

文章编号: 1001-0920(2007)11-1201-07

## 可重构技术及其在网络控制系统中的应用综述

周纯杰, 向纯洁, 陈 辉, 方华京  
(华中科技大学 控制科学与工程系, 武汉 430074)

**摘 要:** 可重构技术在对系统的维护、优化、扩展以及自适应、自组织、自恢复上有其独特的优势. 对此, 介绍了可重构技术的基本原理、实现方法和应用领域; 阐述了其在网络控制系统中分别对系统、节点和协议的应用, 并对其发展趋势进行了展望.

**关键词:** 可重构; 动态重构; 网络控制系统  
**中图分类号:** TP13      **文献标识码:** A

## Reconfiguration and its applications in networked control systems

ZHOU Chun-jie, XIANG Chun-jie, CHEN Hui, FANG Hua-jing

(Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China. Correspondent: ZHOU Chun-jie, E-mail: cjiezhou@sina.com)

**Abstract:** Reconfiguration has its special advantages in the maintenance, optimization, expanding and the self-adaptation, self-organization and self-resuming of the system. Therefore, the principal, implementation and application field of the reconfiguration are introduced. Then the implementation of reconfiguration to the system, network nodes and the protocols in the networked control systems is presented. Finally, the prospect of its development is given.

**Key words:** Reconfiguration; Dynamic reconfiguration; Networked control systems

### 1 引 言

可重构概念最早由美国加利福尼亚大学的 Geraid Estrin 在 20 世纪 60 年代末提出. 由于当时实现技术尚不完善, Estrin 研制的可重构系统只是理论设计的粗略近似<sup>[1]</sup>. 直到 1975 年, Merlin 和 Back 将重构技术应用于配电网<sup>[2]</sup>, 才得以与实际系统相结合, 并显示出其对系统性能的优化和维护上的优势. 此后, 越来越多的研究者对重构技术进行了不断探索, 得出了很多宝贵结论, 并将该技术推广应用用于很多领域.

重构是指“对软件内部结构的一种调整, 目的是在不改变软件之可察行为前提下, 提高其可理解性, 降低其修改成本”<sup>[3]</sup>. 基于上述定义, 本文提出了可重构概念. 可重构就是使用一系列准则和手法, 对系统内部的结构或代码进行调整或改进, 以满足系统对环境变化和应用对象变化的适应性要求. 可重构技术提供了一种高效且可受控的系统整理技术, 从

而使系统更容易被理解和修改.

重构一般可分 4 个阶段: 重构检测, 重构规划, 重构实现和重构评价. 其中: 重构规划和重构实现是重点, 在线重构检测和在线重构评价是难点.

可重构技术之所以被广泛应用, 是因为系统或软件的规模越来越大, 对其用常规方法进行优化和扩展存在一定困难. 其次, 系统或软件的工作环境复杂多变, 可重构技术可使系统自动适应这种变化. 具体原因可归为如下 4 类:

1) 对系统维护的需要. 很多系统在最初设计时无法全面考虑到日后一些功能扩展的需要, 或者设计人员在开始设计时并没有完全理解整体系统的结构, 因而需要可重构技术对系统进行维护和扩展.

2) 降低系统成本的需要. 功能强成本低的系统在市场占有绝对优势. 可重构技术可使系统自适应工作环境和对象的变化, 而无需增加额外的费用, 这样在增强功能的同时还降低了成本.

3) 系统智能化的需要. 智能化是目前控制领域

收稿日期: 2006-09-26; 修回日期: 2007-03-08.

基金项目: 国家自然科学基金项目(60674081, 60574088).

作者简介: 周纯杰(1965—), 男, 湖北黄冈人, 教授, 博士, 从事现场总线技术及应用、计算机网络等研究; 向纯洁(1982—), 女, 湖北武汉人, 硕士生, 从事网络控制系统的研究.

发展的大方向,设计具有自组织、自学习和自愈能力的系统,节点和协议需要应用可重构技术。

4) 帮助找到 bugs. 一个结构良好、可读性强的系统,很容易发现系统的 bugs.

## 2 可重构技术的实现原理和方法

### 2.1 可重构技术的实现原理

可重构技术按照其实现原理可分为静态重构和动态重构两种. 如果重构必须在中断程序的情况下运行,称其为静态重构;如果重构过程可与程序执行同时进行,则称其为动态重构<sup>[4]</sup>. 在实际应用中,静态重构常与动态重构一起混合使用:静态重构产生一个系统运行的初始配置和方案;在系统运行过程中,动态重构根据系统的对象的运行状况和环境的变化产生一个优化的配置或方案.

