

## ET:一种能耗有效的高性能嵌入式处理器\*

杨乾明,伍楠,管茂林,张春元,全巍,黄达飞  
(国防科技大学 计算机学院,湖南 长沙 410073)

**摘要:**随着标准和算法的不断演进,高端嵌入式应用对性能和能耗提出了越来越高的要求。然而,能耗问题成为将 VLSI 潜力转换为实际应用需求的最大挑战,基于此,提出 ET(Embedded Tera-scale Computing) 处理器设计。ET 以众多轻量级处理器(称为小核)来搭建目标处理器,每个小核都是一个基于显式数据和指令管理的 VLIW 处理器,能单独执行一个线程,采用层次化的寄存器文件和非对称全分布式指令寄存器来分别降低数据和指令的供应能耗。为了进一步降低功耗,ET 处理器采用了较短的运算流水线和简单的循环控制结构,并面向应用领域针对循环体进行优化。初步的实验结果表明,在 40nm 工艺下,ET 处理器可以获得单芯片 1TOPS 以上的性能,同时保持操作能效比在 100GOPS/W 以上。

**关键词:**嵌入式计算;能耗有效;层次化寄存器文件

**中图分类号:**TP302 **文献标识码:**A

## ET: An Energy-efficient Processor Architecture for Embedded Tera-scale Computing

YANG Qian-ming, WU Nan, GUAN Mao-lin, ZHANG Chun-yuan, QUAN Wei, HUANG Da-fei  
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

**Abstract:** As criterions and algorithms evolve and become more complex, high performance embedded application demands the high performance and energy efficiency. The challenge, however, is how to turn the VLSI capability into the actual computing performance. This research proposed an energy efficient processor architecture named ET (Embedded Tera-scale Computing), which is composed of many lightweight VLIW processor cores, also named small cores. Each core executes a thread with the mechanisms for explicitly managing the data and instructions. ET uses a hierarchical data registers to reduce the cost of delivering data, and the asymmetric and distributed instruction registers to deliver the instructions. In order to further reduce the energy, ET employs non-deep pipeline and simple control flow and optimizes the execution of loop body of applications. The primary result shows that ET can achieve the 1TOPS performance and the 100GOPS/W efficiency when scaled to 40nm.

**Key words:** embedded computing; energy-efficient; hierarchy register file

随着通信标准、压缩算法的不断演进,典型高性能嵌入式应用,比如基带信号调制、高清视频压缩、高清电视等,对处理器提出了越来越高的性能和能耗需求。例如,1.44Mb/s 的 3G 移动通信手持设备只需要 35~40GOPS 的处理能力,而对于采用 100Mb/s OFDM 通道的 4G 移动通信,其手持设备性能则需要 210~290GOPS<sup>[1]</sup>,基站信号处理的性能需求更是达到了 1TOPS<sup>[2]</sup>,同时对能耗的需求也达到了 100~1000GOPS/W<sup>[2]</sup>。另一方面,VLSI 工艺的发展,也为设计满足应用需求的嵌入式处理器提供了可能,纳米工艺将使单芯片中晶体管密度达到每平方厘米包含百亿至千

亿,2010 年已出现集成度超 10 支亿晶体管的高性能 CPU (Intel 的 Teraflops 处理器<sup>[3]</sup>)。

然而,由于在不抑制亚阈值泄漏电流的情况下,阈值电压不能进一步减小<sup>[4]</sup>,过去几十年用于实现“立方”能量缩放(每器件的能量按器件基本特征尺寸减少的三次方缩减)的方法走到了终点,现有的处理器结构无法在借助工艺提升性能的同时保持相对较低的能耗,满足不了未来嵌入式应用发展的需求。当前主流嵌入式处理器的性能位于 10GOPS~100GOPS,能耗有效性也仅在 4GOPS/W<sup>[5]</sup>左右,与应用的实际需求相差甚远。而最近涌现的采用新型体系结构的前沿处理器,

\* 收稿日期:2011-06-20

基金项目:国家自然科学基金资助项目(61033008,60903041);博士点基金资助项目(20104307110002)

