可重配置系统使用大型FPGA计算域

作者: Igor A. Kalyaev
南联邦大学
Kalyaev 科研所多处理器计算机系统部主任
Kaliaev@mvs.sfedu.ru

Ilya I. Levin
南联邦大学
Kalyaev 科研所多处理器计算机系统部主任
Levin@superevm.ru

Evgeniy A. Semernikov
俄罗斯科学院南部科研中心实验室负责人
semernikov@mvs.tsure.ru

设计人员将赛灵思多款 FPGA 互联，可以开发出新一类超级计算机，并通过定制用于多种应用领域的工作。

在过去的20年里，许多超级计算机架构都同时采用了微处理器和 FPGA，由微处理器发挥系统大脑的作用，FPGA则通常负责从 CPU 处卸载部分计算任务。事实上，对一代又一代超级计算机而言，这种将 FPGA 与独立的微处理器相搭配的方式被证明是一种成功的组合方案。

但近期，我们在南联邦大学Kalyaev科研所多处理器计算机系统部（位于俄罗斯Taganrog）工作的科研小组，主要以赛灵思 FPGA 为基础，设计出了一种可重配置的计算机系统 (RCS)。通过将多个FPGA互联，我们就能够开发出专用的计算结构，非常适合用于解决大量应用领域中存在的问题。

基于 FPGA 的 RCS 有几项值得注意的设计事项与优势。其核心部分是我们连接在一起以构成单个计算系统的数个FPGA。在我们的可重配置系统中，我们使用了正交通信系统，将 FPGA 布置在矩形格点上。相邻的 FPGA 采用直接链路相互连接。

另外，我们采用计算机辅助设计工具和结构化编程工具，在 FPGA 计算域内创建了并行流水线计算机构。这种并行流水线计算结构架构与任务架构相似，而这种相似性可确保 RCS 的最高性能。