#### 2.1.1 静态重构

静态重构在更新系统信息时,为避免死锁现象的出现,需要中断程序或停止用户数据的传输. 比较典型的是 Auto-net,在这种网络中,一旦拓扑结构发生了重大的变化,就会触发启动一个分布式重构协议,并将其下载到整个网络,更新主机和交换机中的信息<sup>[5,6]</sup>. 在重构过程中为了避免死锁,该协议将会在重构过程开始之前停止用户的应用程序数据的相关传输任务,待重构过程结束后再继续运行,其后果是使系统的整体性能下降. 但由于静态重构技术的应用和配置相对简单,该技术在很多系统中还是得到了广泛的应用. 例如,在一些相对较为单一或对实时性要求不高的系统上可以继续使用,或者寻求一种更快速的且能满足随机性的重构算法以减少更新消息的时间,最大限度地降低重构时间,减少静态重构过程对系统性能的损坏.

#### 2.1.2 动态重构

当系统信息变化相当频繁时,采用静态重构算法需要多次打断系统程序,这对系统的性能会造成很大的影响. 因此研究人员提出了动态重构的概念,在更新系统信息时无需中断系统程序.

无论是静态还是动态重构技术,都要考虑死锁的问题. 如前所述,静态重构技术是通过停止程序运行来避免死锁的. 而在动态重构技术中死锁是极其重要的问题,要实现动态重构的无死锁,可采用以下3种基本途径<sup>[5,6]</sup>: 1) 避免系统资源之间的周期相关性; 2) 允许系统资源之间的周期相关性,但要避免永久死锁; 3) 将系统资源分成两部分,分别进行静态重构.

## 2.2 可重构技术的具体实现方法

控制系统本身是由硬件和软件组成的,要使控制系统可重构,其具体实现方法是使硬件和软件均

可重构,或两者协同重构.

### 2.2.1 硬件重构

硬件重构是通过硬件来实现可重构功能. 这里的硬件可以是整个全局系统,也可以是具体的某个设备,或者一个接口.

目前主要有以下3种硬件重构的方法:

1) 利用 CPLD 芯片,用户根据不同的需要而自行构造有逻辑功能的数字集成电路. 经过几十年的发展,许多公司都开发出了 CPLD 可编程逻辑器件<sup>[7]</sup>.

2) 利用 FPGA (Field Programmable Gate Array) 可无限次反复擦写编程的功能.

3) 利用 ISP (In System Programmability) 逻辑器件. 该种器件具有在用户自己设计的目标系统中或线路板上,为重新构造设计逻辑,对器件进行编程或反复编程的能力. 采用 ISP 技术后,硬件设计变得如同软件设计一样灵活且易于修改. 硬件功能也可以实时地加以更新或按预定的程序改变配置.

考虑到成本以及重构能力的原因,采用 FPGA 进行重构得到了广泛应用. 所谓 FPGA 可重构技术,就是对基于 SRAM 编程技术 FPGA 的全部或部分逻辑资源实现系统的功能转换<sup>[8]</sup>. 就其实现重构面积的不同,可分为 FPGA 全局重构和局部重构. 在已有的可重构 FPGA 结构中,比较典型的有:

1) M1 芯片. 由美国 California 大学的 Morphoing System 工程提出的一种粗粒度、多重 context 可重构结构<sup>[9]</sup>.

2) SIDSA FIPSOC 结构. SIDNA 工程提出的一种粗粒度结构的 FPGA<sup>[10]</sup>.

3) DPFGA. 由麻省理工大学的 Transit 工程提出的划分结构为二维阵列单元的 DR-FPGA<sup>[11]</sup>.

4) Virtex 系列的 FPGA 结构. 由 Xilinx 公司推出的一系列可实现动态局部重构的 FPGA 芯片.

针对以上4种典型的 FPGA 结构,有两种主要的重构方法: 1) Off-Chip 重构<sup>[12]</sup>,通过串口或并口将配置信息传送到芯片的配置存储器中实现重构; 2) Context 重构,通过交换配置存储器和片上 context 存储器的部分或全部的配置数据实现重构.

