

文章编号: 1001- 2486(2009) 05- 0033- 05

低功耗微处理器中异步流水线设计*

石 伟, 王友瑞, 陈芳园, 任洪广, 陆洪毅, 王志英
(国防科技大学 计算机学院, 湖南 长沙 410073)

摘要: 随着工艺的不断改进及芯片上资源的不断增加, 微处理器设计遇到了一系列问题: 为芯片提供一个全局时钟网络越来越困难, 时钟扭曲等问题越来越突出, 芯片的功耗问题越来越严重。上述这些因素促使人们将注意力逐渐转向异步电路设计。在设计异步微处理器过程中, 异步流水线的设计是一个非常重要的问题。首先总结了微处理器设计中出现的各种流水线结构, 并给出了相应的异步实现; 然后提出了一种异步流水线设计流程, 用于加速异步流水线的设计; 最后利用提出的流程设计实现了几种异步功能单元, 实验结果表明异步电路能够有效降低电路的功耗。

关键词: 低功耗; 流水线; 异步电路; 设计流程

中图分类号: TP332. 2 **文献标识码:** A

Design of Asynchronous Pipelines for Low-power Microprocessor

SHI Wei, WANG You-rui, CHEN Fang-yuan, REN Hong-guang, LU Hong-yi, WANG Zhi-ying
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: As the CMOS technology enters the deep submicron design era, the richness of the computational resources brings about a lot of problems, such as clock distribution, clock skew and high power dissipation. Asynchronous circuit style is an efficient approach to solve the problems, and it is becoming significantly attractive to the designers. The design of asynchronous pipelines is a very important issue in the process of designing asynchronous microprocessors. In this paper, various pipeline structures are summarized and their asynchronous equivalents are presented, and then an asynchronous flow is proposed, aiming at speeding up the asynchronous circuit design. Finally, the flow is used to design several asynchronous pipelines. The experimental results show that the asynchronous technique can reduce the power consumption of microprocessor effectively.

Key words: low-power; pipeline; asynchronous circuit; design flow

随着半导体工艺进入到深亚微米阶段, 同步微处理器设计遇到了前所未有的挑战。复杂时钟网络带来的功耗增加、面积增大、时钟扭曲等问题日益突出。在采用同步电路设计的微处理器中, 功耗在很大程度上是由时钟树消耗的。文献[1]称片上时钟树的功耗占芯片功耗的比重达到 25%~30%。Alpha 21264 微处理器中时钟网络的功耗达到了总功耗的 32%^[2]。由于同步电路设计遇到了越来越多的问题, 异步电路设计方法开始得到更广泛的关注。异步电路采用局部通信模式, 以局部握手协议取代全局时钟, 不需要设计庞大的时钟分布网络, 很好地解决了时钟扭曲等问题。其次, 由于异步电路中空闲电路几乎没有动态功耗, 使得整个系统的功耗得到有效控制。最后, 异步电路还具有模块化与可重用性的优点, 可以采用异步电路连接不同速度的同步电路, 从而解决了大规模系统中的多时钟域问题。

国外大学和研究机构目前正对异步数字电路展开深入的研究, 并取得了许多创新成果。英国曼彻斯特大学由 Steve Furber 教授领导的先进处理器技术研究小组从 20 世纪 90 年代初期就开始异步电路设计研究。该小组设计实现了和 ARM 系列兼容的 AMULET^[3-5] 系列异步微处理器, 他们的工作表明异步数字电路设计是非常有潜力的, 在性能和功耗上完全可以超越相同工艺的同步数字电路。国内对异步

* 收稿日期: 2009- 07- 03

基金项目: 国家 863 计划资助项目(2007AA01Z101); 国家自然科学基金资助项目(60873015, 60773024); 国防科技大学校资助项目(JC- 08- 06- 02); 教育部“高性能微处理器技术”创新团队资助项目(IR06164)

作者简介: 石伟(1982-), 男, 博士生。

数字电路和异步微处理器设计研究尚处于起步阶段。复旦大学专用集成电路与系统国家重点实验室曾经实验设计了 8 位的异步 PIC16C16 芯片^[6]。国防科技大学计算机学院也正在对异步微处理器设计进行较深入的研究,并与曼彻斯特大学和剑桥大学等研究同行进行多方面的交流和合作。

