微程序控制器实验

1. **实验类型**

本实验类型为原理型+分析型+设计型

**二、实验目的**

⑴掌握微程序控制器的原理

⑵掌握TEC-8模型计算机中微程序控制器的实现方法，尤其是微地址转移逻辑的实现方法。

⑶理解条件转移对计算机的重要性。

⑷掌握时序产生器的组成原理。

**三、实验设备**

TEC-8实验箱 一台

直流万用表 一个

逻辑笔（在TEC-8实验箱上） 一支

## 四、实验电路

微程序控制器与硬连线控制器相比，由于其具有规整性、易于设计，以及需要的时序发生器相对简单，在20世纪七、八十年代得到广泛应用。在计算机系统中，微程序设计技术是利用软件方法来设计硬件的一门技术。本实验通过一个具体微程序控制器工作过程的跟踪，理解控制器在计算机中的作用，使学生从实践上掌握微程序控制器的一般实现方法。

## 微程序控制的基本思想，就是仿照通常的解题程序的方法，把操作控制信号编成所谓的“微指令”，存放到一个只读存储器里，当机器运行时，一条又一条地读出这些微指令，从而产生全机所需要的各种操作控制信号，使相应部件执行所规定的操作。

## 微命令和微操作

## 一台数字计算机基本上可以划分为两大部分—控制部件和执行部件。控制器就是控制部件，而运算器、存储器、外围设备相对控制器来讲，就是执行部件。那么两者之间是怎样进行联系的呢？

## 控制部件与执行部件的一种联系，就是通过控制线。控制部件通过控制线向执行部件发出各种控制命令，通常把这种控制命令叫做微命令，而执行部件接受微命令后所进行的操作，叫做微操作。

## 控制部件与执行部件之间的另一种联系，就是通过反馈线。执行部件通过反馈线向控制部件反映操作情况，以便使控制部件根据执行部件的“状态”来下达新的微命令，这也叫做“状态测试”。

## 微操作在执行部件中是最基本的操作。由于数据通路的结构关系，微操作可分为相容性和相斥性两种。所谓相容性的微操作，是指在同时或同一个CPU周期内可以并行执行的微操作。所谓相斥性的微操作，是指不能在同时或不能在同一个CPU周期内并行执行的微操作。

## 图5.1简单运算器数据通路图

## 如图5.1所示，给出了一个简单运算器模型，其中ALU为算术逻辑单元，R1、R2、R3为三个寄存器。三个寄存器的内容都可以通过多路开关从ALU的X输入端或Y输入端送至ALU，而ALU的输出可以送往任何一个寄存器或同时送往R1，R2，R3三个寄存器。

## 在我们给定的数据通路中，多路开关的每个控制门仅是一个常闭的开关，它的一个输入端代表来自寄存器的信息，而另一个输入端则作为操作控制端。一但两个输入端都有输入信号时，它才产生一个输出信号，从而在控制线能起作用的一个时间宽度中来控制信息在部件中流动。

## 图5.1中每个开关门由控制器中相应的微命令来控制，例如，开关门4由控制器中编号为4的微命令控制，开关门6由编号为6的微命令控制，如此等等。三个寄存器R1、R2、R3的时钟输入端1、2、3也需要加以控制，以便在ALU运算完毕而输出公共总线上电平稳定时，将结果打入到某一寄存器。

## 另外，我们假定ALU只有“+”、“-”、“M”（传送）三个微操作。信号CY为最高进位触发器，有进位时该触发器状态为“1”。

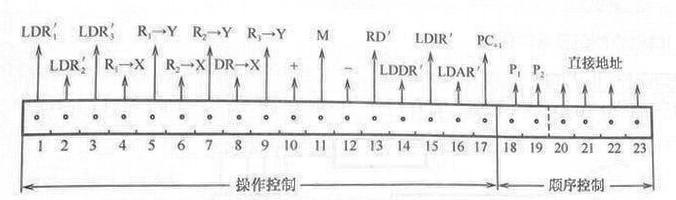
## ALU的操作（“加”、“减”、“传送”）在同一个CPU周期中只能选择一种，不能并行，所以“+“、“-”、“M”（传送）三个微操作是相斥性的微操作；类似地，4、6、8三个微操作是相斥性的；5、7、9三个微操作也是相斥性的。

## 微操作1、2、3是可以同时进行的，所以是相容性的微操作。另外，ALU的X输入微操作4、6、8分别与Y输入的5、7、9任意两个微操作，也都是相容性的。

## 微指令和微程序

## 在机器的一个CPU周期中，一组实现一定操作功能的微命令的组合，构成一条微指令。

## 图5.2表示一个具体的微指令结构，微指令字长为23位，它由操作控制和顺序控制两组构成。



## 图5.2 微指令基本格式

## 操作控制部分用来发出管理和指挥全机工作的控制信号。为了形象直观，在我们的例子中，该字段为17位，每一位表示一个微命令。每个微命令的编号同图5.1所示的数据通路相对应，具体功能显示于微指令格式的左上部。当操作控制字段某一位信息为“1”时，表示发出微命令；而某一位信息为“0”时，表示不发出微命令。

## 例如，当微指令字第1位信息为“1”时，表示发出信号LDR1´的微命令，那么运算器将执行ALU→R1的微操作，把公共总线上的信息打入到寄存器R1。同样，当微指令第10位信息为“1”时，则表示向ALU发出进行“+”的微命令，因而ALU就执行“+”的微操作。

## 注意：图5.2中微指令给出的控制信号都是节拍电位信号，它们的持续时间都是一个CPU周期。如果要用来控制图5.1所示的运算器数据通路，势必会出现问题，因为前面的三个微命令信号（LDR1´、LDR2´、LDR3´）既不能来得太早，也不能来得太晚。为此，要求这些微命令信号还要加入时间控制.

## 例如同节拍脉冲T4相与，而得到LDR3- LDR1信号，如图5.3右图所示。在这种情况下，控制器最后发给运算器的12个控制信号中，3个是节拍脉冲信号（LDR1、LDR2、LDR3），其他9个都是节拍电位信号，从而保证运算器在前600ns时间内进行运算。600ns后运算完毕，公共总线上输出稳定的运算结果，由LDR1（或LDR2、LDR3）信号打入到相应的寄存器，其时间关系如图5.3所示。

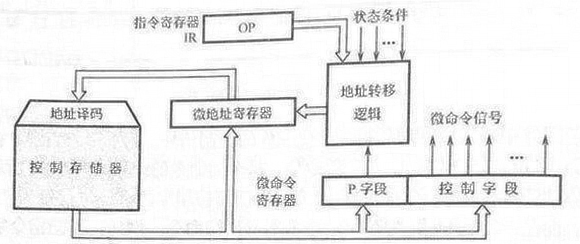
## 图5.3 运算器操作时序与产生逻辑

## 微指令格式中的顺序控制部分用来决定产生下一条微指令的地址。下面我们将会知道，一条机器指令的功能是用许多条微指令组成的序列来实现的，这个微指令序列通常叫做微程序。既然微程序是由微指令组成的，那么当执行当前一条微指令时，必须指出后继微指令的地址，以便当前一条微指令执行完毕后，取出下一条微指令。

## 决定后继微指令地址的方法不只一种。在我们所举的例子中，由微指令顺序控制字段的6位信息来决定。其中4位（20-23）用来直接给出下一条微指令的地址。第18,19两位作为判别测试标志，当此两位为“0”时，表示不进行测试，直接按顺序控制字段第23-20位给出的地址取下一条微指令：当第18位或第19位为“1”时，表示要进行P1或P2的判别测试，根据测试结果，需要对第20-23位的某一位或几位进行修改，然后按修改后的地址取下一条微指令。

## 微程序控制器的组成

## 微程序控制器原理框图如图5.4所示，它主要由控制存储器、微指令寄存器和地址转移逻辑三大部分组成，其中微指令寄存器分为微地址寄存器和微命令寄存器两部分。



## 图5.4微程序控制器组成原理框图

## 控制存储器

## 控制存储器用来存放实现全部指令系统的微程序，它是一种只读型存储器，一旦微程序固化，机器运行时则只读不写。其工作过程是：每读出一条微指令，则执行这条微指令；接着又读出下一条微指令，又执行这一条微指令……读出一条微指令并执行微指令的时间总和称为一个微指令周期。

