

可重构路由器研究的现状与展望

李玉峰, 邱 菡, 兰巨龙

(国家数字交换系统工程技术研究中心, 郑州 450002)

[摘要] 随着新业务和新应用的不断推出,传统路由器在业务升级能力及升级响应速度方面已不能满足网络发展的需求。实时电路重构技术利用可编程器件可多次重复配置逻辑状态的特性,能够在系统运行时根据需要动态地改变系统的电路结构,利用可重构技术实现的路由器能够支持新业务模块的硬件电路在芯片内动态加载,同时又不影响其他功能模块的正常运行,为传统路由器功能扩展能力差问题提供了解决思路。在介绍可重构技术的基本概念和支撑器件的当前发展概况后,对以往的可重构路由器技术和当前的研究现状进行了总结,并从构建可重构网络的角度分析了进一步的研究方向。

[关键词] 可重构路由器;可重构网络;可编程器件;互联网

[中图分类号] TP393 [文献标识码] A [文章编号] 1009-1742(2008)07-0082-08

1 前言

1.1 可重构的基本思想

可重构是指系统中的硬件功能或硬件功能模块间通信连接可以重新配置,如图 1 所示。

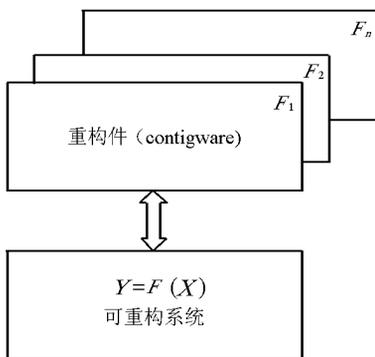


图 1 可重构系统的概念示意图

Fig. 1 Model of the reconfiguration

基本的可重构硬件是一种可重新配置的大规模

集成电路。针对这一硬件可以设计多个硬件电路 F_1, F_2, \dots, F_n , 这些电路的配置数据可以预先设计好并存储在系统的存储器中,称为重构件 (configware)。重构件在系统运行之前或系统运行过程中,根据系统需要配置到可重构硬件中,一旦重构件配置到可重构硬件中,硬件就具有某个重构件规定的功能。

在软件实现的系统中,处理器可以动态地调用当前所需的程序段,具有相当的灵活性,却损失了一定的性能;在硬件实现的系统中,算法被固化成电路,其性能优越但却几乎完全丧失了灵活性。可重构电路可在系统的工作状态下,动态改变电路的结构。这主要是通过对系统中的可编程器件进行重配置或部分重配置实现的。利用这一技术设计的可重构系统,在只增加少量硬件资源或不改变原有硬件设计的情况下,就能将软件实现和硬件实现的优点合二为一,既保留了硬件实现的性能,又兼具软件实现系统的灵活性。表 1 就这三种系统实现方式进行了比较。

[收稿日期] 2007-04-29;修回日期 2007-09-03

[基金项目] “九七三”国家重点基础研究发展计划资助项目(2007CB307102)

[作者简介] 李玉峰(1976-),男,山东烟台市人,国家数字交换系统工程技术研究中心博士研究生,主要研究领域为宽带信息网络,高速路由器核心技术;邱菡(1981-),女,湖北随州市人,国家数字交换系统工程技术研究中心博士研究生,主要研究领域为宽带信息网络,流媒体技术

表1 各种实现方式的性能比较

Table 1 Performance comparison of hardware – based implementation, software – based implementation and run – time reconfiguration

算法实现方式	硬件实现方式	软件实现方式	可重构系统
处理速度	快	慢	较快
并行性	可实现任何意义上的并行	仅能实现指令级并行	可实现大部分的并行方式
资源消耗	根据应用变化	不变	不变
开发难易	难	易	(发展中)
一次性投入成本	大	小	小
可升级性	无	有	有

1.2 可重构系统分类

就目前来说,可重构系统的历史很短,其标准分类形式还没有形成。

一种可重构系统的分类方法是按照重构单元的大小来划分。重构粒度大的为模块级重构,重构时改变某一个或若干个子模块的结构。此时不仅电路逻辑改变,连线资源也重新分配,通常重构时系统可能需要暂停工作。这种重构系统设计简单,但灵活性不足,不能完全发挥出硬件运算的效率。重构单元粒度较小的称为元件级重构,重构时仅改变若干元件的逻辑功能。通常情况下,重构时连线资源的分配状况不作修改,重构时系统可以不暂停,边重构边工作。这种重构系统设计复杂,但灵活性大,能充分发挥出硬件运算的效率,较适合应用于高速数字滤波器、演化计算、定制计算等方面。

