

DJS-050 微型电子计算机

清华大学微型计算机组

一、前言

DJS-050 电子计算机是一台微型电子计算机。采用 P-MOS 集成电路器件。根据四机部的指示,由安徽无线电厂,清华大学电子系,四机部六所三单位于1974年8月开始联合设计工作。

在设计过程中,考虑到下列几个原则:

1. 作为系统核心的中心处理器(CPU)应具有较强的通用性和灵活性,以便适应各种低速的数据处理和过程控制;

2. 为了便于推广,机器的基本结构尽量简单,但系统上灵活可变,必要时可组成多机系统来提高可靠性和处理效能;

3. 配备必要的软件和小型化的入/出设备;

4. 为了缩短研制周期和加速投产推广的过程,第一台样机设计重点是解决功能和可靠性。器件的集成度和速度根据国内先进而有把握的水平来决定。

在安徽无线电厂,清华大学电子工程系,中国科学院自动化研究所,北京四机部六所,北京前门器件厂,北京崇文电子仪器厂等单位的共同努力下,76年研制出第一台样机,77年4月份在合肥开了微型电子计算机交流会。目前试制了六台样机。

二、050微型电子

计算机的性能

1. 050中心处理器(CPU)的主要性能

基本字长8位,内部运算能力可扩充到16位,并行操作;被处理的数据字长可由程序任意连接到若干字节。

微程序控制,有多重转子返回能力,还有256字 \times 32位的控制存储器可供扩充指令。

有五种不同字长的指令,长度分别为1,2,3,4,5字节,有直址,间址访内存能力;基本指令76条;还可以根据需要而编制扩充指令或专用指令。

通用寄存器32只。

后进先出程序堆栈,容量32字 \times 16位,可供四级中断和16重转子程序之用。

存储器最大容量可配65K,可配任意类型、任意速度的存储器。

4级中断。

输入输出异步控制,采用标准接口,可连接32种以上的设备;具有快速通道的能力。

机器主频150 KC/S,基本指令执行时间6--29节拍。

能简单地组成多机系统。

采用中大规模的P-MOS集成电路,050系列组件约40片。

2. 050机的存储器是多种存储器的复合体

磁心随机存储器8K字,存取周期2 μ S。

静态MOS存储器256字,周期4 μ S,集成度16字 \times 8位/片。

动态MOS存储器4K字,集成度1K字 \times 1位/片。

MOS只读存储器ROM,集成度1K字 \times 8位/片。

磁环半固定存储器。

只读存储器的容量，由用户决定。

3. 050微型计算机的外部设备

键盘：53键，68字符。

光电式纸带输入机，每秒200排孔，5/8单位通用。

行式打印机：64种字符，每秒打印2行，每行24位。

正在研制的外围设备有：串行打印机（作为控制打字机），软磁盘（作为外部存储器）和字符显示器。

4. 050微型计算机的电源

+12 V	2.5 A
-12 V	1.5 A
+5 V	1.5 A

-5 V	0.5 A
-6 V	0.3 A
-3 V	0.3 A
+18 V	0.2 A
+22 V	0.2 A

电源种类多是由于配的存储器的类型多。

5. 050机的软件

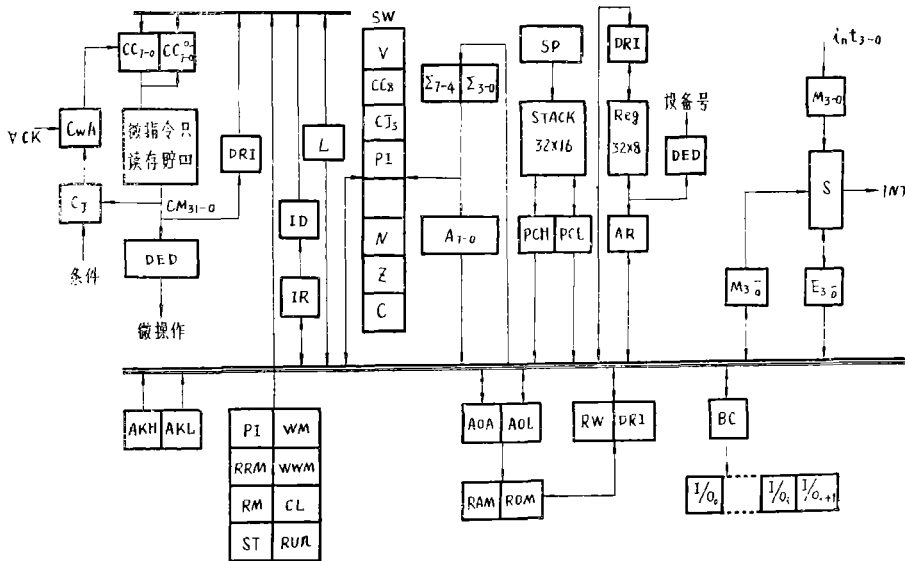
外部设备的管理程序。

汇编程序。

交叉模拟程序。

