微型计算机译文集

（一）

上海工业自动化仪表研究所

前　言

近年来，随着大面积集成电路制造技术的发展，小型中央处理单元可以集成在一个硅片上（叫做微处理机），从而使微型计算机（以下简称微型机）作为一个新的领域崭露头角。它的出现给工业过程控制提供了重要工具，目前在发展的一种总体分散控制系统（也叫微控系统），其核心部件就是微型机。该系统从根本上提高了工业过程控制系统的可靠性。

微型机是由单片中央处理机(即微处理机）加上适当的IC存储器、I/O接口等大面积电路部件构成。微型机的主要特点是部件集成度高、体积小、功耗小和成本低。

目前出现的一些微型机，从制造工艺来看有PMOS、NMOS、CMOS和I^2L电路做成；从机器构成来看有4位、8位、12位和16位字长的微型机。关于国外微型机发展概况请参考《计算机译文集》第四集附录。

根据“洋为中用”的原则和工作的需要，我们编译出版了这本参考资料。本集重点翻译了东芝PLCS——12微型机。PLCS——12微型机是专为工业控制而设计的，它可以在相当恶劣的环境下使用。此外，PLCS——12在功能上已与小型计算机并驾齐驱。PLCS——12A是PLCS——12的改进型，主要是单片的12位并行运算中央处理单元作了改进，所以将改进后的T3190和PLCS——12A的指令系统收入本集。TC5005C是东芝电机公司76年2月份发表的16位字长的CMOS电路微处理机，也收入本集。由于篇幅的限制，微型机的应用仅收入二篇，以后将编集微型机应用专集。

由于水平有限，加之时间仓促，译后未能详细校对，所以文内难免有错误和不当之处，译文仅供参考，并欢迎批评指正。

上海工业自动化仪表所控制机室

1976·10

目　　录

东芝TLCS——12装置动作说明书（摘要）

TOSHIBA LSI Computev System——12

I. TLCS——12构成概要

TLCS——12所用主要MOS／LSI如表1所示。

表1 TLCS——12用MOS／LSI

序号	LSI 电路名称	制作号码	封装出脚 DIPO-6"	电源	功耗 常温mW
1	12位并行处理机（PU）	T3153	42	±5 地	300
2	128×4 bdt静态读/写存贮器（RAM）	T3151	16	±5	330
3	512×4位可写只读存贮器（PROM）	T3181	24	±5 -9	370
4	512×4位屏蔽只读存贮器（ROM）	T3233	24	±5 地	200
5	存贮器控制组件（MCU）	T3216	42	±5 地	300
6	输入／出控制组件（DCU）	T3218	42	±5 地	300
7	12位双向总线驱动（BDU）	T3217	3.6	±5. 地	300
8	通用4／8位寄存器（G10R）	T3220	42	±5 地	250
9	8位中断闩锁组件（INTU）	T3219	36	±5 地	2

