

# 同济大学实验报告纸

软件工程专业 22 届 4 班 姓名 胡峻玮 第 组 同组人员

课程名称 计算机组成原理实验 实验名称 简单模型机设计实验 实验日期 2023 年 12 月 28 日

## [实验目的]

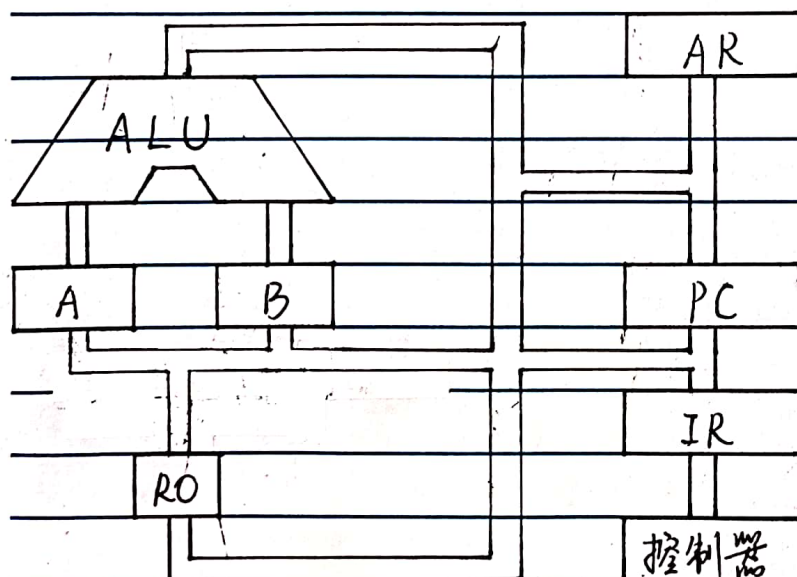
1. 在掌握部件单元的基础上, 构建一台基本模型计算机
2. 调试并理解机器指令及其相应的微程序运行过程

## [实验设备]

TD-CMA 组成原理实验箱、PC 机

## [实验原理]

CPU 由运算器 (ALU)、微程序控制器 (MC)、通用寄存器 (RO)、指令寄存器 (IR)、程序计数器 (PC) 和地址寄存器 (AR) 组成。CPU 在写入相应微指令后, 就具备了执行机器指令的功能。基本 CPU 构成原理图如下:



运算器: ALU: 完成数据计算或处理 寄存器 A、B

控制器: MC: 控制机器指令的分析和译码

寄存器: 用于暂存数据或指令 包含通用寄存器 RO: 用于临时存放中间数据;

程序计数器 PC: 存放下一条指令或数据的地址; 地址寄存器 AR: 存放当前指令或数据的地址;

指令寄存器 IR: 存放当前正在执行的机器指令。

# 同济大学实验报告纸

专业\_\_\_\_ 届\_\_\_\_ 班\_\_\_\_ 姓名\_\_\_\_ 第\_\_\_\_ 组 同组人员\_\_\_\_

课程名称\_\_\_\_ 实验名称\_\_\_\_ 实验日期\_\_\_\_ 年\_\_\_\_ 月\_\_\_\_ 日

CPU工作原理：取指 → 译码 → 执行 → 写回

1. 取指：CPU通过总线将PC中的地址发送给内存，内存将该地址上存储的指令发送给CPU，CPU将指令装载进指令寄存器IR中

2. 译码：指令寄存器IR中存储的二进制指令被CPU识别，CPU根据指令的不同类型和格式，将指令解码成具体的操作

3. 执行：CPU根据操作码判断指令类型，如算术运算、逻辑运算、分支等，CPU从主存或I/O设备读取操作数，根据指令类型执行相应操作

4. 写回：CPU将执行阶段得到的结果保存在通用寄存器中，根据需要，将执行结果写回主存或I/O设备。

本实验中各控制信号和计算机数据通路控制都是在微程序控制下自动产生的，实现特定指令的功能。如运算器中决定对操作数进行何种操作的控制信号S3-S0，存储器读写控制信号WR、RD、IOM0，对输入、输出设备的控制信号WR、RD、IOM1，对微程序强制分支的控制信号P<1>。机器指令的执行是一条机器指令从取出到执行结束由一段相应的微程序序列完成。本实验中共有五条机器指令如下：

