第33卷 第23期 计 算 机 工 程 2007年12月
Vol.33 No.23 Computer Engineering December 2007
·博士论文· 文章编号：1000—3428(2007)23—0028—03 文献标识码：A 中图分类号：TP331

VelociTI结构浮点DSPs的流水线异常处理方法

胡正伟[1,2]，仲顺安[1]，陈 禾[1]
(1. 北京理工大学电子工程系，北京 100081；2. 华北电力大学电子与通信工程系，保定 071003)

摘 要：在采用基于VelociTI结构浮点DSPs流水线模式设计具有自主知识产权的数字信号处理器中，为了正确有效地实现对流水线异常的控制，提出了一种该结构流水线发生异常时的处理方法。对引起流水线异常的情况进行了合理的分类，存储器阻塞、多执行包和多周期NOP指令采用通过控制流水线寄存器的时钟信号实现控制。采用控制指令的执行条件实现了中断引起的流水线队列中部分指令的废除。对提出的方法采用VHDL语言建模设计，仿真结果验证了其正确性。

关键词：VelociTI结构；数字信号处理器；流水线异常

Processing Method for Pipeline Exception of Floating-point DSPs Based on VelociTI Architecture

HU Zheng-wei[1,2], ZHONG Shun-an[1], CHEN He[1]
(1. Department of Electronic Engineering, Beijing Institute of Technology, Beijing 100081;
2. Department of Electronic and Communication Engineering, North China Electric Power University, Baoding 071003)

【Abstract】 In order to design digital signal processor with own property right, one kind of pipeline structure of floating point digital signal processor based on VelociTI architecture is studied and a new processing method for pipeline exception is proposed. Abnormal things affecting pipeline are classified reasonably. Pipeline registers' clock signals of pipeline registers are controlled for memory blocking, many execute packets, many cycles NOP instructions. Some instructions should be disabled when interruption occurs, this work is done by controlling execute condition of these instructions. The correctness can be confirmed by simulation results.

【Key words】 VelociTI architecture; DSPs; pipeline-exception

数字信号处理器(DSPs)是高速、实时信息处理的核心，广泛应用在军事、工业、通信、图像处理等众多领域。但国内处理器设计还相对落后，DSPs产品主要依靠进口。借鉴国外的设计方式是提高国内微处理器设计和开发能力的一种有效途径。本文主要介绍与TI公司高性能DSPs——TMS320C6701指令集兼容的具有自主知识产权DSPs的重要组成部分。

流水线是目前高速CPU中采用的关键技术，主要是采用分时复用硬件资源的设计原理，在不增加或少量增加硬件资源的情况下，实现系统吞吐率的提高。不同体系结构的处理器之间的流水线异常情况差别很大。本文研究了VelociTI结构浮点DSPs的流水线模式，对引起流水线异常的情况进行了合理的分类总结，并提出了流水线异常情况的处理方法。

1 VelociTI结构的浮点流水线介绍

VelociTI结构的浮点流水线分为取指、译码和执行3个级。取指级分为PG, PS, PW, PR 4个节拍，译码级分为DP, DC 2个节拍，执行级分为E1~E10共10个节拍[1]。图1表示了浮点操作的各级流水线节拍。PG(program address generate)：程序地址产生；PS(program address send)：程序地址发送；PW(program address ready wait)：程序访问等待；PR(program fetch packet receive)：程序取指包接收；DP(instruction dispatch)：指令分配；DC(instruction decode)：指令译码；E1~E10(instruction execution)：指令执行。

PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10

图1 浮点流水线节拍

2 VelociTI结构的浮点流水线的异常情况

VelociTI结构是一种典型的VLIW(very long instruction words)结构，该类结构指令之间的并行性完全是在编译时确定的，在CPU动态执行程序时指令之间的结构冒险、数据冒险、控制冒险等已经被解决[2~3]，因此不会引起流水线异常。编译器通过路径调度(trace scheduling)、循环展开(loop unrolling)、软件流水(software pipeline)、预测(prediction)等技术解决指令之间的各种冒险[4]。但是编译时为了消除各种冒险而增加的多周期NOP(no operation)指令会引发流水线异常。同时为了方便取指，编译器将不一定并行执行的8条指令压缩为一个取指包FP(fetch packet)，这样就引发了多执行包EP(execution packet)引起的流水线异常。取指包FP是可以同时进行取指的指令集合，执行包EP是可以同时执行的指令集合。因此，经过编译后的程序引发的流水线异常只有两种情况：多周期NOP指令和多执行包。在CPU运行程序期间，硬件的问题也会引起流水线异常，主要有存储器阻塞和中断两种情况。综上，引起VelociTI结构流水线异常的情况共有下述4种。