利用 FPGA 的重构能力,可以设计很多可重构的硬件设备. 文献[13]设计了一种可重构系统来兼容不同的现场总线,减少了组合的费用. 这套可重构系统包括总线仲裁器、总线检测、TBx 和标准电阻匹配电路,并且给出了系统的工作协议 WorldFIP. 文献[14]把动态 FPGA 重构的应用扩展到接口领域,使用 RCU 在运行时自动触发重构进程.

可重构计算也是 FPGA 的一个应用领域,是指

依靠软件来改变硬件结构,以适应具体应用的计算平台,又称之为自适应计算.目前有以下两种方案:

1) Garp. 由美国加州大学伯克利分校的 BRASS 研究小组开发的将一个 MIPS 处理器和细粒度 FPGA 结合在一起的系统<sup>[15]</sup>.它通过将可重构计算单元嵌入到传统的 RISC 处理器中来对某些领域的计算进行加速.

2) Morphsys. 另一个由加州大学提出的方案<sup>[16,17]</sup>,采用可重构元素(RC)阵列作为其可重构处理单元.

### 2.2.2 软件重构

软件重构技术是软件工程近 10 年来非常活跃的一个研究领域,这些研究主要包括软件重用、软件维护和软件的重新组织.软件重构是指在不改变软件的功能和外部可见性的情况下,为了改善软件的结构,提高清晰性、可扩展性和可重用性而对软件进行的改造.

软件重构可通过支持脚本语言、虚拟机、原生代码和可加载模块等多种形式来实现.具体的重构方法有:重构函数,在对象之间搬移特性;重构数据,简化条件表达式,简化函数调用,处理概括关系等.美国著名学者马丁·福勒<sup>[3]</sup>在其著作《重构——改善既有代码的设计》中对软件重构的原因、方法和应用都做了详细的介绍,这里不再赘述.

### 2.2.3 软硬件协同重构

嵌入式系统在控制系统中得到了广泛应用.嵌入式系统中传统的先硬件后软件的设计方法很难充分利用系统软硬件资源,难以适应现代复杂的、大规模的系统设计任务.

随着微电子技术的发展,嵌入式系统软硬件协同设计成为现实,这也使软硬件协同重构成为可能.所谓软硬件协同重构就是让软件重构和硬件重构作为一个整体来考虑,并行重构,找到软硬件的最佳结合点,从而使系统高效低成本工作.目前这方面的研究工作才起步,文献较少.

### 2.3 可重构技术的应用

由于可重构技术满足系统对环境 and 应用对象变化的适应性要求,提高了系统的智能化和可维护性,可重构技术在系统的设计中得到了广泛应用,如在制造业、各种工业过程的控制系统、各种应用软件的开发、各种网络系统开发和维护等.

## 3 可重构技术在网络控制系统中的应用

网络控制系统(NCS)是信息时代控制系统的主要形式,是网络控制节点通过通信介质按照一定的通信协议进行互连的.由于网络控制系统所控制的应用对象的多样性和应用环境的不可预知性,使得

传统的设计方法所设计的网络控制系统要么不能适应应用对象和环境的变化,要么设计的网络控制系统(考虑所兼容的各种对象和环境)成本过高.另外网络控制系统还没有一个统一的通信协议标准,即多种通信协议标准共存(至少几十种以上),这也使得网络控制系统在应用时互连存在问题.可重构技术为解决以上问题提供了一条有效实用的重要途径和方法.在网络控制系统中,可重构技术可在 3 个不同层次上进行应用:网络控制系统整体重构,节点重构以及节点的局部接口或功能重构(如通信协议栈重构).下面分别就系统重构、节点重构和协议栈重构来介绍可重构技术在网络控制系统中的应用.

### 3.1 网络控制系统的系统重构

将 NCS 作为一个整体来重构可使整个系统更好地适应工作环境和控制对象的变化.对 NCS 系统重构的研究内容主要有:可重构系统的体系结构,可重构系统的建模和评价及优化,可重构系统的重构算法,可重构网络系统中节点协调与控制等.

文献[18]结合机器人集成系统,给出一种可重构网络系统的体系结构,该结构可对物理结构的重构进行快速响应;文献[19]介绍了采用 Petri 网对可重构网络制造系统的建模和分析方法;文献[20]介绍了一种可重构网络制造系统的资源分配算法,利用着色 Petri 网,在资源分配结果的基础上,对网络制造系统的建模和性能进行评价;文献[21]探讨用于加工过程的快速重组网络制造系统的原理、建模技术、结构体系和运行决策的优化规划,提出了适合敏捷制造环境下,实现系统重组的组态式柔性制造单元的原型框架和应用案例.

