实 验 报 告

课程名称 CPU 设计

实验项目 实验模型机的研制

实验仪器 微程序控制板（C 板）、数据通路版（B 板）、控制信号版（A 板）32 位微指令读写板各一块

系　　别 计算机学院

专　　业 计算机科学与技术

班级/学号 计科 1206/

学生姓名

联系电话

实验日期 2014.03-26

成　　绩

指导教师 沈美娥

一．设计总要求

模型计算机设计的总要求主要有以下几个方面：

（1） 模型机采用暂存器型的计算器结构。

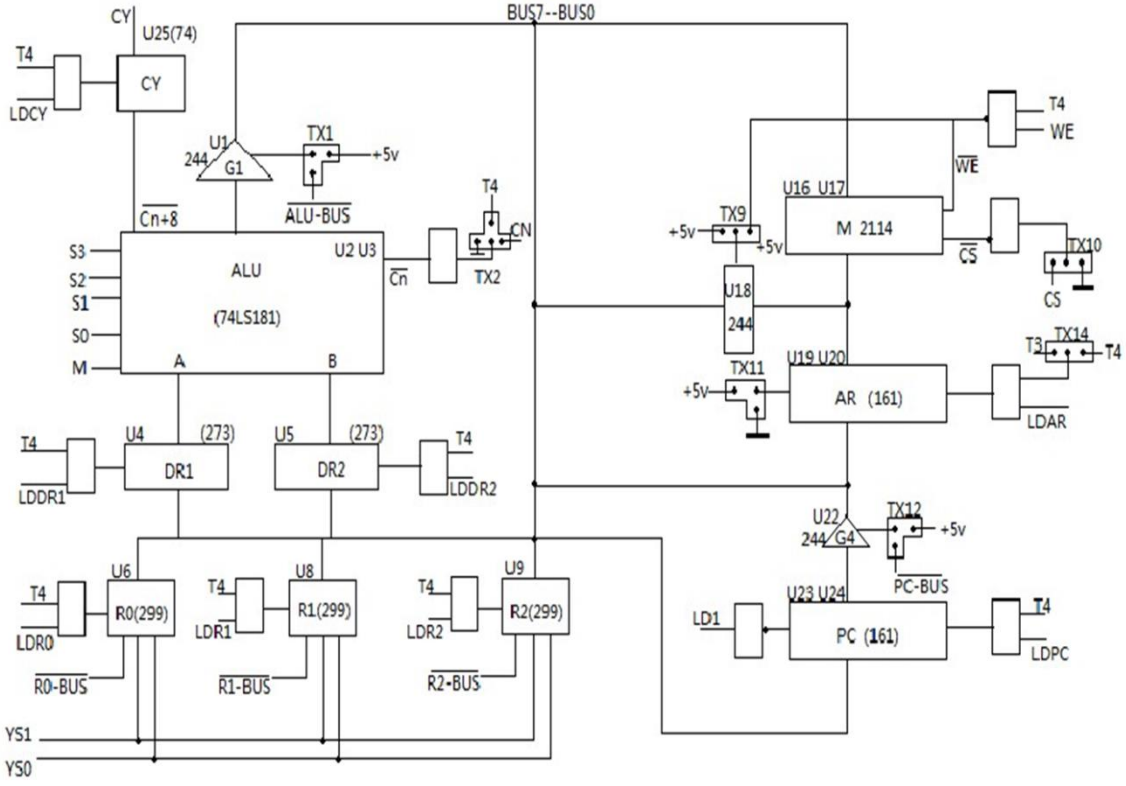
（2） 设计一个十六条指令的指令系统，包括单字长指令和双字长指令，其指令寻址方式包括立即寻址、直接寻址、间接寻址、寄存器直接寻址等；指令类型包括算逻类指令、传送类指令、控制类指令、停机指令等。

