

上海大学 计算机学院

《计算机组成原理实验》报告十三

姓名 严昕宇 学号 20121802

时间 2022. 05. 31 机位 19 指导教师 刘跃军

实验名称: 建立指令流水系统

一、实验目的

1. 了解指令流水系统的设计方式
2. 编制一条可以流水方式运行的指令

二、实验原理

1. 硬件件的并行工作

当一条微指令的子操作使用的硬件（包括总线）互不相同时，可以同时工作。这一特点也表现在微指令编码上，就是子操作的微指令码中为低电平（有效）的都不相同。于是可以将子操作的微指令码合并成一个微指令。

2. 指令流水执行

指令的流水线执行模式下，同一时间有多条指令各自在不同的硬件中执行，而对同一条指令而言，不同时间顺序在不同的硬件中执行。

要形成指令流水模式，每条指令都应该分成几个独立的子操作，当前趋指令的后几个子操作与后继指令的前几个子操作不使用同样的硬件时，系统就可设计成流水线方式。

现代计算机大都采用指令流水模式，但这个模式会使中断响应过程变得复杂，所以实时系统中多是有限地采用它。

3. 实验箱系统的指令流水硬件基础

实验箱系统中的很多操作可以在不同的硬件中同时执行，典型的是“取指令”的微操作，其微指令码微 CBFFFF，与大多数的微操作无关。在厂家给的默认指令系统中这个操作编在了每条指令的最后一个状态，即每条指令的操作完成后就取进下一条指令。这是典型的“取指、执行、取指……”模式。即一条指令先被“取指”，再执行其他微操作，完成后再取下一条指令。

如果一条指令的最后一个微操作与取指无关，就可以把二者合并成一个微指令，于是这个指令的最后一个微操作与取下一条指令并行进行。对下一条指令而言，其“取指”与“其他操作”在不同硬件中顺序执行——指令二级流水。

三、实验内容

1. 实验任务一（分析流水指令集 insfile2.MIC）

(1) 实验步骤

在 CP226 环境，按以下步骤操作，建立相应文件：

- ① 在下拉菜单中选择：文件→打开指令系统/微程序。
- ② 在出现的对话框中选择目录 C:\Program Files\CP226 计算机组成原理\data\ 下的 insfile2.MIC 调入系统
- ③ 在指令系统窗口逐条观察指令的各微指令码，与 insfile1.MIC（指导书 103 到 110 页）的对应指令的微指令码比较，分析二者的异同

(2) 实验现象

通过对比，发现 insfile2.MIC 为流水指令集，与 insfile1.MIC 存在相同与差异之处。

(3) 数据记录、分析与处理

指令 ADD A, #*

insfile1.mic			
ADD A, #*	T2	1C	C7FFEF
	T1	1D	FFFE90
	T0	1E	CBFFFF
		1F	FFFFFF

insfile2.mic			
ADD A, #*	T2	1C	C7FFEF
	T1	1D	CBFE90
		1E	FFFFFF
		1F	FFFFFF

通过对比可以发现，insfile1.mic 的指令 ADD A, #* 中有三条有效微指令，而 insfile2.mic 中仅有两条有效微指令。

其原因是：由于微指令 FFFE90 与 CBFFFF 中的 CB 与 FF 没有相同的 0，因此 insfile2.mic 的 ADD A, #* 的取值微指令 CBFFFF 与其前面的微指令 FFFE90 已经合并，即为 CBFE90。

指令 MOV A, R?

insfile1.mic			
MOV A, R?	T1	70	FFF7F7
	T0	71	CBFFFF
		72	FFFFFF
		73	FFFFFF

insfile2.mic			
MOV A, R?	T1	70	CBF7F7
	T0	71	FFFFFF
		72	FFFFFF
		73	FFFFFF

指令 ADD A, #*中相类似的，insfile2.mic 的 MOV A, R?的取值微指令 CBFFFF 与其前面的微指令 FFF7F7 已经合并，即为 CBF7F7。