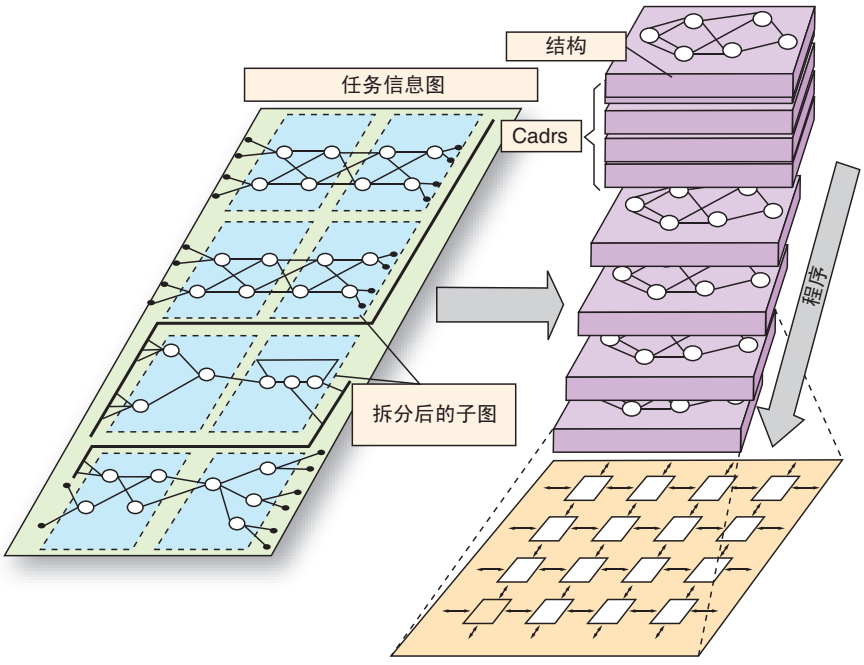

图 1　可重配置计算机系统 (RCS) 的计算结构流程图

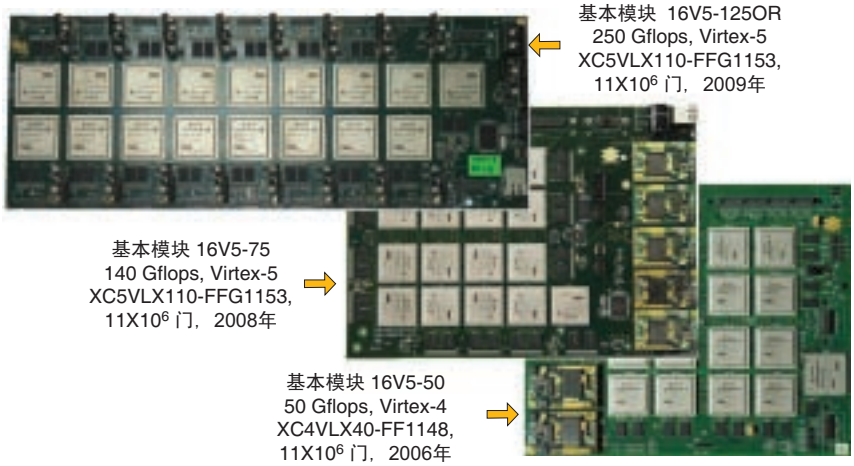

图2　围绕Virtex FPGA构建的几种基本RCS模块

如果由于硬件的限制，不能映射我们力图完成的任务或尝试解决的问题的整个信息图，那么系统将自动把信息图拆分为一系列子图。该系统的计算域将每个子图视为独立的计算结构，并将子图按顺序组织起来并逐个处理。图 1 显示了该 RCS 的计算结构流程图及数据输入和输出的程序。您可以看到，每个子图的大小都与一个 FPGA 域的大小相称。我们针对结构化编程开发的工具则能够自动将问题信息图拆分为数量较少的较大子图，并将其映射给硬件进行有效处理，从而显著提升 RCS 的性能。

详细地说，就是系统将给定工作或问题的信息图解读成为一系列同构的基本子图，子图之间既可彼此独立，也可相互依存（即具有信息相关性）。该系统将每个信息子图转换为某种特殊的计算结构，称为“cadr”。基本上，一个 cadr 就是一个硬连接的任务子图，有一个操作数流 (operands flow) 从其间经过。每一组操作数（或结果）与序列中具体子图的输入（或输出）节点相对应。由专门的并行程序（我们也称“过程”）负责执行 cadr 序列。整个 RCS 只需要一个并行程序。

我们采用模块化架构设计我们用于高性能计算的 RCS。RCS 由基本模块构成，基本模块内含计算域片段，并与其他模块和辅助子系统相连。图 2 展示了南联邦大学多核计算机系统部 Kayaev 科研所（SRI MCS SFU）设计的几种基本模块。

基本模块中的 FPGA 通过 LVDS 通道进行通信， 运行频率为1.2GHz。在 RCS 中的所有模块均通过这些通道通信。每个基本模块还包含有离散的分布式存储单元和用于执行资源管理的基本模块控制器，其功能包括将并行应用的片段加载到分布式存储控制器中以及处理数据输入和输出。

我们采用模块化架构设计我们用于高性能计算的 RCS。RCS 由基本模块构成，基本模块内含计算域片段，并与其他模块和辅助子系统相连。

我们的 16V5-75 是我们 RCS 系列的主模块。RCS 系统的性能水平可从 50 giga flop （使用 16 个赛灵思 Virtex®-5 FPGA）到 6 teraflop （使用 1,280 个Virtex-5 FPGA）。我们的高端型号采用五个 ST-1R 机架，每个机架安装了四个 RCS-0.2-CB 的 6U 模块，每个模块由四个 16V5-75 基本模块组成。每个基本模块的峰值性能可达 200 gigaflop （对单精度数据而言）。故 ST-1R 机架的计算域包含了 256 个由高速 LVDS（图 3）连接的 Virtex-5 FPGA （XC5VLX110）。