（3） 微程序控制器采用断定方式，微指令编码采用直接控制和字段编码相结合的方式，设计完成微程序流程图，编写微指令。

（4） 自己编写一段小程序，完成某个小的功能，以验证指令的正确性。

二．设计模型机的数据通路

本实验仪的 B 板已经给我们提供了带单总线的数据通路，如图一所示。



图一

三．模型机指令系统

一． 指令类型

指令系统系统可包含以下类型的指令：

1. 算术/逻辑运算类指令

算术/逻辑运算类指令，如取反、加 1、加法、减法、与、或异或等指令。

2. 移位操作类指令

移位操作类指令，如左移指令和右移指令。

3. 数据传送类指令

数据传送类指令，通常指 CPU 内部寄存器之间传送或累加器与寄存器之间的数据传送指令，如 MOV、LDI

4. 程序转移控制类指令

存储器操作类指令，即存储器读/写指令。一般指的是把内存某单元内容读到某通用寄存器，或将某通用寄存器内容写入某内存单元。如 LAD、STA。

5. 程序转移控制类指令

转移指令分无条件转移和有条件转移。“条件”可以是运算器的一些状态标志（如进位标志 C_y 等）。如 JMP、JC。

二. 指令操作数寻址方式及其编码

X 字段	有效地址	说明
00	E=D	直接寻址
01	E=(D)	间接寻址
10	E=(PC)+D	相对寻址
11	E=(R_2)+D	变址寻址

指令系统汇总表

指令长度	类型	助记符	功能	指令格式
单字长指令	零地址指令	HLT	停机	0000 ** **
	单地址指令	COM R_d	$\overline{(R_d)}\rightarrow R_d$	0001 ** R_d
		INC R_d	$(R_d)+1\rightarrow R_d$	0010 ** R_d
		RAL R_d	$2(R_d)\rightarrow R_d$	0011 ** R_d
		RAR R_d	$(R_d)/2\rightarrow R_d$	0100 ** R_d
	二地址指令	ADD R_s, R_d	(R_s)加$(R_d)\rightarrow R_d$	0101 R_s R_d
		SUB R_s, R_d	(R_s)减$(R_d)\rightarrow R_d$	0110 R_s R_d
		AND R_s, R_d	$(R_s)\cdot(R_d)\rightarrow R_d$	0111 R_s R_d
		OR R_s, R_d	$(R_s)+(R_d)\rightarrow R_d$	1000 R_s R_d
		EOR R_s, R_d	$(R_s)\oplus(R_d)\rightarrow R_d$	1001 R_s R_d
		MOV R_s, R_d	$(R_s)\rightarrow R_d$	1010 R_s R_d
双字长指令	二地址指令	LDI DATA, R_d	DATA$\rightarrow R_d$	1011 ** R_d DATA
	三地址指令	LAD X,D, R_d	$(MD)\rightarrow R_d$	11 X 00 R_d D
		STA X, D, R_d	$R_d\rightarrow$MD	11 X 01 R_d D

	二地址指令	JMP X, D	D→PC	11 X 10 ** D
		JC X, D	若 C_y=1,则 D→PC	11 11 ** D

四．指令执行流程

以条指令从内存取出到执行完成，需要若干个机器周期，任何指令的第一个机器周期都是取指令周期，并且伴随着程序计数器(PC)的自增，其余的机器周期与每条指令具体的操作有关。比如算术类的指令，先把寄存器中的数据经过数据总线放入暂存器，然后再经过逻辑运算单元放到数据总线上，如果需要写入寄存器，就把相关的微命令执行。如果遇到不同的寻址方式，如直接寻址，只需把地址送入 AR 即可，而如果是间接寻址，就需要把相应的内存单元的数据再次作为地址，进行寻址，相对寻址就是根据 PC 给一个偏移量，最后的地址为 PC+偏移量，变址寻址把 R_2 作为变址寄存器，最后的地址为变址寄存器的值加上偏移量然后进行寻址。在测试相对寻址的时候要注意计算当时 PC 的值，然后确定最后的地址单元中存放的值。我们这次使用的模型机的字长为 8 位，操作码有 4 位，总共有 16 种操作，具体的操作及其编码如下表。

指令流程

1. HLT

1. 停机　　TJ=1

2. COM R_d

1. PC→AR　　PC→BUS,LDAR=1
   PC+1→PC　　LDPC=1
2. RAM→BUS　　CS=1
   BUS→IR　　LDIR=1
3. (R_d)→BUS　　Rd→BUS
   BUS→DR_1　　LDDR1=1
4. $\overline{DR_1}$→BUS　　$S_3S_2S_1S_0MC_n$=000010, ALU→BUS
   BUS→R_d　　LDR_i=1, YS_1YS_0=11, DZ=1

3. INC R_d

1. PC→AR　　PC→BUS,LDAR=1
   PC+1→PC　　LDPC=1
2. RAM→BUS　　CS=1
   BUS→IR　　LDIR=1
3. (R_d)→BUS　　Rd→BUS
   BUS→DR_1　　LDDR1=1
4. (DR_1)加 1→BUS| $S_3S_2S_1S_0MC_n$=000001, ALU→BUS
   BUS→R_d　　LDR_i=1, YS_1YS_0=11, DZ=1

