

32 位嵌入式 CISC 微处理器设计

王得利 高德远 张 骏 王党辉
(西北工业大学计算机学院 西安 710072)

摘 要 LongtiumC2 微处理器是西北工业大学自主产权设计的嵌入式 32 位 CISC 微处理器,与 Intel 486DX2 完全兼容,工作频率 133MHz,规模约 100 万门,功耗小于 1W。在微体系结构方面,提出硬连线和微程序相结合的控制通路设计方案,增强了处理器的灵活性和扩展能力。在流水线方面,为了实现精确中断,提出了基于微操作的指令指针跟踪方案,不但可以精确地保存中断现场,而且省去了等待指令边界的时间,实现了中断的快速响应。为了实现 LongtiumC2 的低功耗特性,提出了译码控制核心的低功耗设计方案,使译码器和微内核的功耗分别下降 26% 和 19%。最后,为了快速、完备、有效地对 LongtiumC2 进行功能验证,提出了一种微处理器的系统级验证策略,使用虚拟系统和 FPGA 来同时搭建系统级验证平台,并采用 Vera 进行基于功能点的覆盖率验证,提高了验证工作的效率和置信度。

关键词 微处理器,低功耗,流水线,CISC,验证

中图分类号 TP302 **文献标识码** A

Design of a 32-bit Embedded CISC Microprocessor

WANG De-li GAO De-yuan Zhang Jun WANG Dang-hui

(College of Computer Science, Northwestern Polytechnical University, Xi 'an 710072, China)

Abstract LongtiumC2 is an embedded 32 bit Complex Instruction Set Computer microprocessor designed by Northwestern Polytechnical University, which is compatible with Intel 486DX2 in instruction level, designed to run in 133MHz, and includes about one million gates. Its power consumption is less than one watt. In micro-architecture level, this paper proposed a control logic design scheme, which combined PLA and micro-operation together to improve the flexibility and expansibility of LongtiumC2. In pipeline design, to realize the precise interrupt, this paper proposed an instruction pointer track method based on micro-operation, which leave out the time to wait current instruction's completion, and realize interrupt faster. To implement the low power characteristic for LongtiumC2, this paper proposed a low power design strategy for decode and control core logic, which reduces about 26% and 19% respectively for decoder unit and micro-core unit. Finally, to verify LongtiumC2 fast, completely and effectively, this paper proposed a system level verification strategy for microprocessor, which improves the efficiency and confidence of verification by constructing two kinds of system level verification platform for LongtiumC2 microprocessor using virtual system and FPGA, and processing a function points based coverage verification with Vera.

Keywords Microprocessor, Low power, Pipeline, CISC, Verification

1 引言

嵌入式微处理器在过程控制、工业控制、智能仪器、数据检测与采集等军事和民用领域都有广泛应用,是整个嵌入式系统的核心。目前,国内的处理器研制水平和国外仍存在很大差距,关键芯片受制于人,存在极大安全隐患。因此,设计研制具有自主知识产权的嵌入式微处理器对提高国产微处理器的竞争力和提高国内持续研发能力具有重要意义。

西北工业大学在微处理器设计方面开展了大量工作,设计了 LongtiumC2 微处理器。该处理器兼容 Intel 486DX2 指

令集,时钟频率 133MHz,规模约 100 万门,功耗小于 1W,具有以下特点:

- 采用硬连线和微操作相结合的控制方式,更有效地利用流水线,增强了处理器的灵活性和扩展能力。
- 具有实时、精确的中断响应能力,将中断响应时间减到最低限度,满足嵌入式应用需求。
- 提供存储空间的保护机制,保证了在多任务和虚拟存储器环境下存储器的完整性,为软件诊断错误提供了支持。
- 采用低功耗的控制通路和数据通路设计方案,使处理器的整体功耗小于 1W。

到稿日期:2008-06-25 本文受国家自然科学基金项目(60573107),国家自然科学基金项目(60573143),国家 863 项目(2007aa010402),国家十一五研究项目资助。

王得利(1981-),男,博士研究生,CCF 学生会员,主要研究方向为计算机系统结构、专用集成电路设计,E-mail:wld900@mail.nwpu.edu.cn;高德远(1946-),男,博士生导师,CCF 会员,主要研究方向为计算机系统结构、专用集成电路设计;张 骏(1979-),男,博士研究生,主要研究方向为计算机系统结构、专用集成电路设计;王党辉(1975-),男,博士,副教授,主要研究方向为计算机系统结构、专用集成电路设计。