按照重新配置方式的不同,可重构系统又可以分为动态可重构系统和静态可重构系统。静态重构必须中断当前系统运行任务来进行新的数据流配置,主要针对具有重构能力但配置数据时速度较慢的器件。动态重构在改变电路的功能同时仍然保证电路的动态接续。常规 SRAM FPGA (static random access memory field programmable gate array) 都可实现静态重构。随着 FPGA 产品和技术的逐渐成熟,动态重构 FPGA 的设计理论和设计方法已经逐渐成为新的研究热点,就其实现重构的面积不同,又可以分为全局重构和局部重构。

1)全局重构。对 FPGA 器件或系统能且只能进行全部的重新配置,在配置过程中,计算的中间结果必须取出存放在额外的存储区,直到新的配置功能全部下载完为止,重构前后电路相互独立,没有关联。

2)局部重构。对重构器件或系统的一部分进

行重新配置,而在此过程中,其余部分的工作状态不受影响。这种重构方式减小了重构范围和单元数目,从而可以大大缩短重构时间,具有相当的优势。

1.3 可重构系统的应用

可重构系统被广泛应用到了很多方面,例如,空间技术的容错系统、军事目标匹配、大数运算、声纳波束合成、基因组匹配、图像纹理填充、集成电路的计算机辅助设计等,已有文献对这些应用进行详细描述。随着应用场合的不同,可重构系统对性能的提高程度也大不相同。一般情况下,按系统中 FP-GA 的芯片数量进行平均,每块 FPGA 能使运算速度提高 7 ~ 30 倍左右,只有少量特别的例子能使速度提得更高^[1]。

2 FPGA—可重构技术的发展基础

2.1 基于 FPGA 的可重构技术简介

专用集成电路 (application specific integrated circuits ,ASIC) 具有高性能、高速度和高可靠性,但是功能固定,微处理器具有强大的可编程功能,但性能很难达到很高。可重构技术基于现场可编程门阵列,即 FPGA。从 FPGA 的结构来看,主要有 2 种:1) 基于反熔丝技术的 FPGA;2) 基于 SRAM 编程的 FP-GA。基于反熔丝技术的 FPGA 用反熔丝开关作基本元件,具有非易失性,不能重构。基于 SRAM 编程的 FPGA 通过阵列中的 SRAM 单元对 FPGA 进行编程。SRAM 单元由一个 RAM 和一个叫做可编程内部连接点 (PIP) 的晶体管组成,RAM 中存储着 PIP 晶体管的通断信息,系统上电时,这些信息由外部电路写入到 FPGA 内部的 RAM 中,电源断开后 RAM 中的数据将丢失。因此 SRAM 编程型 FPGA 是易失性的,每次重新加电,FPGA 都要重组数据,这一特点成为了可重构系统发展的持续驱动力量。

严格来讲,基于 FPGA 的系统重构同样分为静态系统重构和动态系统重构。FPGA 动态可重构技术是指基于 SRAM 编程和专门结构的 FPGA,在一定条件下,不仅具有在系统重新配置电路功能的特性,同时还具有在系统动态重构电路逻辑的能力。基于 FPGA 的动态可重构技术可以使数字系统单片化的设计,从追求逻辑规模转向追求逻辑的分时复用;从专用的固定功能逻辑系统转向功能可自适应进化的逻辑系统。从时间轴或外部看上去,这种新型的逻辑系统虽然和原有整体功能一样;但从资源利用率来讲,由于可以动态重构地复用资源,资

源利用率将成倍地提高,实现系统功能所需的硬件规模大大下降。

2.2 当前 FPGA 对动态可重构技术的支持

以 Xilinx 公司的 FPGA 为例,介绍当前 FPGA 器件对可重构技术的支持水平。

目前, Xilinx 公司已有多款 FPGA,不仅提供了传统的全局重构功能,还能够提供局部动态重构功能(partial reconfiguration, PR)^[2],即 FPGA 能够在运行状态下对系统的一部分进行重新配置,在此过程中,其余部分的工作状态不受影响。这几款支持 PR 的 FPGA 如图 2 所示。Xilinx 公司提供的 PR 功能包括两种模式:基于模块(module-based)模式和基于差异(difference-based)模式^[3]。当局部动态重构模块需要与其他模块进行通信时,Module-based 模式适用。该模式下,利用 Xilinx 公司提供的总线宏(bus macro)技术可以为内部各个模块间提供一个固定的总线,从而保证重构后的连接保持正常。当新设计仅是在原有设计的基础上作微小修改时, Difference-based 模式适用。

