**计算机组成原理实验箱操作实验**

所谓手动控制，就是用二进制拨动开关模拟微控制信号，以手动方式设置相互关联的逻辑

控制电位，建立“源与目”的有效状态，实现和完成实验制定的控制仼务。

手动控制是原理计算机的基础，我们从部件单元电路入手，围饶单元部件、关联部件及微控

制器由浅入深地逐一展开，为原理模型机的设计与实现奠定基础。

本系统手控态提供“在线”与“搭接”两种实验方式。“在线”态按微控制器设计规范定义和命名控制信号，实验时必须遵循，不得愈越，否则有误。在“搭接”态可忽略微控制器组成环节，自行设计和构造原理计算机部件，自由定义和搭接单元部件与关联部件电路，力求提高学生的动手能力，亦为课程设计与超越创新实验的展开预留一个灵活而又便捷的可实施平台

**2.1 手动实验环境的建立**

1. **初始待令状态**

上电或按复位，系统无条件进入初始待令状态，LCD显示器按原设定的摸式显示如下画面：

**Dais-CMX16+ ’kld**

**K**

**Dais-CMX16+ ’mud**

**M**

**Dais-CMX16+ ’pld**

**P**

单元手动模式　　　　　　　　　　微控制器模式　　　　　　　　　　组合逻辑模式

第1行包含了产品型号和控制字，第2行的光标闪动位显示提示符，表示实验系统处于初始待令状态，可以进入系统按键命令所定义的操作。

1. **工作摸式设置**

在初始待令状态下，按【减址】键，LCD显示器提示工作模式选项：

按【增址】键，将光标移到“KLD”单元手动模式，按【减址】键确定后，询问用户是否使用搭接方式的选项：

**Dais-CMX16+ ’XXX**

**KLD MUD PLD**

**选择手动模式**

**Dais-CMX16+ ’kld**

**lst y/n**

**Dais-CMX16+ ’kld**

**lst y/n**

**是，选择搭接方式，须连线**

**否，选择在线方式，零连线**

按【增址】键，将光标移到“y”选择**搭接方式**、或将光标移到“n”选择**在线方式**，按【减址】键确定设置，返回待命状态。

**Dais-CMX16+ ’kld**

**K**

**设置完毕，返回待命状态**

设置工作状态亦可借助PC系统在达爱思集成开发环境下进行，使用方法可参阅第六章。

**2.2　手控实验提示**

1. **初始化操作**

一旦进入手控状态，首先应把实验系统左下方“二进制开关单元”的24位微控制开关拨至下方（即高电平信号“1”），使24位微控制状态指示灯处“暗态”，关闭全部控制信号，完成微控制器的初始化操作。

1. **控制信号的建立**

⑴有效状态的特征：本系统提供的是“负逻辑”控制电路，通常情况下把低电平“0”定义为有效状态，以点亮绿色发光二极管为标志。

⑵有效状态的建立：结合实验项目，按实验要求把相关的二进制开关拨向上方，点亮对应的发光二极管。

⑶有效状态的控制

①源操作：为了避免总线的冲突与竞争，保证其唯一性，系统以编码方式定义总线来源，实验时必须按照源编码表的要求选择当前总线源。例如：IN AX,i/o 源编码开关X2、X1、X0对应二进制开关设为“100”，再把“XP”、“W”对应的二进制开关拨向上方，此时I/O口单元右上方的奇偶绿色指示灯“亮”，然后按动I/O口单元的数据开关，总线单元的显示器及二进制发光管应随之变化。

②目的操作：首先应把与控制目的操作部件相对应的二进制开关拨向上方，即O2、O1、O0对应二进制开关设为“100”，再把“OP”拨向上方，建立目的寻址的有效状态，其特征是该部件单元周边的黄色指示灯“亮”，然后按动【单拍】按钮向目的部件提供写脉冲，把数据打入黄色指示灯处“亮”旁边的部件中。例如：IN AX,I/O在源状态已建立的基础上，首先令I/O口单元的数据开关为“00010010 00110100”，以AX部件处黄色指示灯“亮”为条件按动【单拍】按钮，把I/O的内容打入AX，该部件单元显示器显示“1234H”。

③源与目的编码

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **表2.2 源与目的寻址编码表** | | | | | | | | |
| 总线源编码 | | | |  | 在线态目的编码 | | | |
| M10 | M9 | M8 | 功能 | M19 | M18 | M17 | 功能 |
| X2 | X1 | X0 | O2 | O1 | O0 |
| 1 | 1 | 1 | 禁止 | 1 | 1 | 1 | 禁止 |
| 1 | 1 | 0 | ALU | 1 | 1 | 0 | MAR |
| 1 | 0 | 1 | SP | 1 | 0 | 1 | BX |
| 1 | 0 | 0 | IOR | 1 | 0 | 0 | AX |
| 0 | 1 | 1 | MRD | 0 | 1 | 1 | SP |
| 0 | 1 | 0 | XRD | 0 | 1 | 0 | IOW |
| 0 | 0 | 1 | RRD | 0 | 0 | 1 | XWR |
| 0 | 0 | 0 | PC | 0 | 0 | 0 | RWR |

上表中源编码无“在线”与“搭接”之分，八中选一属必须遵守的潜规则；之于目的编码表的定义仅适用于“在线”方式；在“搭接”方式，由实验者自行定义。

**3）设计与连接环节提示**

1. **实验方式设置规定**

由于“在线”方式下，控制器与部件之间已建立了“主从式”控制关系，也就是说控制器已进入载体工作状态，此时若出现对控制器已定义部件的“实验搭接”，从控制角度称谓“重叠定义”，造成总线混乱；从电路的层面讲称谓超负荷加载，危及器件安全。因此“在线”方式下对于控制器已定义部件的“实验电路连接”是不允许的，属非法操作。在实验方式选择与相互转换中应做到：

①由“搭接”方式转向“在线”方式时，首先卸去所有的实验连接，然后才能进入在线方式的设置操作。

②在进入“实验电路连接”操作之前，必须判断当前的实验环境是否处“搭接”状态，若否首先设置搭接实验方式，然后才能进入“实验电路的搭接”。

1. **“搭接”态并行性设计制约**

二进制模拟控制开关K23~K0的负载能力为四个TTL电平，在构造并行输岀控制信号时它的受控部件通常以三个以下为宜。为了延缓模拟控制电路的使用寿命，对于三个以上TTL电平的控制应通过追加隔离驱动电路的途径耒实现。

1. **连接与拆除环节的注意事项**
2. 实验连接环节

实验连接环节的失误轻者影响实验结果导致实验失败，重者危及设备受损延误实验进程，为此我们提倡“开电源”操作，在设备上电的环境下，首先判断当前的实验环境是否处“搭接”状态，若否先设置搭接实验方式，然后才能进入“实验电路的搭接”。完成实验电路的连接。

1. 实验拆除环节

在实验连线拆除过程中，实验连线头与头之间的单边碰撞不可避免，容易引发短路危及设备安全。因此切断电源，坚持无源拆除是唯一的选择。另外为了延缓双头实验导线的使用寿命，在拆除过程中应做到抓头不拉线，只拔不拉，垂直发力，杜绝横向硬拆。

* 1. **手控态按键命名**

⑴【减址】：模式选择命令，主要用于更改“控制字”与“连接字”。

⑵【返回】：系统复位命令，主要用于带预置或清零控制部件的复位。涉及的清零器件为PC、uPC、AR、A、B寄存器及CY与零标志；涉及置位操作的是十六位I/O。

**2.3　手动“搭接”实验示例**

在手动“搭接”态可忽略微控制器组成环节，自行设计和构造原理计算机部件，自由定义和搭接单元部件与关联部件电路，力求提高学生的动手能力。

所谓手控“搭接”实验是以自行定义和连接单元部件与关联部件电路为前提，在进入“实验电路连接”操作之前，必须判断当前的实验环境是否处“手动”、“搭接”状态，若否首先按照本实验指导书P7页提供的操作方法，把实验箱工作状态设置为“手动”、“搭接”，然后才能进入“实验电路的搭接”。

**2.3.1　十六位机运算器实验**

**一、实验目的**

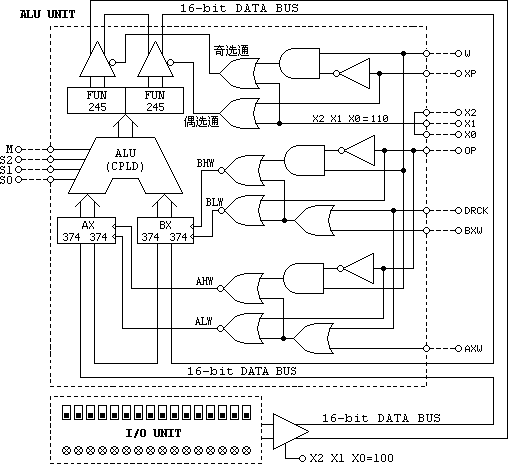
掌握十六位机字与字节运算的数据传输格式，验证运算功能发生器及进位控制的组合功能。