2 LongtiumC2 的体系结构

LongtiumC2 是与 X86 指令集兼容的 CISC 微处理器,采用全同步设计策略,包括一个 32 位的整数处理单元;一个浮点单元;一个 32kB 大小,8 路组相连的指令数据混合 CACHE;一个存储器管理单元;一个总线接口单元;一个指令译码器和一个微程序控制器;还包括一个保护测试单元和一个异常检测单元。LongtiumC2 的流水线划分为取指、指令译码 I、指令译码 II、物理地址计算、取操作数、执行和写回七级流水,整个指令流水过程由多个功能部件协同完成。图 1 是 LongtiumC2 结构框架。

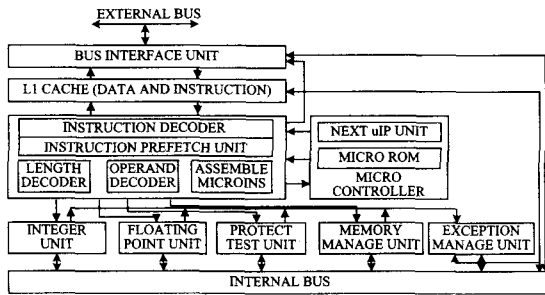


图 1 LongtiumC2 结构框架

32 位的整数单元可以完成 8 位,16 位,32 位数据的完整算术和逻辑运算。大多数的指令可以单拍执行。整数单元内集成了一个 ALU,一个桶式移位器,一个乘法器和一个保护测试单元。装载、写回、加、减、逻辑运算和移位操作可以在一个周期内执行完毕;为了减小对工作频率的压力,乘法固定在 3 个周期执行完毕;除法根据不同类型的除法指令在 8~18 个周期内执行完毕。整数单元和浮点单元间使用一条 32 位的数据线相连。使用这条数据线,两次传送可以组成浮点单元的 64 位操作数。

寄存器堆的规模为 $4 \times 32 \times 32\text{bit}$,包括 4 组 32 位寄存器,每组 32 个寄存器,一共有 128 个 32 位寄存器,外加一个 32 位标志位寄存器。4 组寄存器中包括通用寄存器组、保护寄存器组、常数寄存器组、控制、测试、调试寄存器组。寄存器堆采用两读一写的设计方案,保证一个时钟周期能同时提供两个寄存器源操作数并写回一个结果操作数。

浮点单元支持 16 位整数、32 位整数、64 位整数、单精度浮点数、双精度浮点数、扩展精度浮点数和压缩 BCD 码整数数据格式,符合 IEEE 754 标准,提供不同数据类型的算术运算指令,并且支持超越函数的运算功能。

指令和数据混合 CACHE 大小为 32kB,8 路组相连,每行 16 字节,写通策略,地址命中时一个周期内读出数据。采用伪随机最少使用替换策略。

存储器管理单元同时支持分段和分页,最大寻址空间 4GB。它对地址的管理和对存储空间的保护保证了在多任务和虚拟存储器环境下存储器的完整性。

总线接口单元负责进行数据传输、指令预取和在处理器内部单元与外部进行控制,提供存储器读写,CACHE 行的填充等功能,支持突发传送,带有写缓冲功能,并能动态地在 8 位、16 位和 32 位之间切换总线宽度。

指令译码器从指令预取队列中取出指令进行译码,得到有效地址和微程序入口地址等信息。将整个译码过程分为长

度译码和地址译码两个阶段进行流水译码,在指令不带前缀的情况下单拍完成长度译码,支持任意两条指令并行译码。

微内核接收来自译码器的入口地址,产生一组微操作,这些微操作对执行单元进行控制完成指令执行。能够实现微程序中的微程序跳转、微程序调用、微程序返回和微程序循环执行。提供微操作级的预取和分支预测操作,从而在更大程度上保证微指令的单拍执行。

3 高效的译码控制核心

微处理器中的控制通路对数据通路中各功能单元的操作进行协调控制,规定了每条指令在每个时钟周期中可执行的操作。控制通路的实现有两种基本方式,分别是硬连线方式和微程序控制方式。相应于这两种实现方式,可以分别使用有限状态机和微程序来描述控制通路的设计。