同时我们的 Orion RCS（图 4）具有 20 tera flops 的性能。它由布置在 19 英寸机架上的 1,536 个 Virtex-5 FPGA 构成。该系统的主要组件是性能为 250 giga flops 的 16V5-125OR 基本模块。基本模块通过高速 LVDS 连接到单个计算资源。四个 16V5-125OR 基本模块可连接成一个 1U 模块。与 16V5-75 基本模块不同， 16V5-75 基本模块内的 16 个 FPGA 中，只有 4 个可以与其它基本模块连接，而 16V5-125OR 基本模块的所有 16 个 FPGA 都可通过 LVDS 扩展计算域。这样就能够为 Orion 模块中的基本模块间提供较高的数据交换速率。

表 1 显示了 RCS-0.2-CB 和 Orion 模块在解决数字信号处理、线性代数和数学物理等不同任务时的实际性能。

任务	特定性能 (Gflops/dm3)	执行任务时的实际性能 (Gflops)		
		DSP	线性代数	数学物理
(RCS-0.2-CB)	16.7	647.7	423.2	535.2
ORION	64.9	809.6	528.7	669.0

表 1　RCS-0.2-CB 模块和 Orion 计算块的性能

RCS 的编程

RCS 的编程与传统的基于集群的超级计算机架构编程大相径庭。传统的超级计算机架构采用固定的硬件电路，一般只能由软件工程师在软件层编程。而基于 FPGA 的 RCS 则同时拥有可编程的硬件和软件。因此，我们可把 RCS 的编程工作分为两大类：结构化编程和过程化编程。结构化编程就是把信息图映射到 RCS 结构的计算域（FPGA 编程）。而过程化编程则指的是传统的编程工作，由用户在系统中描述计算过程的组织架构。一般来说，当提及 FPGA 编程时，软件工程师常常有所畏惧，因为要想胜任此项工作必须具备特定的硬件设计技能。

基于此，我们力图通过开发一个专用软件套件来降低 RCS 编程的难度。该套件可以让用户在不具备任何 FPGA 和硬件设计的专业知识的情况下即可完成 RCS 系统的编程。软件工程师在采用该套件后，解决问题的速度要比通过线路设计、或采用传统方法和工具来解决相同问题的工程师快 2 到 3 倍。

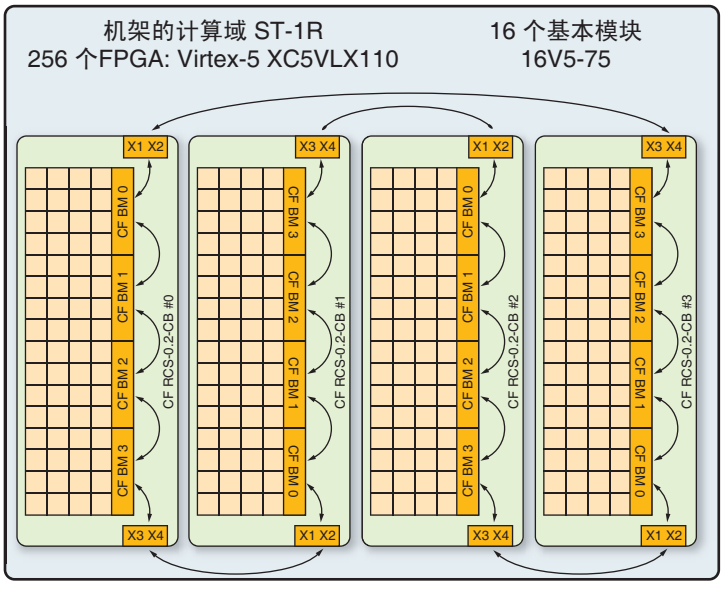

图 3　包含 256 个由高速 LVDS 连接的 Virtex-5 FPGA 的 ST-1R 机架的计算域

我们力图通过开发一个专用软件套件来降低 RCS 编程的难度。该套件可以让用户在不具备任何 FPGA 和硬件设计的专业知识的情况下即可完成 RCS 系统的编程。

该套件系从用于线路设计的 Fire!Constructor 开发环境发展而来，采用名为 COLAMO 的高级编程语言和 Argus 集成开发环境。用户可先在 COLAMO 中完成程序的开发工作，然后内置的编译器/转译器/合成器利用计算单元和接口的 IP 库可自动为结构组件和过程组件创建代码。

