

动态可重配置的星上嵌入式实时计算系统*

刘 勇^{1,2}, 李华旺¹, 尹增山¹, 杨根庆¹

(1. 中国科学院 上海微系统与信息技术研究所, 上海 200050; 2. 中国科学院 研究生院, 北京 100039)

摘要: 卫星上由于特殊条件的限制, 计算机处理速度满足不了对信号处理的需要, 而且不能在有限的硬件规模和功耗的情况下灵活地实现各种计算处理功能。提出了一种基于嵌入式微处理器配合大规模现场可编程门阵列 (FPGA) 的动态可重配置结构的星上实时计算系统的体系结构设计, 可在一块 FPGA 资源上通过动态重配置实现不同的信号处理功能。实际应用证明, 处理速度和性能得到了大幅度提高。

关键词: 嵌入式; FPGA; 动态可重配置; 计算系统; 信号处理

中图法分类号: TP273 文献标识码: A 文章编号: 1001-3695(2006)01-0204-02

Dynamically Reconfigurable Embedded Real-time Computing System on Satellite

LIU Yong^{1,2}, LI Hua-wang¹, YIN Zeng-shan¹, YANG Gen-qing¹

(1. Shanghai Institute of Microsystem & Information Technology, Chinese Academy of Sciences Shanghai 200050, China; 2. Graduate School, Chinese Academy of Sciences, Beijing 100039, China)

Abstract: At present, the speed of the micro-processor on the satellite cannot satisfy the need for the signal processing ability, and the high degree of flexibility can't be obtained based on the limited hardware resource and power consuming. An architecture dedicated to real-time signal computing system on satellite with dynamically reconfigurable architecture based on the embedded micro-processor and large scale Field Programmable Gate Arrays (FPGA) is presented. Based on the architecture, one piece of FPGA can be reconfigured dynamically to execute various signal processing algorithms. And the real application shows that the performance and computing speed can be improved in large range.

Key words: Embedded; FPGA; Dynamically Reconfigurable; Computing System; Signal Process

近年来, 由于卫星通信的信息业务变得越来越复杂和多样化, 对卫星上的信号处理功能也就提出了越来越高的要求。在地面上的大数据量的处理主要采用的是两种技术手段, 即并行处理和专门的处理电路(如 ASIC)^[1,2]。前者提供了很大的灵活性, 但是增大了体积, 而且功耗也会相应增加; 而后者可以大大提高系统的处理速度, 但是一旦应用, 就不能升级或者改变应用功能^[6,7]。这两种技术途径都不能满足目前卫星上信号处理单元对体积、功耗的严格要求和对应用功能的灵活性的需要, 发展一种更新更适合的信号处理方法成为当前的迫切需要。

在过去的 10 年中, 可配置计算 (Reconfigurable Computing) 在很多方面得到了广泛的应用, 而在可配置计算中应用最普遍的则是基于现场可编程门阵列 (Field Programmable Gate Arrays, FPGA) 的。FPGA 是由大量的可编程逻辑单元组成的, 这些硬件资源可以根据不同的应用设计成不同的计算系统, 并且可以实现数据级 (Data-level) 的并行处理来提高系统性能^[3]。

1 基于 FPGA 的可配置计算

基于 SRAM 的 FPGA 利用 SRAM 来保持配置数据, 这些配置数据规定了该器件的互连性和逻辑功能。由于 SRAM 是易

失的, 当器件启动或重新启动时必须对 FPGA 进行配置, 把保存在 FPGA 外部的配置数据重新装入 FPGA。一般是利用 FPGA 外部的专用 ROM 来保存配置数据, 系统启动时由 FPGA 的配置逻辑从 ROM 中读取配置数据来对 FPGA 进行配置。

由于 FPGA 的规模有限, 可用资源不能满足大型信号处理算法的需要, 一般是将算法分成几部分, 分别配置在多块 FPGA 上面, 由这些 FPGA 来共同完成整体的信号处理^[4]。一种简单的一维多 FPGA (Multi-FPGA) 结构^[5], 如图 1 所示。

