面向SDR应用的向量存储器的设计与优化*

陈海燕,刘 胜,刘 仲,陈书明
(国防科技大学 计算机学院,湖南 长沙 410073)

摘 要: 针对面向SDR应用的SIMD数字信号处理器高带宽数据访存需求,提出并实现了一种新型的向量存储结构。该向量存储器由16路向量存储块构成,每路采用两组多体低位地址交叉编址存储结构,减少了访存体冲突,充分利用多存储体带宽,以较小的功耗代价实现并行访问多个向量数据。在此基础上,还设计了一种向量访存重整理单元,使向量存储器可灵活支持多路SIMD结构向量处理单元的非对齐访问,实现了其对向量存储器的共享。测试结果表明,该向量存储器能有效减少或消除向量处理单元之间的数据混洗操作,加速相关应用算法。

关键词: 向量处理;单指令流多数据流;访存冲突;多存储体交叉;混洗;非对齐访问
中图分类号: TP368.1 **文献标志码:** A **文章编号:** 1001-2486(2012)03-0098-05

Design and optimization of the vector memory applying for SDR

CHEN Haiyan, LIU Sheng, LIU Zhong, CHEN Shuming
(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: To meet the high memory bandwidth of SDR-oriented SIMD DSPs, a novel Vector Memory(VM) architecture is proposed. The VM consists of 16-way Vector Memory Blocks(VBs), and each VB contains two groups of multi-bank memory structure with low-order interleaved addressing. This structure aims at reducing the memory access conflicts, making best use of the bandwidth of the multi-bank memory, and realizing the parallel vector data access at the cost of low power consumption. Besides, a vector rearrangement unit is designed and implemented in the VM to support the 16-way unaligned SIMD vector access and share the VM space. Experimental results show that the proposed VM architecture can efficiently reduce or eliminate the data shuffling operations and speed up SDR applications.

Key words: vector processing; SIMD; memory access conflict; multi-bank interleave; shuffle; unaligned access

随着无线通信技术的不断发展,通信协议的快速演进以及多种通信协议间的互操作性要求,传统的基于硬件的ASIC加速器的解决方案代价高昂而且复杂,越来越不能满足应用需求[1],软件无线电(SDR,Software Defined Radio)技术成为业界新的应用研究热点。由于DSP强大的计算能力、较低的功耗预算和灵活的可编程性,可更好支持未来软件升级、降低硬件成本,将其作为核心处理引擎的SDR技术平台成为支持无线通信多模应用的理想解决方案。

SDR应用面向音频、视频等流媒体数字信号处理,是典型的高计算密集型应用,具有很高的数据并行性。为了满足其应用需求,越来越多的高性能DSP采用单指令流多数据流(SIMD)结构,单片集成多宽度向量处理单元来开发应用的数据级并行,以提供强大的实时数据处理能力。多宽度SIMD结构处理器中的向量处理单元需要更高的访存带宽,面临着更为严重的"存储墙"问题[2]。

由于流媒体数据的时空局部性较差,通常的Cache存储层次结构不能很好匹配其访存特点;而且Cache存储层次结构中的访存缺失造成的流水线停顿将严重降低多宽度SIMD结构DSP的处理性能,无法满足实时处理要求。因此SIMD结构DSP一般配置了大容量片内存储器,尽量将运算数据保存在片内存储器中,以避免频繁的外存访问[3],降低访存延迟。如何为向量处理单元提供持续、充足的数据访存带宽,提高算法的访存效率和降低功耗,成为存储系统设计面临的重要问题。为了高效完成SIMD操作,数据必须按向量处理单元对齐,而有的存储系统仅提供按SIMD宽度地址对齐的访问,如Stanford的流处理器Imagine[4],每个运算处理单元(PE, Process Element)只能访问自己对应的存储体,对于较复

* **收稿日期:** 2011-06-02
基金项目: 国家"核高基"重大专项(2009ZX01034-001-001-006)
作者简介: 陈海燕(1967—),女,四川南充人,研究员,硕士,硕士生导师,E-mail:hychen608@163.com

杂的数据访存模式,PE 间的数据交互需要大量的混洗、打包、解包等额外操作[5];有的在存储器和 PE 间提供全交叉开关网络,如 AnySP[6],但随着 SIMD 宽度的增加,交叉开关的时延和功耗也大幅增长。