目前对 NCS 整体重构的方案还不是很多,主要原因在于对系统整体重构需要全局考虑问题,需要各节点的协调,模型非常复杂,涉及的范围很广,对技术人员的要求很高.如果能突破这个瓶颈,可重构的研究工作将会达到一个新的高度.

### 3.2 NCS 的节点重构

在 NCS 中,网络控制节点通过通信介质的连接组成了网络控制系统.有时并不需要重构整个 NCS,而只需重构系统的节点就能满足系统需要.网络控制系统的节点设计主要包括硬件设计、软件设计和软硬件协同设计 3 个方面,因此对于节点的重构也主要从这 3 个方面考虑.

节点的硬件重构可参考 2.2.1 中的方法重构节点设备或接口.文献[14]给出的通信接口重构方法应用了 Xilinx FPGA 芯片提供的部分重构机制,同时也开发了一种新的控制单元,在系统运行时自动触发重构进程.该重构方法包括以下 5 个内容:1)怎

样看待接口的重构要求; 2) 重构要求如何被提出和具体化; 3) 怎样暂时终止被重构接口的通信; 4) 怎样确保重构后新接口的通信连接; 5) 怎样不丢包。这种接口的重构方法可应用于 NCS 的接口重构, 使之满足通信的需求。文献[22]将应用于 NCS 节点的硬件重构方法三维化, 即任务和模块为两个空间维, 第 3 维是时间, 在 3 者的共同作用下, 利用 LEF (Less flexibility first principle) 算法对指定硬件进行动态重构。文献[23]利用嵌入式芯片和 FPGA 的重构能力对 NCS 的硬件层进行实时动态重构。

节点的软件重构包括 NCS 操作系统的重构, 控制软件的重构等。文献[24]在传感器网络中提出用 SNRAMoLa (Sensor Network Reconfiguration Architecture Modeling Language), 一种传感器网络可重构构架模块化语言, 来重构软件。这种思路可以借鉴于 NCS 的软件重构中。

软硬件的协同重构是节点重构的重要方向, 通信协议栈的重构属于此类, 具体内容将在 3.4 节介绍。

### 3.3 可重构过程中的任务调度问题

在网络控制系统中, 不同的控制任务有不同的实时性要求, 节点的重构和应用程序任务的重新分配都会改变实时通信的要求, 继而影响信息的调度。

文献[25]在理论上研究了网络化控制系统的调度问题, 分析了静态调度方法的种种弊端, 然后提出一种基于事件触发的动态调度方法, 并通过建模与仿真说明其性能的优越。文献[26]针对网络化过程控制系统分析了网络化控制系统通信调度的一个基本参数: 最大允许延迟时间。该参数要满足系统的稳定性要求。在此基础上提出了调整采样周期, 分配带宽来传输 3 种不同类型的实时数据(周期数据、非周期数据以及报文)。文献[27]研究了多现场总线实时系统的任务调度分配问题, 采用的方法是, 任务通过按照时间触发机制的固定优先级调度表进行调度, 使任务的分配与调度可同时实现, 并提出了一个自动进行分配任务及其优先级的方法。

信息调度发生在网络层或传输层之上, 这就关系到协议和协议栈的问题。

### 3.4 NCS 的协议及协议栈重构

NCS 是由一系列网络节点按照一定的通信协议进行互连的, 目前是一个多种通信协议标准共存的局面。一般情况下, 协议是静态的, 即通信系统的通信协议构造完成后, 以硬代码的形式连同节点的其他代码存放在节点内, 协议栈的代码不能单独修改或被替换。但随着人们对通信系统维护性能和节点协议兼容性要求的提高, 静态协议已不能满足实

际应用的需要, 协议重构是解决该问题的有效途径和方法。

构造通信协议是进行协议重构的前提和基础, 协议是节点互相通信行为的规则和格式, 协议模型技术是协议重构的核心和基础。形式描述语言总是以某种模型技术为基础, 协议正确性验证必定基于某种模型技术, 协议自动化实现以及协议测试等都以某种模型技术为基础<sup>[28]</sup>。