由基本模块描述、单元描述和整个 RCS描述（RCS 通行证）组成的库可帮助将并行应用从各种架构平台上导入 RCS。 COLAMO 转译器可将系统描述拆分为四个数据类别：控制、过程、流和结构。

控制数据被转译成 Pascal 语言，在主控制器中执行。根据 RCS 硬件平台的不同，主控制器或位于基本模块中，或位于计算块中。控制数据的作用是根据手头的计算任务配置 FPGA，对分布式存储器进行编程，以及映射计算块和基本模块之间的数据交换。

同时，过程组件和流组件被转译成 Argus 汇编语言。它们由分布式存储器控制器负责执行。分布式存储器控制器的作用是规定 cadr 的执行，以及在 cadr 计算结构中创建并行数据流。

该套件中最值得一提的特性是名为 Fire!Constructor 的综合工具，帮助用户对 FPGA进行编程。通过使用IP库，Fire!Constructor 可为 RCS 中的每一个 FPGA 生成一个 VHDL 描述（*.vhd 文件）。赛灵思 ISE 套件使用这些文件可为所有的 FPGA 创建配置文件(*.bit)。

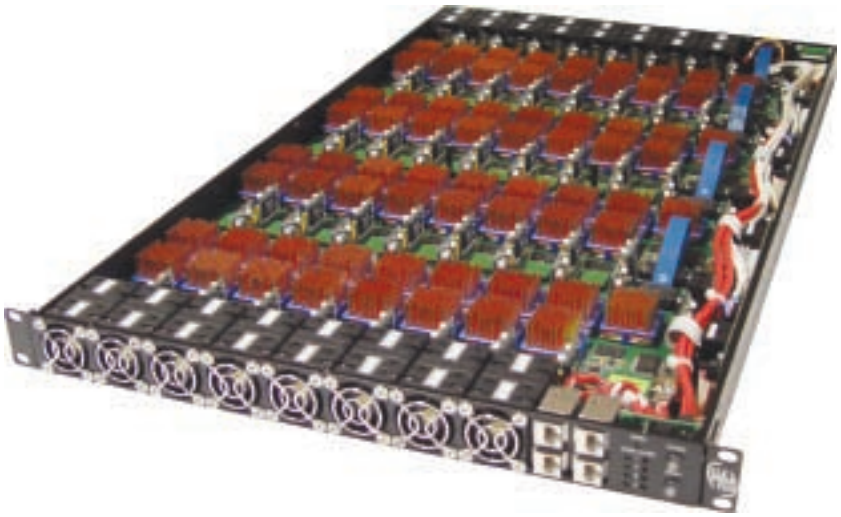

图 4　20-Tflops 的 Orion 计算块

有效解决问题

在“结构-过程”计算方法的帮助下，我们能够在 RCS 上有效地解决各种问题。最适宜解决的是那种算法曲线参数为常数，输入的数据变化的问题。举例来说，药物设计、数字信号处理、数学仿真等类似问题特别适用于 RCS 解决方案。

一家名为超级计算机及神经计算机科研中心（俄罗斯 Taganrog）的分立公司负责交付这些可重配置系统，每套 RCS 都按照预订的规格制造。该公司每月可制造大约 100 套。我们的 RCS 的性能，根据采用 Virtex-5 和 Virtex-6 FPGA 的情况，范围可从100 Gflops 到 10 Tflops 不等。

总而言之，采用 FPGA 的可重配置计算机系统在解决各类问题方面具有得天独厚的优势。采用 RCS 后，用户可定制计算，迅速解决问题。如欲了解有关 RCS 的更多信息，敬请访问 *http://paralle.ru/index_eng.html*。