

同济大学实验报告纸

2353814

软件工程 专业 2027届 1 班 姓名 马小龙 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 2024年12月12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的运行过程

【实验设备】

组成原理实验箱 TD-CMA

【实验原理】

1. CPU的组成结构及工作原理

① CPU的组成

① 运算器 ALU：完成数据计算或处理，需要计算的数据从寄存器 A、B 中获取

② 微程序控制器 MC：控制机器指令的分析和译码

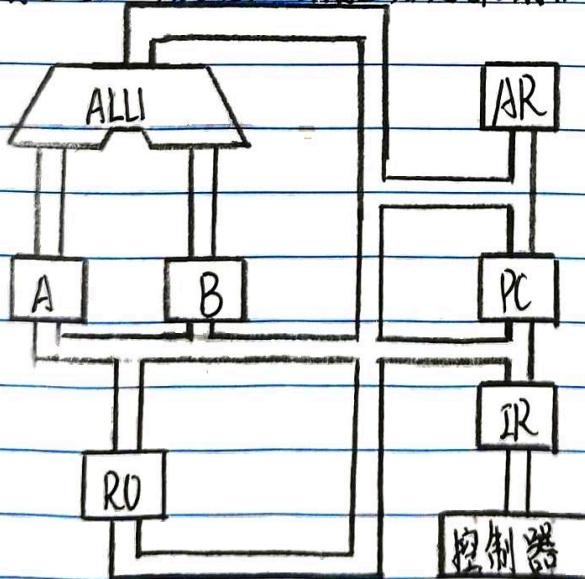
③ 寄存器：用于暂存数据或指令

通用寄存器 RD：用于临时存放中间数据

程序计数器 PC：用于存放下一条指令或数据的地址

地址寄存器 AR：用于存放当前指令或数据的地址

指令寄存器 IR：用于存放当前正在执行的机器指令



基本CPU构成原理图

2) CPU的基本功能：读取并执行指令

3) CPU需要完成的工作：

① 取指令，读主存，装入寄存器（公共）

② 分析指令：指令译码，决定接下来的操作

③ 执行指令

· 取数据：从主存或 I/O 读取操作数，如有

· 处理数据：对操作数进行算术或逻辑运算

· 写数据：将执行结果写到主存或 I/O 设备

2. 机器指令的执行过程

一条机器指令从取出到执行结束，由相应的微指令序列完成。本实验中，微程序控制器通过指令译码产生一系列微指令，微指令序列有次序地控制各个部件的工作状态，实现指令的执行过程。

3. 微指令格式与微代码表

本实验新增了三个部件，包括 PC, AR, MEM，对 A、B、C 寄存器进行了相应功能调整，具体微指令格式如下：

23	22	21	20	19	B-15	14-12	11-9	8-6	5-0
M23	M22	WR	R1	IOM	S3-S0	A字段	B字段	C字段	MA5-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	ALUB	0	0	1	R1>
0	1	0	LDI	0	1	0	RD_B	0	1	0	保留
0	1	1	LDRI	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	LDAD	1	0	1	保留	1	0	1	LDPC
1	1	0	LDAR	1	1	0	PCB	1	1	0	保留
1	1	1	LDIR	1	1	1	保留	1	1	1	保留