本文针对面向 SDR 应用的一款多宽度 SIMD 结构的高性能 DSP 设计了一种新型的向量存储器,该向量存储器由 16 路向量存储块构成,每路包含两组采用多体低位交叉编址的 4 个单口 SRAM 存储体,不仅有利于降低功耗,还能减少访存冲突次数,支持多路并行向量访存。在此基础上,还实现了一种向量访存重整理单元,不仅支持常规地址对齐的向量数据访存,还以较小的硬件代价实现了非对齐方式的向量访问,可大幅减少或完全消除 SDR 应用中一些常见算法的混洗操作,因而压缩了代码密度,提高了向量数据访存效率。

1　相关工作

1.1　YHFT-Matrix 结构框图

根据 SDR 多模应用需求,我们自主研发了一款高性能 16 路 SIMD 结构的 32 位数字信号处理器 YHFT-Matrix,设计主频 500MHz,8/16/32 位定点峰值性能为 32/16/8GMAC/s。YHFT-Matrix 采用标量处理单元(SPU)和向量处理单元(VPU)并行结构,由共同的取值派发部件完成标、向量指令派发,该 DSP 采用 10 发射超常指令字(VLIW)执行包结构,包含 5 条标量和 5 条向量指令。SPU 负责标量指令的执行,完成标量数据处理;VPU 负责向量指令的执行,完成向量数据运算。为同时满足指令、标量和向量数据并行访问,片上存储器分为标量存储器及向量存储器。标量存储器实现指令访问和标量数据访问,由标、向量处理单元共享的指令 Cache(ICache)和标量数据 Cache(DCache)组成;向量存储器主要实现 VPU 的向量数据访问,由 16 路向量存储块构成。VPU 由 16 个同构的 PE 阵列构成,每个 PE 集成了 MAC、ALU 和位处理 3 个子功能部件,分别对应执行 VLIW 指令包中的一条向量指令;其余两条向量指令为向量访存指令,由向量存储器(VM)完成;所有 PE 按 SIMD 的方式执行向量指令。片内存储器通过 DMA 与外设或外部存储器进行数据交换。该 DSP 的总体结构框图如图 1 所示。图中,VM 除了可支持两条向量访存指令,还可支持 SPU 及 DMA 的并行访问。

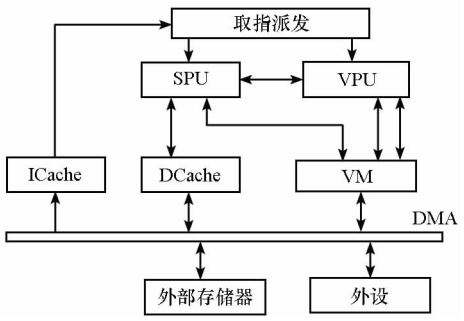

图 1　YHFT-Matrix 结构框图
Fig. 1　Framework of YHFT-Matrix DSP

1.2　向量存储器 VM 的总体结构

为了满足 VPU 内 16 个 PE 中 MAC 部件的数据带宽要求,VM 需要并行支持两条向量访存指令操作(LS0、LS1),同时还支持标量处理单元 SPU、DMA 对向量数据的访问,即 VM 需要支持 4 请求并行访问。VM 由专用的向量地址产生单元(VAGU)、与 PE 总数相同的向量存储块(VB0 ~ VB15)、向量访存控制器和向量输出选择器等主要功能模块组成,其组成结构及接口框图如图 2 所示。VAGU 实现向量访存指令的译码和地址计算;向量访存控制器实现 4 请求访存流水线的控制,即对仲裁、访存、输出选择和写回等各访存流水站进行控制;向量输出选择器将读出的向量数据输出至各请求端。