X86 指令集是典型的 CISC 指令集,一般都使用微程序的方式来实现其控制通路。但 X86 指令集中超过 5 个字节的指令很少,其中 1 到 5 个字节的指令占到了 90% 以上^[6],同时根据大量动态程序执行结果,处理器大量的时间仅用于执行简单的指令^[3]。由于简单指令的执行通常在 1~4 个时钟周期范围内,采用硬连线方式将不会显著增加硬连线逻辑的复杂度。因此,提出一种控制通路设计方案,采用硬连线方式直接产生简单指令执行所需的数据通路控制信号,而复杂指令则采用微程序方式实现。这种设计方案结合了 RISC 处理器控制简洁,执行高效和 CISC 处理器微程序控制方式的灵活性好、扩展能力强的特点,既能简化控制逻辑的设计复杂度,又有较好的性能。图 2 是 LongtiumC2 译码控制核心的逻辑结构图,包括指令译码单元(IDU)和微内核(MCU)。

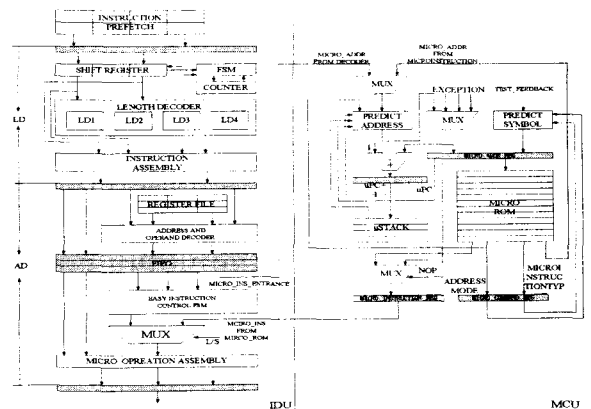


图 2 LongtiumC2 的译码控制核心

指令译码器是微处理器流水线的的数据源,是流水线连续,高效执行的重要保证。针对 X86 系列兼容微处理器串行译码^[4,5]速度慢、效率低的缺点,作者提出了一种并行译码器设计方案。根据 X86 指令编码的特征,任何一条指令,至多得到不是前缀的前 4 个字节就能知道整个指令的长度。所以设置 4 个并行的长度译码单元对一条指令所有可能出现的组合形式进行并行译码,在指令不带前缀的情况下单拍得到指令长度。长度译码和地址译码流水执行,支持任意两条指令并行译码。

LongtiumC2 译码器包括指令预取队列、长度译码部件、地址译码部件。指令预取部件负责产生预取指令的地址,从 CACHE 或者主存中取指令。在 CACHE 不使能或者不命中

的情况下,预取指令总是从主存取指;如果 CACHE 命中,则直接从 CACHE 中取指。在程序发生转移时清空指令预取队列,从转移目标处重新开始取指令。4 个并行的长度译码单元在指令移位寄存器首字节不是前缀的情况下启动译码,单拍得到指令长度。由于 X86 指令编码的正交性,因此不可能出现 4 个并行的长度译码单元对同一条指令译出两个长度的情况。根据得到的指令长度,一条完整的指令被送入地址译码级,并同时开始下条指令的长度译码。地址译码单元和操作数译码单元分别计算访存指令使用的段地址、有效地址、操作数、微程序入口地址等信息,并将结果送入已译码队列等待执行。

根据已译码队列中等待执行指令的信息,分别进入简单指令和复杂指令两个不同的控制流程。简单指令控制状态机接收简单指令,产生一组各流水级使用的控制信号送入微操作流水线,不同流水级选取本级使用的信号,对下一流水级使用的信号仅仅进行传递。执行部件的控制由执行部件自己的控制状态机实现,写回级根据执行部件的控制信号完成相应写回操作。所有流水级由统一的信号控制推进,协调工作。复杂指令则根据微程序入口地址由微内核产生一组微程序进入微操作流水线控制指令执行。

4 流水线设计

LongtiumC2 的流水线划分为取指、指令译码 I、指令译码 II、物理地址计算、取操作数、执行和写回七级流水,其中前三级为指令级流水线,后四级为微操作流水线,如图 3 所示。