(4) 实验结论

通过与指导书的对比，成功分析流水指令集 insfile2.MIC 与 insfile1.MIC 的差异之处。

完成实验任务一的目的。

2. 实验任务二（改造实验十二中自己编制的指令集，使其中至少一条指令成流水方式）

(1) 实验步骤

在 CP226 环境，按以下步骤操作，建立相应文件：

- ① 在下拉菜单中选择：文件→打开指令系统/微程序。
- ② 在出现的对话框中选择目录 C:\Program Files\CP226 计算机组成原理\data\
- ③ 在 data 目录中选择 z12.mic 文件。
- ④ 系统在屏幕的“源程序”窗口中打开选中的程序。
- ⑤ 在源程序窗口中改造实验十二中自己编制的指令集，使其中至少一条指令成流水方式。

z13.mic			
FATCH	T0	00	CBFFFF
		01	FFFFFF
		02	FFFFFF
		03	FFFFFF
LD A, #*	T1	04	C7FFF7
	T0	05	CBFFFF
		06	FFFFFF
		07	FFFFFF
A-W A, #*	T2	08	C7FFE7
	T0	09	CBFE91
		0A	FFFFFF
		0B	FFFFFF
跳到	T1	0C	C6FFFF
	T0	0D	CBFFFF
		0E	FFFFFF
		0F	FFFFFF
OUTA	T0	10	CBDF9F
		11	FFFFFF
		12	FFFFFF
		13	FFFFFF

延时	T3	14	FFFFFF
	T2	15	FFFFFF
	T1	16	FFFFFF
	T0	17	CBFFFF

z13.dat		
LD A, #*	04	2
A-W A, #*	08	2
跳到	0C	2
OUTA	10	1
延时	14	1

z13.mac	
FATCH	000000xx
LD A, #*	000001xx
A-W A, #*	000010xx
跳到	000011xx
OUTA	000100xx
延时	000101xx

⑥ 删除原来程序的多余符号，以 z13 和正确后缀存储新编制的文件到 D 盘。

(2) 实验现象

在之前的程序中，尝试使用上述五个助记符替换原助记符。程序正常运行，即完成了对原助记符的替换。

且通过改造实指令集，使其中的部分指令成流水方式

(3) 数据记录、分析与处理

助记符	机器码	指令意义描述
FATCH	000000xx	实验机占用，不可修改，复位后，所有寄存器清 0 (IR 除外)，首先执行 _FATCH_ 指令取指
LD A, #*	000001xx	将立即数打入累加器 A
A-W A, #*	000010xx	累加器 A 减立即数
跳到	000011xx	无条件跳转指令
OUTA	000100xx	累加器 A 输出到 OUT
延时	000101xx	延长显示时间

(4) 实验结论

成功改造实验十二中自己编制的指令集，使其中部分指令成流水方式。完成实验任务二的目的。

3. 实验任务三（在自己编制的两个指令集中运行同一个程序，观测运行情况和效率。程序来源自定）

(1) 实验步骤

- ① 打开实验箱电源，检验实验箱的基本功能是否正常。确认无误后，启动 CP226 软件。连接 PC 机与实验箱的通信口 COM4。
- ② 在 CP226 汇编语言程序集成开发环境下编写如下汇编程序，即实验十二中的试验任务二“OUT 寄存器交替显示 55、22 和 55-22 的值”：

```
LOOP: LD A, #55H
      OUTA
      延时
      延时
      延时
      延时

      LD A, #22H
      OUTA
      延时
      延时
      延时
      延时

      LD A, #55H
      A-W A, #22H
      OUTA
      延时
      延时
      延时
      延时

      跳到 LOOP
      END
```

- ③ 保存文件，文件扩展名为.asm。
- ④ 编译并下载源程序至实验箱，调试并运行程序。在自己编制的两个指令集中运行同一个程序，观测运行情况和效率，记录实验结果。

