上海大学　计算机学院

《计算机组成原理实验》报告十三

姓名　孔馨怡　　学号　22122128

时间　周一 9-11　　机位　17　　指导教师　顾惠昌

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＋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=1111 1111 1101 11 11 1011 1000=ff df b8H

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。

三、实验内容

1．实验任务一：分析流水指令集 insfle2.MIC，找出所有流水指令（与 insnle1.MIC 比较），进行分析利总结。

两个文件中相同指令已省略，指令不同之处如下表（第一栏结束后接同页第二栏，本页所有拦结束接下一页。

	insfile1.MIC		insfile2.MIC	
ADD A, R?	T2	FFF7EF	T1	FFF7EF
	T1	FFFE90	T0	CBFE90
	T0	CBFFFF		FFFFFF
		FFFFFF		FFFFFF
ADD A, @R	T3	FF77FF	T2	FF77FF
	T2	D7BFEF	T1	D7BFEF
	T1	FFFE90	T0	CBFE90
	T0	CBFFFF		FFFFFF
ADD A, *	T3	C77FFF	T2	C77FFF
	T2	D7BFEF	T1	D7BFEF
	T1	FFFE90	T0	CBFE90
	T0	CBFFFF		FFFFFF
ADD A, #*	T2	C7FFEF	T1	C7FFEF
	T1	FFFE90	T0	CBFE90
	T0	CBFFFF		FFFFFF
		FFFFFF		FFFFFF
ADDC A, @R?	T3	FF77FF	T2	FF77FF
	T2	D7BFEF	T1	D7BFEF
	T1	FFFE94	T0	CBFE94
	T0	CBFFFF		FFFFFF

ADDC A, *	T3	C77FFF	T2	C77FFF
	T2	D7BFEF	T1	D7BFEF
	T1	FFFE94	T0	CBFE94
	T0	CBFFFF		FFFFFF
ADDC A, #*	T2	C7FFEF	T1	C7FFEF
	T1	FFFE94	T0	CBFE94
	T0	CBFFFF		FFFFFF
		FFFFFF		FFFFFF
SUB A, R?	T2	FFF7EF	T1	FFF7EF
	T1	FFFE91	T0	CBFE91
	T0	CBFFFF		CBFFFF
		FFFFFF		FFFFFF
SUB A, @R?	T3	FF77FF	T2	FF77FF
	T2	D7BFEF	T1	D7BFEF
	T1	FFFE91	T0	CBFE91
	T0	CBFFFF		FFFFFF
SUB A, *	T3	C77FFF	T2	C77FFF
	T2	D7BFEF	T1	D7BFEF
	T1	FFFE91	T0	CBFE91
	T0	CBFFFF		FFFFFF
SUB A, #*	T2	C7FFEF	T1	C7FFEF

	T1	FFFE91	T0	CBFE91
	T0	CBFFFF		FFFFFF
		FFFFFF		FFFFFF
SUBC A, R?	T2	FFF7EF	T1	FFF7EF
	T1	FFFE95	T0	CBFE95
	T0	CBFFFF		FFFFFF
		FFFFFF		FFFFFF
SUBC A, @R?	T3	FF77FF	T2	FF77FF
	T2	D7BFEF	T1	D7BFEF
	T1	FFFE95	T0	CBFE95
	T0	CBFFFF		FFFFFF
SUBC A, *	T3	C77FFF	T2	C77FFF
	T2	D7BFEF	T1	D7BFEF
	T1	FFFE95	T0	CBFE95
	T0	CBFFFF		FFFFFF
SUBC A, #*	T2	C7FFEF	T1	C7FFEF
	T1	FFFE95	T0	CBFE95
	T0	CBFFFF		FFFFFF
		FFFFFF		FFFFFF
AND A, R?	T2	FFF7EF	T1	FFF7EF
	T1	FFFE93	T0	CBFE93