- Virtex™, Virtex™-E
- Virtex™-II
- Virtex™-II Pro/X
- Virtex™-4/FX/LX/SX
- Spartan™-II, Spartan™-II E
- Spartan™-3, Spartan™-3E, Spartan™-3L

图 2 支持 PR 功能的 FPGA 系列

Fig. 2 The FPGA families supporting the PR

下面以 Module-based 模式为例,简要介绍 FPGA 动态局部重构的设计方法和实现思路。

首先,从功能上对整个设计进行系统级的动态模块和静态模块的顶层划分,在 FPGA 内部为动态重构模块划定相应的动态配置区域、管脚、逻辑资源和存储资源,定义动态重构模块的接口,如图 3 所示。需要说明的是,该区域划定后不能动态更改,除非下一次重新实现整个 FPGA 设计时方能进行更改。动态模块和静态模块重构后的布线结果示例见图 4。

系统上电并正常运行后才能对局部可重构模块进行动态重新配置,在实施动态配置的过程中,其余正在运行的静态模块可不受影响。

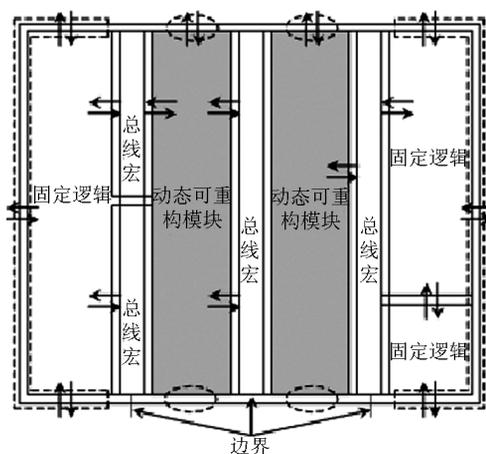


图 3 FPGA 内部区域支持 PR 的设计示意图

Fig. 3 Design with two reconfigurable modules in Xilinx FPGA

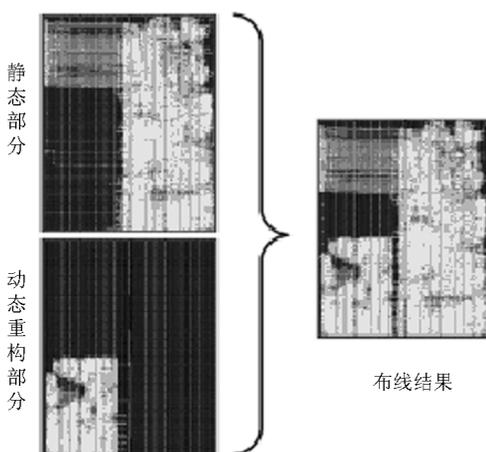


图 4 重构后的布线结果示意图

Fig. 4 Demonstration of routed design for reconfiguration

2.3 小结

近年来,虽然 FPGA 动态重构技术取得了快速的发展,但是有一个问题却始终未能得到有效的解决:当对某一个 PR 模块进行动态重构时,若有别的模块的运行依赖于与该 PR 模块相连接的信号,则在重构期间必然会产生转换时间(transition time),将影响系统功能的连续性,严重时还能影响系统的正常运行。因此,如何减少甚至克服 transition time,是目前 FPGA 动态重构所亟需解决的问题。

还应当看到的是,目前 FPGA 所支持的可重构本质是一种受限制的重构,要求划定重构区域,分模块实现重构,从某种意义上说,仍然是一种基于空分

方式的重构,距离真正意义上的完全重构(时分复用)还有一定距离。

3 以往研究及当前现状

可重构技术的研究主要集中在可重构计算上,许多大学和公司机构进行了相应的研究,给出了很多可重构计算系统的成果及实例。例如:美国加州大学伯克利分校的 BRASS 研究小组开发的 Garp 系统^[4],加州大学提出的 Morphsys 方案^[5, 6],PACT 公司提出的 XPP(extreme processing platform)结构等。

与之相对的是,针对网络和路由器的可重构技术研究很少。概括来说,目前关于该方向的研究成果主要集中于华盛顿大学的可重构网络研究小组。此外,还有许多学者在不同时期对路由器可重构技术做出了很多有意义的探索,下面首先对这些研究作以汇总,然后着重介绍华盛顿大学的研究成果。