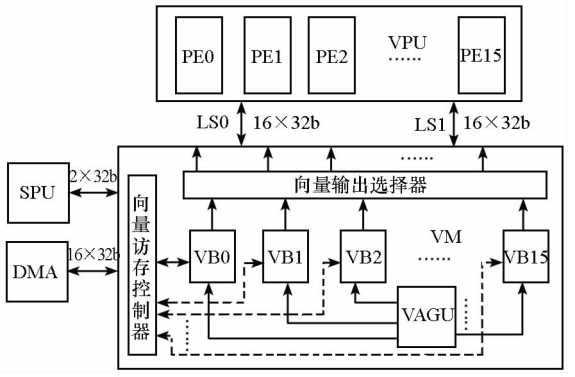

图 2　VM 的组成结构
Fig. 2　Components of VM

VM 总容量设计为 1MB,16 个 VB 完全同构,单个 VB 容量为 64KB,访存地址按 16 个 VB 低位地址交叉编址。由于嵌入式应用对 DSP 功耗的严格限制,每个 VB 均选择单端口 SRAM 作为存储单元。为支持 4 路并行访问,减少并行访存冲突,每个 VB 由 8 个单端口 SRAM(每个大小为 8KB)存储体构成,按两组 4 体低位交叉编址组织成上下两个存储空间,上半存储区由 bank0 ~ bank3 构成,下半存储区由 bank0' ~ bank3' 构成。理论上,访存地址不存在体冲突时,每个 VB 最多

可同时响应 8 个访存请求。假设某个 VB 的访存地址为 ADDR,如果(ADDR) mod(16 × 4)的值不同,就可实现无冲突的并行访存;如果将 DMA 访问和内核访问分布在上下不同的存储区,DMA 与内核则不存在访存冲突,而且如果待处理数据的 DMA 传输和向量数据访存并行,则可隐藏大部分 DMA 访问延时。

2 集中式向量地址产生单元

为了支持多种寻址模式和有符号/无符号的字节、半字、字等多粒度的向量数据访问,专门的地址产生单元的设计非常重要。对于 16 路 SIMD 结构的 DSP,我们设计了一个高带宽的向量地址产生单元(VAGU),实现两条向量访存指令的译码和地址计算功能。VAGU 根据支持的寻址模式,采用集中式方式进行向量访存地址的计算,再根据向量访存地址按访问粒度生成 16 个 VB 的访问地址和其他访存数据。

3 向量访存控制器的设计

访存控制器控制访存流水线的正确执行,决定着流水线的性能。对于片上大容量多路向量存储器,既要支持多请求并行访存流水,又要保证性能,向量访存控制器采用了分级的控制方法,设计总控模块和 16 个 VB 的访存控制器 VMC0 ~ VMC15。总控模块负责 16 个 VB 的访存同步,即访问请求仲裁开始的同步及向量数据写回的同步;VMC0 ~ VMC15 分别控制 16 个 VB 的访存流水线,实现对单个 VB 的访问控制。

整个向量访存流水线分为 6 站:地址计算站、仲裁站、访存译码站、访存站、输出选择站和写回站。其中,VAGU 在地址计算站完成地址计算和访存请求数据生成;仲裁站负责接收和仲裁 VPU、SPU 和 DMA 对 VM 的访存请求,并在发生体地址冲突时缓存访存请求、选择访存信号并输出到下一站;访存译码站负责将访存信号译码到 VB 内部各存储体端口,准备下一拍的访存操作。对于 VPU 的向量读访存请求,还需要输出选择站对各 VB 读出的数据进行选择和对齐整理,并在写回站输出至各请求端。

多请求并行访问的关键是要解决访存冲突问题,在 VM 中这是由向量访存控制器在仲裁站完成的。访存冲突主要发生在 VB 的 4 个请求的访问地址产生存储体冲突的情况。YHFT-Matrix 为 DMA 请求设置了 2 位可编程优先级,当 DMA 和其他访存请求发生冲突时,若 DMA 为高优先级时(优先级 > 0),则先处理 DMA 请求,并暂停其他内核请求;否则先处理内核请求,并暂停 DMA 请求。由于内核访问需要同步,同时为了简化仲裁逻辑,仲裁控制器中设计了两个状态机:内核向量访问(LS0、LS1、SPU)请求仲裁状态机和内核向量访问与 DMA 访问的仲裁状态机,其状态转换图分别见图 3、图 4。