	T0	CBFFFF		FFFFFF	OR A, *	T3	C77FFF	T2	C77FFF			FFFFFF		FFFFFF
		FFFFFF		FFFFFF		T2	D7BFEF	T1	D7BFEF			FFFFFF		FFFFFF
AND A, @R?	T3	FF77FF	T2	FF77FF		T1	FFFE92	T0	CBFE92	RR A	T1	FFFFF7	T0	CBFCB7
	T2	D7BFEF	T1	D7BFEF		T0	CBFFFF		FFFFFF		T0	CBFFFF		FFFFFF
	T1	FFFE93	T0	CBFE93	OR A, #*	T2	C7FFEF	T1	C7FFEF			FFFFFF		FFFFFF
	T0	CBFFFF		FFFFFF		T1	FFFE92	T0	CBFE92			FFFFFF		FFFFFF
AND A, *	T3	C77FFF	T2	C77FFF		T0	CBFFFF		FFFFFF	RL A	T1	FFFCD7	T0	CBFCD7
	T2	D7BFEF	T1	D7BFEF			FFFFFF		FFFFFF		T0	CBFFFF		FFFFFF
	T1	FFFE93	T0	CBFE93	MOV A, R?	T1	FFF7F7	T0	CBF7F7			FFFFFF		FFFFFF
	T0	CBFFFF		FFFFFF		T0	CBFFFF		FFFFFF			FFFFFF		FFFFFF
AND A, #*	T2	C7FFEF	T1	C7FFEF			FFFFFF		FFFFFF	RRC A	T1	FFFEB7	T0	CBFEB7
	T1	FFFE93	T0	CBFE93			FFFFFF		FFFFFF		T0	CBFFFF		FFFFFF
	T0	CBFFFF		FFFFFF	MOV R?, A	T1	FFFB9F	T0	CBFB9F			FFFFFF		FFFFFF
		FFFFFF		FFFFFF		T0	CBFFFF		FFFFFF			FFFFFF		FFFFFF
OR A, R?	T2	FFF7EF	T1	FFF7EF			FFFFFF		FFFFFF	RLC A	T1	FFFED7	T0	CBFED7
	T1	FFFE92	T0	CBFE92			FFFFFF		FFFFFF		T0	CBFFFF		FFFFFF
	T0	CBFFFF		FFFFFF	IN	T1	FFFF17	T0	CBFF17			FFFFFF		FFFFFF
		FFFFFF		FFFFFF		T0	CBFFFF		FFFFFF			FFFFFF		FFFFFF
OR A, @R?	T3	FF77FF	T2	FF77FF			FFFFFF		FFFFFF	CPL A	T1	FFFE96	T0	CBFE96
	T2	D7BFEF	T1	D7BFEF			FFFFFF		FFFFFF		T0	CBFFFF		FFFFFF
	T1	FFFE92	T0	CBFE92	OUT	T1	FFDF9F	T0	CBDF9F			FFFFFF		FFFFFF
	T0	CBFFFF		FFFFFF		T0	CBFFFF		FFFFFF			FFFFFF		FFFFFF

分析：

如上表所示，在未合并的指令操作中，可以看到在所有的非流水指令集中，每条指令在写完有效指令之后，会写一条“CBFFFF”的取指指令，而在流水指令集中，这一条会与上一条指令合并，具体操作是将两条指令做“并”。可以发现所有可以改成流水线指令的指令都是在做“并”操作后不影响原功能的，也就是前两位是 FF，与 CBFFFF 的前两位取“并”就可以提前完成取指的操作，加快进度，而 CBFFFF 的后四位也不会影响原来指令的后四位功能。这就是预习时学习到的流水方式。

2.在实验十二编制的指令集中，增加一条 A 减立即数右移送 OUT 的指令（4 条微指令）。

在原来的指令集中，找到 MIC 文件在最后的位置添加如下指令：

猪猪 A A,#*	T3	18	C7FFEF
	T2	19	FFFE91
	T1	1A	FFFCB7
	T0	1B	CBDF9F

分析：

18、19 的 C7FFEF、FFFE91 完成 A 减立即数的操作，1A 的 FFFCB7 完成右移操作。原先完成送 OUT 的指令是 FFDF9F，最后加上取指的指令 CBFFFF，但由于题目要求和指令集四条微指令的限制，合并最后送 OUT 和取指的两个操作，为 CBDF9F，这里也用到了流水线方式。

