

基于 TTA 技术的专用处理器设计

刘 俊, 谢 憬, 王 琴

(上海交通大学 微电子学院, 上海 200240)

摘 要: 描述了一种基于传输触发体系架构(TTA)的处理器, 对其指令集的设计和体系结构的优化进行了讨论, 并给出了 H. 264 帧内预测算法的 ASIP 处理器设计实例, 有效的克服了当前专用处理器和微处理器的局限性. 实验表明: 基于 TTA 体系结构的 H. 264 帧内预测 ASIP 处理器较之通用处理器运行周期快 4 倍以上.

关键词: 传输触发结构; ASIP; Move Framework; H. 264 帧内预测算法

中图分类号: TN4

文献标识码: A

文章编号: 1000-7180(2009)11-0161-04

The Design of ASIP Based on TTA Methodology

LIU Jun, XIE Jing, WANG Qin

(School of Microelectronics Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: This paper describes the design of an Application Specific Instruction Processor (ASIP) for the intra-predict of H. 264 utilizing the paradigm of Transport Triggered Architecture (TTA), and giving its specific design of instruction and exploration of architecture. The design of ASIP based on TTA methodology has overcome the limit of ASIC and MCU obviously. The simulation results show that TTA processor can achieve four times or much higher performance than general processor.

Key words: transport triggered architecture (TTA); ASIP; Move Framework; H. 264 intra-predict

1 引言

多媒体技术的发展与嵌入式多媒体设备的广泛应用, 要求微处理器具有高效性和灵活性, 以满足实时处理要求. ASIC 和微处理器(MCU)两种实现方式均显示出各自的不足. 基于传输触发体系结构(Transport Triggered Architecture, TTA)^[1] ASIP^[2] 设计方法, 就是针对目标应用程序集合, 开发出具有定制指令集和体系结构的处理器, 该处理器能够高性能运行目标应用程序集合. 这种设计具有可裁减性、简单性、自动化的特点, 有效地克服了当前 ASIC 和 MCU 设计的局限性. 荷兰代尔夫特理工大学提出了 TTA 结构并设计出基于 TTA 的可配置处理器和一套半自动化的设计工具 Move Framework^[3-5].

文中以多媒体技术中 H. 264 视频编码的帧内预测部分为应用对象, 运用 Move Framework 半自动工具实现其在 TTA 传输触发体系结构上算法的

映射. 研究 TTA 架构的主要目的在于以下两点: (1)发掘映射算法的复杂性和计算的并行性; (2)探索 Move Framework 的各种应用功能, 例如对处理器资源优化、性能分析.

2 TTA 体系结构

从指令集的特点分析, RISC 是 SISO(单指令单操作)类型的体系结构, VLIW 是 SIMO(单指令多操作)类型的体系结构, 而 TTA 则是 SIMT(单指令多传输)类型的体系结构. 传统的计算机体系是操作数触发架构(Operation Triggered Architectures), 即数据的传递在执行中完成. TTA 体系结构核心思想是利用数据传输触发具体操作, 即功能单元操作均以数据传输作为触发激励, 也就是说任何数据源中的数据经过互连网络的传输后, 在写入功能单元数据寄存器的同时会触发一次完整的操作.

TTA 体系架构, 如图 1 所示. 包括逻辑功能单

元, 寄存器功能单元, 指令功能单元, 立即数功能单元, 系统总线以及功能单元和总线的连结点 socket. 数据通路由功能单元和寄存器文件通过 socket 互连在多条总线上. 寄存器单元看成特殊的功能单元来降低旁路路径的复杂度. 互连网络、寄存器文件与功能单元三者间的互连结构非常适合于开发三者流水线之间的并行性, 从而大幅度提高硬件利用率.

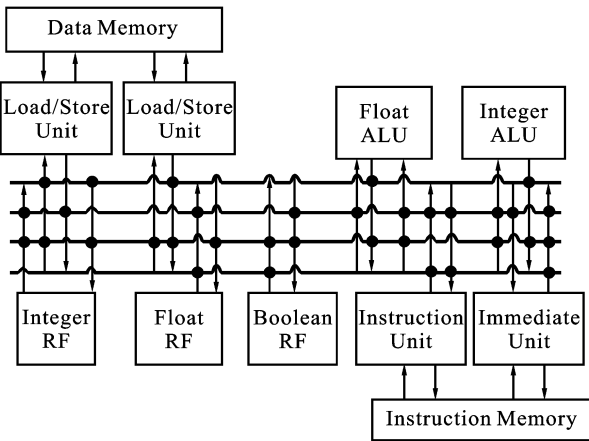


图 1 TTA 体系架构

3 TTA 的 ASIP 设计流程

Move Framework 是一套半自动生成 ASIP 的工具集, 研究探索了 TTA 体系架构的可裁剪性, 灵活性. 这套工具设计流程由三部分组成^[6]: 软件子系统、硬件子系统、优化系统, 如图 2 所示.

