|  |
| --- |
| **北 京 邮 电 大 学**  **实 验 报 告**  **课程名称 计算机组成原理实验**  **实验名称 CPU&微程序&中断实验**  **人工智能学院2020219111班 姓名 刘帅**  **教师 刁婷 成绩\_\_\_\_\_\_**  **2022年6月1日** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. **实验任务及目的** 2. **熟悉CPU的组成和基本功能** 3. **熟悉TEC-8实验台上的控制台的使用** 4. **熟悉控制台、时序部件、数据通路和微程序控制器的连接** 5. **理解计算机如何取指令、执行指令、如何在一条指令结束后自动取下一条指令** 6. **理解CPU中断的原理，模拟单级中断和中断返回的过程** 7. **熟悉中断控制器、中断向量、中断屏蔽等概念** 8. **了解中断子程序和一般子程序的本质区别，掌握中断的突发性和随机性。** 9. **通过模型机，构建计算机整体概念**   **二、实验环境**  **Tec-8仿真软件，ALU运算单元，双端口存储器单元、微程序控制器单元**   1. **实验过程描述（包括但不限于：实验过程记录表、实验要求中思考问题的操作流程、仿真软件连线截图、遇到的问题及解决方案）**   **1、连线**  **将控制器与数据通路进行连接，微指令操作码IR7-I~IR4-I直接与数据通路IR7-O~IR4-O连接，ALU的标志位C-O和Z-O与微程序控制器的C-1和Z-I相连。DP=1，使用单拍模式。**  **2、读取数据**   * 1. **将数写进寄存器中**   **取SWCBA=100，为写寄存器模式。按下QD，此时为09H，SBUS有效。此时，SEL0-SEL3为1000，代表写入的寄存器是R0。DRW有效，可以写入寄存器。STOP有效，执行完该步骤后暂停。为08H，代表下一条执行的微指令是写寄存器R1。由于R0和R1不需要数据，因此数据开关为00H，连续按两次QD，到达写入R2的步骤。**  **同理，在数据开关这里输入12H,SBUS打开，SEL选中R2，DRW有效，12H就被送入了R2中。**  **相同地，把OFH写到R3中，写寄存器步骤结束。**  **②将指令写入存储器中**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **指令** | **16进制机器代码** | **IR7~IR4** | **IR3~IR2** | **IR1~IR0** | | LD R0，[R3] | 53H | LD~0101 | RO~00 | R3~11 | | INC R3 | 4CH | INC~0100 | R3~11 |  | | LD R1，[R3] | 57H | LD~0101 | R1~01 | R3~11 | | SUB R0,R1 | 21H | SUB~0010 | RO~00 | R1~01 | | JZ 0BH | 86H | JZ~1000 | OFFSET=0BH-当前PC值（05H）=06H | | | ST R0,[R2] | 68H | ST~0110 | R2~10 | R0~00 | | INC R3 | 4CH | INC~0100 | R3~11 |  | | LD R0，[R3] | 57H | LD~0101 | R1~01 | R3~11 | | ADD R0,R1 | 11H | ADD~0001 | RO~00 | R1~01 | | JC OCH | 72H | JC~0111 | OFFSET=0CH-当前PC值（0AH）=02H | | | INC R2 | 48H | INC~0100 | R2~10 |  | | ST R2,[R2] | 6AH | ST~0110 | R2~10 | R2~10 | | AND R0,R1 | 31H | AND~0011 | RO~00 | R1~01 | | OUT R2 | A2H | OUT~1010 | XX | R2~10 | | STP | E0H | STOP~1110 | XX |  |   **通过写存储器，将机器代码的16进制形式写入双端口存储器中。