

S698 系列处理器中指令流水的设计与实现

蒋晓华, 梁宝玉, 王祝金, 颜军

(珠海欧比特控制工程股份有限公司, 广东 珠海 519080)

摘要: 介绍了指令流水的概念和基本原理, 并指出应用流水线主要作用是提高吞吐量, 以及其优点和缺点。分析了 S698 系列单核处理器的 5 级流水线结构, 其划分为取指、译码、执行、存储和回写, 并给出了相应的设计方案和仿真结果。目前该处理器已实现了量产, 实际硬件的测试结果证实了指令流水确实能提高处理器的吞吐量。

关键词: 指令流水; 流水线; 吞吐量; S698; SPARC

中图分类号: TP274 **文献标识码:** A **文章编号:** 1003-353X (2009) 01-0024-03

Design and Implementation of the Instruction Pipeline in S698 Series Processors

Jiang Xiaohua, Liang Baoyu, Wang Zhujin, Yan Jun

(Zhuhai Orbita Control Engineering Co., Ltd., Zhuhai 519080, China)

Abstract: The concept and basic principle of instruction pipeline was introduced, and it was pointed out that the function of applying pipeline was for improving throughput, and the advantage and disadvantage were also given. The structure of 5-stage pipeline in S698 series single core processors was analyzed, and it was divided into fetch, decode, execute, memory and write. The corresponding scheme of design and the simulation result were given. Now the processor is in lots production, and the result of test of the real hardware confirms that the instruction pipeline can improve the throughput of a processor indeed.

Key words: instruction pipeline; pipeline; throughput; S698; SPARC

EEACC: 1280

0 引言

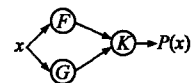
指令流水技术是现代高性能处理器普遍采用的技术, 是工业生产流水线原理在处理器中的应用。流水线是指通过若干个阶段执行一个任务的一组功能单元的序列, 其中每个阶段可以同时工作, 类似于工厂里的生产流水线。采用流水线技术的优点是可以提高吞吐量 (输出或输入的处理速率), 其本质思想是允许一个任务的各个阶段以并行的方式执行, 从而以空间换取时间^[1]。

1 基本原理

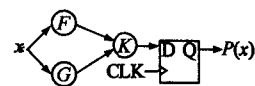
流水线的电路实现原理是通过把规模较大、层次较多的组合逻辑电路分为若干级, 在每一级插入

寄存器组并暂存中间数据。 K 级流水线就是从组合逻辑的输入到输出恰好有 K 个寄存器, 上一级的输出是下一级的输入而又无反馈的电路, 其中每一级电路就是流水线的的一个阶段, 因此, 组合逻辑电路是 0 级流水线。

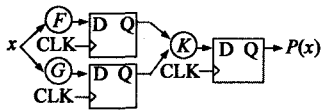
图 1 表示了如何把组合逻辑设计转换为相同组合逻辑功能的流水线设计。



(a) 0 级流水线



(b) 1 级流水线



(c) 2 级流水线

图 1 组合逻辑设计转化为流水线设计

Fig.1 Combinational logic design converted to pipeline design

电路 (a) 是组合逻辑电路, 为 0 级流水线, 包括两级: 第一级的延迟是 t_F 和 t_G 两个延迟中的最大值; 第二级的延迟等于 t_K 的延迟。为了通过这个组合逻辑得到稳定的计算结果输出, 需要等待的传输延迟为 $\max(t_F, t_G) + t_K$, 吞吐量为 $1/[\max(t_F, t_G) + t_K]$, 其时序图如图 2。

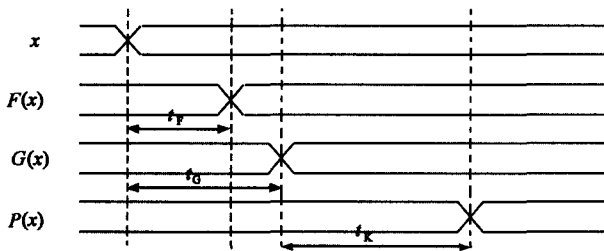


图 2 电路 (a) 的时序图

Fig.2 Timing diagram of schematic (a)

