

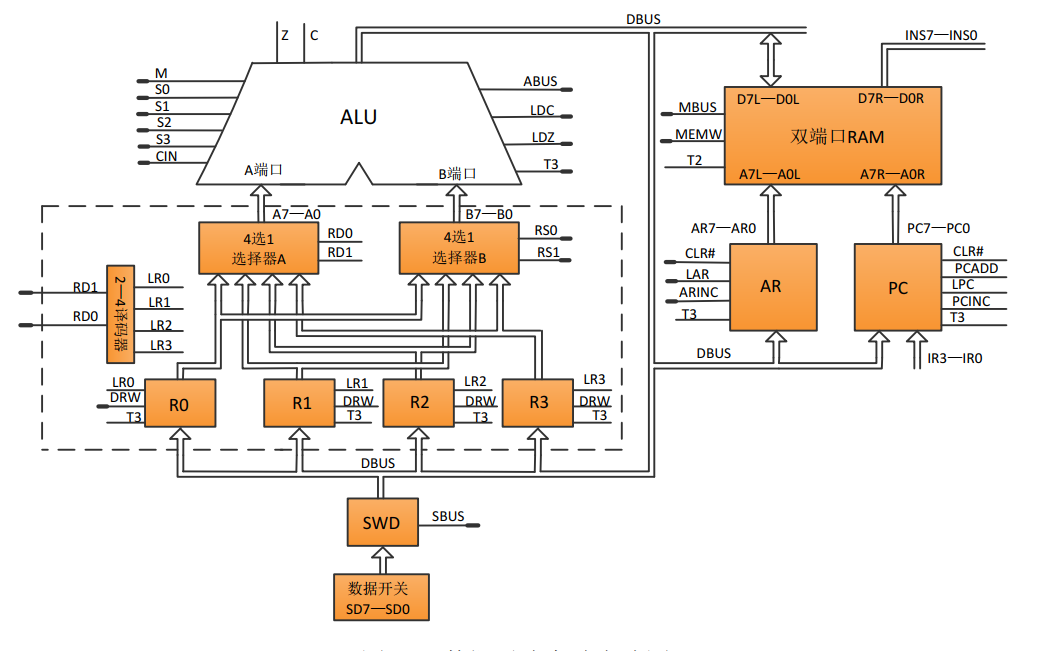
**计组课程设计报告**

|  |  |  |
| --- | --- | --- |
| 小组成员： | 涂旭 | 20171000246 |
| 牟鑫一 | 20161001764 |
| 殷信宇 | 20171002743 |
| 院（部）： | 计算机学院 | |
| 专业： | 计算机科学与技术 | |
| 指导老师： | 胡成玉 | |

2019 年 1 月

**第一部分**

**实验原理：**



数据运算操作：由 RD1、RD0 选中的寄存器通过 4 选 1 选择器 A送往 ALU 的 A端口，由 RS1、RS0 选中的寄存器通过 4 选 1 选择器 B 送往 ALU的 B 端口；信号 M、S3、S2、S1、S1 和 S0 决定 ALU 的运算类型。ALU 对 A 端口和 B 端口的两个数连同 CIN 的值进行算数逻辑运算，得到的数据运算结果在信号 ABUS 为 1 时送往数据总线 DBUS。

寄存器之间进行数据传送操作：由 RS1、RS0 选中的寄存器通过 4 选 1 选择器 B 送往 ALU 的 B 端口；ALU 将 B 端口的数在信号 ABUS 为 1 时送往数据总线 DBUS；在 T3 的上升沿将数据总线上的数写入由 RD1、RD0 选中的寄存器。ALU 进行数据传送操作由一组特定的 M、S3、S2、S1、S0、CIN 的值确定。

运算操作：由 RS1、RS0 选中的寄存器通过 4 选 1 选择器 B 送往 ALU 的 B 端口；由 RD1、RD0 选中的寄存器通过 4 选 1 选择器 A 送往 ALU 的 A 端口；ALU 对数 A 和 B 进行运算，运算的数据结果在信号 ABUS 为 1 时送往数据总线 DBUS；在 T3 的上升沿将数据总线上的数写入由 RD1、RD0 选中的寄存器。ALU 进行何种运算操作由 M、S3、S2、S1、S0、CIN 的值确定。

存储器中取数操作：由地址 AR7~AR0 指定的存储器单元中的数在信号 MEMW 为 0时被读出；在 MBUS 为 1 时送数据总线 DBUS；在 T3 的上升沿写入由 RD1、RD0 选中的寄存器。