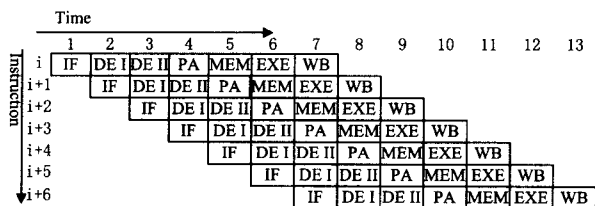


图 3 LongtiumC2 的流水线

LongtiumC2 的流水线设计是依照指令执行的过程按功能和逻辑来进行划分和定义的。这样,可以使得流水线中每个流水级完成的功能明确,各流水级之间的界限清晰,有利于减少流水线中各级之间的存贮信息要求;在保证各流水级功能完整的条件下,尽可能地平衡各流水级的操作。在流水线的功能划分确定以后,流水线可以工作的最高频率是由流水线中延时最大的流水级决定的。实现流水线在时序上的优化的关键是尽可能地平衡流水线各级的延时。

与 RISC 处理器相比,带有变长指令格式的 CISC 体系结构的处理器需要很复杂的指令译码逻辑,这导致了指令的译码逻辑会有较大延迟,为了避免对工作频率产生压力,提高流水线的吞吐率,采用二级译码:第一个译码段确定指令的长度并译出指令的操作码;第二个译码段计算操作数和有效地址并形成控制信号。

流水线相关是影响流水线性能发挥的主要因素,LongtiumC2 的流水线存在 3 种相关,分别是寄存器数据相关、LOAD 相关和条件码寄存器相关。设计中采用了最简单有效的内部推前技术解决寄存器数据相关,使用寄存器记分牌来解决 load 相关,使用条件码记分牌的方式解决条件码相关。对于 X86 指令中一次写回两个结果的指令,在不影响指令执

行延时的情况下,依照结果有效的时机,分次写回至结果寄存器,有效地减少了旁路逻辑的复杂性。

快速、精确的响应中断是嵌入式微处理器的设计需求。为了实现精确中断,作者提出了基于微操作的指令指针跟踪设计方案。在 LongtiumC2 的流水线中,可能有多条微操作同时位于流水线的不同执行部件,这些微操作可能属于一条或者多条指令,所以为了在发生中断时能够正确保存现场,快速响应中断和正确恢复现场,必须设置多个指令指针,从而跟踪各流水级的微操作所对应指令的地址。通过设置的多个指令指针可以明确地得到指令边界,中断到来时等待第一个指令边界出现在微操作流水线中后冲刷第一个指令边界以上的流水级,位于指令边界以下的微操作流水级中的微操作正常执行,后续微操作转入中断转移流程。采用这种基于微操作的指令指针跟踪设计方案处理中断不但可以精确地保存中断现场,而且省去了等待指令边界的时间,实现了中断的快速响应。

5 LongtiumC2 的低功耗设计

在数字 CMOS 电路中,动态功耗占整体功耗的 80%~90%。在无法改变电压的情况下,设计者只能通过降低电路的跳变次数和频率来达到降低功耗的目的。为了降低处理器的功耗,设计者往往将目光集中在数据通路的功能部件上,因为这些功能部件通常包含大量跳变频率极高的寄存器和存储器,如整数执行单元、CACHE 等。LongtiumC2 的低功耗设计不但应用于数据通路,还应用于控制通路。LongtiumC2 的指令译码器和微内核协同工作构成了译码控制核心。针对 X86 指令集的特点,结合 32 位微处理器 LongtiumC2 体系结构,作者分别提出了指令译码器和微内核的低功耗设计方法,相应模块的功耗分别降低了 26% 和 19%。以下着重阐述 LongtiumC2 译码控制核心的低功耗设计策略。

译码器的低功耗设计主要从指令预取部件和已译码队列两方面来进行。根据 SPEC 测试程序,X86 指令集中转移指令大约占到 20%^[2]。由于 LongtiumC2 微处理器没有集成指令级的分支预测单元,因此无论是直接转移还是条件转移,转移的目标地址都将在转移指令的最后一条微操作给出。在执行转移指令并且转移目标不明确的情况下同时进行顺序的指令预取相当于花费大量时钟周期取入了可能不会被执行的指令数据,由此产生的无效预取操作将使指令预取队列中的两级寄存器产生大量无效翻转。设计中在处理器执行转移指令的同时停止无效的指令预取操作,也就是采用一种精确的预取策略,从而降低预取队列寄存器的翻转次数。为了在遇到转移指令时能立即冲刷指令预取队列和指令移位寄存器并停止无效的指令预取和译码操作,采用一个 Branch_en 信号来指明遇到转移指令到转移指令执行完成并给出转移目标地址的过程。在 Branch_en 信号为 1 期间停止指令预取和后续指令译码操作,在 Branch_en 信号被重新置为 0 后再根据得到转移的目标地址重新取指令。设计中只对直接转移指令应用精确预取策略。由于条件转移的不确定性,在执行条件转移时停止指令预取可能会对性能和设计复杂度造成一定影响,而直接转移指令执行时无论是否同时进行预取,最终指令预取队列都会被清空,因此对直接转移指令应用精确预取策略不会对性能造成任何影响。

