

DJS—040 系列微型计算机 用 户 手 册

上海无线电十四厂
温州市自动化研究所 合 编
温州市电子技术研究所

浙江省温州市科委情报所

前 言

由四机部联合设计组设计的，以日本SHARP公司的四位机SM—2为蓝本的国产第一个四位微处理机DJS—040（原称DJS—010），已由上海无线电十四厂试制成功，开始批量生产。它是一个面向计算兼有简单控制功能的通用微处理机，具有结构简单、性能稳定，使用方便，价格便宜等优点，有着十分广泛的用途；如各种智能仪表、专用或通用的计算器，低速、简单的数据处理设备，简易控制装置，商用电子仪器（如出纳机、记帐机）等等。

为了帮助广大用户较深入地了解DJS—040机的具体情况，我们共同编写了《DJS—040系列微型计算机用户手册》，本书详细介绍了该机的系统结构及它各部件的工作原理和性能。叙述了它的指令系统及数据结构，并举例说明了DJS—040机的各种程序的编制方法。考虑到某些从事其它行业的读者对计算机比较陌生，本书以适当的篇幅介绍了微型计算机的一般原理及数制、语言等基础知识。

本书第一章、第四至第八章由赵汝斌、曾春霖执笔，第三章、第九至十二章由缪晓胜执笔，第二章、第十三章由郭金贵执笔，并得到有关同志的支持和帮助。

由于我们水平有限，经验不足，书中谬误一定不少，恳切希望广大读者批评指教，共同为发展我国微型计算机事业而努力。

上海无线电十四厂

温州市自动化研究所

温州市电子技术研究所

一九八一年元月

目 录

第一章 绪 论.....	1
第二章 DJS—040 微型计算机简介.....	7
第三章 DJS—040 的指令系统.....	15
第四章 中央处理器 CPU ₁	31
第五章 中央处理器 CPU ₂	47
第六章 数据存储器 RAM	61
第七章 程序存储器 ROM	67
第八章 时钟发生器及显示部件	75
第九章 程序与软件的概念	79
第十章 数制与数据格式	81
第十一章 程序设计基本方法	87
第十二章 DJS—040 机程序实例	91
第十三章 DJS—040 A 微型算计机	102
附录 1 文字符号说明	105

第一章 绪 论

为了对以下各章所讲内容的理解获得必要的概念，本章叙述了计算机的一些基础知识和定义。对计算机已经很熟悉的读者，可以按自己意愿跳过这一部分阅读。

所谓微型计算机 (Microcomputer 或简写 μC 为) 就是由大规模集成电路构成的微型化的计算机。它同传统的电子计算机一样，主要是由中央处理部件 (运算器、控制器)，存储器、输入和输出接口装置和其他辅助电路 (时钟等) 组成的数据处理系统。所不同的是，微型计算机的中央处理部件是具有运算、控制功能的一片或几片大规模集成电路构成的，即所谓微处理器 (Microprocessor)。所以微型机技术本质就在于微处理器，计算机的重大发展也正是从这个微处理器开始的。

§ 1—1 计算机的基本结构与动作

计算机之所以能对一连串的数值和信息自动地进行运算处理，是按照事先编制好的程序，在计算机内部产生控制信号，再根据这些信号控制计算机的各部件依次进行工作。

在近代的计算机系统中，这种程序也采取与数据相同的形式，存放在计算机内部，按照构成程序的指令依次取出，并按指令指定的操作产生控制信号来控制各部件的动作以进行信息处理。采取所谓存贮程序方式。即，把程序作为信息输入到计算机内，存贮起来，使其交换或变更极为方便。这样，计算机就可以具有非常高的灵活性。

从使用计算机的观点看来，程序是极为重要的部分，它与计算机本体即所谓硬件相对应而称之为软件，而硬件和软件形成一体之后才使计算机成为一个系统，从而发挥它的能力。

一、计算机的基本结构

图 1—1 是一台存贮程序式计算机的基本结构图。它有五个基本组成部分。图中心所示的是中央处理器 (CPU)，CPU 又分两部分：控制器 (CU) 和运算器 (ALU)。

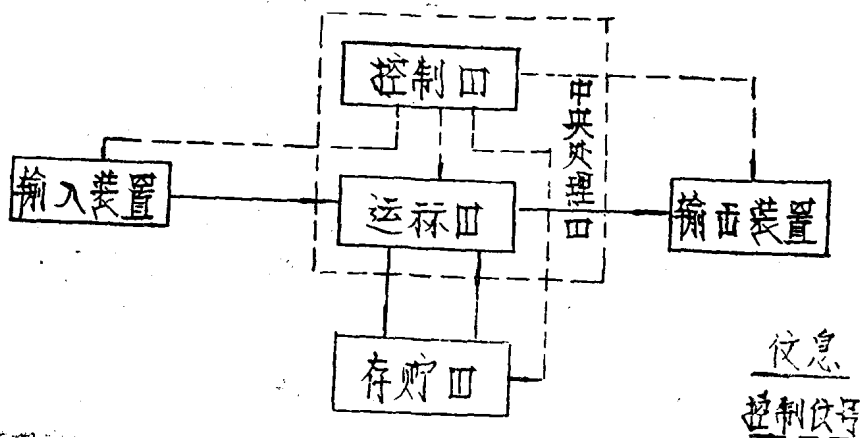


图 1—1 计算机的结构

(1) 运算器 (ALU)

运算器是一台计算机的数据加工核心,是按指令对数据进行算术运算和逻辑运算的处理装置。它是由进行运算操作的加法和寄存与运算有关的数据(即应加以处理的数据和已处理的中间或最后结果)的寄存器,以及计数器等为中心构成的。

算术运算主要为加、减法,而逻辑运算主要是“与”、“或”及移位。计算过程在控制器指挥下,从存储器中读出的数据不断地进入运算器,经处理后,又把运算结果写入存储器保存起来以备以后再用或通过输出装置送到机外。

(2) 控制器 (CU)

控制器具有安排整个系统操作次序的功能,使整个机器能自动地按人们预先编排好的程序进行工作。在无中断和转移的情况下,它从存储器中依次取出组成程序的指令,经过译码,按指令的要求,在适当时间,向不同部件,发出控制信号,使指令得以执行,此外还要接受各部件发回的反馈信息以便决定下一步操作的去向和对意外事故的处理,保证机器工作顺利进行。控制器是由保存下次应该取出指令地址的指令计数器(亦称程序计数器)、存放取出指令的指令寄存器,识别存放在指令寄存器中的指令操作部分的译码器、根据译码结果而发出控制信号的控制线路以及数据地址寄存器和其它附带的线路构成的。