正在编制一种高级语言的编译程序。

三、050微型计算机的结构



图中 CWA 应为 C_{WA}、VCK 应为 ACK、A0A 应为 AOH。

图 1 050微型计算机方框图

符号说明：C_{WA} 节拍等待触发器

C_J 条件转移触发器

CC₇₋₀ 微程序地址寄存器

CC₀₋₀ 微程序返回地址寄存器

CM₃₁₋₀ 微指令只读存储器 输出码

DED 译码器

DR1 双向驱动器

ID 指令编码器

IR 指令寄存器

L 常数寄存器

AKH 高位排码开关寄存器

AKL 低位排码开关寄存器

PI 光电机输入键

WM	写内存单元键
WWM	写内存下一个单元键
RM	读内存单元键
RRM	读内存下一个单元键
CL	总清键
ST	启动键
RUN	继续启动键
SW	状态字寄存器
Σ_{7-4} , Σ_{3-0}	运算器
A_{7-0}	累加器
PCH	指令计数器高位寄存器
PCL	指令计数器低位寄存器
STACK	堆栈
SP	堆栈指针
AR	通用寄存器的地址寄存器
Reg	通用寄存器
AOH	内存地址高位寄存器
AOL	内存地址低位寄存器
RAM	随机存储器
ROM	只读存储器
RW	内存重写寄存器
int_{3-0}	中断请求信号
INR	中断源寄存器
S	中断排队器
M_{3-0}	中断屏蔽触发器
E_{3-0}	正在执行的中断级别寄存器
INT	有中断标志
BC	外设标准接口
I/O	外设
B	数据总线
B'	数据总线

CPU 内部各系统之间, CPU 与存贮器的接口, CPU 与输入输出设备接口处之间的连接都通过双向总线。这种结构有损失操作速度的缺点。但也有许多重要的优点。

由于封装引线较少, 器件集成度改变比较

方便。集成度可根据工艺水平而不断提高, 但结构可以相对不变。

简化了系统内部的关系。

功能部件及指令系统容易增减和修改, 使得系统比较灵活。

表中符号说明:

A: 累加器

M: 存贮器地址

R: 通用寄存器 Reg 的地址

D: 直接数

δ : 进位

R_0 : 通用寄存器 Reg 的间址寄存器

R_2R_1 : 作为访存贮器的变址寄存器

与指令系统有关的若干约定如下:

1. CPU 状态字 SW

位序: T 6 5 4 3 2 1 0

V CC₈ CJ₃ PI N Z C

PI: 执行纸带输入指令时, 区分奇数排孔, 偶数排孔的标志。

CJ₃: 累加器的低四位的进位, 称半进位。

CC₈: 扩充微程序运算标志。

C: 累加器进位数。

Z: 累加器 A 全另。

V: $A_7 \oplus A_6$, 在双符号位补码运算时标志结果溢出, 在其它情况下没有“溢出”的意义。

N: 运算结果为负数。

2. CPU 内部操作寄存器设备号(十六进制)

部件: SW E M AKH AKL

设备号: 00 01 02 03 04

这些寄存器可以用输入输出指令来操作, 像操作外部设备的接口寄存器一样。

3. 静态逻辑系统

050 微型计算机电路系统采用双相时钟脉冲的准静态系统。仅从逻辑功能的角度上说, 是一种完全静态的系统, 它与人们熟知的电位——脉冲组合系统截然不同, 完全不用脉冲表示信息。系统中, 所有逻辑讯息都用直流电平的高、低来表示, 高电平表示“0”, 低电平表示“1”。系统中的时钟脉冲不带逻辑讯