TLCS——12的速度大约是小型机的$\frac{1}{3}$～$\frac{1}{4}$。

专用的入／出电路对应每个要求而专门设计，尽管如此，对其中通用性高的电路也考虑了将来的MOS／LSI化。

TLCS——12用12位双向公用总线构成。

TLCS——12的基本构成如图1所示，大型构成如图2所示。

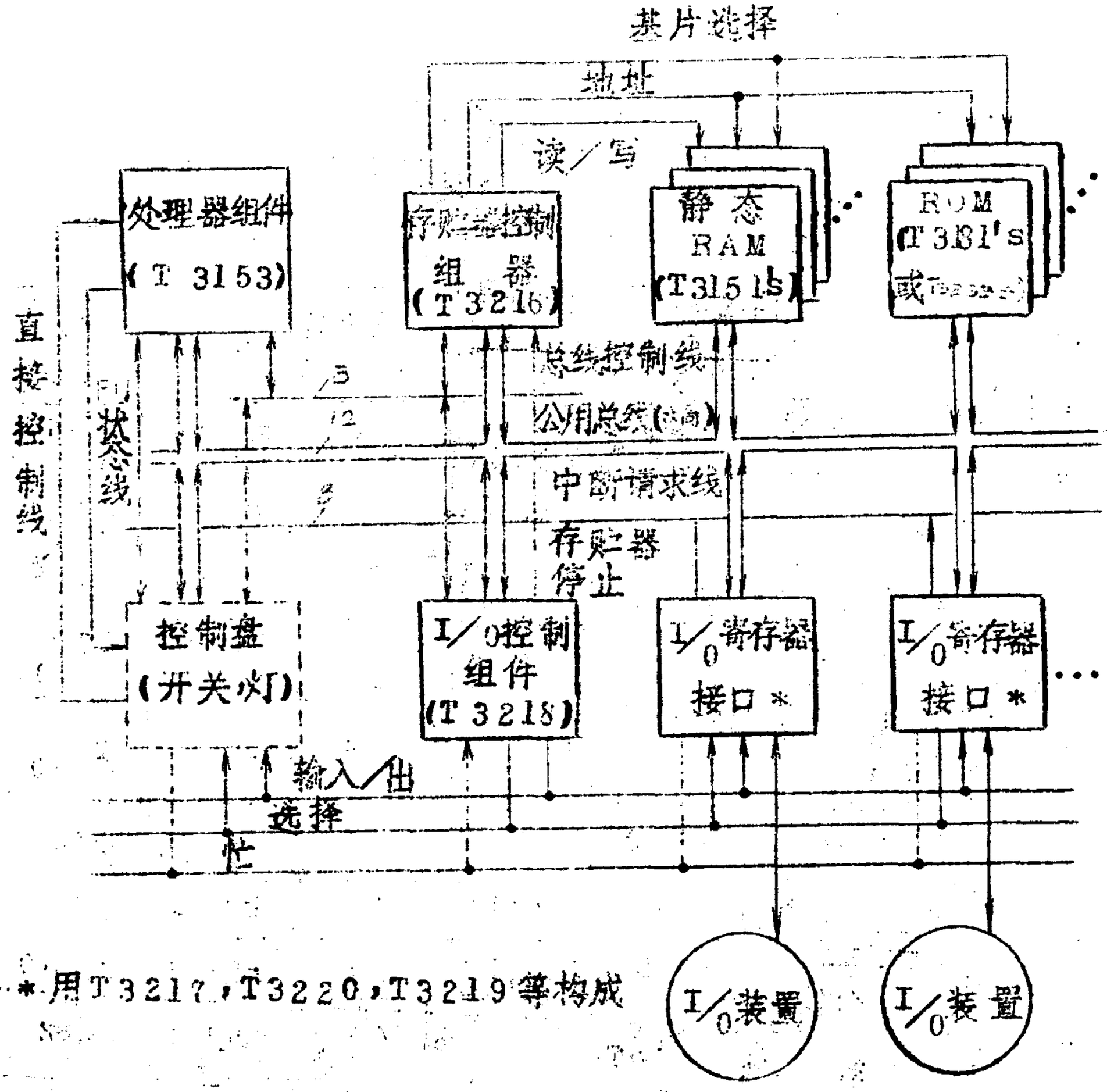


图1　TLCS——12基本构成

（通用寄存器，RAM，ROM和I/O寄存器的总数小于4096字）

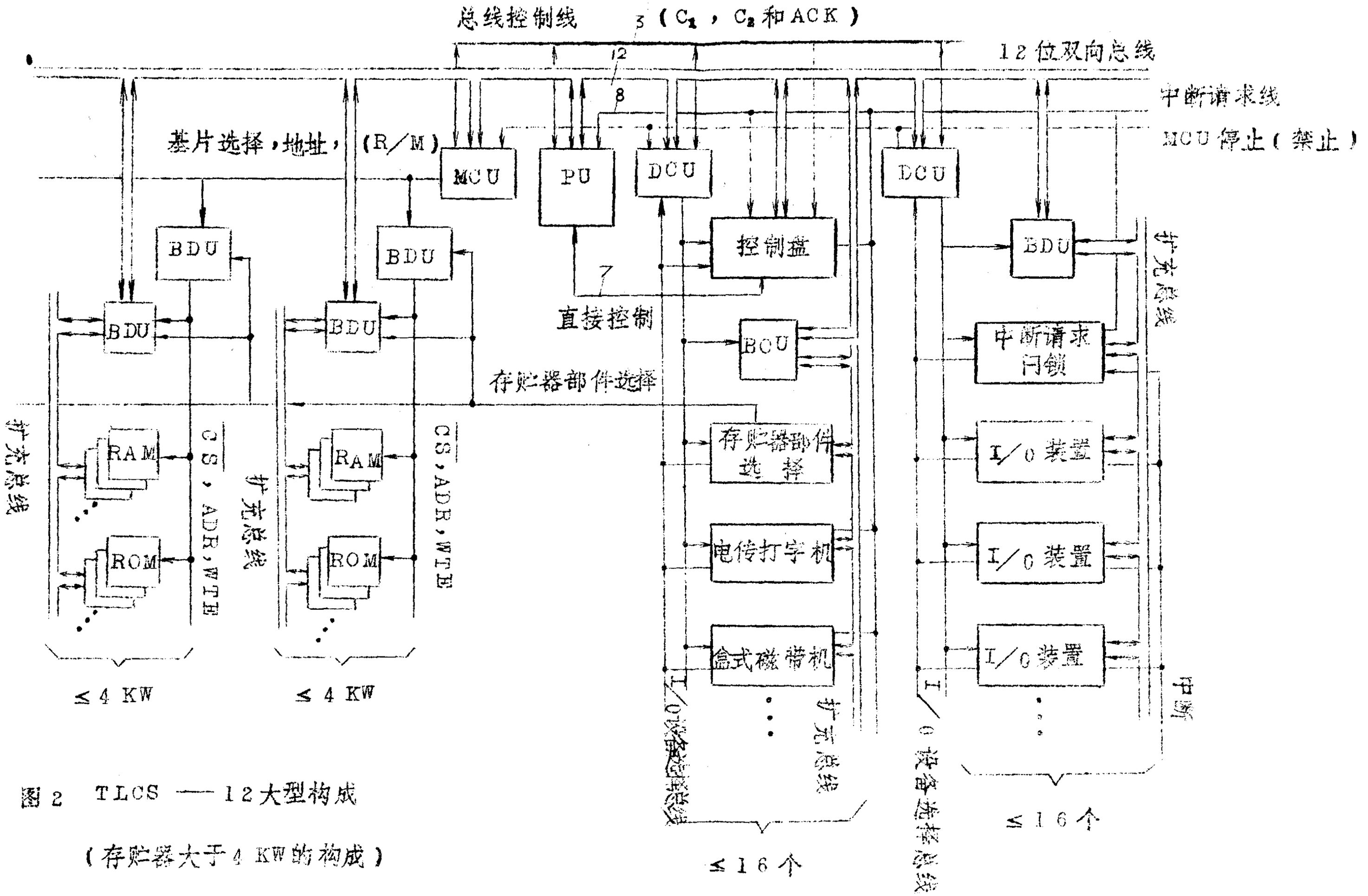


图2 TLCS —— 12大型构成

（存贮器大于4 KW的构成）

小规模的装置使用一个处理机（PU），3个RAM，3个ROM或PROM，1个存贮器控制组件MCU，1个I/O控制（DCU）和电传打字机的接口等构成。这种装置的存贮容量（包括通用寄存器8个和I/O装置数）小于4K字，如果使用存贮器部件选择（1种I/O装置），可以将容量扩充到4K字以上。

该装置由于备有自动启动逻辑，所以可以不要控制盒，但如果要用的话，可以作为1个I/O装置来连接。用控制盒可以直接利用处理器的控制端子而控制整个装置。

该装置的中心是处理器T3153，它是12位并行运算的处理部件，完全集成在1个基片上，可以不需要任何外接电路。

RAM，ROM基片都是以4位并行构成，分别为128字和512字，在常温下，它们的存取时间最大是1μs。PROM和ROM可以互换。

MCU是通过总线来控制PU和RAM/ROM的数据传输组件，它具有地址和数据分离，确保存取时间和对总线控制信号的回答等功能。

DCU是通过总线来控制PU和I/O装置的数据传输组件，主要是进行I/O的地址选择，数据传输时序的调整，具有和MCU大致一样的功能。

在该装置的构成中，由于所有的组件都和总线连接，虽然具有统一的灵活构成，但另方面总线的负载（特别是电容负载）变大了，所以适于MOS电路的阻抗大和数据速度低的场合。为了构成大的装置，要插入双向公用总线驱动电路（BDU），以减轻总线上的负载。