根据对 SPEC 中对 X86 指令集整数和浮点测试程序的统计表明,整数指令中大约 84%的情况下已译码队列中的立即数字段是无效的;在 61%的情况下偏移量字段是无效的。而在浮点指令中大约有 94%的情况下立即数字段是无效的;在 28%的情况下偏移量字段是无效的^[2]。LongtiumC2 译码器中包含长度为 3、宽度为 158 的已译码队列,其中包括每条指令所带的立即数和偏移量并占到队列总宽度的 40%左右。设计中采用独立使能信号的方法来精确控制写入已译码队列时在队列内部进行移位时对立即数和偏移量字段的更新,这样只在指令带有立即数或者偏移量的情况下相应字段的寄存器才会发生跳变,从而减少了已译码队列中寄存器的跳变次数。

微内核的低功耗设计主要从微程序 ROM、微堆栈和微程序编码几个方面来进行。LongtiumC2 的微程序 ROM 容量为 44×8192 bit,通过 13 位地址线控制。如果采用整块 ROM 存放微程序,每次访问中一个 13 输入 8192 输出的译码电路都将工作,功耗很大。设计中采用分区访问 ROM 的工作方式,将微操作分为 3 类,即 MMU 微操作、IEU(整数部件)微操作、FPU 微操作。根据 SPEC 程序的测试,整数指令有 39%使用存储器操作数,浮点指令中有 72%使用存储器操作数^[2]。在整个微操作系统中,MMU 和 FPU 微操作所占的地址空间均小于 1024,也就是各占整个 ROM 空间的约 12.5%,但它们占了所有 ROM 访问的 40%左右,因此设计时将微操作 ROM 分为 8 块 $44 \text{ bits} \times 1024$ 的分区,将 MMU 和 FPU 类微操作分别放在两个分区中,将 IEU 微操作放在其余 6 个分区中。任何一次 ROM 访问都只有一个分区处于活动状态。采用这种结构后,每个分区的地址宽度下降到 10 位,可以降低分区译码器消耗的功耗,同时因为存储器深度的减小,ROM 中数据线所串联存储单元的数目也将减小为原来的 1/8,该连线的负载电容将极大缩小,因此 ROM 本身的功耗也会降低。

下地址产生逻辑是微内核的核心部分,主要用来产生下一条微操作的地址。而微堆栈是为了实现在微程序级调用和返回所设置的结构,它模拟指令系统中的堆栈结构,是下地址产生逻辑的一部分,使用频率非常高。设计中提出了一种通过指针的变化来完成数据后进先出的堆栈结构,这种结构相对于栈顶位置固定不变、通过数据进行移位操作来完成后进先出的堆栈结构来说只有在进行写操作时,才会对堆栈中一组寄存器进行更新;读操作通过指针的变化来完成,不会使堆栈中寄存器有任何跳变。虽然读逻辑中选择电路增加了部分功耗,但总体功耗还是有明显的下降。

复杂指令执行时微程序 ROM 总是处在被读取的状态,微操作寄存器几乎每拍都要被更新,因此合理地安排微操作编码将有效地降低微操作寄存器和微操作总线的翻转频率,并降低每次读取微程序 ROM 时的功耗。只读存储器单元存储信息的方式是通过在数据线和地之间是否存在充放电通路决定的。LongtiumC2 所使用微程序 ROM 的数据线连接在 V_{CC} 上,所有存储的数据缺省为 1。当要读出一个 0 时,就必然存在一次放电过程,因此一条微操作中包含的 0 位越多,ROM 在完成这次读出操作时需要的功耗也就越大。所以对 LongtiumC2 的微操作编码遵循了两个原则来进行功耗的优化:一个是每条微操作中应该尽量多地包含 1 的个数,这样可以减少 ROM 的读操作功耗;另一个是在执行时,相邻输出的两条微操作应该尽量保证具有较多的相同位,这样可以降低微操作寄存器和微操作总线的翻转频率。