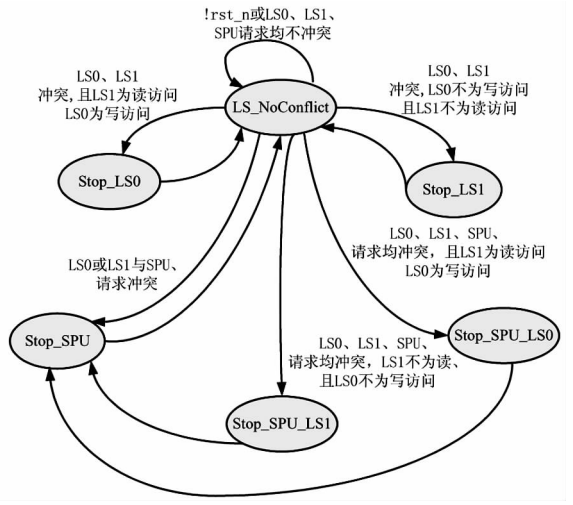

图 3 内核向量访存仲裁状态机

Fig. 3 State transition diagram for VB access arbitration of DSP core

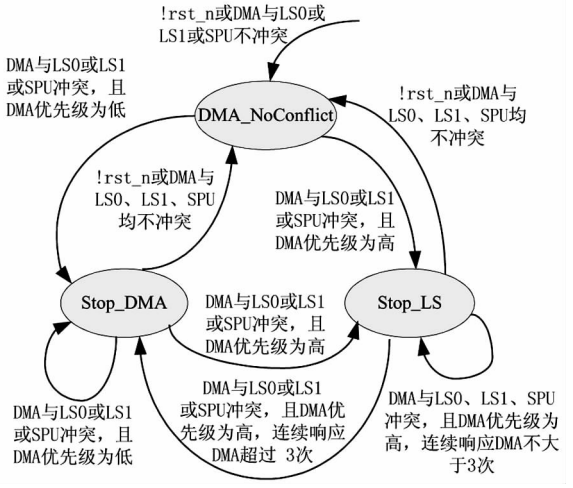

图 4 DMA 与内核向量访存仲裁状态机

Fig. 4 State transition diagram for VB access arbitration between DSP core and DMA

在仲裁 3 个内核访问时,YHFT-Matrix 设置了以下仲裁原则:当向量 LS0/LS1 与 SPU 请求发生访存体地址冲突时,将按照向量 LS0/LS1 为高,SPU 为低的优先级进行串行访问;当两个向量访问 LS0/LS1 发生访存冲突时,则读访问优先,若均为读或写访问,则 LS0 具有较高优先级。因此,图 3 所示的实现 VB 的 3 个内核访问请求仲裁的状态机共有 6 个状态:内核访问无冲突状态 LS_NoConflict、SPU 访问停顿状态 Stop_SPU、向量访

问 LS0 停顿状态 Stop_LS0、向量访问 LS1 停顿状态 Stop_LS1、SPU 和 LS0 同时停顿状态 Stop_SPU_LS0、SPU 和 LS1 同时停顿状态 Stop_SPU_LS1。如果仅存在内核访存冲突,则访问会延迟 1 ~ 2 拍完成。

图 4 所示状态机只判断 DMA 请求是否与 3 个内核访问冲突,用 3 个状态表示:即不冲突则进入 DMA_NoConflict 状态,DMA 请求与内核访问可并行;存在冲突时,根据 DMA 优先级,进入 DMA 请求停顿状态 Stop_DMA,或内核访问停顿状态 Stop_LS,此时为避免内核请求得不到响应,设计规定当 DMA 优先级为高时,可连续响应 DMA 请求的次数由其优先级确定,即按其最高优先级算,最多可连续响应 3 次;若 DMA 优先级被设置为 0,则内核访存始终具有高优先级。

为了减少 DMA 和 VPU 的访存冲突,DMA 和 VPU 处理的数据应该尽量放在一个 VB 中上下两个不同的存储区,这样可以使外部数据加载和内核计算并行,减少访存冲突,隐藏外部数据加载延时,提高向量数据处理的性能。

4　支持非对齐向量访问的优化设计

为了高效完成 SIMD 操作,一般 N 路的 SIMD 结构处理器要求向量访存数据按 N 进行边界对齐,即向量存储器的各个向量存储块 VB 一般与相应的运算处理单元 PE 一一对应。对于访存模式复杂的应用,各 PE 之间需要的数据交互则完全由混洗等指令完成,使操作数的组织过程复杂化,严重降低了实际的向量访存效率[5]。因此折中考虑性能和硬件实现代价,本文提出了一种优化的 VM 结构,支持向量数据的非对齐连续访问,使一个 PE 能有条件地访问到 VM 内的所有 VB,实现所有 PE 对 VM 存储空间的共享,大幅提高相关应用算法的访存效率。