II. 存贮器空间的分配

该装置中PU和存贮器及I/O装置间的数据传送都通过总线用12位并行，对地址分配来说，12位的2进制地址最大可直接指定4096字，在4096个字中还包含有作为特殊使用的寄存器。

参考图3。

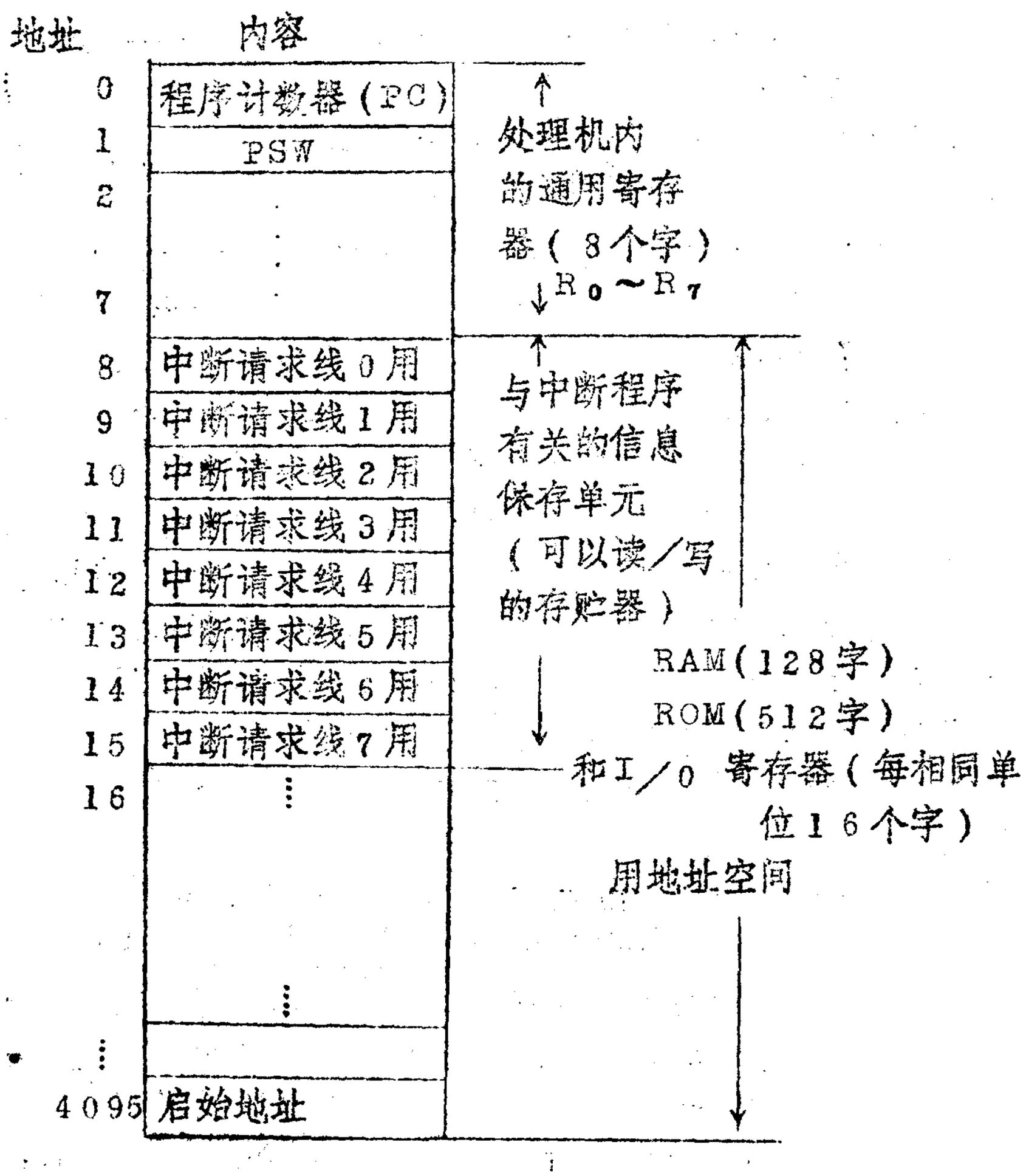


图3　地址空间的分配

开始的8个字分配给PU内的通用寄存器R_0～R_7，R_1为程序计数器PC，R_1为程序状态字PSW。

紧接着的8个字，即8～15，使用于保存与中断程序有关的信息，因此可以方便的进行8级多重中断。不需要的那一级中断的对应

地址可以自由地作为其他目的使用。

地址8～4095可以任意分配给RAM，ROM或I/O装置内的寄存器。在TLCS—12中，RAM占128个字（2^7），ROM占512（2^9）个字。I/O寄存器按照输入／出控制组件（DCU）来选择。对DCU来说分配给16个字。

由于DCU可以直接控制I/O装置内的I/O寄存器，所以1个I/O装置内可以设置任意个的I/O寄存器。

I/O装置内的寄存器也和存贮器一样被分配给一定的地址，因此不需要另外设置I/O指令，所有的指令都可以使用于访问I/O装置。（这是该装置的一大特点）

装置在启动的时候，将4095的内容送给PC，由于ESW被清除，所以必须实装4095地址。（尽管架空比较好，但必须回答）

在启动时要直接执行程序（或自启动）时，地址4095的内容必须是程序的启始地址。

III．公用总线的控制方式

TLCS—12以12位的公用总线为轴，由它来进行各组件间的数据传送。另外还附加有3根控制线，来控制总线上数据的性质和时序。

通过公用总线进行数据的传输是由PU内部的总线控制器（BSC）来控制的。向BSC的总线要求由于只从PU内输出，所以公用总线的数据传输都是在PU的控制下进行的。

2根控制线C_1和C_2为从PU内的BSC输出的控制信号，另外一根ACK（acknowledge），为MCU或DCU给的回答信号。

公用总线的动作由PU内的设备周期按独立的3根控制线（C_1、C_2和ACK）完全异步控制。因此对该总线来说，可以比较方便地连接任意速度或任意类型的存贮器或I/O装置。

