

99,26(9)  
1-4

FPGA 可编程逻辑器件 设计

# FPGA 计算<sup>\*</sup>

FPGA Computing

杨超峰 胡铭曾

(哈尔滨工业大学计算机科学与工程系 哈尔滨 150001)

TP332.1

**Abstract** Since the late 1980s, with the advent of the new kind of programmable devices named FPGA (Field-Programmable Gate Array), a new class of computing systems named FPGA computing systems is proposed. They possess the computing power of application-specific computing systems and the versatility of general-purpose computing systems. This paper is dedicated to FPGA computing systems. First, several representative FPGA computing systems are introduced to characterize the concept of FPGA computing. Second, some important problems in this field are discussed. At last, our current work is outlined.

**Keywords** FPGA, FPGA computing, Reconfigurable computing

## 1. 引言

设计计算机系统时,一个首要的折衷是速度和通用性,具体体现于系统功能的软硬件划分,即什么功能由硬件实现,什么功能由软件实现。硬件实现速度快,而软件实现灵活性大。基于此,计算机系统可以分为两类:通用计算系统和专用计算系统。

不同计算系统的特点集中体现于其处理部件,它代表了由硬件实现的那部分功能。通用计算系统的处理部件是通用的微处理器,如 Intel Pentium、Motorola PowerPC 等,这些处理器能处理一个基本操作集,通过软件编程能完成各种各样复杂的任务,因为任何任务均可分解为一系列基本操作。专用计算系统是面向特定应用设计的,其处理部件是专用芯片,提供了一些针对特定任务的操作,因而处理速度相对于通用系统能高达几个数量级,其弱点是适用面很窄,对于其它一些任务,处理速度可能非常低,甚至无能为力。

概括地说,微处理器在空间维上是固定的,而在时间维上是可变的,因为其功能是固定的,但是通过编程在不同时刻可实现不同的功能;硬连的专用芯片在空间维和时间维上都是固定的。

80年代后期,出现了一类新型的可编程器件——FPGA(Field-Programmable Gate Array,现场可编程门阵列),基于它产生了第三类计算系统,本文称之为FPGA计算系统,它结合了通用和专用计算系统的优点,在空间维和时间维上都是可变的。

## 2. FPGA 计算

FPGA<sup>[1]</sup>是大规模的可编程器件,其基本结构如图1,由逻辑单元和互连线网络两部分组成,两者均可编程,逻辑单元能完成一些基本操作,不同的产

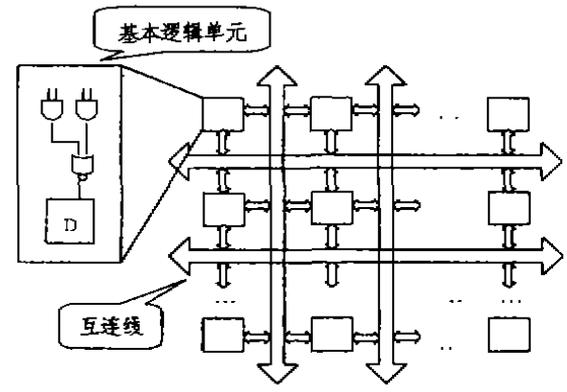


图1 FPGA 结构示意图

\* ) 杨超峰 博士生,研究方向是高性能计算,高性能体系结构,可重构计算等,胡铭曾 博导,研究方向是高性能计算,高性能体系结构,可重构计算,网络计算等。

品其功能不同,逻辑单元通过互连线网络互连,通过对基本逻辑块和互连线网络编程,可以实现非常复杂的逻辑功能。

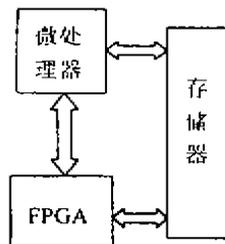


图2 FPGA 计算系统结构框图

FPGA 计算系统如图 2,其基本结构是通用微处理器外加一个 FPGA。新一代基于 SRAM 的 FPGA 不同于以往的可编程器件(如 EPROM),其编程速度非常快,并且还有可能采用其它的实现技术以获得更快的编程速度,因而,根据任务的需要能快速地重构 FPGA 的功能,提供面向任务的运算部件,提高任务的处理速度。可见,FPGA 是系统的计算引擎,对于任务中计算关键(Computation-critical)的部分,由 FPGA 提供的面向任务的功能部件来处理。微处理器仅完成任务中非计算关键的部分,如计算流程控制、简单的标量运算等。因而,用户有两重的灵活性,即微处理器的软件可编程及 FPGA 的硬件可编程,从而可获得专用计算系统的性能,同时又克服了其弱点。

