|  |  |  |  |
| --- | --- | --- | --- |
| **学生学号** | 0121902960525 | **实验课成绩** |  |

**武汉理工大学**

**学 生 实 验 报 告 书**

**实验课程名称** 计算机组成与体系结构

**开 课 学 院 计算机科学与技术学院**

**指导老师姓名 陈建军**

**学 生 姓 名 周航**

**学生专业班级 物联网1902班**

20 — 20 学年 第 学期

实验课程名称： 计算机组成与体系结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验项目名称 | 系统总线和具有基本输入输出功能的总线接口实验 | | | 实验成绩 |  |
| 实验者 | 周航 | 专业班级 | 物联网1902班 | 组别 |  |
| 同组者 |  | | | 实验日期 | 年 月 日 |
| 第一部分：实验分析与设计（可加页）   1. 实验内容描述（问题域描述）     （1）理解总线的概念及其特性。  （2）掌握控制总线的功能和应用。   1. 实验基本原理与设计（包括实验方案设计，实验手段的确定，试验步骤等，用硬件逻辑或者算法描述）   由于存储器和输入、输出设备最终是要挂接到外部总线上，所以需要外部总线提供数据信  号、地址信号以及控制信号。在该实验平台中，外部总线分为数据总线、地址总线、和控制总  线，分别为外设提供上述信号。外部总线和 CPU 内总线之间通过三态门连接，同时实现了内外总线的分离和对于数据流向的控制。地址总线可以为外部设备提供地址信号和片选信号。由地址总线的高位进行译码，系统的 I/O 地址译码原理见图 4-1-1（在地址总线单元）。由于使用 A6、A7 进行译码， I/O 地址空间被分为四个区，如表 4-1-1 所示：      为了实现对于 MEM 和外设的读写操作，还需要一个读写控制逻辑，使得 CPU 能控制 MEM和 I/O 设备的读写，实验中的读写控制逻辑如图 4-1-2 所示，由于 T3 的参与，可以保证写脉宽与 T3 一致，T3 由时序单元的 TS3 给出（时序单元的介绍见附录 2）。IOM 用来选择是对 I/O 设备还是对 MEM 进行读写操作，IOM=1 时对 I/O 设备进行读写操作，IOM=0 时对 MEM 进行读写操作。RD=1 时为读，WR=1 时为写。      在理解读写控制逻辑的基础上我们设计一个总线传输的实验。实验所用总线传输实验框图  如图 4-1-3 所示，它将几种不同的设备挂至总线上，有存储器、输入设备、输出设备、寄存器。  这些设备都需要有三态输出控制，按照传输要求恰当有序的控制它们，就可实现总线信息传输。    三、主要仪器设备及耗材  PC机一台，TD-CMA实验系统一套 | | | | | |