4. RAL R_d

1. PC→AR　　PC→BUS,LDAR=1
   PC+1→PC　　LDPC=1
2. RAM→BUS　　CS=1
   BUS→IR　　LDIR=1
3. 左移(R_d)　　LDR_i=1, YS_1YS_0=10
4. R_d→BUS　　YS_1YS_0=00, R_d→BUS, DZ=1

5. RAR R_d

1. PC→AR　　PC→BUS,LDAR=1
   PC+1→PC　　LDPC=1
2. RAM→BUS　　CS=1
   BUS→IR　　LDIR=1
3. 右移(R_d)　　LDR_i=1, YS_1YS_0=01
4. R_d→BUS　　YS_1YS_0=00, R_d→BUS, DZ=1

6. ADD R_s, R_d

1. PC→AR　　PC→BUS,LDAR=1
   PC+1→PC　　LDPC=1
2. RAM→BUS　　CS=1
   BUS→IR　　LDIR=1
3. (R_s)→DR_1　　R_s→BUS, BUS→DR_1
4. (R_d)→DR_2　　R_d→BUS, BUS→DR_2
5. (DR_1)加(DR_2)→BUS　　$S_3S_2S_1S_0MC_n$=100100, ALU→BUS
   BUS→R_d　　LDR_i=1, YS_1YS_0=11, DZ=1

7. SUB R_s, R_d

1. PC→AR　　PC→BUS,LDAR=1
   PC+1→PC　　LDPC=1
2. RAM→BUS　　CS=1
   BUS→IR　　LDIR=1

3. $(R_s)\rightarrow DR_1$ $R_s\rightarrow BUS$, $BUS\rightarrow DR_1$
4. $(R_d)\rightarrow DR_2$ $R_d\rightarrow BUS$, $BUS\rightarrow DR_2$
5. $(DR_1)-(DR_2)\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=011001, ALU→BUS
   $BUS\rightarrow R_d$ LDR_i=1, YS_1YS_0=11, DZ=1

8. AND R_s, R_d

1. PC→AR PC→BUS,LDAR=1
   PC+1→PC LDPC=1
2. RAM→BUS CS=1
   BUS→IR LDIR=1
3. $(R_s)\rightarrow DR_1$ $R_s\rightarrow BUS$, $BUS\rightarrow DR_1$
4. $(R_d)\rightarrow DR_2$ $R_d\rightarrow BUS$, $BUS\rightarrow DR_2$
5. (DR_1) $(DR_2)\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=101110, ALU→BUS
   $BUS\rightarrow R_d$ LDR_i=1, YS_1YS_0=11, DZ=1

9. OR R_s, R_d

1. PC→AR PC→BUS,LDAR=1
   PC+1→PC LDPC=1
2. RAM→BUS CS=1
   BUS→IR LDIR=1
3. $(R_s)\rightarrow DR_1$ $R_s\rightarrow BUS$, $BUS\rightarrow DR_1$
4. $(R_d)\rightarrow DR_2$ $R_d\rightarrow BUS$, $BUS\rightarrow DR_2$
5. $(DR_1)+(DR_2)\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=111010, ALU→BUS
   $BUS\rightarrow R_d$ LDR_i=1, YS_1YS_0=11, DZ=1

10. EOR R_s, R_d

6. PC→AR PC→BUS,LDAR=1
   PC+1→PC LDPC=1
7. RAM→BUS CS=1
   BUS→IR LDIR=1
8. $(R_s)\rightarrow DR_1$ $R_s\rightarrow BUS$, $BUS\rightarrow DR_1$
9. $(R_d)\rightarrow DR_2$ $R_d\rightarrow BUS$, $BUS\rightarrow DR_2$
10. $(DR_1)\oplus(DR_2)\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=011010, ALU→BUS
    $BUS\rightarrow R_d$ LDR_i=1, YS_1YS_0=11, DZ=1

11. MOV R_s, R_d

1. PC→AR PC→BUS,LDAR=1
   PC+1→PC LDPC=1
2. RAM→BUS CS=1
   BUS→IR LDIR=1
3. $(R_s)\rightarrow DR_2$ $R_s\rightarrow BUS$, $LDDR_{2=1}$
4. $DR_2\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=101010, ALU→BUS
   $BUS\rightarrow R_d$ LDR_i=1, YS_1YS_0=11, DZ=1

12. LDI DATA, R_d

1. PC→AR PC→BUS,LDAR=1
   PC+1→PC LDPC=1
2. RAM→BUS CS=1
   BUS→IR LDIR=1
3. PC→AR PC→BUS, LDAR=1
   PC+1→PC LDPC=1
4. RAM→BUS CS=1
   $BUS\rightarrow R_d$ LDR_i=1, YS_1YS_0=11, DZ=1