实际上,固定功能计算部件加可重构部件的计算结构,思想并不新,Estrin 所研究的可重构计算<sup>[2]</sup>就是如此,只是当时可重构器件的技术还不成熟。由于其历史原因,很多文献[3~6]对目前这一类新型计算结构仍沿袭可重构计算这一叫法,我们则称其为 FPGA 计算,以区别于其它的可重构技术(如可重构互连网络)。

### 3. FPGA 计算现状

在这一节,介绍几个典型的 FPGA 计算系统,这些实例基本上刻画了 FPGA 计算这一概念。

SPLASH<sup>[7,8]</sup> 是一个 32 个线性互连的 FPGA 构成的扩展板,它插在 Sun 工作站的扩展槽上,编程可实现各式各样的线性 SYSTOLIC 阵列,将任务中计算密集的部分提交给它,可显著地加速 Sun 工作站的计算能力。测试表明,对于字符串匹配、语音识别、DNA 匹配等一些任务可获得很高的加速比,如 DNA 匹配,相对于 Cray-2,加速比达 330。

PRISM-I<sup>[9]</sup> 包括两部分:1)10MHz Motorola 68010 加上 4 个 Xilinx 3090 FPGA 构成的硬件平台;2)一个编译器。编译器是主要部分,它接受一个 C 源程序,输出一个可执行代码及一些硬件影像(Hardware Image),硬件影像对应程序中计算密集的部分,是它们的 FPGA 实现,可执行代码是原程序剔除计算密集部分后的编译结果。可执行代码由微处理器执行,执行过程中将硬件影像装入 FPGA,提供数据,并回收结果。可见,微处理器仅作简单的计算以及任务调度,而 FPGA 给微处理器提供面向任务的扩展指令。实验表明,相对于单一 10MHz Motorola 68010,海明距离、位取反、求 2 位底的对数等十几个任务均获得 10~50 倍的加速比,从而验证了 FPGA 计算这一概念的可行性。

DFPGA (Dynamic FPGA)<sup>[10]</sup> 认为微处理器和 FPGA 应集成在同一芯片中,才能在指令级加速任务的执行;另外,目前商用 FPGA 技术的编程速度太低(约数十毫秒),为了在指令级加速,还需寻求新技术——即 DFPGA。FPGA 中分布着大量的可编程点,在这些点中写入相应的信息,就是编程,所有这些点决定了 FPGA 的功能。当前的 FPGA 中,每个可编程点仅能存储一个信息,而 DFPGA 可存储多个信息,即是多上下文(Multi-Context)的,工作方式采用 SIMD 广播指令的原理,工作时通过广播上下文选择命令来激活相应的上下文,每个上下文使得 DFPGA 具有不同的功能。因此,DFPGA 每个时钟节拍都能改变一次功能,支持指令级重构以及细粒度多线程。

OneChip<sup>[11]</sup> 强调仅仅将微处理器与 FPGA 集成到同一芯片内是不够的,两者还应紧耦合,才能充分开发 FPGA 的潜力。它研究了紧耦合方式:微处理器的固定功能部件与 FPGA 应享有同样的数据流和控制流带宽。OneChip 的原型系统实现于用商用 FPGA 构成的 TM-1 系统上,微处理器类 MIPS,工作频率 1.5M,预期最终实现的工作频率可达 50M,经模拟,对于 DCT 算法,相对于 MIPS R4400,OneChip 可获得 50 倍的加速比。

Garp<sup>[12]</sup> 其基本结构类似 OneChip,即同一芯片上集成紧耦合的一个标准 MIPS 处理器和一个 FPGA。它详细地研究了 FPGA 的实现,提出一种 FPGA 结构:逻辑单元组织成  $N \times 24$  的方阵, $N$  是行数,不同的实现可以不等, $N$  越大 FPGA 可实现的功能越强,Garp 中  $N$  为 32。每个逻辑单元能完成(每个输入数据两位);任意 4 个输入数据的按位逻辑

辑操作、移位、4-路选择器及3个输入数据的加/减/比较运算。整个FPGA可以完成宽达32位(可变)的数据的操作。目前,Garp还没实现,只有一个仿真器,模拟表明,133MHz Garp相对于167MHz Ultra-SPARC1/170,1Mb数据的DES获得24倍加速比,640\*480图像的抖动为9.4,1M记录的排序为2.1。

更多的信息可见:FPGA计算系统列表:[http://www.io.com/~guccione/HW\\_list.html](http://www.io.com/~guccione/HW_list.html);MIT's Transit Project:<http://www.at.mit.edu/projects/transit/rc-home-page.html>;会议FCCM的网址:<http://www.fccm.org>。