**二、实验要求**

完成算术、逻辑、移位运算实验，熟悉ALU运算控制位的运用。

**三、实验原理**

实验中所用的运算器数据通路如图2-3-1所示。ALU运算器由CPLD描述。运算器的输出经过2片74LS245三态门与数据总线相连，2个运算寄存器AX、BX的数据输入端分别由4个74LS574锁存器锁存，锁存器的输入端与数据总线相连，准双向I/O输入输出端口用来给出参与运算的数据，经2片74LS245三态门与数据总线相连。



**图2-3-1　运算器数据通路**

图2-3-1中，AXW、BXW在“搭接态”由实验连接对应的二进制开关控制，“0”有效，通过【单拍】按钮产生的负脉冲把总线上的数据打入，实现AXW、BXW写入操作。

**四、运算器功能编码**

**表2.3.1 ALU运算器编码表**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 算术运算 | | | | |  | 逻辑运算 | | | | |
| M15 | M13 | M12 | M11 | 功能 |  | M15 | M13 | M12 | M11 | 功能 |
| M | S2 | S1 | S0 | M | S2 | S1 | S0 |
| 0 | 0 | 0 | 0 | A+B+C | 1 | 0 | 0 | 0 | B |
| 0 | 0 | 0 | 1 | A—B—C | 1 | 0 | 0 | 1 | /A |
| 0 | 0 | 1 | 0 | RLC | 1 | 0 | 1 | 0 | A-1 |
| 0 | 0 | 1 | 1 | RRC | 1 | 0 | 1 | 1 | A=0 |
| 0 | 1 | 0 | 0 | A+B | 1 | 1 | 0 | 0 | A#B |
| 0 | 1 | 0 | 1 | A—B | 1 | 1 | 0 | 1 | A&B |
| 0 | 1 | 1 | 0 | RL | 1 | 1 | 1 | 0 | A+1 |
| 0 | 1 | 1 | 1 | RR | 1 | 1 | 1 | 1 | A |

**五、实验连线**

K23~K0置“1”，灭M23~M0控位显示灯。然后按下表要求“搭接”部件控制电路。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **连线** | **信号孔** | **接入孔** | **作用** | **有效电平** |
| 1 | DRCK | CLOCK | 单元手动实验状态的时钟来源 | 下降沿打入 |
| 2 | W | K6(M6) | 总线字长：0=16位字操作，1=8位字节操作 |  |
| 3 | XP | K7(M7) | 源部件奇偶标志：0=偶寻址，1=奇寻址 |  |
| 4 | X2 | K10(M10) | 源部件定义译码端X2 | 三八译码  八中选一  低电平有效 |
| 5 | X1 | K9(M9) | 源部件定义译码端X1 |
| 6 | X0 | K8(M8) | 源部件定义译码端X0 |
| 7 | M | K15(M15) | 运算控制位：0=算术运算，1=逻辑运算 |  |
| 8 | S2 | K13(M13) | 运算状态位S2 |  |
| 9 | S1 | K12(M12) | 运算状态位S1 |  |
| 10 | S0 | K11(M11) | 运算状态位S0 |  |
| 11 | OP | K16(M16) | 目标部件奇偶标志：0=偶寻址，1=奇寻址 |  |
| 12 | AXW | K17(M17) | AX运算寄存器写使能，本例定义到M17位 | 低电平有效 |
| 13 | BXW | K18(M18) | BX运算寄存器写使能，本例定义到M18位 | 低电平有效 |

**示例1　算术运算**

* 1. **字算术运算**
  2. 字写操作

通过I/O单元“S15~S0”开关向累加器AX和暂存器BX置数，操作步骤如下：

置数

I/O=1122h

数据来源

I/O单元

寄存器AX

(1122h)

K10~K6=10000

K18~K16=100

按【单拍】按钮

置数

I/O=3344h

寄存器BX

(3344h)

K18~K16=010

按【单拍】按钮

* 1. 字读操作

关闭AX、BX写使能，令K18=K17=“1”，按下流程分别读AX、BX。

读FUN

关闭写控位

数据总线

显示AX值

K18~K17=11

K10~K6=11000

读BX

K13~K11=000

数据总线

显示BX值

K15~K11=11111

* 1. 字加法与减法运算

令M S2 S1 S0（K15 K13~K11=0100），FUN及总线单元显示AX+BX的结果。

令M S2 S1 S0（K15 K13~K11=0101），FUN及总线单元显示AX－BX的结果。

* 1. **字节算术运算**

1. 偶字节写（置数操作）

通过I/O单元“S15~S0”开关向累加器AL和暂存器BL置数，具体操作步骤如下：

置数

I/O=XX22h

数据来源

I/O单元

寄存器AL

(22h)

K10~K6=10001

K18~K16=100

按【单拍】按钮

置数

I/O=XX44h

寄存器BL

(44h)

K18~K16=010

按【单拍】按钮

1. 偶字节读操作（运算寄存器AL和BL内容送总线）

关闭AL、BL写使能，令K18=K17=1，按下流程分别读AL、BL。

读FUN

关闭写控位

数据总线

显示AL值

K18~K17=11

K10~K6=11001

读BL

K13~K11=000

数据总线

显示BL值

K15~K11=11111

1. 字节减法算术运算（不带进位加）

令M S2 S1 S0（K15 K13~K11=0100），FUN及总线单元显示AL+BL的结果。

令M S2 S1 S0（K15 K13~K11=0101），FUN及总线单元显示AL－BL的结果。

**示例2　逻辑运算**

1. **字逻辑运算**
2. 字写操作（置数操作）

通过I/O单元“S15~S0”开关向累加器AX和暂存器BX置数，操作步骤如下：

置数

I/O=1122h

数据来源

I/O单元

寄存器AX

(1122h)

K10~K6=10000

K18~K16=100

按【单拍】按钮

置数

I/O=3344h

寄存器BX

(3344h)

K18~K16=010

按【单拍】按钮

1. 字读操作（运算寄存器AX和BX内容送总线）

关闭AX、BX写使能，令K18=K17=“1”，按下流程分别读AX、BX。

读FUN

关闭写控位

数据总线

显示AX值

K18~K17=11

K10~K6=11000

读BX

K13~K11=000

数据总线

显示BX值

K15~K11=11111

1. 字逻辑运算

令M S2 S1 S0（K15 K13~K11=1101），为逻辑与，FUN及总线显示AX逻辑与BX的结果。

令M S2 S1 S0（K15 K13~K11=1100），为逻辑或，FUN及总线显示AX逻辑或BX的结果。

1. **字节逻辑运算**
2. 偶字节写操作（置数操作）

通过I/O单元“S15~S0”开关向累加器AL和暂存器BL置数，具体操作步骤如下：

置数

I/O=XX22h

数据来源

I/O单元

寄存器AL

(22h)

K10~K6=10001

K18~K16=100

按【单拍】按钮

置数

I/O=XX44h

寄存器BL

(44h)

K18~K16=010

按【单拍】按钮

1. 偶字节读操作（运算寄存器AL和BL内容送数据总线）

关闭AL、BL写使能，令K18=K17=1，按下流程分别读AL、BL。

读FUN

关闭写控位

数据总线

显示AL值

K18~K17=11

K10~K6=11001

读BL

K13~K11=000

数据总线

显示BL值

K15~K11=11111

1. 若运算控制位设为（M S2 S1 S0=1111）则F=AL，即AL内容送到数据总线。
2. 若运算控制位设为（M S2 S1 S0=1000）则F=BL，即BL内容送到数据总线。
3. 偶字节逻辑运算

令M S2 S1 S0（K15 K13~K11=1101），为逻辑与，FUN及总线显示AL逻辑与BL的结果。

令M S2 S1 S0（K15 K13~K11=1100），为逻辑或，FUN及总线显示AL逻辑或BL的结果。

1. 奇字节写操作（置数操作）

通过I/O单元“S15~S0”开关向累加器AH和暂存器BH置数，具体操作步骤如下：

置数

I/O=XX22h

数据来源

I/O单元

寄存器AH

(22h)

K10~K6=10001

K18~K16=101

按【单拍】按钮

置数

I/O=XX66h

寄存器BH

(66h)

K18~K16=011

按【单拍】按钮

1. 奇字节读操作（运算寄存器AH和BH内容送数据总线）

关闭AH、BH写使能，令K18=K17=1，按下流程分别读AH、BH。

读FUN

关闭写控位

数据总线

显示AH值

K18~K17=11

K10~K6=11011

读BH

K13~K11=000

数据总线

显示BH值

K15~K11=11111

1. 若运算控制位设为（M S2 S1 S0=1111）则F=AH，即AH内容送到数据总线。
2. 若运算控制位设为（M S2 S1 S0=1000）则F=BH，即BH内容送到数据总线。
3. 奇字节逻辑运算