通常一个微处理器系统应具备三条标准总线、数据总线、地址总线和控制总线。控制器将控制着地址总线和数据总线上的数据流,并且管理和解译控制总线上出现的信号。

(3) 存储器

任何计算机系统的存储器都用于存储信息,它存储计算机要处理的程序和数据,并将运算的中间结果以及处理后的结果贮存起来。

在存储器中,把要处理的信息按单位(字,字节或字位等)分别指定一个地址,在计算机内,访问存储器和存取信息是按地址进行的。

一个存储器能贮存的全部信息量称为存储容量,处理的信息的单位是由字或字节数来表示的。例如能容纳4096个字的存储器,共存储容量为4096字,对每个字从0~4095分别指定一个地址。

存储器的主要性能指标是以取数时间或周期时间表示的。取数时间是从给出读出信号到获得信息的时间。周期时间是从给出读出或写入信号及其动作終了到下一个信号开始出现的时间。

从使用计算机的观点来看,存储器的容量越大越好,但由于高速存储器价格昂贵,从经济方面考虑自然会受到限制,所以大多采用低速廉价的大容量存储器作为附加的辅助存储器。这个辅助存储器的信息,可以向高速存储器(相对于辅助存储器又称为主存储器)传送。这种传送是以几个字乃至数千字的字块为单位进行的。

存储器可分成两种基本类型,即随机存储器(RAM)和只读存储器(ROM)。

RAM是一种能写入和读出数据的存储器。ROM是一种只能读出预先安排在它之中数据的存储器。它的好处是具有永久性。而一般RAM却是易失性的,比如一旦电源切断,所存

的数据就随之消失。

(4) 输入——输出装置

输入输出装置是计算机与外界进行联系的桥梁和通道，是人与计算机之间交换信息所必须的装置，需要处理的数据及程序等信息通过输入装置送至计算机。计算的结果通过输出装置，如打印机，显示器等告诉人们。

二、计算机的动作：

存储程序式计算机，是将贮存在存储器中程序的指令一条一条地取出来并执行这些指令。即如图 1—2 所示，通过取出指令和执行指令两个阶段的重复，一步一步地执行程序。

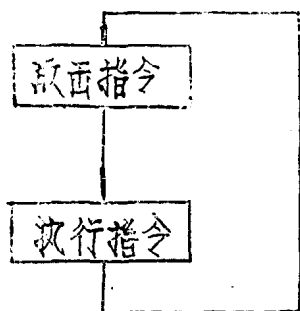


图 1—2 程序的进行

首先，将指令计数器的内容送入存储器的地址寄存器，然后将该地址的内容读出后送入指令寄存器（即下次应执行的指令）；其次，利用译码器对送入指令寄存器的指令进行译码，接着，进行各条指令的操作。每条指令的操作，如寄存器间的信息传送，加法运算，移位，计数内容的增减等都是依靠在计算机内进行基本操作的组合来实现的，控制实现这些操作的信号依次在控制线路中产生并送往计算机的各部分。由这一连串的控制信号形成的基本操作组合，使指令操作完成后就转移到取出下条指令。

程序的控制，若是按存储在存储器中指令的地址顺序进行基本的顺序控制时，仅需将指令计数器的内容加“1”即可。但仅用这种顺序控制则无法根据计算机的内部条件对程序的执行处理进行变更。为此，就要有所谓转移或跳越指令，这类指令出现在程序的指令序列（A）中时，就可依其条件调整计算机的内部状态，如条件满足，可转向程序的另一指令序列（B），条件未满足时，则按原指令序列（A）进行控制，

（如图 1—3）。

依照转移指令，计算机随着程序执行中出现的状态，可以改变下一次应当进行的处理，形成依靠程序本身进行条件判断来改变程序的流程，这就给计算机以判断的能力。用转移指令，就可以在条件

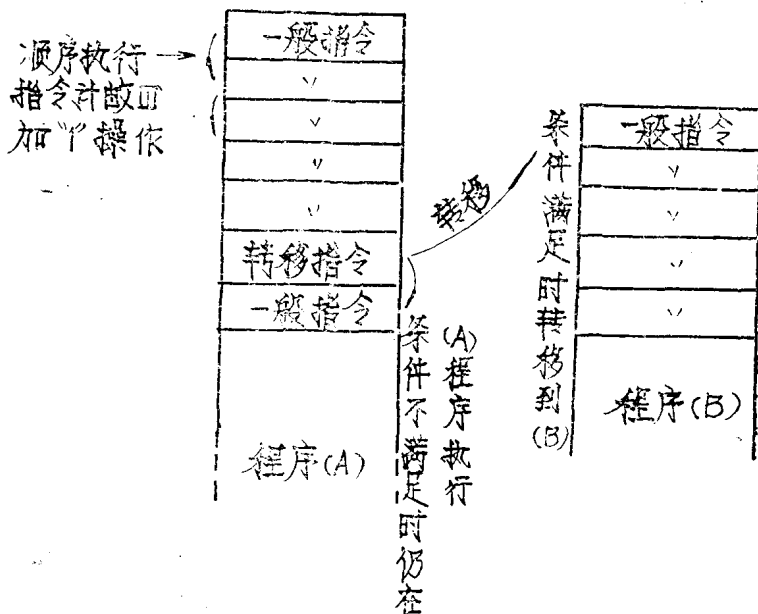


图 1—3 转移指令

满足时, 根据送入指令计数器的转移指令的实际地址, 改变指令的存贮顺序从而改变程序执行的顺序。条件不满足时, 仅将指令计数器的内容加“1”, 程序仍按原指令贮存的顺序执行。

当“调用”一个子程序时, 就要发生一种特殊的程序转移。在这种转移中, 处理器需要“记住”发生转移时主程序的指令计数器的内容。这就使处理机能够在子程序完成最后一条指令后恢复执行主程序。

所谓子程序就是程序中的程序。它通常是在主程序执行中必须反复执行的一组指令。它在很多计算中能共同使用。最好的例子如计算乘方、计算正弦函数或一个顺序变量的对数等, 通常都是编成子程序的。其他例子还有, 如清除存贮器程序、存贮器交换数据的程序等。