作者简介:杨乾明(1984—),男,博士生。

如 Fermi<sup>[6]</sup>、Tile-Gx100<sup>[7]</sup>、CELL<sup>[8]</sup>、Storm<sup>[9]</sup>等,虽然在性能上有了很大提高,但是其能耗有效性与实际需求仍有着较大的差距。当然,如果在处理器中采用大量的专用设计,可以获得 200GOPS/W<sup>[10]</sup>的能耗有效性,同时保证有一个相当可观的性能,但是随着算法复杂度的增加,其实现成本将变得难以承受。如何在体系结构作出创新,在提升处理器性能的同时保证能耗的较低增加,将 VLSI 潜力转换为实际的运算能力是一个亟须解决的问题。

当前有些研究在能耗有效的体系结构方面进行了积极的探索,得到了一些有益的结论。ELM<sup>[11]</sup>采用层次化的寄存器设计和分布式指令存储,有效降低了数据和指令供应的能耗,使得整个处理器的能耗效率达到普通 RISC 处理器的 23 倍。Feenecs<sup>[12]</sup>是一种基于定制指令集的异构多核体系结构模板,它由多个多线程 SIMD 并行加速单元和普通标量控制单元组成,采用多级数据存储层次,在 40nm 的工艺下,使用特定指令集可获得 900GOPS/W (8 位操作)的性能。AnySP<sup>[13]</sup>则是一款面向 3G/4G 无线通信和高清视频应用的可重构 SIMD 处理器模型,它由 SIMD 核和标量核组成。SIMD 核的基本运算单位称为 FFU,共 64 个,分为 8 组,每组 8 个,每个 FFU 含有乘法单元、加法单元、Swizzle 单元和 ALU 单元组成,可以支持三种执行模式:宽向量模式(宽度为 64)、窄向量模式(8 个,每个宽度为 8)、向量流水模式(2 级流水,每级宽度为 32)。AnySP 采用了寄存器分块、寄存器读写旁路、多流出货量、指令对打包等低功耗技术,可以获得 200 ~ 1000GOPS/W (8 位操作)的能效。Rehan hameed<sup>[14]</sup>则研究了应用在通用 CMP 上执行时的能耗开销,发现面向应用采用 SIMD 或 VLIW 加速阵列、定制指令和特定功能单元可以极大提高处理器的性能,同时有效降低功耗。

本文提出面向高性能嵌入式应用的能耗有效的可编程嵌入式处理器(ET: Embedded Tera-scale Computing)设计。ET 采用异构多核结构,融合了层次化寄存器文件、分布式指令存储器、循环指令缓冲、简单的多指令流出机制(VLIW)、显式的指令和数据传输、较浅的流水线等低功耗思想。实验结果显示,在 40nm 的工艺下,ET 有望获得超过 1TOPS 的性能和 100GOPS/W 的能效。

## 1 ET 体系结构

ET 的整体结构如图 1 所示,它是一个大规模

显式并行的异构多核结构,重量级的标量核(Scalar Core)负责控制任务在各个线程簇(cluster)上执行,并显式控制数据在各个线程簇之间移动;线程簇由多个微线程处理器完成,每个微线程处理器含有独立的指令执行单元,可以执行一路独立的 VLIW 线程,而位于同一个线程簇内部的微线程处理器可以根据需要配置成 SIMD 向量执行模式:4 个 4 路 SIMD,2 个 8 路 SIMD 或 1 个 16 路 SIMD。

ET 从嵌入式应用的特征出发,采用硬件尽量简单、软件足够智能的设计思想,综合考虑了各种能耗有效的体系结构技术,在设计上做出很多创新,下面详细介绍。

### 1.1 以小核为中心

目前国际上对于多(众)核的设计有两种典型的设计理念,一是主张采用少量的重量级处理器核(俗称大核),这样可以提高单线程的性能,并且降低线程调度的难度;二是主张采用大量的轻量级处理器核(俗称小核),这样虽然单线程的性能较低,但是对于并行度较高的应用来说,可以通过多个小核来获得应用性能的提升,而对于并行度较低场合,系统的吞吐率可以达到很高,同时由于每个小核都非常简单,可以精心地设计,在能耗、面积等方面进行最大的优化。

