

计算机技术

JI
SUAN
JI
SHU

1
1979

上海无线电十三厂

计 算 技 术

1979年第1期

总第34期

目 录

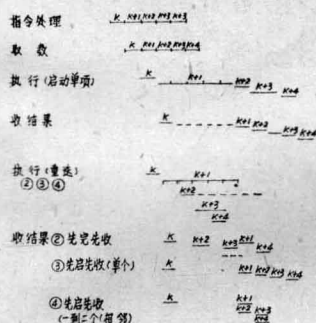
1. 在多个执行单元重迭工作的计算机中实现顺序接收结果	1
2. 向量计算机	5
3. TQD——1高速模数转换器	11
4. 4位微型计算机概述	21
5. 流集估算分析——试述不丢失信息的必要充分条件	27
6. 模块化并发程序设计的体会	33
7. 现代半导体存贮器	39
8. 微型计算机程序编制方法	57

在多个执行单元重迭工作的计算机中实现顺序接收结果

设计科 陈国尧

在流水线计算机中,一般浮点运算的执行时间是大于指令处理的时间。为了使执行速度跟上指令处理的速度,通常采用多个专用执行单元(如变址量运算,浮点算术运算,逻辑运算和移位等),它由指令处理相继启动后,同时进行工作。但由于各单元执行速度不相同,有可能后面指令提前执行结束(例如浮点加法后的修改变址指令),送出结果。因此在发生内中断时,就可能有前面的指令未做完而后面指令已先完成的情形。参看图一中的例2,这样使中断点上各寄存器的信息有错。处理中断后,不能立即从断点做下去,国外的CDC—6600/7600和IBM.360.91/195²等机器都有这个问题。

要避免上述问题,最简单的方法是在任何时间只允许执行一项运算,待它执行完毕送出结果后,再启动下一项运算。但这样做就取消了多个执行单元重迭运算的功能,使执行速度下降。参看图一所示工作时的例1。



图一 重迭执行的时序图

从上面分析可知,内中断时寄存器中信息出错的原因是:多条指令同时执行时,后面指令先执行完就先送出结果所造成。倘若规定执行器送出结果要严守次序,即先启动发出;后启动的运算,即使先完成也不能送出结果,要等前面指令的执行结果送出后,它才能送出结果。参看图一中例3。其次要规定,在某执行器出错后,它不能立即发内中断请求,而是建立出错标志。当轮到它送结果时,就不送结果而改发内中断请求,从而建立中断。这样,中断后寄存器内容就不会有上述错误而保持正确。

下面要研究实现顺序接收结果的具体方法。

大家知道,在流水线计算机(如CDC·6600)中,当某一指令发往执行时,在启用相应执行单元 E_n 和送去二个操作量的同时,还要把目的寄存器 R_i 的编号 i 送 E_n ,并把 R_i 所附更新位 gi 置1。当执行完成后,执行单元按照其所附编号 i 送结果到寄存器 R_i 和清 gi 位。(如果 R_i 接收结果是用(1BM—360/91的)特约号符合方式,则 R_i 旁要附加输入特约号 e_n ,它存放执行单元编号 e_n 。当某执行单元 E_n 送结果(E_n)到结果总线时,还附上其编号 e_n 。结果总线上的 e_n 位跟各待收寄存器(其 $gi = 1$)所附 e_n 相比较,若有符合,则 R 输入总线内容。)图2a是其流程图。

为了使执行结果的输出序列保持正确,指令处理后启动执行单元时要附送启算序号(CA),这由启算计数器CA提供。另外在寄存器组的结果接收总线上也设一个结果序号计数器 C_z 。它们的最大计数值 N_c 应大于执行单元的个数 N_E 加上在它前面或后面的缓冲级的总数 N_B ,即 $N_c (N_E + N_B)$ 。例如 N_E 加 N_B 小于15,则 C_A 和 C_z 用4位已够。如超过15(但小于31)要用5位。开始时令 $(C_A) = (C_z)$ 。当指令经过处理并发往相应执行单元 E_n 时,还附着送去 (C_A) 值,即 $(C_A) \rightarrow C_{En}$;然后把 C 空加一,即 $(C_A) + 1 \rightarrow C_A$ 。

当执行单元做完送出结果到寄存器时，要先把它所附的 C_{En} 跟结果总线的 (Z) 相比较，看是否相等；如果 $(C_{En}) = (C_Z)$ ，则执行结果 (E_n) 可以联接到结果总线，再送往目的寄存器 R_i 。同时 C_Z 内容加一，即 $(C_Z) + 1 \rightarrow C_Z$ 。若 $(C_{En}) \neq (C_Z)$ ，说明 (E_n) 是属于后启动先完成的结果，它不能送入寄存器。要等前面指令的执行结果逐个送入对应的寄存器，同时 (C_Z) 也随着增值，直到 $(C_{En}) = (C_Z)$ ，才把 (E_n) 送入 R_i 。