在这种优化的 VM 结构中,设计了一种新的向量访存重整理单元,即在向量地址产生单元 VAGU 中增加了实现地址对齐整理功能的向量地址整理单元(VARU, Vector Address Reorder Unit),在向量输出选择器中增加了数据输出对齐整理功能的向量数据整理单元(VDRU, Vector Data Reorder Unit)。在地址计算站,VARU 根据 VAGU 计算出的向量地址,完成向量地址分解、复制和重整理,即对向量访存地址进行判断,对于地址未对齐的访存请求进行循环移位(按 PE0 ~ PE15 的顺序)和 VB 地址跨行加 1 操作,使得向量访存操作可以从任意非 N 路对齐的位置开始 N 个数据的连续访问,同时将该移位信息传递到下面的访存流水线。若本次操作为向量读操作,则在 VM 的输出选择站,需要 VDRU 根据流水线传递的移位信息对读出数据进行与 VARU 相反的循环移位操作后,再输出到对应的 PE 中。增加的 VARU 和 VDRU 这两个功能单元都主要实现循环移位功能,均由一个 16 项输入的循环移位器实现,硬件实现代价小,并未降低对应访存流水线的性能。

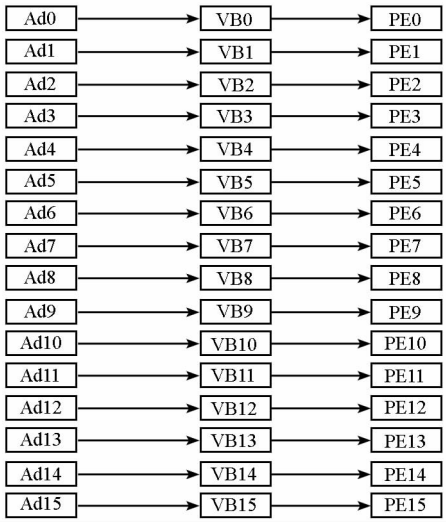

图 5　向量访存地址按 16 路对齐时 PE 对 VB 的读访问
Fig. 5　16-way aligned address load for PEs

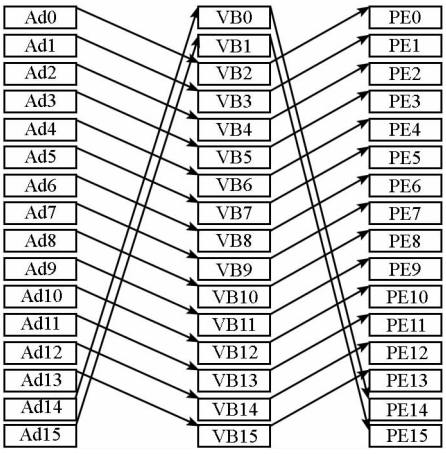

图 6　向量访存地址 = 2 时 PEs 对 VBs 的非对齐读访问
Fig. 6　Unaligned address load for PEs

图 5 和图 6 分别为向量读访存地址按 16 路边界对齐和向量读访存地址等于 2 时的向量读操作中 16 个 VB 与 16 个 PE 的关系流图,其中 Ad0 ~ Ad15 为 VAGU 产生的 16 个访问 VB 的地址。可见,通过向量访存重整理单元,VM 可支持地址对齐和非对齐方式的访问,每个 PE 可以访问 VM 的任何一个 VB。

下面以 SDR 中常用的有限滤波算法(FIR)为例,分析这种非对齐访问的访存性能。

基本的 FIR 算法如下：

$$y(n) = \sum_{i=0}^{M-1} h(i)x(n-i)$$

假设样本数 $n=1024$，抽头 $M=16$，数据均为定点 32 位，YHFT-Matrix 采用 16 路 SIMD 操作方式，计算出全部 $y(n)$，系数 $h(i)$ 需要 1 条向量 load 指令，样本数 $x(n-i)$ 加载需要 1024 条向量 load 指令，保存结果需要 1024 条向量 store 指令，即可完成算法的访存操作。如果 VM 只支持 SIMD 宽度对齐的向量数据访问，样本数 $x(n-i)$ 的加载只需要 1024/16 = 64 条向量 load 指令，比前者减少了 960 条，但会额外增加 15 种混洗模式共 960 条混洗指令，而每次混洗操作都需要使用向量 load 指令配置混洗模式，所以两者向量 load/store 指令条数相同，但前者无需混洗操作，后者不仅会增加 960 条混洗指令，而且多 SIMD 宽度 PE 间的混洗操作开销较大，其软件流水实现困难，无疑将大幅降低系统的计算效率。