子程序的配置可以有二种不同方法, 我们以所谓闭合子程序进行说明。其过程如图 1—4 所示。使用闭合子程序的场合, 从主程序移向子程序和以子程序返回主程序时, 要按照必要的顺序进行。

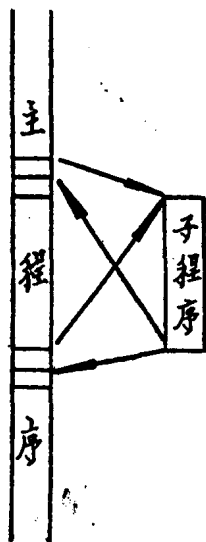


图 1—4 闭合子程序的配置

转子前, 要明确子程序地址同时要把转子前的主程序地址保存起来, 从子程序返回时, 取出转子时保存的主程序地址, 回到主程序继续执行。

处理器有一种专门处理子程序的方法, 以保证有秩序地返回主程序。当处理器收到某一条“调用”指令时, 它就将程序计数器加“1”, 并将其内容存放在一个保留的存贮区, 叫做“堆栈”。因此, 堆栈就保留着子程序完成以后要执行的那一条指令的地址。有些处理器将返回地址存放在处理器内部的一个专用寄存器——栈寄存器里。然后处理器就将“调用”指令所规定的地址输入程序计数器。于是取出的下一条指令将是被调用子程序的第一条。

任何子程序的最后一条指令都是一个“返回”操作, 当处理器取出一条“返回”指令时, 就把存贮在栈寄存器的返回地址送主程序计数器。如系堆栈, 则将堆栈顶端的地址取代程序计数器的现行内容。就能使处理器在原来调用子程序执行完毕后立即恢复执行主程序。

子程序常常是“嵌套”的, 也就是, 有时一个子程序能调用第二个子程序。第二个子程序还可以调用第三个子程序, 依次类推。只要处理器具有足够以存贮必要的返回地址的容量和备有这样做的逻辑条件, 这是完全可以解决的。换句话说, 嵌套的深度取决于堆栈本身的深度。假若堆栈有存放三个返回地址的空间, 那末, 子程序嵌套可以达到三级, 就能实现三级转子。

根据计算机的内部或外部条件, 有时有必要从正在执行的程序转向另一个程序的控制, 这时, 依靠上述的转移指令进行程序转移处理, 比较烦琐, 而且使程序的处理效率显著降低, 因此, 很多计算机设有“中断”的功能。“中断”是在产生某种条件及原因时, 中止正在执行的程序(A), 转入另一个与其无关的强行指定地址以控制执行特意安排的中断处理程序(B), 在完成中断处理后, 再返回原来执行的程序(如图 1—5)。这在原则上和调

用子程序十分相似，不同之处只在于中断的启动来自外部，而不是来自程序。

由于有了“中断”这种功能，提高了程序的处理效率，使多道程序设计，输入输出同时动作，联机处理等成为可能，从而提高计算机系统的利用效率，扩大了计算机的应用领域。

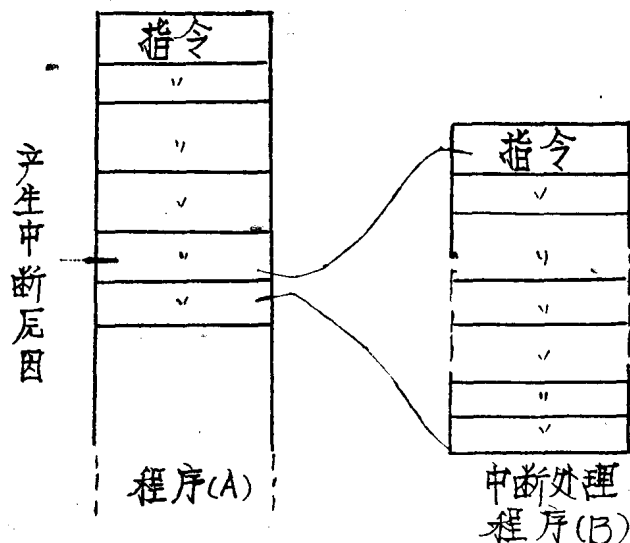


图 1—5 中断处理

§ 1—2 4 位微处理器和 4 位微型计算机

1971年，美国Intel公司在台式计算器的基础上设计了一个微处理器 Intel4004，它是世界上最早的一个4位微处理器，并以它为核心构成了MCS—4微型机，主要用于十进制串行运算和简单的数据处理，基本上是一种可编程序的高级台式计算器。1975年，美国德克萨斯仪器公司发表了TMS—1000 4位单片微型计算机系列。由于，TMS—1000系列的开发和在很多领域的推广应用，促进了4位机的新发展。所以，从1975年以来，新的4位机系列不断地开发出来，到1979年10月为止，已有一百多种，这些4位机，基本上都是单片微型计算机，即一块LSI中包含CPU，RAM，ROM，时钟发生器CG，I/O通道接口等部件。

4位微型计算机的销售量是极大的，根据日本统计，78年和79年两年，4位微型机的销售额占微型机销售总额的50%以上。78年的销售额比77年增长80%。

4位微型机之所以能快速发展和大量销售，是由它的独特性能所决定的。单片4位微型机具有下列特点：

(1)单片4位微型机不同于通用性较强的8位微型计算机，具有一定的专用性。针对具体用途，目的程序预先写入固定式ROM中。这样做，虽然影响了灵活性，但是，方便了用户，降低了成本，提高了可靠性。特别是它价格低廉，是8位机无法比拟的，至少在目前是这样。因此，可以认为，只要8位微型机在经济上无法和4位微型机竞争时，4位微型机的生产和发展是必然的。

(2) 单片 4 位微型机的 ROM 容量, 一般是 0.5K~2K 字节, RAM 容量为 128~512 位。ROM 容量(字节)和 RAM 容量(位)的比, 在一般情况下是 4:1。如日本 SHARP 公司的单片 4 位微型机 SM-2 的 ROM 容量是 1K 字节, RAM 容量 256 位, SM-3 的 ROM 容量是 2K 字节, RAM 容量是 512 位, 它们的容量比都是 4:1。