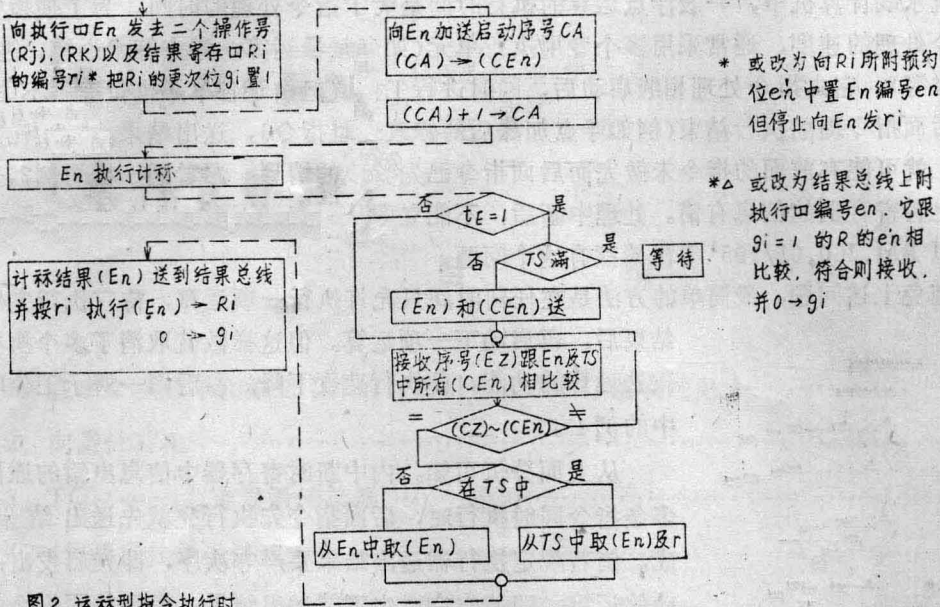


图2 转移型指令执行时的流程图

按照上面所述执行结果 (E_n) 是经过总线直送寄存器组的，若遇到 $(C_{En}) \neq (C_Z)$ ，则 (E_n) 就送不出去而滞留存执行器 E_n 内。这时若有新启指令要用 E_n ，则由于 E_n 仍被占用而停止启发新指令。为了消除这个缺点，在结果总线上添设结果缓冲站 TS 。凡是执行时间为一个拍的单元，其结果附上目的寄存器编号 r_i 及序号 CE_n 送入 TS 。其他执行单元的结果仍直送结果总线。因此 C_Z 要跟 E_n 中 C_{En} 和 TS 中所有 CE 相比较，有符合就把对应数送到其 r_i 所指定的寄存器 R_i 。空出的 E_n 或 TS 中单元就可以接受新内容。因实现顺序接收结果而使工作流程扩充的部份参看图 2b。

如果让所有执行结果都先送 TS 再转送 R 。在遇到一条执行时间 t_e 较长的指令 I_1 后跟上多条 t_e 短的指令时，后面指令的结果可能把 TS 送满，到指令 I_1 得出结果时就送不进 TS ，但它的序号 C_{E1} 在 TS 中的 C_E 之前，这样 TS 的内容也出不来，造成死结。若是 TS 中缓冲站数 n_{TS} 大于最长执行时间 t_{Emax} (拍数) 就能免除死结，但这样 TS 器材较费，还要使 CA, C_E 位数增大。现用方法也能避免死结，且 TS 器材少。其他方法还有，但这里从略。

对取数指令，为保证顺序，取数也要通过 TS 。因此在发访存地址的同时把 r_i 及 CA 也送入 TS 。待数取到 TS 后，再按序送寄存器 R 。

采用启/收计数器整序的多执行单元计算机的框图可以参看图 3。从图可以看到，添加器材有启动执行的序号计数器 CA ，把它发到各执行单元时，保存它的寄存器 C_E ，和控制接

收结果的序号计数器 C_z 。和结果缓冲站 TS。以及多个符合线路供 C_E 跟 C_z 比较用的。

结果缓冲站 TS 可由多个寄存器组成，如图 3 所示，也可以用寄存器堆做，它只能存放结果及 r_i ，而所附序号 C_{Ei} 要放在寄存器中， C_{Ei} 各联到一个符合线路以跟 C_z 相比较，若有符合则取出对应数，并清除占用位。新结果的输入地址可用一个计数器指出，还要一个全满指示位。只要未全满，则遇到输入计数器所指单元已占用时，计数器能继续加一，直到指向未占用单元为止。参看图 4。

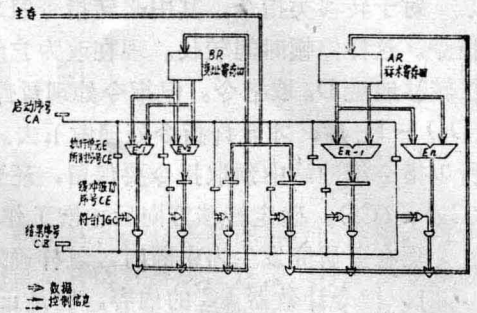


图3 实现顺序接收结果的逻辑框图

在有多组寄存器的计算机中，上述 TS 可以每组寄存器设置一个，这样设备较费，但输入/出联线较少。若多组寄存器合用一个 TS，则其输入要输到所有执行单元，输出要联到所有寄存器组，故联线多。为区分各组寄存器，在寄存器编号 r 中要添加组号，增加位数，输出选地也复杂些。

现在研究采用上述整序措施后所引起的速度损失及其补偿方法。第一种情况是用 TS 后，如果遇到新启指令要用寄存器 R_i 中内容，但它还在 TS 中并且还未轮到输出，这时怎样取用它。解决方法是在 TS 中另设一路输出线，专门通向执行单元。它的选址是用新启指令的 R_i 号 r_i (或 R_i 所附预约号 e') 跟 TS 中所有 r 项相比较，若有符合，则取出读单元到 E。因此 TS 的所有 r 项也要联上符合门用作第二路选址。它跟用 C_z 符合取数所不同之处是它取用 TS 内容后不注销它(清占用位)。若 R_i 内容还在执行器 E 中，那末对于用预约号 e 接收结果时，可把 R_i 的 $e'n$ 转发到执行单元的预定站，当 E_n 有输出时，其编号 e_n 跟预定站中待收单元的 e' 比较，有符合则 (E_n) 直送预定站。虽然送往 R 还要通过 TS 顺序送 R_i 。

第二种情况是指令串中出现执行时间长的指令后跟着几条时间短的指令。这时后者都被推迟输出而拉长工作时间。现在介绍一个补偿方法，即允许缓冲站 TS 中二个序号相邻的结果可以同时送出，这样能让 TS 中积层的存数加速送出，如图一中例 4 所示。实现方法是在 TS 中对每个 C_E 另加一个符合线路，它控制 TS 的第二路输出门，在 C_z 上添加一个加一线路，让 (C_z) 及 (C_z) + 1 同时跟 TS 中所有 C_E 项进行比较；当 (C_E) 或 (C_z) 及 (C_z) + 1 都有符合，就从 TS 中取出一个或二个结果到寄存器。在送出二个结果时，(C_z) + 2 $\rightarrow$ C_z 。但 C_z 项不符合，即使 (C_z) + 1 项符合也不能出结果。图 4 是它的逻辑框图。为了同时输入二个结果，寄存器也要二组输入门，故结构较复杂。如改为二个寄存器组可同时接收二个相邻结果，则结构可以少复杂些，参看图 4。

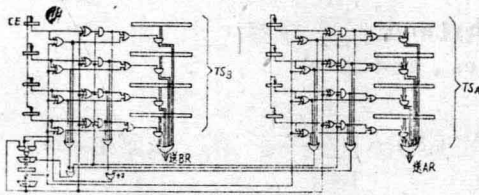


图4 结果缓冲站：可同时取出结果。

采用上述措施后，对某些指令的执行有所影响。例如送存指令，原来是指令处理后，地址即送往访存排队，排上后把数送往存储器，而现在是把地址送缓冲级，并附上 C_A ，等它跟 C_z 符合后，才能把地址送去排队，排上后才送存数，并把 C_z 加一。排不上就要等，而后面指令的结果也不能送出。

