

中央处理器 与

部分外围电路说明

0 5 1 中央处理器及部分外围电路说明

0 5 1 微处理器和 DJS-051 微型计算机是在四机部的组织下，由上无十四厂与安徽无线电厂协作会战的项目。在会战中受到 207 所的大力支持，由于各级领导的大力支持和兄弟单位的积极配合，使会战得以顺利进行，大约一年的时间内完成了 CPU 及部分外围电路的研制工作。

这里仅将中央处理单元 (CPU) 及现有的几种外围电路片作一简单说明，以供用户参考。由于编写匆忙，错误之处望指正。

上无十四厂技术科

目 录

第一部分

概 要	1
-----	---

一、051CPU的总体结构

1. 051-1 微程序控制 ROM-CROM	3
2. 通用寄存器 (GR)	5
3. 算术及逻辑单元 (ALU)	9
4. 051-4 微程序控制单元 MCU	9

二、指令系统

1. 指令和数据格式	12
2. 寻址型式	13
3. 条件标志	14

三、外围电路

1. 051-12.8 位输入/输出	15
2. 051-05.0~7 时序译码器	19
3. 051-16.4 位双向总线驱动器	20
4. 百板管理程序 ROM	21

第二部分

一、DJS-051 CPU 与存储器及 I/O 接口简解

1. ROM 的接口	28
2. RAM 的接口	29
3. I/O 接口	29

三 DJS-051 存储器

- 1. 概述----- 31
- 2. 存储基片----- 32
- 3. 系统结构----- 36

概 要

051 微处理机是由四片 N 沟硅栅 MOS LSI 组成的完整的八位中央处理单元 (CPU)。这种片子为用户提供了能高性能解决控制和处理的手段。在功能上代替了 8080A 和系统控制器 8228。

051 有三个总线:

1. 具有三态能力八位双向数据总线 (D_0-D_7) 用以传递数据和信息。

2. 具有三态能力的十六位地址总线 (A_0-A_{15}) 来发送存贮器和外部设备的地址, 因此寻址范围可达 64K 字节。

3. 控制总线由读 (R)。写 (W)。(存贮器外设 IO/M) 中断响应 (INTA) 保持响应 (HLDA), 等待 (WAIT), 停机指示 (HLTA), 指令取指示 (FETCH) 等八根输出线及五个控制输入 (HOLD, ϕ , WAIT, READY, INT, RESET) 及时钟信号 ϕ_1, ϕ_2 和电源输入 (+12V+5V-5V))

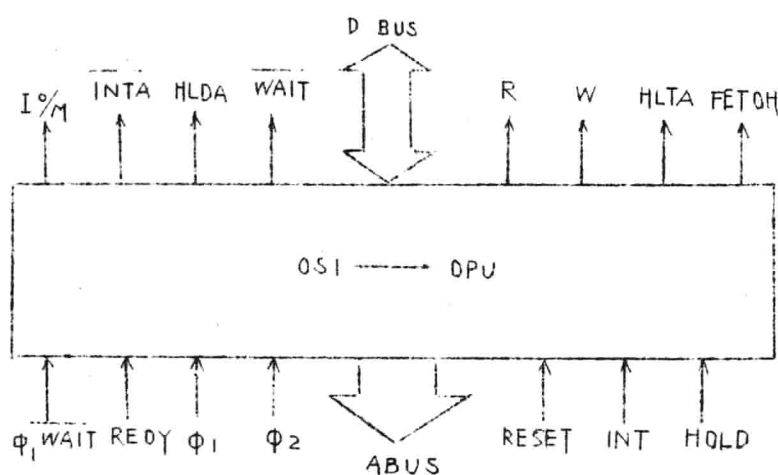


图 1 051 CPU 引出线路

051 CPU 其软件和 8080A 完全相容。CPU 的输出引线与 TTL 电路相容 (CPU 四片电路之间互相连接仍为 MOS 电平)。

051 CPU 采用了微程序设计, 在完成基本指令 (CPU 由四片组成) 的同时再增加一片控制 ROM (CPU 由五片组成) 还可以再扩充若干条指令 (扩充指令将在 79 年底完成)。

051 CPU 引出线功能说明

A, $\bar{A}$ 。地址总线输出三态; 地址总线提供高达 64K 的八位字节的存储器地址或者 256 个输出设备及 256 个输入设备的 I/O 设备代码, $\bar{A}_0$ 是最低的地址位, 当 HOLD 时三态。