**实验步骤**

⑴设置数据通路实验模式首先将“控制转换”开关拨到最下方位置既“微程序”灯亮。按复位按钮 CLR，使 TEC-8实验系统复位。指示灯 µA5~µA0 显示 00H。将操作模式开关设置为 SWC=1、SWB=1、SWA=1，准备进入数据通路实验。按一次 QD 按钮，进入数据通路实验。

⑵将数 75H 写到寄存器 R0、数 28H 写到 R1、数 89H 写到 R2、数 32H 写到 R3。。指示灯 µA5~µA0 显示 0FH。在数据开关 SD7~SD0 上设置数 75H。在数据总线 DBUS 指示灯D7~D0 上可以看到数设置得正确不正确，发现错误需及时改正。数设置正确后，按一次 QD按钮，将 SD7~SD0 上的数写入寄存器 R0，进入下一步。依照写 R0 的方式，在指示灯 µA5~µA0 显示 32H 时，在指示灯 B7~B0 观测寄存器 R0 的值，将数 28H 写入 R1；在指示灯 µA5~µA0 显示 33H 时，在指示灯 B7~B0 上观测 R1 的值，将数 89H写入 R2；在指示灯 µA5~µA0 显示 34H 时，在指示灯 B7~B0 上观测 R2 的值，将数 32H 写入 R3。

⑶设置存储器地址 AR 和程序计数器 PC指示灯 µA5~µA0 显示 35H。此时指示灯 B7~B0 显示寄存器 R3 的值。在数据开关 SD7~SD0上设置地址 20H。在数据总线 DBUS 指示灯 D7~D0 上可以看到地址设置得正确不正确。地址设置正确后，按一次 QD 按钮，将 SD7~SD0 上的地址写入地址寄存器 AR 和程序计数器 PC，进入下一步。

⑷将寄存器 R0、R1、R2、R3 中的数依次写入存储器 20H、21H、22H 和 23H 单元。指示灯 µA5~µA0 显示 36H。此时指示灯 AR7~AR0 和 PC7~PC0 分别显示出存储器左、右两个端口的存储器地址。指示灯 A7~A0、B7~B0 和 D7~D0 都显示寄存器 R0 的值。按一次 QD 按钮，将 R0 中的数写入存储器 20H 单元，进入下一步。依照此法，在指示灯 µA5~µA0 显示 37H 时，在 INS7~INS0 上观测存储器 20H 单元的值，将 R1 中的数写入存储器 21H 单元；在指示灯 µA5~µA0 显示 38H 时，在 INS7~INS0 上观测存储器 21H 单元的值，将 R2 中的数写入存储器 22H 单元；在指示灯 µA5~µA0 显示 39H 时，在INS7~INS0 上观测存储器 22H 单元的值，将 R3 中的数写入存储器 23H 单元。

⑸重新设置存储器地址 AR 和程序计数器 PC指示灯 µA5~µA0 显示 3AH。此时指示灯 PC7~PC0 显示 23H，INS7~INS0 显示存储器 23H单元中的数。在数据开关 SD7~SD0 上设置地址 20H。按一次 QD 按钮，将地址 20H 写入地址寄存器 AR 和程序计数器 PC，进入下一步。

⑹将存储器 20H、21H、22H 和 23H 单元中的数依次写入寄存器 R3、R2、R1 和 R0。指示灯 µA5~µA0 显示 3BH。此时指示灯 AR7~AR0 和 PC7~PC0 显示 20H，指示灯 D7~D0 和INS7~INS0 同时显示存储器 20H 中的数，按一次 QD 按钮，将存储器 20H 单元中的数写入寄存器 R3，进入下一步。依照此法，在指示灯 µA5~µA0 显示 3CH 时，在指示灯 B7~B0 上观测 R3 的值，将存储器21H 单元中的数写入寄存器 R2；在指示灯 µA5~µA0 显示 3DH 时，在指示灯 B7~B0 上观测 R2的值，将存储器 22H 单元中的数写入寄存器 R1；在指示灯 µA5~µA0 显示 3EH 时，在指示灯B7~B0 上观测 R1 的值，将存储器 23H 单元中的数写入寄存器 R0。

