

文章编号: 1001- 2486(2009) 05- 0006- 06

软件无线电数字信号处理器体系结构研究^{*}

刘衡竹, 莫方政, 张波涛, 赵恒, 刘冬培, 陈艇, 周理

(国防科技大学 计算机学院, 湖南长沙 410073)

摘要: 软件无线电因被认为是无线通信技术未来的发展趋势而受到广泛关注。目前数字信号处理器是软件无线电发展的瓶颈。通过分析、比较目前几种较为典型的软件无线电数字信号处理器结构, 归纳总结各种结构各自设计出发点和优缺点, 并对软件无线电数字信号处理器的发展趋势做了展望。

关键词: 软件无线电数字信号处理器; 可重构; 单指令多数据; 超长指令字

中图分类号: TP301 文献标识码: A

Software Defined Radio Digital Signal Processor Architecture Research

LIU Heng-zhu, MO Fang-zheng, ZHANG Bo-tao, ZHAO Heng, LIU Dong-pei, CHEN Ting, ZHOU Li

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Software defined radio (SDR) has won much interest for being considered to be in line with the trend of wireless communication development. Now the digital signal processor (DSP) is the bottleneck of software defined radio. The advantages and disadvantages of diverse architecture of software defined radio digital signal processor are summarized, and then the trends of software defined radio digital signal processor are discussed.

Key words: software defined radio digital signal processor; reconfigurable; SIMD; VLIW

支持多种通信标准是下一代无线通信终端发展的重要驱动力。软件无线电作为一种潜在的满足未来支持多种通信标准需求的解决方案而受到广泛关注, 预计到 2014 年软件无线电手持终端市场将达到 270 亿美元^[1]。无线通信是一种计算非常密集的应用, 软件无线电对计算性能的需求远超目前主流通用处理器和数字信号处理器的性能水平, 数字信号处理器是目前软件无线电发展的瓶颈。本文通过分析比较目前几种较为典型的软件无线电数字信号处理器结构, 归纳总结各种结构各自设计的出发点和优缺点, 最后对我们当前在软件无线电数字信号处理器结构研究上的进展作简单介绍。

1 SDR-DSP 及其面临的挑战

传统无线通信系统采用 ASIC 或 FPGA 方式硬件实现。ASIC 方式实现的缺点是缺乏灵活性和扩展性。FPGA 方式实现相对 ASIC 方式实现虽具有一定的灵活性和扩展性, 但其缺点是芯片硬件资源利用率低、面积功耗大, 难以应用到手持终端中。软件无线电改变传统无线通信系统中采用硬件实现的方式, 以可编程数字信号处理器为核心, 采用软件方式实现无线电功能。软件无线电的目标是在单一硬件平台上能通过升级软件适用于新型通信标准, 具有高度的灵活性和扩展性。软件无线电对数字信号处理器性能提出更高的要求, 为此针对软件无线电应用的数字信号处理器——软件无线电数字信号处理器(简称 SDR-DSP)应运而生。

目前全球尚在使用的第二代(2G)移动通信技术通信速率在 100Kbps 以下, 正在大规模部署的第三代无线通信技术(3G)通信速率提升到 10Mbps, 开始试验部署的第四代无线通信技术(4G)通信速率将达

^{*} 收稿日期: 2009- 07- 03

基金项目: 国家 863 计划资助项目(2007AA01Z287); CAST 基金资助项目(20080302); 国家自然科学基金资助项目(60970037); 教育部“高性能微处理器技术”创新团队资助项目(IRT0614)

作者简介: 刘衡竹(1963-), 男, 研究员, 博士生导师。

到 100Mbps。每一次无线通信技术的更新换代都会给通信速率带来 1~2 个数量级的提升, 通信速率提升的同时导致无线通信终端的数据处理能力需求大幅提高。针对无线通信算法的研究表明^[2-4], 如果以普通 RISC 处理器作为目标实现, 3G WCDMA 的计算负载达 60Gops, 而 4G 通信技术中仅核心算法的计算负载就超过 1000Gops, 远远高于目前主流通用处理器和主流数字信号处理器的性能水平, 计算性能是目前 SDR-DSP 面临的巨大挑战。