6 LongtiumC2 的验证与综合

Longtium C2 微处理器的验证使用自底向上和层次化验证相结合的方式,将整个验证过程抽象为 3 层,即子模块层、子系统层和系统层。验证从最底层最基本的模块做起,然后将各个基本模块组成子系统做子系统级的验证,最后将整个系统集成起来作系统级的验证。在进行系统级验证的时候,作者提出了一种微处理器的系统级验证策略,使用瑞典计算机科学院和 Virtutech AB 公司开发的命名为 Simics 的虚拟系统^[10]和 FPGA 来同时搭建 Longtium C2 微处理器的系统级验证平台,在不同层次上进行软硬件的协同验证。另外,除了不同形式的系统级验证外,作者还利用 SYNOPSIS 公司的 VERA 软件对电路状态的正确性和多个状态之间转换的合法性进行了验证,更全面地覆盖到了设计中的边角情况,这对仿真验证和 FPGA 验证是一个很好的补充。

Simics 是一种全系统仿真平台,不仅包括多种处理器还包括可以被操作系统和相关设备驱动访问的硬件设备,可以运行真实的固件和完全未经过修改的内核和驱动代码,它在指令级进行抽象,以一条指令的粒度来运行操作系统和应用软件,因而具有很好的性能和自由度。另外,为了缩短发现并解决错误的时间,加快开发进度,我们分别搭建了基于 XILINX VERTEX2 XC2V8000-FF1152-4 和基于 ALTERA STRATIX II EP2S180-F1020-C3 的系统级 FPGA 验证平台,容量分别达到 800 万门和 1000 万门以上,布局布线后运行速度均达到 66MHz。在两个 FPGA 验证平台上,Longtium C2 微处理器均能启动 BIOS,运行 DOS6.22 操作系统,进而运行 DOS 下的应用程序。Longtium C2 微处理器使用这两个平台进行了大量的应用程序测试,纠正了发现的错误之后,运行情况良好。

LongtiumC2 处理器设计频率 133 MHz,使用 SMIC CMOS 0.18 工艺加工,所有 RTL 设计采用 Verilog-HDL 进行描述,使用 LEDA 进行代码检查,VCS 工具进行仿真,VERA 进行功能点覆盖率验证,使用 Design Compiler Ultra 在 SMIC CMOS 0.18 工艺库下进行逻辑综合。综合时采用 TOP_DOWN、子模块级结构化、子模块内逻辑展开的策略。时钟约束 7.4ns,最大扇出数为 30,线负载模型使用 smic18_wl30,面积无约束。根据综合结果,关键路径出现在译码控制通路和浮点处理部件上。关键路径长度达到 7.28 ns,实际运行频率 137.36 MHz,功耗小于 1W,规模约 100 万门,并给 DFT 设计留有余量。采用 Encounter SOC 进行了后端实现。采用该处理器核的一款 SOC 芯片 LongtiumS2 已经流片成功,使用 PQFP256 封装,在样板上已经成功运行 DOS 操作系统,并且实测功耗仅为 0.555W。图 4 为 LongtiumS2 的芯片布局图。

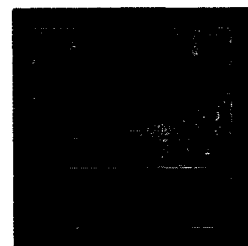


图 4 LongtiumS2 的芯片布局图

(下转第 298 页)