我们设计实现了一款异步微处理器 ADTA-1,在设计实现该异步微处理器的过程中,异步流水线设计是一个必须解决的难题。首先总结了微处理器设计中出现的各种流水线结构,并给出了相应的异步电路实现;然后提出了一种异步流水线设计流程,用于加速异步流水线的设计;最后在 0.18 μm 工艺下设计实现了几种异步功能单元,并进行了比较分析。

1 异步电路设计方法

由于缺少成熟的 EDA 工具支持,异步电路设计一直没有得到广泛的应用。早在 20 世纪五、六十年代,研究人员就已经开始了异步电路设计方法研究。现有的异步电路设计方法可以分为控制通路和数据通路分离的设计方法及基于语法驱动转换的设计方法两种。解同步方法^[7]从已有的同步电路网表着手,将全局时钟网络替换为本地握手电路,然后使用与同步电路设计一样的 EDA 工具实现相应的异步电路。我们结合现有的同步及异步工具提出了一种解同步异步设计流程^[7]。该设计流程可以与现有的同步 EDA 设计工具进行整合,采用设计人员熟悉的语言与工具,支持工艺厂商提供的标准单元库,大大缩短了开发周期,提高了设计效率。本文采用四段握手数据打包协议,该协议不需要对数据通路中的寄存器进行修改,并且具有面积开销小的优点。

1.1 控制通路设计方法

由于异步电路中各个模块采用握手的方式进行通信,因此控制通路需要生成异步电路的局部时钟,使整个电路能够正常工作。简单四段握手协议(Four-Phase Latch Control, FLC)是异步电路握手通信的最直接实现,但是该协议会使异步流水线出现一段满一段空的现象。为了解决简单四段握手协议性能低的问题,文献[9]提出了全解耦四段握手协议(Full-Decoupled FCL, FDFLC)。另外,我们提出了冗余四段握手协议(Redundant FLC, RFLC)。与 FDFLC 一样,RFLC 的基本思想是降低不同流水段间的耦合度,从而提高流水线性能。RFLC 在相邻流水段的握手电路之间增加冗余握手电路,其 STG 图如图 1(a)所示。图 1(b)是 RFLC 协议电路实现的符号图。本文对 FDFLC 与 RFLC 两种协议的具体实现进行比较,结果如表 1 所示。从表中可以看出,RFLC 的性能与面积要全面优于 FDFLC。

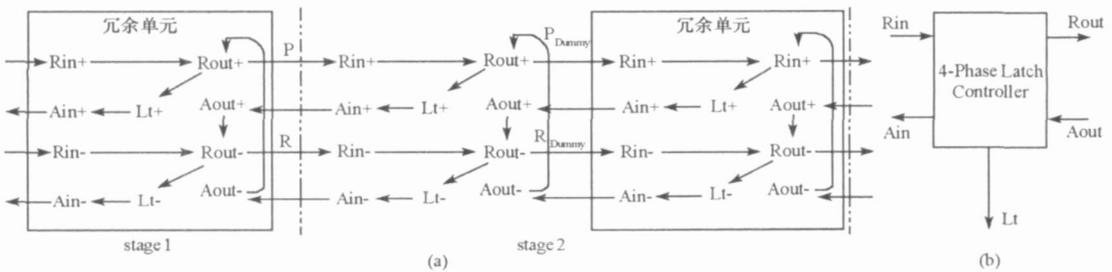


图 1 (a) 冗余四段握手协议的 STG; (b) RFLC 控制器符号图

Fig. 1 (a) The STG of RFLC; (b) The symbol of RFLC controller

表 1 四段握手协议控制电路比较

Tab. 1 Comparison of four-phase handshake protocols

协议	FLC	RFLC	FDFLC
面积(μm^2)	112.3	213.6	257.7
延迟(ns)	0.39	0.71	0.93

由于四段握手协议存在信号复位的固有缺陷,四段握手协议的一次握手周期开销较大。为了进一步提高异步流水线的性能,需要将延迟匹配单元设计成非对称延迟单元,减少复位信号的传输延迟。