(1)多周期NOP指令

NOP指令是不使用功能单元的空操作指令。延迟间隙是

基金项目：国家部委预研基金资助项目(200205)
作者简介：胡正伟(1978－)，男，博士研究生，主研方向：微处理器设计，大规模数字集成电路设计；仲顺安，教授、博士生导师；陈 禾，副教授
收稿日期：2006-12-10 **E-mail：**hzwwizard@sohu.com

指一条指令在第 1 个执行节拍 E1 以后占用的 CPU 周期数。具有延迟间隙的指令，在最后一个延迟间隙之前，其结果不能被使用。指令 NOP *n*(1≤*n*≤9)使它本身执行包的指令和所有在它前面的执行包都插入了 *n* 个额外的延迟间隙。因此，如何插入额外的延迟间隙是控制多周期 NOP 指令对流水线影响的关键。通过研究，提出采用延迟执行多周期 NOP 指令所在执行包以后的所有执行包，延迟的周期数等于多周期 NOP 指令插入的额外延迟间隙。这样变相地增加了多周期 NOP 指令所在执行包和所有在它前面的执行包的延迟间隙，可以很方便地实现对流水线的控制。

流水线控制方案：DC 节拍经过译码可以判断多周期 NOP 指令是否存在和需要插入的延迟间隙。由于可能存在与多周期 NOP 指令并行执行的指令，因此要等待进入 E1 节拍后，才停止流水线寄存器 DC/E1 及其以前的所有流水线寄存器和 FPC(fetch packet counter)的时钟信号，直至延迟间隙结束，重新启动时钟信号。而流水线寄存器 E1/E2 及其以后的流水线寄存器的时钟信号不受影响。图 2 说明了 NOP3 指令引起流水线异常的情况。每个 FP 含 1 个 EP，FP*n* 为含有 NOP3 的 EP，译码信号 M_NOP(many cycles NOP instruction)在 E1 节拍变为有效，则 FP*n* 以后的执行包插入 3 个延迟间隙。多周期 NOP 指令如处在跳转指令的延迟间隙内时，跳转指令会影响多周期 NOP 插入的延迟间隙。当跳转指令的延迟间隙结束后时，没有结束的多周期 NOP 指令的延迟间隙必须结束。

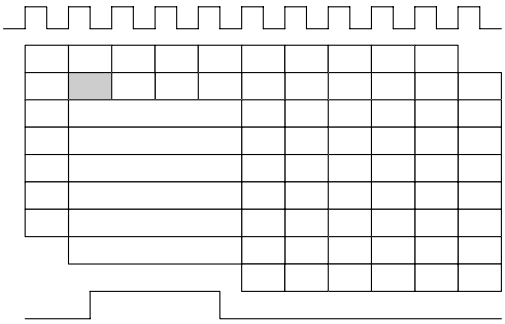


图 2　多周期 NOP 指令引起的流水线异常

(2)多执行包

为了取指的方便，编译器将不足 8 条指令的执行包压缩在一起组成 256bits 的取指包。1 个取指包含多个执行包的优点是方便取指，但因此也带来了新的流水线异常。由于一个取指包的所有指令不能在 1 个流水线周期内进入 DC 节拍，因此必须占用 DP 节拍多个流水线周期，为了在 DP 节拍的取指包不被覆盖，取指级的 4 个节拍必须停止，而 DP 节拍及其以后的流水线节拍仍正常工作。

流水线控制方案为：在 DP 节拍对取指包的 8 条指令的并行控制位进行测试，若含有多个执行包，则将取指级的 4 个节拍的流水线寄存器和 FPC 的时钟信号停止，其余流水线寄存器的时钟信号仍正常工作。图 3 表示了多执行包引起的流水线异常。FP*n* 为含有 3 个执行包取指包，信号 M_E(many execution packet)为高电平时表示 1 个取指包含有多个执行包，低电平时说明 1 个取指包只含有 1 个执行包。当 FP*n* 中的最后 1 个执行包处在 DP 节拍时，M_E 信号变为低电平。

(3)存储器阻塞

当存储器没有做好相应 CPU 访问的准备时，流水线将产生存储器阻塞。对于程序存储器，存储器阻塞发生在 PW 节拍，数据存储器则发生在 E3 节拍。存储器阻塞引起处于该流水线的所有节拍延长一个时钟周期以上，从而使执行增加额外的时钟周期。存储器阻塞的控制要求在发生阻塞时，整个流水线处在停止工作状态。