通过该总线进行信息交换时的控制线状态迁移如图4所示，概要叙述如下（按序）：

(1) 总线操作从PU的读／写把“地址”配置在公用总线上开始，地址全部送到总线被确认后，PU把C_1和C_2置高电位（逻辑“1”）。

(2) MCU和DCU在$C_1 \cdot C_2 = 1$时接收总线上的地址，然后判断该地址是否是在各自本身的控制下，如果是的则被选中，并保持这个地址（为了以后的说明，该信号叫做SL），把ACK置高电平。

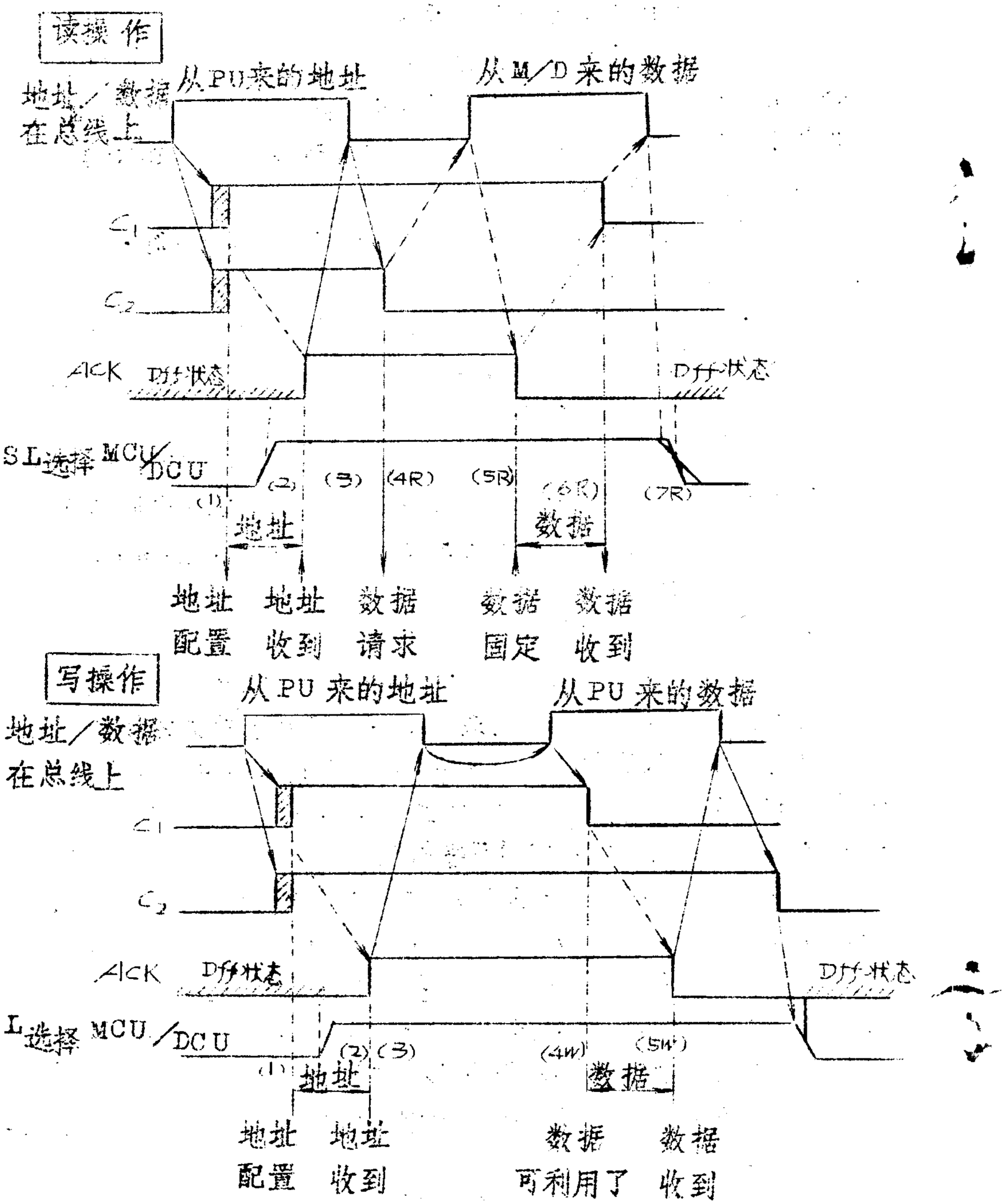
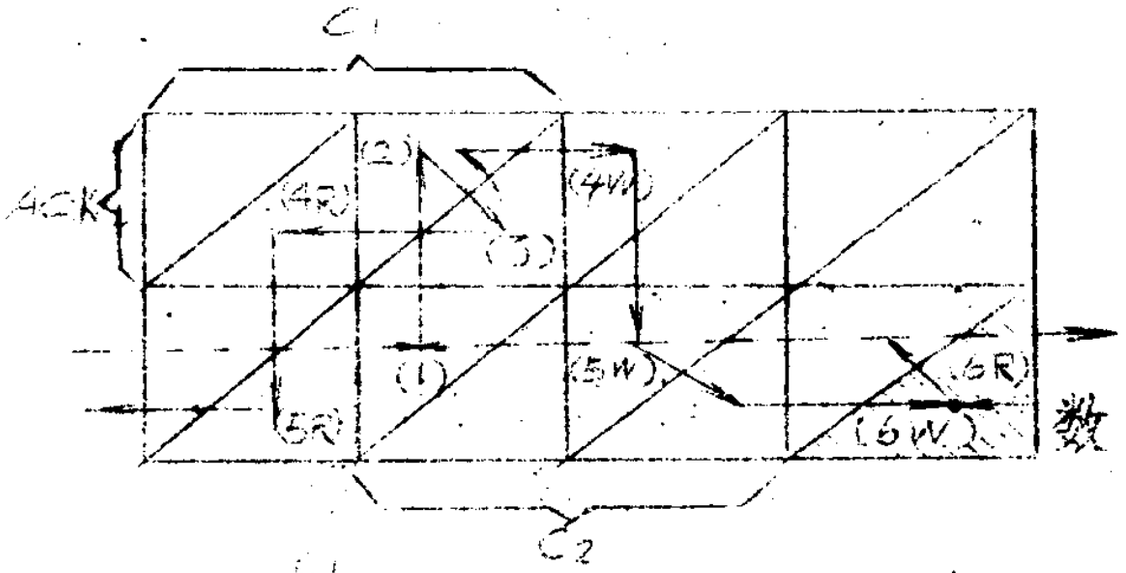