## 通常，在串行方式的微程序控制器中，微指令周期就是只读存储器的工作周期。控制存储器的字长就是微指令字的长度，其存储容量视机器指令系统而定，即取决于微程序的数量。

## 对控制存储器的要求是速度快，读出周期要短。

## 微指令寄存器

## 微指令寄存器IR用来存放由控制存储器读出的一条微指令信息。其中微地址寄存器uAR决定将要访问的下一条微指令的地址，而微命令寄存器则保存一条微指令的操作控制字段和判别测试字段的信息。

## 地址转移逻辑

## 在一般情况下，微指令由控制存储器读出后直接给出下一条微指令的地址，通常我们简称微地址，这个微地址信息就存放在微地址寄存器中。如果微程序不出现分支，那么下一条微指令的地址就直接由微地址寄存器给出。当微程序出现分支时，意味着微程序出现条件转移。在这种情况下，通过判别测试字段P和执行部件的“状态条件”反馈信息，去修改微地址寄存器的内容，并按改好的内容去读下一条微指令。地址转移逻辑就承担自动完成修改微地址的任务。

## 一条机器指令是由若干条微指令组成的序列来实现的。因此，一条机器指令对应着一个微程序，而微程序的总和便可实现整个的指令系统。

## CPU周期与微指令周期的关系

## 在串行方式的微程序控制器中，微指令周期等于读出微指令的时间加上执行该条微指令的时间。为了保证整个机器控制信号的同步，可以将一个微指令周期时间设计得恰好和CPU周期时间相等，图5.5示出了某计算机中CPU周期与微指令周期的时间关系。

## 图5.5 CPU周期与微指令周期的关系

## 一个CPU周期为0.88μs，它包含四个等间隔的节拍脉冲T1-T4，每个脉冲宽度为200ns，用节拍脉冲T4作为读取微指令的时间，用节拍脉冲T1+T2+T3时间作为执行微指令的时间。

## 例如，在前600ns时间内运算器进行运算，在600ns时间的末尾运算器已经运算完毕，可用节拍脉冲T4上升沿将运算结果打入某个寄存器。与此同时可用节拍脉冲T4时间间隔读取下条微指令，经200ns时间延迟，下条微指令又从只读存储器读出，并用节拍脉冲T1上升沿打入到微指令寄存器。如忽略触发器的翻转延迟，那么下条微指令的微命令信号就从节拍脉冲T1上升沿起就开始有效，直到下一条微指令读出后打入微指令寄存器为止。因此一条微指令的保持时间恰好是0.8μs，也就是一个CPU周期的时间。

## 机器指令与微指令的关系

## 经过上面的讲述，应该说，我们能够透彻地了解机器指令与微指令的关系。也许读者会问：一会儿取机器指令，一会儿取微指令，它们之间到底是什么关系？

## 现在让我们把前面内容归纳一下，作为对此问题的问答。

## 第一，一条机器指令对应一个微程序，这个微程序是由若干条微指令序列组成的。因此，一条机器指令的功能是由若干条微指令组成的序列来实现的。简言之，一条机器指令所完成的操作划分成若干条微指令来完成，由微指令进行解释和执行。

## 第二，从指令与微指令，程序与微程序，地址与微地址的应关系来看，前者与内存储器有关，后者与控制存储器有关，与此相关，也有相对应的硬设备，如图5. 6所示。

## 

## 图5.6 机器指令与微指令的关系

## 第三，我们在讲述本章节时，曾讲述了指令与机器周期概念，并归纳了五条典型指令的指令周期（参见图5.7）。现在我们看到，图5.7就是这五条指令的微程序流程图，每一个CPU周期就对应一条微指令。这就告诉我们如何设计微程序，也将使我们进一步体验到机器指令与微指令的关系。

## 图5.7 用方程框图语言表示指令周期

## 微程序设计技术

## 已经了解了微程序控制器的基本原理，这使我们认识到，如何确定微指令的结构，乃是微程序设计的关键。

## 设计微指令结构应当追求的目标是：①有利于缩短微指令字长度：②有利于减小控制存储器的容量；③有利于提高微程序的执行速度；④有利于对微指令的修改；⑤有利于提高微程序设计的灵活性。

## 1）.微命令编码

## 微命令编码，就是对微指令中的操作控制字段采用的表示方法。通常有以下三种方法。

## 直接表示法

## 采用直接表示法的微指令结构如图5.2所示，其特点是操作控制字段中的每一位代表一个微命令，这种方法的优点是简单直观，其输出直接用于控制。缺点是微指令字较长，因而使控制存储器容量较大。

## 编码表示法

## 编码表示法是把一组相斥性的微命令信号组成一个小组（即一个字段），然后通过小组（字段）译码器对每一个微命令信号进行译码，译码输出作为操作控制信号，其微指令结构如图5.8所示。

## 图5.8 字段直接译码法

## 采用字段译码的编码方法，可以用较小的二进制信息位表示较多的微命令信号，例如3位二进制位译码后可表示7个微命令，4位二进制位译码后可表示15个微命令。与直接控制法相比，字段译码控制法可使微指令字大大缩短，但由于增加译码电路，使微程序的执行速度稍稍减慢。目前在微程序控制器设计中，字段直接译码法使用较普遍。

## 混合表示法

## 这种方法是把直接表示法与字段编码法混合使用，以便能综合考虑微指令字长、灵活性、执行微程序速度等方面的要求。

## 另外，在微指令中还可附设一个常数字段。该常数可作为操作数送入ALU运算，也可作为计数器初值用来控制微程序循环次数。

## 2）.微地址的形成方法

## 微指令执行的顺序控制问题，实际上是如何确定下一条微指令的地址问题。通常产生后继微地址有两种方法。

## 计数器方式

## 这种方法同用程序器计数来产生机器指令地址的方法相类似。在顺序执行微指令时，后继微地址由现行微地址加上一个增量来产生；在非顺序执行微指令时，必须通过转移方式，使现行微指令执行后，转去执行指定后继微地址的下一条微指令。在这种方法中，微地址寄存器通常改为计数器。为此，顺序执行的微指令序列就必须安排在控制存储器的连续单元中。

## 计数器方式的基本特点是：微指令的顺序控制字段较短，微地址产生机构简单。但是多路并行转移功能较弱，速度较慢，灵活性较差

## 多路转移方式

## 一条微指令具有多个转移分支的能力称为多路转移。例如，“取指”是指微指令根据操作码OP产生多路微程序分支而形成多个微地址的过程。

## 在多路转移方式中，当微程序不产生分支时，后继微地址直接由微指令的顺序控制字段给出；当微程序出现分支时，有若干“后选”微地址可供选择：即按顺序控制字段的“判别测试”标志和“状态条件”信息来选择其中一个微地址。

## 其原理如图5.3所示，“状态条件”有1位标志，可实现微程序两路转移，涉及微地址寄存器的一位；“状态条件”有2位标志，可实现微程序4路转移，涉及微地址寄存器的两位。依此类推，“状态条件”有n位标志，可实现微程序2n路转移，涉及微地址寄存器的n位。因此执行转移微指令时，根据状态条件可转移到2n个微地址中的一个。