鉴于通信系统的复杂程度不断提高, 通信协议模型一般采用层次模型结构, 参照 ISO 七层协议的参考模型, 协议模型可分为若干层<sup>[29]</sup>。协议模型应能准确描述协议的 4 个方面: 活动性、安全性、一致性和完整性。运用抽象模型技术, 包含协议的输入/输出模型、协议内部/外部模型、状态转移模型等, 理想的协议模型应能方便并充分地表示协议元素和通道的各种性质, 并且容易观察协议性质。目前主要的协议模型技术有: 状态转移机 (FSM), 通信进度演算 (CCS) 和 Petri 网等<sup>[28]</sup>。Petri 网是一个图形化的模型工具, 它适宜描述那些并发、异步的不确定性的分布式信息处理系统, 可应用于协议模型的分析 and 评价<sup>[30,31]</sup>, 如文献[32]提出的基于着色 Petri 网的 TCP 的拥塞控制模型, 并给出了模型的校验算法。

当前对照 ISO-OSI 网络模型的 7 层协议, 从最底层的物理层到第 7 层应用层都有相应重构方法的相关探索。文献[33,34]介绍了应用层协议重构框架的设计, 文献[35]介绍了基于 FPGA 的物理层协议重构方法。文献[36]研究了网络协议构件化的问题, 把构件思想与网络协议相结合, 给出了一个面向对象的 MCS 基本框架, 并以此提出了基于构件的网络通信系统模型。该模型可将单块式设计的协议转变成一个基于构件的协议, 基于构件的协议具备更好的可复用性和可扩展性, 为应用层通信协议的重构提供了便利的方法。文献[37]提出了一种基于标签的树形结构数据格式, 它类似于 XML 的 DOM 树结构, 能进行节点寻址, 具有即插即用功能, 兼容各种不同的现场总线, 支持多播通信, 是属于整个网络的统一语法格式。但该方法也有极大的局限性, 它只适用于能被这种结构压缩的数据, 当遇到某些报文结构固定且不能改变的现场总线系统时, 它也不得不使用相关硬件进行协议转换。尽管如此, 它毕竟为从应用层协议的角度解决相关问题进行了初步探索, 积累了相当多的经验。

协议栈中的网络层负责解决网络拓扑和路由方面的问题。因此网络拓扑的重构和路由的变化都在这一层决定。网络重构技术是指采用各种重构技术使网络在网元出现故障或失效的情况下, 恢复到可

接受的性能指标的水平,或重构成另一网络,以保证整个网络的不间断运行.可见网络重构技术是保证网络生存性的一个重要内容,其目的是为了提 高控制系统的性能、网络的容错能力和通信的可靠性.目前对这方面的研究很多,如文献[5,38,39]提出了 UP/DOWN 路由算法加上 PPR (Partial Progressive Reconfiguration) 协议的动态拓扑重构方法.UP/DOWN 路由算法是一种部分自适应的无死锁路由算法,适用于规则和不规则拓扑结构,能实时重构路由表.在改变路由表之前,需修正每个链接的方向,以从旧的拓扑图转为新的拓扑图.而 PPR 协议是一个动态重构协议,当拓扑结构改变时,它无需像静态重构方法那样停止应用程序的数据传输,可直接改变链接方向,直到找到符合新的拓扑结构定义的有向图.文献[40]则针对实时变化的拓扑结构提出了一套自己的网络节点自动注册协议.该协议允许新接入的节点通过与相邻节点通信进行注册,生成新的路由表.该协议能避免冲突,减少调度表中的通道数目,而无需知道初始拓扑情况.传感器一旦检测到拓扑变化,就确认自己的邻居,然后根据一套自定义的规则进行相互的通信以确认该事件.这样就无需对所有节点进行广播通信,从而减少了产生冲突的可能性,并基于该变化产生新的数据.文献[41]为分布式传感器网络设计了一种以数据为中心的网络路由协议——定向传播协议.它能实现分布式服务与拓扑结构动态改变的网络数据存取,它对网络拓扑结构的变化非常敏感,按要求更新路由信息.文献[42]又提出一种使用动态重构算法 NetRec 的扩展算法.该算法用来动态寻找并替换失效路径的路由算法,能异步更新路由表,只需每个节点有单独的 ID 号并知道其邻居以及两个跳跃以内所有节点的 ID 号,而无需知道拓扑整体情况.

协议重构和协议栈重构是近几年网络系统研究的热点课题,尤其是无线网络和流媒体<sup>[43,45]</sup>.目前,协议重构的研究热点是动态协议重构,其难点和重点在于协议栈的体系结构、重构的在线触发、协议在线评价及优化等方面.文献[42]介绍了基于组件技术的通信系统的动态协议栈的设计及实现.文献[43]介绍了无线网络终端的动态协议栈体系结构的设计、实现及校验.