微指令格式表

同济大学实验报告纸

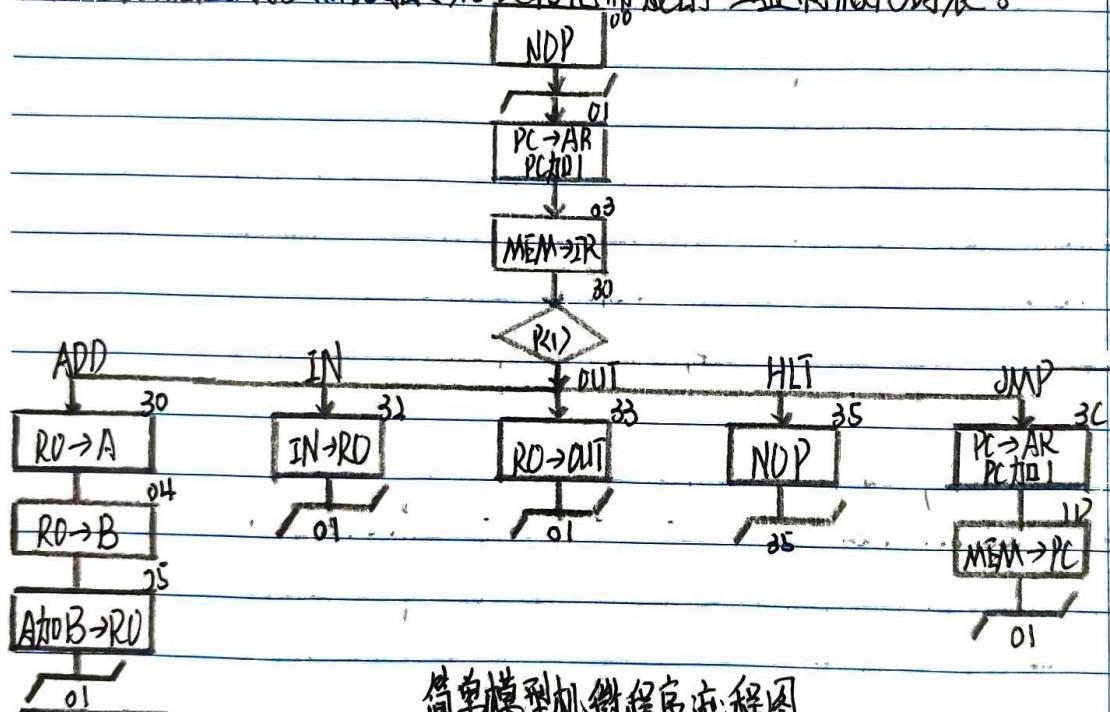
2353814

软件工程 专业 2021 届 班 姓名 马小龙 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 2024 年 12 月 12 日

当拟定“取指”微指令时，该微指令的判别测试字段为 R11 测试，由于“取指”微指令是所有微程序都使用的公共微指令，因此 R11 测试结果出现多路分支。本实验机器使用指令寄存器的高 6 位 (IR1-IR2) 作为测试条件，出现 5 路分支，占用 5 个固定的微地址单元，剩余其他地方就可以使用一条微指令用控存的一个微地址单元的方法填写。

当全部微程序设计完成后，应将每条微指令代码化。以下为微程序流程图和微程序流程图按照微指令格式转化而成的“二进制微代码表”。



简单模型机微程序流程图

地址	十六进制	高五位	S3-S0	A字节	B字节	C字节	MA5-MA0
00	00 00 01	00000	0000	000	000	000	000001
01	00 61 43	00000	0000	110	110	101	000011
03	10 10 70	00010	0000	111	000	000	110000
04	00 24 05	00001	0000	010	010	000	100101
05	04 B2 01	00000	1001	011	101	000	001001
10	10 51 H1	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	1010	000	000001
35	00 00 35	00000	0000	000	000	000	110101
3C	00 61 5D	00000	0000	110	110	101	011101

二进制微代码表

3. 实验中的五条机器指令

实验共有5条机器指令：IN（输入）、ADD（二进制加法）、OUT（输出）、JMP（无条件转移）、HLT（停机），其指令格式如下。

助记符	机器指令码	说明
IN	0010 0000	IN \rightarrow RD
ADD	0000 0000	RD + RD \rightarrow RD
OUT	0011 0000	RD \rightarrow OUT
JMP	1110 0000 **** ****	addr \rightarrow PC
HLT	0101 0000	停机

其中JMP为双字节指令，其余均为单字节指令，****为addr对应的二进制地址码。微程序控制器实验的指令是通过手动给出的，现在要求CPU自动从存储器读指令并执行。

