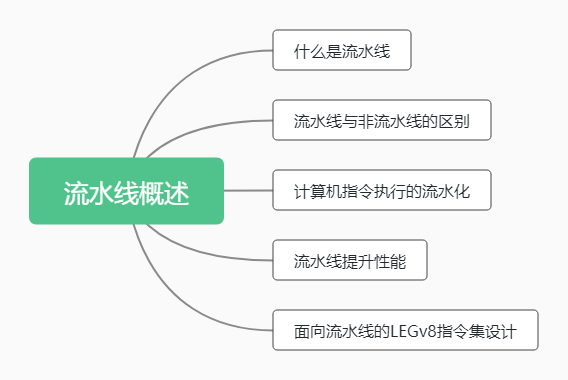
<第一部分start>

第一部分内容：4.5流水线概述先导内容、4.5.1面向流水线的指令集设计

负责人：2012679王娇妹



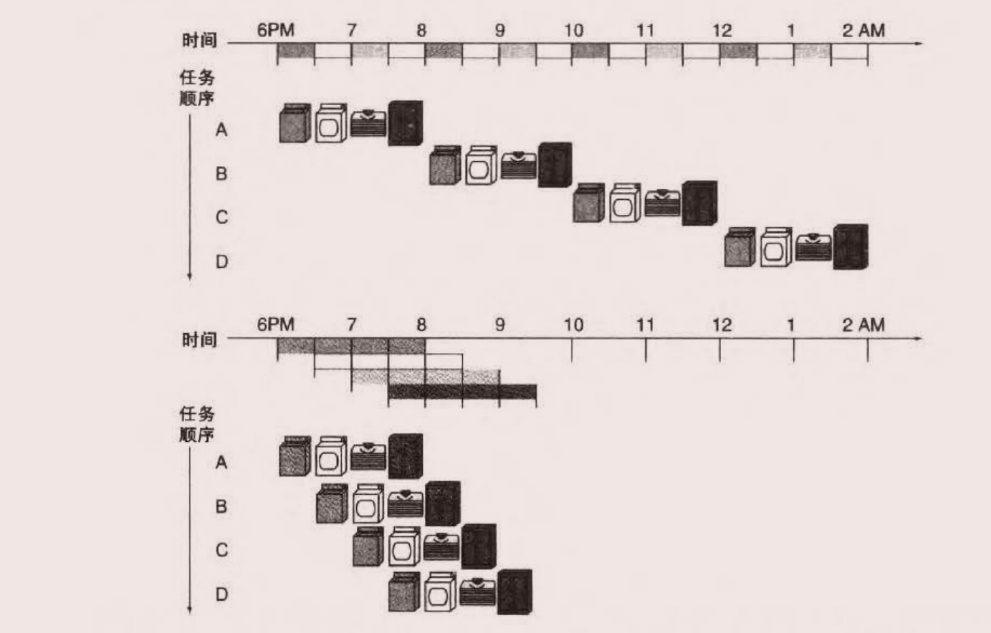
4.5流水线概述

什么是流水线？

流水线（pipelining）是一种实现**多条指令重叠执行**的技术。其使用与单周期类似的数据通路，但因吞吐率更高而更高效。

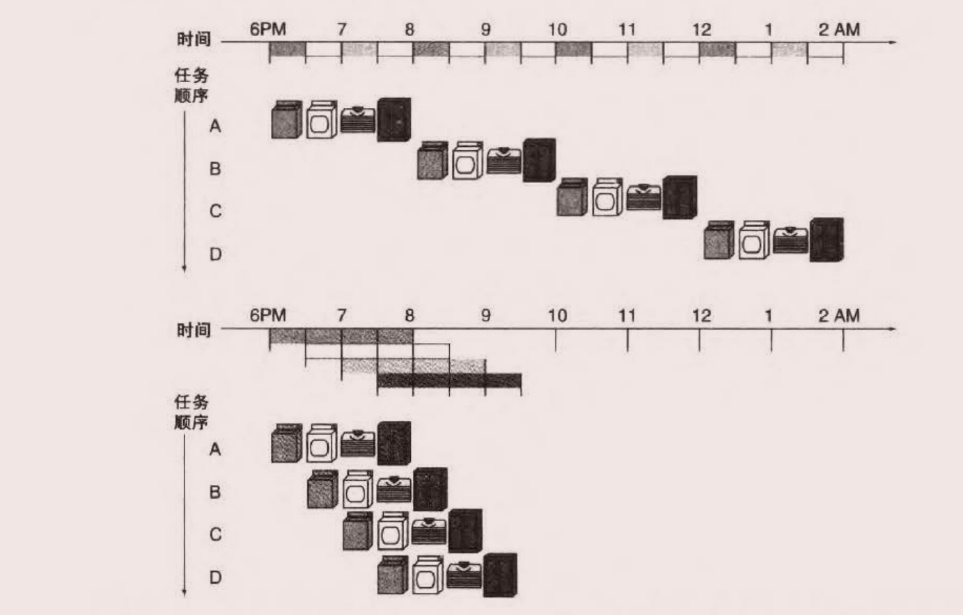
流水线与非流水线的区别？

**举例**（非流水线式的洗衣过程）：

1. 把一批脏衣服放入洗衣机里清洗。
2. 洗衣机洗完后,把湿衣服放人烘干机中。
3. 烘干机任务完成后,将干衣服放到桌子上并折叠起来。
4. 衣服叠好后,请室友将衣服拿走。
5. 当你的室友把衣服拿走后,再开始洗下一批脏衣服。