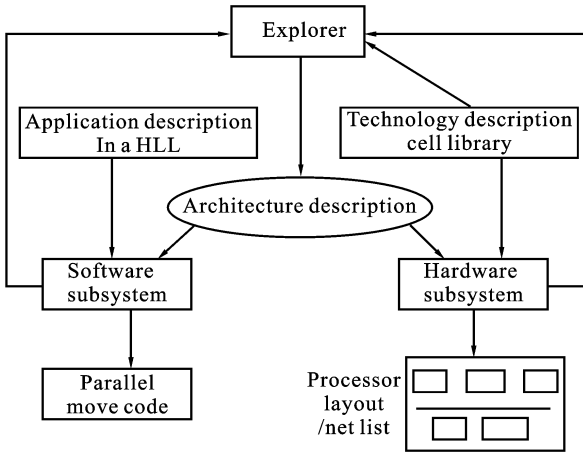


图 2 Move Framework

3.1 软件子系统

软件子系统主要负责生成指令级的并行代码. 它包括 Front-end、Back-end 和仿真器. Front-end 基于 GNU 编译器的 GCC 工具将 HLL(C 语言、汇编)代码编译成串行代码. Back-end 读取系统架构配置文件和串行代码生成并行代码. 仿真器提供性能分析,

例如执行周期数, 指令数以及硬件资源的利用率.

3.2 硬件子系统

Move Framework 的硬件子系统主要负责目标处理器的硬件 HDL 的实现. 包含两部分: Estimator 和 Processor Generator.

Estimator 通过读取系统体系架构文件估计芯片面积, 功耗和性能等主要的三项指标, 从而评估一个处理器核的性能.

Processor Generator 的作用是通过设计者所指定的系统体系架构文件生成所需要的硬件 HDL 代码. 生成 HDL 代码的同时也生成了测试文本. 任何工业界使用的综合工具和仿真工具都可以对这些代码进行综合和仿真.

3.3 优化系统

Move Framework 的系统优化目的是对于某些特定的应用, 自动的寻找一个满足特定要求的系统架构文件.

系统优化包括资源优化和互联优化. 资源优化的目的是通过某种优化算法, 在不影响系统性能的前提下删除功能单元, 输入输出接口, 寄存器单元, 寄存器单元的输入输出端口的数目和总线等等. Estimator 对优化后的系统体系架构文件进行性能评估. 在多个优化后的系统体系架构文件中, 根据优化质量函数 quality 选择使得 quality 值最大的系统体系架构文件.

$$quality = ((\frac{t_0}{t})^a \cdot (\frac{A_0}{A})^b)^{\frac{1}{a+b}}$$

其中, t_0 、 A_0 分别为原系统架构文件执行的时钟周期数和面积, t 、 A 为优化后的系统架构文件执行的时钟周期数和面积.

文中利用 simulated annealing^[2] 资源优化算法, 计算出初始系统架构的 quality 函数, 然后将第一次优化后的系统架构函数与初始系统架构函数比较, 接受条件为:

$$acceptance = r < e(\frac{\Delta q}{t})$$

其中, Δq 为新的 quality 函数与前一次的差值. $r \in [0, 1]$ 为随即数. t 为 simulated annealing 算法参数, 称为数据接受阈值.

互联优化是指连接优化将全相联的总线和 socket 连接转换成部分相联的方式. 其目的是使生成的处理器具有, 并使性能上损失较小.

4 ASIP 设计

4.1 H.264 帧内预测算法

H.264 帧内预测编码^[7]的基本原理是利用解

码重构的邻近块像素来实现对当前编码块的预测, 对预测块和实际块的残余差值进行变换, 量化, 熵编码, 以消除空域冗余. 本设计的目标程序采用 8×8 色度块的帧内预测算法 (Intra_chroma), 其中残余差值的计算采用均值法.