四、050微型计算机的指令系统

1 算 术 运 算 指 令

指令形式	说 明	A 留否	C = 1 条件	N = 1 条件	Z = 1 条件	V = 1 条件	微入 程 序口	节拍 数
1 1000 RRRR	$(A) + (R) \Rightarrow A, R \Leftarrow 0$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	42	10
2 1000 0000	$(A) + [(R_0)] \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	44	10
3 1001 RRRR	$(A) - (R) \Rightarrow A, R \Leftarrow 0$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \ominus A_6 = 1$	42	10
4 1001 0000	$(A) - [(R_0)] \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \ominus A_6 = 1$	44	10
5 1010 RRRR	$(A) + (R) + (\delta) \Rightarrow A, R \Leftarrow 0$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	44	10
6 1010 0000	$(A) + [(R_0)] + (\delta) \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	44	10
7 1011 RRRR	$(A) - (R) - (\bar{\delta}) \Rightarrow A, R \Leftarrow 0$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \ominus A_6 = 1$	42	10
8 1011 0000	$(A) - [(R_0)] - (\bar{\delta}) \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \ominus A_6 = 1$	44	10
9 0101 0111	加 6 修改		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	51	7
10 0000 0011 D ₇D ₀	$(A) + D \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	03	8
11 0000 0100 D ₇D ₀	$(A) + D + (\delta) \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	04	8
12 0000 0101 D ₇D ₀	$(A) - D \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \ominus A_6 = 1$	05	8
13 0000 0110 D ₇D ₀	$(A) - D - (\bar{\delta}) \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \ominus A_6 = 1$	06	8
14 0001 1001 M ₁₅M ₈ M ₇M ₀	$(A) + (M) \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	48	13
15 0001 1010 M ₁₅M ₈ M ₇M ₀	$(A) + (M) + (\delta) \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	48	13
16 0001 1010 M ₁₅M ₈ M ₇M ₀	$(A) - (M) \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \ominus A_6 = 1$	48	13
17 0001 1100 M ₁₅M ₈ M ₇M ₀	$(A) - (M) - (\bar{\delta}) \Rightarrow A$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \ominus A_6 = 1$	48	14
18 0000 1101 R'R'R'R' RRRR	$(R' + 1, R') + (R + 1, R) \Rightarrow R' - 1, R'$	留	破	$A_7 = 1$	破	$(R' + 1)_7 \oplus (R' + 1)_6 = 1$	03	27
19 0000 1110 R'R'R'R' RRRR	$(R' + 1, R') - (R + 1, R) \Rightarrow R' - 1, R'$	留	破	$A_7 = 1$	破	$(R + 1)_7 \oplus (R + 1)_6 = 1$	04	28

2 逻辑运算指令

指令形式	说明	A 留否	C = 1 条件	N = 1 条件	Z = 1 条件	V = 1 条件	微入 程序 序口	节拍 数
20 1100 RRRR	$(A) \Delta (R) \Rightarrow A \quad R \neq 0$		留	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	42	10
21 1100 0000	$(A) \Delta [(Ro)] \Rightarrow A$		"	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	44	10
22 1101 RRRR	$(A) V (R) \Rightarrow A, \quad R \neq 0$		"	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	42	10
23 1101 0000	$(A) V [(Ro)] \Rightarrow A$		"	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	44	10
24 1110 RRRR	$(A) \oplus (R) \Rightarrow A \quad R \neq 0$		"	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	42	10
25 1110 0000	$(A) \oplus [(Ro)] \Rightarrow A$		"	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	44	10
26 0000 0111 D ₇D ₀	$(A) \Delta D \Rightarrow A$		"	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	07	8
27 0000 1000 D ₇D ₀	$(A) V D \Rightarrow A$		"	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	08	8
28 0000 1001 D ₇D ₀	$(A) \oplus D \Rightarrow A$		"	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	09	8
29 0001 1101 M ₁₅M ₈ M ₇M ₀	$(A) \Delta (M) \Rightarrow A$		"	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	48	13
30 0001 1110 M ₁₅M ₈ M ₇M ₀	$(A) V (M) \Rightarrow A$		"	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	48	13
31 0001 1111 M ₁₅M ₈ M ₇M ₀	$(A) \oplus (M) \Rightarrow A$		"	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	48	13

3 移位指令

指令形式	说明	A 留否	C = 1 条件	N = 1 条件	Z = 1 条件	V = 1 条件	微入 程序 序口	节拍 数
32 0101 0000	左移(A ₇) $\Rightarrow \delta$ (A _{k-1}) $\Rightarrow A_k$ 0 $\Rightarrow A_0$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	50	7
33 0101 0001	循环左移(A ₇) $\Rightarrow \delta$ (A _{k-1}) \Rightarrow A _k (k = 7,1) (δ) $\Rightarrow A_7$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	51	7
34 0101 0010	右移(A ₀) $\Rightarrow \delta$ (A _k) $\Rightarrow A_{k-1}$ 0 $\Rightarrow A_7$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	52	7
35 0101 0011	循环右移(A ₀) $\Rightarrow \delta$, (A _k) $\Rightarrow A_{k-1}$ (k = 7.....1) (δ) $\Rightarrow A_7$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	53	7
36 0101 1111	算术右移(A ₀) $\Rightarrow \delta$, (A _k) $\Rightarrow A_{k-1}$ (k = 7.....1) (A ₇) $\Rightarrow A_7$		$\delta = 1$	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	55	7