4. CPU与简单模型机实验指令文件

联机软件提供了微程序和机器程序下载功能，以代替手动读写微程序和机器程序，微程序和机器程序的格式如下：

机器指令格式：

微指令格式

\$P XX XX

\$M XX XXXXXX

机器指令代码

微指令代码

十六进制地址

十六进制地址

机器指令标志

微指令标志

本实验程序如下（“；”后为注释内容）

; // Start of Main Memory Data //

\$P 00 20 ; START: IN RD //IN单元读入数据并送入RD

\$P 01 00 ; ADD RD, RD //RD和自身相加，结果送入RD

\$P 02 30 ; OUT RD //RD中的内容送入OUT单元并显示

\$P 03 ED ; JMP START //跳转至0DH地址

\$P 04 00 ;

\$P 05 SD ; HLT //停机

; // End of Main Memory Data //

同济大学实验报告纸

2353814

软件工程 专业 2021 届 1 班 姓名 张小虎 第 1 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 2024 年 12 月 12 日

; // Start of MicroController Data[]

\$M 00 000001 ; NOP

\$M 01 006143 ; PC→AR, PC加1

\$M 03 107070 ; MEM→IR, P<1

\$M 04 002405 ; RD→B

\$M 05 04B201 ; A加B→R0

\$M 10 105141 ; MEM→PC

\$M 30 001404 ; RD→A

\$M 32 183001 ; IN→RD

\$M 33 280401 ; RD→OUT

\$M 35 000035 ; NOP

\$M 3C 006050 ; PC→AR, PC加1

; // End of MicroController Data[]

选择联机软件的“[转储]-[装载]”功能，在打开文件对话框中选择上面文件，软件自动将机器程序和微程序写入指定单元，选择联机软件的“[转储]-[刷新指令区]”可以读出下位机所有的机器指令和微指令，并在指令区内显示。

实验内容】

1. 五条机器指令

① ADD 二进制加法 ② IN 输入 ③ OUT 输出 ④ HLT 停机 ⑤ JMP 转移

2. 实验步骤

① 分析并理解机器指令和微指令

② 连接实验线路

③ 联机装载微程序和机器指令

④ 手动运行或联机运行，观察指令执行过程

本机手动运行：

将时序与操作台单元的开关KK1, KK3置为“运行”档，按动CON单元的总清按钮CLR，将程序计数器PC，地址寄存器AR和微程序地址寄存器内容置为00H，使程序可以从头开始运行，暂存器A,B，指令寄存器IR和DUT单元也会被清零。

将时序与操作台单元的开关KK2置为“单步”档，每按动一次ST按钮，即可单步运行一条微指令，对照微程序流程图，观察微地址显示灯是否和流程一致。每运行完一条微指令，观测一次CPU内总线和地址总线，对照数据通路图，分析总线上的数据是否正确。

当模型机执行完JMP指令后，检查DUT单元显示的数据是否为IN单元输入的值的倍，按下CON单元的总清零按钮CLR，改变IN单元的值，再次执行机器程序，并判断程序执行是否正确。

② 联机运行

将时序与操作台单元的开关KK1, KK3置为“运行”档，进入软件界面，选择菜单命令“[实验]-[简单模型机实验]”，打开简单模型机数据通路。

按动CON单元总清按钮CLR，然后通过软件运行程序，选择相应功能键，即可联机运行、监控、调试程序，当模型机执行完JMP指令后，检查DUT单元显示的数据是否为IN单元的倍，在数据通路和微程序中观测指令的执行过程，并观测软件中地址总线，数据总线以及微指令显示和下机位是否一致。

实验考核

问题一：微地址为01和3C的微指令对应的操作均为 $(PC \rightarrow AR, PC+1)$ ，那么这两条微指令的内容是否相同，为什么？

答：不相同。这两条微指令的下地址字段不同，微地址为01的微指令下地址字段为03，微地址为3C的微指令下地址字段为1D。

问题二：将MEM中地址为04的存储单元的内容由00改为01，会产生怎样的影响？

答：MEM中地址为04的存储单元中的内容为JMP指令第二个字节的内容，为跳转的目標地址，将其由00改为01后，下一条执行的指令为加法指令ADD而不是原来的输入指令IN。
实际是