令M S2 S1 S0（K15 K13~K11=1101），为逻辑与，FUN及总线显示AH逻辑与BH的结果。

令M S2 S1 S0（K15 K13~K11=1100），为逻辑或，FUN及总线显示AH逻辑或BH的结果。

**示例3　移位运算**

标准运算器的累加器A在运算中通常含有移位的功能，为了规范运算器的设计，Dais-CMX16+在移位运算时把累加器A视为移位的源寄存器，也就是说移位是通过累加器A实现的，这种规范的设计理念使我们的运算器能够与标准机微机系统相吻合，达到可以面向源程序级（汇编及C语言）的水准。

1. **移位控制编码**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表2.3.2 移位编码表** | | | | |
| M15 | M13 | M12 | M11 | 功能 |
| M | S2 | S1 | S0 |
| 0 | 1 | 1 | 1 | RR |
| 0 | 1 | 1 | 0 | RL |
| 0 | 0 | 1 | 1 | RRC |
| 0 | 0 | 1 | 0 | RLC |

1. **移位执行过程**
2. 循环左移
3. 循环右移
4. 带进位循环左移

CY

1. 带进位循环右移

CY

所谓循环移位，就是指移位时数据的首尾相连进行移位，即最高（最低）位的移出位又移入数据的最低（最高）位。根据循环移位时进位位是否一起参加循环，可将循环移位分为不带进位循环和带进位循环两类。其中不带进位循环是指进位“CY”的内容不与数据部分一起循环移位，也称小循环。带进位循环是指进位 “CY”中的内容与数据部分一起循环移位，也称大循环。

◆不带进位循环左移：各位按位左移，最高位移入最低位。

◆不带进位循环右移：各位按位右移，最低位移入最高位。

◆带进位循环左移：各位按位左移，最高位移入C中，C中内容移入最低位。

◆带进位循环右：各位按位右移，最低位移入C中，C中内容移入最高位。

循环移位一般用于实现循环式控制、高低字节的互换，还可以用于实现多倍字长数据的算术移位或逻辑移位。

1. **字移位运算**
2. 向AX寄存器置数

拨动“I/O输入输出单元”开关向移位源寄存器AX置数，具体操作步骤如下：

置数

I/O=0080h

数据来源

I/O单元

AX写选通

K10~K6=10000

K18~K16=100

AX=0080

按【单拍】按钮

1. AX寄存器移位

S2 S0方式选择

移位控制

M=0 S1=1

数据来源

AX

移位选通

K10~K6=11000

移位结果

打入AX

按【单拍】按钮

K18~K16=100

令M=0 S1=1参照表2.3.2改变S2、S0的状态，再按动【单拍】按钮，观察AX移位变化。

1. **字节移位运算**
   1. AL寄存器置数

拨动“I/O输入输出单元”开关向移位源寄存器AL置数，具体操作步骤如下：

置数

I/O=XX80h

数据来源

I/O单元

AL写选通

K10~K6=10001

K18~K16=100

AL=80

按【单拍】按钮

* 1. AL寄存器移位

S2 S0方式选择

移位控制

M=0 S1=1

数据来源

AL

移位选通

K10~K6=11001

移位结果

打入AL

按【单拍】按钮

K18~K16=100

令M=0 S1=1参照表2.3.2改变S2、S0的状态，再按动【单拍】按钮，观察AL移位变化。

* 1. AH寄存器置数

拨动“I/O输入输出单元”开关向移位源寄存器AH置数，具体操作步骤如下：

置数

I/O=XX80h

数据来源

I/O单元

AH写选通

K10~K6=10001

K18~K16=101

AH=80

按【单拍】按钮

* 1. AH寄存器移位

S2 S0方式选择

移位控制

M=0 S1=1

数据来源

AH

移位选通

K10~K6=11011

移位结果

打入AH

按【单拍】按钮

K18~K16=101

令M=0、S1=1参照表2.3.2改变S2、S0的状态，再按动【单拍】按钮，观察AH移位变化。

**实验思考**

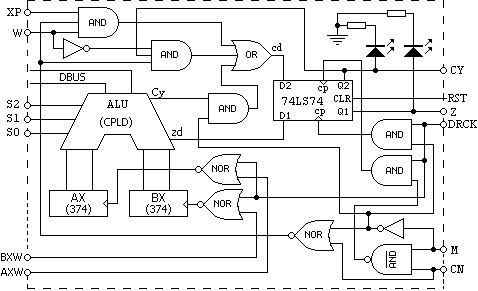
验证表2.3.3 ALU运算器编码表所列的运算功能。

在给定AX=6655h、BX=AA77h的情况下，改变运算器的功能设置，观察运算器的输出，填入下页表格中，并和理论分析进行比较、验证。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **表2.3.3　ALU运算器真值表** | | | | | | | | |
| **运算控制** | **运算表达式** | **M** | **S2** | **S1** | **S0** | **AX** | **BX** | **运算结果** |
| 带进位算术加 | A+B+C | 0 | 0 | 0 | 0 | 6655 | AA77 | FUN=( 10CC ) |
| 带借位算术减 | A-B-C | 0 | 0 | 0 | 1 | 6655 | AA77 | FUN=( BBDE ) |
| 带进位左移 | RLC A | 0 | 0 | 1 | 0 | 6655 | AA77 | FUN=( CCAA ) |
| 带进位右移 | RRC A | 0 | 0 | 1 | 1 |  |  | FUN=( ) |
| 算术加 | A+B | 0 | 1 | 0 | 0 |  |  | FUN=( ) |
| 算术减 | A-B | 0 | 1 | 0 | 1 |  |  | FUN=( ) |
| 左移 | RL A | 0 | 1 | 1 | 0 |  |  | FUN=( ) |
| 右移 | RR A | 0 | 1 | 1 | 1 |  |  | FUN=( ) |
| 取BX值 | B | 1 | 0 | 0 | 0 | 6655 | AA77 | FUN=( AA77 ) |
| AX取反 | NOT A | 1 | 0 | 0 | 1 | 6655 | AA77 | FUN=( 99AA ) |
| AX减1 | A-1 | 1 | 0 | 1 | 0 | 6655 | AA77 | FUN=( 6654 ) |
| 清零 | 0 | 1 | 0 | 1 | 1 |  |  | FUN=( ) |
| 逻辑或 | A OR B | 1 | 1 | 0 | 0 |  |  | FUN=( ) |
| 逻辑与 | A AND B | 1 | 1 | 0 | 1 |  |  | FUN=( ) |
| AX加1 | A+1 | 1 | 1 | 1 | 0 |  |  | FUN=( ) |
| 取AX值 | A | 1 | 1 | 1 | 1 |  |  | FUN=( ) |

**示例4　进位控制与零标志**

1. **标志控制原理**



**图2-3-2　标志位锁存原理图**

1. **进位标志CY**

运算标志CY是带复位可预置的进位标志，在运算时由M信号控制，当M=0时，按【单拍】按钮，在DRCK下降沿把当前运算溢出位（进位或借位）打入CY的锁存输出端Q2。在M=1时，由CN位控制CY的“位操作”。当CN=0时，按【单拍】按钮，在DRCK下降沿执行CY的置“1”、清零、取反操作；遇M=1、CN=1时，CY保持原始状态。CY的复位端由管理CPU直接控制。

1. **零标志Z**

零标志Z是带复位端的运算结果判零标志，取源于运算器输出FUN。当FUN=00h时，zd为“1”，反之为“0”。；零标志由M信号控制，当M=0时，按【单拍】按钮，在DRCK下降沿把当前zd打入Z（零标志）的锁存输出端Q1。遇M=1，零标志Z保持原始状态。

1. **标志位的初始化**

标志位的清零端由系统掌控与控位无关，在手动实验中需要清标志时，通过按动【返回】键来实现标志位的初始化。

系统提供CY与Z的状态灯，其显示特征为：进位标志CY=1时，灯“亮”，反之灯“灭”；零标志Z=1时，灯“亮”，反之灯“灭”。

1. **实验连线**
   1. 按实验指导书P11页表2.3.2的要求搭接运算器。
   2. 进位控制信号CN与K14所对应的引出端M14相连。
2. **进位标志CY实验**
   1. *CY标志控制实验*
   2. 进位控制流程

进位控制

标志初始化

CY=0

方式选择

按【返回】键

改变K7 K6

位操作

K15 K14=10

按【单拍】按钮

关闭控位

K23~K0=全”1”

* 1. 按上流程建立控制状态，参照下表改变方式控制，按【单拍】观察CY灯的变化

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **表2.3.4 进位控制编码** | | | | | | |
| 进位标志位操作 | | | | | 进位 | 功能说明 |
| K15 | K14 | K7 | K6 | DRCK | CY |
| M | CN | XP | W |
| 1 | 0 | 0 | 1 | ↓ | 0 | 清零 |
| 1 | 0 | 1 | 1 | ↓ | 1 | 置位 |
| 1 | 0 | X | 0 | ↓ | /CY | 取反 |