D, $\bar{D}$ 。(输入/输出三态) 数据总线; 数据总线供 CPU 和存储器 I/O 设备之间进行指令和数据传达的双向道路。D₀ 为最低位。

IO/M (输出): 区分是 I/O 设备还是存储器操作。

INTA (输出): 中断响应, 在中断请求被接受后代替读 (R)。

H LDA (输出): 对保持的响应, 表示 CPU 已接收了保持请求。数据总线和地址总线挂起 (高阻状态)。

WAIT (输出) 等待: 此信号用来识别 CPU 处于等待状态之中。

R (输出) 读: 表示被选中的存储器或 I/O 设备被读出。

W (输出) 写: 表示在总线上的数据要写入被选择的内存或 I/O 设备。

HLTA (输出) 停机响应: 表示 CPU 处于暂停状态 (开漏输出)

FETCH (输出) 取指令: 表示取指令 (开漏输出)

WAIT (输入): 非等待时使微程序连续运行

READY (输入) 准备完毕: READY 信号向 051 CPU 指明存

贮器或输入数据可以供给 051 CPU 数据总线, 该信号常用来与慢速存贮器或 I/O 设备同步工作。

Φ_1 (输入) 输入时钟 1

Φ_2 (输入) 输入时钟 2

RESET (输入) 复位: 当 RESET 信号出现时, 程序计数器的内容被清除; RESET 之后程序从零开始。且内部 INT 触发器复位。

INT (输入) 中断申请; CPU 在执行现行指令时, 或是在暂停时, 不断在查询此输入, 当有中断申请, 且前已执行过“开中断”(EI)指令时, 产生中断。

HOLD (输入) 保持: 保持信号请求 CPU 进入保护状态, 当 HOLD 输入时, 微程序执行到查保持, 则使 CPU 在下一个节拍处于保持状态(数据总线, 地址总线处于三态)。

V_{DD} (输入); N-MOS 电路电源 +12V

V_{CC} (输入); 与 TTL 相容的电源 +5V

V_{SS} (输入); N-MOS 电路衬底偏置电源 -5V

 ; 接“0”

一 051 CPU 的总体结构

051 CPU 由下列四片 LSI 组成

* 051-1 微程序 ROM (CROM₁)

* 051-2 通用寄存器 (GR)

* 051-3 算术逻辑单元 (ALU)

* 051-4 微程序控制单元 (MCU)

二 051-1 微程序控制 ROM-CROM₁,

051 CPU 采用微程序设计, 由该电路发出二十四条微程序控制信息, 去分别控制通用寄存器 (GR) 和算术逻辑单元以及读 (R),

写(W)和查保持(H)及MCU的微程序转移信息。

该电路实际上为6.0K容量的ROM，并带有二十四条并行输出缓冲寄存器，构成256条微程序×24输出的阵列。

输入微程序地址九条(MA_0-MA_8)· MA_0-MA_8 ，产生256条微程序地址， MA_0 可以控制二十四位并行输出为三状态，以使微程序再扩充256条，使在基本指令73条的基础上再增加一片CROM，将指令大大扩充。

工作原理：

在第一个 Φ_1 期间，对输入地址译码电路进行预充，在 Φ_2 期间ROM阵列输出码点信息。而在第二个 Φ_1 期间信息打入输出缓冲寄存器并输出，形成6根输出至GR的 K_0-K_5 ，6根线送到ALU的 K_0-K_5 ，9根输入到MCU的JC，—，以形成微程序转移，以及H.W。