#### 4 发展趋势和研究方向

可重构技术是面向技术开发的应用基础研究,也是理论和实际相结合的研究,因此可重构技术的研究应以实际应用为导向,理论研究与技术开发并重.

可重构技术是面向系统设计和开发的应用技

术,目标是提高系统的开放性、可维护性、可扩展性及系统对对象和环境的适应性.以下是可重构技术的重要研究方向:

1) 可重构系统的建模和性能评价.可重构系统是根据应用对象和环境的变化,在现有系统的基础上,在系统规划和设计规定的范围内,通过系统自身或第 3 方调整系统的控制策略,以较低的重构成本满足系统控制要求.当应用对象和环境发生变化时,如何根据一定的指标快速确定系统的重构方案,是可重构系统的关键.为了评价一个待选方案的性能,需首先描述系统资源分布的模型,建立能精确揭示可重构系统的动态随机模型是可重构系统的基础.在此基础上,评价重构方案的性能,是该系统的关键技术之一,也是可重构方案优化的基础.

2) 可重构系统的自组织理论及智能化优化算法.可重构系统主要是依靠自身完成对应用对象和环境的自适应,所以系统的自组织是系统可重构的前提.其研究内容包括:自检测、自规划、自实现和自评价及自优化,前面提到的系统建模及评价是系统自组织的基础.在系统的自组织过程中,系统的智能优化算法是决定可重构系统性能的关键技术之一.

3) 可重构系统中网络节点的协调与控制理论及方法.网络控制系统将是控制系统的主要形式,尤其是复杂控制系统.网络控制系统是通过网络节点的协调与控制来完成控制任务的,所以对于可重构的网络控制系统而言,网络节点的协调与控制是其核心关键技术,它的研究内容主要有:可重构网络控制系统的体系结构,多智能体的分布式通信任务调度,分布式网络节点的协同控制算法等.

4) 可重构节点的开发.网络控制系统是由网络节点组成的,因此网络节点的开发是网络控制系统开发的关键所在,同样,可重构网络节点的开发是可重构网络系统开发的关键所在.面向各类对象的可重构网络节点的开发是可重构系统应用的核心和根本.

5) 计算机技术和微电子技术在可重构系统中的应用研究.控制系统是由硬件和软件构成的,所以计算机技术和微电子技术的发展对可重构系统的研究关系密切.随着微电子技术和计算机技术的发展,软硬件的协同设计和协同重构应引起开发者的关注.

#### 5 结 语

可重构技术是目前系统控制领域重要的研究方向之一,随着系统的复杂化,可重构技术的应用领域不断扩大.目前,对可重构技术的研究,国内还处于发展阶段,还有许多原理和实际问题有待深入研究.