由于单片 4 位微型机的外引线有限, 一般不考虑 ROM 和 RAM 的扩充能力。这也是和 8 位微型机不同的地方。

ROM 和 RAM 的读写控制, 地址控制分开, 在器件已定的情况下, 这种系统结构, 可能提高工作速度。但是, 8 位微型机中, ROM 和 RAM 的地址是连续编号的, 读写控制也是统一的(如 Intel 8080)。

(3) 单片 4 位微型机通道接口的端数受管脚的接点数的影响。所以输入, 输出的端数就较少, 而且扩展能力也较差。

(4) 单片 4 位微型机总是成系列地发展。同一系列中, 以指令数的多少, ROM 和 RAM 容量的不同, 通道接口能力强弱, 分成几个机种。用户可以根据不同的用途, 在系列中选择合适的机种来使用。这样做, 对生产厂和用户都有好处。

同一系列的不同机种, 系统结构和单元电路基本相似, 生产厂在设计周期上, 可以缩短; 在生产上, 有继承性, 这有利于合格率的提高和成本的降低。同时, 一个系列中的机种互相补充, 适应性很强, 而每一机种的系统结构又不是太复杂。因为, 它的针对性较强。这样, 每一机种的集成度就可以适当地降低。这对于合格率和成本同样带来有利因素。

由于成本降低, 价格便宜, 就便于普及推广应用, 用户可以花不多的钱, 得到一个合适的 4 位微型机, 因而就显示了 4 位机的生命力。

多数的 4 位微型机, 基本上都是单片的。最简单的 4 位机, 单片电路的集成度也在一万个管子左右。当前, 我们要立即批量生产单片 4 位微型机, 是有一定困难的。因此, 我们在设计、试制 4 位微型机时, 采用了多片电路(如 4 片左右)构成的方案。这样, 可以缩短试制周期, 尽快转入批量生产。

由于 RAM 和 ROM 分别单独做成一块电路, 因而产生了独立使用的可能性。对于各用户不同要求, 编制专用程序。通过计算机辅助设计, 图象发生器制版, 能方便制成相应的 ROM, 组成各种专用微型计算机, 满足不同用户的需要。

第二章 DJS—040微型计算机简介

§ 2—1 概 况

DJS—040是参照日本SHARP公司四位单片MC族中SM—2的逻辑结构和指令系统而研制的产品，是部颁4位微型机系列之一，尽管8位，16位单片微处理器不断问世，但由于4位机具有价格便宜，设计灵活，并具有丰富的输入输出，集成度高，可靠性好等优点，在一些简单应用场合无疑是竞争的优胜者。根据78年日本的统计数字4位机在整个微型机的销售量中占据了78%的比例，而且4位机族是一个发展系列，在竞争中又不断有新的更完善的成品加入。

表2—1是日本SHARP公司SM族的简介

表2—1 SM 族 功 能 表

品 名 目 项	SM—2	SM—2/1	SM—3	SM—3/1	SM—4	SM—5
管 脚	42	28	60	42	60	60
工 艺	P—E/D	P—E/D	P—E/D	P—E/D	CMOS	CMOS
电源电压	—15V	—15V	—15V	—15V	—3V	—3V
指 令 数	45	45	57	57	54	57
ROM容量	1008×8bit	1008×8bit	2268×8	2268×8	2268×8	1972×8
RAM容量	64×4	64×4	128×4	128×4	96×4	65×4
子 程 序	1	1	2	2	1	1
输 入 码	8	8	8	8	6	6
输 出 码	27	16	40	23	4	36
双 向 端	—	—	4	4	4	—
指令周期	10μS	10μS	10μS	10μS	61μS	61μS
		SM—2的 28脚化	可扩CMOS SRAM直 接荧光管驱 动	SM—3的 42脚化	液晶直接驱动 板低功耗分频 器内藏石英晶 体振荡	液晶直接驱动板低功耗 分频器内藏石英振荡电 路。

4位机在民用领域占据了绝对优势，象电子炉灶，自动洗衣机，自动空调设备，电子缝纫机，电子玩具等等，在家用电气制品中已被受欢迎。

同时在一些简单工业控制，数字通讯，汽车仪表，财会出纳，科学计算，打印机控制，数字式仪器、仪表，数字血压计，数字体温计等医疗设备中都有广泛的应用，而且科技人员正在不断开辟新的应用领域，可以预料4位微型机具有强大的生命力和广阔的发展前景。

§ 2—2 硬件特点

DJS—040是一个具有4位并行处理功能,包括RAM, ROM, I/O通道及显示的微型计算机。根据我们现在的生产能力,分成五片电路分别取名: CPU₁, CPU₂, RAM, ROM及时钟和显示。管脚排列和详细说明及技术指标请看上无14厂“双岭”手册或书本中各分图。

DJS—040输入端共8个,5个是适用于键盘输入的同步信号,3个是适用于打字机与外部设备输入的异步信号。输出端共是27根,由程序可控的15根(W₁~W₁₅)。累加器内容直接输出4根,另外备有扫描键盘使用的输出端8根。

本机内部采用负逻辑概念,接近-12V(⌋)电平表示“1”,接近0V电平表示“0”(⌋),但是,在机器的输入、输出部分,即与外界信息有关的管脚处采用正逻辑概念。这样在作逻辑分析时能与TTL及CMOS和NMOS兼容。这就是说,在电路片管脚上的信号(用□表示)采用正逻辑。 $\overline{CL}=1$ 表示CL信号在管脚上且为高电平(⌋),在电路片内部采用负逻辑,CL=1为逻辑“1”信号(⌋)。因此 $\overline{CL}$ 管脚上正逻辑—▷○—CL片内负逻辑,即管脚信号倒相后才变成相应的片内信号。

基本指令共45条其中有很多指令是复合指令,即一条指令同时完成几个操作,因而能组成高效的应用程序,提高了运算速度降低了ROM容量。

一、DJS—040的主要特点如下:

- (1) 4位并行处理;
- (2) 8位指令字长;
- (3) 45种含有复合指令的指令系统;
- (4) 丰富的输入、输出;

输入8根	{	5根同步输入;
		3根异步输入;
输出27根	{	7根选通,扫描输出;
		20根通用输出;

- (5) 指令执行周期10μS
- (6) 程序存储器ROM容量1008×8(63×8×16)可扩充一倍;
- (7) 数据存储器RAM容量256(16×4×4)可扩充一倍;
- (8) 子程序一级;
- (9) 内含时序及振荡器;
- (10) P沟硅栅E/D—MOS工艺;
- (11) 单一电源V_{DD}: -12V。