在原来的指令集中，找到 DAT 文件在最后的位置添加如下指令：

猪猪 A A,#*	18	2

分析：

18 代表的是该条指令开始地址,对应该指令在 MIC 文件中的位置，2 的含义是所需的机器码数量，在之前的学习过程中可以知道，这时候一条需要的是实现这个操作的机器码，另一个是要传进去的立即数大小。

接下来就可在我们编写的指令集中使用可爱猪猪来完成操作了！

3. 将其中能改流水指令的改成流水方式。在自己编制的两个指令集运行同一个程序，观测运行情况和效率。(完成 OUT 寄存器交替显示 A-01H 右移的值与 A-03H 右移的值，A 的值自定，能清晰显示）。

为了体现两个指令集的运行时间和效率，打开原来的 MIC 文件修改，并攥写流失方式的指令集，两个指令集如下：

非流水线指令集			
FATCH	T3	00	FFFFFF
	T2	01	FFFFFF
	T1	02	FFFFFF
	T0	03	CBFFFF
输入 A A,#*	T3	04	C7FFF7
	T2	05	FFFFFF
	T1	06	FFFFFF
	T0	07	CBFFFF
减法 A,#*	T3	08	C7FFEF
	T2	09	FFFE91
	T1	0A	FFFFFF
	T0	0B	CBFFFF
跳到 *	T3	0C	C6FFFF
	T2	0D	FFFFFF
	T1	0E	FFFFFF
	T0	0F	CBFFFF
输出	T3	10	FFDF9F
	T2	11	FFFFFF
	T1	12	FFFFFF
	T0	13	CBFFFF
零跳转 *	T3	14	C6FFFF
	T2	15	FFFFFF
	T1	16	FFFFFF
	T0	17	CBFFFF
猪猪 A A,#*	T3	18	C7FFEF
	T2	19	FFFE91
	T1	1A	FFFCB7
	T0	1B	CBDF9F

流水线指令集			
FATCH	T0	00	CBFFFF
		01	FFFFFF
		02	FFFFFF
		03	FFFFFF
输入 A A,#*	T1	04	C7FFF7
	T0	05	CBFFFF
		06	FFFFFF
		07	FFFFFF
减法 A,#*	T1	08	C7FFEF
	T0	09	CBFE91
		0A	FFFFFF
		0B	FFFFFF
跳到 *	T1	0C	C6FFFF
	T0	0D	CBFFFF
		0E	FFFFFF
		0F	FFFFFF
输出	T0	10	CBDF9F
		11	FFFFFF
		12	FFFFFF
		13	FFFFFF
零跳转 *	T1	14	C6FFFF
	T0	15	CBFFFF
		16	FFFFFF
		17	FFFFFF
猪猪 A A,#*	T3	18	C7FFEF
	T2	19	FFFE91
	T1	1A	FFFCB7
	T0	1B	CBDF9F

分析：

在上面两个表格中，左边表格代表了非流水的指令集，其中还通过增加一些无效指令来增长运行时间，右边的表格代表了流水方式的指令集。

橘色框的含义是：为了减缓开始进行下一条指令而被放到最后的的取指指令。在这条指令和原先有效指令的中间用 FFFFFF 的无效指令填充，增加运行时间。

蓝色框的含义是：为了运用流水线方式，早一点开始取指开始做下一条指令，更快运行，而被放到前面的取指指令。而后面的无效指令都不需要保留，所以将用于计数有效指令条数的 T3、T2、T1、T0 进行更改。