1.2 数据通路设计方法

在本文采用的异步集成电路设计方法中,数据通路的设计方法与同步电路中的数据通路设计方法一致,只需要在设计过程中将同步数据通路的全局时钟替换为多个局部时钟。这样做的好处是异步电路的数据通路可以采用成熟的同步 EDA 工具进行设计、综合、布局、布线以及时序分析,更重要的是时序分析能够精确地统计出数据通路中各流水段逻辑的延迟,为控制通路的设计提供指导。另外,由于异步数据通路的设计与同步电路一致,对于有同步设计经验的设计人员而言,可以按照同步电路设计的方法直接进行异步数据通路的设计。

2 异步流水线结构设计

2.1 同步流水线结构

流水线是一种在连续指令流中开发指令级并行性的技术。目前,流水线已经成为高性能微处理器中的一种关键技术。根据不同的观点,流水线具有多种不同的划分方法。按照数据的流动是否存在反馈回路,流水线可以划分为线性流水线与非线性流水线^[10]。

通过对同步微处理器的结构进行分析,得出微处理器中同步流水线结构的分类如表 2 所示。线性流水线可以分为单分支线性流水与多分支线性流水两种。单分支线性流水线是一种最简单且常见的流水线结构,如采用流水方式实现的浮点功能单元。多分支线性结构一般在实现具有多种功能的功能单元中采用,如 ALU 单元中可以实现加减法和各种逻辑操作等,因此可以将 ALU 单元看成多分支线性结构。非线性流水线中一般具有环形反馈通路,如状态机、采用迭代方式实现的乘法及除法单元等。

表 2 微处理器中流水线结构

Tab. 2 The pipeline structures in microprocessor

流水线分类	物理结构	实例	异步握手部件
线性流水线	单分支线性结构	浮点单元	RFLC 控制器
	多分支线性结构	ALU	DEMUX、MERGE
非线性流水线	环形结构	状态机	FORK、JOIN
		迭代乘/除法	MUX、DEMUX

2.2 异步流水线设计流程

控制通路中的基本异步握手部件主要包括 FORK、JOIN、MUX、DEMUX 与 MERGE^[11]。表 2 给出了各种不同流水线结构的异步控制通路可以采用的基本握手部件。简单的单分支线性结构流水线可以采用基本的冗余四段握手控制器,而各种状态机则需要采用 FORK-JOIN 握手部件。

在分析了流水线结构及异步握手部件以后,本文提出了一个异步流水线设计流程,该流程能够有效加速并简化异步流水线设计。异步流水线设计流程详细步骤如下:

- (1) 设计实现冗余四段握手协议控制器,在 RFLC 控制器基础上设计实现 FORK、JOIN、MUX、DEMUX 与 MERGE 等基本握手部件模块,形成一个基本的握手部件库;
- (2) 设计实现多个具有不同延迟的非对称延迟单元,形成一个延迟单元库;
- (3) 分析同步流水线的的数据流,并将全局时钟替换成多个局部时钟,得到数据通路;
- (4) 对数据通路进行综合,得到每一流水段的精确延迟;
- (5) 根据(2)中分析得到的数据流图及(4)中综合获得的精确延迟,从库中选取各种相关的基本握手模块与非对称延迟单元组合生成完整的控制通路;
- (6) 将控制通路的局部时钟信号与数据通路中相应的时钟信号相连,得到异步流水线代码。

2.3 设计实例

上述异步设计流程为快速异步流水线设计提供了有效保障,本节利用该异步设计流程对多分支线性流水线进行异步设计。同步的多分支线性流水线结构如图 2 所示,流水线中的所有寄存器由一个全

局时钟控制。当该流水线对数据进行处理时,可以根据操作码选择不同的分支。

通过对同步流水线的数据流进行分析,可以将全局时钟信号替换成多个局部时钟,得到异步流水线的数据通路,如图 3 所示,全局时钟被替换成三个局部时钟信号,它们将由异步握手控制通路产生。