移动无线通信终端的可持续工作时间是一项重要设计指标。软件无线电方案要在移动终端上实施, 数字信号处理器除了满足超高的性能要求外, 还需具有极低的功耗。若以性能功耗比(Mops/mW, 百万次操作每秒每毫瓦) 衡量, 目前主流通用处理器大致处于 0.4Mops/mW 水平, 主流数字信号处理器大致处于 10Mops/mW 水平, 而普通 3G 无线通信的要求高于 100Mops/mW^[5], 4G 无线通信的要求甚至高于 1000Mops/mW^[6]。随着通信速率的提高, 无线通信系统所需求的计算性能越来越高, 而电池技术发展有限, 软件无线电对处理器所需求的性能功耗比与实际水平之间的差距有进一步变大的趋势, 功耗是 SDR-DSP 面临的另一重要挑战。

2 SDR-DSP 结构比较分析

针对 SDR-DSP 的性能和功耗需求, 业界提出了多种类型的微结构。执行效率和灵活性是处理器设计过程中的一对矛盾目标, 专用集成电路和通用处理器是这对设计目标的两个极端。专用集成电路强调芯片的性能和面积, 执行效率是首要设计因素, 代价是几乎不具备灵活性; 通用处理器强调芯片的通用程度, 灵活性是首要设计目标, 代价是执行效率较低。以处理器的执行效率和灵活性作为参考, 可将 SDR-DSP 结构划分为两大类: 一类是可重构式流式结构, 该类结构的特点是有较高的执行效率但是灵活性相对较低; 另一类是可编程处理器结构, 该类结构主要特点是具有较高的灵活性但是执行效率相对较低。

2.1 可重构流式结构

国外学者提出的 MorphoSys、ADRES、Montium^[7-9] 以及国内学者提出的 ReModCSP^[10] 等是目前典型的可重构流式结构。可重构流式结构通常包含多个功能单元, 组成某种阵列形式, 如图 1(a) 所示。可重构流式结构中功能单元可以是较小的简单功能单元, 也可以是较大的复杂功能单元, 功能单元之间以及功能单元和寄存器之间通过可配置的数据通路相连接, 如图 1(b) 所示。根据连接形式的不同, 原有的基本功能单元组合成不同的、更加复杂的功能单元。数据从通路的一端流入途经多个功能单元后从