二、主要电参数表

参 数 名 称	符 号	单 位	技 术 规 范			测 试 条 件
			最小值	典型值	最大值	
电 源 电 压	V_{DD}	V	-10	-12		
单 片 平 均 功 耗	P_D	mW	100	150	200	$V_{DD} = -12^V$
输 出 高 电 平	V_{OH}	V	-0.8	-1	-1.5	$R_L = 20M\Omega$
输 出 低 电 平	V_{OL}	V	-8	-8.5	-10	$R_L = 20M\Omega$
输 入 高 电 平	V_{IH}	V		-2		
输 入 低 电 平	V_{IL}	V		-7.5		
输 入 电 流	I_{IN}	μA		≤ 1		-12^V
输 入 电 容	C_{IN}	P F		≤ 8		
上 升 时 间	t_r	μS		≤ 1		
下 降 时 间	t_f	μS		≤ 1		

§ 2—3 系统结

构及其框图

DJS-040的系统，由CPU、ROM、RAM，时钟及显示部件，四大部分组成，它们之间的操作控制是通过四十五条指令编制的指令译码器完成的，由于采用了PLA那样的两级与、或译码，因而结构紧凑、明确简单图2—1是DJS-040系统框图，图中虚线表示的是指令控制与方块图的关系。图2—2是分片框图。