问题三：在实验中，某同学发现执行加法指令时，先将通用暂存器RD中的数据存入暂存器B，再将暂存器RD中的数据存入暂存器A，而不是理论上的“先A后B”。

同济大学实验报告纸

2353814

专业 2027 届 1 班 姓名 马小龙 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 2024 年 12 月 12 日

造成这种现象的原因是什么？

答：有两种可能。第一种可能是装载的程序被修改了，即将微地址为30的微指令和微地址为04的微指令的A字段进行了修改，使得微地址为30的微指令A字段由001变为010，功能由LDA变为LDB；微地址为04的微指令A字段由010变为001，功能由LDB变为LDA。由于微程序执行过程中，微地址为30的微地址先执行，所以功能由先向暂存器A中存数据变为先向暂存器B中存数据。虽然这种情况存在，但出现的概率并不大，更有可能的情况是第二种，即该同学将MC单元与ALU®单元间LDA与LDB的连接反了，即MC单元的LDA与ALU®单元的LDB相接；MC单元的LDB与ALU®单元的LDA相接。

问题四：构建任意进制计数器的三个方法是什么？

答：反馈清零法，反馈置数法，级联法

【实验小结】

本简单模型机设计实验是计算机组成原理部分的实验的综合与应用，在前实验的基础上连接实验线路，观察指令的执行过程及对应微程序的运行。

每条机器指令对应一般微程序，这个微程序包含一系列微指令，一条微指令包含多个微命令，每条微命令对应一种微操作。在本次实验 ADD, IN, OUT, JMP, HLT 等机器指令执行过程中，微程序首先控制实现取指过程，对应 $PC \rightarrow AR$, $PC + 1 \rightarrow IR$ 等操作，然后控制器对指令译码，微程序根据译码结果进行分支，产生对应入口地址，继续实现机器指令对应功能。

本次实验中我在使用实验软件进行调试的能力有了进一步提升，同时可以准确掌握各个指令的执行过程，进一步提高了我的实验操作能力和实验素养。

【实验课程学习心得】

在入门课程的实验过程中，最先接触到的是组合逻辑实验部分。起初，我对于各种逻辑门和布尔代数相关的内容还相对陌生，对于很多抽象的概念与公式均不太熟悉，很难想象它们应当如何在实际电路中发挥作用。然而，当我学习并使用各种门电路、译码器、选择器、数码显示器等相关芯片，尤其是在自己动手实验验证这些芯片的功能时，这些晦涩的理论仿佛变得“鲜活”了起来，我逐渐理解了逻辑设计的美妙所在。

随着对数字逻辑学习的不断深入，我开始学习时序逻辑实验的相关内容。在此阶段，我主要学习了触发器和计数器两大部件。在使用74LS74构成二、四分频电路中，我深刻理解了D触发器的工作机制；在构成任意进制计数器实验中，我学习到了反馈清零法、反馈置数法和级联法三种常用方法。在每一次的构建中，常常需要反复推敲和实验，在抽象和具体之间来回切换，虽然耗心力，却也让人沉浸其中。

在期中之际，我们进行了数字逻辑设计实验，这是对前半学期所学知识的总结与归纳。我选择了电子锁作为设计实验实现的内容。在本次设计实验中，我多次使用了D触发器用于锁存输出状态，让我深刻理解了D触发器在时序电路进行控制方面的重要功能。同时，在本次设计实验中，我也认识了74LS192四位同步加法/减法计数器和74LS194四位双向移位寄存器两种新的芯片，拓展了我的知识面。本次设计实验使我收获颇丰。

在数字逻辑奠定的基础之上，组成原理部分实验让我进一步看到了一个完整的计算机系统全貌。在计算机系统认识实验中，我熟悉了实验室设备的基本操作，理解了一些基本概念，包括总线体系以及计算机系统结构和时序发生器的工作原理。在之后的实验中，我逐渐理解了存储器、运算器、控制器等部件的工作原理，总线中的数据流动以及各个控制信号的作用，不仅对于相关概念，原理有了深入理解，并且提高了自己的实验动手能力，并最终完成简单模型机设计实验，理解了指令、微程序执行过程。这部分内容与理论课程联系紧密，通过学习与实践，我深刻理解了各种概念与知识，为理论课学习提供了便利。