X=00

1. PC→AR PC→BUS, LDAR=1
   PC+1→PC LDPC=1
2. RAM→BUS CS=1
   $BUS\rightarrow DR_2$ $LDDR_2$=1
3. $(DR_2)\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=101010, ALU→BUS
   BUS→AR LDAR=1, P(2)

X=01

1. PC→AR PC→BUS,LDAR=1
   PC+1→PC LDPC=1
2. RAM→BUS CS=1
   $BUS\rightarrow DR_2$ $LDDR_2$=1
3. $(DR_2)\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=101010, ALU→BUS
   BUS→AR LDAR=1
4. RAM→BUS CS=1
   $BUS\rightarrow DR_2$ $LDDR_2$=1
5. $(DR_2)\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=101010, ALU→BUS
   BUS→AR LDAR=1, P(2)

X=10

1. PC→AR PC→BUS,LDAR=1
   PC+1→PC LDPC=1
2. RAM→BUS CS=1
   $BUS\rightarrow DR_1$ $LDDR_1$=1
3. (PC)→BUS PC→BUS
   $BUS\rightarrow DR_2$ $LDDR_2$=1
4. (DR_1)加$(DR_2)\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=100100, ALU→BUS
   BUS→AR LDAR=1
   $BUS\rightarrow DR_2$ $LDDR_2$=1, P(2)

X=11

1. PC→AR PC→BUS, LDAR=1
   PC+1→PC LDPC=1
2. RAM→BUS CS=1
   $BUS\rightarrow DR_2$ $LDDR_2$=1
3. $R_2\rightarrow DR_1$ $R_2\rightarrow BUS$, $LDDR_1$=1
4. (DR_1)加$(DR_2)\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=100100, ALU→BUS
   BUS→AR LDAR=1
   $BUS\rightarrow DR_2$ $LDDR_2$=1, P(2)

13. LAD X, D, R_d

1. RAM→BUS CS=1
   $BUS\rightarrow R_d$ LDR_i=1, YS_1YS_0=11, DZ=1

14. STA X, D, R_d

1. $(R_d)\rightarrow BUS$ $R_d\rightarrow BUS$
   BUS→RAM CS=1, WE=1, DZ=1

15. JMP X, D

1. $(DR_2)\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=101010, ALU→BUS
   BUS→PC LD_1=1, LDPC=1, DZ=1

16. JC X, D

1. 若 C_y=1,则$(DR_2)\rightarrow BUS$ $S_3S_2S_1S_0MC_n$=101010, ALU→BUS
   BUS→PC,否则 NOP LD_1=1, LDPC=1, DZ=1

五．微程序流程图

（见附页）

六．微程序表

（见附页）

七．调试程序

先将 4 片芯片按照 3 2 1 4 的顺序插在 C 板上，然后开启电源，进行调试，一定要注意断电插拔芯片。要想执行上面指令系统中的程序，要使用控制台指令先把程序写入内存，然后执行。有的时候还需要检查相关内存单元中的数据是否正确。

控制台指令有：

1. 控制台写内存指令 WM：代码 02H(0000 0010)
2. 控制台读内存指令 RM：代码 01H(0000 0001)
3. 控制台启动指令 QD：代码 00H(0000 0000)

下面介绍如何使用控制台指令：

1. 控制台写内存指令 WM
   （1） 按下 CLR 总清键
   （2） 置 SW=0000 0010，然后 KQD，进入写内存指令
   （3） 置 SW=(要存入的内存单元)，然后 KQD
   （4） KQD，把 PC 送 AR
   （5） 置 SW=(要存入的数据)，然后 KQD
   （6） KQD，把数据送入 RAM，并且 PC+1
   （7） KQD，返回第（5）步
2. 控制台读内存指令 RM
   （1） 按下 CLR 总清键
   （2） 置 SW=0000 0001，然后 KQD，进入读内存指令
   （3） 置 SW=(要读取的内存单元)，KQD
   （4） KQD，把 PC 送 AR
   （5） KQD，空操作
   （6） KQD，将相关内存的内容读出，并在总线上显示，并且 PC+1 为读出下一个内存单元做准备
   （7） KQD，返回第（5）步
3. 控制台启动指令
   （1） 按下 CLR 总清键
   （2） 置 SW=0000 0000，然后 KQD，进入启动指令
   （3） KQD，空操作
   （4） KQD，把 PC 送 AR
   （5） 置 SW=(要启动的程序所在的内存单元)，KQD