* 1. *进位运算实验*

1. 按【返回】键，通过I/O单元向AX、BX分别置数5678h、ABCDh，操作如下：

置数

I/O=FFFEh

数据来源

I/O单元

AX=FFFEh

K10~K6=10000

按【单拍】按钮

置数

I/O=0003h

BX=0003h

按【单拍】按钮

K18~K16=100

K18~K16=010

1. 带进位加减法运算流程

运算控制

数据来源

FUN

加减选择

K10~K6=11000

改变K11

运算标志

CY

K18~K16=100

K15 K13 K12=000

关闭控位

K23~K0=全”1”

1. 在完成上流程中规定的设置操作后，按下列顺序执行带进位加减法运算实验：

带进位加法 令K11=0 AX=FFFEH,BX=0003H,CY=0,DBUS=FUN=0001H

按【单拍】钮 AX=0001H,BX=0003H,CY=1,DBUS=FUN=0005H

再按【单拍】钮 AX=0005H,BX=0003H,CY=0,DBUS=FUN=0008H

带进位减法 令K11=1 AX=0005H,BX=0003H,CY=0,DBUS=FUN=0002H

按【单拍】钮 AX=0002H,BX=0003H,CY=0,DBUS=FUN=FFFFH

再按【单拍】钮 AX=FFFFH,BX=0003H,CY=1,DBUS=FUN=FFFBH

* 1. 零标志实验（字识别）

1. 向寄存器AX和BX置数

通过“I/O输入输出单元”向AX、BX均置数0001h，操作步骤如下：

置数

I/O=0001h

数据来源

I/O单元

AX=0001h

K10~K6=10000

按【单拍】按钮

置数

I/O=0001h

BX=0001h

按【单拍】按钮

K18~K16=100

K18~K16=010

1. 验证零标志锁存功能

按【返回】键清Z标志，按下流程建立零标志运算控制状态。K11=1时减法运算FUN=0000，按【单拍】钮Z标志灯亮（Z=1）；K11=0时加法运算FUN=0002按【单拍】钮Z标志灯灭（Z=0）。

选择减法

关闭控位

标志灯亮

Z=1

K23~K0=全”1”

选择加法

标志灯灭

Z=0

按【单拍】按钮

K15~K11=01101

K11=0

K10~K6=11000

按【单拍】按钮

**2.3.2　通用寄存器实验**

**一、实验目的**

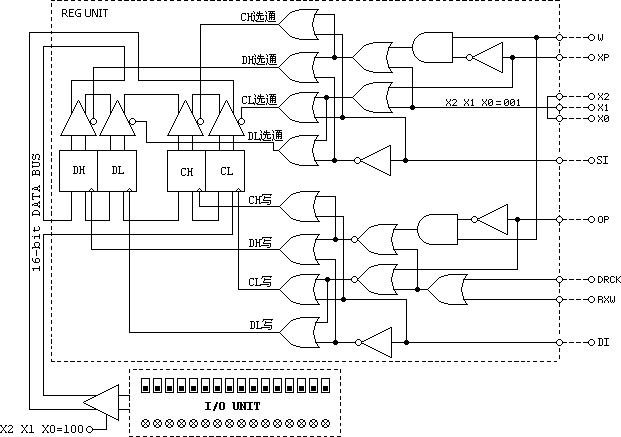
1. 熟悉通用寄存器的数据通路。
2. 掌握通用寄存器的构成和运用。

**二、实验要求**

在掌握了AX、BX运算寄存器的读写操作后，继续完成CX、DX通用寄存器的数据写入与读出。

**三、实验原理**

实验中所用的通用寄存器数据通路如下图所示。由四片8位字长的74LS574组成CX（R1 R0）、DX（R3 R2）通用寄存器组。图中X2 X1 X0定义输出选通使能，SI、XP控制位为源选通选择。RXW为寄存器数据写入使能，OP、DI为目的寄存器选择。DRCK信号为寄存器写脉冲，下降沿有效。准双向I/O输入输出端口用于置数操作，经2片74LS245三态门与数据总线相连。



**图2-3-3通用寄存器数据通路**

**四、实验内容**

1. **实验连线**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **连线** | **信号孔** | **接入孔** | **作用** | **有效电平** |
| 1 | DRCK | CLOCK | 单元手动实验状态的时钟来源 | 下降沿打入 |
| 2 | W | K6(M6) | 总线字长：0=16位字操作，1=8位字节操作 |  |
| 3 | X2 | K10(M10) | 源部件定义译码端X2 | 三八译码  八中选一  低电平有效 |
| 4 | X1 | K9(M9) | 源部件定义译码端X1 |
| 5 | X0 | K8(M8) | 源部件定义译码端X0 |
| 6 | SI | K19(M19) | 源寄存器编址：0=CX，1=DX，定义到M19 |  |
| 7 | XP | K7(M7) | 源寄存器奇偶位：0=偶寻址，1=奇寻址 |  |
| 8 | RXW | K18(M18) | 寄存器写使能，本例定义到M18位 | 低电平有效 |
| 9 | DI | K17(K17) | 目标寄存器编址：0=CX，1=DX，定义到M17 |  |
| 10 | OP | K16(M16) | 目标寄存器奇偶位：0=偶寻址，1=奇寻址 |  |

1. **寄存器的读写操作**
   1. 目的通路

当RXW=0时，由DI、OP编码产生目的寄存器地址，详见下表。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **表2.3.5　通用寄存器“手动／搭接”目的编码** | | | | | |
| 目标使能 | 通用寄存器目的编址 | | | | 功能说明 |
| RXW | DI | OP | W | DRCK |
| 0 | 0 | 0 | 0 | ↓ | CX字写 |
| 0 | 0 | 0 | 1 | ↓ | CL(R0)偶字节写 |
| 0 | 0 | 1 | X | ↓ | CH(R1)奇字节写 |
| 0 | 1 | 0 | 0 | ↓ | DX字写 |
| 0 | 1 | 0 | 1 | ↓ | DL(R2)偶字节写 |
| 0 | 1 | 1 | X | ↓ | DH(R2)奇字节写 |

* 1. CX、DX寄存器的写入

通过“I/O输入输出单元”向CX、DX分别置数1122h、3344h，操作步骤如下：

置数

I/O=1122h

数据来源

I/O单元

寄存器CX

(1122h)

K10~K6=10000

按【单拍】按钮

置数

I/O=3344h

寄存器DX

(3344h)

按【单拍】按钮

K18~K16=000

K18~K16=010

* 1. 源通路

当X2 X1 X0=0 0 1时，由SI、XP编码产生源寄存器，详见下表。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **表2.3.6　通用寄存器“手动／搭接”源编码** | | | | | | |
| 源使能 | | | 通用寄存器源编址 | | | 功能说明 |
| X2 | X1 | X0 | SI | XP | W |
| 0 | 0 | 1 | 0 | 0 | 0 | CX字读 |
| 0 | 0 | 1 | CL(R0)偶字节读 |
| 0 | 1 | X | CH(R1)奇字节读 |
| 1 | 0 | 0 | DX字读 |
| 1 | 0 | 1 | DL(R2)偶字节读 |
| 1 | 1 | X | DH(R2)奇字节读 |

* 1. CX、DX寄存器的读出

关闭CX、DX写使能，令RXW=1，按下流程分别读CX、DX。

读CX

数据来源

REG

数据总线

显示CX值

K19 K18=01

K10~K6=00100

读DX

K19=1

数据总线

显示DX值

**2.3.3　准双向I/O口实验**

**一、实验目的**

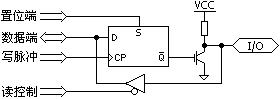
熟悉与了解准双向I/O口的构成原理。

**二、实验要求**

掌握准双向I/O口的输入／输出特性的运用。

**三、实验原理**

Dais-CMX16+向用户提供的是按准双向原理设计的十六位输入/输出I/O口，当该位为“1”时才能用作输入源，上电或复位（手动态按【返回】键），该十六位I/O口被置位（即为“0FFFFh”）。通常情况下，在用作输入的时候就不能再有输出定义。电路结构如图2-3-4所示。该口外接十六位二进制数据开关，适用于外部数据的输入，该口跨接十六个发光二极管，经缓冲驱动四个七段显示，能以二进制和十六进制两种方式显示I/O口的输入输出状态。发光管在高电平“1”时发光点亮。



**图2-3-4 准双向I/O电路**

实验中所用的I/O口数据通路如图2-3-5所示。I/O的输入经2片74LS245缓冲与数据总线相连，I/O口的输出由2片74LS574锁存后输出，锁存器的输入端与数据总线相连。