4 传 送 指 令

	指令形式	说 明	A 留否	C = 1 条件	N = 1 条件	Z = 1 条件	V = 1 条件	微入 程 序 口	节 拍 数
37	0111 RRRR	$(A) \Rightarrow R \quad R = 0$	留	留	留	留	留	42	8
38	0111 0000	$(A) \Rightarrow (R_0)$	//	//	//	//	//	44	8
39	0110 RRRR	$(R) \Rightarrow A \quad R = 0$	//	//	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	42	10
40	0110 0000	$[(R_0)] \Rightarrow A$	//	//	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	44	14
41	0101 1000	$(A) \Rightarrow (R_2, R_1)$	留	//	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	58	14
42	0101 1010	$[(R_2, R_1)] \Rightarrow A$		//	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	50	12
43	0101 1101	$(R_2)(R_1) \Rightarrow \text{STACK } SP + 1$ PC 不变	留	//	留	留	留	53	12
44	0101 1110	$SP - 1, (\text{STACK}) \Rightarrow R_2 R_1$ PC 不变	留	留	留	留	留	54	12
45	0100 RRRR D ₇ ...D ₀	$D \Rightarrow R \quad R = 0$	//	//	//	//	//	42	8
46	0100 0000 D ₇ ...D ₀	$D \Rightarrow (R_0)$	//	//	//	//	//	44	8
47	0000 0001 D ₇ ...D ₀	$D \Rightarrow A$		//	$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	01	8
48	0000 0010 D ₇ ...D ₀	$D \Rightarrow R$	留	留	留	留	留	02	7
49	0001 0000 M ₁₅ ...M ₈ M ₇ ...M ₀	$(A) \Rightarrow M$	//	//	//	//	//	10	14
50	0001 0010 M ₁₅ ...M ₈ M ₇ ...M ₀	$(A) \Rightarrow M + (R_2, R_1)$	//	破	//	//	//	12	27
51	0001 0011 M ₁₅ ...M ₈ M ₇ ...M ₀	$(M + (R_2, R_1)) \Rightarrow A$			$A_7 = 1$	$A \equiv 0$	$A_7 \oplus A_6 = 1$	13	25
52	0001 1000 M ₁₅ ...M ₈ M ₇ ...M ₀	$(M) \Rightarrow A$		留	//	//	//	43	13
53	0010 RRRR M ₁₅ ...M ₈ M ₇ ...M ₀	$(R) \Rightarrow M - (R_2, R_1) \quad R = 0$	留	破	留	留	留	45	30
54	0011 RRRR M ₁₅ ...M ₈ M ₇ ...M ₀	$(M - (R_2 R_1)) \Rightarrow R \quad R = 0$	//	//	//	//	//	44	29
55	0000 1100 RRRR R' R' R'	$(R') \Rightarrow R$	//	留	//	//	//	02	9

5 停 机 指 令

	指令形式	说 明	Λ 留否	$C = 1$ 条件	$N = 1$ 条件	$Z = 1$ 条件	$V = 1$ 条件	微入 程序口	节 拍数
56	0101 0100	Ω^* (动态停机)	留	留	留	留	留	54	最少6
57	0101 0101	Ω (停机)	//	//	//	//	//	55	5

6 修 改 设 备 寄 存 器 内 容 指 令

	指令形式	说 明	Λ 留否	$C = 1$ 条件	$N = 1$ 条件	$Z = 1$ 条件	$V = 1$ 条件	微入 程序口	节 拍数
58	0010 0000 $I_7 \dots I_0$ $D_7 \dots D_0$	(第 I 号设备寄存器) $\oplus D$ \Rightarrow 第 I 号设备寄存器	留	留	第 I 号设备寄存器最高位 = 1	第 J 号寄存器全“0”	第 I 号寄存器两位按位加 = 1	20	7