5 性能分析与结论

设计使用 Synopsis 公司的 Design Compiler，基于某厂家 65nm 工艺库进行逻辑综合，结果表明，VM 的工作主频可达 500MHz 以上，因此 VM 最大可同时提供 512Gbps 的向量、32Gbps 的标量、256Gbps 的 DMA 访存带宽。

目前基于 4 个 YHFT-Matrix 的多核 DSP 芯片已投片成功。下面在该 DSP 芯片的测试环境下，基于其单核 YHFT-Matrix 的向量运算结构，使用 SDR 中音频、视频信号处理中常用的有限脉冲滤波算法（FIR）、无限脉冲滤波（IIR）、自相关（Autocor）和差的绝对值和（SAD）等典型算法，针对利用 VM 的非对齐访问和只利用 VM 的对齐访问功能两种情况进行编程测试，分析 YHFT-Matrix 完成向量运算所需的访存和混洗指令数，结果如图 7 所示。

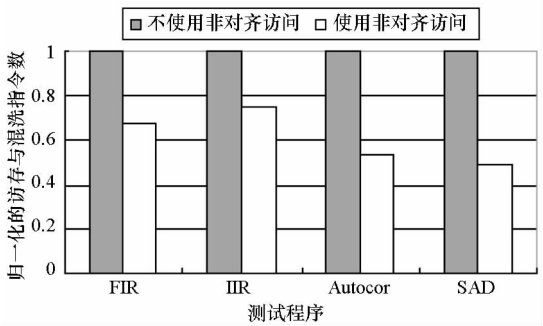

图 7 不使用 VM 对齐和使用 VM 对齐功能的性能对比

Fig. 7 The performance compare between the unaligned VM and aligned VM

其中 FIR、IIR 均为 1024 点、16 抽头的 32 位样本数据，Autocor 为 512 点、8 抽头的 16 位样本数据，SAD 为实时图为 256×256 像素、模板为 64×64 像素的 8 位数据。所有算法所需源操作数均为定点数，采用手工编写的经过优化的汇编代码实现。

从图 7 中可见，对于滤波、自相关一类算法，由于运算时需要大量向量访存地址跨步为 1 的连续向量数据，使用 VM 的非对齐访问功能后，上述 4 个测试程序的向量访存和混洗指令数之和将降低 31%、25%、47% 和 49%，具有较高的访存效率。

因此，基于多路 SIMD 结构所设计的 VM 不仅能支持多端口并行访存，有效降低访存冲突，还支持非对齐方式的向量数据访问，对于需要大量的向量访存地址跨步为 1 的向量访存的一类算法，实现了多路运算单元对向量存储器的共享，大幅减少或消除了相关算法的混洗操作，压缩了代码密度，从而提高了访存效率，加速了相关算法。

参考文献（References）

[1] Lin Y, Lee H, Woh M, et al. SODA: A High-Performance DSP architecture for software-defined radio[J]. IEEE Micro, 2007(1): 114 – 122.

[2] Cho J, Chang H, Sung W Y. An FPGA based SIMD processor with a vector memory unit[C]// ISCAS 2006: 525 – 528.

[3] Talla D, John L K, Burger D. Bottlenecks in multimedia processing with SIMD style extension and architectural enhancements[J]. IEEE Transactions. Computers, 2003, 42(8): 1015 – 1031.

[4] Khailany B, Dally W J, Chang A, et al. VLSI design and verification of the imagine processor[C]// Proceedings of the IEEE International Conference on Computer Design, 2002: 289 – 296.

[5] Chang H, Cho J, Sung W Y. Performance evaluation of an SIMD architecture with a multi-bank vector memory unit[C]// IEEE workshop, SIPS'2006: 71 – 76.

[6] Wob M, Seo S W, Mahlke S, et al. AnySP: anytime anywhere anyway signal processing[C]// ISCA'2009: 128 – 139.