紫色框的含义是：由于 CBFFFF 取指指令的前一条指令的前两位是 FF 所以可以将两条指令取“并”，更加加快我们的操作进程，也是运用流水线方式的一种方法。在任务一中已经进行分析。

综合来分析两个 MIC 文件，不仅使用两条指令“并”操作提前取指操作，还通过移动取指操作指令和填充无效指令增加运行时间等方式，我们也可以从第二列“T2、T1”等的有效条数中看到流水线的方式条数更少，运行更快，将会加快我们的实验进程，感受到本次实验的意义。

接下来我们将上一个实验的 ASM 文件拷贝，在此基础上进行更改，加入我们这次刚刚编写的 A 减立即数右移 OUT 的指令，并且实现实验中

要求的 OUT 寄存器交替显示 A-01H 右移的值与 A-03H 右移的值的操作，要求流水和非流水使用同一个 ASM 文件，我编写如下：

指令		分析
LOOP:	输入 A A,#05H	我们的指定数字 05H
	猪猪 A A,#01H	实现减去立即数右移 OUT 的操作
	输入 A A,#10H	延时操作
T3:	减法 A,#01H	
	零跳转 T4	
	跳到 T3	
T4:	输入 A A,#05H	我们的指定数字 05H
	猪猪 A A,#03H	实现减去立即数右移 OUT 的操作
	输入 A A,#10H	延时操作
T5:	减法 A,#01H	
	零跳转 T10	
	跳到 T5	
T10:	跳到 LOOP	交替实现以上操作

在上表中已经分析该文件的编写思路，在这里不过多赘述，因为也是原先几次实验早已掌握的内容。

接下来掉入两个不同的 MIC 文件，并对以上的同一个 ASM 文件进行编译下载、运行。

实验结果可以发现，使用流水线方式的指令集运行速度很明显大于非流水线方式的指令集速度。我们设定的实验数据是 05H，OUT 也成功交替显示 A-01H 右移的值与 A-03H 右移的值，即交替显示 02H、01H。

实验成功！

四、建议和体会

在原本的预习题目（没改的题目）中，有一条指令是延时指令，内容是三条 FFFFFF 和一条取指 CBFFFF，意义是通过增加无意义的指令来延长时间来达到延时效果，在这个指令的流水线方式下的变化是将 CBFFFF 提到了最前面，由这里我们也可以同样看出，流水线方式加快运行进度的原理就是提前开始下一条指令的取指以更快进行操作。

我们本次实验中还为了使显示时间变得相差更明显，所以才故意在非流水指令集中增加了很多条 FFFFFF 的无意义指令，可能实际运用场景没有这样做的意义，但是可以在本次实验中对这次实验的学习有更深的掌握。

但是流水线方式一些时候因为将操作功能的指令和 CBFFFF 的取指指令做“并”操作了，所以在中断等操作时，就会增加难度，变得复杂，有可能会出错。所以一个操作总是有利有弊的，要适当使用。

五、思考题

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：这里要感谢顾老师的教学方式，每次把很多重要的点说的很清楚，并且顾老师虽然每次改题目会让我做实验突然觉得有点压力和挑战的感觉哈哈哈，但是也是实实在在地在这个过程中学到了机组实验的很多知识，做到了举一反三和完全掌握实验目的要求的内容。我觉得这样的方式很好，感谢顾老师的教导。

我在这两个学期的机组实验课中，也有一些不足，比如在面对学校老旧设备或者不熟悉的编译系统的时候，有时候不知道报错是为什么。PS：软件里经常编译下载时会提示各种报错，但是从来都不会显示原因，这样找起来很痛苦，报错有好几种，可能是“不合规字符”“编译不成功”“error：一堆英文”，有时候刚接触不知道是哪里的问题，虽然自己摸索也可以慢慢找出问题锻炼自我解决问题的能力，但是还是希望对实验用到的编译环境有一个更全面的介绍和了解。

机组实验课学到很多，做的也很快乐，再次谢谢老师，祝愿老师样样好，希望学校早点换好一点的机子。