## 多路转移方式的特点是，能以较短的顺序控制字段配合，实现多路并行转移，灵活性好，速度较快，但转移地址逻辑需要用组合逻辑方法设计。

## 3）.微指令格式

## 微指令的编译方法是决定微指令格式的主要因素。考虑到速度、成本等原因，在设计计算机时采用不同的编译法。因此微指令的格式大体分成两类：水平型微指令和垂直型微指令。

## 水平型微指令

## 一次能定义并执行多个并行操作微命令的微指令，叫做水平型微指令。例如本节中所讲的微指令即为水平型微指令。

## 水平型微指令的一般格式如下：

|  |  |  |
| --- | --- | --- |
| 控制字段 | 判别测试字段 | 下地址字段 |

## 按照控制字段的编码方法不同，水平型微指令又分为三种：一种是全水平型（不译码法）微指令，第二种是字段译码法水平型微指令，第三种是直接和译码相混合的水平型微指令。

## 垂直型微指令

## 微指令中设置微操作码字段，采用微操作码编译法，由微操作码规定微指令的功能，称为垂直型微指令。

## 垂直型微指令的结构类似于机器指令的结构。它有操作码，在一条微指令中只有1-2个微操作命令，每条微指令的功能简单，因此，实现一条机器指令的微程序要比水平型微指令编写的微程序长得多。它是采用较长的微程序结构去换取较短的微指令结构。

## 下面用4条垂直型微指令的微指令格式加以说明，设微指令字长为16位，微操作为3位。

## 寄存器-寄存器传送型微指令

## 15 13 12 8 7 3 2 0

|  |  |  |  |
| --- | --- | --- | --- |
| 000 | 源寄存器编址 | 目的寄存器编址 | 其他 |

## 其功能是把源寄存器数据送目标寄存器，15-13位为微操作码，源寄存器和目标寄存器编址各5位，可指定31个寄存器（25）。

## 【2】运算控制型微指令

## 15 13 12 8 7 3 2 0

|  |  |  |  |
| --- | --- | --- | --- |
| 001 | 左输入源编址 | 右输入源编址 | ALU |

## 其功能是选择ALU的左、右两个端口输入源信息，按ALU字段所指定的运算功能（8种操作）进行处理，并将结果送入暂存器中。ALU左、右端口输入源编址可指定31种信息源之一。

## 【3】访问主存微指令

## 15 13 12 8 7 3 2 1 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 010 | 寄存器编址 | 存储器编址 | 读写 | 其他 |

## 其功能是将主存中一个单元的信息送入寄存器或者将寄存器的数据送往主存。存储器编址是指按规定的寻址方式进行编址。第1、2位指定读操作或写操作（取其之一）

## 【4】条件转移微指令

## 15 13 12 4 3 0

|  |  |  |
| --- | --- | --- |
| 011 | D | 目的寄存器编址 |

## 其功能是根据测试对象的状态决定是转移到D所指定的微地址单元，还是顺序执行下一条微指令。9位D字段不足以表示一个完整的微地址，但可以用来替代现行μPC的低位地址。测试条件字段有4位，可规定16种测试条件。

## （3）水平型微指令与垂直型微指令的比较

## 【1】 水平型微指令并行操作能力强，效率高，灵活性强，垂直型微指令则较差。

## 在一条水平型微指令中，设置有控制信息传送通路（门）以及进行所有操作的微命令，因此在进行微程序设计时，可以同时定义比较多的并行操作的微命令，来控制尽可能多的并行信息传送，从而使水平型微指令具有效率高及灵活性强的优点。

## 在一条垂直型微指令中，一般只能完成一个操作，控制一两个信息传送通路，因此微指令的并行操作能力低，效率低。

## 【2】水平型微指令执行一条指令的时间短，垂直型微指令执行时间长。

## 因为水平型微指令的并行操作能力强，因此与垂直型微指令相比，可以用较少的微指令数来实现一条指令的功能，从而缩短了指令的执行时间。而且当执行一条微指令时，水平型微指令的微命令一般直接控制对象，而垂直型微指令要经过译码，会影响速度

## 【3】由水平型微指令解释指令的微程序，有微指令字较长而微程序短的特点。垂直型微指令则相反，微指令字较短而微程序长。

## 【4】水平型微指令用户难以掌握，而垂直型微指令与指令比较相似，相对来说，比较容易掌握。

## 水平型微指令与机器指令差别很大，一般需要对机器的结构、数据通路、时序系统以及微命令很精通才能设计。

## 垂直微指令的设计思想在 Pentium4、安腾系列机中得到了应用。

## （4）动态微程序设计

## 微程序设计技术还有静态微程序设计和动态微程序设计之分。对应于一台计算机的机器指令只有一组微程序，而且这一组微程序设计好之后，一般无须改变而且也不好改变，这种微程序设计技术称为静态微程序设计，本节前面讲述的内容基本上属于静态微程序设计的概念。

## 当采用 E2PROM作为控制存储器时，还可以通过改变微指令和微程序来改变机器的指令系统，这种微程序设计技术称为动态微程序设计。采用动态微程序设计时，微指令和微程序可以根据需要加以改变，因而可在一台机器上实现不同类型的指令系统。

## TEC-8模型机微程序控制器介绍

## 1.微指令格式

根据机器指令功能、格式和数据通路所需的控制信号，TEC-8采用如图5.9所示的微指令格式。微指令字长40位，其中，顺序字段11位(判别字段P4-P0，后继微地址NµA5-NµA0)，控制字段29位，由微命令直接控制。

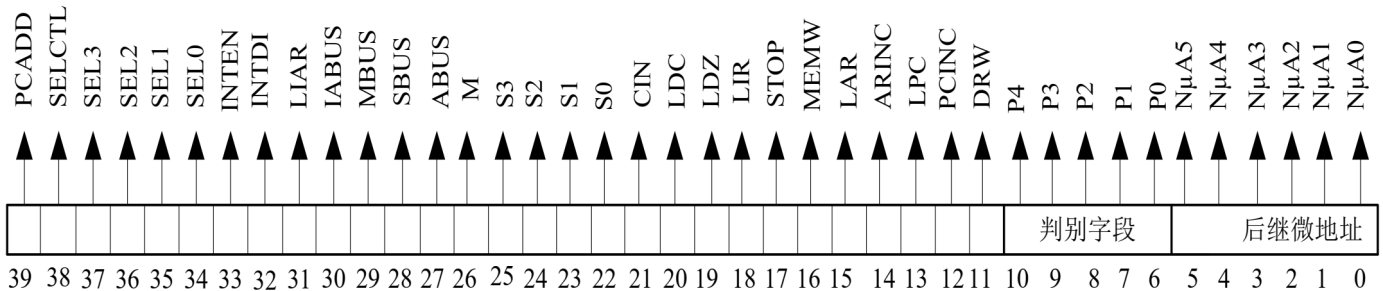


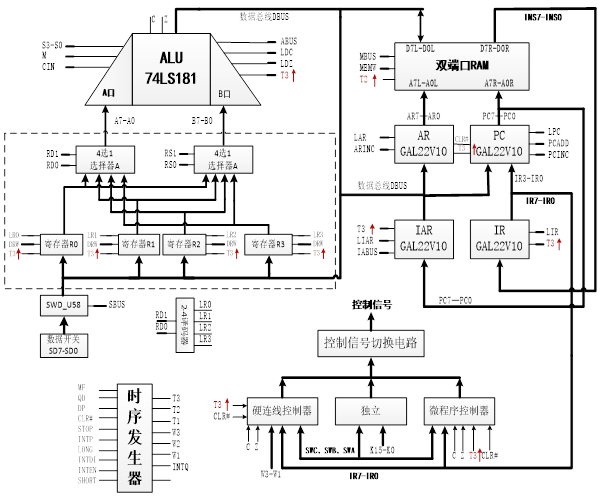
图5.9 微指令格式

|  |  |
| --- | --- |
| NµA5-NµA0 | 后继微地址，在微指令顺序执行的情况下，它是下一条微指令的地址。 |
| P0 | =1时，根据后继微地址 NµA5-NµA0 和模式开关 SWC、SWB、SWA 确定下一条微指令的地址。见图 5.2 微程序流程图。 |
| P1 | =1 时，根据后继微地址 NµA5-NµA0 和指令操作码 IR7-IR4 确定下一条微指令的地址。见图 5.2微程序流程图。 |
| P2 | =1 时，根据后继微地址 NµA5-NµA0 和进位 C 确定下一条微指令的地址。见图5.2 微程序流程图。 |
| P3 | =1 时，根据后继微地址 NµA5-NµA0 和结果为 0 标志 Z 确定下一条微指令的地址。见图 5.2 微程序流程图。 |
| P4 | =1 时，根据后继微地址NµA5-NµA0和中断信号INT确定下一条微指令的地址。在TEC-8模型计算机中，中断信号INT由时序发生器在接到中断请求信号后产生。 |
| STOP | =1 时，在节拍脉冲T3 结束后时序发生器停止输出节拍脉冲 T1、T2、T3。 |
| LIAR | =1 时，在节拍脉冲T3的上升沿，将程序计数器PC7-PC0 写入中断地址寄存器 IAR。 |
| INTDI | =1 时，置允许中断标志(在时序发生器中)为“0”，禁止 TEC-8 模型计算机响应中断请求。 |
| INTEN | =1 时，置允许中断标志(在时序发生器中)为“1”，允许 TEC-8 模型计算机响应中断请求。 |
| IABUS | =1 时，将中断地址寄存器中的地址送数据总线DBUS。 |
| PCADD | =1 时，将当前的 PC 值加上相对转移量，生成新的PC。 |
| SETCTL | =1 时，实验系统处于实验台状态。 |
| =0 时，实验系统处于运行程序状态。 |

前面的4个实验已经介绍了主要的微命令(控制信号)，介绍过的微命令不再重述，这里介绍后继微地址、判别字段和其它的微命令。

**2. 微程序流程图**

根据TEC-8模型计算机的指令系统和控制台功能（见表2.2）以及各部件之间的数据通路（见图5.10），TEC-8模型计算机的微程序流程图如图5.11所示。在图5.11中每一个方框代表1条微指令其现行微地址在方框右上角，（默认）后继微地址在方框右下角。