对于转移类指令，其中必转指令按规定要等前面指令接收结果后，才能执行转移并取新指令。这样停顿时间较长。现在改为它经过指令处理后，先清除指令缓冲站剩余内容，然后按转移地址 D_B 取指令。但指令处理暂停工作，要等转移地址的 C_E 跟 C_Z 符合，执行转移 $(D_B) \rightarrow PC$ 后，才允许指令处理做下去。而对于条件转移指令，它要从二个方向予取指令，所以指令缓冲站中预取指令要保留，还要往转去方向取指令。至于指令处理也要停下来，等 $(C_E) = (C_Z)$ ，确定转移方向后才恢工作。

现在要研究发生内中断时，怎样确定断点。为此需要回收下列参数：

1. 指令计数器 PC 的内容，它是正在访存取到指令缓冲站的指令地址。
2. 予取指令条数 L_i ，它包括指令站中和正在访取的指令。
3. 启动执行序号 C_A 。
4. 接收结果序号 C_B 。

从1, 2, 项 $(PC) - L_i$ 可得到指令处理点的指令地址。从而 $(C_A) - (C_Z)$ 可得到还在执行中的指令条数，参看图5。

根据这些参数就能确定 C_Z 点的指令序号 PC_E

$$(PC_E) = ((PC) - L_i) - ((C_A) - (C_Z))$$

当执行中主机发生出错(不管是访存或运算)都在数码傍附上出错位，到接收结果时建立内中断。因此中断点 PC_I 就是 PC_E 。

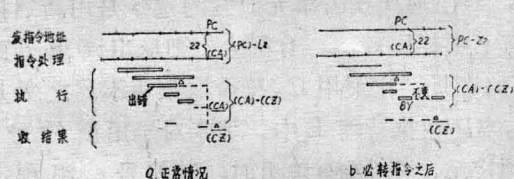


图5 流水线时序图

但是，在指令处理时遇到必传指令并停下来后，但指行单元还在做转移前的指令，而指令站已在予取新的指令，这时发生内中断时怎样确定断点？

为了先要确定必转指令地址，这就需要保留原来 PC 和 L_i (尽管予取指令都丢弃)。

其次指令处理遇必转指令后要置必转标志 $bu = 1$ ，则上面算式中的 PC 和 L_i 都要用保留值。

在遇到条件转移指令时，因指令站是按不转方向予取指令，而从转移方向取来的是另外存放的。故计算断点跟一般指令相同。

参 考 资 料

1. J.E.Thornton
Design of a Computer The Control Data 6600
2. D. W. Anderson
IBM System 360—91, Machine Philosophy and Instruction Handling IBM Journal of Res. and Dev. v—11. PP8—27, 1967. NO.1.
3. R.M Tomasolo
An Efficient Algorithm for Exploiting Multiple Arithmetic Units. IBM Journal of Res. d Dev. v—11. PP 25—33(1967. NO.1)

向量计算机

设计科 陈行祥

为了提高计算机的速度, 更多的发挥重叠操作, 流水线和并行多功能部件的作用, 对于某些矩阵或可进行重复叠代运算的题目采用了称为向量运算的计算机。这是一种单指令多数数据流的计算机, 目前要使速度超过数千万次/秒以上的机器均具有这种性能, 如TI的ASC, STAR—100, ILLIAC IV, CRAY—1 等。现对下列几个问题进行讨论。

- ①向量计算机的优点。
- ②向量计算机需要增加的参数和硬件。
- ③向量计算机的局限性及需要注意点。
- ④向量计算机的优点。

向量不是物理上力学力的向量概念, 而是表示一串序列数据, 向量运算就是一串序列数据与另一半序列数据的运算。

例如:

(i) 传送 $A \rightarrow B$ 即 $a_1 \rightarrow b_1, a_2 \rightarrow b_2, a_3 \rightarrow b_3 \dots a_n \rightarrow b_n$ 。

(ii) 四则运算: $A + B \rightarrow C$ 即 $a_1 + b_1 \rightarrow c_1, a_2 + b_2 \rightarrow c_2 \dots a_n + b_n \rightarrow c_n$ 。

(iii) 向量和及内积。向量和 $\sum a_i \rightarrow c, \sum a_i = a_1 + a_2 + \dots + a_n \rightarrow c$, 内积 $\sum a_i b_i \rightarrow c, \sum a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n \rightarrow c$

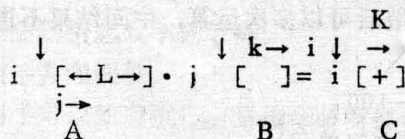
$\sum a_i^2 \rightarrow c, \sum a_i^2 = a_1^2 + a_2^2 + \dots + a_n^2 \rightarrow c$

(iv) 平均 $\frac{1}{2}(A + B) \rightarrow C$ 即 $\frac{1}{2}(a_1 + b_1) \rightarrow c_1 + \frac{1}{2}(a_2 + b_2) \rightarrow c_2 \dots \frac{1}{2}(a_n + b_n) \rightarrow c_n$

也可 $\frac{1}{2}(a_i - 1 + a_i + 1) \rightarrow c_i$

(v) 比较 $A \sim B, a_1 > b_1, a_2 > b_2 \dots a_n > b_n$

(vi) 卷乘与卷加 · 卷乘 $\sum_{j=1}^L A_{ij} B_{jk} \rightarrow C_{ik}$



卷加 $\sum_{j=1}^L |A_{ij} + B_{jk}| \rightarrow C_{ik}$

(vii) 多项式 $\sum_{j=1}^{L_A} A_j (B_i)^{L_A-j} \rightarrow C_i \quad i = 1 \sim \min(L_C, L_B)$

$C_i = a_1 b_i^{L_A-1} + a_2 b_i^{L_A-2} + \dots$

(viii) 部份矩阵乘

$$\sum_{j=1}^{L_A} A_j B_{j-1} L_{A+j} \rightarrow C_i \quad i=1 \sim \min L_C$$

$$L_B = m L_A \cdot L_A \cdot$$

$$C_i = a_1 b_{i-1} L_{A+1} + a_2 b_{i-1} L_{A+2} \cdots \cdots \leftarrow$$

$$m \updownarrow C = \updownarrow A \updownarrow B$$

$$\cdots \cdots + \cdots + a L_A b_{i-1} (L_A + L_A) \quad \begin{array}{l} A \text{ 为一维向量} \\ B \text{ 为二维矩阵} \end{array}$$