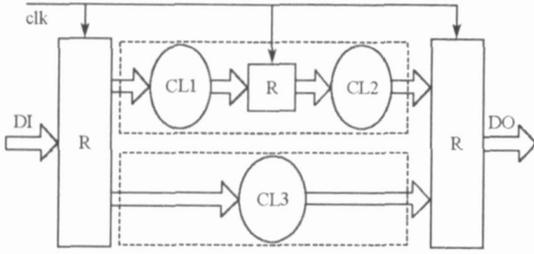


图 2 多分支线性流水线结构
Fig. 2 Multi-branch linear pipeline

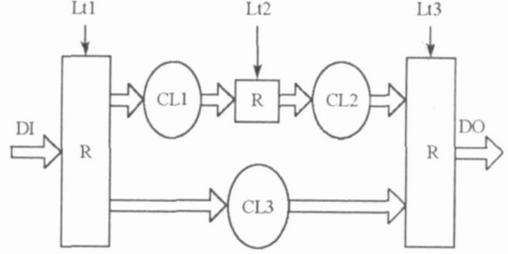


图 3 多分支线性流水线数据通路
Fig. 3 The datapath of multi-branch linear pipeline

由图 3 知道,控制通路需要产生 3 个局部时钟 L_{t1} 、 L_{t2} 与 L_{t3} 。在 L_{t1} 处数据流发生条件分离现象,可以采用 DEMUX 握手部件来产生局部时钟;在 L_{t2} 处数据流未发生变化,可以直接采用冗余四段握手控制器来产生局部时钟;多个数据流在 L_{t3} 处进行合并,因此可以用 MERGE 握手部件来产生局部时钟。DEMUX 握手部件与 MERGE 握手部件的结构分别如图 4 与图 5 所示。在确定了各局部时钟信号的产生部件以后,可以将这些部件按照数据流的流向连接起来获得多分支线性流水线控制通路,其结构如图 6 所示。实际上,可以通过对同步流水线分析得出的数据流图的特性直接选取各种异步握手模块组成异步控制通路。

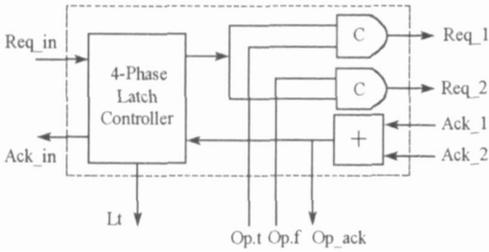


图 4 DEMUX 握手部件结构图
Fig. 4 Structure of DEMUX handshake component

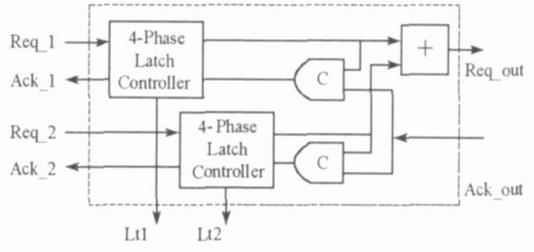


图 5 MERGE 握手部件结构图
Fig. 5 Structure of MERGE handshake component

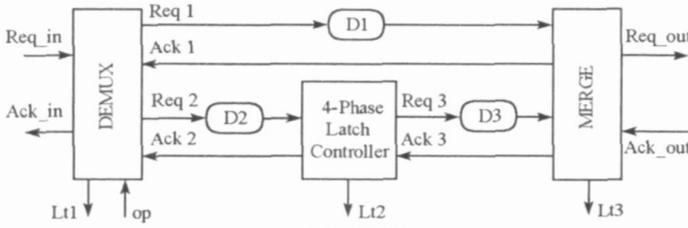


图 6 多分支线性流水线控制通路
Fig. 6 The control path of multi-branch linear pipeline

3 实验结果及分析

本节采用上述异步流水线设计流程对单分支线性、多分支线性与环形三种结构的流水线进行异步化设计。单分支线性流水线的设计实例为线性流水乘法器,多分支线性流水线的设计实例为多功能 ALU,环形流水线的设计实例为迭代除法器。乘法器基于改进的 Booth 编码和 Wallace 树结构实现,具有较高的性能;ALU 集成了基本逻辑操作、加减法操作及多媒体操作;除法器基于不恢复余数除法算法实现。