|  |
| --- |
| 第二部分：实验调试与结果分析（可加页）   1. 调试过程（包括调试方法描述、实验数据记录，实验现象记录，实验过程发现的问题等）   1．读写控制逻辑设计实验。  （1）按照图 4-1-4 实验接线图进行连线。    （2）具体操作步骤图示如下：  首先将时序与操作台单元的开关 KK1、KK3 置为‘运行’档，开关 KK2 置为‘单拍’档，  按动 CON 单元的总清按钮 CLR，并执行下述操作。  ① 对 MEM 进行读操作（WR=0，RD=1，IOM=0），此时 E0 灭,表示存储器读功能信号有效。  ② 对 MEM 进行写操作（WR=1，RD=0，IOM=0），连续按动开关 ST，观察扩展单元数据指示灯，指示灯显示为 T3 时刻时，E1 灭,表示存储器写功能信号有效。  ③ 对 I/O 进行读操作（WR=0，RD=1，IOM=1），此时 E2 灭,表示 I/O 读功能信号有效。  ④ 对 I/O 进行写操作（WR=1，RD=0，IOM=1），连续按动开关 ST，观察扩展单元数据指示灯，指示灯显示为 T3 时刻时，E3 灭,表示 I/O 写功能信号有效。  2．基本输入输出功能的总线接口实验。  （1）根据挂在总线上的几个基本部件，设计一个简单的流程：  ① 输入设备将一个数打入 R0 寄存器。  ② 输入设备将另一个数打入地址寄存器。  ③ 将 R0 寄存器中的数写入到当前地址的存储器中。  ④ 将当前地址的存储器中的数用 LED 数码管显示。  （2）按照图 4-1-5 实验接线图进行连线。  （3）具体操作步骤图示如下：  进入软件界面，选择菜单命令“【实验】—【简单模型机】，打开简单模型机实验数据通路”  图。  将时序与操作台单元的开关 KK1、KK3 置为‘运行’档，开关 KK2 置为‘单拍’档，CON  单元所有开关置 0（由于总线有总线竞争报警功能，在操作中应当先关闭应关闭的输出开关，再打开应打开的输出开关，否则可能由于总线竞争导致实验出错） 按动 CON 单元的总清按钮，CLR，然后通过运行程序，在数据通路图中观测程序的执行过程。  ① 输入设备将 11H 打入 R0 寄存器。  将 IN 单元置 00010001，K7 置为 1，关闭 R0 寄存器的输出；K6 置为 1，打开 R0 寄存器的输入；WR、RD、IOM 分别置为 0、1、1，对 IN 单元进行读操作；LDAR 置为 0，不将数据总线的数打入地址寄存器。连续四次点击图形界面上的“单节拍运行”按扭（运行  一个机器周期），观察图形界面，在 T4 时刻完成对寄存器 R0 的写入操作。  ② 将 R0 中的数据 11H 打入存储器 01H 单元。  将 IN 单元置 00000001（或其他数值）。K7 置为 1，关闭 R0 寄存器的输出；K6 置为 0，  关闭 R0 寄存器的输入；WR、RD、IOM 分别置为 0、1、1，对 IN 单元进行读操作；LDAR  置为 1，将数据总线的数打入地址寄存器。连续四次点击图形界面上的“单节拍运行”按扭，观察图形界面，在 T3 时刻完成对地址寄存器的写入操作。  先将 WR、RD、IOM 分别置为 1、0、0，对存储器进行写操作；再把 K7 置为 0，打开  R0 寄存器的输出；K6 置为 0，关闭 R0 寄存器的输入； LDAR 置为 0，不将数据总线的数  打入地址寄存器。连续四次点击图形界面上的“单节拍运行”按扭，观察图形界面，在 T3  时刻完成对存储器的写入操作。  ③ 将当前地址的存储器中的数写入到 R0 寄存器中。将 IN 单元置 00000001（或其他数值）K7 置为 1，关闭 R0 寄存器的输出；K6 置为0，关闭 R0 寄存器的输入；WR、RD、IOM 分别置为 0、 1， IN 单元进行读操作；1、 对LDAR置为 1，不将数据总线的数打入地址寄存器。连续四次点击图形界面上的“单节拍运行”按扭，观察图形界面，在 T3 时刻完成对地址寄存器的写入操作。  将 K7 置为 1，关闭 R0 寄存器的输出；K6 置为 1，打开 R0 寄存器的输入；WR、RD、  IOM 分别置为 0、1、0，对存储器进行读操作；LDAR 置为 0，不将数据总线的数打入地址  寄存器。连续四次点击图形界面上的“单节拍运行”按扭，观察图形界面，在 T3 时刻完成  对寄存器 R0 的写入操作。    注：由于采用简单模型机的数据通路图，为了不让悬空的信号引脚影响通路图的显示结果，将  这些引脚置为无效。在接线时为了方便，可将管脚接到 CON 单元闲置的开关上，若开关打到‘1’，等效于接到‘VCC’；若开关打到‘0’，等效于接到‘GND’。  ④ 将 R0 寄存器中的数用 LED 数码管显示。  先将 WR、RD、IOM 分别置为 1、0、1，对 OUT 单元进行写操作；再将 K7 置为 0，  打开 R0 寄存器的输出；K6 置为 0，关闭 R0 寄存器的输入； LDAR 置为 0，不将数据总线的数打入地址寄存器。连续四次点击图形界面上的“单节拍运行”按扭，观察图形界面，在  T3 时刻完成对 OUT 单元的写入操作。     1. 实验结果及分析（包括结果描述、实验现象分析、影响因素讨论、综合分析和结论等）   实验结果：       1. 实验小结、建议及体会      通过这次实验，使得自己对系统总线和总线接口有了更深的了解。总线作为CPU与外设、存储器之间传递数据的“桥梁”，有着举足轻重的作用。在数据/地址多路复用总线上，可以输入输出数据，起着沟通的作用。在不同的总线周期，执行不同的指令操作，总线上也传递着不同的数据。为了使CPU能够具有更高的工作效率，产生了中断这种技术。原来CPU必须不断查询外设，询问其状态，以便在需要的时候运行子程序。但是这种方法特别耗时，CPU一直在查询外设状态，无法执行自己的工作。有了中断之后，CPU可以做着自己的工作，一旦外设发出中断请求，CPU便转向子程序，执行指令，执行完后再回到中断处，这里一来CPU的工作效率大大提高。虽然中断的产生使得CPU在需要的时候执行子程序，但是在传送数据的时候，都是由CPU进行监控，一旦数据量非常大的时候，这时对CPU的负担是非常大的。因为请求都是“外设-CPU-存储器”或者“存储器-CPU-外设”，无论何种方式，CPU都必须无时无刻对每一个数据进行监控。DMA是一直使得数据可以直接从“外设-存储器”或者“存储器-外设”，这样一来，CPU便可以在数据传输过程中，继续执行自己的操作，数据传送过程则由DMA控制。这样一来CPU的工作效率又大大提高了许多。通过这次的三个实验，自己对计算机内部数据传递的方式有了深刻的了解，不同是方式有着不同的优缺点，在不同的需求场合下，需要合理的选择适合的方法。 |