符号	机器指令码	说明
IN	0010 0000	IN → R0
ADD	0000 0000	R0 + R0 → R0
OUT	0011 0000	R0 → OUT
JMP addr	1110 0000 xxxxxxxx	addr → PC
HLT	0101 0000	停机

其中JMP为双字节指令，其余均为单字节指令。





# 同济大学实验报告纸

专业\_\_\_\_ 届\_\_\_\_ 班\_\_\_\_ 姓名\_\_\_\_ 第\_\_\_\_ 组 同组人员\_\_\_\_  
 课程名称\_\_\_\_ 实验名称\_\_\_\_ 实验日期\_\_\_\_ 年\_\_\_\_ 月\_\_\_\_ 日

微指令格式

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0	
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MAS-MA0	

A字段

B字段

C字段

14	13	12	选择		11	10	9	选择		8	7	6	选择	
0	0	0	NOP		0	0	0	NOP		0	0	0	NOP	
0	0	1	LDA		0	0	1	ALU-B		0	0	1	P<1>	
0	1	0	LDB		0	1	0	RO-B		0	1	0	保留	
0	1	1	LDRO		0	1	1	保留		0	1	1	保留	
1	0	0	保留		1	0	0	保留		1	0	0	保留	
1	0	1	LOAD		1	0	1	保留		1	0	1	LDPC	
1	1	0	LDAR		1	1	0	PC-B		1	1	0	保留	
1	1	1	LDIR		1	1	1	保留		1	1	1	保留	

A、B、C为三个译码字段。C字段中P<1>为测试字位，其功能是根据机器指令及相应微代码进行译码使微程序转入相应的微地址入口，从而实现对机器指令的识别，并实现微程序分支。地址转移在一般情况下，微指令由控制存储器读出后直接给出下一条微指令的地址，即微地址，存放在微地址寄存器中。如果微程序未出现分支，那么下一条微指令的地址就直接由微地址寄存器给出；如果微程序出现分支，表明微程序出现条件转移，在该种情况下，通过判别测试字段P和执行部件的状态条件信息，去修改微地址寄存器中的内容，并按修改后的微地址读取下一条指令，从而实现地址转移。

MAS-MA0为6位后继地址(前2后4)