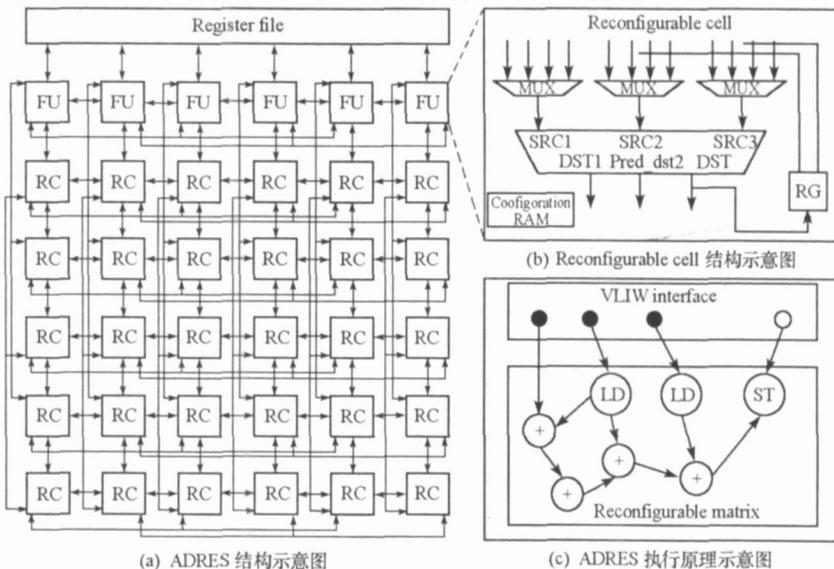


图 1 ADRES 结构图

Fig. 1 ADRES architecture diagram

另一端流出,完成对数据的处理,如图1(c)所示。可重构结构相对于ASIC有较高的灵活性,相对于可编程处理器又有较高的执行效率,是处理器设计过程中对灵活性和执行效率的折中。根据功能单元的粒度大小,可重构流式结构可划分为粗粒度可重构和细粒度可重构。可重构流式结构的重构粒度对芯片的面积、硬件利用率、功耗和可重构程度有直接影响,可重构流式结构面临的一个困难就是无论从消费者还是从市场设计者都很难得到一个可重构程度的确切答案^[11]。可重构流式处理器通过配置功能单元间的连接通路确定处理器的功能,因此程序流既包含数据流又包含处理器本身的配置流,导致编程模型很难同处理器结构本身解耦,这对编译器的智能程度提出了更高的要求。

2.2 可编程处理器结构

与可重构流式结构着力提高执行效率不同,可编程处理器结构主要通过提高并行度来提升处理器性能。并行可分成线程级并行、指令级并行和数据并行,根据提高并行对象的不同,可将可编程处理器结构划分为多线程结构、超长指令字(VLIW)结构和单指令多数据(SIMD)结构。

2.2.1 多线程结构

MF-ADRES和Sandbridge Sandblaster等是支持多线程的SDR-DSP^[12-13]。多线程结构能同时或间隔发射不同线程的指令,从而减少因为指令间的数据相关导致的流水线停顿,提高处理器功能单元的利用率。针对无线通信算法的研究表明^[2-3],无线通信系统中对数据的处理过程可看作是多个可流水的算法核。这些算法核可开发成多个线程执行,也可以开发成多个任务分配到不同的处理器核上执行,即多核结构也能有效开发无线通信算法中天然存在的线程级并行。无线通信系统中,算法核之间的通信是一种高吞吐率的数据通信,且多是单向通信,即多个算法核流水对数据进行处理。相对多核结构,多线程结构增加处理器结构的复杂程度和功耗。无线通信系统中各个算法核在编译时是已明确的,算法核之间的通信及调度情况也是可确定的,因此采用多核静态调度相对同时多线程的动态调度更符合通信系统的实际模型。目前提出的SDR-DSP倾向于采用多核结构。

2.2.2 VLIW结构

VLIW技术是利用静态调度手段将不相关的多条指令封装成一个执行包,程序执行时处理器将整个执行包的指令同时发射执行。由于静态调度,VLIW结构不需要额外硬件资源进行动态调度及相关检测,结构复杂性和功耗大大降低,因此VLIW结构是开发指令级并行的一种有效手段。无线通信算法中通常存在部分串行操作和宽度较小的SIMD操作,这些操作在大宽度的SIMD功能单元中执行的效率较低。由于在无线通信算法中数据并行占主导,单纯VLIW结构不能完全有效开发数据并行,因此目前提出的各种SDR-DSP多采用VLIW+SIMD或者VLIW+可重构流式结构的组合方式^[14]。

2.2.3 SIMD结构

国际的SODA、EVP、MuSIC^[5,11,15-16]和国内的SPVA^[17]是比较典型的SIMD结构。SIMD结构的核心是SIMD执行单元,如图2所示是SODA PE结构示意图。SIMD执行单元能同时对多个数据执行相同的操作,提高了程序的执行效率,是开发数据并行的有效手段。在普通应用中是否存在足够的并行是影响SIMD执行单元执行效率的关键因素。普通应用中数据并行方式多种多样,进行SIMD操作前往往需要多个操作来进行数据搬移和对齐,导致实际上SIMD执行单元的利用率并不高。