预测模式有 4 种: DC 预测, 水平预测, 垂直预测, PLANE 预测. 设一个 8×8 预测块的像素值用 $\text{pred}(i, j)$ 表示, 其中, i 为块的行坐标, j 为块的列坐标, $(i, j = 0 \sim 7)$. 则 $\text{pred}(i, -1)$ 是当前块的上方邻近像素值; $\text{pred}(-1, j)$ 是当前块的左边邻近像素值.

(1) 模式 0: DC 预测

① 若上方邻近像素和左边邻近像素都可用时, 各像素的预测值为:

$$\text{pred}(i, j) = \{ \sum_{i,j=0}^7 [P(-1, j) + P(i, -1)] \} / 16 \quad (1)$$

② 若只有上方邻近像素可用时, 各像素的预测值为:

$$\text{pred}(i, j) = \sum_{i=0}^7 P(i, -1) / 8 \quad (2)$$

③ 若只有左边邻近像素可用时, 各像素的预测值为:

$$\text{pred}(i, j) = \sum_{j=0}^7 P(-1, j) / 8 \quad (3)$$

④ 若上方邻近像素和左边邻近像素都不可用时, 所有像素的预测值为 128.

(2) 模式 1: 水平预测

$$\text{pred}(i, j) = P(-1, j) \quad (4)$$

(3) 模式 2: 垂直模式

$$\text{pred}(i, j) = P(i, -1) \quad (5)$$

(4) 模式 3: 平面预测

$$\text{pred}(i, j) = \text{Clipl}((a + b \times (i - 3) + c \times (j - 3) + 16) / 32) \quad (6)$$

其中, $\text{Clipl}(x)$ 表示将 x 位于 0 到 255 之间, 即

$$a = 16 \times (p[-1, 7] + p[7, -1]) \quad (7)$$

$$b = (17 \times H + 16) / 32 \quad (8)$$

$$c = (17 \times V + 16) / 32 \quad (9)$$

$$H = \sum_{x=0}^3 (x + 1) \times (p[4 + x, -1] - p[2 - x, -1]) \quad (10)$$

$$V = \sum_{y=0}^3 (y + 1) \times (p[-1, 4 + y] - p[-1, 2 - y]) \quad (11)$$

由于两个色度分量 (u, v) 各包含一个色度块,

而色度分量又只能有一个帧内预测模式, 因此色度分量的最佳帧内预测模式选择两个分量中残余差值最小的那个模式. 其中 $\text{pred}(i, j)$ 为预测块像素值, $\text{now_pred}(i, j)$ 为实际块像素值:

$$\text{SAD} = \min \sum_{i=0}^7 \sum_{j=0}^7 |\text{pred}(i, j) - \text{now_pred}(i, j)| \quad (12)$$

4.2 程序集特征操作数

Intra_chroma 目标程序经过 Front-end 编译器得到的顺序 move 代码, 同时生成目标代码性能指标. 它主要包括执行顺序 move 代码的周期数和操作数应用的次数. 如表 1, 这些性能指标作为配置系统体系架构文件的依据.

表 1 程序集特征操作数提取

Operation	count
add	3030
ld	1074
gt	928
sub	494
st	455
abs	448
shr	103
and	64
eq	64
shl	13
mul	8

4.3 初始系统体系架构

系统体系架构文件采用了 Python 语言对处理器组件特性进行定义, 包括: 总线、输入输出接口、功能单元种类、立即数、寄存器单元、指令单元. 对于设计者来说, 最重要的是提供一个能有满足, 甚至有冗余资源的初始系统架构文件, 以此来保证并行编译. 对于比较简单的设计, 在体系结构文件中, 主要关注以下内容: 功能单元的类型、数量及功能单元延迟; 总线数量; 输入输出接口的数量; 寄存器单元的大小和数量.

使用顺序 Move 代码和表 2 系统体系架构描述文件, 输入到 Back-end 编译器进行并行调度, 得到并行 Move 代码和性能统计信息.