如何将可重构技术用于生产实际而不仅仅局限于仿真,是当前最迫切的任务之一。

### 参考文献(References)

- [1] Estrin G, Bussell B, Turn R, et al. Parallel processing in a restructurable computer system[J]. IEEE Trans Electronic Computers, 1963, 12(5): 747-755.
- [2] Merlin A, Back H. Search for a minimal-loss operating spanning tree configuration in urban power distribution systems[C]. Proc of 5th Power System Computation Conf. Cambridge, 1975: 2-6.
- [3] Fowler M. 重构——改善既有代码的设计[M]. 北京: 中国电力出版社, 2003.  
(Fowler M. Refactoring — Improving the design of existing code[M]. Beijing: China Electric Power Press, 2003.)
- [4] 段然, 樊晓桢, 高德远, 等. 可重构计算技术及其发展趋势[J]. 计算机应用研究, 2004, 21(8): 14-17.  
(Duan Ran, Fan Xiao-ya, Gao De-yuan, et al. Reconfigurable computing technology and developing trends[J]. Application Research of Computers, 2004, 21(8): 14-17.)
- [5] Casado R, Bermudez A, Duato J, et al. A protocol for deadlock-free dynamic reconfiguration in high-speed local area networks [J]. IEEE Trans on Parallel and Distributed Systems, 2001, 12(2): 115-132.
- [6] Lysne O, Pinkston T M, Duato J. A methodology for developing deadlock-free dynamic network reconfiguration processes [J]. IEEE Trans on Parallel and Distributed Systems, 2005, 16(5): 428-443.
- [7] Amine B, Dominique M. A reconfigurable hardware implementation of tree classifiers based on a custom chip and a CPLD for gas sensors application[C]. 2004 IEEE Region 10 Conf: Analog and Digital Techniques in Electrical Engineering. Chiang Mai, 2004: A32-A35.
- [8] 覃祥菊, 朱明程, 张太镒, 等. FPGA 动态可重构技术原理及实现方法[J]. 电子器件, 2004, 21(2): 277-282.  
(Qin Xiang-ju, Zhu Ming-cheng, Zhang Tai-yi, et al. Analysis of the fundamental and implementation method about dynamic reconfigurable FPGA [J]. J of Electron Devices, 2004, 21(2): 277-282.)
- [9] Rafael M, Fadi J K, Milagros F, et al. A framework for reconfigurable computing: Task scheduling and context management[J]. IEEE Trans on VLSI Systems, 2001, 9(6): 858-873.
- [10] Moreno J M, Madrenas J, Cabestany J, et al. Realization of self-repairing and evolvable hardware structures by means of implicit self-configuration[C]. 1st Proc of NASA/DoD Workshop on Evolvable Hardware. Pasadena, 1999: 182-187.
- [11] Andre Dehon. DPGA-Coupled microprocessors: Commodity ICs for the early 21st century[C]. The Proc of IEEE Symposium on FPGAs for Custom Computing Machines. Napa Valley, 1994: 31-39.
- [12] Lysaght P, Stockwood J. A simulation tool for dynamically reconfigurable field programmable gate arrays [J]. IEEE Trans on Very Large Scale Integration (VLSI) Systems, 1996, 4(3): 381-490.
- [13] Valdes Maria D, Moure Maria J, Dominguez Miguel A, et al. Improving industrial communications using reconfigurable devices [C]. Proc of the 2002 28th Annual Conf of the IEEE Industrial Electronics Society. Sevilla, 2002: 2387-2391.
- [14] Meisel A, Visarius M, Hardt W, et al Self-reconfiguration of communication interfaces[C]. Proc of the 15th IEEE Int Workshop on Rapid Systems Prototyping. Geneva, 2004: 144-150.
- [15] Hauser John R, Wawrzyniek John. Garp: A MIPS processor with a reconfigurable coprocessor[C]. Proc of the 1997 5th Annual IEEE Symposium on Field-Programmable Custom Computing Machines. Napa Valley, 1997: 12-21.
- [16] Singh H, Lee M, Lu G. M, et al. MorphoSys: An integrated reconfigurable architecture[C]. Proc System Concepts and Integration. Monterey, 1998: 134-139.
- [17] Singh H, Lee M H, Lu G M, et al. MorphoSys: An integrated reconfigurable system for data-parallel and computation-intensive applications[J]. IEEE Trans on Computers, 2000, 49(5): 465-481.
- [18] Sugi M, Maeda Y, Aiyama Y, et al. A holonic architecture for easy reconfiguration of robotic assembly system [J]. IEEE Trans on Robotics and Automation, 2003, 19(3): 457-464.
- [19] Park E, Tilbury D M, Khargonekar P P. A modeling and analysis methodology for modular logic controllers of machining systems using Petri net formalism [J]. IEEE Trans on Systems, Man and Cybernetics, Part C: Applications and Reviews, 2001, 31(2): 168-188.
- [20] Kuo C H. Resource allocation and performance evaluation of the reconfigurable manufacturing systems [C]. Proc of the IEEE Int Conf on Systems, Man and Cybernetics. Tucson, 2001: 2451-2456.
- [21] 盛伯浩, 罗振璧, 俞圣梅, 等. 快速重组制造系统动态建模和结构体系[J]. 中国机械工程, 1999, 10(9): 1055-1060.  
(Sheng Bo-hao, Luo Zhen-bi, Yu Sheng-mei, et al. Study on dynamic modeling and structure for rapidly reconfigurable manufacturing system [J]. China Mechanical Engineering, 1999, 10(9): 1055-1060.)
- [22] Zhou Zhe, Dong She-qin, Hong Xian-long, et al. A new approach based on LFF for optimization of