另外当一个题目可以分解成具有相同处理过程的重复叠代而相互各分量间又互不相关时，那么同样也可以用向量处理机来进行解算。因此一个串行计算的题目就可以变成一定次数的平行运算。这种算题特别适合于流水线和多功能部件的计算机，它能既重叠又平行，所以向量计算机的优点之一是能更好地发挥重叠和平行的功能，即取操作数与执行过程重叠，在不发生存储冲突情况下取操作数可以不化时间，省却了标量的变址运算，减少了执行部件的瓶颈口，同时也解决了执行加工与变址的矛盾。

向量计算机的另一优点是一条指令可以执行多个数据，因此减少了指令，提高了速度。例如求 $a_i + b_i \rightarrow c_i$ ，原来要求取 a ， $+b$ ，送 c ，变址，测试及转移要五条指令才能得到一个结果，而向量运算仅要一条指令就可以得到全部结果。不过并非所有算题均能符合向量运算的要求。对于有相关的不适用于向量运算的题目要加快速度只能用多指令多数据流（多机系统去解决，向量运算还要增加不少设备如 ILLIAC IV 要 64 个 PE，（器材太多），STAR—100 要宽字四条总线来提高存储器流量（相对于 STAR—100）更好地发挥平行重叠的功能，国内有人提出了所谓纵横加工向量机（北京所 757 机①它相当于国外 CRAY—1 的缩小型）。

例如要求计算 $(x_i + a)^2 \rightarrow y_i, i=0 \cdots \cdots L-1$ 可以有三种加工方式

(a) 一般单指令单数据流（标量）机器，为横向加工（用：表示清加 ↑ 表示平方）

则计算过程如下：

$$: x_0, + a, \uparrow, \rightarrow y_0,$$

$$: x_1, + a, \uparrow, \rightarrow y_1,$$

.....

横向加工的优点是内存取出数后可以多次运算，中间结果不进行存取存储器，但缺点是每次运算有依赖于上次结果。

(b) 纵向加工：如 STAR—100

先： $x_0 + a \rightarrow L_{00}$ （内存作为中间单元）

$$: x_1 + a \rightarrow L_{01}$$

⋮

其次 $L_{00} \uparrow \rightarrow L_{00}$

$$L_{01} \uparrow \rightarrow L_{01}$$

再次 $L_{00} + y_0 \rightarrow L_{00}$

$$L_{01} + y_1 \rightarrow L_{01}$$

纵向加工的优点是可以平行无关化，但中间要进行存取存储器增加了对存储器流量的需要。

(c) 纵横向加工：CRAY—1 及国内 757 机

求解 $(x_i + a)^2 \rightarrow y_i$ 以 CRA—1 机为例

先纵向加工 $x_0 + a \rightarrow L_{00}$

$x_1 + a \rightarrow L_{01}$

$\vdots$

$x_{63} + a \rightarrow L_{063}$

其次横向加工

$L_{00} \uparrow \rightarrow y_0$

$L_{01} \uparrow \rightarrow y_1$

$\vdots$

$L_{063} \uparrow \rightarrow y_{63}$

再执行纵向加工

$x_{64} + a \rightarrow L_{00}$

$x_{65} + a \rightarrow L_{01}$

$\vdots$

$x_{127} + a \rightarrow L_{0163}$

其次横向加工

$L_{00} \uparrow \rightarrow y_{64}$

$L_{01} \uparrow \rightarrow y_{65}$

$\vdots$

$L_{063} \uparrow \rightarrow y_{127}$

.....

利用纵横向加工交替则在纵向加工时可以高度平行无相关，而在横向加工时，则可以减轻存储器流量，可以使并行重叠的优点充分发挥，但是这要靠增加大量寄存器组来解决(757机为 $8 \times 16 \times 64$ 位，CRAY—1 为 $8 \times 64 \times 64$ 位)且当向量长度超过 16 (757机)，64(CRAY) 时，由于要重新启动新的运算过程必然引起效率的下降。

② 向量计算机需要增加的参数和硬件。

从上面介绍向量运算的例子可以知道执行向量指令必需提供许多额外讯息(参数)。

(i) 二个源向量和一个结果向量的起始地址(或基地址)。

(ii) 向量中那些分量要进行操作控制称步长(三个向量要三个)

(iii) 向量操作分量的长度称向量长度(三个向量要三个)。

但一般以结果向量长度为准

$L_{源_1} \quad L_{源_2} > L_{结果}$

到 $L_{结果}$ 结束

$L_{源_1} \quad L_{源_2} < L_{结果}$

到 $L_{源_1}, L_{源_2}$ 后，填另

(iv)为了减少运算节约内存,对稀疏向量要去除另元素,因此要有稀疏尺或称有序向量寄存器来指示所选取的分量存入存储器进行压缩。

例: $a = 00a_300a_500a_8a_9000\cdots\cdots\cdots$

压缩后 $a = a_3a_5a_8a_9$

提供这些额外讯息(参数)有各种不同方式:

(a) 指令后附参数

手编方便,但指令字长要增加,且不能用于再入式程序,所以一般不使用

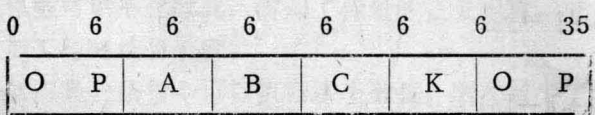
(b) 指令地址指出参数

可以解决再入式程序问题,但要多访问一次内存,对形成指令也较麻烦, MU 5 机采用该方式。

(c)使用通用寄存器或变址寄存器作为向量参数寄存器。

FACOM 230—75 ② AP及STAR—100 ③ 采用该方式。

FACM 230—75 AP 向量指令形式



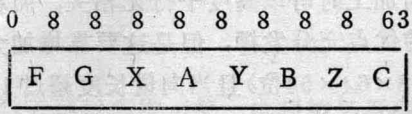
OP 表示操作码

A, B, C, K各带描述字地址进行 $A \leftarrow (B \times c) \cdot K$ 运算, A, B, C, K 的描述字地址从数据寄存器 DR 中取出参数(DR为 36 位 64 个寄存器)

DR:	类 型	数 形 式 变 址
i + 1		基 地 址
i + 2		长 度 L
i + 3		步 长 Δ

K 是屏蔽场指出位向量是“1”或“0”,即表示重新送入运算结果还有保留原值。这种屏蔽是运行一直进行不屏蔽,但结果送与不送受屏蔽控制。这种方式控制简单一些,但增加了不必要的运算时间。

STAR—100指令形式:



F 操作码

G 辅助操作码: 决定操作数和结果的读写总线的用法。

X, A 第一向量 A 基地址和长度 XA 的地址位移。

Y, B第二向量 B基地址和长度YB的地址位移

基地址+ 地址位移= 起始地址

Z, C 第三向量。

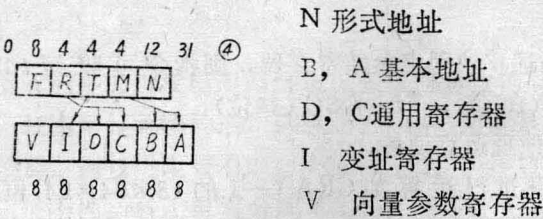
Z 表示控制向量的基地址控制向量的每位指示相应结果分量是否要 存入。“1”表示 要 存入,“0”表示保留原结果。

C 结果向量基地址和长度

C+1:C 和 Z 的位移量。

STAR—100 无步长 Δ ，只能逐个依次变化。若要变化步长只能通过稀疏向量来实现。

(d)用专用寄存器放置向量参数称向量参数寄存器，在指令执行前预先送入，这样要多一条指令，但灵活性好且对再入式程序也有保证，如TI的ASC就为一例。



向量指令执行之前预先将参数放入V内(有辅助操作码，向量开始地址，长度，向量的步长，内循环和外循环长度等等)。在向量指令流向运算部件时，把参数也流入，各个专用寄存器的功能是固定的，可以快速存取而不考虑冲突。V的内容如下③

操 作 码		类 型	长 度
一维	位移 A	基地址 A	
	位移 B	基地址 B	
	位移 C	基地址 C	
	内增量长 A		内增量长 B
二维	内增量长 C		迭 代 次 数
	外增量长 A		外增量长 B
	外增量长 C		迭 代 次 数

迭代次数 = 场长度

在 STAR—100中只能处理一维向量，向量的增量不可变，要变更向量增量仅可通过稀疏向量来实现，指令译码后就由微码进行控制，但在 TI—ASC 中它可以处理二维向量，且各向量的步长可变，但 ASC 没有控制向量，稀疏向量微码仅对运算部件起作用。

从提供向量参数来看 TI—ASC 段 STAR—100为 佳，但 STAR—100 具有向量宏指令(即不产生结果向量而把结果存放在一个或二个寄存器中，它不但控制存与否且可禁止对某些分量进行操作)及稀疏向量，所以指令功能较强。

向量宏指令举例：比较 $A_i > B_i$ 项并把比较对数的项写入 C

当 $A_i > B_i$ 被找出，停止比较，已比较对数写入 C

或当 $A_i = A_L =$ 长度到达长度 L，停止比较，已比较好对数写入 C，若控制向量禁止 $A_i > B_i$ 项比较，则该指令可以继续。

稀疏向量举例：如有一向量 $a = 00a_3 00a_6 00a_8 a_9 0 \cdots \cdots$ 仅 $a_3 a_6 a_8 a_9$ 为 1，其余为 0。将该向量用比较指令全 0 比较，相同为 0，不相同为 1，得有序向量，再将有序向量所选取的分量存入存储器压缩成稀疏向量。有序向量在稀疏向量存在期间保留不变，以便指示各分量在向量中的位置。

因向量计算机所需提供的附加参数及向量机的运算功能的需要, 向量计算机所需增加的硬件是:

(I) 指令字长要增长, 且为三地址以提供向量参数: 向量基本地址, 向量类型, 控制向量, 长度, 增量。

如 STAR—100 字长要 64。但若采用描述字或固定参数寄存器, 则指令可较短 (32 位) 如 FACOM—230—75 AP (36 位), MU 5 (16 位), TI—ASC (32 位)

(II) 要增加专用向量参数寄存器及地址运算设备。

(III) 要增加寄存器组来存放中间结果进行运算如 CRAY—1 的 $(8 \times 64 \times 64)$ 位寄存器组, 国内 757 机的 $(8 \times 16 \times 64)$ 位寄存器组。

(IV) 存储器要宽字, 多总线以满足存储流量需要, 即要有足够源向量操作数读出及其结果的写入或有超高速存储器。

③ 向量计算机的局限性及注意点。

为了更好地发挥向量机的作用, 要满足几个条件。

(I) 要减少建立时间即减少送参数、控制及检查时间, 这在向量长度较短时较为器重, 其次是通过流水线的时间。对于类似国内 757 机或 CKAY—1 计算机向量长度刚超过 16 及 64 时, 要重新启动新的运算, 则必然引起处理能力的下降。

(II) 对于不连续地址存取的向量运算, 若不用超高速缓存则降低速度, 这是由于 STAR—100 及 TI—ASC 等计算机一次从内存取 4/8 字但实际只用一个字, 所以速度受到损失 (FACOM 230—75 AP 因每次取一字故没有这个问题)。如向量运算中的卷乘。

$$\sum_{j=1}^L A_{ij} B_{jk} \rightarrow C_{ik} \quad i \downarrow \quad \begin{array}{c} \xrightarrow{L} \\ [j] \\ A \end{array} \times j \quad \begin{array}{c} \downarrow \\ [B] \\ B \end{array} = i \quad \begin{array}{c} \xleftarrow{K} \\ [C] \\ C \end{array}$$

B 中向量地址不连续

若 B 向量步长为 2 的整倍数时将引起存访冲突, 也将使取数率下降。

(III) 向量机的编译程序及算法均要适应平行运算的要求, 否则不能发挥特长, 快速算法在向量机上不一定适用。⑥

参 考 文 献

① 高庆狮等: 1025 机主机总体功能设计报告。

② FACOM 230—75 アレイプロセッサハードウェア 解説编 (富士通)

③ (Computer Survey) v09. N+1 March 1977.

PP 75—102 "Pipeline Architecture" C.V. Ramamoorthy and H.F. Li

④ 出国参观考察报告—美国计算机 74 年 8 月

⑤ 电子计算机动态 74 年 4 月。

⑥ "Star Trex" Tad Kshi and Tim Rudy (Compcon 75—10 th IEEE Computer Society International Conference) Feb. 1975. PP. 185—188.