7 转 移 指 令

	指令形式	说 明	Λ 留否	$C = 1$ 条件	$N = 1$ 条件	$Z = 1$ 条件	$V = 1$ 条件	微入 程序口	节 拍数
59	0001 0001 $M_{15} \dots M_7$ $M_7 \dots M_0$	按 C 条转 若 $(C) = 1$, $M_{15} \sim M_0 \Rightarrow PC$ 若 $(C) = 0$, $(PC - 1) \Rightarrow PC$	留	留	留	留	留	11	条转 12 顺序 10
60	0001 0100 $M_{15} \dots M_8$ $M_7 \dots M_0$	无转: $M_{15} \sim M_0 \Rightarrow PC$	留	//	//	//	//	14	11
61	0001 0101 $M_{15} \dots M_8$ $M_7 \dots M_0$	转子 $(PC) \Rightarrow STACK$ $(SP - 1) \Rightarrow SP$ $M_{15} \sim M_0 \Rightarrow PC$	留	//	//	//	//	15	11
62	0001 0110 $M_{15} \dots M_8$ $M_7 \dots M_0$	按 N 条转 若 $(N) = 1$, $M_{15} \sim M_0 \Rightarrow PC$ 若 $(N) = 0$, $(PC + 1) \Rightarrow PC$	留	//	//	//	//	16	条转 12 顺序 10
63	0001 0110 $M_{15} \dots M_8$ $M_7 \dots M_0$	若 $(Z) = 1$, $M_{15} \sim M_0 \Rightarrow PC$ 若 $(Z) = 0$, $(PC - 1) \Rightarrow PC$	留	//	//	//	//	17	条转 12 顺序 10
64	0101 0110	子程序返回 $(SP) - 1 \Rightarrow SP$ $(STACK) \Rightarrow PC$	留	//	//	//	//	56	8
65	0101 1011	间转 $(R_2)(R_1) \Rightarrow PC \Rightarrow AO$	留	//	//	//	//	57	11

8 I/O 指令

	指令形式	说明	A 留否	C = 1 条件	N = 1 条件	Z = 1 条件	V = 1 条件	微入 程 序 口	节 拍 数
66	0000 1010 I ₁I _n	(第 I _i 号设备寄存器) = A		留	A ₇ = 1	A ≡ 0	A ₇ ⊕ A ₆ = 1	00	8
67	0000 1011 O ₁O _n	(A) ⇒ 第 O _i 号设备寄存器	留	留	留	留	留	01	6

9 计算指令

	指令形式	说明	A 留否	C = 1 条件	N = 1 条件	Z = 1 条件	V = 1 条件	微入 程 序 口	节 拍 数
68	1111 RRRR	(R) + 1 ⇒ R R = 0	留	留	(R) ₇ = 1	(R) ≡ 0	(R) ₇ ⊕ (R) ₆ = 1	42	15
69	1111 0000	[(R ₀)] + 1 ⇒ (R ₀)	留		[(R ₀)] ₇ = 1	[(R ₀)] ≡ 0	[(R ₀)] ₇ ⊕ [(R ₀)] ₆ = 1	44	15
70	0101 1001	(R ₀) + 1 ⇒ R ₀	留	//	(R ₀) ₇ = 1	(R) ≡ 0	(R ₀) ₇ ⊕ (R ₀) ₆ = 1	59	12
71	0101 1100	(R ₂ R ₁) + 1 ⇒ R ₂ , R ₁	留	//	(R ₂) ₇ = 1	(R ₂ R ₁) ≡ 0	(R ₂) ₇ ⊕ (R ₂) ₆ = 1	52	16

10 比较指令

	指令形式	说明	A 留否	C = 1 条件	N = 1 条件	Z = 1 条件	V = 1 条件	微入 程 序 口	节 拍 数
72	0000 1111 M ₁M _n M ₁M _n	(A) ≡ [(R ₂)(R ₁)] 否 若全同 M ₁ ~ M _n ⇒ PC 若不全同 (R ₂ R ₁) + 1 ⇒ R ₂ R ₁ (PC) + 1 ⇒ PC	留	留	全同时 留 不同时 (R) ₇ = 1	全同时 留 不同时 (R ₂ R ₁) ≡ 0	全同时 留 不同时 (R ₀) ⊕ (R ₀) = 1	05	全同 20 不同 28

11 中断返回指令

	指令形式	说明	A 留否	C = 1 条件	N = 1 条件	Z = 1 条件	V = 1 条件	微入 程 序 口	节 拍 数
73	0000 0000 1100 0111	将进入中断处理时, 保留于 STACK 中的 SW, A ₅ , R ₂ R ₁ , E, R ₀ , PCH, PCL 取回		δ = 1	A ₇ = 1	A ≡ 0	A ₇ ⊕ A ₆ = 1	00	22