3.1 路由器可重构技术的以往研究小结

D. C. Lee 等人在文献[7]中已经提出了可重构路由器(reconfigurable router)的概念,指出动态可重构路由器能够提高网络的性能,具有更好的适应性。在此基础上,文献[8]中提出了一个基于流的可重构路由器的原型结构,并将结构中的可重构硬件部分用 FPGA 实现。此外,宾夕法尼亚大学的 I. Hadzic, J. Smith 等人提出的可编程协议处理流水线 P4 (programmable protocol processing pipeline)^[9, 10],试图通过可配置硬件技术与软件技术的结合,使得设备具有可重构性,并且获得较好的性能和安全性。图 5(a)是 P4 的原理框图。P4 系统包括一系列处理单元 (processing element, PE) 组成的一条流水线和一个用于连接这些 PE 的交换阵列。PE 以基于 RAM 的 FPGA 实现,每个 PE 完成一种协议处理功能。交换阵列的作用是将功能单元进行连接和重组,达到流水线结构的改变,从而达到重构的目的。图 5(b)为 P4 系统的一个接口板实现实例。

考虑到高速路由器的 IP 地址查表模块不仅需要满足高速查表要求,还需要应对频繁的表项更新,而动态可重构的本质特性能够自然的满足这种要求,因此,有部分学者针对这个方向进行了深入的研究,取得了一定的研究成果^[11~13]。还有学者将可重构技术引入到网络协处理器的设计中,这种引入一方面能够减轻网络处理器的任务,提高路由器的性能,同时利用协处理器的动态重构特性还能为路由器提供更灵活的功能支持^[14, 15]。另外,波士顿大学

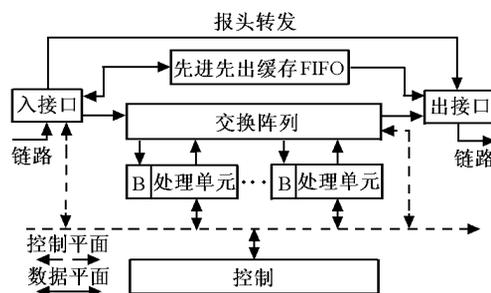


图 5(a) P4 的原理框图

Fig. 5(a) The functional schematic of the P4

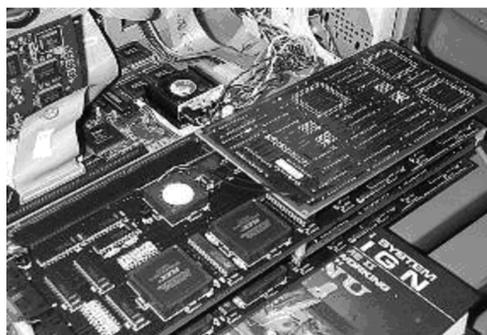


图 5(b) P4 的实现实例

Fig. 5(b) A photograph for the P4

还有学者对可重构的路由器交换设备曾展开过相关项目的研究,但未见公开文献发表。

最后,斯坦福大学的研究人员利用 NETFPGA 和虚拟网络系统 (the virtual network system, VNS) 搭建了一个教学系统,利用 NETFPGA 的远程可重新配置特性为学生提供一个实验平台,使学生能够直接面对路由器软、硬件并对其进行重新进行设计、编程和实验,从而获得第一手的研发经验和知识^[16]。虽然在文献[16]中采用了“reconfigurable networking hardware”的提法,但是,该教学系统并不能实现真正的动态可重构,“可重新配置”的教学系统的提法或许更确切。

3.2 华盛顿大学可重构路由器的主要研究成果

华盛顿大学可重构研究小组的主要研究目标是充分利用可重构硬件的高性能和灵活性,开发新的网络应用。其研究在当前处于领先水平。

3.2.1 FPX 系统概述

在众多的研究成果中,该小组所提出的基于 FPX (field-programmable port extender) 实现的路由器动态可重构解决方案是这些成果的核心和基础^[17~22],如图 6 所示。其中 FPX 系统是实现重构

的核心,当新的网络功能需求产生后,针对该需求所设计的新硬件电路可在 FPX 系统内快速的动态重构。与 FPX 相连的是线卡和交换(washington university gigabit switch, WUGS)。