一、TQD—1 高速模数转换器

(一) 引言

以前我们参加过权电阻网络电压负反馈模数转换器设计和调整。倘若有高速度,高精度比较器,权电阻是无感的话,那么 A/D 编码速度是高的;但是由于精密电阻种类多,例如 11 位转换器,则权电阻要有 $2^{n-1}R$ ($n=1, 2, \dots, 11$) 种,精密电阻生产难度大,另外由于权电阻阻值不同,各位电流分配不均匀,网络开关饱和压降调试工作量很大。对于 T 型网络 A/D,权电阻种类虽少,但是有转换速度低的缺点。认为以上二种形式 A/D 都不适合 TQD—1 的高速 A/D。

TQD—1 高速 A/D 不但有精度要求,而且还有速度要求,因为 TQD—1 高速模入采用双发射极晶体管做采样开关,为了提高部件抗共模能力,电子开关输出接隔离变压器,为了不使变压器输出波形平顶跌落太多,采样开关闭合时间不能太长,我们决定 $50\mu s$, $50\mu s$ 闭合时间,扣除输入变压器上升边 $20\mu s$,余下来时间给 A/D 转换,而希望在 A/D 转换过程中,输入信号维持不变;由于变压器平顶下降之故,事实上是不可能的。如果 A/D 转换时间越短,变压器平顶下降对 A/D 转换精度影响就越小。对于带有取样保持电路的 A/D 转换也有速度要求, A/D 转换速度越快,保持电容器上信号跌落对 A/D 精度影响越小,因此要求提高 A/D 转换速度。我们参阅一些国外资料,分析了日本横河 73 年产品 YODIC—100 计算机中 A/D 方案,认为有参考价值,根据国内现有元件情况,经过反复试验,在 YODIC—100 A/D 基础上,作一些修改,现已设计成功,第一台已为 TQD—II 配套。这种方案还有许多不足之处,有待以后改正。

(二) 技术指标

① 转换电压范围: $\pm 1V$

② 二进制 11 位,最高位为符号位,10 位数码位,当符号位为“0”时,二进制代码为原码,符号为“1”时代码以补码表示。

③ 精度: $\leq 0.1\% \pm$ 最后一位

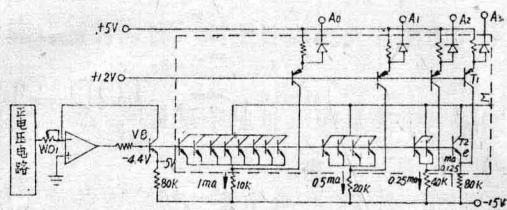
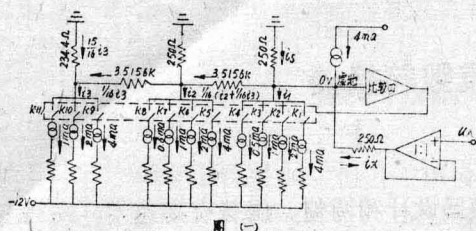
④ 编码时间: $\leq 12\mu s$

⑤ 转换速度: 75000 次/S

(三) 简单工作原理

电路框图如图(一)所示:

把 u_x 模拟量信号转换成 11 位二进制信息, $i_x = u_x / 250\Omega$, 当 u_x 输入某一电压值时,能有相应的 i_x , 初始状态,解码网络电子开关 k_1 到 k_{11} 都是断开,即 $i_1, i_2, i_3 = 0$, 时序节拍脉冲把 $k_1 \sim k_{11}$ 逐一接通, $i_s = 4^{m-1} i_x + i_1 + 1/16 i_2 + 1/256 i_3$, 当某一 k 接通时, $i_s \leq 0$, 这时比较器输出一个脉冲,把这一 k 断开,对应这位为“0”,若某一 k 接通, $i_s \geq 0$, 则比较器没有脉冲输出,这个 k 仍接通,对应这位为“1”,这样一直比较到 k_{11} 为止,就把一个 u_x 电压值,转换成相应的 11 位二进制代码。



(四) 主要单元电路

1. 恒流源开关电路(即数——模网络)

恒流源开关电路由恒流源及开关电路组成,恒流源电路产生 4mA、2mA、1mA、0.5mA 电流,开关电路是控制恒流源是否流到相加点。

①YODIC——100 恒流源开关电路

四位恒流源开关电路如图(二)所示:

A_0, A_1, A_2, A_3 为各电流开关控制端, 当 A_3 为低电平时, T_1 截止, T_2 导通, 因 T_2 的 e 极电位为 $-5V$, 所以 $0.125mA$ 电流流向 Σ 点, 相反 A_3 为高电平时, T_1 导通, T_2 截止, Σ 点无电流。其他三位控制原理是一样的。

图(二)虚线内是集成块电路, 温度均匀, 因此温漂小, 而且又根据每位恒流源大小, 恒流源电路中分别采用1只、2只、4只、8只并列使用, 这样每只管子流过电流是相等的, 都是0.125 mA 电流, 保证了各只管子 V_{BE} 一致, 因为 V_{BE} 对于不同 I_E 是不相同的。另外为了消除 V_{BE} 的温度漂移, 用反馈放大器将 V_B 保持恒定, WD_1 电位器是用来调整基极电压变化量。例如: V_B 由于某原因, V_B 下降, V_E 下降, I_E 减少, I_C 也随着减少, 放大器倒相输入端电压上升, V_B 也上升, 保证 V_B 稳定在一个恒定值。

②TQD—1 高速 A/D 的恒流源开关电路

四位恒流开关电路如图(三)所示:

根据国内元件生产情况,没有这样四个电流开关集成电路,为此,我们曾经同有关元件厂联系过,认为目前生产有困难,因此只能立足于现有元件,采用分离元件组装,在电路上作一些改动。为了保证恒流源稳定性,用 5G24 接成负反馈放大器,使得 L 点电位稳定, R 两端电压稳定在 6V, $R=1.5K$, 可得电流 4mA, $R=3K$ 、6K、12K, 分别得到恒流电流 2mA, 1mA, 0.5mA。5G24 输出接场效应管, 场效应管栅流等于 0, $I_S=I_D$ 。T₂ 是高 β 管 ($i_C=1mA$, $\beta>2000$), 集电极漏电流 $<20\mu A$, 接成共基电路, 提高恒流源输出阻抗, N 点电位变化时, L 点电位不受影响。D₁、D₂ 是开关二极管、反向漏电流控制在 $30\mu A$ 以下。

当 A_3 为高电平时, T_1 截止, B 点为 $-3.4V$, Σ 点电位在 $+0.7 \sim -0.7V$ 之间变化, (Σ 是接比较器输入, 又比较器输入接有二只双向限幅二极管), 所以 N 点变化范围是 $0V \sim -1.4V$, D_1 是截止的, i_1 由 D_2 流向相加点, 相反 A_3 为低电平时, T_1 导通饱和, B 点为 $+2V$ 左右, D_1 导通, N 点为 $+1.4V$ 左右, D_2 截止, i_1 经过 D_1 , 这时相加点 Σ 没有电流。