读操作
地址/数据
在总线上
从PU来的地址
从M/D 来的数据
C1
C2
ACK
D触发状态
D触发状态
SL选择 MCU/DCU
(1)
(2)
(3)
(4R)
(5R)
(6R)
(7R)
地址
数据
地址 地址 数据 数据 数据
配置 收到 请求 固定 收到
写操作
从PU 来的地址
从PU 来的数据
地址/数据
在总线上
C1
C2
ACK
D触发状态
D触发状态
SL选择 MCU/DCU
(1)
(2)
(3)
(4W)
(5W)
地址
数据
地址 地址 数据 数据
配置 收到 可利用了 收到

图 4--a) 逻辑波形

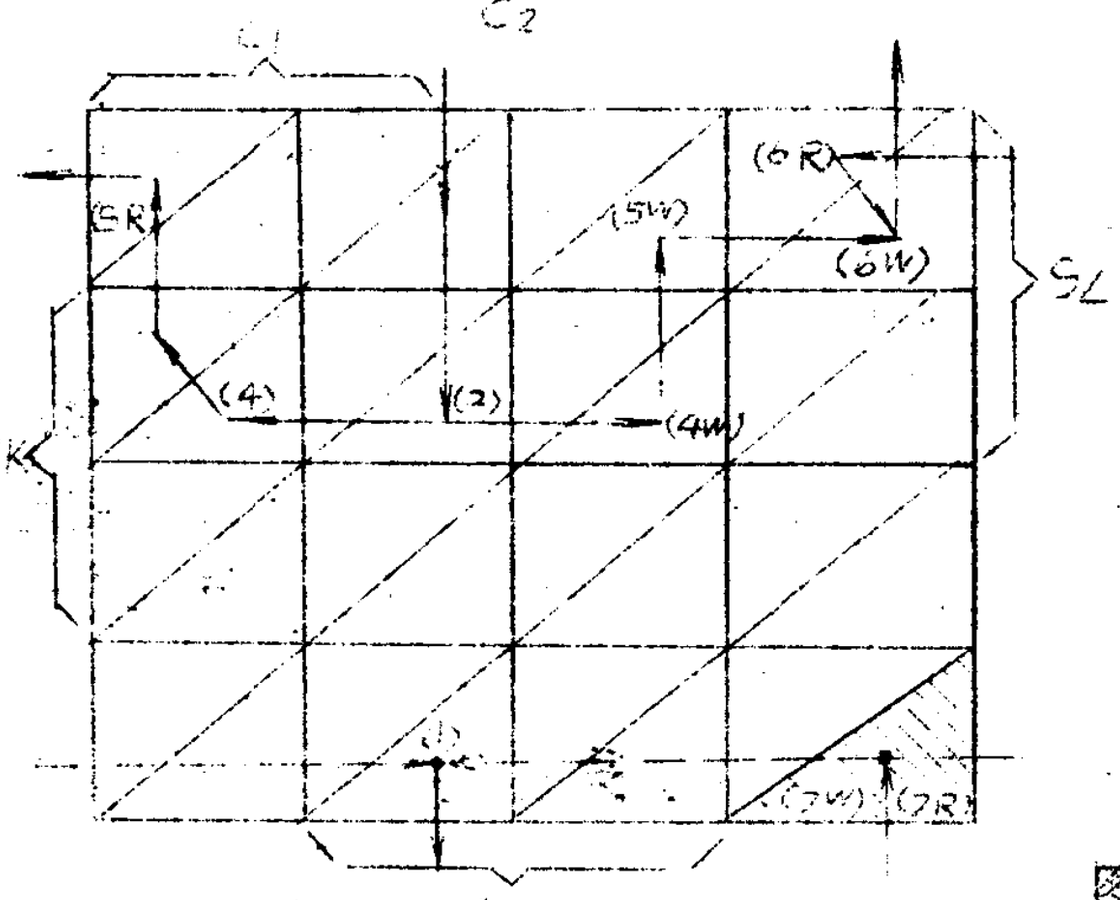
(1)
处理机的
状态图



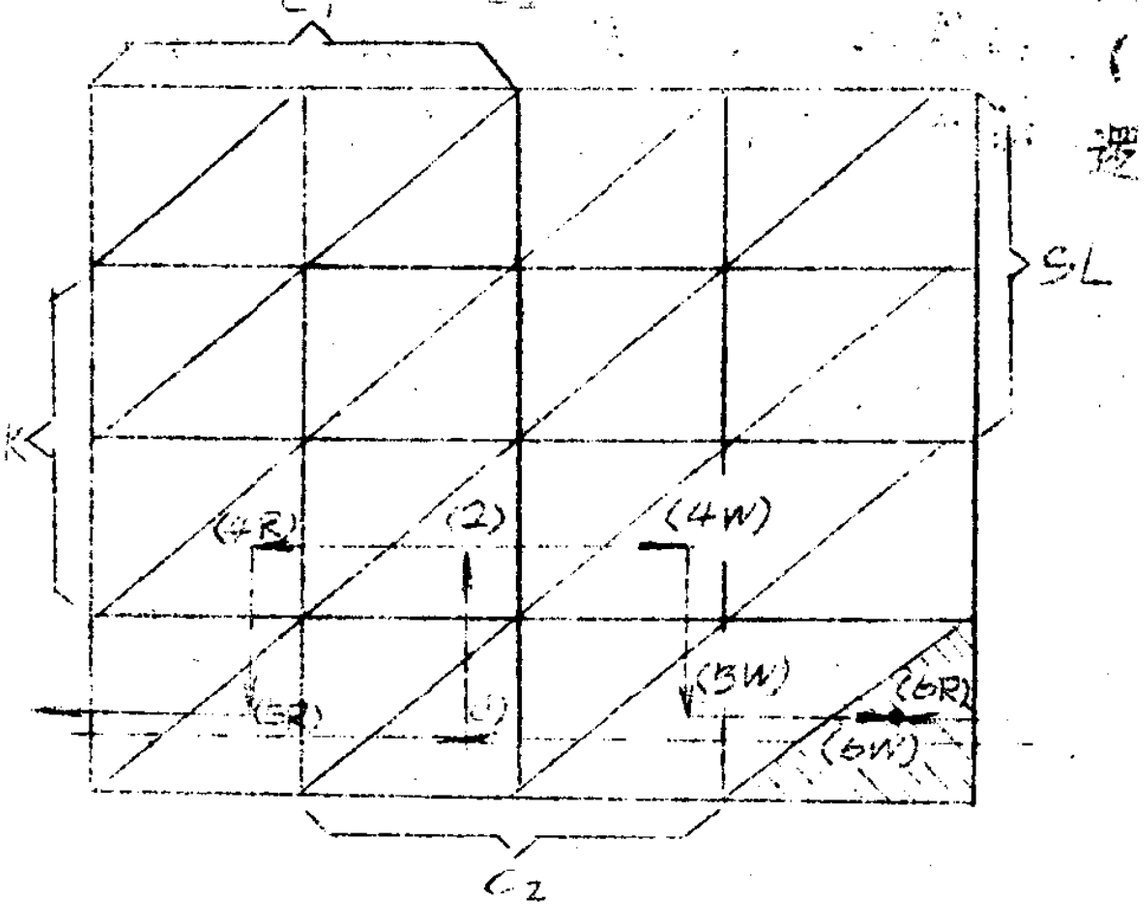
数据收到

BUS 驱动

(2)
被选择的
MCU/DCU
状态图



(3)
未选择的
MCU/DCU
状态图



注：
图中(1)(2)…(6R)(6W)等的说明见逻辑波形图中的说明。

图4 b) 状态迁移图

图4 公用总线控制方式

(3) 如果ACK = 1，PU切除地址的输出。因此，地址的正确值在总线上应该是C_1，C_2，$\overline{ACK = 1}$的时间。

(4R) 下面如果是读操作，PU把C_2置“L”电平，要求读数据。

(5R) 如果SL = 1（即ACK为“H”电平），MCU或DCU则变为$SL \cdot C_1 \cdot \overline{C_2} = 1$，对前面接收的地址进行读操作，把读出数据送到总线上。如果数据完全被送到总线上了，SL = 1，MCU或DCU把ACK置“L”电平，告知PU读出数据已送到总线上。

(6R) 如果ACK = 0，PU接收读数据，把C_1置“L”电平，读操作结束。

(7R) SL = 1，如果$C_1 = 0$（即$C_1 + C_2 = 0$），则MCU/DCU切除读出数据，（把SL复位为“0”），结束有关的动作。因此，读出数据在总线上的正确时间是$SL \cdot C_1 \cdot \overline{C_2} \cdot \overline{ACK} = 1$。

(4W) 另方面，在写操作的时候，继〔(1)～(3)〕的地址传送之后，PU把写数据送到总线上，在确认之后把C_1置“L”电平。

(5W) SL = 1，如果$C_1 = 0$，MCU/DCK开始写操作。如果写操作结束了，把ACK置“L”，并告知PU。