针对无线通信算法的研究表明^[2-4],无线通信中计算密集的数字信号处理算法存在大量的数据并行,并且数据并行方式以及并行的宽度都是在编译时可确定的,因此在SDR-DSP中采用SIMD技术不存在由于数据对齐而导致的执行效率下降的问题,也不会因为需要支持复杂的可变宽度SIMD操作而导致结构复杂化。比如下一代无线通信技术中普遍采用的OFDM技术,通常需要进行1024个点以上的FFT/IFFT运算,1024个点的FFT/IFFT运算的并行度达1024。SIMD技术被认为是解决软件无线电计算性能需求过高问题的有效方法。

3 SDR-DSP发展趋势

目前提出的各种SDR-DSP结构尚处于研究阶段,没有公认的标准。SDR-DSP面临性能和功耗的双重挑战,无论是可重构流式结构还是可编程处理器结构,未来的SDR-DSP必将更多地融合各种结构的

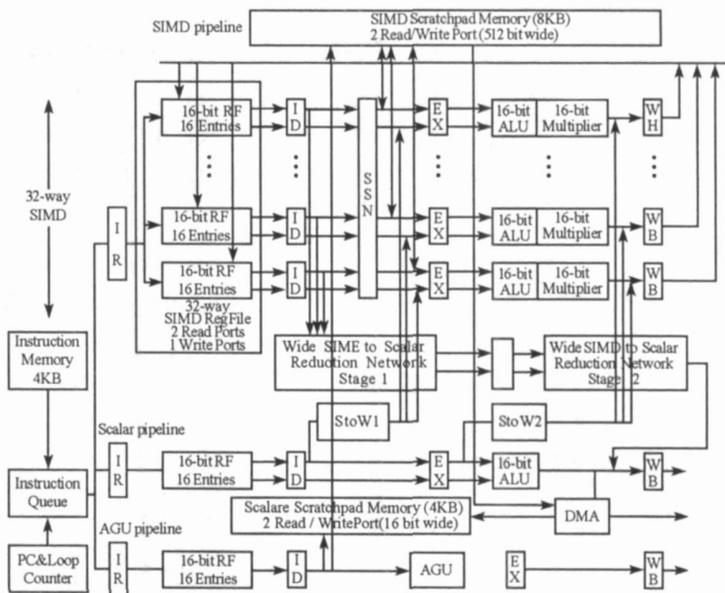


图 2 SODA PE 结构示意图^[5]

Fig. 2 SODA PE architecture diagram

优点。融合多核结构、VLIW 结构和 SIMD 结构, 增加协处理单元和新型片上互连网络(NoC), 可能是未来 SDR-DSP 结构的显著特征。

3.1 算法驱动的处理器的体系结构设计

以目前的技术水平实现一款能胜任理想软件无线电计算性能要求的数字信号处理器难度较大, 因此一般都是针对几个或几类无线通信标准, 并以此为预期目标进行结构设计。限于几个或几类无线通信标准, 使得更多地挖掘不同标准间存在的共同特征成为可能。抽取不同标准的共同特征对 SDR-DSP 的结构设计优化有重要意义。目前针对通信算法的分析研究多受计算负载的驱动, 即通过对通信算法进行分析得到相应算法的计算负载指标, 并以此为设计目标进行处理器设计。除计算负载外, 通信算法特征对设计 SDR-DSP 参数——指令集、功能单元结构、SIMD 宽度、片内存储大小、数据混洗模式等都有重要指导意义。针对无线通信中的信道编解码算法特征进行指令集扩展和算术执行单元优化设计的实验表明^[10, 18-19], 扩展指令集和优化设计的算术执行单元使得处理器在执行信道编解码算法时获得极大性能的提升, 同时将面积和功耗增加限制在很小的幅度。最新提出的 AnySP 结构根据对不同算法进行特征分析的结果, 设计可配置的 SIMD 单元以及数据交换网络, 在特定应用中具有极高的执行效率^[9]。