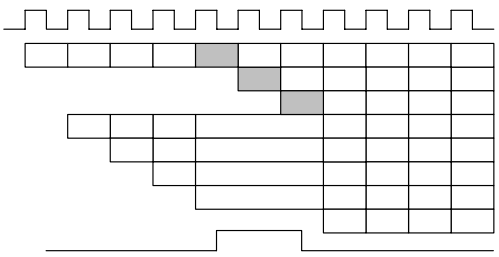


图 3　多执行包引起的流水线异常

流水线控制方案为：发生存储器阻塞时，停止流水线中所有的流水线寄存器和 FPC 的时钟信号，使整个流水线处在停止工作状态。在此期间，存储器仍然处在工作状态，为 CPU 的正确访问做准备。等存储器准备好后，阻塞取消，启动流水线时钟。若程序存储器和数据存储器同时发生阻塞，则阻塞周期与两者阻塞周期的长度一致。图 4 为存储器引起的流水线异常的情况。FP*n* 在 E3 节拍发生数据存器阻塞，FP*n*+1 在 PW 节拍发生程序存储器阻塞。

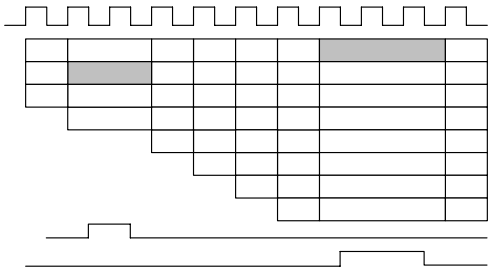


图 4　存储器阻塞引起的流水线异常

(4)中断

中断要求CPU暂停当前的工作，转而处理引起中断的事件，处理完这些事件后，再回到原来被中断的地方，继续原来的工作。复位中断发生时，正在执行的指令紧急停止，所有的寄存器返回到它们的缺省状态。复位中断将CPU复位，是比较特殊的中断，它使CPU重新执行程序，流水线中所有指令被清除。非复位中断发生时，为了能够在中断执行完毕后正确返回到被中断的地方，需要将处在E1 节拍的执行包地址写入中断寄存器NPR或IPR，所以等中断结束后返回到中断发生时处在E1 节拍的执行包所在地址。中断发生时，不需要停止流水线时钟，但为了防止CPU状态被修改，需要将处在E1 节拍之前的执行包中所有指令废除。VelociTI结构DSPs的指令都有自己的执行条件[1]，功能单元根据执行条件决定是否执行该条指令，因此提出通过控制指令的执行条件实现流水线的排空。当非复位中断发生时，将指令的执行条件设置为无效，则指令将不会被执行，实现了废除的指令不改变任何CPU状态。

图 5 中，在周期 T+1 时刻发生中断，此时处在 E1 节拍的为取指包 FP*n*，因此取指包 FP*n* 的地址保存至 NRP 或 IRP。取指包 FP*n* 至 FP*n*+6 将被废除，在周期 T+7 所有被废除的取指包的指令执行条件被设置为无效，因此可以认为完成了指

令的废除。执行条件控制信号 E_ctr 作为废除信号，需要保持 6 个有效的时钟周期，以保证取指包 FP*n* 至 FP*n*+6 的指令不被执行。

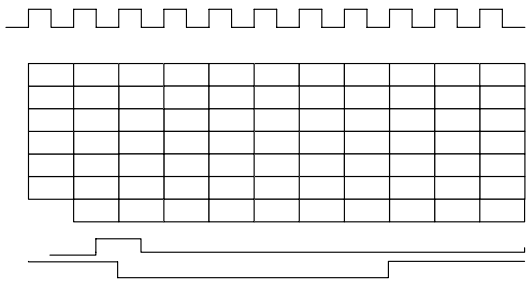


图 5　中断引起的流水线指令废除

3　流水线异常处理的设计与仿真

为了验证理论的正确性，用 VHDL 对流水线异常处理方法进行 RTL 设计，设计方案如图 6 所示。

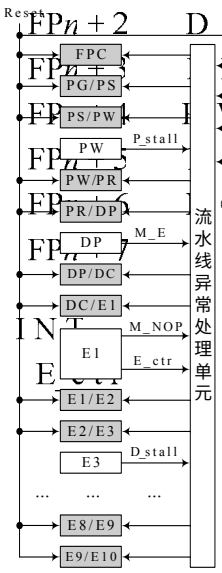


图 6　流水线异常处理单元的设计方案

流水线异常处理单元的输入信号为系统时钟信号 clk 和复位信号 Reset，引起流水线异常的信号 P_stall, D_stall, M_E, M_NOP 和 INT，由于跳转指令影响多周期 NOP 指令而间接影响流水线，因此跳转延迟间隙标志信号 Branch 也作为流水线异常处理单元的输入之一。输出的信号为 15 个流水线寄存器和 FPC 的时钟信号、中断时的指令废除信号。根据影响流水线寄存器时钟信号的数量将 stall(P_stall or D_stall), M_E, M_NOP 进行优先级排列，得到流水线寄存器和 FPC 的时钟控制信号，如表 1 所示。