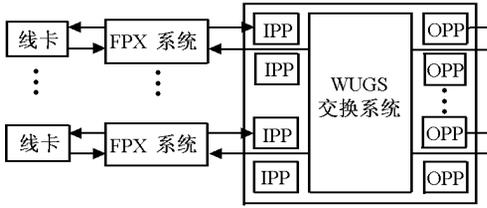


图 6 基于 FPX 平台实现的动态可重构路由器系统

Fig. 6 FPX platform supporting run - time reconfiguration

FPX 系统中,所有逻辑用两片 FPGA 实现,其逻辑结构如图 7(a) 所示。其中一片 FPGA 被称作网络接口设备 NID (networking interface device),NID 利用内部的交换单元实现数据包在线卡、交换和可编程应用设备 RAD (reprogrammable application device)间的灵活传送,此外,其重构控制模块可对 RAD 实施动态重构。另外一片 FPGA 被称 RAD, RAD 中所有数据包处理功能都以模块化方式实现,每一个模块都支持动态重构,可通过全局重构或者局部动态重构的方式对这些模块实现重新配置。图 7(b)是 RAD 的硬件实物图,核心芯片采用 XILINX 的 XCV2000E 实现。

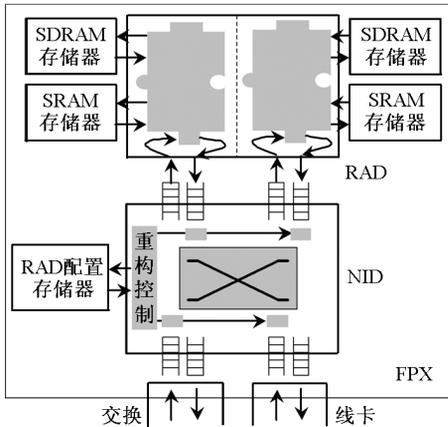


图 7(a) FPX 系统的逻辑结构图

Fig. 7(a) Major components of the FPX

3.2.2 NID 介绍

NID 中利用一个无阻塞的内部交换单元实现了对数据包传输方向的灵活控制。如图 8 所示。在 RAD 全处理模式下,从线卡输入的某一数据包首先通过内部交换单元交换到 RAD 中的某一个功能模

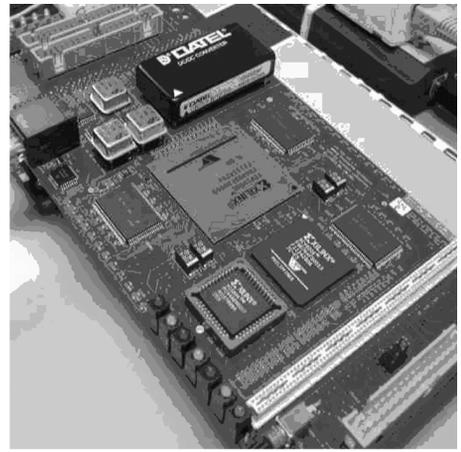
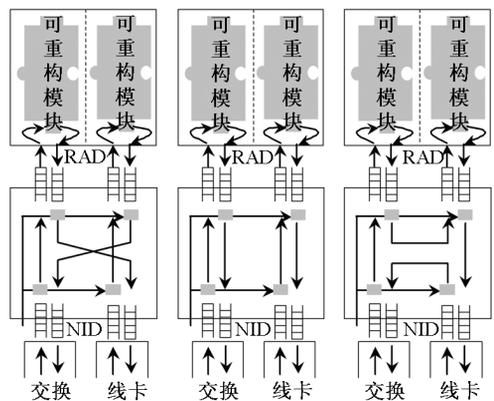


图 7(b) RAD 硬件实物图

Fig. 7(b) A photograph of the RAD

块(例如转发处理模块,实现 IP 包的转发查表,报头检查和替换等功能),完成该模块的处理后输出,然后再经过内部交换单元交换到外部交换网络(Washington university gigabit switch, WUGS)的接口上,最后通过 WUGS 交换到目的端口上。RAD 全回环测试模式是一种支持内部环回测试的模式,可对系统的功能进行自测。若输入的数据包要求 RAD 中的 2 个功能模块都对其进行处理(例如,在传统转发处理模块的基础上,该包还要求完成安全模块的处理),则内部交换结构可采用图 8 中的串行处理模式。



(a) RAD 全处理流程; (b) 全回环测试流程; (c) 重构模块串行处理的流程

图 8 RAD 全处理、全回环测试和重构模块串行处理的流程

Fig. 8 Full RAD processing & full loopback testing & dual egress processing

FPX 系统支持基于网络的远程重构。远程网络重构时,NID 首先利用可靠的传输协议从网络上接收对 RAD 的配置数据,并将配置数据存入 RAD 配置存储器中,配置数据接收完毕后,便可对 RAD 进行初始化和动态配置。