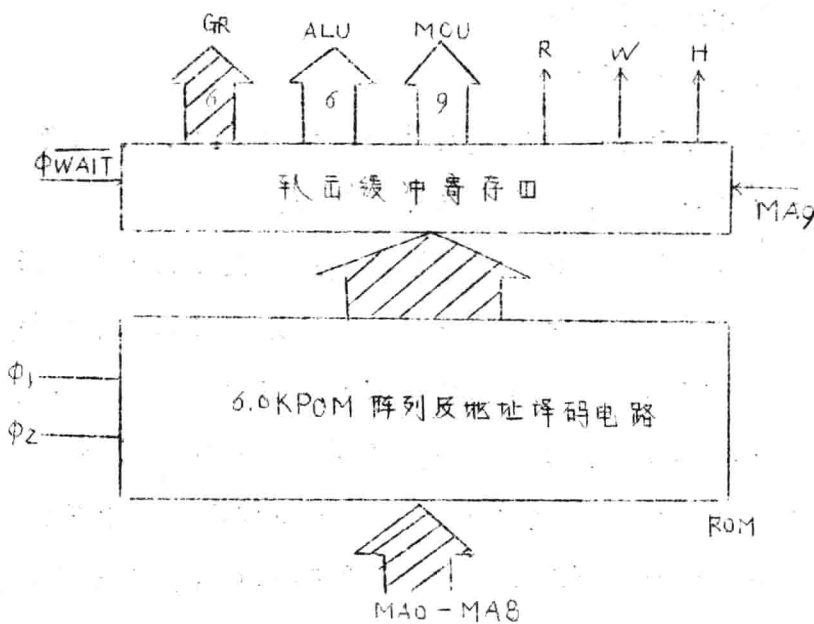


图2 CROM框图

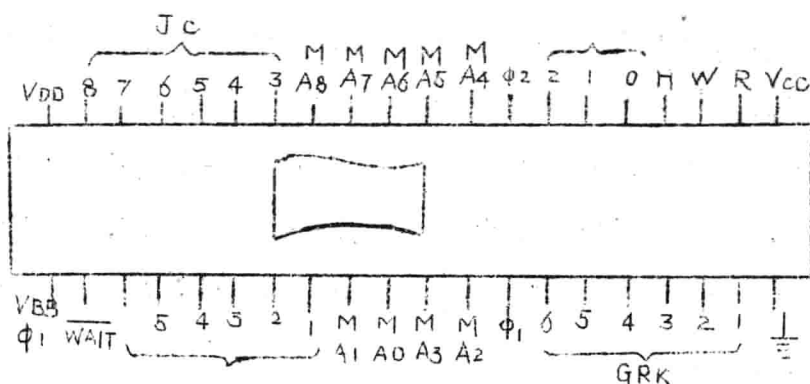


图3 051-1 管脚引线排列图

R 三根控制信息输出，详见微程序码点图。

2 通用寄存器 (GR)

通用寄存器 (GR) 包含一个静态读写存贮 (RAM) 阵列，它由 6 个十六位寄存器构成：

- * 程序计数器 (PC)
- * 堆栈指示器 (SP)
- * 六个八位通用寄存器成对排列，称为 B、C、D、E 及 H、L。
- * 一个称为 W、Z 的成对暂存寄存器。

程序计数器保存现行指令的存贮器地址，每当指令取出后，它就自动加 1。堆栈指示器保存下一个可供使用的存贮器中堆栈单元的地址。堆栈指示器能够设置初值，因而可使用读写存贮器的任何部分作为堆栈，堆栈指示器在数据被“压入” (PUSH) 堆栈时减 1。当数据被“弹出” (POP) 堆栈时加 1 (即堆栈是向下增长的)。

6 个通用寄存器既可以作为单个的寄存器 (八位) 使用，也可作为寄存器对 (十六位) 使用，(如 BC、DE、HL)。暂存寄存器对

W. Z. 不能用程序来选址，只在内部执行指令时才用到它。

通用寄存器（八位）和CPU的内部数据总线之间的数据传送，通过多路开关来进行。十六位寄存器对的数据能在加1、减1器和地址锁存器之间传送，地址锁存器可以从寄存器对中的任何一个接收数据并送至十六位地址输出缓冲器（ $A_0 - A_{15}$ ）作为地址信息输出。