本文使用 Mentor 公司的 Calibre 工具从流水线的版图中提取带寄生参数的网表,然后使用 Synopsys

公司的 Nanosim 工具对网表进行功耗仿真。由于功能单元的功耗与其处理的数据输入密度相关,本文分别测量 10 组不同测试向量,然后计算出流水单元的平均功耗。三种流水单元的异步与同步版本的面积、性能及功耗比较如表 3 所示。

表 3 三种流水单元的同步及异步比较

Tab. 3 Comparison of asynchronous and synchronous pipelines

	ALU		流水乘法		迭代除法	
	同步	异步	同步	异步	同步	异步
面积(nm^2)	0.321	0.342	0.153	0.171	0.125	0.141
平均周期(ns)	2.13	2.21	1.97	1.91	2.03	2.18
平均功耗(mW)	45.58	23.85	25.37	12.43	17.87	9.89

从表 3 可以看出,异步单元的面积要稍大于同步单元,这部分面积主要是由控制通路增加的。实际上,在进行同步芯片设计时需要建立一个大的全局时钟树,时钟树占据了整个芯片面积相当大的部分,因此实际的同步单元面积需要考虑全局时钟树的面积。同步功能单元的性能受限于延迟最大的流水段在工艺最坏情况下的延迟,而异步功能单元根据不同的输入数据及工作环境具有不同的延迟,因此异步控制通路逻辑的延迟开销并没有使异步功能单元的平均性能降低太多。功耗仿真结果表明,异步单元的功耗比同步单元要小得多,这充分体现了异步电路在功耗方面的优势。异步电路功耗低主要是因为异步电路没有时钟,在无数据输入的情况下几乎不产生动态功耗,而同步电路将在全局时钟的驱动下产生较大的动态功耗。

4 结论

异步流水线设计是异步微处理器设计必须解决的一个难题。本文分析了微处理器中存在的各种流水线结构,并给出了相应的异步流水线结构。然后提出了一种异步流水线设计流程,该设计流程能够快速地将同步流水线设计成异步流水线,提高了设计效率。根据提出的异步流水线设计流程,本文在 $0.18\mu\text{m}$ 工艺下设计实现了具有单分支线性、多分支线性与环形结构的流水线,并进行面积、性能及功耗比较。实验结果表明,异步流水线与同步流水线具有相当的面积与性能,但具有更低的功耗。

参考文献:

- [1] Friedman E G. Clock Distribution Networks in Synchronous Digital Integrated Circuits[J]. Proceedings of the IEEE, 2001, 89(5): 665- 692.
- [2] Gowan M K, Biro L L, Jackson D B. Power Considerations in the Design of the Alpha 21264 Microprocessor[C]// Proceedings of the 35th ACM/IEEE DAC, San Francisco, CA, USA, 1998(6): 726- 731.
- [3] Furber S B, Day P, Garside J D, et al. AMULET1: A Micropipelined ARM[C]// Proceedings of the IEEE Computer Conference, San Francisco, CA, USA, 1994(2): 476- 485.
- [4] Furber S B, Garside J D, Riocreux P, et al. AMULET2c: An Asynchronous Embedded Controller[J]. Proceedings of the IEEE, 1999, 87(2): 243- 256.
- [5] Garside J D, Bainbridge W J, Bardsley A, et al. AMULET3i—An Asynchronous System-on-chip[C]// Proceedings of Async 2000, Eilat, Israel, 2000(4): 162- 175.
- [6] 俞颖,等.采用异步电路的低功耗微控制器的 VLSI 设计与实现[J]. 半导体学报,2001, 22(10): 1346- 1356.
- [7] Cortadella J, Kondratyev A, Lavagno L, et al. De-synchronization: Synthesis of Asynchronous Circuits from Synchronous Specifications[J]. IEEE Transactions on Computer-aided Design, 2006, 25(10): 1904- 1921.
- [8] 李勇,等.一种 32 位异步乘法器的研究与实现[J]. 计算机研究与发展,2006, 43(10): 2152- 2157.
- [9] Furber S B, et al. Four-phase Micropipeline Latch Control Circuits[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1996, 4(2): 247- 253.
- [10] 张晨曦,等.计算机系统结构[M].北京:高等教育出版社,2005.
- [11] Spars J, Furber S. Principles of Asynchronous Circuit Design — A Systems Perspective[M]. Boston: Kluwer Academic Publishers, 2001.