(一)程序存储器与程序控制：

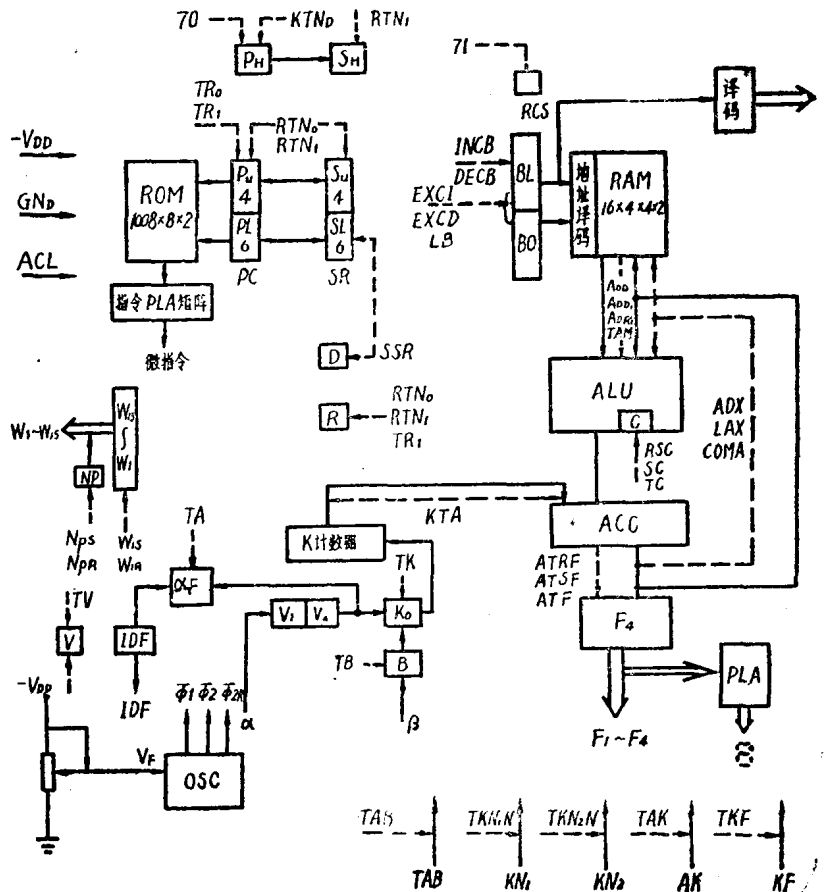


图 2—1 DTS-040 系统框图

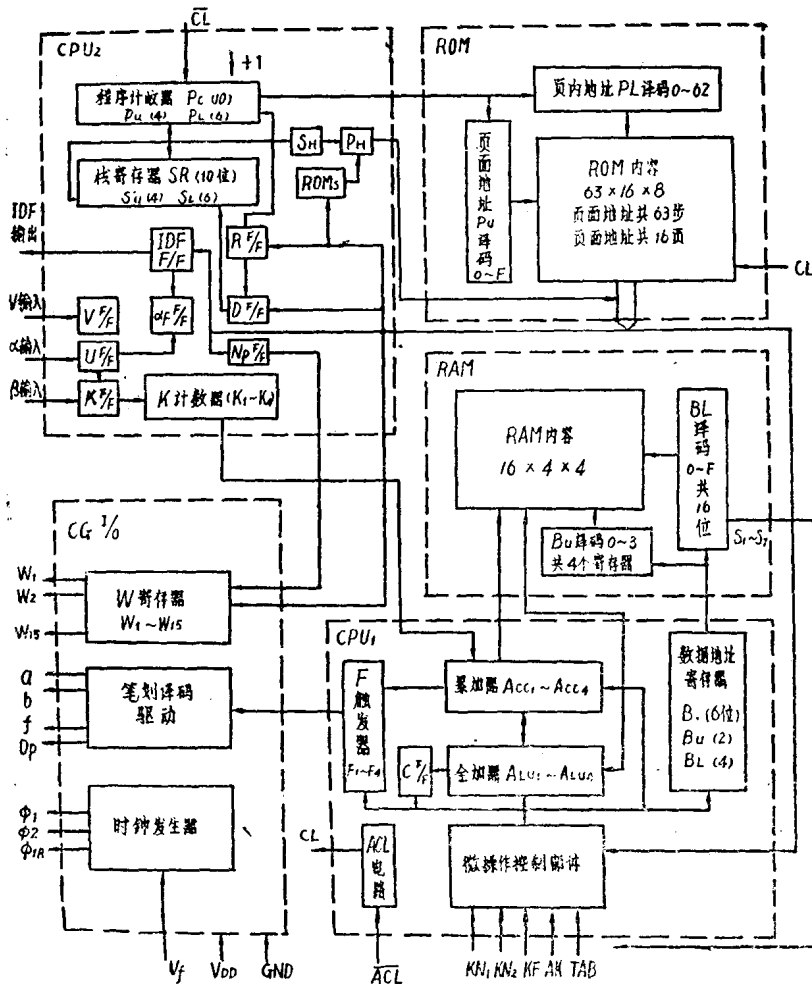


图 2—2 040 机分片框图

用户程序及常数是存在只读存储器 (ROM) 中, 程序地址寄存器分 6 位页内地址 (PL), 和 4 位页面地址 (PU) 组成 10 位程序地址寄存器 (Pc), 6 位页内地址总共有 63 步前进状态, (注意不是 64 步), 除了执行转移指令和返回指令, 在其它程序执行过程中都自动增加 1, 指向下一步程序地址, PU 是 4 位页面地址, 总共有 16 页, 本机页面分配如下: 0~3 页为子程序页, 4~15 是主程序页, 图 2—3 为 ROM 配置图。因而本机一块 ROM 容量是 $63 \times 8 \times 16 = 8064$ (bit) 页面地址和页内地址, 不是 63 进一关系, 在任何情况下, 页面地址只能由指令来修改, 这一点特别要注意, DJS-040 是如何转页的以后要专门讨论。SR 是为一级转子而设置的栈寄存器, 在调用 0 页子程序时, 将下条程序地址 PL_{+1} 及本页页数 PU 推入 SR 寄存器, SL 和 SU 中在从 0~3 页子程序返回时, 自动将 SR 内容弹出至 Pc 指示器。

本机 ROM 最大可扩至 2 K 字节它是通过设置的片选触发器 ROMs 完成的。开机时自动清洗 ROMs, 此时自动指向 0* 片 ROM, 若要转向 1* 片 ROM 时, 须先发 71 指令, 使 ROMs 翻转一次, 切换至 1* 片 ROM, 因而根据触发器 ROMs 状态是 “0” 还是 “1” 决定 0* 或 1* 片。ROM 数据送上指令总线, 这样就完成了 ROM 扩充。同样在从 1* 转 0*

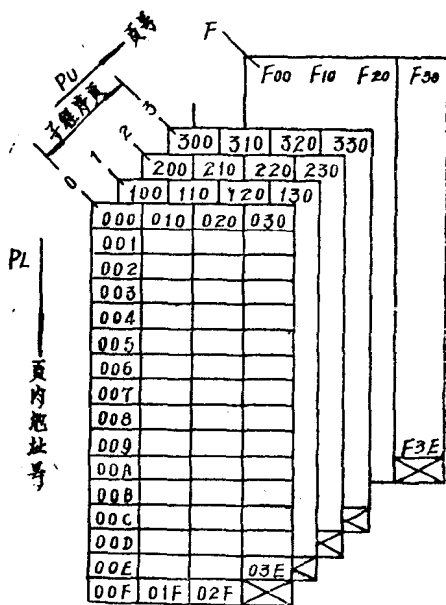


图 2-3 ROM 页配置

ROM 时也必须先发 71 指令, 使 ROM_E 触发器再翻转一次, 同样为了调用任意一块 ROM 子程序, 增设了 ROM_E 的栈保护器 SP。

二、数据存储器和数据地址控制:

DJS-040 机的数据存储器采用 6 管静态单元组成的随机存储器 (RAM), 它是本机数据存储器区域和特征寄存器, B 寄存器是 RAM 地址指示器, 分页内地址 (BI) 4 位及页面地址 (BU) 2 位, BI 可以由指令加 1 减 1, 总共有 0~15 位, 页面地址仅由指令修改, BI 与 BU 关系也是非 16 进一。RAM 总共可以由 64 个字组成, 4 位并行输出, 它的组成形式 $16 \times 4 \times 4$, 在 RAM 当作寄存器使用时, 可以组成 4 个 16 位字长的寄存器, 当然根据用户需要, 还可以编成其它各式各样形式, 指令可以对 RAM 某些位进行位操作, 对各自位可以分别置 “1” 复 “0” 或测试。BI 通过译码器输出至 $S_1 \sim S_7$, $S_1 \sim S_7$ 的状态分别对应的 BI 值为 F, E, D, C, B, A, 9, 共 7 种, 并且当进位触发器是 “1” 的情况, $S_1 \sim S_7$ 与 BI 无关, 全部输出为 “1”。S 通常作为扫描地址使用, 它与 I/O 可测试输入端, 即: KF, AK, TAB, KN₁, KN₂, 构成矩阵, 以作键盘或其输入转换用扫描阵列。

三、算术逻辑部件:

DJS-040 的 ALU 是一个 4 位并行处理的算术逻辑单元, 含有进位触发器 C, 根据指令对累加器 ACC 立即数, RAM 中的数据, K 计数器进行传送和加算操作, 进位触发器可以由指令置 “1” 或复 “0” 并进行测试, 进位触发器可以参加运算, 也可接收最高位的进位值, 4 位的 F 触发器接收来自 ACC 的数据, 也可以由累加器对它进行屏蔽输出。

四、输出输入通道:

DJS-040 有丰富的输入、输出, 所有 I/O 线都可以由指令来调度或被测试。W₁~W₁₅ 是一个串行输入, 并行输出 15 位的移位寄存器它的输出可以用 NP 触发器禁止。异步输入端 α 、 β 、V 机内都有时序同步及触发器相对应, 因而这三个异步输入端是可以测试的。4 位 K 计数器用以对 α 输入进行计数, 由于 K₀ 对 α 信号一次分频, 因而 K₀ 与 4 位计数器对 α 输入的计数值最大是 $2^4 - 1$ (除去原始状态 “0”) K 计数器值可转移到累加器 ACC 中。IDF 可以由指令置 “1” 或复 “0”。 α 、 β , 可以分别作为打字机的字符信号和印刷终止信号, 利用 W₁~W₁₅ 输出作 15 行台式打字机字锤驱动信号几乎可以无需其它任何组合逻辑芯片, (W₁~W₁₅) 也可作为动态扫描荧光数码管栅极控制信号, 当然在程序控制下这些 I/O 出线端还可以灵活使用于其它场合。

五、振荡及显示 PLA 矩阵:

振荡及时序电路是内藏的, 通过调节 V₁ 的输入电压可以改变时钟振荡频率, 从而改变了

本机三相时钟 ϕ_1 , ϕ_2 , ϕ_{12} 的周期, 显示 PLA 矩阵是紧接在 $\overline{F}$ 输出端上的, 由它组成 7 段笔划信号。

译码输出, 用它的 W 输出作扫描显示, 既可以有 $\overline{F}$ 的直接输出, 又可以有数码管 7 段输出。

DJS-040 系统的进一步了解请参照图 2-4 这里对本机时钟 ROM, RAM 扩充, ALU 单元及数据流通方向都绘出了一个轮廓的说明, 它对于熟悉本机详细硬件系统提供了帮助。