回顾整个实验历程，数字逻辑部分培养了我更严谨的思维方式，对于故

同济大学实验报告纸

2353814

软件工程 专业 2021 届 1 班 姓名 张小龙 第 1 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 2024 年 12 月 12 日

故障排除、设计电路都有了初步经验；组成原理实验让我深刻理解了计算机各个部件之间的协调、配合，深刻认识到在硬件设计和软件开发的相互配合下，计算机在保持简单结构的同时可以展现出卓越的功能和性能。毫无疑问，本学期的实验经历激发了我对计算机相关知识更深层次的热爱。

在这段学习和实践中，我格外感谢张老师的悉心教导、细致引导和耐心帮助。无论是在概念讲解、实验思路指导还是在解决调试过程中遇到的问题，老师始终循循善诱，为我们指点迷津。尤其是在解决实验中的问题时，您带领我从错误之处逐步寻找源头，不断提问让我思考，而不是直接抛出答案。您的每一次讲解、每一次提问，都拓宽我的思路，让我对知识有了更深的理解，深深地影响着我。

本学期学习的内容较为丰富，让我认识到计算机的基本工作原理。未来，我希望可以借助现有知识，进行进一步深入学习，例如操作系统、如何将高级语言翻译为机器指令来实现一个程序的功能，以及计算机相关的各种协议等。逐步将所学硬件知识与软件相结合，提升自己的软件设计素养能力。

补充：

机器指令执行过程

1. IN 指令

① PC 将指令地址传给 AR，同时 PC+1；② 从 MEM 中取出 AR 中地址对应的存储单元中的数据，即 IN 指令，并传给指令寄存器 IR，控制器进行指令译码，微程序转入相应的微地址入口，实现微程序分支；③ 控制器控制 IN 单元数据流入 RD 中；

2. ADD 指令

① PC 将指令地址传给 AR，同时 PC+1；② 从 MEM 中取出 AR 中地址对应的存储单元中的数据，即 ADD 指令，并传给 IR，控制器进行指令译码，微程序转入相应的微地址入口，实现微程序分支；③ DC 将数据传给暂存器 A；④ RD 将数据传给暂

在器B；⑤在运算器中对A,B中的数据进行加法运算并将结果传入RD中。

3. DLT 指令

①PC将指令地址传给AR，同时PC+1；②从MEM中取出AR中地址对应存储单元的数据，即DLT指令，并传入IR中，控制器进行指令译码，微程序转入相应的微地址入口，实现微程序分支；③RD将数据传送给DLT单元，并在数码显示器管显示。

4. JMP 指令

①PC将指令地址传给AR，同时PC+1；②从MEM中取出AR中地址对应存储单元的数据，即JMP指令，并传入IR中，控制器进行指令译码，微程序转入相应的微地址入口，实现微程序分支；③PC将当前地址传给AR，同时PC+1；④从MEM中取出AR中地址对应存储单元的数据并传入PC，实现跳转。

5. HLT 指令

⑥PC将指令地址传给AR，同时PC+1；⑦从MEM中取出AR中地址对应存储单元的数据，即HLT指令，并传入IR中，控制器进行指令译码，微程序转入相应的微地址入口，实现微程序分支；⑧由于该入口地址对应的微指令的下地址仍为该入口地址，该条微指令被反复执行，不进行任何操作，实现停机。

机器指令和微指令的对应关系

一条机器指令对应一个微程序，一个微程序对应一系列微指令。这就意味着一个机器指令所完成的操作可以划分为若干微指令来完成，由微指令进行解释和执行。微指令首先控制实现机器指令的“取指”操作，微程序控制器根据取出的机器指令进行译码，微程序根据译码结果转入相应的微地址入口，实现微程序分支，以此来决定后续需要执行的微指令序列，来实现机器指令的相关功能和机器指令的结束或停机。总的来说，微指令序列通过有次序地控制各个部件的工作状态，实现机器指令的执行过程。

具体的对应关系可以参见简单模型机微程序流程图。[实验原理部分]