电路 (b) 是 1 级流水线, 同步时序电路中时钟周期应该满足条件: $t_{clk} > t_{co} + t_{pd} + t_s$, 其中 t_{clk} 是时钟周期, t_{co} 是从时钟边沿到寄存器输出有效的延迟, t_s 是寄存器的建立时间, t_{pd} 是两个寄存器之间的组合逻辑延迟, 在本文中等于 $\max(t_F, t_G) + t_K$ 。因此电路 (b) 的最小时钟周期是 $t_{co} + [\max(t_F, t_G) + t_K] + t_s$, 相应其吞吐量为 $1/\{t_{co} + [\max(t_F, t_G) + t_K] + t_s\}$, 比电路 (a) 的吞吐量小。因此采用 1 级流水线并不能提高吞吐量。

电路 (c) 为 2 级流水线, 该电路的最小时钟周期是 $t_{co} + \max[\max(t_F, t_G), t_K] + t_s$, 吞吐量是 $1/\{t_{co} + \max[\max(t_F, t_G), t_K] + t_s\}$, 只要 $t_{co} + t_s$ 小于 $\max(t_F, t_G)$ 和 t_K , 则吞吐量一定比电路 (a) 大。由此看出, 采用 2 级及以上的流水线可以提高吞吐量^[1]。

然而, 采用流水线技术会使延迟增加, 其时钟周期比不采用流水线时相应的各级组合电路的延迟都大, 而 K 级流水线的延迟就是时钟周期的 K 倍, 这显然比不采用流水线的组合逻辑的延迟要大。同时, 采用流水线技术的另一个代价是消耗更多的寄存器, 使电路复杂度增加。

处理器是一个复杂的时序电路, 其任务是执行

指令。一条指令的执行一般需要经过 K 个阶段, 假设执行 1 个阶段需要 1 个时钟周期, 则当不采用流水线技术时, 完成一条指令的执行需要 K 个时钟周期; 当采用了 K 级流水线后, 因为执行各阶段的部件在同时工作, 每经过 1 个时钟周期流水线的最末端便输出一组指令执行完毕的数据, 因此从宏观效果上看, 完成一条指令的执行只需要 1 个时钟周期, 效率提高了 K 倍, 也就是把 MIPS 值提高 K 倍。但是在实际应用的中当遇到数据相关指令或者转移指令时, 流水线的满载工作会被打断, 已经进入流水线的后面的指令变成无效, 必须等到该指令执行完毕后才能继续执行后面的指令, 这样会造成吞吐量的降低。

2 S698 系列处理器中指令流水的设计

S698 系列处理器是基于 SPARC V8 架构的 32 位 RISC 高性能嵌入式微处理器, 分为单核处理器和 4 核并行处理器两种类型。其中单核处理器的内部整数单元 (IU) 采用了 5 级指令流水结构, 而 4 核并行处理器则采用了 7 级流水线结构, 均可实现强大高效的整数运算功能^[2]。

以下着重介绍 S698 系列单核处理器的 5 级流水线, 流水线的各个阶段分别是: (1) 取指 (Fetch-FE); (2) 译码 (Decode-DE); (3) 执行 (Execute-EX); (4) 存储 (Memory-ME), (5) 回写 (Write-WR)^[3], 其工作示意图如图 3。

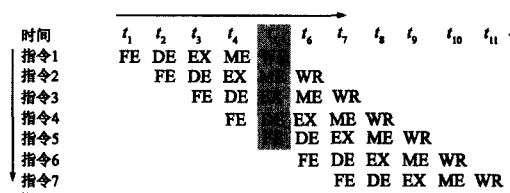


图 3 S698 系列处理器 5 级指令流水工作示意图

Fig.3 Illustrative diagram of 5-stage pipeline of S698 series processors

从图 3 可以看出, 最开始的前 4 个时钟周期 $t_1 \sim t_4$, 流水线处于初始化阶段, 有部分执行流水线阶段的部件处于空闲状态, 流水线没有满载工作。从 t_5 开始, 流水线便工作于满载状态, 在每个时钟周期里可以同时执行 5 条相邻指令的 5 个不同阶段, 这样的效果相当于在 1 个时钟周期里同时执行 1 条指令的 5 个完整阶段, 实现了单周期指令^[4]。

3 S698 系列处理器中指令流水的实现与验证

S698 系列处理器中指令流水的实现集中体现在整数单元中。整数单元的 VHDL 描述定义了流水线的 5 个阶段的关键信号：fe、fein、de、dein、ex、exin、me、mein、wr、wrin。其中 fein、dein、exin、mein、wrin 是组合逻辑输出；fe、de、ex、me、wr 是寄存器输出。指令流水的 5 个阶段的组合逻辑部分用 5 个独立的不含时钟信号的进程 fetch_stage、decode_stage、execute_stage、memory_stage 和 write_stage 分别描述。各个阶段中所用到的寄存器如 fe、de、ex、me、wr 等，在另外的含有时钟信号的进程里描述。因为在 VHDL 中各个进程是并行执行的，所以被描述的指令流水线的 5 个阶段也是并行执行的，这体现了流水线技术的特点。用 ModelSim 对其进行仿真，其结果如图 4 所示。