更节省时间的洗衣方式（流水线）：

当第一批脏衣服在洗衣机中完成洗涤后,第二批脏衣服就可以进行清洗了；当第一批衣服烘干后,洗净的第二批湿衣服放入烘干机了,并将第三批批脏衣服放入洗衣机里清洗……



只要每一个步骤中都有独立的资源(工作单元)时,任务就可以流水化执行。

流水线方式下,单独一批衣服从放进洗衣机到烘干、折叠、打包取走的总时间并没有缩短。多任务时流水线更快的原因是,所有的工作都在**并行**执行,每小时内能够完成更多的工作量。**流水线提高了洗衣系统的吞吐率**。吞吐率的改进可以减少完成所有工作的时间。

计算机指令执行如何进行流水化？

同样的原理也可以应用到处理器中,我们将指令的执行流水化。LEGv8指令通常包含如下五个处理步骤:

1、从指令存储器中取指。

2、读寄存器并对指令译码。

3、执行操作或计算地址。

4、从数据存储器中读取操作数。

5、将结果写回寄存器（如果有需要）。

本章讨论的指令只有7条：取数（LDUR）、存数（STUR）、加（ADD）、减（SUB）、与（AND）、或（ORR）、比较为0分支（CBZ）。

**举例题目**：

假设主要功能单元的操作时间为:访问指令或数据存储器200ps,ALU操作200ps,读写寄存器文件100ps。单周期模型中,每条指令执行都只需一个时钟周期,因此需要延展时钟周期以满足最慢的指令。

将单周期实现(所有的指令都在一个时钟周期内完成执行)中指令执行的平均时间,与流水实现下指令执行的平均时间进行对比。

**answer**：



单周期设计必须支持最慢的指令。计算每条指令的总执行时间，如上图所示，LDUR用时最长（800ps）,因此,每条指令所需的执行时间记为 800ps。



执行LDUR指令，如上图所示，两条指令之间的时间差从800ps降到200ps，第一与第四条指令之间的时间差缩短为3 \* 200ps=600ps，流水线执行可以达4倍的加速比。（PS：4倍加速比并不准确，后面有解释）

流水线中每一级都需要一个时钟周期,因此,时钟周期必须能够满足最慢的操作的执行,所以流水线设计中时钟周期也必须满足最坏情况（200ps）。

流水线如何提升性能？

**总结例题**：

上面讨论的流水线性能加速可以归纳为一个公式。如果流水线各级的操作平衡,那么流水线处理器上的指令执行时间为(在理想情况下):

**指令执行时间（流水线）=**

在理想情况且有大量指令的情况下,流水线所带来的性能加速比与流水线的级数近似相等。

但是！一般来说，流水线各级间并不是完全平衡的（如本例题，LDUR指令中有的耗时100ps，有的200ps）。那么流水线处理器中每条指令的执行时间可能会大于最小值,流水线的加速比将小于流水线的级数。

例题中，三条指令的总执行时间上并没有反映出最大性能加速比,实际获得的加速比为240Ops/1400ps ≈ 1.71。

当然,这是因为执行指令的数量不够多。

**增加指令数**：

我们在上面的流水线例子中增加1000000条指令。

非流水线式总执行时间：2400 + 1000000 \* 800 = 800002400 ps

流水线式总执行时间：每条指令都将导致整体执行时间增加 200ps，

故1400 + 200 \* 1000000 = 200001400 ps

那么加速比 = 800002400/200001400 ≈ 4

可以看出，流水线通过**增加指令的吞吐率**而不是减少单条指令执行的时间来提高性能。

实际程序通常都会执行成千上万条指令,因此,指令的吞吐率是一个很重要的度量标准。

面向流水线的LEGv8指令集设计

4.5.1面向流水线的指令集设计

根据前面关于流水线的例子，我们对面向流水线执行设计的LEGv8指令集有了一定了解：

第一、所有的LEGv8指令长度相同。这一限制简化了流水线第一级的取指与第二级的译码。

第二、LEGv8只有很少的几种指令格式,并且指令中源寄存器和目的寄存器字段位于相同的位置。

第三、LEGv8中的存储器操作数仅出现在 load和 store指令中。这一限制意味着可以利用执行级计算存储器地址,接着在下一级访问存储器。

练习题与解答：

假设主要功能单元的操作时间为:访问指令或数据存储器150ps，ALU操作130ps，读写寄存器文件80ps。将单周期实现(所有的指令都在一个时钟周期内完成执行)中指令执行的平均时间，与流水实现下指令执行的平均时间进行对比。

计算：相较于非流水线单周期实现，流水线实现能否带来4倍的性能加速比，如果能，至少需要多少条指令呢？

解答：

1. 根据各功能单元所需时间计算出每条指令的总执行时间，如下图所示。



（2）LDUR用时最长，所以每条指令所需的执行时间记为590ps。

（3）假设有k条指令。非流水线式总执行时间为 590k ps。对于流水线式，每条指令都将导致整体执行时间150ps，故总时间为590+(k-1)\*150 = 150k+440 ps

那么加速比 = ，当k→+∞时，加速比→3.933。

所以流水线无法达到4倍加速比。

<第一部分end>