寄存器（或对）的选择以及数据的传送，由控制信息码 $K_6 - K_1$ 来控制。由控制信息码 K_6 、 K_5 、 K_4 产生的IO/M输出信息，用以区别系统是I/O操作还是M操作。

$A_0 - A_{15}$ 地址总线具有三态能力，由输出线TS(three state)控制，当 $TS = 1$ 时三态。

为了扩充指令，该电路增加了一条 Z_{out} 输出头，它意味着当加1“减”1电路作减1时，减到全0该输出头为“0”（基本指令不用）， $K_6 - K_1$ 由微程序CROM送来，以控制GR的操作。

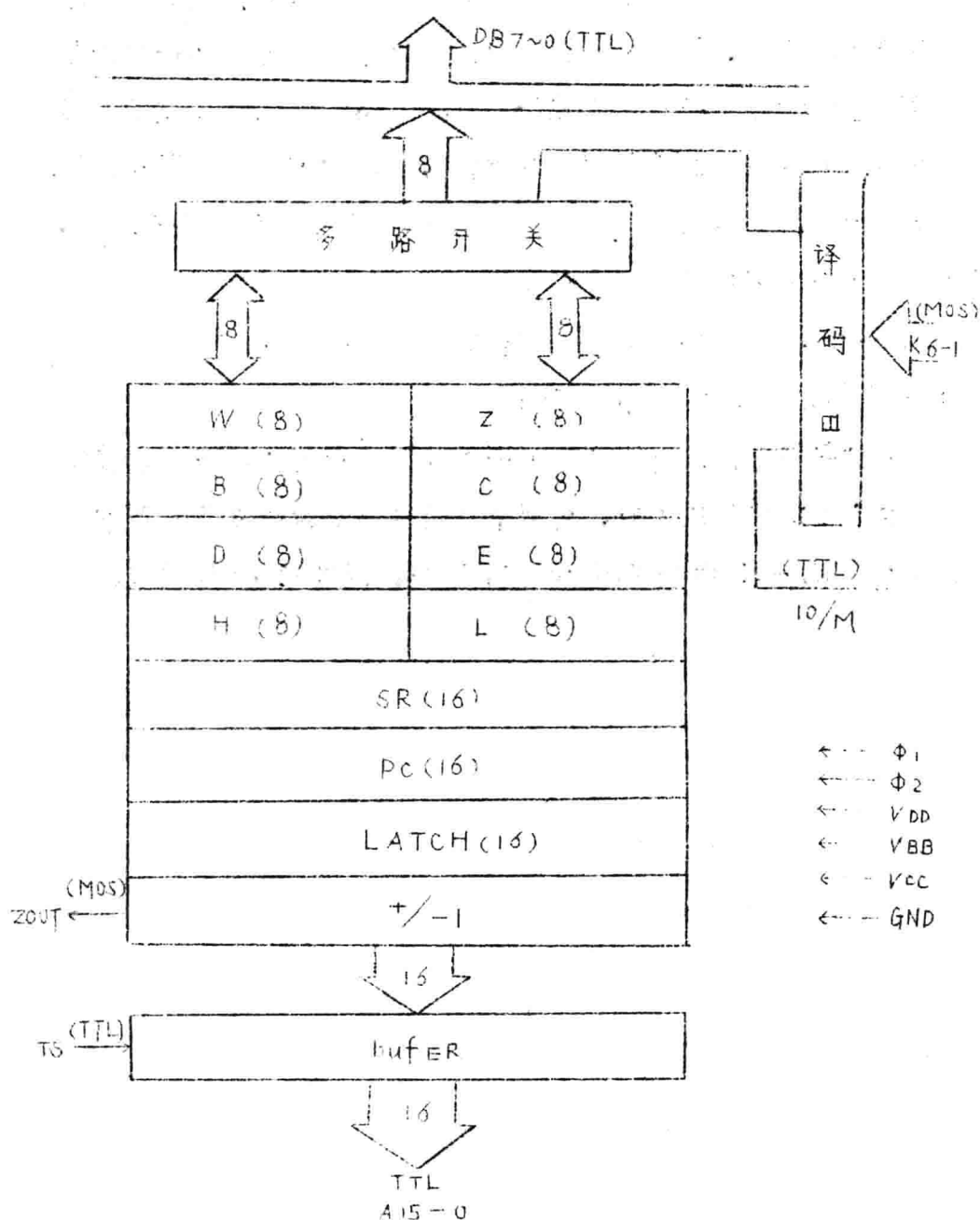


图 4 CPU-2 逻辑框图

GR 控制信号 $K_0 - K_7$ 功能说明

K_0	K_1	K_2	寄存器对选择
0	0	0	无操作
0	0	1	BC
0	1	0	DE
0	1	1	HL
1	0	0	PC
1	0	1	SP
1	1	0	WZ
1	1	1	WZ, 1/0

K_3	K_4	K_5	操作方式
0	0	0	LA 写入 YP
0	0	1	YP 读入 LA 且 LA+0
0	1	0	YP 读入 LA 且 LA+1
0	1	1	YP 读入 LA 且 LA-1
1	0	0	数据写入 YPH
1	0	1	数据由 YPH 读出
1	1	0	数据写入 YPL
1	1	1	数据由 YPL 读出
1	X	1	数据线输出三态

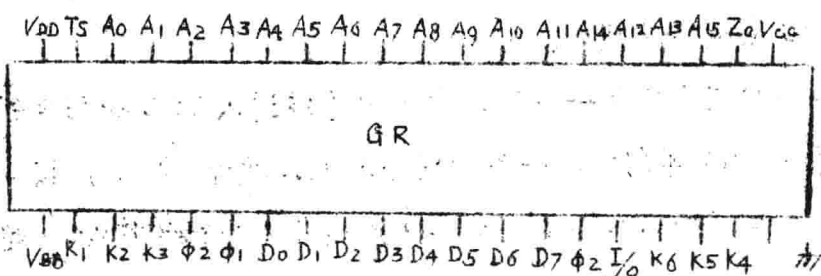


图5 GR的管脚图

3. 算术及逻辑单元 (ALU)

ALU 包括一些寄存器:

- * 一个八位累加器 A