3.2.3 RAD 介绍

路由器的功能(例如,转发处理,安全检测等)在 RAD 中被设计成标准化的可重构模块,这些标准化模块具有统一的内部数据接口和外部存储器接口,如图 9a 所示,具有相同的时序关系。每一个模块都被划分在 RAD 内部的一个区域内,占用该区域的逻辑资源、存储资源和管脚资源等。当存在多个标准化可重构模块后,可利用图 9b 所示的结构对所有可重构模块分别动态重构。需要指出的是,由于各个标准化可重构模块都是独立实现的(占用独立的 FPGA 区域,并且各个模块间不存在相互通信),因此在系统动态增加新的功能模块时,其他功能模块和整个系统的运行将不受影响。

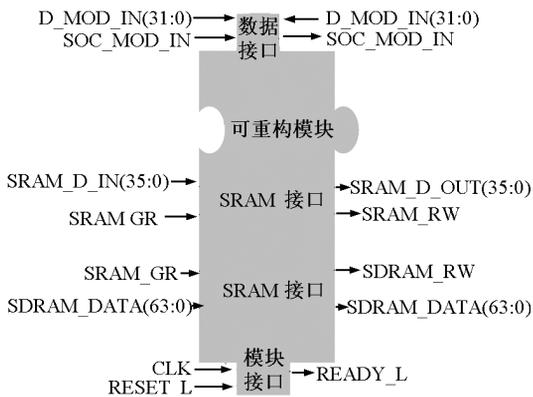


图 9a RAD 内标准化的模块接口

Fig. 9a Modular Component of FPX

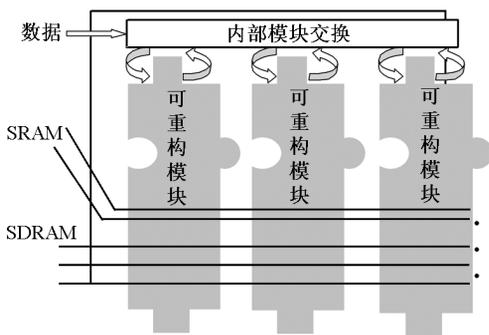


图 9b RAD 内的多模块重构

Fig. 9b Larger Configuration of FPX

3.2.4 系统的软硬件协同

基于 FPX 的硬件包处理模块与软件包处理模块(SPC)可方便地实现协同。如图 10 所示,需要正常处理的数据包可由 FPX 硬件高速处理,而对非正常处理的数据包(不适合硬件实现,或硬件实现代价大的数据包),例如包含扩展选项的 IPv6 数据包则可由 FPX 交给 SPC 处理^[23]。

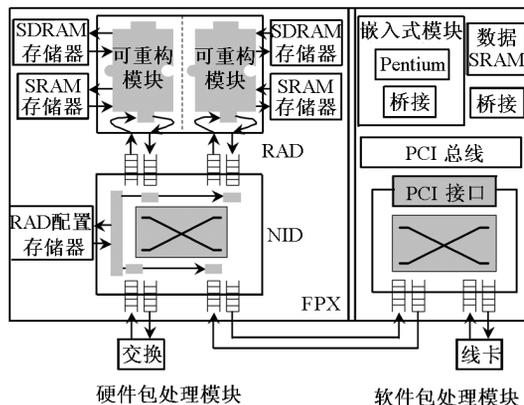


图 10 FPX 系统与基于软件的包处理系统的协同

Fig. 10 Combined FPX/SPC logical configuration

3.3 小结

总的来说,路由器的可重构技术研究正处于起步阶段,以往研究大都是针对路由器的某一关键技术(如查表技术等)给出相应的研究成果。

华盛顿大学的可重构研究小组给出了一种基于 FPX 系统的可重构实现方案。在 NID 内部引入交换单元,控制数据包传输方向;在 RAD 内部定义可重构模块的标准化接口,实现了各功能模块的动态重构。目前,在 FPX 系统之上,许多传统的路由器实现难题已可利用动态重构技术实现。例如,利用可重构技术实现了高性能的文件归类系统^[24],结果表明,该系统能够对 512k 的文件进行快速分类,在 FPGA 采用 Xilinx Virtex4 LX200 的情形下,比基于软件的实现方法快 328 倍。此外,针对基于内容的路由提出了一种可重构的结构,有效提高了速度性能^[25]。针对高速网络中数据流的语义检测提出了一种可重构结构,结果表明该结构可对 1 Gbps 乃至更高网络速率上的数据流进行语义检测^[26]。

另外,由于可重构技术面向的使用环境不同,其在网络研究中的应用策略与可重构计算中的应用策略不尽相同。可重构计算中的重构技术偏重的是时分方式的,而在网络设备中的重构技术则偏