图5.10 TEC-8 模型计算机框图

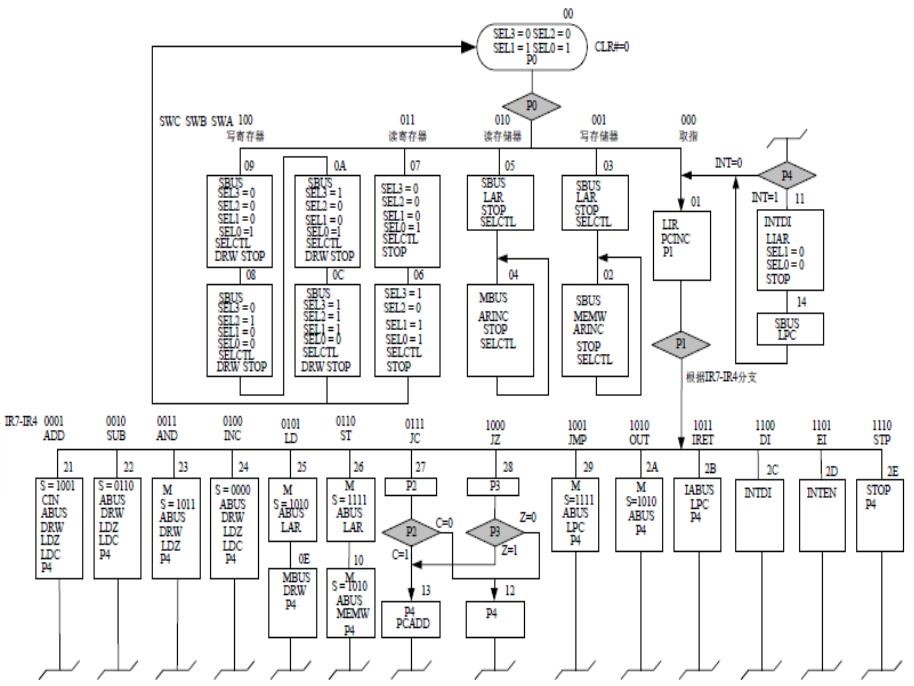


图5.11 TEC-8 模型计算机微程序流程图

由于TEC-8模型计算机在非“独立”控制方式（在此方式下，模型及各部分之间相互独立）有微程序控制器和硬连线控制2个控制器，因此微程序控制器产生的控制信号以前缀“A-”标示，以便和硬连线控制器产生的控制信号区分。硬连线控制器产生的控制信号以前缀“B-”标示。图5.11中，为了简洁，将许多以“A-”为前缀的信号，省略了前缀。

需要说明的是，图5.11中没有包括运算器组成实验、双端口存储器实验和数据通路三部分。这三部分的微程序很简单，微程序都是顺序执行的，根据这四个实验很容易画出。

## 3．微程序控制器电路

根据TEC-8模型计算机的指令系统、控制台功能、微指令格式和微程序流程图，TEC-8 模型计算机微程序控制器电路如图 5.12所示。

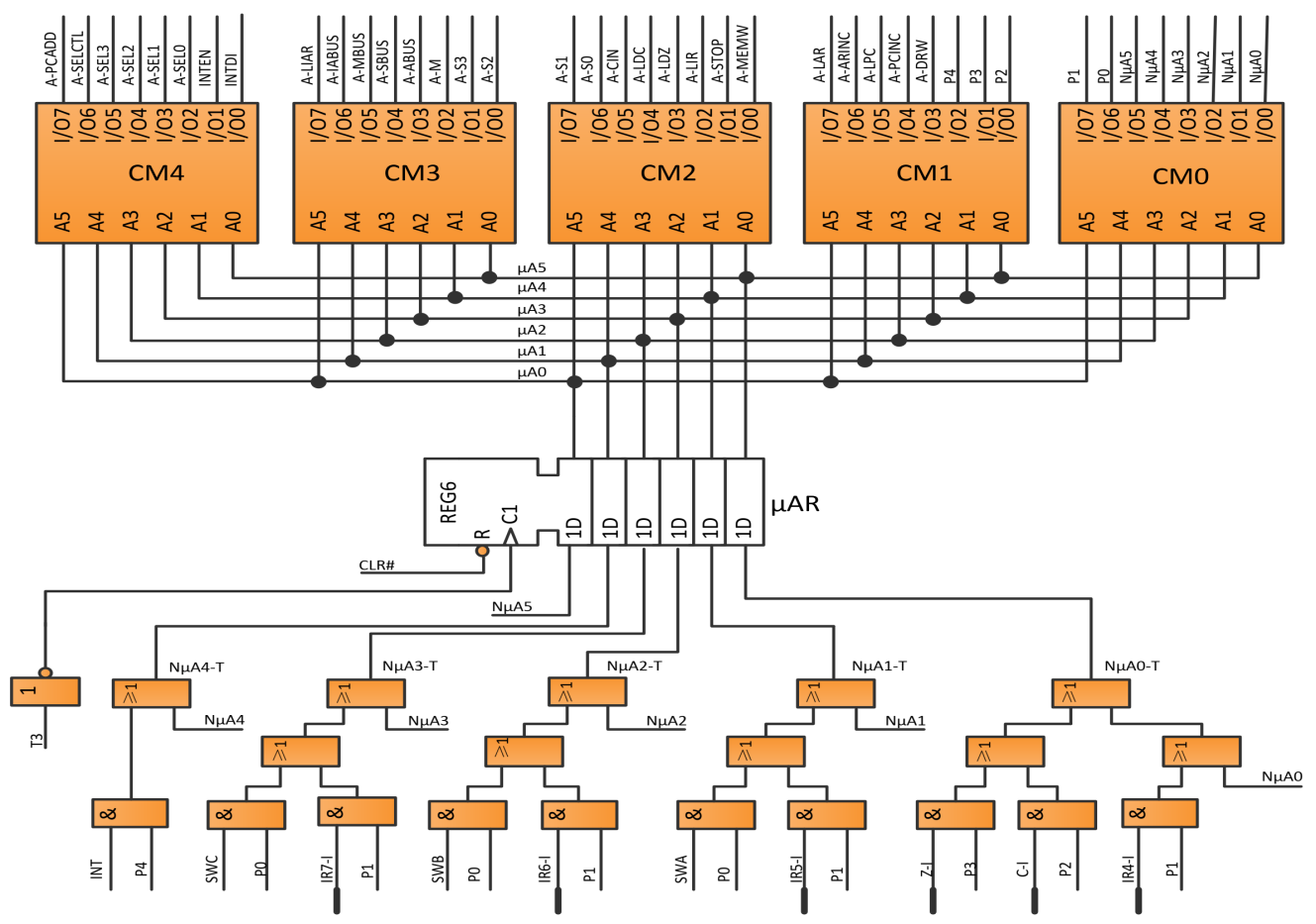


图5.12 微程序控制器电路图

图 5.12中，以短粗线标志的信号都有接线孔。信号IR4-I、IR5-I、IR6-I、IR7-I、C-I 和Z-I的实际意义分别等同于IR4、IR5、IR6、IR7、C和Z。INT信号是时序发生器接收到中断请求脉冲PULSE(高电平有效)后产生的中断信号。

1. 控制存储器

控制存储器由5片HN58C65（【39】、【40】、【41】、【42】、【43】）组成，在图5.12中表示为CM4-CM0。其中CM0存储微指令最低的8位微代码，CM4存储微指令最高的8位（实际使用7位）微代码。控制存储器的微代码必须与微指令格式一致。HN58C65是一种8K×8(bit)位的E2PROM器件，地址位为A12-A0。由于TEC-8模型计算机只使用其中64(B)个字节作为控制存储器，因此将地址位A12-A6接地，地址位A5-A0接微地址µA5-µA0。在正常工作方式下，5片E2PROM处于只读状态；在修改控制存储器内容时，5片E2PROM处于读、写状态。

1. 地址寄存器

微地址寄存器µAR由1片74LS174【22】组成，74LS174 是一个6D触发器，异步清零。当按下复位按钮CLR时，产生的信号CLR#(负脉冲)使微地址寄存器复位，指示灯µA5-µA0为00H，供读出第一条微指令使用。在一条微指令结束时，用节拍电位T3的下降沿将微地址转移逻辑产生的下条微指令地址NµA5、NµA4-NµA0写入微地址寄存器。

⑶微地址转移逻辑

微地址转移逻辑由若干与门（三个74LS08【23】、【33】、【61】）和或门（三个74LS32【24】、【25】、【32】）组成，实现“与-或”逻辑。深入理解微地址转移逻辑，对于理解计算机的本质有很重要的作用。计算机现在的功能很强大，但是它是建立在两个很重要的基础之上，一个是最基本的加法和减法功能， 令一个是条件转移功能。

设想一下，如果没有条件转移指令，实现10000（一万）个数相加至少需要20000（二万）条指令，还不如用算盘计算速度快。可是有了条件转移指令后，一万个数相加，不超过20条指令就能实现。

因此可以说，最基本的加法和减法功能以及条件转移功能给计算机后来的强大功能打下了基础。本实验中微地址转移逻辑的实现方法是一个很简单的例子，对于理解条件转移的实现方法大有益处。

显然，TEC-8采用多路转移方式形成微程序所需要的多路分支，其可能的分支转移点共有5处，分别用判断为P0-P4指示（图5.11），很容易地可以写出如图5.12所示的微程序控制电路的形成逻辑：

NµA5-T = NµA5 （NµA5接到微地址寄存器µAR的D5输入端）

NµA4-T = NµA4 or (P4 and INT)