- * 一个八位暂存累加器 (ACT)
- * 一个八位运算缓冲器 (ABU)
- * 一个八位暂存器 (TMP)
- * 一个五位条件标志寄存器、零位、进位位、符号位、奇偶位及辅助进位位。

算术、逻辑及循环移位等操作是在 ALU 中进行的, ALU 的数据由暂存寄存器, 加入 ACT 以及进位触发器来供给, 操作结果能送到 ABU 缓冲寄存器; ALU 也能传送信息给标志寄存器。

该 ALU 电路具有六十四微操作, 由 K_0-K_6 六个操作码组成的 PLA 阵列来控制, 实际上基本指令所用的微操作仅 46 条, 其余供扩充指令用, 微操作详见表一 ALU 微指令表。

为执行 DAA 指令, 内部布有十进制调整线路。

此外在该电路中, 尚有一个查保持信号的逻辑门, 实际上是为组成系统用的, 而与本电路无关系 (仅为利用 ALU 的管脚)。

4. 051-4 微程序控制单元 MCU

051 CPU 采用了微程序设计思想。每一条基本指令都由若干条微操作按程序执行—即微程序来完成, 基本指令的微程序详见 051-CPU 微程序流程图。

每条基本指令的第一字节, 由数据总线送入 MCU 的指令寄存器 (IR) 再由 PLA 阵列译出一条微程序的入口地址 MA_0 。该入口地址被送到微程序控制 ROM (CROM) 产生 24 个微操作码, 其中六

个($MJR_0 - MJR_5$)用以控制通用寄存器堆(GR), 六个($MIR_0 - MIR_5$)用以控制算术逻辑单元, 九个反馈送回到微程序控制单元以产生下一条微操作的地址形成微程序转移, 这样由微程序控制单元(MCU)和微程序ROM组成就形成了一直循环着的微程序操作。