⑺观测 R0 的值指示灯 µA5~µA0 显示 00H。此时指示灯 A7~A0 显示 R0 的值，指示灯 B7~B0 显示 R3 的值。

**实验结果**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| μA5~μA0 | A7~A0 | B7~B0 | D7~D0 | AR | PC | INS7 | R0 | R1 | R2 | R3 |
| 0FH | 00H | 00H | 20H | 00H | 00H | FFH | 0 | 0 | 0 | 0 |
| 32H | 89H | 32H | 75H | 00H | 00H | FFH | 75H | 0 | 0 | 0 |
| 33H | 28H | 28H | 28H | 00H | 00H | FFH | 75H | 28H | 0 | 0 |
| 34H | 75H | 89H | 89H | 00H | 00H | FFH | 75H | 28H | 89H | 0 |
| 35H | 00H | 32H | 32H | 00H | 00H | FFH | 75H | 28H | 89H | 32H |
| 36H | 00H | 00H | 00H | 00H | 00H | D5H | 75H | 28H | 89H | 32H |
| 37H | 00H | 28H | 28H | 20H | 20H | 00H | 75H | 28H | 89H | 32H |
| 38H | 00H | 89H | 89H | 21H | 21H | 28H | 75H | 28H | 89H | 32H |
| 39H | 00H | 32H | 32H | 23H | 22H | 89H | 75H | 28H | 89H | 32H |
| 3AH | 00H | 20H | 20H | 24H | 23H | 32H | 75H | 28H | 89H | 32H |
| 3BH | 32H | 00H | 00H | 20H | 23H | 32H | 75H | 28H | 89H | 75H |
| 3CH | 89H | 28H | 28H | 20H | 23H | 32H | 75H | 28H | 28H | 75H |
| 3DH | 28H | 89H | 89H | 89H | 23H | 32H | 75H | 89H | 28H | 75H |
| 3EH | 00H | 32H | 32H | 32H | 23H | 32H | 32H | 89H | 28H | 75H |
| 00H | 00H | 00H | 00H | 24H | 23H | 32H | 32H | 89H | 28H | 75H |

**第二部分**

**实验目的**

（1）用微程序控制器控制数据通路，将相应的信号线连接，构成一台能运行测试程序的 CPU。

（2）执行一个简单的程序，掌握机器指令与微指令的关系。

（3）理解计算机如何取出指令、如何执行指令、如何在一条指令执行结束后自动取出下一条指令并执行，牢固建立的计算机整机概念。

**测试代码**

00H LD R0[R2] 0101 0010 （【R2】->R0）

01H LD R1[R3] 0101 0111 （【R3】->R1）

02H ADD R1,R2 0001 0110 （R1+R2->R1）

03H SUB R0,R3 0010 0011 （R0-R3->R0）

04H INC R0 0100 00XX （R0+1）

05H STOP 1110 XXXX

06H INC R3 0100 1100

07H ADD R0，R1 0001 0001

**测试结果**

RO R1 R2 R3

0 01H 01H 08H 09H

1 12H 00H 08H 09H

2 12H 30H 08H 09H

3 12H 2BH 08H 09H

4 0AH 2BH 08H 09H

5 0BH 2BH 08H 09H

6 STOP

实验测试和计算数值一样，结果无误。

**第三部分**

**教学目的**

（1） 融会贯通计算机组成与体系结构课程各章教学内容，通过知识的综合运用，加深对 CPU 各模块工作原理及相互联系的认识。

（2） 掌握硬连线控制器的设计方法。

（3） 学习运用当代的 EDA 设计工具，掌握用 EDA 设计大规模复杂逻辑电路的方法。

（4） 培养科学研究能力，取得设计和调试的实践经验。

**实验设备**

1 实验系统 1台

2 双踪示波器 1台

3直流万用表 1块

4逻辑测试笔 1支

5 PC机 1台

**CPU设计代码**

LIBRARY ieee; --头文件

USE ieee.std\_logic\_1164.all;

ENTITY CPU IS

PORT(SW:IN std\_logic\_vector(2 Downto 0);

IR:IN std\_logic\_vector(7 Downto 4);

W1,W2,W3:IN std\_logic;

C,Z,CLR,T3:IN std\_logic;

ARINC,CIN,DRW,LPC,LAR,LIR,LDZ,LDC,PCINC,PCADD,SELCTL,M,MEMW,STOP:OUT std\_logic;

SHORT,LONG,ABUS,SBUS,MBUS:OUT std\_logic;

S,SEL:OUT std\_logic\_vector(3 Downto 0));