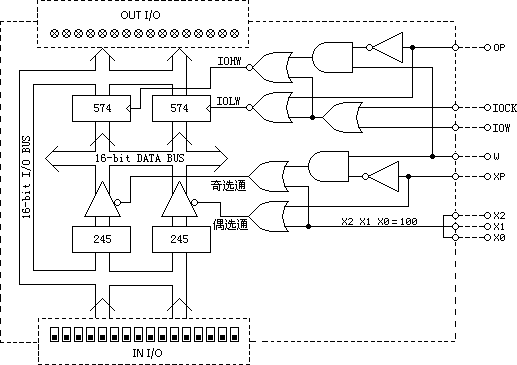


图2-3-5 I/O口数据通路

**四、实验内容**

1. **实验连线**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **连线** | **信号孔** | **接入孔** | **作用** | **有效电平** |
| 1 | IOCK | CLOCK | 单元手动实验状态的时钟来源 | 下降沿打入 |
| 2 | W | K6(M6) | 总线字长：0=16位字操作，1=8位字节操作 |  |
| 3 | X2 | K10(M10) | 源部件定义译码端X2 | 三八译码  八中选一  低电平有效 |
| 4 | X1 | K9(M9) | 源部件定义译码端X1 |
| 5 | X0 | K8(M8) | 源部件定义译码端X0 |
| 6 | XP | K7(M7) | 源奇偶位：0=偶寻址，1=奇寻址 |  |
| 7 | IOW | K17(M17) | I/O写使能，本例定义到M17位 | 低电平有效 |
| 8 | OP | K16(M16) | 目的奇偶位：0=偶寻址，1=奇寻址 |  |
| 9 | RXW | K19(M19) | 寄存器写使能，本例定义到M19位 | 低电平有效 |
| 10 | DI | K18(K18) | 目标寄存器编址：0=CX，1=DX，定义到M18 |  |

1. **搭接方式I/O的寻址定义**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **I/O口源编址** | | | | | |  | **I/O口目的编址** | | | | |
| 源编码 | | | I/O编址 | | 注释 |  | 目的I/O编址 | | | | 注释 |
| X2 | X1 | X0 | W | XP | IOW | W | OP | IOCK |
| 1 | 0 | 0 | 0 | 0 | IO字读 | 0 | 0 | 0 | ↓ | IO字写 |
| 1 | 0 | IOL偶字节读 | 1 | 0 | ↓ | IOL偶字节写 |
| X | 1 | IOH奇字节读 | X | 1 | ↓ | IOH奇字节写 |

1. **I/O口写操作（输出）**

执行OUT I/O,AX 指令，把AX内容送I/O口。

允许I/O写

AX送

数据总线

I/O=AX

(0000)

K10~K6=11000

按【单拍】按钮

I/O开关

S15~S0置1

按【返回】键

K17 K16=00

1. **I/O口读操作（输入）**

执行IN CX，I/O指令，把I/O口内容送CX。

CX=I/O

(0000)

寄存器

写选通

关闭

寄存器写

K10~K6=11000

按【单拍】按钮

I/O送总线

K19=1

K19~K16=0010

1. **I/O口的字节操作（偶输入，奇输出）**

执行OUT IOH，IOL 指令，把S7~S0送S15~S8。

数据总线

(6666)

置数

I/O=FF66h

I/O口

(6666)

按【返回】键

按【单拍】按钮

初始化

I/O=FFFFh

K10~K6=10001

K17 K16=01

**2.3.4　地址总线组成实验**

**一、实验目旳**

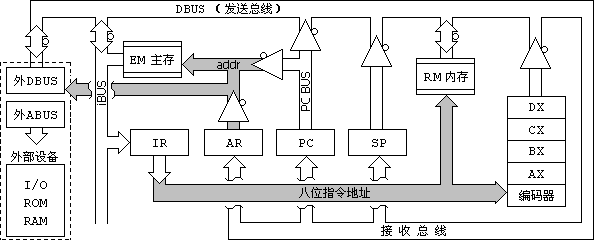
1. 熟悉和了解地址总线的组成结构、地址来源及集合原理。
2. 掌握程序段与数据段的寻址规则及地址部件的运用技巧。

**二、实验要求**

通过地址形成部件实验，建立“段”概念，学会“段”运用。

**三、实验原理**

地址总线的作用是传递地址信息，输出当前数据总线上发送信息的源地址或接收信息的目的地址。如下图所示本系统设有内存与外设两条地址总线，通过PC计数器提供内存（程序存储器）地址，并由地址寄存器AR传递内存（数据存储器）地址与外设地址。另外堆栈寄存器SP亦可视为地址寄存器，它的堆顶指向数据与程序指针存取地址。



**图2-3-6　地址总线组成通路**

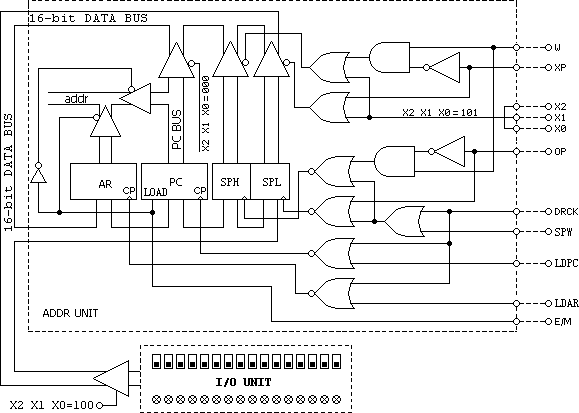
1. **11位程序地址**

如图2-3-6所示，本系统从提高信息存取效率的角度设计主内存地址通路，按现代计算机体系结构中最为典型的分段存取理念合成主存及外设地址总线addr，在指令操作“时段”（取操作码与取操作数），以当前程序指针PC为址，遇主存数据传递“时段”以当前数据指针AR为址。addr地址的合成通路见图2-3-6。其寻址范围为0~7FFh。

1. **16位数据地址**

如图2-3-6所示，本系统数据指针由地址锁存器AR直接提供，当LDAR=0时，在DRCK下降沿把数据总线打入AR。其寻址范围为0~FFFFh，可达64KB。

**四、地址部件电路**



**图2-3-7　地址部件控制电路**

**五、实验内容**

1. **程序计数器实验**

|  |  |  |  |
| --- | --- | --- | --- |
| **表2.3.7　PC程序计数器目标编码** | | | |
| 目标部件定义 | | 按钮 | 功能说明 |
| E/M | IP | DRCK | DRCK下降沿打入 |
| 1 | 1 | X | PC保持 |
| 1 | 0 | ↓ | PC加1 |
| 0 | 0 | ↓ | PC装载 |
| 说明：“↓”表示下降沿有效 | | | |

图2-3-7所示的PC框由3片161构成按字方式寻址的11位PC计数器，计数器的输入端与总线相连构成置数通路，计数器的输出端途经三态门缓冲分离为两条通路，其一与总线相连构成可读通路，其二与地址寄存器（数据）集合组成主存EM地址总线。它的清零端由中央外理器单元直控，上电时PC计数器自动淸零，实验中按复位钮亦可实现计数器的手动淸零。

手控状态，本实验由表2.6.1定义的目的编码控制PC计数器的预置与加1操作，并以准双向I/O部件的S10~S0为计数器预置源。当IP=0时按单拍按钮，遇E/M=0在脉冲下降沿把S10~S0的内容装入PC计数器；遇E/M=1在脉冲下降沿PC计数器加1。

PC计数器的读出操作由表2.3.7所列的源编码表定义。

**1）实验连线**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **连线** | **信号孔** | **接入孔** | **作用** | **有效电平** |
| 1 | DRCK | CLOCK | 单元手动实验状态的时钟来源 | 下降沿打入 |
| 2 | W | K6(M6) | 总线字长：0=16位字操作，1=8位字节操作 |  |
| 3 | XP | K7(M7) | 源奇偶位：0=偶寻址，1=奇寻址 |  |
| 4 | OP | K16(M16) | 目的奇偶位：0=偶寻址，1=奇寻址 |  |
| 5 | X2 | K10(M10) | 源部件定义译码端X2 | 三八译码  八中选一  低电平有效 |
| 6 | X1 | K9(M9) | 源部件定义译码端X1 |
| 7 | X0 | K8(M8) | 源部件定义译码端X0 |
| 8 | LDPC | K22(M22) | PC程序计数器刷新位 | 低电平有效 |
| 9 | E/M | K23(M23) | 当IP有效时，E/M：0=PC装载，1=PC加1 |  |
| 10 | SPW | K20(M20) | SP堆栈指针写使能 | 低电平有效 |
| 11 | LDAR | K19(M19) | AR地址寄存器写使能 | 低电平有效 |