该电路片中包含有中断请求、中断响应、保持响应、复位、准备完毕等控制电路。

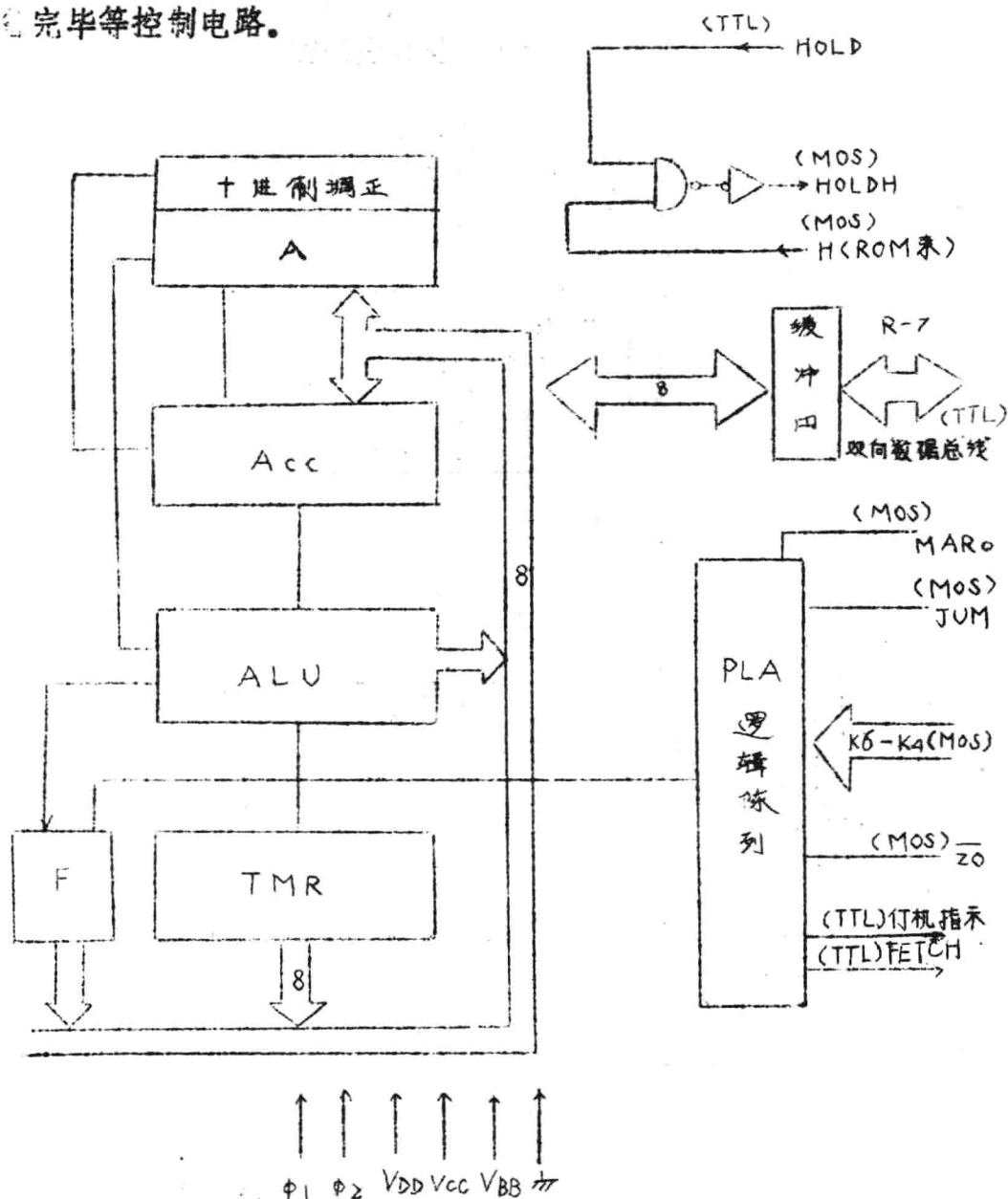


图6 ALU的逻辑框图

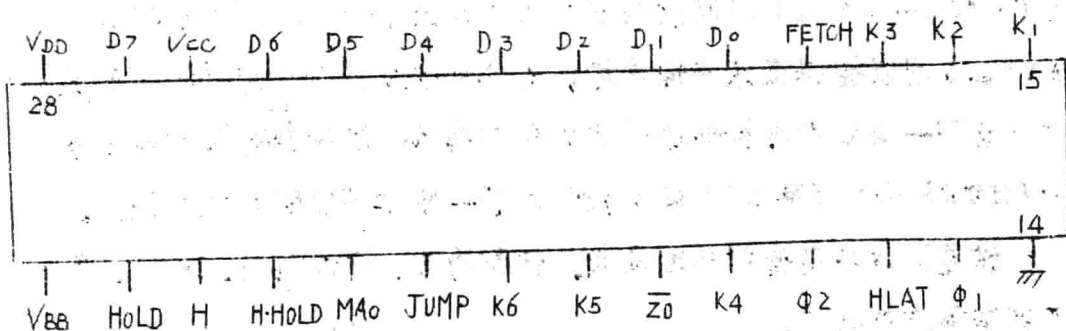


