的指令（图 1-4）

（I）在执行 I 寄存器的指令之前，为了给取下一个指令作准备，先把 P 寄存器的内容加 1。

（Ⅱ）I寄存器的内容 060300_8 被解释为将 300_8 地址的内容放入A寄存器的指令，将操作数地址 300_8 送到M寄存器。

操作数地址也叫做放入运算数据的地址，随计算机的种类不同而异，在有的计算机中，这个地址由T寄存器给出。

（Ⅲ）在主存储器中，按M寄存器的内容指定 300_8 地址。

（Ⅳ）将 300_8 地址的内容 000001_8 放入T寄存器。

（Ⅴ）由ALU（运算器），取T寄存器的内容 000001_8 与 000000_8 作“或”运算，也有的计算机进行加法运算。不论是“或”运算还是加法运算，数据 000001_8 不变。

（Ⅵ）将运算的结果 000001_8 由ALU送入A寄存器。

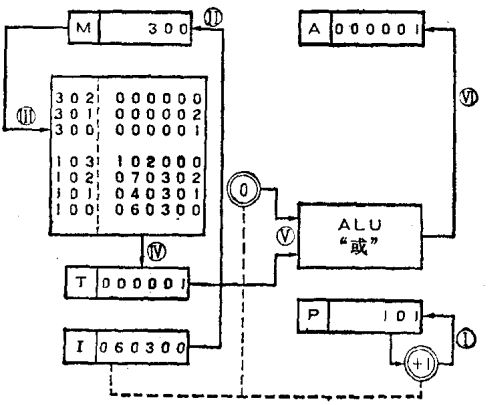


图1-4 LDA X的执行指令阶段

图 1-4 表示上述操作过程。这是执行指令一次的操作，通常叫做执行指令阶段（Execute Phase）或第Ⅱ阶段。

iii）ADAY的取指令阶段（图1-5）

① 将P寄存器的内容送到M寄存器。

② 在主存储器中，按M寄存器的内容指定 101_8 地址。

③ T寄存器接受 101_8 地址的内容。

④ I寄存器接受T寄存器的内容作为执行指令。

上述取指令阶段如图 1-5 所示。

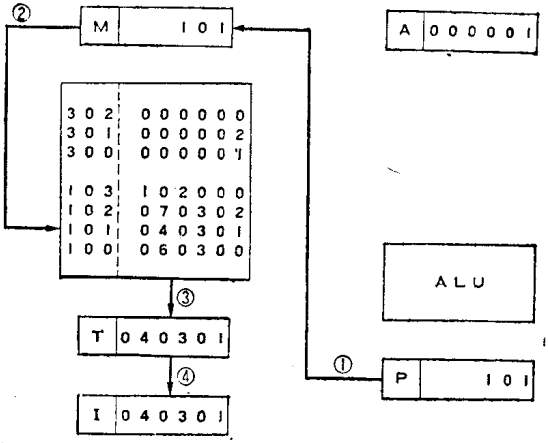


图1-5 ADA Y的取指令阶段

iv) ADA Y的执行指令阶段 (图1-6)

(I) 将P寄存器的内容加 1。

(II) I寄存器的内容 040301_8 被解释为将 301_8 地址的内容与A寄存器的内容相加，并放入A寄存器的指令，将操作数地址 301_8 转送到M寄存器。