END CPU;

ARCHITECTURE cont OF CPU IS

signal ST0:std\_logic;

signal SST0:std\_logic;

BEGIN

PROCESS(ST0,SST0,SW,IR,CLR,T3,W1,W2,W3)

BEGIN

IF (CLR='0') THEN --清空

ST0<='0';

SST0<='0';

CIN<='0';

DRW<='0';

LPC<='0';

LAR<='0';

LIR<='0';

LDZ<='0';

LDC<='0';

PCINC<='0';

PCADD<='0';

SELCTL<='0';

M<='0';

S<="0000";

SEL<="0000";

MEMW<='0';

STOP<='0';

SHORT<='0';

LONG<='0';

ABUS<='0';

MBUS<='0';

ARINC<='0';

ELSE

CIN<='0';

DRW<='0';

LPC<='0';

LAR<='0';

LIR<='0';

LDZ<='0';

LDC<='0';

PCINC<='0';

PCADD<='0';

SELCTL<='0';

M<='0';

S<="0000";

SEL<="0000";

MEMW<='0';

STOP<='0';

SHORT<='0';

LONG<='0';

ABUS<='0';

SBUS<='0';

MBUS<='0';

ARINC<='0';

SST0<=(NOT ST0)AND ((SW(2)AND(NOT(SW(1)))AND(NOT SW(0))AND W2)OR((NOT SW(2))AND SW(1)AND(NOT SW(0))AND W1)OR((NOT SW(2))

AND (NOT SW(1)) AND SW(0) AND W1));

IF(SST0='1' ) THEN

IF (T3'EVENT AND (T3='0'))THEN

ST0<='1';

END IF;

END IF;

CASE SW IS

WHEN "100" =>

IF (ST0='0') THEN

IF (W1='1') THEN

SBUS<='1';

SEL<="0011";

SELCTL<='1';

DRW<='1';

STOP<='1';

END IF;

IF (W2='1') THEN

SBUS<='1';

SEL<="0100";

SELCTL<='1';

DRW<='1';

STOP<='1';

SST0<='1';

END IF;

END IF;

IF (ST0='1') THEN

IF (W1='1') THEN

SBUS<='1';

SEL<="1001";

SELCTL<='1';

DRW<='1';

STOP<='1';

END IF;

IF (W2='1') THEN

SBUS<='1';

SEL<="1110";

SELCTL<='1';

DRW<='1';

STOP<='1';

END IF;

END IF;

WHEN "011" =>

IF (W1='1') THEN

SEL<="0001";

SELCTL<='1';

STOP<='1';

END IF;

IF (W2='1') THEN

SEL<="1011";

SELCTL<='1';

STOP<='1';

END IF;

WHEN "010"=>

IF (ST0='0') THEN

IF (W1='1') THEN

SBUS<='1';

LAR<='1';

STOP<='1';

SHORT<='1';

SELCTL<='1';

SST0<='1';

END IF;

END IF;

IF (ST0='1') THEN

IF (W1='1') THEN

MBUS<='1';

ARINC<='1';

STOP<='1';

SHORT<='1';

SELCTL<='1';

END IF;

END IF;

WHEN "001"=>

IF (ST0='0') THEN

IF (W1='1') THEN

SBUS<='1';

LAR<='1';

STOP<='1';

SHORT<='1';

SELCTL<='1';

SST0<='1';

END IF;

END IF;

IF (ST0='1') THEN

IF (W1='1') THEN

SBUS<='1';

MEMW<='1';

ARINC<='1';

STOP<='1';

SHORT<='1';

SELCTL<='1';

END IF;

END IF;

WHEN "000"=>

IF (W1='1') THEN

LIR<='1';

PCINC<='1';

END IF;

CASE IR IS

WHEN "0001"=> --add

IF (W2='1') THEN

S<="1001";

CIN<='1';

ABUS<='1';

DRW<='1'; --将数据送往指定寄存器

LDZ<='1';

LDC<='1';

END IF;

WHEN "1111"=> --取数加1

IF (W2='1') THEN

M<='1';

S<="1010";

ABUS<='1';

LAR<='1'; --将数据总线上的数写入地址寄存器AR

LONG<='1';

END IF;

IF (W3='1') THEN

DRW<='1';

MBUS<='1';

LONG<='1';

END IF;

IF (W1='1') THEN

S<="0000";

ABUS<='1';

DRW<='1';

LDZ<='1';

LDC<='1';