NµA3-T = NµA3 or (P0 and SWC) or (P1 and IR7-I)

NµA2-T = NµA2 or (P0 and SWB) or (P1 and IR6-I)

NµA1-T = NµA1 or (P0 and SWA) or (P1 and IR5-I)

NµA0-T = NµA0 or (P1 and IR4) or（P2 and C）or（P3 and Z）

两级与或逻辑构成微地址转移逻辑，用于产生下一微指令地址。在每一个节拍电位T1的上升沿，新的微指令地址置入微地址寄存器，控存随即输出微指令的控制信号。

下面分析根据后继微地址NµA5-NµA0、判别位P1和指令操作码如何实现微程序分枝的。

微地址NµA5-NµA0中的微指令是一条功能为取指令的微指令。将取指微指令（如图5.11中微地址20H处所示）在T3的上升沿从双端口存储器中取出的指令写入指令寄存器IR。在这条微指令中，默认后继微地址为20H，判别位P1为“1”、其他判别位均为“0”。因此根据微地址转移逻辑， 很容易就知道，下一条微指令的微地址是：

NµA5-T = NµA5

NµA4-T = NµA4

NµA3-T = NµA3 or （P1 and IR7-I）

NµA2-T = NµA2 or （P1 and IR6-I）

NµA1-T = NµA1 or （P1 and IR5-I）

NµA0-T = NµA0 or （P1 and IR4-I）

新产生的微地址NµA5-T、（NµA4-T）-（NµA0-T）在T3的下降沿写入微地址寄存器µAR，实现了图5.12所要求的根据指令操作码进行微程序分支。

微地址转移逻辑中，SWC、SWB、SWA为控制台指令的定义开关，区分控制台指令对应的微程序流程。C-I为进位标志信号，Z-I为结果为0标志信号，IR7-IR4为机器指令操作码字段，区分不同机器指令对应的微程序流程。

（4）机器指令与微程序

TEC-8使用14条机器指令，见表5.1所示，均为单字长（8位）指令。指令的高4位IR7-IR4是操作码，提供给微程序控制器用作地址转移，低4位提供给数据通路。

14条指令的微程序流程图如图5.12所示。每条微指令可按前述的微指令格式转换成二进制代码，然后写入微程序控制器的控制存储器中。

表5.1 TEC-8模型计算机指令系统

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 助记符 | 功 能 | 指令格式 | | |
| IR(7-4) | IR(3-2) | IR(1-0) |
| 加法 | ADD Rd, Rs | Rd ← Rd + Rs | 0001 | Rd | Rs |
| 减法 | SUB Rd, Rs | Rd ← Rd -Rs | 0010 | Rd | Rs |
| 逻辑与 | AND Rd, Rs | Rd ← Rd andRs | 0011 | Rd | Rs |
| 加1 | INCRd | Rd ← Rd + 1 | 0100 | Rd | XX |
| 取数 | LD Rd, [Rs] | Rd ← [Rs] | 0101 | Rd | Rs |
| 存数 | ST Rs, [Rd] | Rs → [Rd] | 0110 | Rd | Rs |
| C条件转移 | JCaddr | C=1，则 PC←@ +offset | 0111 | offset | |
| Z条件转移 | JZaddr | Z=1，则 PC←@ +offset | 1000 | offset | |
| 无条件转移 | JMP[Rd] | PC ←Rd | 1001 | Rd | XX |
| 输出 | OUTRs | DBUS ←Rs | 1010 | XX | Rs |
| 中断返回 | IRET | 返回断点 | 1011 | XX | XX |
| 关中断 | DI | 禁止中断 | 1100 | XX | XX |
| 开中断 | EI | 允许中断 | 1101 | XX | XX |
| 停机 | STP | 暂停运行 | 1110 | XX | XX |

表5.1中，XX代表随意值。Rs代表源寄存器号，Rd代表目的寄存器号。在条件转移指令中，@代表当前PC的值+1，offset是一个4位的有符号数，第3位是符号位，“0”代表正数，“1”代表负数。**注意：@不是当前指令的PC值，是当前指令的PC值加 1。**

指令系统中，指令操作码0000B没有对应的指令，实际上指令操作码0000B对应着一条nop指令，即什么也不做的指令。当复位信号为0时，对指令寄存器IR复位，使IR的值为0000 0000B，对应一条nop指令。这样设计的目的是适应指令流水的初始状态要求。

## 五、实验任务

1．正确设置模式开关 SWC、SWB、SWA，用单微指令方式(单拍开关DP设置为“1”) 跟踪控制台操作读寄存器、写寄存器、读存储器、写存储器的执行过程，记录下每一步的微地址µA5-µA0、后继微地址NµA5-NµA0、判别位P4-P0和有关控制信号的值，写出这4种控制台操作的作用和使用方法。

2．正确设置指令操作码IR7-IR4，用单微指令方式跟踪表5.1中除停机指令STP之外的所有指令的执行过程。记录下每一步的微地址µA5-µA0、后继微地址NµA5-NµA0、判别位P4-P0和有关控制信号的值。对于JZ指令，跟踪Z=1、Z=0两种情况；对于JC指令， 跟踪C=1、C=0 两种情况。

## 第1组：操作模式的控制

**六、实验步骤**

**1.**操作模式的设置：

将控制器转换开关拨到“微程序”位置,微程序指示灯绿灯亮；

将编程开关设置为正常位置，编程指示灯红灯灭；

将短路子DZ11设置为左边的“正常”位置；

将单拍开关设置为“1”(朝上)。 在单拍开关DP为“1”时，每按一次QD按钮，只执行一条微指令、即DP=1，产生一组节拍电位T1、T2、T3信号。

将信号IR7-I、IR6-I、IR5-I、IR4-I、C-I、Z-I依次通过接线孔与电平开关K5-K0连接。通过拨动电平开关K5-K0，可以对上述信号设置需要的值。按表完成连线，并仔细检查，确保无错接、漏接、虚接。

打开实验设备电源，设备右上方的电源指示灯红灯被点亮。

**2．跟踪控制台操作读寄存器、写寄存器、读存储器、写存储器的执行。**

按复位按钮CLR后，拨动操作模式开关SWC、SWB、SWA到希望的位置，按一次QD按钮，则进入希望的控制台操作模式。控制台模式开关和控制台操作的对应关系如下：

表5.2 控制台模式开关和控制台操作的对应关系

|  |  |  |
| --- | --- | --- |
| 操作模式（SWC、SWB、SWA） | 功能选择 | 备 注 |
| 000 | 启动程序运行 |  |
| 001 | 写存储器 |  |
| 010 | 读存储器 |  |
| 011 | 读寄存器 |  |
| 100 | 写寄存器 |  |

跟踪过程中按一次复位按钮CLR按钮，能够结束本次跟踪操作，开始下一次跟踪操作。

3．跟踪指令的执行

按复位按钮CLR后，设置操作模式开关SWC=0、SWB=0、SWA=0，按一次QD按钮，则进入启动程序运行模式。

设置电平开关K5-K2，使其代表希望的指令操作码IR7-IR4，按QD按钮，跟踪指令的执行。

注意：这里只跟踪每一条指令的执行流程，分析指令执行与相关微操作控制信号的关系，因此与操作数（如果有操作数的话）无关。

跟踪过程中按一次复位按钮CLR按钮，能够结束本次跟踪操作，开始下一次跟踪操作。

**控制台的作用和使用方法：**

## 模式一：写存储器模式：

【1】设置存储器的操作模式 置操作模式：DP=1、SWC=0、SWB=0、SWA=1，按CLR按键进入复位模式

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | AR7-AR0 | SBUS | STOP | MEMW | ARINC |
| 01 | 00H | 01H | 01H |  |  |  |  |  |  |

按QD按键进入写存储器模式

【2】设置存储器的地址

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | D7-D0 | SBUS | STOP | LAR |
| 02 | 03H | 02H | 00H | 20H | 20H | 1 | 1 | 1 |

在数据开关SD7-SD0上设置存储器数据写入的起始地址20H。按一次QD按钮

将20H写入地址寄存器AR。

【3】写第一个数据01H

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | AR7-AR0 | SBUS | STOP | MEMW | ARINC |
| 03 | 02H | 02H | 00H | 01H | 20H | 1 | 1 | 1 | 1 |

在数据开关SD7-SD0上设置需要写入存储器中的数据01H,按一次QD按钮将数据01H写入存储器的20H地址，地址自动加1，下一个数据写入到20H+1的地 址单元。

【4】写第二个数据03H

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | AR7-AR0 | SBUS | STOP | MEMW | ARINC |
| 04 | 02H | 02H | 00H | 03H | 21H | 1 | 1 | 1 | 1 |

在数据开关SD7-SD0上设置需要写入存储器中的数据03H,按一次QD按钮将数据03H写入存储器的21H地址，地址自动加1，下一个数据写入到21H+1的地 址单元。