3.2 更宽的 SIMD 数据通路

半导体工艺的发展使得晶体管特征尺寸下降, 芯片单位面积上集成的晶体管数量增加, 芯片中集成更多的功能单元, 从而使增加 SIMD 宽度成为可能。无线通信算法中存在的大量数据并行使得非常宽的 SIMD 执行单元能够被有效利用。目前针对软件无线电的数字信号处理器 SIMD 执行单元宽度在 32 路以下, 上百甚至上千路的 SIMD 执行单元有可能在未来的 SDR-DSP 中出现^[20]。

3.3 增加可重构协处理单元

不同无线通信标准中往往存在部分相同或相近的数据处理算法, 这些算法往往存在显著的共同特征, 比如不同通信标准中所采用的滤波算法、信道编解码算法等。这些算法往往又是计算密集型的, 采用软件编程方式实现这些算法性能功耗比较低。采用针对某种算法进行优化的处理单元实现方式, 能在保持整体灵活性条件下大大减少整个处理器的计算负载^[11]。

3.4 无线应用领域专用的多核互连结构

出于低功耗和开发线程级并行的原因, SDR-DSP 多设计成多核结构。SIMD 执行单元提升处理器性

能同时提升数据的吞吐量,增加处理器核与存储之间、处理器核与处理器核之间的通信带宽需求。无线通信的实时性要求处理器核的数据交互具有延迟小、响应时间确定的特性。SDR-DSP对灵活性的需求要求处理器具有开放的、易扩展的架构,可以灵活增加处理器核数目和增加协处理单元。与此同时,工艺的进步使得芯片内互连线的功耗所占据的比重越来越大^[21],因此需要在互连结构设计上尽可能采用低功耗设计技术。在SDR-DSP中集成无线通信专用的多核互连结构成为新的趋势。

4 我们当前的工作

国防科技大学计算机学院微电子所SDR小组近年来跟踪国际SDR-DSP的发展,同时开展针对3G、4G无线通信算法特征的研究。在总结现有SDR-DSP体系结构的基础上,提出了一种新型的SDR-DSP结构“GAEA”,如图3所示。GERA是一种异构多核处理器结构,GERA通过NoC方式连接数字信号处理单元MPE、控制单元、加速部件单元和存储控制单元等多种异构核。目前SDR小组已经完成第一版本的设计和描述,得到综合面积和功耗等参数,并对其性能进行评估和改进。

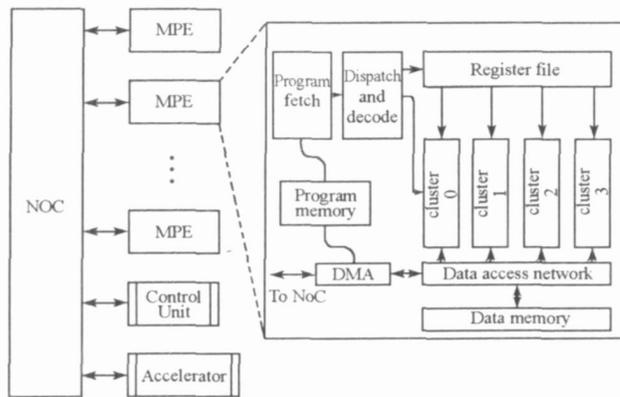


图3 GAEA结构示意图

Fig. 3 GAEA architecture diagram

5 结束语

当前软件无线电距离成熟和大规模部署应用尚有不小的距离,数字信号处理器仍然是软件无线电发展的重点。目前,针对SDR-DSP还没有公认的权威的测试集,难以针对不同结构进行量化比较。提出的各种SDR-DSP也尚未有成功的规模化和商业化应用案例。未来SDR-DSP将会继续沿着提高处理器性能和减少功耗的方向发展,新型的结构和指令集将会不断涌现。

