

广州大学学生实验报告

开课学院及实验室：计算机科学与工程实验室

2019 年 5 月 8 日

学院	计算机科学与 网络工程学院	年级/专 业/班	软件 171	姓名	谢金宏	学号	1706300001
实验课 程名称	计算机组成与系统结构实验课					成绩	
实验项 目名称	存储器与总线传输实验					指导老师	古鹏

实验二 存储器与总线传输实验

一、实验目的

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。
3. 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

1. 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。
2. 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

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。

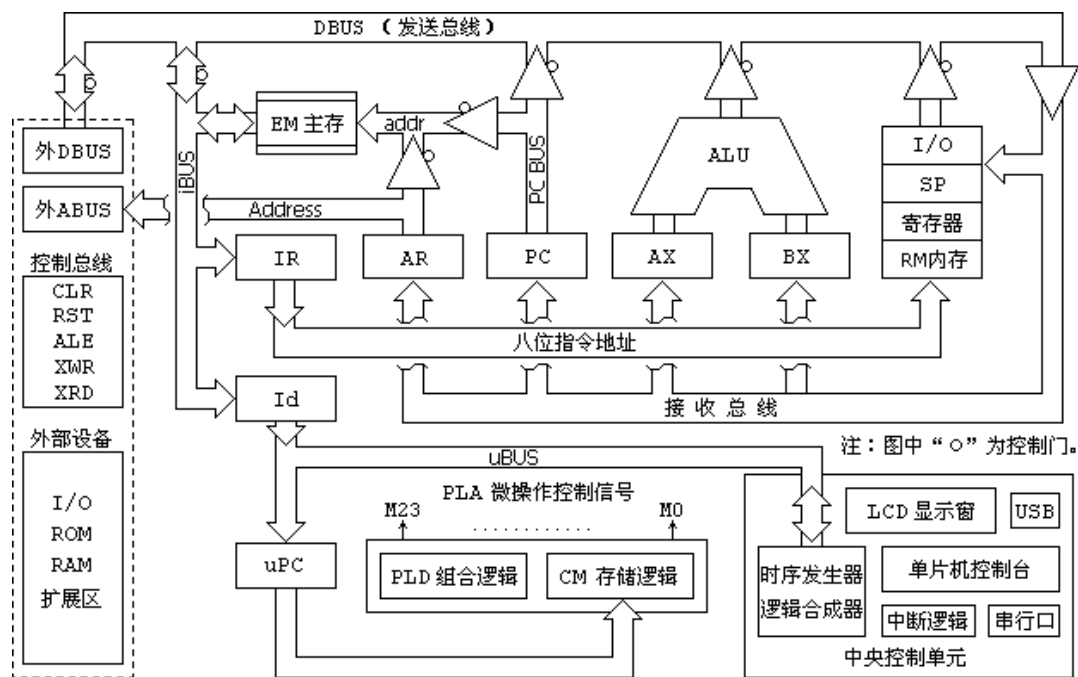


图 1 系统体系结构图

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。

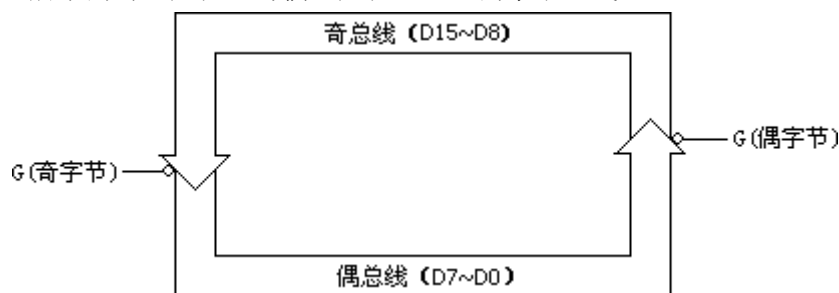


图 2 奇偶互通字节总线体系结构图

2. 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：

$$G(\text{偶字节}) = !W(\text{字长}) \# XP(\text{源奇偶})$$

$$G(\text{奇字节}) = !XP(\text{源奇偶})$$

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：

(1) 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。

(2) 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。

(3)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。

3. 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:

$$!O(\text{偶字节}) = !OP(\text{目的奇偶})$$

$$!O(\text{奇字节}) = !W(\text{字长}) \# OP(\text{目的奇偶})$$

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

(1)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。

(2)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。

(3)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。

4. 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 1 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。

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 3 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。