(6W) 如果ACK = 0，PU切除总线上的写数据，把C_2置“L”，写操作结束。

(7W) SL = 1 如果$C_2 = 0$（即$C_1 + C_2 = 0$），MCU/DCU结束有关的操作（把SL清“0”）。因此写数据在总线上的正确时间是$\overline{SL} \cdot C_1 \cdot C_2 \cdot ACK = 1$。

MCU对所有的地址空间（4096字）都有回答的场合，（例如MCU为T3216的时候），如果DCU被选中，则采用了禁止MCU功能的方法。

在目前的装置中，控制线C_1和C_2由PU输出，但ACK线按地址值，由被选择的MCU/DCU输出，ACK的输出电路必须是3状态的，即SL信号应该是3状态的ACK输出电路的使能信号。

还有，正在执行的MCU/DCU在ACK=“H”电平时，其他的MCU/DCU也必须要能正确的接收地址，因此通常地址正确在总线上的时间（$C_1 \cdot C_2 \cdot \overline{ACK = 1}$）为0.5 μs左右。

在所有的MCU/DCU中，ACK输出为开路状态(即高阻抗输出状态)时，由于ACK必须为“L”电平，所以需要下拉(Pulldown)电阻。

ACK在仅下拉电阻的阻抗时，由于容易接收干扰，所以总线动作结束了的时候必须把SL信号清“0”，下次的地址不在自身的控制下的时候，SL也清“0”，以减少ACK线的“开路”状态。

在某装置构成中，如果PU未选中任何一个MCU/DCU，在输出地址时应该用(C_1，C_2，ACK)=(1，1，0)状态待机。在PU内部由于没有时间输出功能，所以为了解决这个问题，在需要的时候要外加电路。使用象MCUT3216那样包括所有地址的空间时是不会引起这个问题。

Ⅳ. 逻辑信号的电气规格。

TLCS——12各信号线在最恶条件下，要满足表2的规格

表2 TLCS——12逻辑信号的电气规格

(最坏条件下的规格)

输入特性

H 电平	3·5 V以上	作为H电平接收电压
L 电平	0·8 V以下	作为L电平接收电压
λ 电流	±1μA以下	输入电压0·4～4·0V范围内
λ 电容	> pf 以下	

输出特性

打开状态*	H电平	> 4·0 V	输出源电流0·4mA时(标准0·8 mA)
	L电平	< 0·4 V	输出吸收电流0·7mA时(标准1·6mA)
关闭状态**(开路)	输出漏电流	<±5μA	输出端电压在0·4～4·0 V内
	输出电容	<10pf	