指令形式	说 明	A 留否	C = 1 条件	N = 1 条件	Z = 1 条件	V = 1 条件	微入 程 序 口	节 拍 数
0000 0000 1110 0010 I ₇I ₀ M ₁₅M ₈ M ₇M ₀	从第 1 号设备寄存器连续送 (00000000 - M ₇M ₀) 个代码到存贮器的 M ₁₅M ₈ M ₇M ₀ → M ₁₅M ₈ 11111111 单元中 (发一次中断, 传送一次)	破	留	破	破	破	00	
0000 0000 1100 1111 O ₇O ₀ M ₁₅M ₈ M ₇M ₀	将存贮器的 M ₁₅M ₈ , M ₇M ₀ ~ M ₁₅M ₇ 11111111 单元中的 (10000000 - M ₃M ₀) 个代码连续的送到第 0 号设备存贮器中 (发一次中断传送一次)							

13 扩 充 指 令

指令形式	说 明	A 留否	C = 1 条件	N = 1 条件	Z = 1 条件	V = 1 条件	微入 程 序 口	节 拍 数
0011 0000 xxxx xxxx dddd dddd	将扩充指令的广义操作码 xxxxxx 送入 CC7~0, 同时 Z/CL 8xxxxxxx 即为第二 个微程序只读存贮器入口地址将指示信 息 dddd dddd 取出放在 RW 中	留	留	留	留	留	30	9

息, 只有定时的意义。

准静态逻辑系统有许多优点:

- (1) 对于同样性能的器件, 其系统速度可高出一倍;
- (2) 对器件的要求较低, 抗干扰能力强, 有利于系统可靠性;
- (3) 能实现完全的静态校验, 使系统很易调整和维修;
- (4) 与时序逻辑理论结合, 使逻辑设计过程完全算法化, 所有部件的工作过程为该部件的输入方程所描述, 从而整个系统可用有限个方程来作完全的描述。

4. 中 断

考虑到面向实时过程的可能性, 050 系统设计了较强的中断控制器, 这个控制器的功能如下: 当新来的中断源比正在执行的中断处理程序更高级, 且本级不屏蔽, 又没有更高级的中断源时, CPU 才响应新来的中断。这样避免

了非高级中断源时 CPU 运行作无谓的打扰, 提高了系统的实际效率。

为中断程序设置了硬件程序堆栈。每重中断要保护的讯息包括 PC、SW、A、E、R₃、R₁、R₂ 点堆栈中 4 个单元。32 单元的堆栈足以容纳 4 重中断加 16 重转子。

进栈退栈以及形成各级处理程序入口都由微程序实现, 整个中断响应速度很快。

各级中断都分别有自己的屏蔽位, 屏蔽位可以由程序任意修改, 这样中断排队次序, 通过程序可以任意安排。

5. 输入输出通道标准接口

BC 是输入输出通道标准接口部件, 由于 MOS 器件对长线的驱动能力很差, 因此 BC 具有异步控制的能力, BC 中有三个标志触发器 G、a、b:

- G: 总线占用触发器;
- a: 异步时序状态触发器;