- dynamic hardware reconfigurations[C]. Proc of IEEE Int Symposium on Circuits and Systems. Hong Kong, 2005: 1210-1213.
- [23] Ferreira C, Silva M. Run-time reconfiguration support for FPGAs with embedded CPUs: The hardware layer [C]. Proc of the 19th IEEE Int Parallel and Distributed Processing Symposium. Denver, 2005: 165.
- [24] Kogekar S, Neema S, Koutsoukos X. Dynamic software reconfiguration in sensor networks[C]. Proc of Systems Communications in Sensor Networks. Montreal, 2005: 413-421.
- [25] Tan S K, Ge Y, Tan K S, et al. Dynamically loadable protocol stacks: A message parser-generator implementation[J]. IEEE Internet Computing, 2004, 8(2): 19-25.
- [26] Walsh G C, Hong Y. Scheduling of networked control systems[J]. Control Systems Magazine IEEE, 2001, 21(1): 57-65.
- [27] Park H S, Kim Y H, Kim D S, et al. A scheduling method for network-based control systems[J]. IEEE Trans on Control Systems Technology, 2002, 10(3): 318-330.
- [28] Chow C H, Gouda M G, Lam S S, et al. On constructing multi-phase communication protocols[C]. Proc of the IFIP WG 6.1 4th Int Workshop. Skytop Lodge, 1985: 57-68.
- [29] Chiola G, Donatelli S, Solda G. Construction and validation of a petri net model of a layered protocol architecture[C]. Proc of 4th IEEE Region 10th Int Conf. Bombay, 1989: 226-233.
- [30] Suzuki T, Shatz S M, Murata T. A protocol modeling and verification approach based on a specification language and petri nets[J]. IEEE Trans on Software Engineering, 1990, 16(5): 523-536.
- [31] Murata T. Petri nets properties, analysis and applications[J]. Proc of the IEEE, 1989, 77(4): 541-580.
- [32] Bao G F, Liu F, Hong L. Modeling and verification of TCP congestion control based on colored petri nets [C]. The 2003 IEEE Int Conf on Systems, Man and Cybernetics. Washington, 2003: 1045-1050.
- [33] Conte A, Anquetil L P. Design for application protocol stack framework [C]. IEEE Int Conf on Communications. New Orleans, 2000: 565-570.
- [34] Stefan B, Ken M, Brian W. Application-compliant networking on embedded systems [C]. Proc of 5th IEEE Int Workshop on Networked Appliances. Manchester, 2002: 53-58.
- [35] Moon J T, Kim J S, Kim J B, et al. A hardware implementation of distributed network protocol [J]. Computer Standards and Interfaces, 2005, 27(3): 221-232.
- [36] 蔡衍文, 陈天洲, 吴朝晖. 面向通信设备的网络协议构件化[J]. 计算机应用研究, 2004, 21(12): 253-256. (Cai Yian-wen, Chen Tian-zhou, Wu Zhao-hui. Component-based network protocols on communication device[J]. Application Research of Computers, 2004, 21(12): 253-256.)
- [37] Deter S. Fieldbus device description using tag-based trees[J]. Proc of the 6th Africon Conf in Africa. George, 2002: 263-268.
- [38] Casado R, Bermudez A, Quiles F J, et al. Influence of network size and load on the performance of reconfiguration protocols[C]. IEEE Int Symposium on Network Computing and Applications. Cambridge, 2001: 46-57.
- [39] Casado R, Bermudez A, Duato J, et al. Performance evaluation of dynamic reconfiguration in high-speed local area networks[C]. Proc of 6th Int Symposium on High-Performance Computer Architecture. Toulouse, 2000: 85-96.
- [40] Balasubramanian S, Aksoy D. Automatic registration for dynamic and reconfigurable sensor networks [C]. The 13th IEEE Workshop on Local and Metropolitan Area Networks. Miu Valley, 2004: 169-174.
- [41] Lim A. Distributed services for information dissemination in self-organizing sensor networks[J]. J of the Franklin Institute, 2001, 338(6): 707-727.
- [42] Natchev N, Avresky D. Validation of NetRec — A dynamic reconfiguration algorithm for irregular topologies in presence of multiple failures[C]. Proc of the 18th Int Parallel and Distributed Processing Symposium. Santa Fe, 2004: 209-217.
- [43] An L M, Pung H C, Zhou L F. Design and implementation of a dynamic protocol framework[J]. Computer Communications, 2006, 29(9): 1309-1315.
- [44] Thorsten S, Christian M S. Design, implementation and validation of a generic and reconfigurable protocol stack framework for mobile terminals[C]. Proc of the 24th Int Conf Hachioji, 2004: 362-367.
- [45] Richard M, Richard P, Cottet F, et al. Allocating and scheduling tasks in multiple fieldbus real-time systems [C]. Proc of the 9th Emerging Technologies and Factory Automation. Lisbon, 2003: 137-144.