* 3状态输出时的使能状态和2状态输出时

* * 3状态输出时的阻塞状态

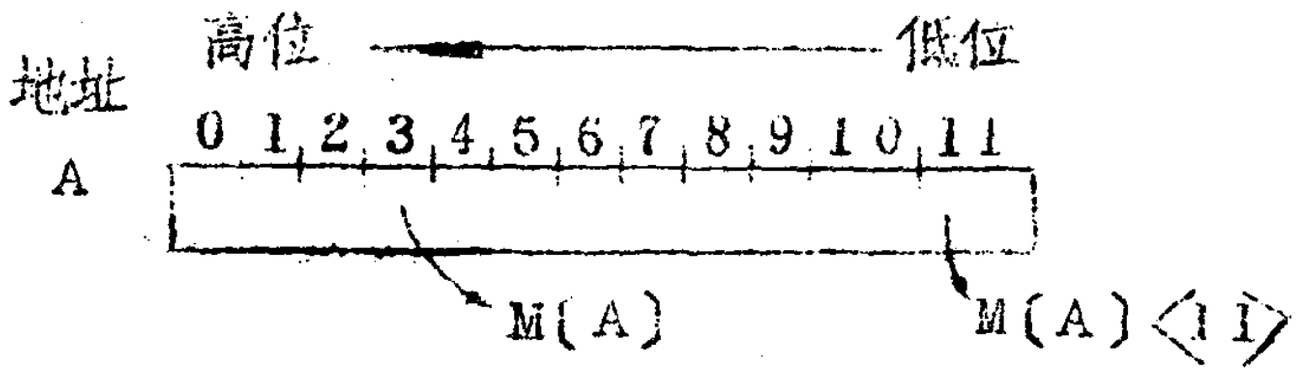
该逻辑信号的入／出特性和5伏电源的CMOS完全可以兼容，而且和附加有下拉电阻的低电力TTL电路也可兼容，除极端使用外，和标准的TTL也可连接。

逻辑信号的上升和下降时间，根据信号线的负载状态而异，但都不影响装置动作，但在标准的状态，为了防止噪音，希望在100 ns左右。

对于异步控制的C_1，C_2和ACK线，要特别注意噪音问题。也就是说，这些线在配线较长的时间，一定要靠近地线，来防止感应的干扰信号。

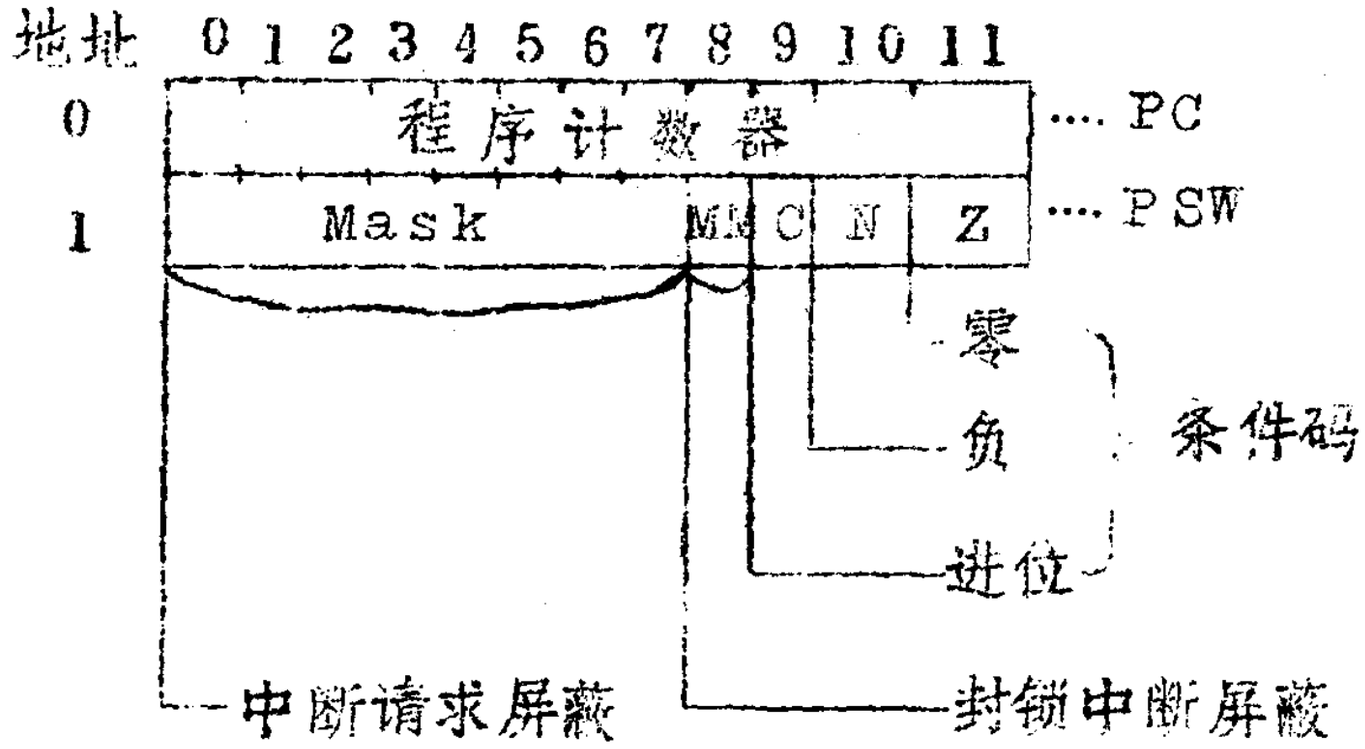
V. 指令和执行时间

为了说明的起见，把地址A的内容用M〔A〕来表示。字的构成如下：



地址“A”的“n”位用M〔A〕〈n〉来表示。但$0 \leq A \leq 4095$，$0 \leq n \leq 11$。

1. 程序计数器（PC）和程序状态字（PSW）



PC 保存正在执行的指令的下一个地址。PSW 由中断屏蔽位和条件代码组成。中断屏蔽位是 8 根中断请求线的屏蔽位 M(1)<0:7> 和封锁所有中断请求的封锁中断屏蔽位 M(1)<8>。关于这个问题后面叙述。

条件代码根据运算处理的结果来置位，使用于条件转移指令的判断。

PSW 的第 9 位 M(1)<9> 叫做“C”，由运算结果的最高位（0位）产生进位输出（为“1”）时置位。还有在循环移位指令时，它作为第13位的寄存器动作。因此C位作为比第1操作数的“0”位更高的位来使用，即第1操作数变为13位的操作。