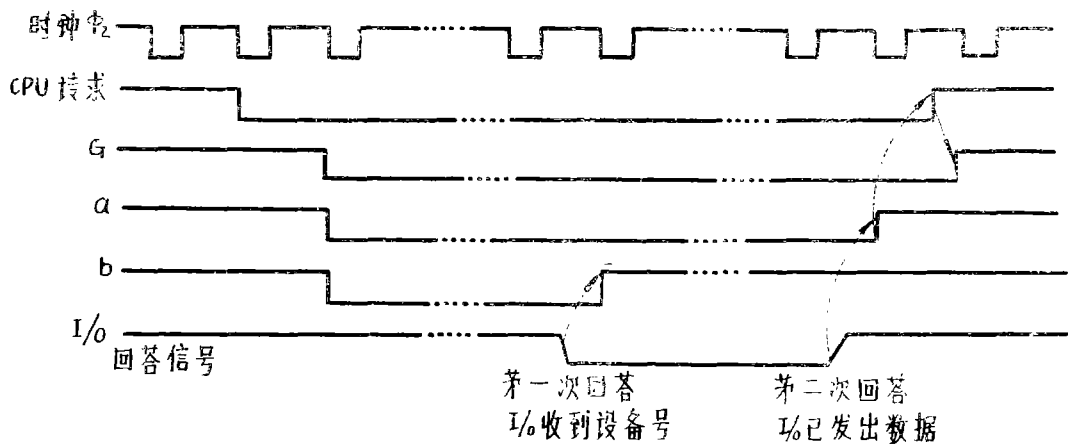


图2 输出过程

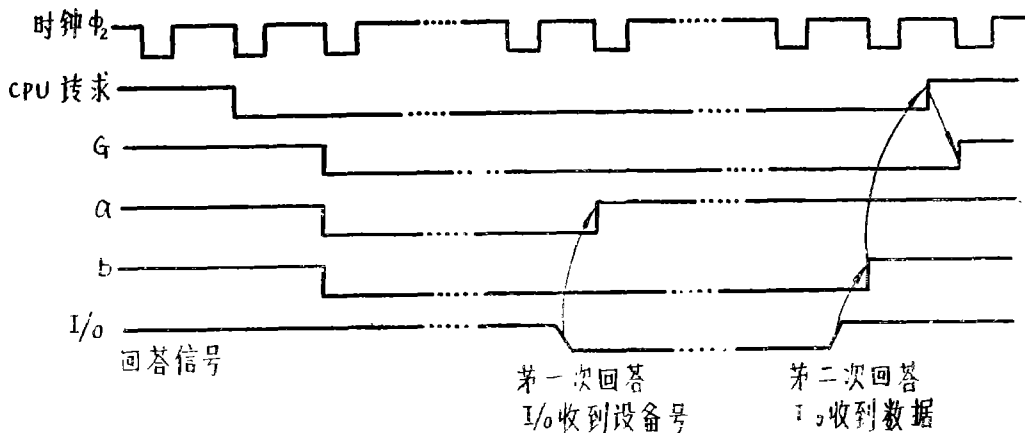


图3 输入过程

b: 异步时序状态触发器;

tab: 送设备号

tāb: 数据从 cpu→I/O (输出)

ta \bar{b} : 数据从 I/O→cpu (输入)

传输过程见图 2、3。

6. 数据传送快速通道

指令中有数据传送快速通道指令,用以与快速外存(如磁盘、磁带等)连接。但因目前还没有这类设备,故快速通道控制器没有安装。

五、050微型计算机

指令执行过程

050微型计算机指令功能的实现,是由若干条有机联系的微指令来完成的。

微指令存放在 256 单元×32位的只读存储器里面。本机共有 256 条微指令,每条微指令的字长32位。微指令经过操作码译码器进行译码,产生微操作,这些微操作,使机器的各个部件协调动作。微操作见下表。