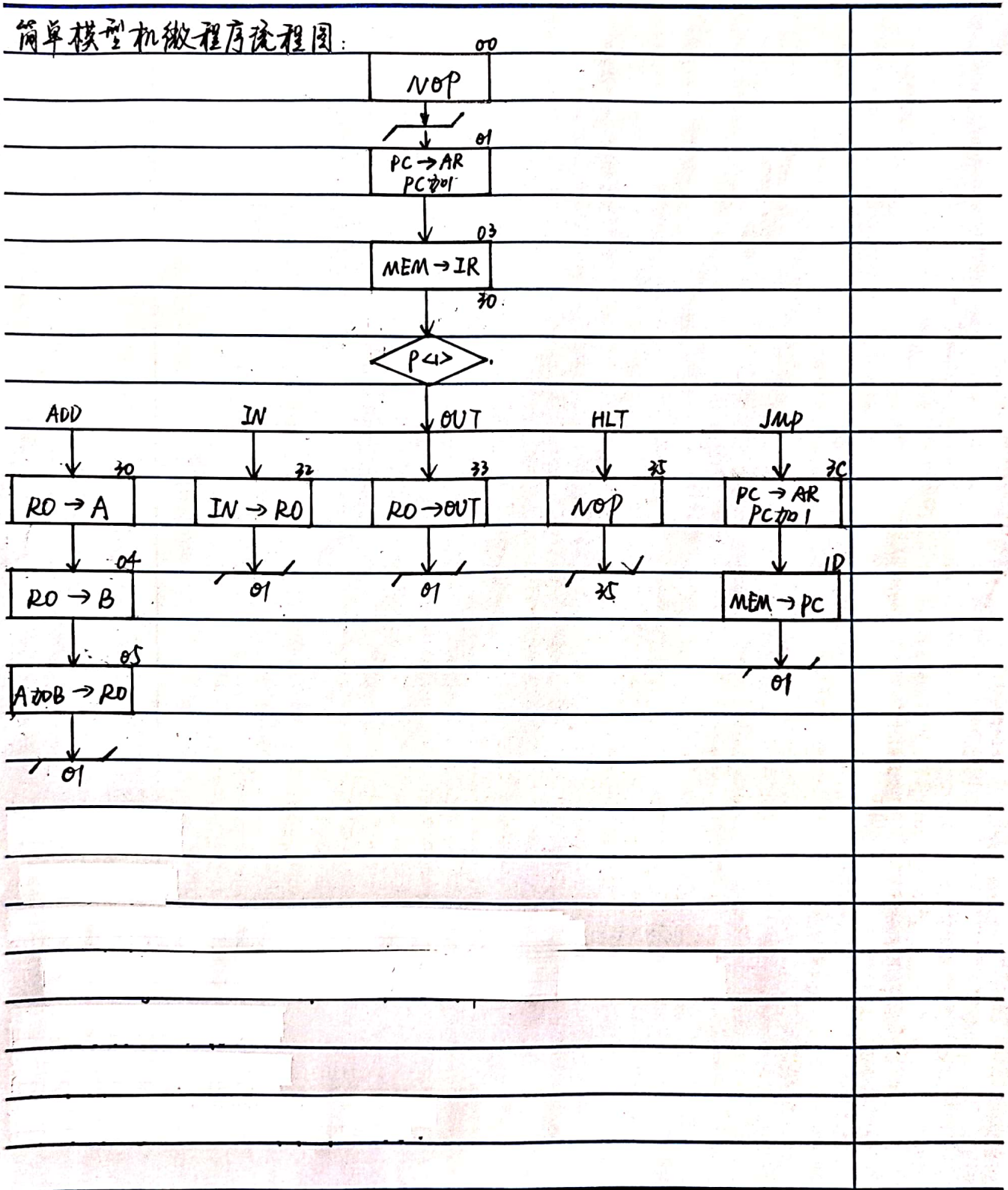


# 同济大学实验报告纸

专业\_\_\_\_ 届\_\_\_\_ 班\_\_\_\_ 姓名\_\_\_\_ 第\_\_\_\_ 组 同组人员\_\_\_\_

课程名称\_\_\_\_ 实验名称\_\_\_\_ 实验日期\_\_\_\_ 年\_\_\_\_ 月\_\_\_\_ 日

简单模型机微程序流程图:





# 同济大学实验报告纸

专业\_\_\_\_ 届\_\_\_\_ 班\_\_\_\_ 姓名\_\_\_\_ 第\_\_\_\_ 组 同组人员\_\_\_\_

课程名称\_\_\_\_ 实验名称\_\_\_\_ 实验日期\_\_\_\_ 年\_\_\_\_ 月\_\_\_\_ 日

二进制微代码表

地址	十六进制	高五位	S3-S0	A字段	B字段	C字段	MA5-MA0
00	00 00 01	00000	0000	000	000	000	000001
01	00 6D 43	00000	0000	110	110	101	000011
03	10 70 70	00010	0000	111	000	001	110000
04	00 24 05	00000	0000	010	010	000	000101
05	04 B2 01	00000	1001	011	001	000	000001
1D	10 51 41	00010	0000	101	000	101	000001
30	00 14 04	00000	0000	001	010	000	000100
32	18 30 01	00011	0000	011	000	000	000001
33	28 04 01	00101	0000	000	010	000	000001
35	00 00 35	00000	0000	000	000	000	110101
3C	00 6D 5D	00000	0000	110	110	101	011101

实验程序:

机器指令格式: \$P XX XX

机器指令标志 十六进制地址 机器指令代码

微指令格式: \$M XX XXXXXX

微指令标志 十六进制地址 微指令地址

机器指令:

\$P 00 20 ; 从IN单元读入数据送入R0 (IN)

\$P 01 00 ; R0与自身相加,结果送R0 (ADD)

\$P 02 30 ; R0的值送OUT单元显示 (OUT)

\$P 03 ED ; 跳转至00H地址 (JMP addr)

\$P 04 00

\$P 05 50 ; 停机 (HLT)



# 同济大学实验报告纸

专业\_\_\_\_ 届\_\_\_\_ 班\_\_\_\_ 姓名\_\_\_\_ 第\_\_\_\_ 组 同组人员\_\_\_\_

课程名称\_\_\_\_ 实验名称\_\_\_\_ 实验日期\_\_\_\_ 年\_\_\_\_ 月\_\_\_\_ 日

\$M 00 000001 ; NOP	
\$M 01 006D43 ; PC → AR, PC加1	
\$M 03 107070 ; MEM → IR, P<1>	
\$M 04 002405 ; R0 → B	
\$M 05 04B201 ; A加B → R0	
\$M 1D 105141 ; MEM → PC	
\$M 30 001404 ; R0 → A	
\$M 32 183001 ; IN → R0	
\$M 33 280401 ; R0 → OUT	
\$M 35 000035 ; NOP	
\$M 3C 006D5D ; PC → AR, PC加1	

## [实验内容]

### 1. 五条机器指令

ADD (= 加进加法)

IN (输入)

OUT (输出)

HLT (停机)

JMP (转移)

### 2. 实验步骤

① 连接实验线路

② 分析并理解机器指令和微指令 (见上方实验程序部分)

③ 联机装载程序

选择联机软件的“[转储]—[装载]”功能,再打开文件对话框中选择上面所保存的文件,软件自动将机器程序和微程序写入指定单元。





# 同济大学实验报告纸

专业\_\_\_\_ 届\_\_\_\_ 班\_\_\_\_ 姓名\_\_\_\_ 第\_\_\_\_ 组 同组人员\_\_\_\_  
课程名称\_\_\_\_ 实验名称\_\_\_\_ 实验日期\_\_\_\_ 年\_\_\_\_ 月\_\_\_\_ 日

## ④运行程序

1) KK1, KK3 置为“运行”档, 进入软件界面, 选择菜单命令“[实验]-[简单模型机]”, 打开简单模型机数据通路图。

2) 按动 CON 单元 CLR 按钮, 实现清零

3) 置 IN 单元为“0001 0011”, 按动软件中单步执行, 观察到 IN 单元数据通过总线流入 RO 中

4) 按动单步执行按钮, RO 中数据加载到暂存器 A, 然后再加载到暂存器 B, A、B 中数据相加写回到 RO, 此时 RO 中数据为 0010 0100。

5) 按动单步执行按钮, RO 中数据通过总线流入 OUT 单元输出。

6) 按动单步执行按钮, 执行 JMP 指令, 跳转列入口地址为 32 的分支, 即观察到 IN 单元数据通过总线流入 RO。(可改变 JMP 指令跳转地址)

### (1) 以 OUT 指令为例描述指令执行过程:

从 PC 取指将地址通过总线发送到地址寄存器 AR 中, PC+1, 然后内存将地址存储的指令通过总线发送到指令寄存器 IR 中, 指令寄存器中的二进制指令被 CPU 识别, 由于此时 P<1>测试字段有效, 此时 CPU 经过译码后形成后续微地址入口并产生分支, 执行地址为 33 的微指令, 将 RO 中数据通过总线送到 OUT 单元输出。