PCW 的第 10 位 M(1)<10> 叫做N位，它与运算结果的最高位（0位）的内容为同一值而被保存。

PCW 的第 11 位 M(1)<11> 叫做Z位，运算结果的12 位都为“0”时，N位被置“1”，其它场合被清“0”。

2 指令形式和操作数

该装置设置有 18 种指令，大致可分为类型I和类型II。

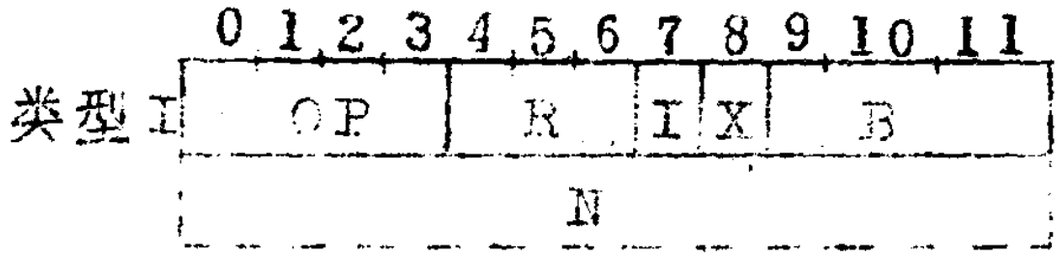
类型I是具有2个操作数的指令，根据修饰有2字指令的情况，这时的第2个字是12 位的地址或直接数据。

类型II是单一操作数和无操作的指令，都是1字指令。

操作代码（OP）为最高4位（0～3），在所有的指令中4～6位指定作为1个操作数（在2操作数指令中的第1操作数）的1个通用寄存器。该操作数的指定通常用R(0≤R≤7)或Rn(0≤n≤7)来表示。

7～11 位，类型I的指令来说，使用于指定第2操作数的地址；对于类型II的指令来说，7～11位作为一种直接操作或操作代码（OP）的扩充使用。

3 类型I的指令



类型I的指令有：LOAD，SWAP，AND，OR，ADD，SUB，MPY和EXEC 8种。第1操作数是M(R)，第2操作数的地址Y根据I、X和B字段来计算。I字段（第7位）指定间接寻址，X字段（第8位）指定变址方式，对变址方式来说为2字指令。B字段（第9～11位）作为寻址的基础指定1个通用寄存器。

有效地址的计算方法：

在各指令的取周期的开始，将有效地址Y清“0”。新的第2操作数的地址Y如下计算：

I X	有效地址	方式
0 0	Y ← Y + B	寄存器方式（RR型）
1 0	Y ← Y + M(B); M(B) ← M(B) + 1	间接方式（R@R型）
0 1	Y ← Y + M(B) + M(PC) PC ← PC + 1	变址方式（RX型）
1 1	Y ← Y + M(M(B) + M(PC)) PC ← PC + 1	间接变址方式（R@X型）

但PC ≡ M(0)，M(PC) ≡ N

注：右边的Y在被执行的指令进行寻址以外的时间为“0”。在这里用B字为0也可以指定PC，这时解释为：

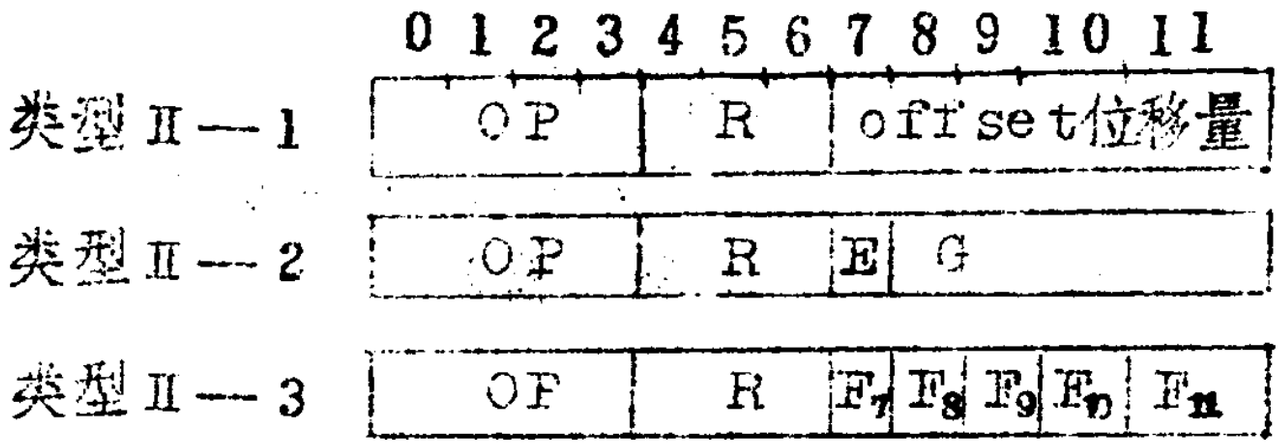
间接方式 →立即方式（2字指令）（RI型）

变址方式 →相对方式（RS型）

间接变址 →相对间接（R@S型）

4. 类型II的指令

类型II的指令分为3组：



类型Ⅱ——1有BCS，BCC，ROT和INCR 4种指令。这种类型的指令，位移量y作为一种直接操作数使用，该直接操作数y把与第7位相同的数值加在左边，作为12 位的数加到y上来使用。*

即：

位移量　Y←Y＋4080×F〈7〉＋F〈8：11〉

*译注：若F〈7〉为1，则左边为111，111，110，000+F〈8：11〉；

若F〈7〉为0，则左边为000，000，000，000+F〈8：11〉。

但，F〈0：11〉是指令，4080＝111 111 110 000，该位移Y用2的补码来表示，也有认为作为代符号的数考虑较方便的。（这时右边的Y仅在被执行的指令时，没到零以外的值）

类型Ⅱ—2有SETB，CLRB，INVB和TSTB 4条指令。这些指令的第7位作为操作代码（OP）的扩充使用。对于这类指令来说，是对被指定的操作寄存器R中的某1位进行逻辑操作。对哪一位进行操作，由指令的8～11（叫做G字段）来决定。

类型Ⅱ—3只有TEST指令，该指令的7～11位作为OP代码的扩充使用，各位分别有独立的意义。

除以上类型外，还有不操作（NOP）指令，它被预定为将来的选择指令（例如DIVIDE或COMPARE指令）所代替。因此，作为用程序来实现不操作，推荐用于特殊的场合。

5. 指令系统

下面叙述TLCS—12的各指令功能。它的概略示于表3。