我们可以看出,采用恒流编码后,有这样一个优点:恒流源始终是存在的,用二极管作恒流控制开关,转换速度比较快,而且精密电阻种类也不多。

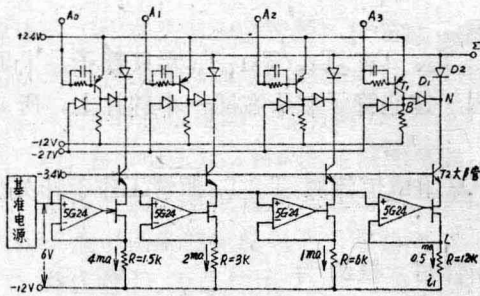


图 (三)

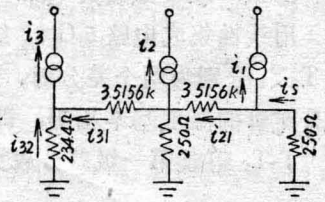


图 (四)

2. 分流器

分流器电路如图(四)所示:

i_2 分配给 i_5 的电流是 $1/16 i_2$ 、 i_3 分配给 i_5 的电流是 $1/256 i_3$ 。设 $\Sigma=0$, $i_3=0$, $i_1=0$ 则图(四)的等效电路如图(五)所示: $R = (234.4 \Omega + 3.5156 K) \parallel 250 \Omega = 0.2344 K \therefore i_{21} = 1/16 i_2$ 又设 $\Sigma=0$, $i_1=0$

$i_2=0$, 同理可求到 $i_{31} = 1/16 i_3$ 、 i_{31} 再分流到 i_5 的电流是 $1/16 i_{31}$, i_3 分流到 i_5 的电流是 $1/256 i_3$, $\therefore i_5 = i_1 + 1/16 i_2 + 1/256 i_3$ 。

3. 比较器

比较器线路图如图(六)所示:

对比较器要求鉴别灵敏度高, 响应速度快。此比较器特点, 当 $u_{\lambda} < 0$ 时, 选通脉冲来时, 比较器发出复另脉冲, 当 $u_{\lambda} \geq 0$ 时, 选通脉冲来时, 不发复另脉冲。它主要包括以下几部份。

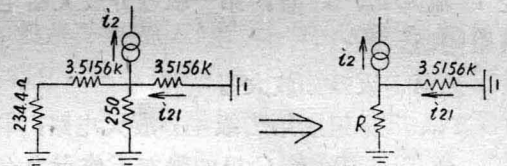


图 (五)

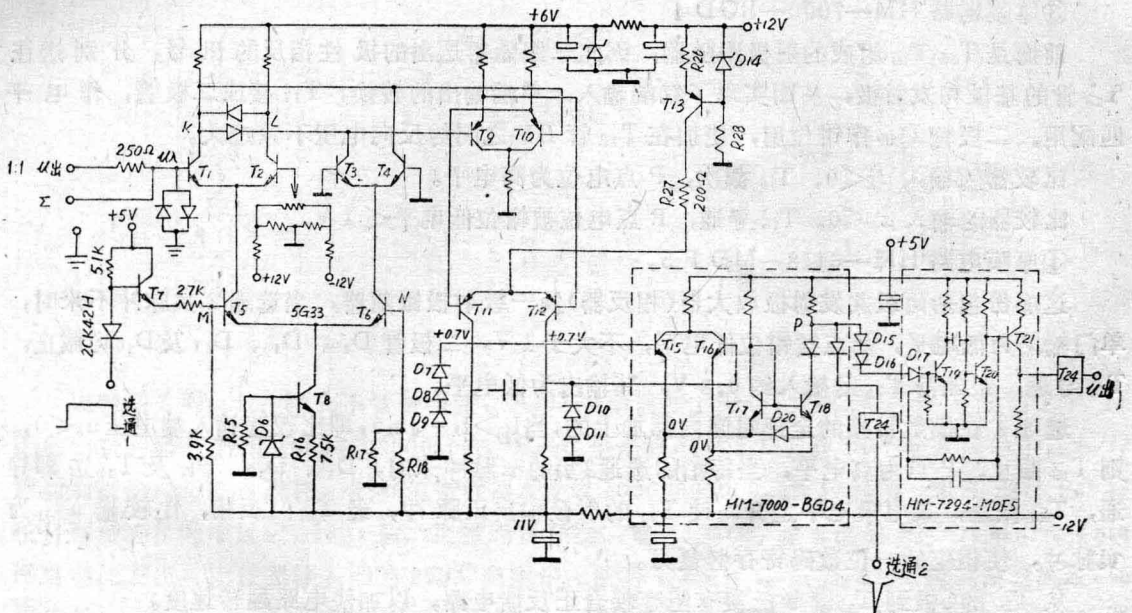


图 (六)

①第一级差分放大电路

用线性集成电路 5G33 组成, 5G33 中包括 T_1 、 T_2 、 T_3 、 T_4 、 T_5 、 T_6 六只管子, T_1T_2 及 T_3T_4 分别为两个差分对, β 与 V_{BE} 是对称的, 由于这些管子制作在同一小硅片上, 所以差分对的温度基本相等, 即温度漂移是一致的。

T_1T_2 组成第一级差分放大器, T_8 、 R_{13} 、 R_{16} 、 D_5 组成恒流源, T_3 、 T_6 两管工作于开关状态。

当选通 1 端为高电位, 即选通正脉冲来时, T_7 管的 e 极约为

$$3.6V. V_M = 3.6 - (11 + 3.6) \frac{2.7K}{3.9K + 2.7K} = -2.3V, \text{当选通 1 端为低电平时, 即}$$

选通正脉冲不来, T_7 管的 e 为 0.3V 左右, M 点电位

$$V_M = 0.3 - (11 + 0.3) \frac{2.7K}{3.9K + 2.7K} = -4.3V.$$

调整 R_{13} 使 N 点电位 -3V 左右 ($R_{18} = 5.6K$, $V_N = -3.2V$), 这样当选通 1 正脉冲不来时, T_6 基极电位比 T_5 基极电位高, T_6 导通, T_5 截止, 恒流源的电流经过 T_6 管流过 T_3 、 T_4 管, T_1 、 T_2 差分放大器电路不工作, 调整 R_{16} , 使 K 及 L 的静态工作点为 +2V 左右。

当选通 1 正脉冲来时, T_3 基极电位比 T_6 基极电位高, T_5 导通, T_6 截止, 恒流源电流经 T_5 流入 T_1 、 T_2 管, 第一级差分放大器电路工作, 其输出信号送到由 T_9 、 T_{10} 管组成的射极跟随器。