针对相对专一的嵌入式应用领域,ET 处理器根据应用并行度较高的特点,采用众多小核来搭建目标处理器,同时引入线程簇的概念来降低调度的难度。每一个小核(微线程处理器)都是一个面向能耗进行优化的 VLIW 处理器核,可以独立执行一个 VLIW 线程。小核选用 VLIW 结构基于两点考虑:(1)相对简单的多指令发射机制。为了获得足够的性能,可编程处理器必须采用大规模并行,典型的并行执行机制有 VLIW、多线程、超标量和 SIMD,而 VLIW 将大量的指令调度工作交给编译,被证明是一种硬件非常简单且非常有效的并行手段。(2)可以降低指令执行过程中流水线寄存器的能耗。传统处理器在执行流水线的每一拍就将指令的所有信息全部取出(操作码、源操作数地址、目的操作数地址等),然后在流水线的每一拍进行寄存,直到使用它的那个节拍才会释放掉,虽然这是一种很自然、简单的方式,但是却带来能量的开销。因为实际上,对于源操作数地址只有在读寄存器段使用,而目的操作数地址更是只会在指令执行完的写回段才会使用,而 VLIW 可以将一条指令的信息打散,借助于编译调度在需要使用的那一拍才流出,从而有效

降低功耗。

每个小核的结构如图 1 右上部分所示,含有 5 个发射槽(slot),其中三个对应于 ALU 运算单元,一个对应于数据加载(保存)单元(DMU:Data Manage Unit),一个对应于指令加载和执行控制单元(IMU:Instruction Manage Unit)。每个 ALU 含有两个超小的寄存器文件(TORF)作为数据的输入;DMU 是小核与外界的数据通道,负责数据的加载和保存,它含有一个较大容量的寄存器组来捕获指令间的数据重用,DMU 能够根据指令的

信息,进行数据地址的自动计算,从而实现数据的预取;IMU 则负责加载指令,并控制程序的执行,在 ET 中只支持简单的循环控制结构,不支持 if/else 结构(所有 if/else 由程序员转化为 select 选择操作),从而最大限度地简化执行控制逻辑,降低能耗。所有的这些单元通过零延迟的通信网络(distributed switch)来互连,为了有效地降低互连的能耗,ET 采用部分连接的 crossbar,通信的性能则依赖于编译高效的调度。