§ 2-4 操作时序

DJS-040 的操作时序在图 2-4 下方已很清楚表达了, 这些波形图能较好的说明本机工作时间关系, 为了进一步阐明本机工作时序, 先对所有寄存器作简单介绍, 寄存器使用的触发器如图 2-5 所示, 图 2-6 是一个转移波形, 它指出了输出和输入波形关系。

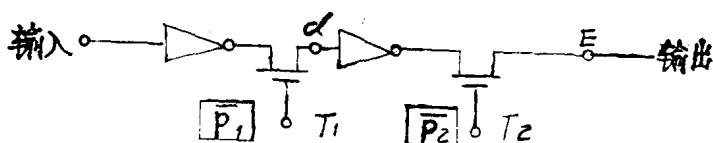


图 2-5

信息从输入转移到输出, 必须在 $\overline{\phi_1}$ 的下降沿之前一段时间稳定, 当 $\overline{\phi_1}$ 结束, 信息便靠 d 点结电容维持, 当 $\overline{\phi_2}$ 前沿一到, 信息便通过导通管 T_2 向 E 输出, 这种触发器的形式如同主从触发器 (D 型触发器)。在本机触发器中输入信息是由 $\overline{\phi_1}$ 存入前段在 $\overline{\phi_2}$ 一到送出, 这仅是一个一位移位寄存器, 作为一个有维持状态的触发器还必须有自保功能, 自保只须将其输出信息再原样送入输入端便可。

DJS-040 的所有寄存器的输出 (除了 ROM 地址寄存器 PC 以外) 全都是 $\overline{\phi_2}$ 的前沿为准, 并且至少维持到下一个 $\overline{\phi_2}$ 前沿, 在看详细的硬件讨论时需注意这一点, 而程序 ROM 地址指示器的输出采用了通常所用方法, 即提前于指令执行周期就放出。本机提前 4 μ S 这是考虑了地址译码稳定及增加指令执行有效周期而采取的措施。从图 2-3 ROM 框图中还可以很清楚地看出, 本机 ROM 扩充方法, 两块 ROM 的输出是通过一个三态门而挂在指令总线上的, 三态门控制是由 P_H 执行, 三态门之前的禁止端 $\overline{R_F}$ 是本机跳步控制, 当执行跳步操作, 因 $\overline{R_F} = 0$, 所以全部输出为 0 是控制空操作, 而空操作时, P_L 照常增 1, 因而跳步被执行。

RAM 的扩充分 2 段控制 (a) 输出端三态控制, 原理如同 ROM, (b) 低位地址选通也受 RAM。状态影响, 没被选中 RAM, 由于地址没有一个被选中, 因而在写入时刻不会有信息进入, 因而达到保护原有信息目的, 这样 RAM 就扩充了一倍。

在图 2-4 中, 将 ALU 的框图提供给读者, 它仅绘出了数据通路, 而所有这些数据都

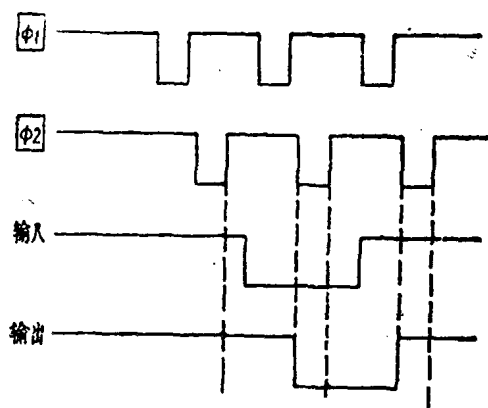


图 2-6 转移波形

是受相应微操作条件控制, 这里省略了, 仅提供了进一步阅读详细硬件图的引导。RAM 工作时序不同于 ROM, 因此在一条指令中对 RAM 可以读一次同时写一次, 如 $M \leftrightarrow A$ 交换指令, $M \rightarrow A$ 和 $A \rightarrow M$ 是在一条指令周期中完成的。

为了使读者对本机有一个了解, 以执行 EXCDy 指令为例说明本机的操作时序见图 2

—4 EXCDy 执行 $\begin{cases} M \leftrightarrow A, \\ BU \forall y_2 y_1 \\ BL-1 \rightarrow BL, \end{cases}$

执行这些操作的步骤如下:

(1) ROM 地址周期: 在 Φ_1 一到即 t_1 时刻, 见图 2—4 本指令所在的地址码 PU, PL。通过 Φ_1 门管送 ROM 地址译码器, 延迟 τ 后, 在 t_1' 时刻稳定, 由 $t_1' \sim t_2$, ROM 中的 8 位指令代码 $R_1 \sim R_8$ 被取出, 送至指令暂存器门控管 X 点 (见图 2—4) ROM 部分当 Φ_2 一来, 即 t_2 时刻指令码输出, 开始指令周期, 注意 ROM 的地址这个周期维持到下一个 Φ_1 来, 即 t_4 时刻结束。

(2) 指令周期: 在 t_2 时刻八位指令代码输出, 在 ROM 中 Φ_2 门管关闭时, 信号以电荷的形式存在结点电容上, 一直维持到下一个 Φ_2 来即 t_6 时刻才改变状态所以指令周期是从 $t_2 \sim t_6$ 。

(3) RAM 的地址周期, 即对 BU, BL 译码指定其中一位, 这个周期是从 $t_3 \sim t_6$ 。

(4) RAM 的读出周期, 在这个周期中先对指定的一位读出即从 $t_3 \sim t_4$, 此时执行 $M \rightarrow A$, 信号暂存在 ALU 中而此信号要到下一个 Φ_2 来即 t_6 时才被 Acc 接收。

(5) RAM 的写入周期, 在这周期中对被指定的一位写入, 是把原来 Acc 的内容, 送至 RAM, 即 $A \rightarrow M$, 这是在 $t_4 \sim t_6$ 时执行的。

在这个指令周期中还要执行 $BU \forall y_2 y_1$ 及执行 $BL-1 \rightarrow BL$ 的操作, 即指定 RAM 的下地址。

(6) 指令执行后的状态, 前面已经讲过 $M \rightarrow A$ 的数要在下一个 Φ_2 来的 t_6 时刻被 Acc 接收, 所以指令执行后的状态是从 $t_6 \sim t_7$ 图形中用 (2) 表示, 见图 2—4。