### (2) 以 JMP 指令为例描述指令执行过程

从 PC 取指将地址通过总线发送到地址寄存器 AR 中, PC+1, 然后内存将地址存储的指令通过总线发送到指令寄存器 IR 中, 指令寄存器中的二进制指令被 CPU 识别, 由于此时 P<1>测试字段有效, 此时 CPU 经过译码后形成后续微地址入口并产生分支, 执行地址为 3C 的微指令, 将 PC 中的地址送入地址寄存器, PC+1, MEM 中数据流入 PC, 修改 PC 的前值, 下次取指有 3 修改, 从而实现跳转。以实验中的程序为例, 在 JMP 指令结束后, 跳转到的是从 IN 单元读取数据送入 RO, 即 IN 这条机器指令。



# 同济大学实验报告纸

专业\_\_\_\_ 届\_\_\_\_ 班\_\_\_\_ 姓名\_\_\_\_ 第\_\_\_\_ 组 同组人员\_\_\_\_  
课程名称\_\_\_\_ 实验名称\_\_\_\_ 实验日期\_\_\_\_ 年\_\_\_\_ 月\_\_\_\_ 日

## [实验小结]

1. 打开CMA软件界面后所做的工作?

答: 打开CMA软件后选择“[转储]-[装载]”功能, 打开文件对话框选择CPU与简单模型机实验.txt进行装载。其中机器指令装载在内存中, 微指令装载在内存中。

2. 描述IN指令和ADD指令执行过程(单步执行)

答: 单步执行每一步是一条微命令。

IN指令: 从PC取指令地址发送到地址寄存器AR中, PC+1, 然后内存将该地址存储的指令发送给指令寄存器IR中, 指令寄存器中的二进制指令被CPU识别, 由于此时PC10测试字段有效, 此时CPU译码后形成后续微程序入口地址并产生分支, 执行地址为32的微指令, 读取IN单元数据存入通用寄存器R0中。

ADD指令: 从PC取指令地址发送到地址寄存器AR中, PC+1, 然后内存将该地址存储的指令发送到指令寄存器IR中, 指令寄存器中的二进制指令被CPU识别, 由于此时PC10测试字段有效, 此时CPU译码后形成后续微程序入口地址并产生分支, 执行地址为00的微指令, R0中数据加载到暂存器A, R0中数据加载到暂存器B, A与B中数据相加写入通用寄存器R0。

3. 一条机器指令结束标志?

答: 下一条微指令的微地址为01时代表一条机器指令结束。

4. 控制器译码的作用?

答: 使微程序转入相应的微地址入口, 实现对指令的识别并实现微程序的分支。





# 同济大学实验报告纸

专业\_\_\_\_届\_\_\_\_班\_\_\_\_姓名\_\_\_\_第\_\_\_\_组 同组人员\_\_\_\_  
课程名称\_\_\_\_实验名称\_\_\_\_实验日期\_\_\_\_年\_\_\_\_月\_\_\_\_日

5. 从开始到 ADD 指令结束共执行了几条机器指令?

答: 2 条 IN 指令和 ADD 指令

6. 简单修改: 如何将通用寄存器中的数据先到寄存器 B 再到寄存器 A

答: 方法一: 修改地址为 00 的徽指令的 A 字段为 010

修改地址为 04 的徽指令的 A 字段为 001

方法二: 修改接线: MC 单元 LDA 接 ALU & REG 单元 LDB

MC 单元 LDB 接 ALU & REG 单元 LDA

## [课程总结]

通过一学期计算机组成原理实验的学习, 我收获颇多, 在前半学期数字逻辑的学习中, 我对组合逻辑和时序逻辑有了了解, 对译码器、选择器、加法器和触发器、计数器的应用有了掌握, 让我感受到了数字逻辑的魅力, 并通过自行设计抢答器对这些知识的应用有了更深的认识。后半学期的组成原理实验让我对计算机内部的大部件之间的工作有了进一步理解, 从存储器到运算器到微程序控制器到总线再到最后的模型机设计实验, 使我对一台计算机的工作有了更深的认识, 为今后学习计算机的有关知识打下了良好的基础, 这门课不仅教会我数字逻辑与组成原理的知识, 更重要的是分析问题、解决问题及迁移知识并应用的能力, 这将会对我以后的学习和工作起到了很好的铺垫作用。