系统体系架构文件配置了 11 条 32 位总线, 利用率最高的为 75%, 最低的为 8%. 依据硬件资源自带的 VHDL 库配置了 9 个功能单元. 第一组功能单元为 $\text{fu_always } 3, \{ \text{fu_o} \}, \text{fu_t}, \{ \text{fu_r} \}, \{ \text{ld, ldb, ldh, st, stb, sth} \}$. 3 表示数据传输延迟, $\text{o}, \text{t}, \text{r}$ 分别代表操作数, 触发数, 结果, $\text{ld, ldb, ldh, st, stb, sth}$

表示 load, store 数据的方式. 利用率最高为 24%, 最低为 9%. 同理, 无论总线、功能单元, 还是联结点 和寄存器单元, 最低利用率都不超过 10%, 说明这个 系统架构文件资源充足, 再增加相应的单元, 也不会 提升系统的性能.

表 2 初始系统体系架构参数提取

Parameter	Num	Max	Min
bus	11	75%	8%
ld ldb ldh st stb sth	2	24%	9%
add sub, eq gt shl shr and abs	6	50%	2%
mul	1	0%	0%
register	16	14%	0%
long immediate	2	16%	4%

4.4 系统优化

采用 simulated annealing 优化算法对初始的系 统体系架构文件进行功能单元、总线、寄存器数量等 的进行优化配置. 如图 3、图 4, 描述了资源优化下性 能、面积、功耗三者的关系.

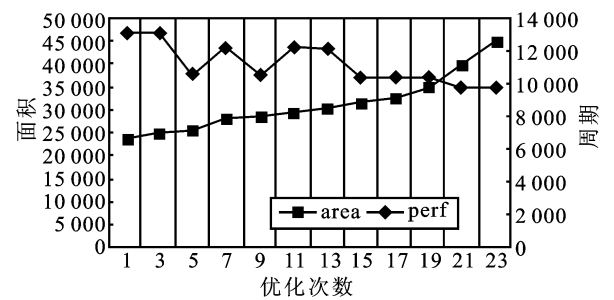


图 3 资源优化性能和面积关系

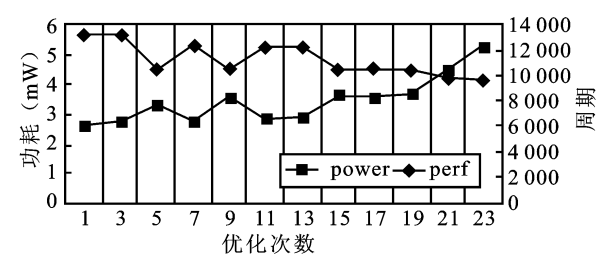


图 4 资源优化性能和功耗关系

从图 3 和图 4 中, 以性能周期数为主要评估标 准, 并权衡三者, 能够分析得到优化的系统架构文件 和相应的参数, 如表 3 所示.

从资源优化结构参数看, 各功能单元的数量有 所改变, 通常将利用率低的裁剪掉. 功能单元的减少 导致并行度的降低, 因此总线数量变少.

4.5 优化结果分析

应用 Move Framework 的 Estimator 对优化的 系统架构进行评估. 如表 4 所示, 经过资源优化, 在 维持性能不变的基础上, 芯片面积减小 50.5%, 功

耗降低 63.3%.

表 3 资源优化后系统架构参数提取

Parameter	Num
32bit _ bus	4
ld ldb ldh st stb sth	2
add sub, eq gt shl shr and abs	2
mul	1
register	8
long immediate	2

表 4 系统架构优化结果

性能参数	初始系统	优化系统
面积	80063	39631
运行周期	9741	9748
功耗(mW)	12.20	4.48

同时在 CCS2000 的 DSP 集成环境下, 使用 TM S320C5402 通用 DSP 和本设计给出的嵌入式处 理器分别运算 H.264 编码的 Intra-chroma 目标程序 进行比较, 本设计仅需要 9748 个周期完成, 比前者 (40058 个周期)快 4 倍以上.

5 结束语

文中给出了基于 TTA 结构的 ASIP 的详细设 计与实现, 研究探索了 Move Framework 半自动化 工具以及其采用 HLL 语言和 Python 语言进行系统 设计的环境, 并有效地设计出了针对多媒体 H.264 视频算法的 Intra _ chroma 部分的高性能专用处理 器. 整个设计思想简单、灵活, 对于实现某一具体领 域的应用具有一定的指导意义.

参考文献:

[1] Henk Corporaal. Transport triggered architectures: design anevaluation[D] . Netherlands: Delft University of Technol- ogy, 1995.