（6） KQD，将内存单元送到 PC

（7） KQD，转入微程序入口地址 55H。

所有指令总调试程序：

样机调试程序

地址	助记符	指令编码	说明	预期结果
10H	LDI 01H,R_1	1011 0001	01H→R_1	R_1=01H
11H		0000 0001		
12H	STA X_1, 01H, R_1	1101 0101	01H→(R_1)	(02H)=01H
13H		0000 0001		
14H	LAD X_0, 02H, R_2	1100 0010	(01H)→R_2	R_2=(02H)=01H
15H		0000 0010		
16H	LAD X_2, 01H, R_0	1110 0000	((PC)+1)→R_0	R_0=(19H)=02H
17H		0000 0001		
18H	LDI 02H, R_2	1011 0010	02H→R_2	R_2=02H
19H		0000 0010		
1AH	LAD X_3, 01H, R_0	1110 0000	((R_2)+1)→R_0	R_0=(03H)=11H
1BH		0000 0001		
1CH	LDI FFH, R_1	1011 0001	FFH→R_1	R_1=FFH
1DH		1111 1111		
1EH	ADD R_2, R_1	0101 1001	(R_2)加(R_1)→R_1	R_1=01H
1FH	JC X0, 21H	1100 1100	若 Cy=1 则 22→PC 否则,无操作	PC=22
20H		0010 0010		
21H	HLT	0000 0000	停机	
22H	COM R_2	0001 0010	($\overline{R_2}$)→R_2	R_2=FDH
23H	INC R_2	0010 0010	(R_2)+1→R_2	R_2=FEH
24H	RAL R_2	0011 0010	(R_2)*2→R_2	R_2=FCH
25H	RAR R_2	0100 0010	(R_2)/2→R_2	R_2=FEH
26H	SUB R_2, R_1	0110 1001	(R_2)减(R_1)→R_1	R_1=FDH
27H	AND R_2, R_1	0111 1001	(R_2)・(R_1)→R_1	R_1=FCH
28H	OR R_2, R_1	1000 1001	(R_2)+(R_1)→R_1	R_1=FEH
29H	EOR R_2, R_1	1001 1001	(R_2)⊕(R_1)→R_1	R_1=00H
2AH	MOV R_1, R_2	1010 0110	(R_1)→R_2	R_2=00H
2BH	JMP X_0, 21H	1100 1000	21H→PC	PC=21H
2CH		0010 0001		

注：

在启动上表的程序之前，要事先把下面几个内存单元用控制台指令写入对应的数据，然后进行调试。

01H	02H
02H	01H
03H	11H
19H	02H

八．综合设计总结

通过本次实践性的《CPU 设计》课程，我真真正正做到了理论结合实际。这次实验结合了《计算机组成原理》课程中学习的指令系统，把课本上的看到的理论以及课堂上听到的方法，从对指令系统的改写开始，到指令流程的设计，再到指令流程图的设计最后写出了完全以‘0’‘1’代码组成的微程序，然后通过 32 位微指令读写板，把所有的微程序代码写入到 4 个 281EEPROM 芯片中，最后在插在 C 板上，进行调试。

由于微程序多，信号多，出错在所难免。我认为解决错位的最有效的方法还是从源头上，先检查一下相应的控制信号是否正确，然后对照微程序表，判断是否有错误，最后，使用 32 位微程序读写板对照微程序表进行排错，做完上述工作后，在拿到 C 板上，对照程序流程图和微程序表进行测试。这样一定会高效得找出所有的错误。比如我在往芯片中输入程序之前，先检查了所有的控制信号与微程序表是否一致，然后把所有的微程序都输入到了芯片中后，又立即对照微程序表看是否输入正确，这个过程中，我找到了几个错误。最后在 C 板上调试的时候，又根据其显示，找出了 1 个不容易发现的错误。一定要注意在插拔芯片之前注意断电，并且不要弄断芯片的管脚，否则将会前功尽弃。

对于程序设计而言，俗话说 3 分编程，7 分调试。的确是这样，本次实验的大部分时间基本上都是在排错。为了测试所有功能都能够正常运行，我设计了本报告第七部分的调试程序。在设计 4 种寻址方式的程序的时候着实废了一番功夫。如果说模型机设计的过程是把理论变成现实的过程，那么调试程序设计的过程，其实是对程序在理论应用层面的一个更加清楚的认识，对理论的更深入理解的过程。通过这次实验，我切身的感受到，动手能力在实践中的重要性，感受到了实践对于帮助理解理论原理的必要性。