

# 上海大学 计算机学院

## 《计算机组成原理实验》报告十三

姓名 赵文宇 学号 23121769  
时间 周三 9-11 机位 8 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、1. 硬部件的并行工作

在实验 3 的“实验过程举例”中我们用一条微指令完成了“A+W”后右移一位的值送 OUT 的操作，这个操作明显地可以分成三个子操作:A 十 W、把“和”右移一位、把值送 OUT。这三个子操作为什么可以同时进行呢?因为这三个子操作使用的硬件(包括总线)互不相同，于是可以同时工作。这一特点表现在微指令编码上，就是这三个子操作的微指令码中为低电平(有效)的都不相同，于是可以将这三个子操作的微指令码合并成一个微指令，就是实验 3 谈到的结果：

c23~c0=11111111 1101 11 11 1011 1000=fdfb8H 这个微指令控制三部分硬件并行工作。

#### 2、指令流水执行

把“使用不同硬件的操作可以同时工作”的概念推广到相继的两条指令之间，就形成“指令的流水线执行模式”。

这个模式下，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，很像在流水型生产线上的产品，不同时刻顺序在不同的工位上加工。这就是指令流水模式的名称来源。

显然，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

实验箱系统的各种基本操作我们已经熟悉，其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，其微指令码为 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。

#### 4、实验箱系统实现指令流水的技巧

由背景知识 3 可知：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。这个原则用微操作码表述就是：若取指令操作 (CBFFFF) 与它前面的微操作码没有相同的位为 0，则这两个微操作码的“与”就是二者合并后的微指令。在程序中这条指令就会和它的后继指令形成二级流水模式。

例如：指导书 108 页的 IN 指令有两个状态 T1 和 T0，T1 状态的微指令码为 FFFF17，T0 状态的微指令码为 CBFFFF，二者为 0 的位都不相同，故 IN 指令可以被改造为流水型指令“CBFF17”——T1 和 T0 微指令码的“与”运算值。

实验箱提供的一个二级流水模式指令示例 insfile2.MIC 就遵循这一原则。

### 三、实验内容

1. 实验任务一（1、分析流水指令集 insfile2.MIC。2、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。3、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。）（由于三个实验一起验收，所以写在一起了）

指令助记符	指令意义描述
LD A,#*	将立即数打入累加器A
A-W A,#*	累加器A减立即数
跳到 *	无条件跳转指令
OUTA	累加器A输出到OUT
延时	延长显示时间

上图为上一次实验的要求，这一次的实验差不多，只需要更改一些指令即可。

#### (1) 实验步骤

先对任务进行分析：

可以翻看上一次已经编好的.mic、.mac、.dat 文件，只需要更改.mic 文件中的一些指令，将其改为 CBxxxx 即可，具体修改过程可参照老师上课所举的例子及 ppt 上后面的介绍。需要更改代码的指令为：A-W A,##\*、OUTA 与延时。分别改为：

T1 08 C7FFE9	T0 10 CBDF9F	T3 14 CBFFFF
T0 09 CBFEE91	11 FFFFFFF	T2 15 FFFFFFF
0A FFFFFFF	12 FFFFFFF	T1 16 FFFFFFF
0B FFFFFFF	13 FFFFFFF	T0 17 FFFFFFF
<u>A-W A,##*</u>	<u>OUTA</u>	<u>延时</u>

至此，即可完成该操作。

下面是操作过程：

操作如下：

. DAT 文件修改如下：

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

图 1. .DAT 文件修改后

. MAC 文件修改如下：

[FATCH_	000000xx
LD A,#*	000001xx
A-W A,#*	000010xx
跳到*	000011xx
OUTA	000100xx
延时	000101xx

图 2 .MAC 文件修改后

. MIC 文件修改如下：

```
_FATCH_ T0 00 CBFFFF
          01 FFFFFFF
          02 FFFFFFF
          03 FFFFFFF
LD A,#*   T1 04 C7FFF7
          T0 05 CBFFFF
          06 FFFFFFF
          07 FFFFFFF
A-W A,#*   T1 08 C7FFEF
          T0 09 CBFE91
          0A FFFFFFF
          0B FFFFFFF
跳到*    T1 0C C6FFFF
          T0 0D CBFFFF
          0E FFFFFFF
          0F FFFFFFF
OUTA     T0 10 CBD9F
          11 FFFFFFF
          12 FFFFFFF
          13 FFFFFFF
延时     T3 14 CBFFFF
          T2 15 FFFFFFF
          T1 16 FFFFFFF
          T0 17 FFFFFFF
```

图 3 . MIC 文件修改后

运行的程序我选择的是上次实验实现的交替显示 55、22、55-22 的程序，只需要将 55 打入 A 寄存器，送到 OUT，延时，将 22 打入 A 寄存器，送到 OUT，延时，用 55-22 打入 A 寄存器，送到 OUT，延时，跳转到开头，整体是一个死循环，即可实现这个程序。具体代码如下所示：

调试窗口 daima.asm | EPROM | 结构

```
OUTA
延时
LD A, #55H
A-W A, #22H
OUTA
延时
跳到 LOOP
END
```

行:34列:6 联机

指令集 | uM微程序 | 跟踪 | 红色

助记符	机器码1	机器码2	机器码3
-----	------	------	------

图 4、5. . 详细代码

### (2) 实验现象

OUT 寄存器交替显示 55, 22, 和 55-22 的值。

可以观察到，在流水指令系统（这次实验新编制的）中，OUT 寄存器中 55, 22, 33 交替显示的速度明显比上次实验编制的 yaokuai 。

本实验无数据分析与处理，数据记录同实验现象。

### (3) 实验结论

可以认为，本次实验正确地完成了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并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。发现流水方式的指令集运行程序的效率更高。

## 四、建议和体会

- 1、实验开始前，要先检查一下导线是否功能正常。
- 2、通过自己的尝试与摸索，自己学会了如何进行 CP226 软件上编写汇编语言与实验箱操作的结合。
- 3、依然还是老样子，提前预习的重要性，即在进行实验课前，需要对该节课实验的内容对应的理论知识（虽然好像与机组理论课关系不大）进行温故与复习并且对实验步骤进行大框架上的构想与思考，最好能够提前学会要用到的知识点。  
(如果不复习、提前构想的话，仅靠课堂上的那两个小时可能不足以完成所有任务，或者说进展的比较慢，没有什么头绪)。

## 五、思考题

- 1、计组实验课临近尾声，请你对该课程的授课形式、实验内容等提出你的建议？

答：我不知道这个建议是否可行==。这门课的教学内容可以稍微与时代接轨一些吗（我不太清楚现在对于计算机组成原理的研究还是需要在这种实验箱以及 win7 上的软件操作嘛）是否有可能使用一些更先进的实验箱或者软件？好像这个软件在 win11 系统上都无法运行。