**2）程序计数器PC的写入、读出与加1**

* 1. PC程序计数器的写入

通过“I/O单元”开关向程序计数器PC置数，操作步骤如下：

置数

I/O=0100h

数据来源

I/O单元

程序计数器

PC=0100h

K10~K6=10000

按【单拍】按钮

K23 K22=00

关闭PC装载

读PC

DBUS=0100

K10~K6=00000

K23 K22=11

* 1. PC程序计数器的读出

在PC置数操作完成后，按上流程中后两步的要求，关闭PC写使能（K23、K22=11），打开PC输出三态门（K10~K6=00000），数据总线单元应显示的PC指针为0100h。

* 1. PC程序计数器加1

在保持PC置数与读出流程的状态下，令K22=0，按【单拍】按钮，在DRCK节拍的下降沿PC计数器加1并送数据总线，PC程序计数器和数据总线单元的显示器应显示0101h。继续按【单拍】按钮，观察PC与数据总线的内容。

**2、地址寄存器实验**

图2-3-7所示的AR框由2片74LS574锁存器构成按字方式寻址的16位数据指针，锁存器的输入端与总线相连构成置数通路，锁存器的输出端途经三态门缓冲分离与PC计数器集合组成主存地址总线。它的清零端由中央外理器单元直控,上电时锁存器自动淸零，手动实验中按【返回】键亦可实现锁存器的手动淸零。按通用计算机设计规范的要求，把数据指针AR定义为字写入寄存器，运用中局限于字写，字节写会引发数据指针的错误侵入，因此在数据指针AR的操作过程中并非不支持而是不允字节写。

**地址寄存器AR打入**

在手控/搭接态，数据指针AR由W、LDAR及DRCK（CP脉冲）三信号组合控制地址的置数操作。本实验以总线上准双向I/O部件的S15~S0为置数源。当W=0、LDAR=0时按【单拍】钮，在脉冲下降沿把S15~S0的内容装入地址锁存器AR。操作步骤如下：

置数

I/O=1234h

数据来源

I/O单元

AR显示

1234

K10~K6=10000

按【单拍】按钮

置数

I/O=5678h

AR显示

5678

按【单拍】按钮

K19=0

**3、堆栈寄存器实验**

图2-3-7所示的SP框由2片74LS574锁存器构成16位堆栈指针，锁存器的输入端与总线相连构成存数通路，锁存器的输出端途经三态门隔离与总线相连构成取数通路。它按先进后出的原则存放需要保留的数据信息与地址信息，在调用中断等突发事件处理中SP指针以间址方式把当前程序指针存入SP-2单元，遇返回指令SP又把栈项所指单元的内容装入程序计数器，然后SP+2退至原始位置。

在手控/搭接态,堆栈指针SP由W、SPW及DRCK（CP脉冲）三信号组合控制栈指针的置数操作。本实验以总线上准双向I/O部件的S15~S0为置数源。当W=0、SPW=0时按单柏钮，在脉冲下降把S15~S0的内容装入SP。

堆栈指针SP的读出操作由P8页表2.2所列的源编码表定义。

**1）堆栈指针SP打入**

拨动“I/O输入输出单元”开关向堆栈指针SP置数，具体操作步骤如下：

置数

I/O=0510h

数据来源

I/O单元

堆栈指针

SP=0510h

K10~K6=10000

按【单拍】按钮

K20 K16=00

**2）堆栈指针SP读出**

关闭SP写使能，令SPW=1 按下流程完成SP送总线。

数据来源

SP

数据总线

显示SP值

K10~K6=10100

写关闭

K20=1

**2.3.5　十六位数据总线实验**

一、实验目旳

1. 熟悉和了解总线的数据通路、双向互递原理及寻址方式与运用规则。
2. 掌握十六位数据总线中“字”与“字节”操作方法及源与目的奇偶效应。

**二、实验要求**

通过总线的数据传递实验，建立“奇偶”概念，领会字寻址中对字节操作的动态定义。

**三、实验原理**

系统数据总线作为计算机传递信息的通道是连接各个功能部件的纽带，在计算机中起着至关重要的作用。模型机的工作过程就是计算机各个功能部件之间的信息，通过数据总线不断有序流动的过程。

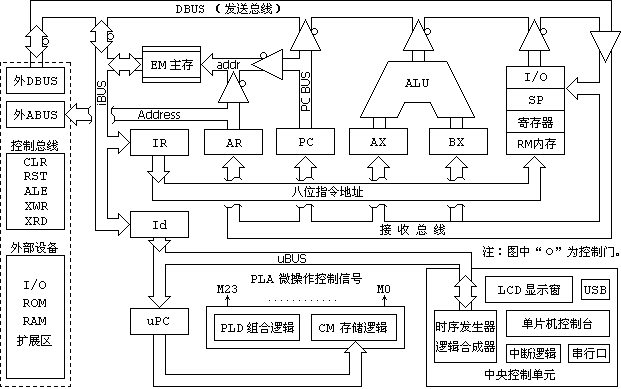
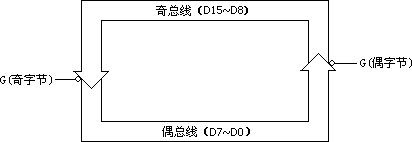


图2-3-8 系统体系结构图

1. **字与字节体系**

本系统总线宽度为十六位，设有字长控位“W”，当W=0，由源寻址的奇偶性决定当前总线宽度，遇源址为偶时其字长宽度为十六位；当源址为奇或W=1时，字长宽度为八位，形成图2-3-9所示的奇（八位）与偶（八位）互通的字节总线。



**图2-3-9 奇偶互通字节总线体系结构图**

1. **源奇偶的运用**

图2-3-9所示，我们按原理计算机的设计规范，以字节为基准把十六位数据总线划分奇与偶俩路八位总线，其中“D15~D8”称为“奇总线”，“D7~D0”称为“偶总线”；在字节传递中由于总线的互通，形成“奇送偶”或“偶送奇”的八位字节总线，其使能端定义为低电平选通，逻辑表达式为：

G（偶字节）= !W（字长）# XP（源奇偶）

G（奇字节）= !XP（源奇偶）

从上俩式可知，图2-3-9所示的奇偶总线由字长“W”和源奇偶“XP”动态呈现以下三状态：

⑴在W=0时遇XP=0，由于G（偶字节）与G（奇字节）处隔离态“1”，形成“D15~D0”十六位字总线源。

⑵在W=1时遇XP=0，由于G（偶字节）为“0”，G（奇字节）“1”，形成“偶送奇‘的八位字节总线源。

⑶在XP=1时，由于G（奇字节）为“0”，G（偶字节）为“1”，无条件形成“奇送偶”的八位字节总线源。

1. **目的奇偶的运用**

在目的寻址中亦由字长控位“W”与目的地址的奇偶性动态定义当前目的字长。在W=0又遇目的址为偶时，其目的传递为字操作，否则均为字节传递，其逻辑表达式为：

!O（偶字节）= !OP（目的奇偶）

!O（奇字节）= !W（字长）# OP（目的奇偶）

上述俩式表明，由字长“W”和目的址奇偶“OP”动态产生以下三种目的寻址操作

⑴在W=0时遇OP=0，由于O（偶字节）与O（奇字节）均为“0”，执行以当前目的偶址为目标的字传递。

⑵在W=1时遇OP=0，由于O（偶字节）=“0”、O（奇字节）=“1”,执行以当前目的偶址为目标的字节传递。

⑶在OP=1时，由于O（奇字节）=“0”、O（偶字节）=“1”，无条件执行以当前目的奇址为目标的字节传递。

1. **数据传递规则**

系统在十六位原理计算机的字操作中动态地融入了字节操作的过程，其源奇偶映射总线宽度，而目的奇偶则制约传递长度。系统在十六位原理计算机的字节操作中运用总线互联机制，以源址的奇偶性形成“奇递偶”或“偶递奇”两者互通的八位字节总线。

|  |  |  |  |
| --- | --- | --- | --- |
| **表2.3.7　十六位总线传递规则** | | | |
| 总线规则 | | | 功能说明 |
| W | XP | OP |
| 0 | 0 | 0 | 字传递（十六位传递） |
| 1 | 0 | 0 | 偶送偶（低位送低位） |
| 1 | 0 | 1 | 偶送奇（低位送高位） |
| X | 1 | 0 | 奇送偶（高位送低位） |
| X | 1 | 1 | 奇送奇（高位送高位） |

说明：上表中“XP”与“OP”仅为原理计算机特定的专用寄存器奇偶标志，适用于AX、BX、SP及I/O的寻址场合；在存储器寻址中应以地址线“A0”为奇偶；在通用寄存器寻址中应从指令格式中所定义的“源与目的”字段动态索取奇偶标志。