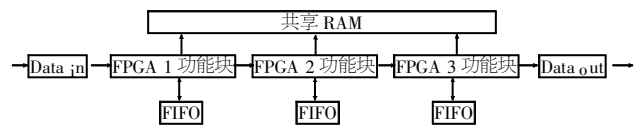


图 1 一维多 FPGA 信号处理结构图

以上系统由多块 FPGA 组成信号处理链, 信号输入和结果输出分别通过 Data in 和 Data out 单元 (采用先进先出 FIFO 芯片), 信号处理过程中的中间变量存储在 FIFO 中, 各个 FPGA 模块可以通过共享 RAM 交换数据。此系统的特点是配置简单, 但是增加多块 FPGA 会增大系统体积, 不利集成, 而且功耗上也会有不小的开销。

2 动态可重配置结构

FPGA 的配置除了采用专用 ROM 外, 还可以把配置数据保存在外部 ROM 或者 Flash 中, 由微处理器读取这些配置数

据, 然后通过 FPGA 的配置总线对 FPGA 进行配置。这样, 简单地载入新的配置数据就可以改变 FPGA 的工作方式。这种基于 FPGA 的动态可重配置结构, 可以将一个比较大的信号处理系统分成几个独立的功能模块, 通过动态重配置, 在一块 FPGA 上完成整体的信号处理功能, 其结构如图 2 所示。

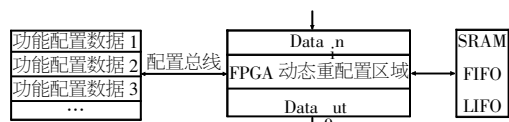


图 2 基于动态可重配置结构的信号处理结构图

如图 2 所示, 基于可重配置技术配置 FPGA 的步骤是: 在 PC 上完成 FPGA 的不同逻辑功能的设计, 仿真通过后生成各个独立的功能配置数据, 一般为二进制位流文件; 将这些配置数据下载到 ROM/Flash 中去, 再由微处理器根据信号处理过程中所需的不同功能模块去读取 Flash 中相应的配置数据, 并按照 FPGA 的 JTAG 配置标准定义的时序要求对 FPGA 进行再配置^[10]。

3 嵌入式实时计算系统

在动态可重配置系统中, 主要有两种结构模型, 即 SoC 模型^[11]和协处理器模型^[12]。前者是 FPGA 中嵌入微处理器内核来控制重配置和数据交换, 优点是大量减少了数据 I/O 和通信时间, 但结构较复杂; 后者是由 FPGA 担任微处理器外部的独立的协处理器, 完成所有的计算功能, 优点是结构简单, 但计算时间较长, 且对 FPGA 的逻辑单元的数量有要求。

本文提出了一种微处理器加可配置逻辑单元的混合系统结构模型 (Hybrid System Architecture Model)^[8]。在此模型基础上, 设计了一种动态可重配置的星上嵌入式实时计算系统。其结构如图 3 所示。

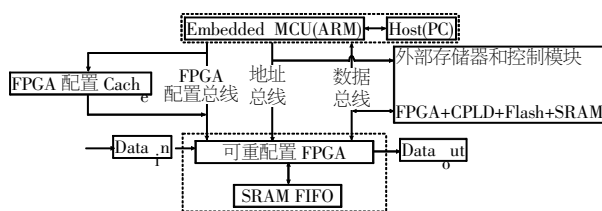


图 3 动态可重配置的嵌入式实时计算系统框图

此系统中, 嵌入式微处理器采用 32 位 RISC 微处理器 ARM, 可配置逻辑单元 (Configurable Logic Unit) 采用 XilinxTM公司的 FPGA^[13]。为了加快配置速度和减少数据传输的消耗, 还使用了 FPGA 配置 Cache。