[2] 邵洋, 刘鸿瑾, 何星, 等. 专用指令集处理器系统级设计 方法 J]. 微电子学与计算机, 2007, 24(7): 102—104.

[3] The move framework user's manual[D] . Finland: Tam- pere University of Technology, 2004.

[4] Hoogerbrugger J. Code generation for transport triggered architectures[D] . Netherland: Delft University of Tech- nology, 1996.

[5] Corporaal H, Arend P A. MOVE32INT, a sea of gates re- alization of a high performance transport triggered architec- ture[J]. Microprocessing and Microprogramming, 1993 (38): 53—60.

(下转第 168 页)

始化配置,设置成绩处理计算机的 IP 地址和裁判员的编号,向成绩处理计算机请求授权,接收 PC 机发出的开始打分消息,显示分数输入界面.在裁判员输入分数并确认后,调用 WebService 接口存入后台数据库,进行业务运算后,显示运动员总分.主要程序模块包括:

(1) 初始化配置模块

首先要求用户输入帐户与密码后登陆.当用户正确输入之后,按下保存.调用程序 SaveConfig 存入本地 XML 文件中.

(2) 网络通信模块

本模块主要实现以下两个函数,① SvrNetComm 函数:该函数向 PC 机发送请求授权信息,并等待 PC 机发回 OK 消息.② SOCKET 函数:该监听程序接收到 PC 的消息后,经函数解析后,赋值给 MsgType 结构体,并将其存入本地 XML 文件.

(3) 评分模块

当 PDA 接收到开始评分消息后,由 ParseMsg 解析后,在界面上显示出评分界面.等待裁判员进行评分,评分完毕后按下确认键.调用 WebService 接口函数 SaveScoreDetail,然后界面切换到显示总分.

4 结束语

针对当前大部分打分类体育项目采用手工输入、速度慢、效率低、处理易出错的缺陷.设计了一种基于 WebService 三层架构的自动化无线打分系统,代替人工进行成绩处理.裁判员将分数通过手持无线打分终端(PDA)传输至成绩处理计算机后,存入后台数据库进行业务处理.不仅减少了错误,提高了效率,而且可以高效地与外围系统如运动会综合信息发布系统(INFO)、电视直播系统(TVG)等进行

数据交换.采用 WebService 技术,把主要的业务逻辑功能在成绩处理计算机中实现,保持了打分终端客户机的特点.该系统数据吞吐量大、速度快、实时性强、运行安全可靠.在 2007 年澳门举行的第二届亚洲室内运动会的部分比赛项目中运行良好,满足了比赛的要求,获得了裁判员和技术代表的一致好评,具有广阔的应用前景.

参考文献:

- [1] Yap C, Q I E, Sood K, et al. Issues with real-time streaming applications roaming in QoS-based secure IEEE 802. 11 WLANs [C] // Mobile Technology, Applications and Systems. Guangzhou, 2005: 1—7.
- [2] Robinson J, Randhawa W. Saturation throughput analysis of IEEE 802. 11e enhanced distributed coordination function [J]. Selected Areas in Communications, 2004, 22(5): 917—928.
- [3] 李海林,王美琴,高振明.基于 linux 的 802. 11b 无线局域网数据包捕获方法 [J]. 计算机应用研究, 2004, 21(12): 270—273.
- [4] Homas Kyte. Effective oracle design: design and build high-performance oracle applications [M]. New York: The McGraw-Hill Companies, 2003.
- [5] 赵慧琴,李秀兰. ORACLE 数据库应用系统的优化策略 [J]. 计算机工程与应用, 2003(23): 213—215.
- [6] 姚文琳,王存刚.基于 Oracle 的分布式数据库设计与技术 [J]. 计算机工程, 2006, 32(20): 89—91.

作者简介:

宋依青 男,(1960—),硕士研究生,副教授.研究方向为通信及自动控制.

李书旗 男,(1965—),工程师.研究方向为电子测量、自动化.

(上接第 164 页)

- [6] 岳虹,沈立,戴葵,等.基于 TTA 的嵌入式 ASIP 设计 [J]. 计算机研究与发展, 2006, 43(4): 752—758.
- [7] 毕厚杰.新一代视频压缩编码标准——H. 264/AVC [M]. 北京:人民邮电出版社, 2006.

作者简介:

刘俊女,(1983—),硕士.研究方向为数字集成电路设计与验证.