【5】写第三个数据07H

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | AR7-AR0 | SBUS | STOP | MEMW | ARINC |
| 05 | 02H | 02H | 00H | 07H | 22H | 1 | 1 | 1 | 1 |

在数据开关SD7-SD0上设置需要写入存储器中的数据07H,按一次QD按钮将 数据07H写入存储器的22H地址，地址自动加1，下一个数据写入到22H+1的地 址单元。

【6】参照如上方法写存储器„。

## 模式二：读存储器模式：

【1】设置存储器的操作模式置操作模式：DP=1、SWC=0、SWB=1、SWA=0，按CLR按键进入复位模式

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | AR7-AR0 | SBUS | STOP | MEMW | ARINC |
| 01 | 00H | 01H | 01H |  |  |  |  |  |  |

按QD按键进入读存储器模式

【2】设置存储器的地址

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | D7-D0 | SBUS | STOP | LAR |
| 02 | 05H | 04H | 00H | 20H | 20H | 1 | 1 | 1 |

在数据开关SD7-SD0上设置存储器数据读出的起始地址20H。按一次QD按钮将20H写入地址寄存器AR。

【3】读出第一个数据01H

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | AR7-AR0 | D7-D0 | MBUS | STOP | ARINC |
| 03 | 04H | 04H | 00H | 20H | 01H | 1 | 1 | 1 |

按一次QD按钮将存储器的20H地址的数据显示在指示灯D7-D0上，AR自动加1。

【4】读出第二个数据03H

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | AR7-AR0 | D7-D0 | MBUS | STOP | ARINC |
| 04 | 04H | 04H | 00H | 21H | 03H | 1 | 1 | 1 |

按一次QD按钮将存储器的21H地址的数据显示在指示灯D7-D0上，AR自动加1。

【5】读出第二个数据07H

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | AR7-AR0 | D7-D0 | MBUS | STOP | ARINC |
| 05 | 04H | 04H | 00H | 22H | 07H | 1 | 1 | 1 |

按一次QD按钮将存储器的22H地址的数据显示在指示灯D7-D0上，AR自动加1。

【6】参照如上方法读存储器„。

## 模式三：写寄存器模式：

【1】设置寄存器的操作模式 置操作模式：DP=1、SWC=1、SWB=0、SWA=0，按CLR按键进入复位模式

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | AR7-AR0 | SBUS | STOP | MEMW | ARINC |
| 01 | 00H | 01H | 01H |  |  |  |  |  |  |

按QD按键进入写寄存器模式

【2】写寄存器R0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | SBUS | STOP | DRW | SEL0 |
| 02 | 09H | 08H | 00H | 03H | 1 | 1 | 1 | 1 |

按一次QD按键将数据03H写入寄存器R0。

【3】写寄存器R1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | B7-B0 | SBUS | STOP | DRW | SEL2 |
| 03 | 08H | 0AH | 00H | 0FH | 03H | 1 | 1 | 1 | 1 |

按一次QD按键将数据0FH写入寄存器R1。

【4】写寄存器R2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | B7-B0 | SBUS | STOP | DRW | SEL3 | SEL0 |
| 04 | 0AH | 0CH | 00H | 3FH | 0FH | 1 | 1 | 1 | 1 | 1 |

按一次QD按键将数据3FH写入寄存器R2。

【5】写寄存器R3

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | B7-B0 | SBUS | STOP | DRW | SEL3 | SEL2 | SEL1 |
| 05 | 0CH | 00H | 00H | FFH | 3FH | 1 | 1 | 1 | 1 | 1 | 1 |

按一次QD按键将数据FFH写入寄存器R3。截止到这一步，寄存器R0=03H、寄存器R1=0FH、寄存器R2=3FH、寄存器R3=FFH。

【6】写寄存器结束

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SD7-SD0 | B7-B0 | SBUS | STOP | DRW | SEL1 | SEL0 |
| 06 | 00H | 01H | 01H | XX | 03H | 0 | 0 | 0 | 1 | 1 |

## 模式四：读寄存器模式：

【1】设置寄存器的操作模式 置操作模式：DP=1、SWC=0、SWB=1、SWA=1，按CLR按键进入复位模式

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | SEL3-SEL0 |
| 01 | 00H | 01H | 01H | 0011B |

按QD按键进入读寄存器模式

【2】读寄存器R0、寄存器R1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | A7-A0 | B7-B0 | SEL3-SEL0 | STOP |
| 02 | 07H | 06H | 03H | 0FH | 0001B | 1 |

按一次QD按键，读寄存器R2、寄存器R3。

【3】读寄存器R2、寄存器R3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | A7-A0 | B7-B0 | SEL3-SEL0 | STOP |
| 03 | 06H | 00H | 30H | FFH | 1011B | 1 |

## 模式五：跟踪指令的执行：单纯的指令执行与相对应的控制信号，与数据无关。

按表所示正确连接控制器部件如下信号与实验平台上的电平开关K15-K10。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 电平开关 | K15 | K14 | K13 | K12 | K11 | K10 | 备注 |
| 信号名称 | IR7-I | IR6-I | IR5-I | IR4-I | C-I | Z-I |  |

按表完成连线，并仔细检查，确保无错接、漏接、虚接。

## 具体指令如下：

表5.1 TEC-8模型计算机指令系统

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 助记符 | 功 能 | 指令格式 | | |
| IR(7-4) | IR(3-2) | IR(1-0) |
| 加法 | ADDRD,RS | RD←RD+RS | 0001 | RD | RS |
| 减法 | SUBRD,RS | RD←RD-RS | 0010 | RD | RS |
| 逻辑与 | ANDRD,RS | RD←RD andRS | 0011 | RD | RS |
| 加1 | INCRD | RD ← RD +1 | 0100 | RD | XX |
| 取数 | LDRD,[RS] | RD ←[RS] | 0101 | RD | RS |
| 存数 | ST RS,[RD] | RS →[RD] | 0110 | RD | RS |
| C条件转移 | JCaddr | C=1，则 PC←@ +offset | 0111 | Offset | |
| Z条件转移 | JZaddr | Z=1，则 PC←@ +offset | 1000 | Offset | |
| 无条件转移 | JMP[RD] | PC ←RD | 1001 | RD | XX |
| 输出 | OUTRS | DBUS ←RS | 1010 | XX | RS |
| 停机 | STP | 暂停运行 | 1110 | XX | XX |
| 空操作 | NOP | 空操作 | 0000 | 00 | 00 |

**【01】NOP指令**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 0 | 0 | 0 | 0 | RD | | RS | | 算术 |
| 指令格式 | NOP | | | | | | | |  |
| 指令功能 |  | | | | | | | |  |
| 功能说明 | 空操作 | | | | | | | |  |

第1步：设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮CLR进入复位模式。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 0000B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。

第2步：取指操作

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | 功能 | LIR | PCINC |
| 02 | 01H | 20H | 02H | 取指令 | 1 | 1 |

按QD按钮完成取指操作。

第3步：执行指令

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | 功能 |
| 03 | 20H | 00H | 00H | 执行NOP |

指令系统中，指令操作码0000B没有对应的指令，实际上指令操作码0000B对应着一条NOP指令，即什么也不做的指令。当复位信号CLR为零时，对指令寄存器IR复位，使指令寄存器IR的值0000 0000B,对应一条NOP指令，这样设计的目的是适应指令流水的初始状态要求。指令系统中，指令操作码0000B对应着一条NOP空操作指令，按QD按钮返回微址00H，循环执行。NOP指令执行完成，微地址：**00012000**循环。

## 【02】ADD加法指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 0 | 0 | 0 | 1 | RD | | RS | | 算术 |
| 指令格式 | ADD RD,RS | | | | | | | |  |
| 指令功能 | Rd ← Rd + Rs | | | | | | | |  |
| 功能说明 | 将寄存器RD与寄存器RS的值求和，结果保存到寄存器RD中 | | | | | | | |  |

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮CLR进入复位模式。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 0001B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。

第2步：取指令

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | K15-K12 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 0001B | 1 | 1 |  |

按QD按钮完成取指操作。

第3步:执行ADD加法指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | S3-S0 | CIN | LDC | LDZ | ABUS | DRW |
| 03 | 21H | 01H | 1 | 1001B | 1 | 1 | 1 | 1 | 1 |

指令系统中，指令操作码0001对应着ADD加法指令，，按QD按钮返回微址01H，循环执行。加法指令执行完成，微地址：**00012101**循环。

## 【03】SUB减法指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 0 | 0 | 1 | 0 | RD | | RS | | 算术 |
| 指令格式 | SUB RD,RS | | | | | | | |  |
| 指令功能 | Rd ← Rd - Rs | | | | | | | |  |
| 功能说明 | 将寄存器RD与寄存器RS的值求差，结果保存到寄存器RD中 | | | | | | | |  |