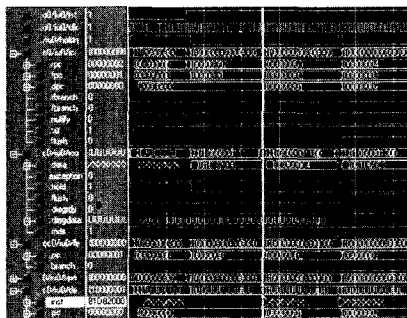


图 4. S698 系列处理器的整数单元仿真波形图

Fig.4 Simulation diagram of integer unit of S698 series processors

S698 系列处理器是基于 SPARC V8 架构的，所以其指令宽度是固定的 32 位，即 4 个字节。复位后处理器从 00000000H 地址开始执行一段固化在 PROM 里的引导程序^[5]。该程序是（16 进制）：

```
81d82000 03000004 821060c0 81884000
81900000 81980000 81800000 01000000
...
```

图 4 中，信号 ico.rpc 是指令缓存的输入的程序计数器的值，即输入地址，信号 fe.pc 和信号 de.pc 分别是在流水线的取指和译码阶段寄存的地址。上述地址的二进制值的宽度是 30 位而不是 32 位，省略了原本的输出给外部存储器的 32 位地址的低 2 位。在缓存命中的条件下，指令和数据分别来自指令缓存和数据缓存。复位后，上述 3 个信号的值都从 00000000H 开始递增，增量为 1，且 fe.pc 落后 ico.rpc 1 个增量，de.pc 落后 fe.pc 1 个增量，这表

明程序以顺序方式执行，地址的值顺序地通过流水线并被寄存，而且指令缓存的每个存储单元是 32 位。因为此时的地址省略了低 2 位，所以其值加 1 相当于原本的 32 位地址加 4。信号 ico.data 是指令缓存的数据输出，即取出的指令，而信号 de.inst 则是在流水线的译码阶段寄存的指令。信号 ico.data 和信号 de.inst 的值依次出现 81d82000, 03000004, 821060c0...，且 de.inst 落后于 ico.data，这表明引导程序的指令正确无误地通过流水线。

在 S698 ASIC 芯片的测试中，在 200 MHz 的时钟频率下，测得指令执行速率为 170 MIPS，也就是说每个时钟周期执行 170/200 = 0.85 条指令，这接近于指令流水在理想情况下的每个时钟周期执行一条指令的情况。假如不采用流水线技术，指令周期的各个阶段不同时执行，则完成一条指令的执行至少需要 5 个时钟周期，也就是说每个时钟周期执行 1/5 = 0.2 条指令。相比之下，采用流水线技术比不采用此技术时的指令执行效率有大幅度提高。

4 结语

目前 S698 系列处理器已经成功实现了量产，在消费电子、工业控制以及航空航天等领域中得到了广泛的应用。实践证明，S698 系列处理器中设计的 5 级流水线技术可以显著地提高 S698 系列处理器的吞吐量，该技术完全是科学合理的。

参考文献：

- [1] 夏宇闻. Verilog 数字系统设计教程[M]. 北京: 北京航空航天大学出版社, 2003.
- [2] 蒋晓华, 李付海, 祁波. SPARC 体系的 S698 系列 SoC 及其应用[J]. 单片机与嵌入式系统应用, 2007, 8: 84-85.
- [3] 珠海欧比特控制工程股份有限公司. 32-bit SPARC V8 Embedded Processor Sailing S698 User's Manual [K]. 2003.
- [4] 龚永红, 梅卫平, 蒋晓华, 等. 32 位嵌入式处理器 S698 的 SPARC V8 指令集[J]. 电子元器件应用, 2007, 9(11): 25-28.
- [5] SPARC International Inc. The SPARC Architecture Manual, Version 8 [K].

(收稿日期: 2008-07-02)



作者简介：

蒋晓华(1978—), 男, 湖南衡南人, 硕士, 现任珠海欧比特控制工程股份有限公司 IC 设计部经理。