早期的系统(SPLASH,PRISM-I)还只是利用商用FPGA构成计算加速板,附加于通用计算系统,以获得高的计算能力。在这些系统中,通用微处理器与FPGA的耦合很松散。后来的系统将微处理器与FPGA集成到同一芯片中,以提高耦合度,并探讨了FPGA的实现,这极大地扩展了FPGA计算这一概念的外延和内涵。如此一来,尽管能进一步开发FPGA计算的潜力,但也引出了很多尚需研究的问题,这在下一节讨论。

#### 4. 一些重要的问题

• **FPGA的结构** 关于FPGA结构的一项重要研究内容是逻辑单元的粒度。如图1示意的逻辑单元是细粒度的,它仅由逻辑门和触发器组成,能完成一些简单的位级逻辑功能,目前的商用FPGA及前面的Garp均属于此类。大粒度FPGA的逻辑单元中包含有ALU和乘法器等功能部件。

细粒度FPGA很灵活,适用于不规则的面向位的逻辑功能,如统计1的个数、找第一个1、复杂的屏蔽操作等。已有细粒度FPGA计算系统的应用表明,加密/解密/数据压缩、序列或串匹配、排序、物理系统的模拟、图像处理等应用领域都能获得很高的加速比。但是,对于复杂的运算,如乘法,细粒度FPGA的利用率很低,速度也低。另外,对其编程,如同设计通常的电路,操作的对象是关于门和触发器的网表,抽象层次太低,将算法的高级行为描述映射成网表的难度很大,特别是对于用户,这是目前的系统均未能走向商业化的一个主要原因。

大粒度FPGA适用于DSP应用领域,如目标识别,其特点是处理的数据为8~12位宽,需要乘法操作。将乘法等复杂功能部件用定制方式实现,所需芯片面积小,速度快。另外抽象层次高,是基于大功能

模块的网表,应用较简单。设计大粒度FPGA的关键是确定必要的功能部件、数据宽度以及功能部件的互连方式,才能有充分的灵活性,支持尽可能多的任务。目前,在实验室中有若干这一类系统<sup>[12~14]</sup>。

很自然地,介于细粒度和大粒度之间应有第三种情况,可是,对前两种情况作出充分的论证之前,不太可能提出有效的方案。

• **FPGA的嵌入** 第3节提到,FPGA计算的研究者倾向于将微处理器和FPGA置入同一芯片内,这产生了如下一些问题:如何分配有限的芯片空间给固定功能部件、片内存储器和FPGA?如何控制?数据带宽多大?目前对这些问题还没有任何结论性的研究成果。

• **FPGA的重构模式** 是指重构速率,有些应用领域,如自动目标识别<sup>[1]</sup>,需要的重构速率很慢(小时、天、星期),而将FPGA作为运算部件时,若干条指令就要重构一次。目前的商用FPGA技术,重构速率大致为几十毫秒,还不足以支持后一种情况。DFPGA是一种解决方案,但是,该方案需要大量存储器来存储配置信息(多上下文)。

当前的FPGA技术,每次编程都是针对整个FPGA的,从而FPGA一个时刻只能实现一个功能。为了提高灵活性,可采用部分可重构技术:将整个FPGA划分成若干子块,对每一块的编程可以独立地进行。这样一来,一个FPGA可以实现几个独立的功能,能很好地支持多线程。另外,若块划分得足够小,编程速度将很快,也解决了前面提到的重构速率问题。但是,这还只处于思想阶段,如何实现仍很不清楚!

• **编译技术** FPGA计算系统中,硬件也是可编程的。但是,对FPGA直接编程,涉及到数字电路的设计,这个工作非常困难且低效,因而,对于用户,这个特性应该透明,用户使用FPGA计算系统应如同使用目前流行的通用计算系统那样,只需利用高级语言编写程序即可。因此需要有有效的编译技术,将用户用高级语言编写的算法的行为描述转化为FPGA实现,并且FPGA资源的调度也是透明的。关于这一点,可以利用现有的很多并行编译技术,如向量编译技术、流水线的编译技术、超长指令字体系结构的编译技术以及SYSTOLIC阵列的综合技术等,不过,如何在FPGA计算中有效地利用、改造、完善这些技术,也是一个难题。

#### 5. 我们正在进行的工作

• **算法到处理器阵列的映射技术**:这项工作对

应于上一节的编译技术,主要内容有:

1) 针对于一些图像处理和图像匹配算法、研究高效的计算结构;2) 研究通用的 SYSTOLIC 阵列的综合技术,特别是线性阵列的综合技术,因为线性阵列结构简单,易于在 FPGA 中实现,并且仅在阵列两端有 I/O 端口,非常易于与其它部件互连。这方面已获得了一些很好的结果。

· 研究 FPGA 的非计算用途,如用 FPGA 实现微处理器的 I/O 接口,灵活性很大,并且可以硬化一些 I/O 指令,加快 I/O 操作。最近开展的一项工作是:利用 DPGA 来实现多线程体系结构的线程切换机制。

· 研究部分可重构 FPGA 的实现技术,并探讨部分可重构 FPGA 对多线程的支持技术,包括 FPGA 资源的组织、管理、调度。从想法上看,这是很有吸引力的,可是,可行性及代价还是一个未知数。

### 参 考 文 献

- 1 Xilinx, The Programmable Gate Array Data Book, 2100 Logic Drive, San Jose, CA 95124, 1994
- 2 Estrin G, et al. Parallel processing in a restructurable computer system. IEEE Trans. Electronic Computer, 1963(Dec.): 747~755
- 3 Turley J. Soft computing reconfigures designer options. Computer Design, 1997(Apr.): 76~81
- 4 Mayer J H. Reconfigurable computing redefines design flexibility. Computer Design, 1997(Feb.): 49~52
- 5 Villasenor J, Mangione-Smith W H. Configurable computing. Scientific American, 1997(June): 66~85
- 6 Mangione-Smith W H, et al. Seeking Solutions in Configurable Computing. IEEE Computer, 1997(12): 38~43
- 7 Gokhale M, et al. SPLASH. A reconfigurable linear logic array. In: Proc. Interl. Conf. on Parallel Processing, 1990. 526~632
- 8 Gokhale M, et al. Building and using a highly parallel programmable logic array. IEEE Computer, 1991, 24(1): 81~89
- 9 Athanas P M, Silverman H F. Processor reconfiguration through instruction-set metamorphosis. IEEE Computer, 1993, 26(3): 11~18
- 10 Dehon A. DPGA-coupled microprocessors: Commodity Ics for the early 21st century. In: Proc. IEEE symposium on FPGAs for custom computing machines, 1994. 31~39
- 11 Wittig R D, Chow P. OneChip. A FPGA processor with reconfigurable logic. In: Proc. IEEE symposium on FPGAs for custom computing machines, 1996. 126~135
- 12 Hauser J R, Wawrzynek J. Garp: A MIPS processor with a reconfigurable coprocessor. In: Proc. IEEE symposium on FPGAs for custom computing machines, 1997. 24~33
- 13 Hartenstein R, Kress R. A datapath synthesis system for the reconfigurable datapath architecture. In: Proc. Asia and South Pacific Design Automation Conf. 1995. 479~484
- 14 Ebeling C, et al. Rapid-Reconfigurable pipeline datapath. In: Proc. Field Programmable Logic. Springer-Verlag, Heidelberg, 1996. 126~135
- 15 Mirsky E, Dehon A. MATRIX: A reconfigurable computing architecture with configurable instruction distribution and deployable resources. In: Proc. IEEE symposium on FPGAs for Custom Computing Machines. IEEE CS Press, Los Alamitos, Calif., 1996. 157~166

(上接第 9 页)

明地回答问题,并通过向用户通报有意义的事件来预测问题。

因此,对数据库研究界今后十年的目标建议如下:把 Web 变成一个在下一个十年中更有用的信息工具,使其易于为每个人使用,以存储、组织、访问和分析大部分的联机人类信息。

为了鼓励在前沿领域的创新性工作,特别是加速研究重点的转变,而且,考虑到以电子方式出版的技术报告已经改变了科技文献的搜集和发布方式,因此建议主要的会议和期刊的程序委员会改变录取文章的方法。

首先,会议和期刊应不再强调其纸介质形式的文集,而应把投稿放在 Web 站点上,加上编辑的评语。建议会议采取所有的文章都张贴、或大部分文章

张贴的方式。大会报告应该分配给那些突破“delta-X”研究模式的新思想,以及在创新领域和开拓新领域方面对最新研究进展的总结。

此外,相信接收更多的文章更有利于信息发布,这将容纳更多的创新性文章,而又不必排除那些会议所钟爱的“delta-X”类结果。

最后,建议要有一个公共审阅过程。一旦一个作者把一篇文章公开,志愿审阅人就应该能在一个适当的论坛上发表他们的审阅意见。这些经过组织的对相关文章的审阅意见对有关的材料进行了比较,将十分有用。

(Asilomar Report 刊于: ACM SIGMOD Record, Volume 27, Number 4, December 1998)