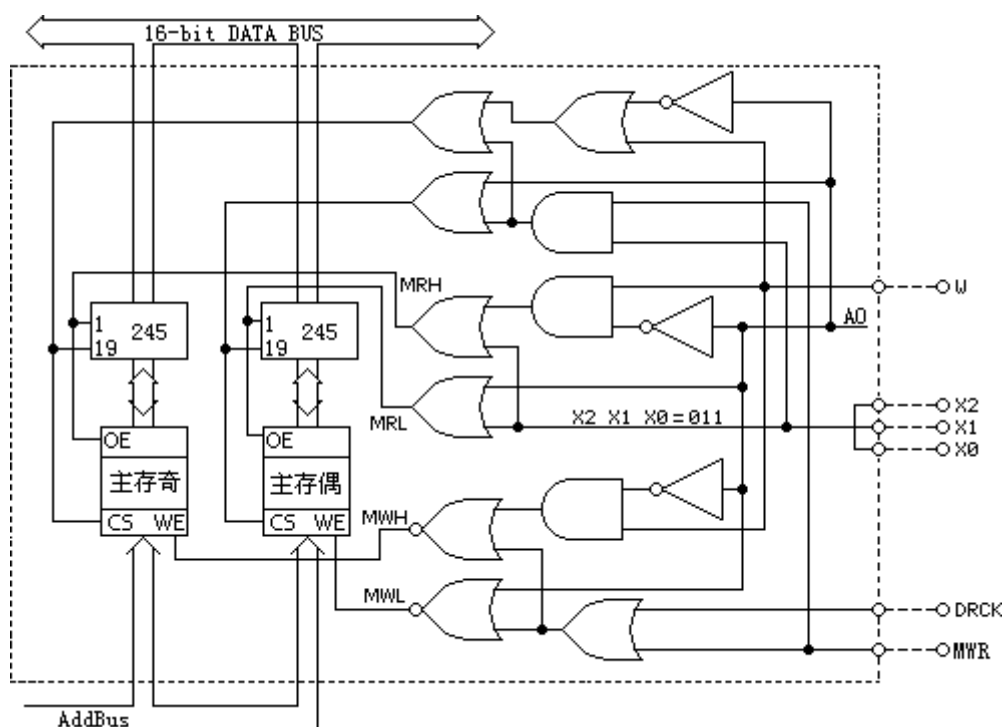


图 3 存储器数据通路

四、实验连线

1. 数据传递实验

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

2. 数据传递实验

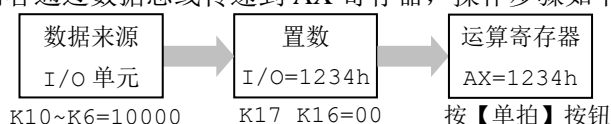
连线	信号孔	接入孔	作用	有效电平
1	DRCK	CLOCK	单元手动实验状态的时钟来源	下降沿打入
2	W	K6(M6)	总线字长：0=16 位字操作，1=8 位字节操作	
3	XP	K7(M7)	源部件奇偶标志：0=偶寻址，1=奇寻址	
4	X2	K10(M10)	源部件定义译码端 X2	三八译码 八中选一 低电平有效
5	X1	K9(M9)	源部件定义译码端 X1	
6	X0	K8(M8)	源部件定义译码端 X0	
7	E/M	K23(M23)	存储器地址段定义：0=AR 指针，1=PC 指针	

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

五、实验内容

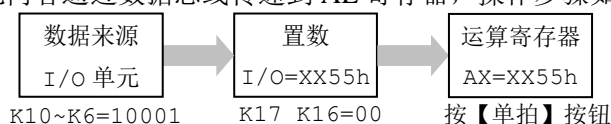
1. 十六位数据传送（字传递）

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：



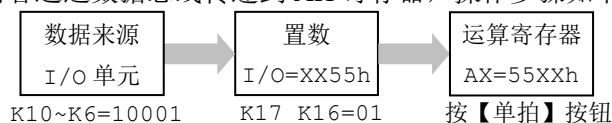
2. 低位到低位（偶送偶）

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：



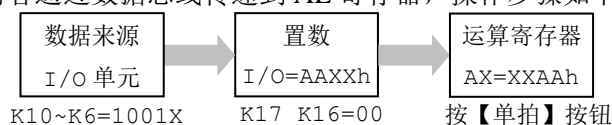
3. 低位到高位（偶送奇）

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：



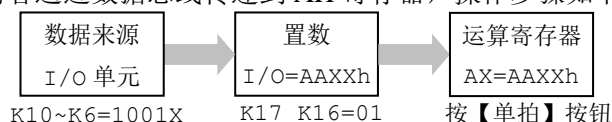
4. 高位到低位（奇送偶）

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：



5. 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：



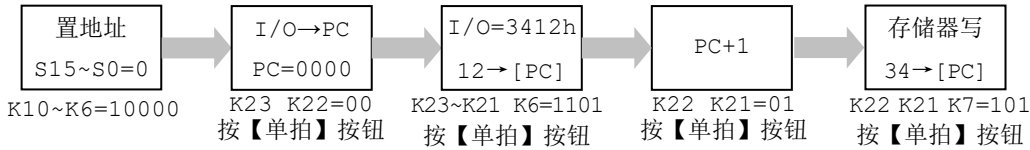
6. 存储器程序段读写操作

(1) 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 指令完成后续地址的读写操作。

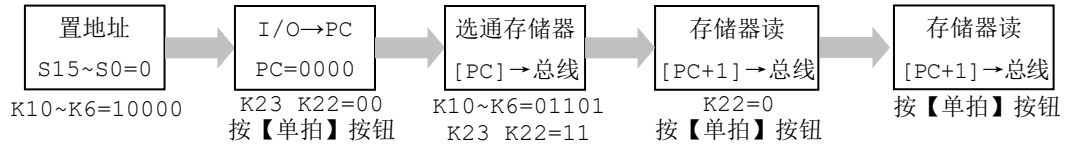
③ PC 地址装载写入与 PC+1 写入流程



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。

(2) 程序段字节读操作

PC 地址装载读出及 PC+1 读出流程



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。

六、实验思考

部分完成了实验要求的内容。基本掌握了在总线中传递数据和将数据写入读出通用寄存器的方法。通过此次实验和先前实验的学习，我基本掌握了图灵机模型中输入输出设备、ALU 和存储设备的使用方法。预计结束下一次实验对控制器的学习之后，我就能完整地创造一台符合图灵模型的“计算机”了。

七、问题及体会

在借助集成开发环境尝试改变指令体系和微操作时，发现自己对指令体系的更改并没有起效果。单拍运行时，根据集成开发环境的反馈，机器似乎跳过了自己创造的指令。猜测是自己的操作出现错误，但时间不足，不能定位具体的错误位置。希望下次实验能够找出原理。目前对实验的拓展练习还毫无头绪。