参考文献:

- [1] Glossner J, Iancu D, Moudgill M, et al. Trends in Low Power Handset Software Defined Radio[C]//Proceedings of the Embedded Computer Systems: Architectures, Modeling, and Simulation, 2007.
- [2] Lee H, Lin Y, Harel Y, et al. Software Defined Radio — A High Performance Embedded Challenge[C]//Proceedings of the 1st International Conference on High Performance Embedded Architectures and Compilers, 2005.
- [3] Woh M, Seo S, Lee H, et al. The Next Generation Challenge for Software Defined Radio[C]//Proceedings of the International Conference on Embedded Computer Systems — Architectures, Modeling and Simulation, 2007.
- [4] Fang X, Chen SM. The Design and Evaluation of a MIMO-OFDM Benchmark[C]//Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008.
- [5] Lin Y, Lee H, Woh M, et al. SODA: A High-performance DSP Architecture for Software-defined Radio[J]. IEEE Micro, 2007, 27(1): 114–123.
- [6] Woh M, Seo S, Mahlke S, et al. AnySP: Anytime Anywhere Anyway Signal Processing[C]//Proceedings of the ISCA'09, Austin, Texas, USA, 2009.
- [7] Parizi H, Niktash A, Kamalizad A, et al. A Reconfigurable Architecture for Wireless Communication Systems[C]//Proceedings of the Third International Conference on Information Technology: New Generations, 2006.
- [8] Mei B F, Vemalade S, Verkest D, et al. ADRES: An Architecture with Tightly Coupled VLIW Processor and Coarse-grained Reconfigurable Matrix

[C]// Proceedings of the 13th International Conference on Field-programmable Logic and Applications, 2003.

[9] Rauwerda G K, Heysters P M, Smit G J M. Towards Software Defined Radios Using Coarse-grained Reconfigurable Hardware[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2008, 16(1): 3- 13.

[10] 高德远. 软件无线电的可重构流处理器体系结构[J]. 航空学报, 2008, 29(6).

[11] Ramacher U. Software-defined Radio Prospects for Multistandard Mobile Phones[J]. Computer, 2007, 40(10): 62- 69.

[12] Schulte M, Glossner J, Junturkar S, et al. A Low-power Multithreaded Processor for Software Defined Radio[J]. Journal of VLSI Signal Processing Systems for Signal Image and Video Technology, 2006, 43(2- 3): 143- 159.

[13] Wu K H, Kanstein A, Madsen J, et al. MF-ADRES: Multi-threading on Coarse-grained Reconfigurable Architecture[J]. International Journal of Electronics, 2008, 95(7): 761- 776.

[14] Lin Y. Realizing Software Defined Radio — A Study in Designing Mobile Supercomputers[D]. United States, Michigan: University of Michigan, 2008.

[15] Woh M, Lin Y, Seo S W, et al. From SODA to Scotch: The Evolution of a Wireless Baseband Processor[C]// Proceedings of the 41st Annual IEEE/ACM International Symposium on Microarchitecture, 2008.

[16] Svan Berkel K, Heinle F, Meuwissen P P E, et al. Vector Processing as an Enabler for Software-defined Radio in Handheld Devices[J]. Eurasip Journal on Applied Signal Processing, 2005(16): 2613- 2625.

[17] Fang X, Wang D, Chen S M. SPVA: A Novel Digital Signal Processor Architecture for Software Defined Radio[C]// Proceedings of the 2008 IEEE/ACS International Conference on Computer Systems and Applications, 2008.

[18] Mamidi S, Blem E, Schulte M J, et al. Instruction Set Extensions for Software Defined Radio[J]. Microprocessors and Microsystems, 2009, 33(4): 260- 272.

[19] Mamidi S, Schulte M J, Xie Z P, et al. Arithmetic Units for Software Defined Radio[C]// 2006 Fortieth Asilomar Conference on Signals, Systems and Computers, 2006: 341- 346.