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮CLR进入复位模式

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 0010B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。

第2步：取指令

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | K15-K12 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 0010B | 1 | 1 |  |

按QD按钮完成取指操作。

第3步:执行SUB减法指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | S3-S0 | LDC | LDZ | ABUS | DRW | 备注 |
| 03 | 22H | 01H | 1 | 0110B | 1 | 1 | 1 | 1 |  |

指令系统中，指令操作码0001对应着SUB减法指令，，按QD按钮返回微址01H，循环执行。加法指令执行完成。微地址：**00012201**循环。

## 【04】AND逻辑与指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 0 | 0 | 1 | 1 | RD | | RS | | 逻辑 |
| 指令格式 | AND RD,RS | | | | | | | |  |
| 指令功能 | Rd ← Rd and Rs | | | | | | | |  |
| 功能说明 | 将寄存器RD与寄存器RS的值进行逻辑与，结果保存到寄存器RD中 | | | | | | | |  |

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮CLR进入复位模式

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 0011B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。

第2步：取指令

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | K15-K12 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 0011B | 1 | 1 |  |

按QD按钮完成取指操作。

第3步:执行AND逻辑与指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | M | S3-S0 | LDZ | ABUS | DRW | 备注 |
| 03 | 23H | 01H | 1 | 1 | 1011B | 1 | 1 | 1 |  |

指令系统中，指令操作码0011对应着AND逻辑与指令，按QD按钮执行微址01H。AND逻辑与指令执行完成。微地址：**00012301**循环。

## 【05】INC加1指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 0 | 1 | 0 | 0 | RD | | XX | | 算术 |
| 指令格式 | INC RD | | | | | | | |  |
| 指令功能 | Rd ← Rd and 1 | | | | | | | |  |
| 功能说明 | 将寄存器RD与1进行和运算，结果保存到寄存器RD中 | | | | | | | |  |

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮CLR进入复位模式

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 0100B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。

第2步：取指令

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | K15-K12 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 0100B | 1 | 1 |  |

按QD按钮完成取指操作。

第3步:执行INC加1指令

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | LDC | LDZ | ABUS | DRW | 备注 |
| 03 | 24H | 01H | 1 | 1 | 1 | 1 | 1 |  |

指令系统中，指令操作码0100对应着INC加1指令，，，，，按QD按钮执行微址01H。INC加1指令执行完成。微地址：**00012401**循环。

## 【06】LD取数指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 0 | 1 | 0 | 1 | RD | | RS | |  |
| 指令格式 | LD RD,[RS] | | | | | | | |  |
| 指令功能 | Rd←[Rs] | | | | | | | |  |
| 功能说明 | 将寄存器Rs所代表的地址中的数据送到寄存器Rd中。 | | | | | | | |  |

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮CLR进入复位模式。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 0101B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。

第2步：取指令

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | K15-K12 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 0101B | 1 | 1 |  |

按QD按钮完成取指操作。

第3步:执行LD取数指令

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | M | S3-S0 | ABUS | LAR | 备注 |
| 03 | 25H | 0EH | 1 | 1010B | 1 | 1 |  |

指令系统中，指令操作码0101对应着LD取数指令，按QD按钮执行微址OEH。

第4步:执行LD取数指令2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | MBUS | DRW | 备注 |
| 03 | 0EH | 01H | 1 | 1 | 1 |  |

按QD按钮进行下一步。LD取数指令执行完成。微地址：**0001250E01**循环。

## 【07】ST存数指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 0 | 1 | 1 | 0 | RD | | RS | |  |
| 指令格式 | ST RS,[RD] | | | | | | | |  |
| 指令功能 | Rs→[Rd] | | | | | | | |  |
| 功能说明 | 将寄存器Rs中的数据送到寄存器Rd指定的地址。 | | | | | | | |  |

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮CLR进入复位模式。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 0110B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。

第2步：取指令

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | K15-K12 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 0110B | 1 | 1 |  |

按QD按钮执行微址20H。

第3步:执行ST存数指令

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | M | S3-S0 | ABUS | LAR | 备注 |
| 03 | 26H | 10H | 1 | 1111B | 1 | 1 |  |

按QD按钮执行微址10H。

第4步:执行LD取数指令2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | M | S3-S0 | ABUS | MEMW | 备注 |
| 03 | 10H | 01H | 1 | 1 | 1010B | 1 | 1 |  |

ST存数指令执行完成。微地址：**0001261001**循环。

**【08】JC跳转指令 第一部分：C=1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 0 | 1 | 1 | 1 | OFFSET | | | | 控制传送 |
| 指令格式 | JC addr | | | | | | | |
| 指令功能 | C=1，则PC←@ + offset | | | | | | | |
| 功能说明 | 若标志寄存器C的值为1，则跳转到@+offset地址 | | | | | | | |

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按钮CLR进入复位模式

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | K11 | K10 | 备注 |
| 01 | 00H | 01H | 01H | 0111B | 1 | 0 |  |

根据IR7-IR4分支进行指令的执行操作，按QD按钮准备进行取指操作。

第2步：取指令

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | K15-K12 | K11 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 0111B | 1 | 1 | 1 |  |

按QD按钮进行取指操作。

第3步: 执行**C=1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P2 | 备注 |
| 03 | 27H | **12H** | 1 |  |

此处下地址有错误，在ROM中无法判断C的数值，所以机械性的下地址认为是12H，只能在按QD按钮进行下一步之后转去相应的微地址23H,执行相应的操作。

第4步: 执行JC跳转指令

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | PCADD | 备注 |
| 04 | **13H** | 01H | 1 | 1 |  |

按QD按钮进行下一步，JCC=0指令执行完成。微地址：0001271301循环。

**第二部分：C=0**

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮CLR进入复位模式

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | K11 | SK10 |
| 01 | 00H | 01H | 01H | 0111B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮准备进行取指操作。

第2步：取指令

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 1 | 1 |  |

按QD按钮进行取指操作。

第3步: 执行C=0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P2 | K11 | 备注 |
| 03 | 27H | 12H | 1 | 0 |  |

按QD按钮进行下一步。

第4步: 执行JC跳转指令

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | 备注 |
| 04 | 12H | 01H | 1 |  |

按QD按钮进行下一步，JCC=1指令执行完成。微地址：0001271201循环。

**【09】JZ跳转指令 第一部分：Z=1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 1 | 0 | 0 | 0 | OFFSET | | | | 控制传送 |
| 指令格式 | JZ addr | | | | | | | |
| 指令功能 | Z=1，则PC←@ + offset | | | | | | | |
| 功能说明 | 若标志寄存器Z的值为1，则跳转到@+offset地址 | | | | | | | |

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮 CLR进入复位模式：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 1000B | 0 | 1 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。信号Z被置为1，绿灯Z被点亮。

第 2步：这一步为取指令步骤，但是指令还没有被取出来，下址机械性的转到 20H这个地址单元。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 1 | 1 |  |

按QD按钮进行取指操作

第3步:执行Z=1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P3 | 备注 |
| 03 | 28H | 13H | 1 |  |

按QD按钮进行下一步。

第4步: 执行JZ跳转指令

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | 备注 |
| 03 | 13H | 01H | 1 |  |

按QD按钮进行下一步。JZZ=1指令执行完成。微地址：0001281301循环。

**第二部分：Z=0**

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，LDZ接GND。按复位按钮 CLR进入复位模式：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 1000B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。

第2步：这一步为取指令步骤，但是指令还没有被取出来，下址机械性的转到20H这个地址单元。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 1 | 1 |  |

按QD按钮进行下一步。

第3步: 执行Z=0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P3 | LDZ | 备注 |
| 03 | 28H | 12H | 1 | 0 |  |

按QD按钮进行下一步。

第4步: 执行JZ跳转指令

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | LDZ | 备注 |
| 04 | 12H | 01H | 1 | 0 |  |

按QD按钮进行下一步。JZZ=0指令执行完成。微地址：0001281201循环。

## 【10】JMP无条件转移指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 1 | 0 | 0 | 1 | RD | |  | | 控制传送 |
| 指令格式 | JMP [RD] | | | | | | | |
| 指令功能 | PC←Rd | | | | | | | |
| 功能说明 | 程序无条件跳转到RD地址单元执行，用于短地址跳转。 | | | | | | | |

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮CLR进入复位模式：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 1001B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。

第2步：取指令

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 1 | 1 |  |

按QD按钮执行微址29H。

第3步:执行JMP 无条件转移指令

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | M | S3-S0 | ABUS | LPC | 备注 |
| 03 | 29H | 01H | 1 | 1 | 1111B | 1 | 1 |  |