(2) 实验现象

OUT 寄存器交替显示 55、22 和 55-22 的值。

通过对比，可以发现流水指令集 z13 的运行效率更高。

(3) 数据记录、分析与处理

汇编程序的具体分析如下所示

LOOP: LD A, #55H	将立即数 55H 打入累加器 A
OUTA	累加器 A 输出到 OUT
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间
LD A, #22H	将立即数 22H 打入累加器 A
OUTA	累加器 A 输出到 OUT
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间
LD A, #55H	将立即数 55H 打入累加器 A
A-W A, #22H	累加器 A 减立即数 22H
OUTA	累加器 A 输出到 OUT
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间
跳到 LOOP	无条件跳转程序，转跳 LOOP
END	汇编程序终止

(4) 实验结论

成功利用 CP226 软件编写汇编程序，并使用实验十二与实验十三实验任务一所改变的指令集，实现了 OUT 寄存器交替显示 55、22 和 55-22 的值。且通过对比，观测得到不同指令集的运行情况和效率。

完成实验任务二的目的。

四、建议和体会

体会

本次实验是建立指令流水系统，要我们用实验十二编制的一个汇编指令系统，进行合并，把取指操作 CBFFFF 和前一个没有相同的低电平，即没有相同的位为 0 的指令合并，形成二级流水模式。然后运行程序，完成 OUT 寄存器交替显示 55，22，和 55-22 的值。对比实验十二明显加快。得出结论，流水指令系统可以加快指令集速度。其实早在在这个实验之前，我就自己在写汇编程序的时候把两个指令合并，没想到里面蕴含了流水的道理。

通过视频讲解与实验演示，我了解到：指令流水为提高处理器执行指令的效率，把一条指令的操作分成多个细小的步骤，每个步骤由专门的电路完成。而且

要使流水线具有良好的性能，必须设法使流水线能畅通流动，即必须做到充分流水，不出现断流。通过观察、分析流水指令集与一般指令集的区别，以及使用之前编写的程序对比其运行效率，我真切领悟到计算机流水技术的高效。

由于疫情的影响，本次实验是居家完成。通过观看线上的实验讲解视频与学习实验内容 PPT，并未能亲身操作实验箱和 CP226 软件来进行实验，这会在一定程度上影响学习效果，因此需要课前高质量的预习与课后的总结以加深理解。

五、思考题

计组实验课接近尾声，请你对该课程的授课形式、实验内容等提出你的建议

答：

计算机组成原理实验是我大学的第二门专业实验课。和数字逻辑实验比起来，计算机组成原理实验难了不少。个人认为，计算机组成原理与计算机组成原理实验是计算机专业所有课程中最重要的。因为在本课程中，通过应用数字逻辑、离散数学的知识，可以设计一台模拟人类思维的机器，这为计算机领域的一切发展奠定了基础。而“实践是检验真理的唯一标准”，计算机组成原理实验则是理论课的实践。

有人可能会问，学习这课程有什么用呢？在我看来，对于芯片开发或者系统级部件开发而言，这部分知识可以为我们提供工程实践的基本理论与实践支持，同时，该门课程也是整个计算机体系结构的基础，掌握好这门课程对于我们从事计算机软硬件方面的工作非常重要。即使将来从事纯软件开发，如果通过实验了解底层的运行原理、工作逻辑，可以设计出更适配的高质量代码，提升效率。

更重要的是，通过本课程的学习，可以掌握不少学习的方法。在计算机组成原理实验课程学习过程中，通过同学间互相探讨，我逐步提高了发现问题、分析问题和解决问题的能力；同时，通过撰写这篇报告，我也提高了文字编辑、排版等方面的能力。

当然实验课接近尾声，我想对该课程的授课形式、实验内容提出自己的建议。从一名学生的角度来看，计算机组成原理实验课带给我从理论走向实践的关键步骤，让我通过在实验箱上动手操作，感受计算机的精密美妙。但是，实验课其内容有时与书本知识的结合程度不是最好，有时存在实验内容超前于课本讲授的情况。因此在我看来，如果两者能同步进行，能通过实验课帮助我更好理解计算机组成原理。

最后，在此我想感谢刘老师的认真教授，帮助我能通过实验，更好掌握课程的知识点。也感谢同学们对我的帮助。在大家的帮助下，我顺利地完成了计算机组成原理实验(2)课程的学习！