重于空分结构下的重构。

4 研究方向

若将可重构设计思想引入到网络中所有路由器的设计中,将构成一个可重构网络。一旦可重构网络需要修改原有网络功能,增加新的应用或者功能,则不需重新设计新的功能板卡,也不需要改动原有硬件结构,仅需在原有硬件结构上统一设计生成新的动态可重构功能模块,然后将该模块的硬件程序通过网络服务器或者固定的门户网站向全网的可重构路由器发布,路由器在收到发布的信息后进行动态重构即可。图 11 是预想的一种网络动态重构图。

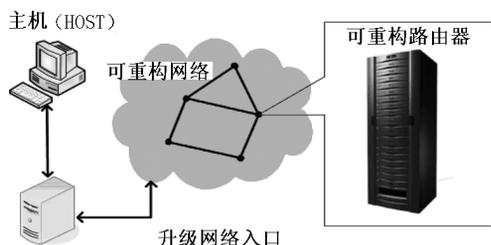


图 11 可重构网络的一种动态重构方案图

Fig. 11 A scheme for dynamic reconfiguration of the network

与传统的网络相比,可重构网络预期具有如下优点:

1) 网络的可扩展性增强。传统网络的特点是规模扩展实现难度小,而网络上的应用功能扩展难度大,且缺乏统一的管理。可重构网络的主要特点就是能够在不改变原有硬件结构的基础上,动态地支持新的网络功能和网络应用加载,且又不影响原有网络功能的运行。这一特点能够从本质上解决传统网络功能扩展难度大的问题,大大降低网络的升级成本,增强网络的可扩展性。

2) 网络可控性增强。可重构网络在传统网络的基础上,不仅能够从软件层面对全网进行控制,而且还能够使得从硬件层面对全网控制成为可能,从而能够有效地消除不同设备提供商网络设备间的异构性互联给网络可控性带来的诸多问题。

3) 网络可管理性增强。在可重构网络中,重构信息的管理中心负责发布全网的动态重构信息,是全网的核心,能够对全网设备从底层硬件电路的基础上进行统一化的管理。

然而,关于可重构网络的研究还未有公开的研究成果。若能够搭建一个可重构的实验网络并在这

个网络上展开相关研究,则可能为未来网络的发展提供一个新的思路和方向。需要完成的相关工作和研究包括:

4) 路由器数据包处理技术的可重构化研究——研究数据包的各种处理任务,包括常规的转发处理,新的安全任务等,研究这些任务形成标准化重构件的可行性。

5) 搭建一个可重构的实验网络,可包括 2 个发展阶段:第一,搭建一个可重构路由器,要求该路由器能够对新的网络应用和网络功能实现硬件的动态重构支持;第二,构建一个可重构的实验网络。预期研究包括可重构网络的拓扑研究,制定全网内可重构硬件的设计规范和要求(要实现全网重构,必须要求网络设备中的可重构硬件设计必须满足一定的设计规范和设计要求),确立全网重构的合理方式,研究全网重构中的重构同步问题,相关的协议制定,重构数据库支持,等等。

当前的网络可管理性差,功能扩展难度大,升级代价高。利用可重构实验网,可展开对全网动态可重构这一设想的深入研究。若该设想能够成立,则可重构网络将能有效的解决这些弊端,实现网络的可扩展、可控制、可管理,而且还将使网络具有更强的适应能力和生命力。

参考文献

- [1] MacVicar D, Singh S. Accelerating DTP with reconfigurable computing engines [A]. Proceedings of the 8th International Workshop on Field-programmable Logic and Applications [C]. Lecture Notes in Computer Science, 1998, 1482(Aug): 391-395
- [2] Xilinx_Corp. Xilinx Development System Reference Guide 8.1i [M/OL]. <http://toolbox.xilinx.com/docsan/xilinx8/books/docs/dev/dev.pdf>, Xilinx data sheet, 2007:113
- [3] Xilinx_Corp. XAPP 290: Two Flows for Partial Reconfiguration: Module Based or Difference Based [M/OL]. <http://www.xilinx.com>, Sept, 2004
- [4] Hauser J, Wawrzynek J. Garp: a MIPS processor with a reconfigurable coprocessor [A]. In IEEE Symp FPGAs for Custom Computing Machines [C]. Napa, California, 1997:16-21
- [5] Singh H, Lee M H, Lu G, et al. MorphoSys: An integrated reconfigurable architecture [A]. Proc NATO System Concepts and Integration [C]. Monterey, CA, 1998
- [6] Singh H, Lee M H, Lu G, et al. MorphoSys: an integrated reconfigurable system for data-parallel and computation-intensive applications [J]. Computers, IEEE Transactions on, 2000, 49(5): 465-481
- [7] Lee D C, Midkiff S F. Reconfigurable Routers: a New Paradigm for Switching Device Architecture. <http://www.cem.ece.vt.edu/>