**四、数据传递实验**

1. **实验连线**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **连线** | **信号孔** | **接入孔** | **作用** | **有效电平** |
| 1 | DRCK | CLOCK | 单元手动实验状态的时钟来源 | 下降沿打入 |
| 2 | W | K6(M6) | 总线字长：0=16位字操作，1=8位字节操作 |  |
| 3 | XP | K7(M7) | 源部件奇偶标志：0=偶寻址，1=奇寻址 |  |
| 4 | X2 | K10(M10) | 源部件定义译码端X2 | 三八译码  八中选一  低电平有效 |
| 5 | X1 | K9(M9) | 源部件定义译码端X1 |
| 6 | X0 | K8(M8) | 源部件定义译码端X0 |
| 7 | OP | K16(M16) | 目标部件奇偶标志：0=偶寻址，1=奇寻址 |  |
| 8 | AXW | K17(M17) | AX运算寄存器写使能，本例定义到M17位 | 低电平有效 |

1. **十六位数据传送（字传递）**

设置数据来源为I/O单元（X2 X1 X0=100），总线规则设为字传递（W XP OP=000），打开AX写使能（AXW(M17)=0），拨动“I/O输入输出单元”十六位数据开关，按【单拍】按钮，将I/O单元内容通过数据总线传递到AX寄存器，操作步骤如下：

置数

I/O=1234h

数据来源

I/O单元

运算寄存器

AX=1234h

K10~K6=10000

按【单拍】按钮

K17 K16=00

1. **低位到低位（偶送偶）**

设置数据来源为I/O单元（X2 X1 X0=100），总线规则设为偶送偶（W XP OP=100），打开AX写使能（AXW(M17)=0），拨动“I/O输入输出单元”十六位数据开关，按【单拍】按钮，将I/O偶单元内容通过数据总线传递到AL寄存器，操作步骤如下：

置数

I/O=XX55h

数据来源

I/O单元

运算寄存器

AX=XX55h

K10~K6=10001

按【单拍】按钮

K17 K16=00

1. **低位到高位（偶送奇）**

设置数据来源为I/O单元（X2 X1 X0=100），总线规则设为偶送奇（W XP OP=101），打开AX写使能（AXW(M17)=0），拨动“I/O输入输出单元”十六位数据开关，按【单拍】按钮，将I/O偶单元内容通过数据总线传递到AH寄存器，操作步骤如下：

置数

I/O=XX55h

数据来源

I/O单元

运算寄存器

AX=55XXh

K10~K6=10001

按【单拍】按钮

K17 K16=01

1. **高位到低位（奇送偶）**

设置数据来源为I/O单元（X2 X1 X0=100），总线规则设为奇送偶（W XP OP=X10），打开AX写使能（AXW(M17)=0），拨动“I/O输入输出单元”十六位数据开关，按【单拍】按钮，将I/O奇单元内容通过数据总线传递到AL寄存器，操作步骤如下：

置数

I/O=AAXXh

数据来源

I/O单元

运算寄存器

AX=XXAAh

K10~K6=1001X

按【单拍】按钮

K17 K16=00

1. **高位到高位（奇送奇）**

设置数据来源为I/O单元（X2 X1 X0=100），总线规则设为奇送奇（W XP OP=X11），打开AX写使能（AXW(M17)=0），拨动“I/O输入输出单元”十六位数据开关，按【单拍】按钮，将I/O奇单元内容通过数据总线传递到AH寄存器，操作步骤如下：

置数

I/O=AAXXh

数据来源

I/O单元

运算寄存器

AX=AAXXh

K10~K6=1001X

按【单拍】按钮

K17 K16=01

**2.3.6 存储器读写实验**

**一、实验目的**

熟悉和了解存储器组织与总线组成的数据通路。

**二、实验要求**

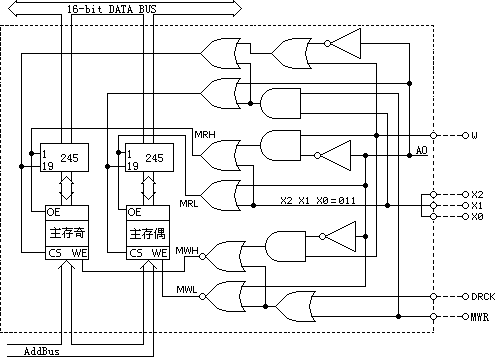
按照实验步骤完成实验项目，掌握存储部件在原理计算机中的运用。

**三、实验原理**

存储器是计算机的存储部件，用于存放程序和数据。存储器是计算机信息存储的核心，是计算机必不可少的部件之一，计算机就是按存放在存储器中的程序自动有序不间断地进行工作。

本系统从提高存储器存储信息效率的角度设计数据通路，按现代计算机中最为典型的分段存储理念把存储器组织划分为程序段、数据段等，由此派生了数据总线（DBus）、指令总线（IBus）、微总线（μBus）等与现代计算机设计规范相吻合的实验环境。

实验所用的存储器电路原理如图2-3-10所示，该存储器组织由二片6116构成具有奇偶概念的十六位信息存储体系，该存储体系AddBus由PC指针和AR指针分时提供，E/M控位为“1”时选通PC，反之选通AR。该存储体系可随机定义总线宽度，动态变更总线结构，把我们的教学实验提高到能与现代计算机设计规范相匹配与接轨的层面。



**图2-3-10 存储器数据通路**

**四、存储器分类与寻址**

1. **存储器组织分类表**

本系统主存储器由两个部分组成，详见下表：

|  |  |  |
| --- | --- | --- |
| 分类 | 存储容量 | 寻址范围 |
| 程序段 | 2K | 0~7FFh |
| 数据段 | 2K | 0~7FFh |

1. **程数存储器源与目的寻址**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **程序段与数据段源寻址** | | | | | | |  | **程序段与数据段目的寻址** | | | | |
| 源使能 | | | 源编址 | | | 注释 |  | 目的编址 | | | | 注释 |
| X2 | X1 | X0 | E/M | W | A0 | MWR | E/M | W | A0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 程序段字读 | 0 | 1 | 0 | 0 | 程序段字写 |
| 1 | 0 | 程序段偶读 | 1 | 0 | 程序段偶写 |
| X | 1 | 程序段奇读 | X | 1 | 程序段奇写 |
| 0 | 0 | 0 | 数据段字读 | 0 | 0 | 0 | 数据段字写 |
| 1 | 0 | 数据段偶读 | 1 | 0 | 数据段偶写 |
| X | 1 | 数据段奇读 | X | 1 | 数据段奇写 |
|  | | | | | | |  | 注：在【单拍】按钮下降沿写入 | | | | |

**五、实验内容**

1. **实验连线**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **连线** | **信号孔** | **接入孔** | **作用** | **有效电平** |
| 1 | DRCK | CLOCK | 单元手动实验状态的时钟来源 | 下降沿打入 |
| 2 | W | K6(M6) | 总线字长：0=16位字操作，1=8位字节操作 |  |
| 3 | XP | K7(M7) | 源部件奇偶标志：0=偶寻址，1=奇寻址 |  |
| 4 | X2 | K10(M10) | 源部件定义译码端X2 | 三八译码  八中选一  低电平有效 |
| 5 | X1 | K9(M9) | 源部件定义译码端X1 |
| 6 | X0 | K8(M8) | 源部件定义译码端X0 |
| 7 | E/M | K23(M23) | 存储器地址段定义：0=AR指针，1=PC指针 |  |
| 8 | MWR | K21(M21) | 存储器写使能：0=存储器写 |  |
| 9 | LDAR | K17(M17) | AR地址寄存器写使能 | 低电平有效 |
| 10 | LDPC | K22(M22) | PC装载与PC+1 | 低电平有效 |

1. **存储器数据段读写操作**
   1. 数据段写操作（字）

在进行数据存储器字操作时，地址线A0必须为0（偶地址）。向数据段的0~0005h存储单元写入112233445566一串数据，以0址单元写入数据1122h为例表述操作流程。

置地址

S15~S0=0

I/O=0000h

AR=0000

K10~K6=10000

K23 K17=00

按【单拍】按钮

置数据

I/O=1122h

存储器写入

(1122h)

K21=0

按【单拍】按钮

K17=1

关存储器写

令K21=1

按照上述操作流程完成0002~0005h单元分别写入33445566的操作。

* 1. 数据段读操作（字）

依次读出数据段0~0005h单元的内容，这里以0址单元读出为例阐述操作流程。

置数

I/O=0000h

数据来源

I/O单元

AR地址写入

(0000h)

K10~K6=10000

K23=0

K17=0

按【单拍】按钮

关闭AR写

令K17=1

读存储器RAM→总线

K10~K8=011

K6=0

执行上述流程总线单元应显示1122h，若正确可按上述流程读出0002~0005h单元的内容。

1. **存储器程序段读写操作**
2. **程序段字节写操作**
   1. 计算机规范的取指操作均以字节为单位。所以本实验以字节操作方式展开。程序段写入必须从定义地址入手，然后再进入程序存储器的写入。
   2. PC指针是带预置加法计数器，因此在输入起始地址后一旦后续地址为PC+1的话就不需重装PC，用PC+1指令完成下续地址的读写操作。
   3. PC地址装载写入与PC+1写入流程