[20] Woh M, Lin Y, Seo S, et al. Analyzing the Scalability of SMD for the Next Generation Software Defined Radio[C]// Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal, 2008.

[21] 车德亮. LS-DSP 数字信号处理器总线的低功耗设计[J]. 国防科技大学学报, 2005, 27(2).

(上接第 5 页)

参考文献:

[1] Rixner S, Dally W J, et al. A Bandwidth-efficient Architecture for Media Processing[C]//The 31st Annual ACM/IEEE International Symposium on Microarchitecture, USA, 1998: 3- 13.

[2] 伍楠. 高效能流体系结构关键技术研究[D]. 长沙: 国防科技大学, 2008.

[3] Stream Processors Inc. Some 1 SP16-G160 Stream Processor Data Sheet[R]. <http://www.streamprocessors.com>, 2007.

[4] <http://www.Tilera.com/products/processors.php>, 2007.

[5] Hofstee H P. Power Efficient Processor Architecture and the Cell Processor[C]// Proceedings of the 11th International Symposium on High Performance Computer Architecture, 2005: 258- 262.

[6] Yang X, Yan X, et al. A 64-bit Stream Processor Architecture for Scientific Applications[C]// Proceedings of the 34th Annual International Symposium on Computer Architecture, 2007: 210- 219.

[7] Karthikeyan S, et al. Distributed Microarchitectural Protocols in the TRIPS Prototype Processor[C]// Proceedings of the 39th Annual International Symposium on Microarchitecture, 2006: 480- 491.

[8] StreamIt Group. StreamIt Wepage[R]. <http://catfish.csail.mit.edu/streamit/>.

[9] Mattson P. A Programming System for the Imagine Media Processor[D]. Dept. of Electrical Engineering, Ph.D. Thesis, Stanford University, 2001.

[10] Foley B T, Horn D, Sugeman J, et al. Brook for GPUs: Stream Computing on Graphics Hardware[C]// SIGGRAPH, 2004: 777- 786.

[11] Fatahalian K, et al. Sequoia: Programming the Memory Hierarchy[C]// Proceedings of SC' 06, 2006: 207- 216.

[12] NVIDIA Inc. <http://www.nvidia.com>.

[13] Munshi A. OpenCL: Parallel Computing on the GPU and CPU[R]. Technique Report, SIGGRAPH2008, 2008.

[14] Otto F, et al. Tichy: Streaming Extensions for Object-oriented Languages[C]// Workshop on Streaming Systems: From Web and Enterprise to Multicore 41st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2008.

[15] Wu N, Wen W, et al. Streaming HD H. 264 Encoder on Programmable Processors[R]. Technique Report, 2009.

[16] Francois Labonte, et al. The Stream Virtual Machine[C]// PACT 2004: 267- 277.

[17] Choi Y, et al. Stream Compilation for Real-time Embedded Multicore Systems[C]// Proceedings of 2009 International Symposium on Code Generation and Optimization (CGO), 2009: 210- 220.

[18] Bouzas B, et al. MultiCore Framework: An API for Programming Heterogeneous Multicore Processors[C]// Workshop on Streaming Systems: From Web and Enterprise to Multicore 41st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2008.

[19] Zhang D, et al. A Lightweight Streaming Layer for Multicore Execution[J]. SIGARCH Computer Architecture News, 2008, 36(2): 18- 27.

[20] Udupa A, et al. Software Pipelined Execution of Stream Programs on GPUs[C]// Proceedings of 2009 International Symposium on Code Generation and Optimization (CGO), 2009: 200- 209.

[21] Khailany B, Dally W J, et al. Exploring the VLSI Scalability of Stream Processors[C]// Proceedings of the 9th Symposium on High Performance Computer Architecture, Anaheim, California, USA, 2003: 153- 164.

[22] Dally W J, et al. Efficient Embedded Computing[J]. IEEE Micro, 2008: 27- 32.