$BU = BU \forall y_2 y_1$ 及 $BL = BL-1$ 也要在 t_7 时形成, 这是决定了 RAM 中下一条指令周期的地址。图 (2—4) 中用 (1) 表示指令执行时的状态。用 (2) 表示指令执行后的状态。

§ 2—5 DJS—040 开发系统

自从 75 年 1 月 TI 公司首次发表 TMS 1000 系列单片 4 位微计算机后 4 位机在国际市场上引起了一场激烈的竞争, 一些具有远见的厂商都在研制开发系统中大力投资, 由于 4 位机的价格和成本不可能象 8 位, 16 位机那样, 用一个通用随机存储器放置用户程序, 因而在开发系统中首先要解决的是程序模拟和实时硬件模拟, 从某种意义上讲, 这些支援系统都比 4 位机本身重要得多, 如果没有支援系统, 对于日益增高的市场需要是无法满足的, 而且将使 4 位机使用变得非常麻烦, 这就限制了 4 位机本身的发展, 由于我们刚开始从事 4 位机的研制和应用, 因而对 4 位机开发系统还没有全面的概念。

我们打算在 4 位机研制系统中作一些工作 (图 2—7) 列出了 DJS—040 开发系统流程框图, 仅供参考。

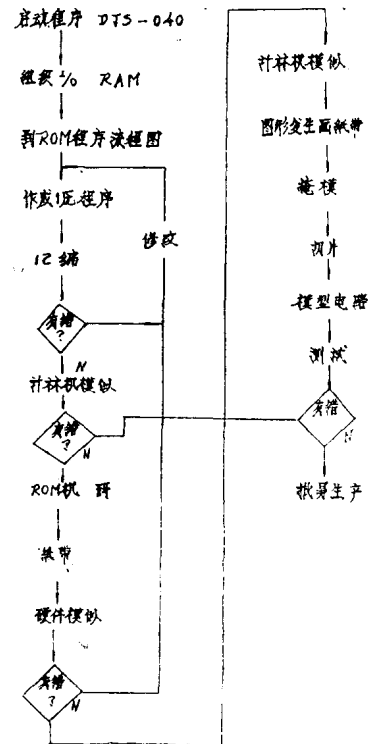


图 2—7 开发系统流程图

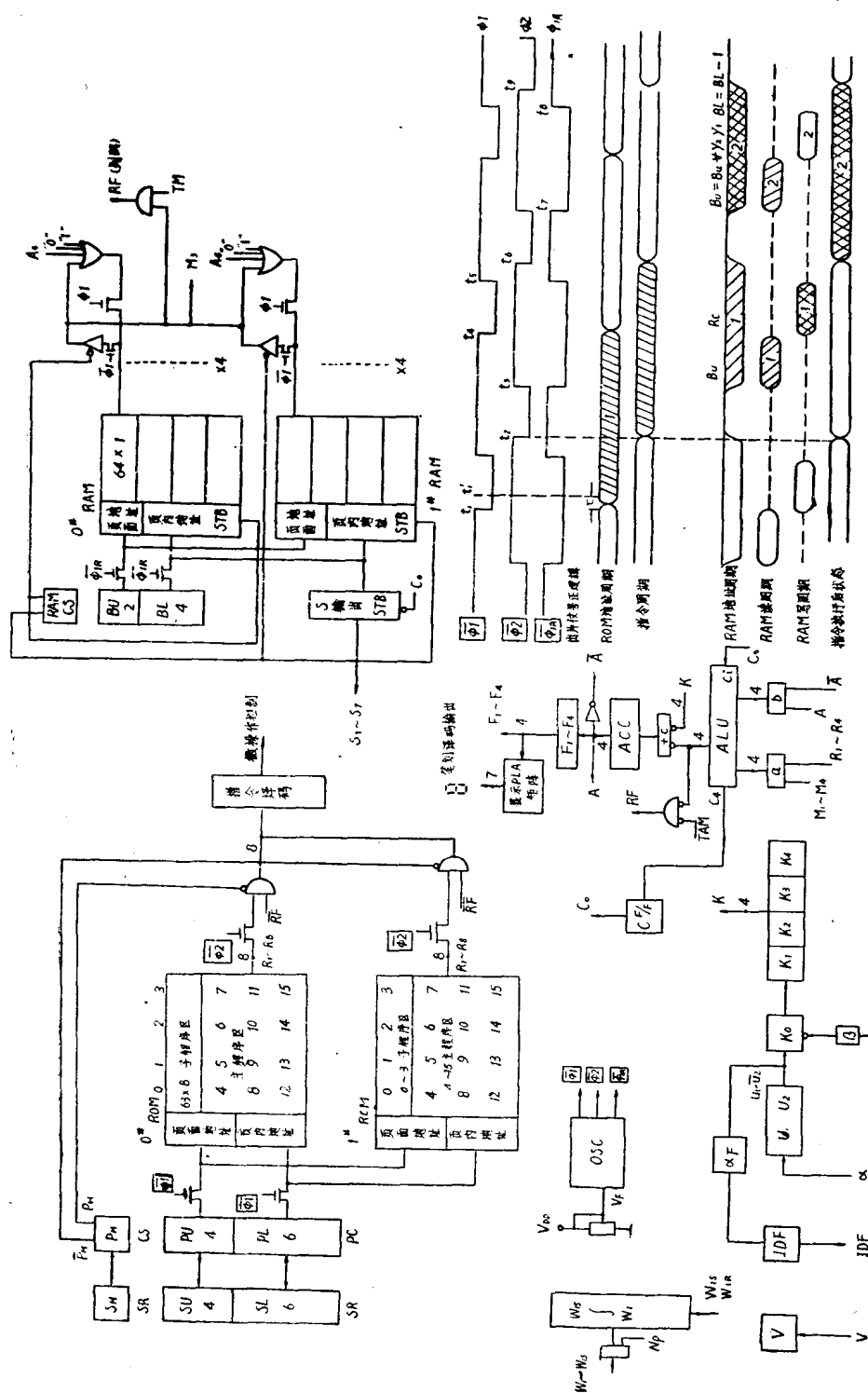


图2-4 DJ S-040主要数据流及工作时序