图中 A₀+1 应为 A₀+1

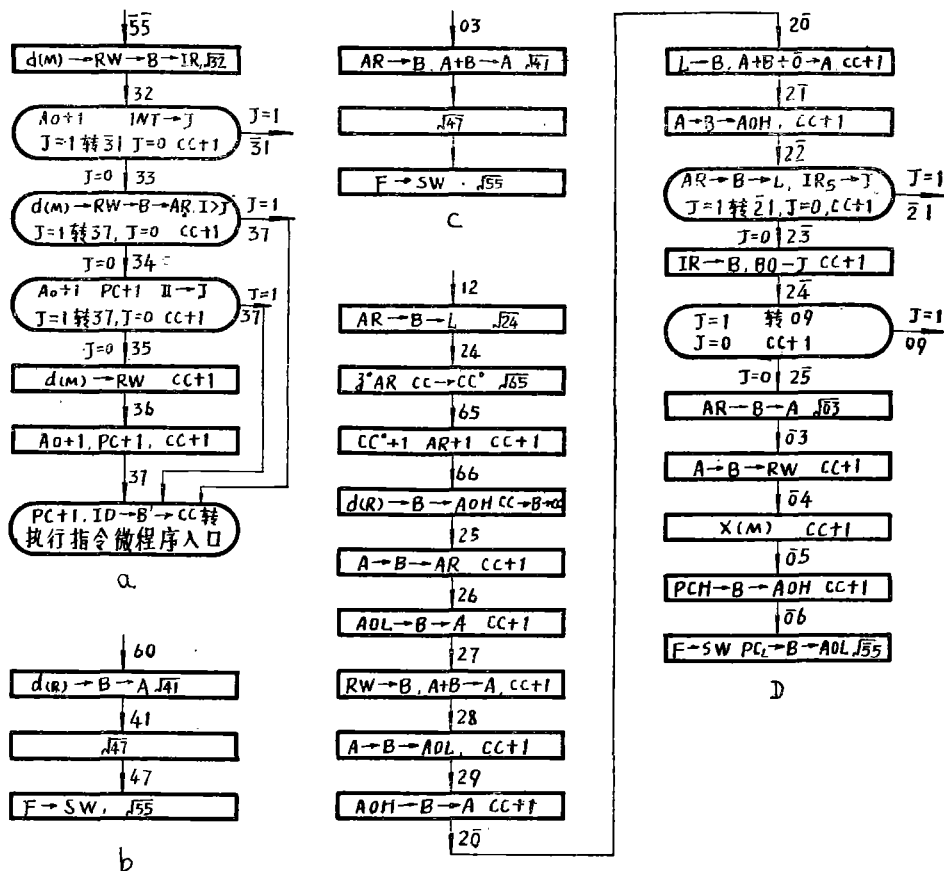


图 4 a. 取指流程 b. (R)→A流程 c. (A)+D→A流程 d. (A)-M+(R₂R₁)流程

完成一条指令的功能由取指令流程和执行本指令功能流程共同完成的。取指令流程见图(四 a) 执行本指令功能流程举(R)→A, (A)+D→A, A→M+(R₂R₁) 三条指令来说明, 流程见图(四、b、c、d)。

六、050微型计算机的器件

1. 器件分片情况

根据当时的工艺条件, 器件划分上采取大、中、小并存, 大、中、小规模集成电路为主的办法。

其中2048位(256×8)的只读存储器, 128位(16×8)静态存储器, 属于大规模集成电路。四位运算器, 八位通用寄存器, 指令编码

器等属于中规模集成电路, 其它属于小规模集成电路。

050微型计算机片子的各种型号、管子数、管芯面积见表

2. 线路设计

触发器采用了以动态移位寄存器为主体的准静态触发器, 它从外特性上来看, 与静态触发器功能一样, 这样方便机器的调整使用。触发器的单元线路工作波形见图(五), 当机器工作于停机状态时, ϕ_1 不产生, 由 T_1 、 T_2 、 T_3 构成一条反馈自保线路。当机器处于工作状态而触发器输入控制信号为“0”时, 由 T_4 、 T_5 构成一条反馈自保线路。二相时钟触发器, 在输入信号有毛刺时, 亦不会产生误动作, 因而工作可靠, 调整方便。

电 路 名 称	编 号	管 子 数	管 芯 面 积 (mm) ²
八位通用寄存器	501	429	6.2
	502		
四位 D 触发器	505	154	3.0
标准译码器	506	221	4.0
四位运算器	507	450	6.1
转移控制部件	508	94	2.7
等待控制部件	509	54	1.8
八位双向驱动器	511	134	3.0
128 位静态存储器	519	1007	6.3
指令编码器	520	284	3.6
2048位只读存储器	521	2500	6.6
	—524		

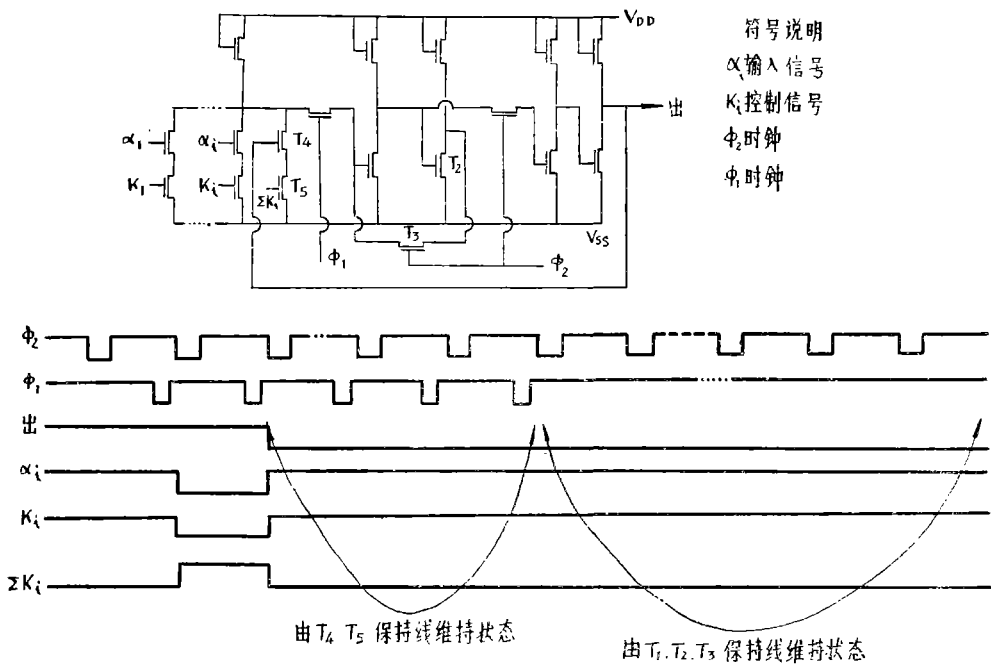


图 5 准静态触发器线路及波形