指令系统中，指令操作码1010对应着无条件转移指令，按QD按钮执行微址01H。JMP无条件转移指令执行完成。微地址：**00012901**循环。

## 【11】OUT输出指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 1 | 0 | 1 | 0 | XX | | RS | |  |
| 指令格式 | OUT RS | | | | | | | |  |
| 指令功能 | DBUS←Rs | | | | | | | |  |
| 功能说明 | 将寄存器RS中的数据直接送往数据总线 | | | | | | | |  |

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮CLR进入复位模式

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 1010B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。

第2步：取指令

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号号 | µA5-µA0 | NµA5-NµA0 | P1 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 1 | 1 |  |

按QD按钮执行微址2AH。

第3步:执行OUT 输出指令

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | M | S3-S0 | ABUS | 备注 |
| 03 | 2AH | 01H | 1 | 1 | 1010B | 1 |  |

指令系统中，指令操作码1010对应着OUT输出指令，按QD按钮顺序执行微址00H。OUT输出指令执行完成。微地址：**00012A00**循环。

## 【12】STP停机指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令编码 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 备注 |
| 1 | 1 | 1 | 0 | XX | | XX | |  |
| 指令格式 | STP | | | | | | | |  |
| 指令功能 | 停止运行 | | | | | | | |  |
| 功能说明 | 停止运行或者空操作 | | | | | | | |  |

第1步设置操作模式：DP=1、SWC=0、SWB=0、SWA=0，按复位按钮CLR进入复位模式

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4-P0 | K15-K12 | SK11 | SK10 |
| 01 | 00H | 01H | 01H | 1110B | 0 | 0 |

根据IR7-IR4分支进行指令的执行操作，按QD按钮进行取指操作。

第2步：取指令

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P1 | LIR | PCINC | 备注 |
| 02 | 01H | 20H | 1 | 1 | 1 |  |

按QD按钮执行微址2EH。

第3步:执行STP 停机指令

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | µA5-µA0 | NµA5-NµA0 | P4 | STOP | 备注 |
| 03 | 2EH | 01H | 1 | 1 |  |

指令系统中，指令操作码1110对应着STP停机指令，按QD按钮返回微址01H。STP停机指令执行完成。微地址：**00012E01**循环。

## 七、实验要求

1．认真做好实验的预习，掌握TEC-8模型计算机微程序控制器的工作原理。

2．写出实验报告，内容是：

⑴实验目的和实验内容。

⑵控制台操作的跟踪过程。写出每一步的微地址 µA5-µA0、后继微地址NµA5-NµA0、判别位P4-P0和有关控制信号的值。

⑶写出这4种控制台操作的作用和使用方法。

⑷指令的跟踪过程。写出每一步的微地址µA5-µA0、后继微地址NµA5-NµA0、判别位 P4-P0 和有关控制信号的值,时序波形图和测量值。

⑸写出TEC-8模型计算机中的微地址转移逻辑的逻辑表达式。分析它和各种微程序分支的对应关系，观察时序信号，熟悉微指令格式的定义和微程序代码表。

**八、可探索和研究的问题**

1．如图5.11所示，在写寄存器操作中写入寄存器R0时，信号SEL1-SEL0=10B，这是必需的吗？可不可以是其他值？

2.试根据运算器组成实验、双端口存储器实验和数据通路实验的实验过程，画出这部分的微程序流程图。

3．图5.11以微序流程图的方式给出了模型机指令执行以及控制台操作的具体过程，它其实是微程序控制计算机中的微程序的一种表达方式。微程序控制计算机中的微程序还有另一种方式是微指令代码表。

如表5.3所示的就是微指令代码表样例，它以表格形式注明了控存中每一个地址中各个微操作控制信号的值，并记录后继微地址。表中已经把TEC-8微程序控制器中00H、01H、02H单元的相关信息填好了，请尝试把微指令代码表填写完整。

在进行微程序控制计算机的设计过程中，填好了这个微指令代码表后，只需要把相关内容送入控制存储器中，微程序计算机的控制器就差不多设计完成了。

4.分析微程序流程图与转移逻辑、时序信号的关系，请解释微操作控制信号STOP的作用。这个信号与单微指令开关DP有无相似之处？

5. 画出运算器、存储器、数据通路、微程序控制器流程图。

6.已知TEC-8模型计算机可用控存空间中11H、2FH、3FH共3个单元空闲，试根据现有的结构、微操作控制信号和机器时序，尝试进行指令扩展，扩展出一条新指令。要求给出该指令对应的微程序，列出每一拍所需的微操作控制信号，分配控存空间，并验证其功能。

提示：可以尝试下列指令，观察能否扩展？如果能，怎样扩展？如果不能，为什么（不定限于下列指令，但是要求所扩展的指令尽量功能比较明确、单一，且应该能完成一些经常会用到的功能）？

OR R0，R1 //或操作，R0与R1进行逻辑或操作，结果送往R0

NOT R0 //按位求反，R0按位取反，结果送往R0

LEET R0 //左移，将R0左移1位，结果送往R0

MOV R0，R1 //数据传送，将R1的值送往R0

LEET RO，X //左移X位，将R0左移x位，结果送往R0

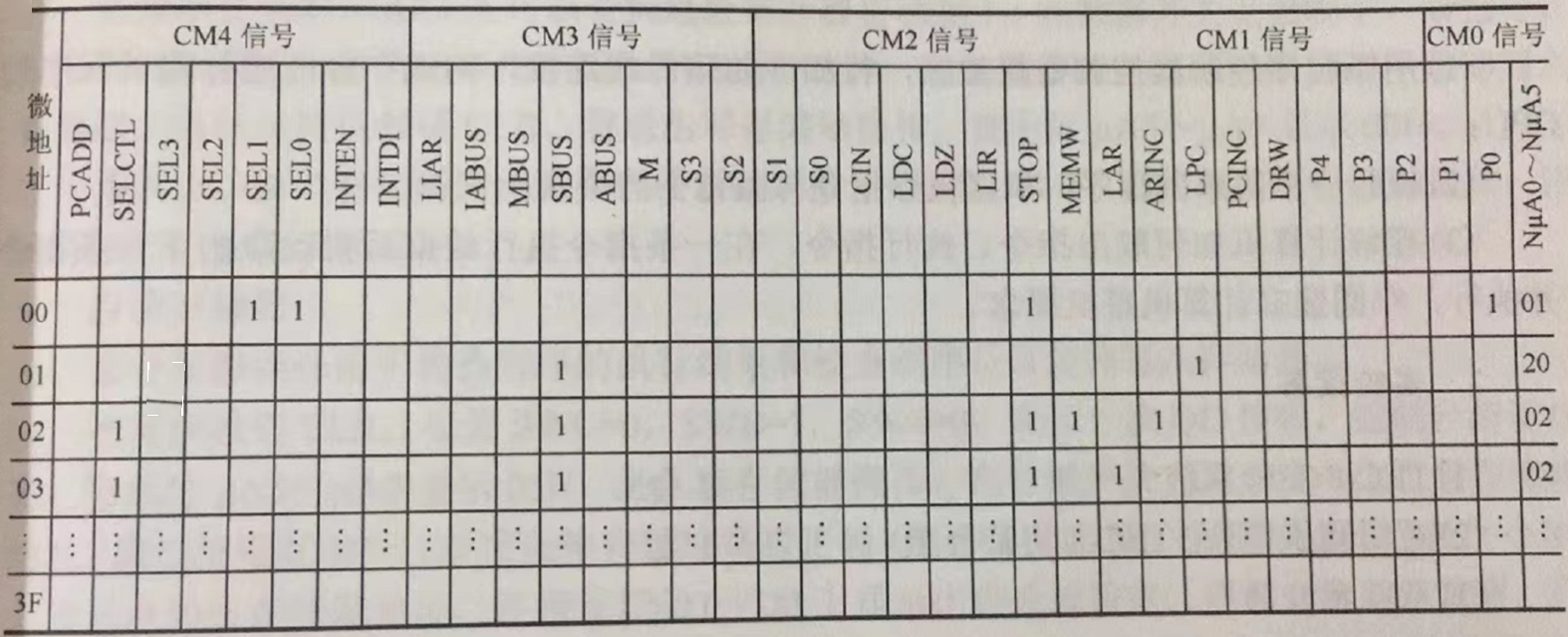
MUL RO，R1 //无符号乘法，R0与R1相乘，结果送往R0

上述指令中，假如乘法指令只是2个无符号4位数相乘呢？假如控存空间不加以限制（即不考虑控存中存储单元的分配）呢？

7.采用字段编制方式设计新的微指令格式，并重新设计几条机器指令的微程序并代码化，经老师允许，可以写入 E2PROM\_HN58C65,进行调试。

8.能将图2.7中的微指令格式重新设计压缩长度吗？

表5.3 微指令代码表样例



实验报告记录：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验题目 |  | | | 备注： |
| 姓 名 |  | 学 号 |  |  |
|  | | | | |