(III) 在主存储器中，按M寄存器的内容指定 301_8 地址。

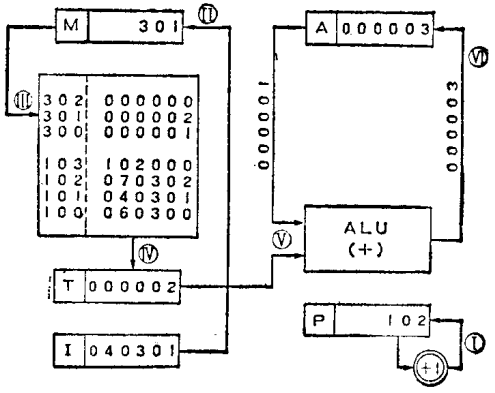


图1-6 ADA Y的执行指令阶段

（Ⅳ）将 301_8 地址的内容 000002_8 送入T寄存器。

（Ⅴ）T寄存器的内容 000002_8 与A寄存器的内容 000001_8 一起在 ALU 中相加。

（Ⅵ）运算结果得 000003_8，由 ALU 送入A寄存器。

上述执行指令阶段示于图 1-6

v）STA Z 的取指令阶段（图 1-7）

① 将P寄存器的内容送到M寄存器。

② 在主存储器中，按M寄存器的内容指定 102_8 地址。

③ T寄存器接受 102_8 地址的内容。

④ I寄存器接受T寄存器的内容作为执行指令。

上述取指令阶段如图 1-7 所示。

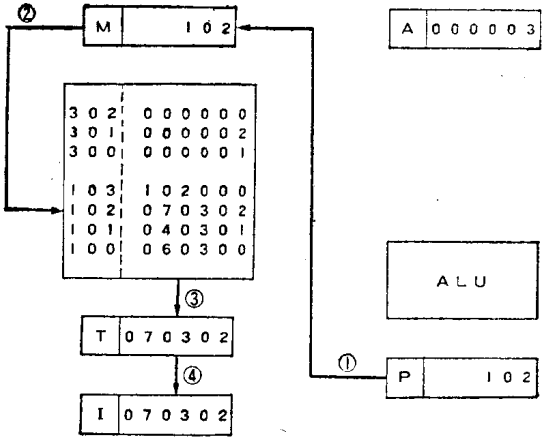


图1-7 STA Z的取指令阶段

vi）STA Z 的执行指令阶段（图 1-8）

（Ⅰ）将P寄存器的内容加 1。

（Ⅱ）I寄存器的内容 070302_8 被解释为将A寄存器的内容写入 302_8 的指令，操作数地址 302_8 转送到M寄存器。

（Ⅲ）在主存储器中，指定 302_8 地址为M寄存器的内 容。

（Ⅳ）为了将A寄存器的内容 000003_8 写入主存储器而送到T

寄存器。

（Ⅴ）将T寄存器的内容 000003_8 写入主存储器的 302_8 地址。

以上执行指令阶段如图 1-8 所示。

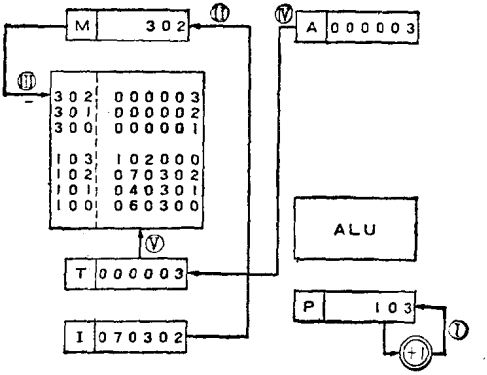


图1-8 STA Z的执行指令阶段

vii）HLT 的取指令阶段（图 1-9）

① 将P寄存器的内容送到M寄存器。

② 在主存储器中，按M寄存器的内容指定 103_8 地址。

③ T寄存器接受 103_8 地址的内容。

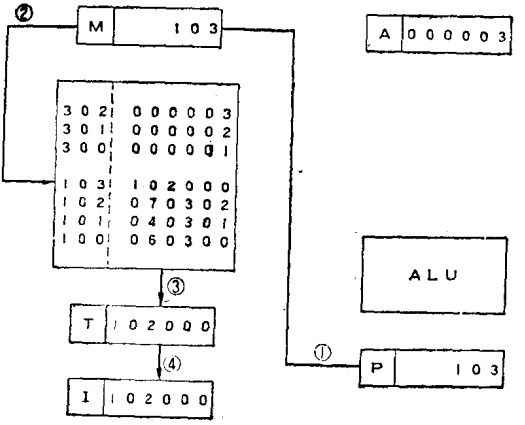


图1-9 HLT的取指令阶段

④ I寄存器接受T寄存器的内容作为执行指令。

以上取指令阶段如图 1-9 所示。

viii) **HLT 的执行指令阶段**（图 1-10）

（I）P寄存器的内容加 1 。

（II）I寄存器的内容 102000_8被解释为使计算机的执行操作停止的指令。程序停止执行。

以上执行指令阶段示于图 1-10 。

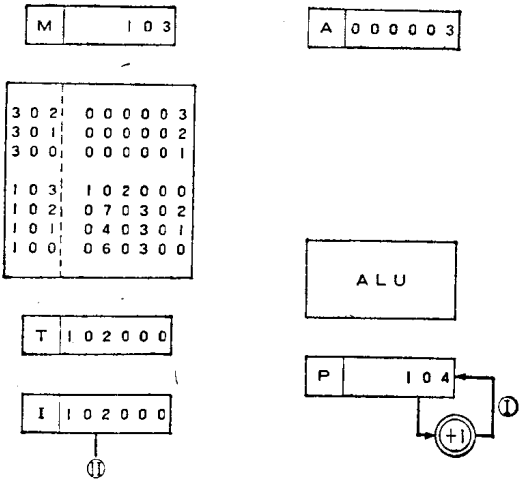


图1-10 HLT的执行指令阶段

第 2 章　总线和控制

2.1　用总线作公用通道

现在考虑图 1-3～图 1-10 中表示计算机操作过程的箭头。

以图 1-5 和图 1-6 所示的加法指令 ADAY 的取指令阶段和执行指令阶段为例，二者可一同表示在图 2-1 中，用脚标 f（取）和 e（执行）来区别。

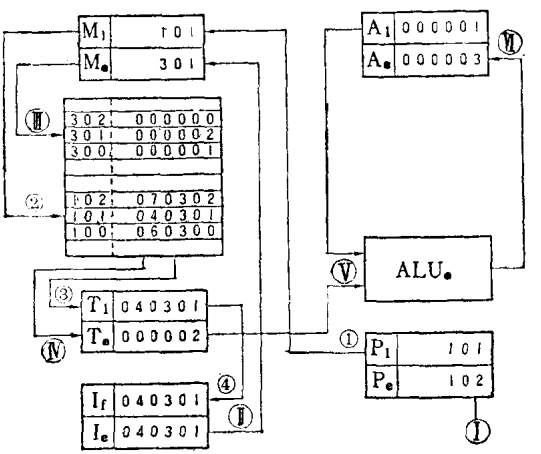


图2-1　加法指令的取指令阶段和执行指令阶段

图 2-1 中的箭头是连接寄存器与寄存器的通道，有规律性（其他指令也一样）。因此，走同样通道的线汇总为总线（Bus）。总线又分 S 总线、R 总线、T 总线。图 2-1 经整理后，用总线表示就变成图 2-2。这样，图中存储器与M寄存器和T寄存器的关系就可分割为总线与存储器的控制操作。

这样一来，计算机的操作过程就能用进、出总线和存储器及