参考文献

- [1] Chen S M, Tan J M. Handling multi - criteria fuzzy decision - making problems based on vague set theory[J]. Fuzzy Sets and Systems, 1994, 67(2):163-172
- [2] Hong D H, Choi C H. Multi-criteria fuzzy decision-making problems based on vague set theory[J]. Fuzzy Sets and Systems, 2000, 114:103-113
- [3] Liu H W, Wang G J. Multi - criteria decision - making methods based on intuitionistic fuzzy sets[J]. European Journal of Operational Research, 2007, 179(1):220-233
- [4] 徐泽水. 直觉模糊偏好信息下的多属性决策途径[J]. 系统工程理论与实践, 2007, 11:62-71
- [5] Chiclana F, Herrera F, Herrera-Viedma E. Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations[J]. Fuzzy Sets and Systems, 1998, 97(1):33-48
- [6] Ishihuchi H, Tanaka M. Multiobjective programming in optimization of the interval objective function[J]. European Journal of Operation Research, 1990, 48:219-225
- [7] 尤天慧, 樊治平. 区间数多指标决策的一种 TOPSIS 方法[J]. 东北大学学报:自然科学版, 2002, 23(2):840-843
- [8] Zadeh L A F. The Concept of a Linguistic Variable and Its Application to Approximate Reasoning [J]. Information Science, 1975, 8(2):199-249, 8(3):301-357, 9(1):43-80
- [9] 张善文, 李晓曼, 雷英杰. 多值直觉模糊集定义[J]. 计算机科学, 2008, 35(1):176-177
- [10] 周启海, 吴红玉. 知识发现与数据挖掘中平均信息测度的创新方法——组合平均[M]. 2006 年科协年会
- [11] 李艳红, 迟忠先, 阎德勤. Vague 相似度量与 Vague 熵[J]. 计算机科学, 2002, 29(12):129-132

(上接第 294 页)

结束语 本文结合 LongtiumC2 处理器设计, 着重对高性能译码控制核心设计、流水线设计、低功耗设计及系统级验证方法几个方面进行论述。分别提出了 PLA 与微操作相结合的控制通路设计方案、并行指令译码器设计方案、基于微操作的指令指针跟踪方案、低功耗指令译码器和微内核设计方案和微处理器系统级验证方案。目前, 这些技术都已在 LongtiumC2 微处理器上成功实现, 为国产嵌入式微处理器的设计积累了宝贵的经验。

参考文献

- [1] 贾琳, 樊晓桢. 32 位 RISC 微处理器流水线设计[J]. 计算机工程与应用, 2005, 14:115-117
- [2] Patterson D A, Hennessy J. Computer Architecture: A Quantitative Approach[M]. Fourth Edition, Morgan Kaufmann Publisher Inc., 2006: B1-B47
- [3] 李三立, 李亚民. RISC 单发射和多发射体系结构[M]. 北京: 清华大学出版社, 1994: 10-30
- [4] 刘诗斌, 高德远, 樊晓桢, 等. 一种嵌入式 MPU 指令译码器设计[J]. 西北工业大学学报, 2001, 2:1-5
- [5] 居小波, 李志斌, 宁兆熙, 等. 一种新型 CISC 微处理器指令译码器设计方法[J]. 微电子学, 2003, 4:154-156
- [6] Stevens K S. An Asynchronous Instruction Length Decoder[J]. Solid state circuits. IEEE, February 2001:217-226
- [7] 张盛兵, 高德远. NRS4000 微处理器的可测试性设计[J]. 西北工业大学学报, 1999, 3:344-349
- [8] 王巍, 高德远. NRS4000 的取指单元的设计[J]. 航空电子技术, 1998, 3:1-4
- [9] 王得利, 高德远. 兼容 X86 指令的 32 位乘法器的分析与设计[J]. 计算机应用研究, 2008, 3:1254-1267
- [10] Magnusson P S, Christensson M, Eskilson J, et al. Simics: A Full System Simulation Platform[J]. Computer, 2002, 35(2):50-58

重视中、英文摘要的编写

国内外公开发行的标准化科技期刊中的文摘已成为科技论文的重要组成部分, 读者可根据文摘提供的信息考虑是否阅读、引用原文; 如能被利用, 才能体现文章的学术价值, 提高原文的引用频次。如此看出学术文章中文摘的重要性, 它所起的作用不可替代。

1. 中文摘要一般为 200~300 字, 英文文摘的长度一般不超过 250 words, 不少于 150 words。

2. 摘要中不涉及图、表、化学结构式以及非公知公用的符号和术语。关键词一般为 3~8 个, 每个关键词首字母大写。

3. 文摘是对文献进行主题分析, 以此体现主题概念、主题内容等该篇文献最重要的信息, 使读者在没有看到全文的情况下, 能够很清楚地了解到该篇文献的中心思想。

4. 文摘语言简洁, 避免重复的单元与措辞; 文摘中的缩写名称在第一次出现时用全称。文字描述中减少对背景信息的介绍; 文摘中不涉及该文献谈及的未来计划; 首句不得简单重复题名中已有的信息。

5. 文摘包含的信息量要完整, 包括目的、过程及方法、结果三方面内容。英文文摘与中文文摘一致, 并使用过去时态叙述作者工作, 现在时态叙述作者结论。