置地址

S15~S0=0

I/O=0000h

I/O→PC

PC=0000

K10~K6=10000

K23 K22=00

按【单拍】按钮

I/O=3412h

12→[PC]

PC+1

K22 K21=01

按【单拍】按钮

K23~K21 K6=1101

按【单拍】按钮

存储器写

34→[PC]

K22 K21 K7=101

按【单拍】按钮

按照上述PC装载写入与PC+1写入的流程分别对0000~0005写入12345678h。

1. **程序段字节读操作**

PC地址装载读出及PC+1读出流程

按【单拍】按钮

置地址

S15~S0=0

I/O=0000h

I/O→PC

PC=0000

K10~K6=10000

K23 K22=00

按【单拍】按钮

选通存储器

[PC]→总线

存储器读

[PC+1]→总线

K22=0

按【单拍】按钮

K10~K6=01101

K23 K22=11

存储器读

[PC+1]→总线

按照上述PC装载读出与PC+1读出的流程分别读出0000~0005h单元内容，应为12345678h。

**2.3.7　指令总线运用实验**

**一、实验目旳**

1. 熟悉和了解指令总线的数据通路与构成途径。
2. 掌握指令部件的“取指”规则及地址段运用技巧。

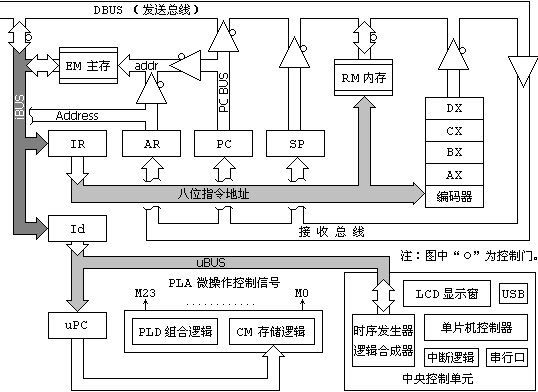
**二、实验要求**

通过指令部件实验，建立“指令暂存”概念，领会“操作码段”和“操作数段”的用途。

**三、实验原理**

指令总线（IBUS）作为传递指令信息的通道是连接指令部件的钮带，如图2-7-1所示，在取指操中指令信息由主存流向指令寄存器IR和指令译码器ID，若取操作数亦可经三态门流向数据总线，指令总线（IBUS）也是主存及IR与数据总线之间的互递通路，在主存读写周期与数据总线双向交換信息，在通用寄存器或内存寻址操作中透过数据总线单向传递地址信息。

**四、数据通路**



**图2-3-11 十六位指令总线数据通路**

**五、实验项目**

1. **指令寄存器IR**

图2-3-11所示的IR框由2片74LS574锁存器构成16位指令寄存器，主要用于存放指令的操作码与操作数，它的输入端经指令总线（IBUS）分别与主存和数据总线构成取数通路。其锁存输出端编码产生通用寄存器地址，并指定由IR15~IR8提供内存地址。

1. **指令编译器ID**

图2-3-11所示的ID框由指令编译电路（CPLD）构成11位微地址寄存器，主要用于存放指令排序器所定义的指令起始微地址（亦可称为指令的微程序入口地址）。ID的输入端经指令总线（IBUS）分别与主存和数据总线构成取数通路。其三态输出端经微总线（uBUS）单向流入微程序计数器的输入端口，在时序电路的控制下形成与当前指令相对应的微程序入口地址。

**指令排序格式**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 微总线 | ud10 | Ud9 | Ud8 | ud7 | ud6 | ud5 | ud4 | ud3 | ud2 | ud1 | ud0 |
| 指令总线 | 1 | 1 | Id7 | id6 | id5 | id4 | id3 | id2 | id1 | id0 | 0 |

由指令排序格式可知，本装置微控制器在“取指”时按“字节”排序，指令系统微程序入口地址的寻范围为600~7FFh，最多可支撑256条指令的微运行，其容纳率达通用计算机控制器的设计水准。控制器支持指令的变长编码，在模型机的设计与实现中，可根据指令的容纳率动态编制与确定机器指令中操作码的长度（简称指令段）。指令段通常存放在机器指令起始字节的高端。在取指时用“与逻辑”保留指令段屏敝地址段，例如设计一个八条以下的指令系统模型机，它的指令段长度为三位，存放在机器指令起始字节的“D7~D5”位置，形成下表：

**三操作码指令排序格式**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 微总线 | ud10 | Ud9 | ud8 | ud7 | ud6 | Ud5 | ud4 | ud3 | ud2 | ud1 | ud0 |
| 指令总线 | 1 | 1 | id7 | id6 | id5 | 0 | 0 | 0 | 0 | 0 | 0 |

上表可产生7C0h、780h、740h、700h、6C0h、680h、640h、600h共八个微程序入口地址，分别对应机器指令的E0h、C0h、B0h、80h、60h、40h、20h、00h。

这里仅阐述了指令系统起始微入口的形成途径与排序格式，它的执行涉及微控制器原理，我们按排在微控制器实验中进行。

1. **IR与ID操作方式**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表2.9.3　IR指令寄存器写入** | | | | |
| IR写 | | 微控制位 | | 功能说明 |
| IR | MWR | OP | W | DRCK下降沿打入 |
| 0 | 0 | 0 | 0 | IR字写 |
| 0 | 0 | 0 | 1 | IRL偶字节写 |
| 0 | 0 | 1 | X | IRH奇字节写 |

|  |  |  |  |
| --- | --- | --- | --- |
| **表2.9.4　IR指令寄存器取指** | | | |
| 取指 | | 按钮 | 功能说明 |
| IR | MWR | DRCK |
| 0 | 1 | ↓ | 打操作码，散转 |
| 说明：“↓”表示下降沿有效 | | | |

**1）实验连线**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **连线** | **信号孔** | **接入孔** | **作用** | **有效电平** |
| 1 | DRCK | CLOCK | 单元手动实验状态的时钟来源 | 下降沿打入 |
| 2 | W | K6(M6) | 总线字长：0=16位字操作，1=8位字节操作 |  |
| 3 | XP | K7(M7) | 源部件奇偶标志：0=偶寻址，1=奇寻址 |  |
| 4 | OP | K16(M16) | 目标部件奇偶标志：0=偶寻址，1=奇寻址 |  |
| 5 | X2 | K10(M10) | 源部件定义译码端X2 | 三八译码  八中选一  低电平有效 |
| 6 | X1 | K9(M9) | 源部件定义译码端X1 |
| 7 | X0 | K8(M8) | 源部件定义译码端X0 |
| 8 | MWR | K21(M21) | EM程序存储器写使能，本例定义到K21位 | 低电平有效 |
| 9 | LDPC | K22(M22) | PC程序计数器更新位，本例定义到K22 | 低电平有效 |
| 10 | E/M | K23(M23) | 当IP有效时，E/M：0=PC装载，1=PC加1 |  |
| 11 | LDIR | K2(M2) | IR指令寄存器写使能，本例定义到K2位 | 低电平有效 |

**2）指令寄存器IR字打入**

拨动“I/O输入输出单元”开关向指令寄存器IR置数，具体操作步骤如下：

置数

I/O=1234h

数据来源

I/O单元

指令寄存器

IR=1234h

K10~K6=10000

K21 K16=00

按【单拍】按钮

令K2=0

令PC=0

按【返回】键

**3）指令寄存器IR偶字节打入**

拨动“I/O输入输出单元”开关向指令寄存器IR偶址置数，操作步骤如下：

置数

I/O=XX56h

数据来源

I/O单元

指令寄存器

IR=XX56h

K10~K6=10001

K21 K16=00

按【单拍】按钮

令K2=0

令PC=0

按【返回】键

**4）指令寄存器IR奇字节打入**

拨动“I/O输入输出单元”开关向指令寄存器IR奇址置数，操作步骤如下：

令PC=0

按【返回】键

置数

I/O=XX78h

数据来源

I/O单元

指令寄存器

IR=78XXh

K10~K6=10001

K21 K16=01

按【单拍】按钮

令K2=0

**5）指令寄存器IR取指与散转**

以PC为地址，向程序存储器EM存放指令码，通过内部指令总线将指令码打入指令寄存器，实现指令码的散转。

置数

I/O=0000h

数据来源

I/O单元

程序计数器

PC=0000h

K10~K8=10000

K16=0

K23 K22=00

按【单拍】按钮

K23 K22=11

打入PC地址

置数

I/O=0020h

存储器

(20XXh)

K6 K21=10

按【单拍】按钮

K21=1

写入存储器

指令码写入成功后，关闭所有微控制位（K23~K0=全“1”），令IR控制位为0，按【单拍】按钮，将指令码20h打入IR指令寄存器，微地址uPC散转为0640h：

指令码20h

(00100000)

左移1位

(01000000)

加基地址0600h

微地址

0640h