表 1　流水线异常的优先级及其时钟控制信号

stall	M_NOP	M_E	时钟 1 控制	时钟 2 控制	时钟 3 控制
1	X	X	0	0	0
0	1	X	0	0	1
0	0	1	0	1	1
0	0	0	1	1	1

由于一些流水线寄存器的时钟控制信号一致，因此可以共享一个时钟。FPC, PG/PS, PS/PW, PW/PR, PR/DP 共享时钟 1，DP/DC 与 DC/E1 共享时钟 2，E1/E2, E2/E3, E3/E4, E4/E5, E5/E6, E6/E7, E7/E8, E8/E9, E9/E10 共享时钟 3。采用门控时钟的设计原则，用生成的控制信号与系统时钟信号进行逻辑“与”，从而得到每个流水线寄存器和 FPC 的时钟信号。中断不影响流水线的时钟信号。

图 7 为部分仿真结果，从中可以验证提出方法的正确性。在周期 2 检测到发生 P_stall，在周期 3 的时钟上升沿到来后，整个流水线阻塞，所有的流水线寄存器时钟信号无效，流水线阻塞 1 个周期。周期 4 发生 NOP4 指令，因为从周期 3 开始处于跳转指令的延迟间隙内，所以 NOP4 指令受到跳转指令的影响，本来应该在周期 8 结束延迟间隙，但由于在周期 7 跳转延迟结束，因此 NOP4 指令此后的延迟间隙被取消。在周期 8 发生多执行包情况，在此处为 2 执行包情况。在周期 9 发生中断，从周期 10 至周期 15 指令执行条件控制信号保持低电平 6 个周期。仿真结果与理论完全一致。
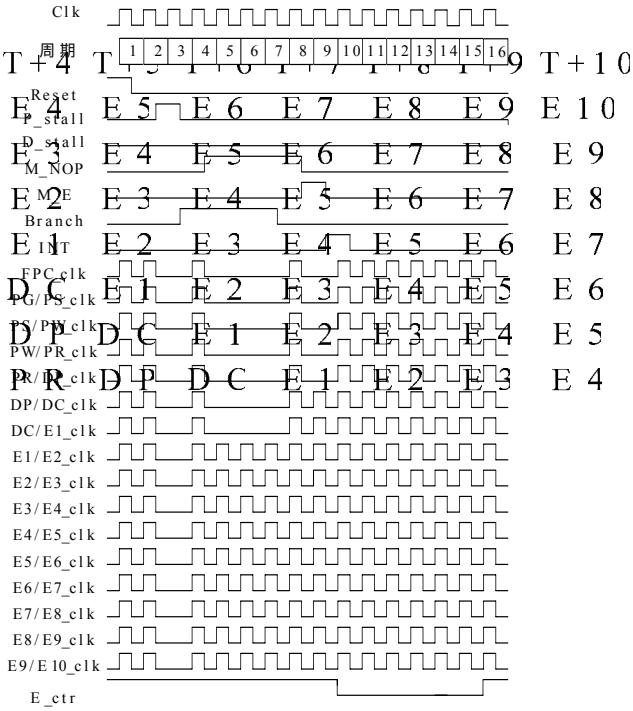


图 7　仿真结果

4　结束语

本文研究了基于 VelociTI 结构的浮点 DSPs 流水线的特点和引起流水线异常的情况，提出了流水线异常情况的处理方法。该流水线异常处理方法已应用于设计具有自主知识产权的数字信号处理器中。本文的设计思想也为采用类似 VLIW 结构的流水线进行微处理器设计提供了参考。研究国外先进的技术原理，可以借鉴他们的科技成果，最终的目的是提出更加合理的流水线结构。

参考文献

1 Texas Instruments. TMS320C6000 CPU and Instruction Set Reference Guide[EB/OL]. (2005-05-20). http://www-s.ti.com/sc/techlit/spru733.pdf .

2 Seshan N. High VelociTI Processing[J]. Signal Processing Magazine, 1998, 15(2): 86-101, 117.

3 Simar R Jr. Codevelopment of the TMS320C6X VelociTI Architecture and Compiler[C]//Proc. of IEEE Int'l conf. on Acoustics, Speech, and Signal Processing, 1998: 3145-3148

4 洪　享. 嵌入式数字信号处理器研究与设计[D]. 杭州: 浙江大学, 2003.