图7 ALU的管脚引线图

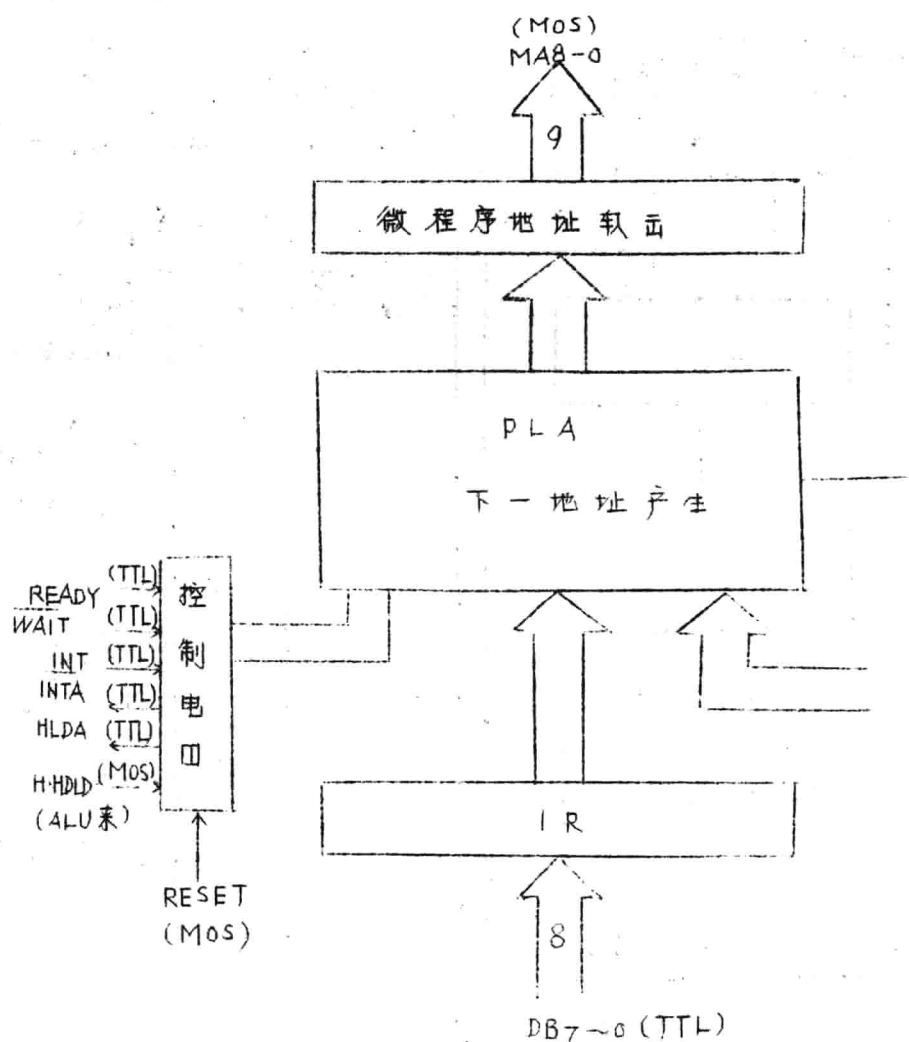
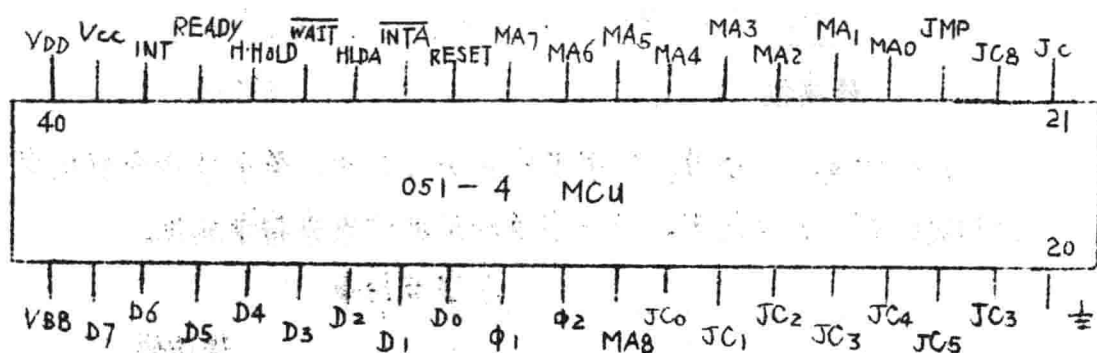