下面的例子是在此系统上实现扩频通信中一种功率控制算法。在直接序列扩频码分多址 (DS-SS) 通信系统中, 由于分配给各用户的扩频序列不可能完全正交, 因此总存在多址接入干扰和远/近效应。为了消除这些干扰, 必须时刻控制每个用户的发射功率, 为此笔者提出一种新的基于最小均方误差 (MMSE) 的自适应功率控制算法^[9], 由式 (1)、式 (2) 组成算法的迭代规则:

$$C_i(n) = (6 \sum_j p_j h_{ij} S_j S_j^T + \sigma^2 I)^{-1} S_i \quad (1)$$

$$p_i^{(n+1)} = \frac{p_i^*}{h_{ij}} \frac{6 \sum_j p_j^{(n)} h_{ij} C_j(n) T S_j^2 + \sigma^2 (C_i(n) T C_i(n))}{(C_i(n) T S_j)^2} \quad (2)$$

以上的自适应迭代算法中, p_i^*, h_{ij} 和矢量 C, S 由初始化条件给出, 每个用户 i 的发射功率由迭代运算收敛时的 $p_i(n+1)$ 控制, n 为迭代数。FPGA 首先被配置成式 (1) 的计算模块, 把 $p_i(n)$ 和计算结果 $C_i(n)$ 保存在 RAM 中, 由 ARM 判断计算结束后, 再把 FPGA 配置成式 (2) 的计算模块, 并从 RAM 中读取 $C_i(n)$ 和 $p_i(n)$ 来计算 $p_i(n+1)$, 并把结果 $p_i(n+1)$ 保存在 RAM 中, 然后 FPGA 再被配置成式 (1) 的计算模块, 如此迭代下去, 数据收敛时输出最终的计算结果。为了避免每次由 ARM 来判断计算结果是否收敛而降低整个信号处理系统速度的瓶颈, 迭代次数可由计算机先模拟得到, 当达到计算次数后, 结束计算, 输出计算结果。

4 功能分析

以上算法 (用户数为 20, 迭代 10 次) 在 XilinxTM Spartan FPGA 上实现。设式 (1) 的计算模块的每次计算时间为 $t_{1i} (i=1, \dots, 10)$, 配置时间为 r_1 ; 式 (2) 的计算模块的每次计算时间为 $t_{2i} (i=1, 2, \dots, 10)$, 配置时间为 r_2 , 那么完成计算所需的时间和为

$$T_{total} = 6 \sum_{i=1}^{10} (t_{1i} + t_{2i}) + (r_1 + r_2) + 9(r_1 + r_2) \cdot c \quad (3)$$

式 (3) 中, c 为读写 RAM 的时间消耗, $c (0 < c < 1)$ 为引入 FPGA 配置 Cache 后对 FPGA 配置时间的加速参数。

表 1 列出了三种计算平台, 即 PC、DSP 和本计算系统上所需的计算时间。

表 1 不同计算平台所需的计算时间

计算平台	PC	DSP	可配置计算系统
硬件条件和运行频率	P4 1.8GHz 512 MB DDR	TMS320 C6711 150MHz	Spartan- 33MHz
计算时间 (s)	6.5	2.3	0.025

根据 Amdahl 计算速度比公式:

$$S = \frac{f_1}{f_2} \cdot \frac{T_2}{T_1} \quad (4)$$

其中, f_1 和 f_2 分别为计算系统的工作频率, T_1 和 T_2 为计算所需的时间。

我们可以得到, 此计算系统相对于 PC 和 DSP 的计算速度比分别为 4.8 和 20.2。可以看出, 采用可动态重配置的 FPGA 进行信号处理计算, 虽然运行的时钟不是很高, 但计算速度可以得到数量级的提高。

5 结论

可配置计算相比传统的计算系统或 DSP, 能够以任意位宽进行计算, 并且可以并发式地并行处理多个计算, 可以最大限度地提高系统性能。本文基于可重配置结构提出了一种应用于卫星上的嵌入式实时信号处理系统的设计, 并通过一种实际算法评估了系统的信号处理性能。结果表明可配置计算具有相当大的速度优势。

当然, 这种动态可重配置还不能完全发挥运行时动态可重配置的计算优势, 因为当 FPGA 上的当前运算还没有结束前是不能对其进行再次配置的, 而且当两种算法有共同之处时, 还不能做到部分重配置来节省配置时间。部分重配置、动态重配置算法和微处理器与 FPGA 之间软硬件任务的自动划分将是下一步研究的重点。

(下转第 209 页)