②第二级差分放大器电路

T_{11} 、 T_{12} 组成第二级差分放大电路, T_{13} 、 D_{14} 、 R_{26} 、 R_{28} 组成此级恒流源电路, 调节 R_{26} , 使 T_{11} 、 T_{12} 管 C 极的静态工作点为 +0.7V 左右, 这样 T_{15} 、 T_{16} 的射极电位约为 0V 左右, 第二级差分放大器电路的输出送往厚膜电路 HM—7000—BGD 4。 D_7 、 D_8 、 D_9 、 D_{10} 、 D_{11} 组成限幅电路, 当 T_{11} 、 T_{12} 偏离太多时, 防止一只管子 C 极太负。

③厚膜电路 HM—7000—BGD 4

前面是 T_{15} T_{16} 组成的射极跟随器, 因此二跟随器送出的极性相反的讯号, 分别送往 T_{18} 管的基极和发射极, 从而实现了双端输入, 单端输出的转换, T_{17} 接成二极管, 作电平匹配用, 二极管 D_{20} 作钳位用, 使加在 T_{18} 管 BE 之间的反向电压不致过大。

比较器的输入 $\mu_A < 0$ 、 T_{18} 截止, P 点电位为高电平。

比较器的输入 $\mu_A \leq 0$, T_{18} 导通, P 点电位被钳位低电平 $< 1V$ 。

④厚膜电路 HM—6118—MD F 5。

这部份包括两级共发射极放大器(相反器)及一级射极跟随器, 当选通 2 负脉冲不来时, 单门输出为低电平, P 点被钳位低电平, 不大于 1V, 二极管 D_{15} 、 D_{16} 、 D_{17} 及 D_{19} 均截止, T_{20} 导通, 跟随器 T_{21} 的输入约 0.3V, 其输出为低电平。

选通 1 在模数转换的全部时间内都加上的, 当 $I_S > I_A + 4ma$, 即比较器输入端电压 $u_A < 0$, 则 T_{18} 截止, P 点为高电平, 当每拍的选通 2 负的窄脉冲来时, D_{15} 、 D_{16} 、 D_{17} 及 T_{19} 立即导通, T_{20} 截止, 集电极电平上升, 使 T_{21} 的射极输出正脉冲, 经 T_{21} 门反相, 比较器 $u_{出}$ 为负脉冲, 使相应的一位数码寄存器复另。

从 T_{20} 的 C 极到 T_{19} 管 b 极接有阻容耦合正反馈电路, 以加快电路翻转速度。

比较器本身是一个高增益多级放大器, 而整个电路中未加负反馈, 所以在连续工作的情

况下，工作点是不易稳定的，极易产生振荡，要求在选通脉冲的瞬间工作，为了避免出现振荡，所以第二级选通用的时间应尽量缩短，一般为 80ns 的负脉冲来选通。

T₂ 管的基极电路中接有调另电路，调节范围约为 ±36 mv。

4. 参考电压及 4 ma 固定偏流电路

前面讲到的恒流源开关电路中，要有 -6V 基准电压稳定，4 ma 稳定。参考电压及 4 ma 固定偏流电路就能完成以上二方面要求。线路图如图(七)所示：

稳压管 D₁ 及晶体原 T₁，T₂ 是用来稳定流过稳压管 D₂，D₃ 的工作电流，D₁ 稳定 T₁，T₂ 两管基极电压，从而使 R₁ 电阻两端的电压基本稳定，使 T₁ 及 T₂ 两管的发射极电流保持恒定，由于 I_c=I_e，而且 5G 23 组件接成 1:1 负反馈放大器的输入阻抗很高，所以集电极电流就是 D₂，D₃ 的工作电流，也是稳定的，A 点对 +12 V 的电压稳定，B 点对 -12 V 的电压也是稳定的，5G 23 及 T₃，T₄ 管接成电流负反馈电路，显然 C 点的电位就是稳定在 A 点电位， $i_{c3} = VD_2 / 1.5K + WD_1$ ，调整 WD₁ 就可得稳定的 4 ma 电流。T₃，T₄ 接成复合电路， $i_{c3} = i_{c3}$ 、 i_{c3} 供 4 ma 固定偏流用。

下面的 5G 23 及 T₃ 管组成电压负反馈电路，保证 D 点电压稳定，调整 WD₂，就能得 6 V 基准电压。

5. 1:1 放大器

1:1 放大器采用 4E 321 C 线性组件放大器，线路图如图(八)所示：放大器输出经过 250 Ω 电阻，接到相加点，A/D 在转换过程中，相加点电位在变化的，无论相加点电位怎么变化，要求 u_λ 不变，才能保证 A/D 精度，就对 1:1 放大器有高频响要求，使得放大器动态输出阻抗低。

(五) A/D 工作过程

A/D 逻辑图如图(九)所示：

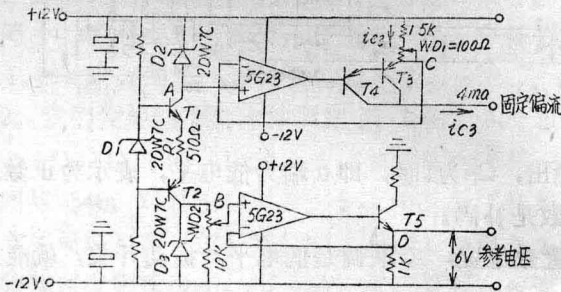


图 (七)

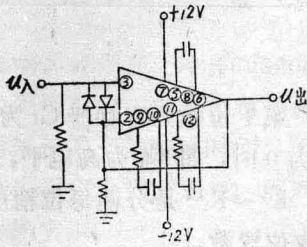


图 (八)

以联调为例：K₁₁ 打到不置数位置，A/D 启动信号上升边把移位寄存器 A 全清“0”，把数码寄存器 C 置成 10000000000，A，B，C 全是高电平、GJ 全“0”也是高电平，A/D 启动后沿形成 100ns 脉冲，把移位寄存器 A 置成 10000000000，比较器第一级选通加上，移位寄存器移位脉冲延迟 100ns 后，把数码寄存器 C₁ 置“1”（C₁ 原来也是“1”），第 1 位 4 ma 网络电流发出，比较器输入端的 250 Ω 电阻中，就有 i_s = 4 ma，i_λ 及 4 ma 固定偏流流过，如果 i_s ≤ i_λ + 4 ma，置“1”数码寄存器 C 的脉冲再延迟 0.6 μs 后，形成 80ns 脉冲加到比较器第二级选通上，这时比较器没有正脉冲输出，第 1 位数码寄存器仍保留为“1”，A/D 启动