**  **③单拍方式执行程序**   |  |  |  | | --- | --- | --- | | **指令** | **操作** | **结果** | | LD R0，[R3] | 将R3地址所指内存的数放到R0里 | [0F]=85H,R0从00H变成85H | | INC R3 | R3加1 | R3从0FH变成10H | | LD R1，[R3] | 将R3地址所指内存的数放到R1里 | [10]=23H,R1从00H变成23H | | SUB R0,R1 | 将R0-R1的数放到R0中 | R0从85H变成（85H-23H）=62H | | JZ 0BH | 如果ALU计算完的数是0，跳到0BH执行 | 不进行Z跳转，继续05H的指令 | | ST R0,[R2] | 将R0的数放到R2所指的内存中 | [12]的数从00H变成62H | | INC R3 | R3自加1 | R3从10H变成11H | | LD R0，[R3] | 将R3所指内存的数放到R0里 | [11]=EFH,R0从62H变成EFH | | ADD R0,R1 | 将R0+R1的数放到R0中 | R0从EFH变成（EFH+23H）=12H，C=1 | | JC OCH | 如果ALU计算完的数有溢出，那么程序跳转到0CH执行 | 跳转至0CH执行 | | INC R2 | R2自加1 | 未执行 | | ST R2,[R2] | 将R2的数存入R2所指的内存中 | 未执行 | | AND R0,R1 | 将R0&R1的数放到R0中 | R0从12H变成（12H&23H）=02H | | OUT R2 | 输出R2 | 将12H输出到DBUS中 | | STP | 停机 | 停机 |   **单拍执行：**  **SWCBA=000，每次都先取指令，然后通过IR7~IR4译码并执行指令。以其中某一时刻的数据通路原理为例，展示数据传送、存储过程。将0FH从R3中送到DBUS中，准备作为地址输入到地址寄存器中，代表下一刻选取的存储器地址为0F，将0F中的数据存入R0之中。从而执行第一条指令LD R0，[R3]。控制信号M=1,S=1010，直接将R3中的0FH打入数据通路中。PC7~0=01H，代表下一条要执行的指令位置是01H。IR7~0=53H，代表此刻执行的指令的机器代码是53H，AR暂时仍为00H，因为0FH还没有送到AR中。INS7~0=4CH，就是下一条执行的指令的机器码。B7~0=0FH，D7~0=0FH，代表0FH通过ALU，从寄存器读到了总线上。为25H，代表正在执行LD指令（第一个CPU周期），即将ABUS打开，LAR有效，将地址从通用寄存器中取到地址寄存器中。为0EH,说明下一个操作是LD指令的第二个CPU周期的操作，从内存读数并放到通用寄存器中。**  **直至程序运行到STOP信号结束。**  **最终实验结果如下，寄存器R0~R3的值分别为02H\23H\12H\11H，数据通路中的数据为12H，内存中13H地址所指的单元内容为62H，符合理论推测的结果：**  **连续执行：再将DP调至连续模式，从头执行程序，发现程序运行结果与单排结果相同。**  **3.2 实验过程记录表** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **四、中断实验**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **指令** | **16进制机器代码** | **IR7~IR4** | **IR3~IR2** | **IR1~IR0** | | EI | D0H | EI~1101 |  |  | | INC R0 | 40H | INC~0100 | R0~00 |  | | JMP [R1] | 94H | JMP~1001 | R1~01 |  | | ADD R0,R0 | 10H | ADD~0001 | RO~00 | RO~00 | | EI | D0H | EI~1101 |  |  | | IRET | B0H | IRET~1011 |  |  |   **在写入00H-09H的指令后，从数据开关设置输入地址45H，从而使地址寄存器在读取到中断信号时跳转到45H地址的内容，再重复刚才的过程，将中断后的三条指令输入。再为R0赋初值00H，R1赋初值01H。以连续方式运行程序。运行后按下暂停，此时PC为05H，R0为04H。QD置1，打码开关输入中断服务程序的入口地址45H，准备进入中断子程序。** |

|  |
| --- |
| **五、实验总结**  **通过本次实验，我对于CPU组成工作原理、对于计算机如何取指令、执行指令、如何通过微指令进行连续指令执行有了更为深刻的理解，同时也了解了CPU中断原理等相关知识。至此，本学期的全部有关计组实验的课程已经全部结束了，我认为这门课程是使我感到最大收获的课程。虽然理论课对于CPU已经有所讲解，可我真正了解CPU的工作机制，体会到其自动化启动、终止的神奇还是在实验课当中（具体来讲是最后两次实验之中），我了解了CPU内部环环相扣的严密逻辑和设计的巧思，同时对于信号的流通，以及周期有了更为深刻的认识。然而，线上实验也存在一定的弊端，比如我没有着手进行实验连线、没有真正通过指示灯观察内部寄存器、数据等实际存储情况，这些都是线上实验所感受不到的。因此我希望，如果以后有机会的话，我一定会线下去体验计组实验，体会一下物理排错过程的乐趣。** |

|  |
| --- |
| **四、实验总结**  本次实验让我对更加熟悉了TEC-8仿真软件，对于微指令及指令、寄存器之间的信息交互方式有了更加系统和完备的了解，同时，微指令的顺序控制部分决定产生下一条微指令地址的原理，使我对于指令状态转移、逻辑转移有了更加清晰的认识。 |