- [8] Lee D C, Harper S J, Athanas P M, et al. A stream - based reconfigurable router prototype [A]. Proceedings of the IEEE International Conference on Communications [C]. Vancouver, B C, Jun 1999
- [9] Hadzic I, Smith J M. On - the - fly programmable hardware for networks [A]. Proceedings of GLOBECOM [C]. Sydney, NSW, Australia, 1998
- [10] Hadzic I, Smith J M. P4: a platform for FPGA implementation of protocol. Boosters [A]. Proc FPL '97 [C]. Springer LNCS 1304, September 1997:438 - 447
- [11] Fadishei H, Zamani M S, Sabaei M. A novel reconfigurable hardware architecture for ip address lookup [A]. ANCS '05 [C]. Princeton, New Jersey, USA, 2005
- [12] Desai M, Gupta R, Karandikar A, et al. Reconfigurable finite - state machine based IP lookup engine for high - speed router [J]. IEEE Journal on Selected Areas in Communications (JSAC), 2003, 21(4): 501 - 512
- [13] Sangireddy R, Somani A. High - speed IP routing with binary decision diagrams based hardware address lookup engine [J]. IEEE Journal on Selected Areas in Communications (JSAC), 2003, 21(4): 513 - 521
- [14] Maehle E, Albrecht C, Hagenau R. Dynamically reconfigurable coprocessor for network processors [A]. In Work in Progress Session, 29th EUROMICRO Conference and EUROMICRO Symposium on Digital System Design [C]. Institute of System Science, University of Linz, Linz, Austria, 2003
- [15] Albrecht C, Foag J, Koch R, et al. DynaCORE - a dynamically reconfigurable coprocessor architecture for network processors [A]. Proc of the 14th Euromicro Conference on Parallel, Distributed and Network - based Processing (PDP 2006) [C]. IEEE Computer Society, Montbeliard - Sochaux, France, 2006: 101 - 108
- [16] Casado M, Watson G, McKeown N. Reconfigurable Networking Hardware: A Classroom Tool [M]. Hot Interconnects 13, Stanford, August 2005
- [17] Lockwood J W, Naufel N, Turner J S, et al. Reprogrammable network packet processing on the field programmable port extender (FPX) [A]. ACM International Symposium on Field Programmable Gate Arrays (FPGA2001), Monterey, CA, 2001:87 - 93
- [18] Lockwood J W. Evolvable internet hardware platforms [A]. NASA/DoD Workshop on Evolvable Hardware (EHW01) [C]. Long Beach, CA, 2001:271 - 279
- [19] Lockwood J W, Turner J S, Taylor D E. Field programmable port extender (FPX) for distributed routing and queuing [A]. ACM International Symposium on Field Programmable Gate Arrays (FPGA2000) [C]. Monterey, CA, 2000:137 - 144
- [20] Lockwood J W. An open platform for development of network processing modules in reprogrammable hardware [A]. IEC Design Con 2001, Paper WB - 19 [C]. Santa Clara, CA, 2001
- [21] Braun F, Lockwood J, Waldvogel M. Protocol Wrappers for Layered Network Packet Processing in Reconfigurable Hardware [S]. IEEE Micro, Volume 22, Number 3, 2002:66 - 74
- [22] Horta E L, Lockwood J W, Taylor D E, et al. Dynamic hardware plugins in an FPGA with partial run - time reconfiguration [A]. Design Automation Conference (DAC) [C]. New Orleans, LA, 2002
- [23] Deering S, Hinden R. Internet Protocol Version 6 (IPv6) Specification [S]. RFC 2460, December 1998
- [24] Covington G A, Comstock C L G, Levine A A, et al. High speed document clustering in reconfigurable hardware [A]. 16th Annual Conference on Field Programmable Logic and Applications (FPL) [C]. Madrid, Spain, 2006:411 - 417
- [25] Moscola J, Cho Y H, Lockwood J W. A reconfigurable architecture for multi - gigabit speed content - based routing [A]. IEEE Symposium on High Performance Interconnects (Hot Interconnects - 14) [C]. Stanford, CA, 2006:61 - 66
- [26] Cho Y H, Moscola J, Lockwood J W. Context - free grammar based token tagger in reconfigurable devices [A]. Proceedings of International Workshop on Data Engineering (ICDE/SeNS) [C]. Atlanta, GA, 2006

(下转 95 页)