END IF;

WHEN "0010"=> --sub

IF (W2='1') THEN

S<="0110";

ABUS<='1';

DRW<='1';

LDZ<='1';

LDC<='1';

END IF;

WHEN "0011"=> --and

IF (W2='1') THEN

M<='1';

S<="1011";

ABUS<='1';

DRW<='1';

LDZ<='1';

END IF;

WHEN "0111"=> --相与加1

IF (W2='1') THEN

M<='1';

S<="1011";

ABUS<='1';

DRW<='1';

LDZ<='1';

LONG<='1';

END IF;

IF (W2='1') THEN

S<="0000";

ABUS<='1';

DRW<='1';

LDZ<='1';

LDC<='1';

END IF;

WHEN "0100"=> --inc

IF (W2='1') THEN

S<="0000";

ABUS<='1';

DRW<='1';

LDZ<='1';

LDC<='1';

END IF;

WHEN "0101"=> --ld

IF (W2='1') THEN

M<='1';

S<="1010";

ABUS<='1';

LAR<='1'; --将数据总线上的数写入地址寄存器AR

LONG<='1';

END IF;

IF (W3='1') THEN

DRW<='1';

MBUS<='1';

END IF;

WHEN "0110"=> --st

IF (W2='1') THEN

M<='1';

S<="1111";

ABUS<='1';

LAR<='1';

LONG<='1';

END IF;

IF (W3='1') THEN

S<="1010";

M<='1';

ABUS<='1';

MEMW<='1';

END IF;

WHEN "1000"=> --jz

IF(W2='1') THEN

IF(Z='1') THEN

PCADD<='1';

END IF;

END IF;

WHEN "1001"=> --jmp

IF (W2='1') THEN

M<='1';

S<="1111";

ABUS<='1'; --运算结果送往数据总线

LPC<='1'; --数据总线送往程序计数器

END IF;

WHEN "1010"=>

IF (W2='1') THEN

M<='1';

S<="0000"; --增添新模块实现A―

ABUS<='1';

DRW<='1';

END IF;

WHEN "1110"=> --STP

IF (W2='1') THEN

STOP<='1';

END IF;

WHEN OTHERS=>SBUS<='0';

END CASE;

WHEN OTHERS=>SBUS<='0';

END CASE;

END IF;

END PROCESS;

END cont;

--新增指令

WHEN "0111"=> --相与加1

IF (W2='1') THEN

M<='1';

S<="1011";

ABUS<='1';

DRW<='1';

LDZ<='1';

LONG<='1';

END IF;

IF (W2='1') THEN

S<="0000";

ABUS<='1';

DRW<='1';

LDZ<='1';

LDC<='1';

END IF;

WHEN "1111"=> --取数加1

IF (W2='1') THEN

M<='1';

S<="1010";

ABUS<='1';

LAR<='1';

LONG<='1';

END IF;

IF (W3='1') THEN

DRW<='1';

MBUS<='1';

LONG<='1';

END IF;

IF (W1='1') THEN

S<="0000";

ABUS<='1';

DRW<='1';

LDZ<='1';

LDC<='1';

END IF;

**结果：**

|  |  |  |
| --- | --- | --- |
|  | R0 | R1 |
| 01 | 10001001 | 10010111 |
| 02 | 00000110 | 10100111 |
| 03 | 01100110 | 01001001 |
| 04 | 00101110 | 01011000 |
| 05 | STOP |  |

**实验感想**

通过这次实验，我们对控制器和数据通路又有了更深刻地认识，CPU的神秘面纱也在这次操作中揭开，与同学们交流与合作的过程也是十分舒适。同时也十分感谢老师的指点和学习资料，确保实验完成的质量，也驱除许多关于操作上的疑云。

通过这次课程设计，我们对硬件设计的步骤有了深刻的了解。当我们拿到任务之后，运用我们掌握的知识对这个任务进行分析，然后划分几个阶段并明确每个阶段的任务如下：首先理解题目，画出流程图；其次进行状态编码并予以化简；再次进行编程，下载；然后是调试，查错，验收；最后完成实验报告。

随着实验的不断深入同学们的配合越来越默契，共同探讨，互补合作，使得大家在知识和能力上都有了很大的提高。总而言之，经过几天的试验我们收获颇丰，同时我们还要感谢实验老师对我们的悉心指导，帮我们分析解决了许多棘手的问题，也为我们拓展思路提供了很好的指导。