图8 051-4 微程序控制器(MCU)方框图



051-4 微程序控制器 (MCU) 管脚图

三 指令系统

051 CPU 的基本指令有五种类型:

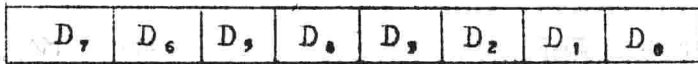
- * 数据传送类—在寄存器之间或存贮器与寄存器之间传送数据。
- * 算术类—各寄存器中或存贮器中的数据做加减, 加 1 或减 1 等。
- * 逻辑类—各寄存器或存贮器中的数据进行“与”“或”, “异或”比较, 循环移位或补码操作。
- * 转移类—条件及无条件的转移指令子程序的调用指令及返回指令。
- * 堆栈。I/O 及机器控制类—包括 I/O 指令以及为管理堆栈和内部控制标志的指令。

1. 指令和数据格式:

051 的存贮器由并行八位 (bit) 称为字节构成, 每个字节对应一个 16 位两进制地址。

051 能直接寻址达 65536 个存贮器字节, 存贮器包括只读存贮器 (ROM) 和随机存取存贮器 (或称为读写存贮器 RAM)。

数据在 051 中按八位二进制整数形式:

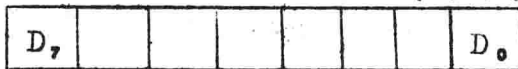


最高位

最低位

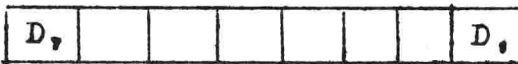
指令长度有一字节、二字节和三字节 3 种, 多字节指令应该存在相继的存贮器单元中, 第一字节的地址带称为指令地址。

单字节指令

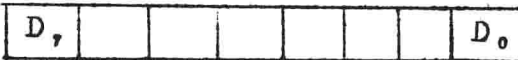


操作码

双字节指令

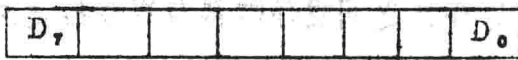


操作码

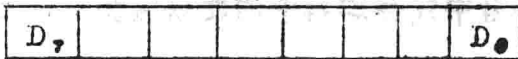


数据或地址

三字节指令



操作码



数据或地址

2 寻址型式:

通常用来操作的数据是存在存贮器里。当使用多字节数字时, 数据和指令一样存在存贮器相继的单元中。低位字节在前, 随着是逐渐增高的高位字节: 051 对存在存贮器或寄存器的数据有四种不同的寻址型式:

- * 直接寻址——指令的第二及第三字节为确切的存贮器地址的数据项 (地址的低位是在第二字节, 高位在第三字节)。

- * 寄存器寻址——在指令中就指定放着数据的寄存器或寄存器对。