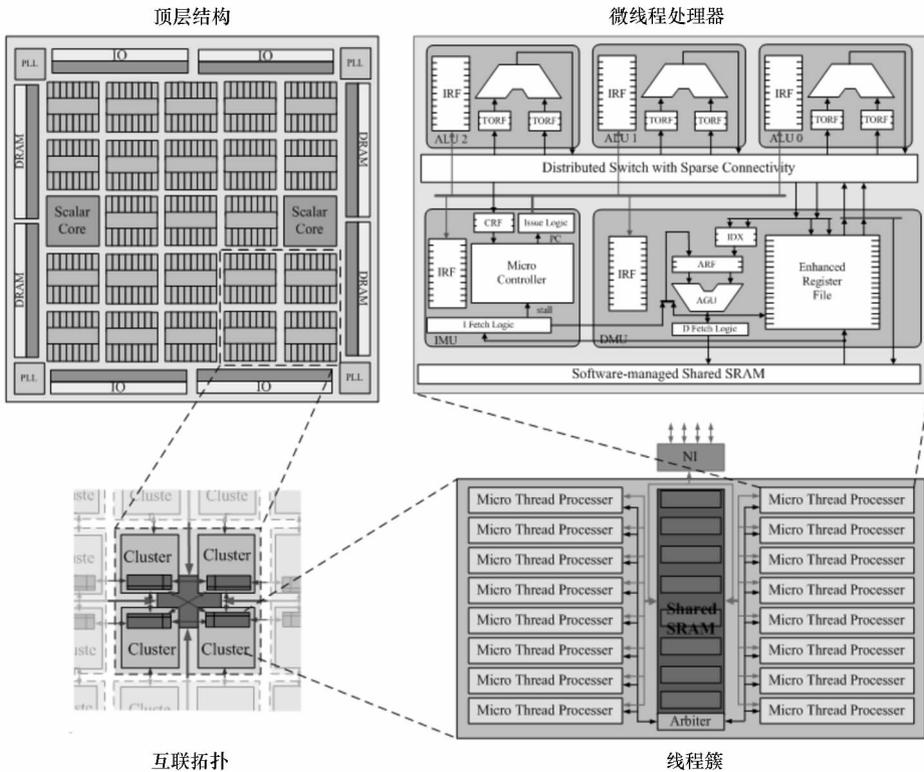


图 1 ET 整体结构  
Fig. 1 The architecture of ET

为了降低全局的互连延迟和长线能耗,所有的小核以局域化的形式组织,每 16 个小核构成一个线程簇,每 4 个线程簇构成一个大的运算核心。线程间的通信分为两种:(1)线程簇内部的所有线程通过共享存储进行通信。ET 的每个线程簇内部含有一个多 bank 的软件管理的片上存储(SM:Shared SRAM),数据在 SM 上的位置和生命周期由编译器进行显式管理和调度。(2)线程簇间的通信通过类似于消息传递机制的流传输协议<sup>[15]</sup>进行通信。由嵌入在众多小核中的大核(标量核)以执行指令的方式来启动批量数据(称为流)的传输,整个传输的过程都由大核来控制,通信的路径由程序员和编译器共同指定。由于不同通信方式的能耗开销不同,编译器在进行线程调

度时按如下策略进行:通信频率较高的线程尽量调度到同一个线程簇的小核上运行,通过本地的共享存储来降低通信的能耗,而对于通信较少的线程,则可以分配到不同的线程簇执行,以获得较高的并行处理效率。

### 1.2 多级数据存储层次

为了降低数据供应的功耗,ET 的小核采用了多级数据存储层次,如图 2 所示。

#### (1) 软件管理的片上存储

传统的嵌入式处理器中通常设有片上 cache 来缓存输入输出的数据,从而有效降低访存的延迟并减少对片外带宽的需求,但同时也消耗了处理器中大部分的能耗,这主要是因为:(1)每次对

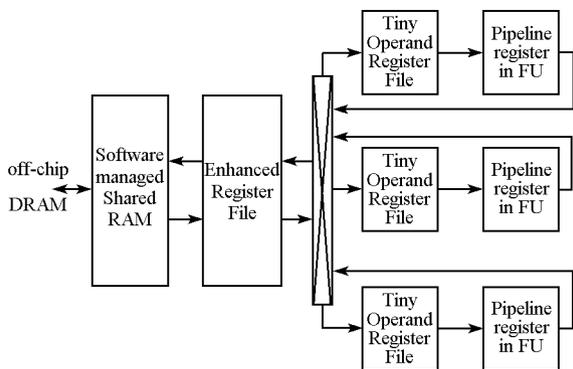


图2 小核存储层次

Fig.2 Memory hierarchy of micro thread processor

cache 的访问都要进行标志位的比较,使得每次访问的能耗很高;(2)自动的预取机制具有盲目性,造成无效的数据预取;(3)固定的替换策略不能有效匹配应用的特征,额外增加了片外访存量。为了降低片上访问的能耗,ET 采用软件管理的片上存储(称为 SM),用户通过关键字 share 来告诉编译器该数据块放在 SM 中,而编译器则负责安排数据在 SM 上的位置和分布。程序执行时,由标量核负责将数据从片外加载到 SM。

## (2) 层次化寄存器文件

随着软流水、循环展开等优化手段在编译器中的广泛采用,程序的并行性得到极大开发,带来了性能显著提高,但同时增加了对寄存器的需求,使得处理器中必须设置大量的寄存器以暂存程序运行的中间结果,而 VLIW 的结构更是增加了寄存器的需求。而我们都知,寄存器每次访问的能耗与寄存器的数目成正比<sup>[16]</sup>,寄存器数目的增加带来了每次寄存器访问能耗的显著增加,使得指令在执行时将花费很大一部分能量用于寄存器访问。如果能将这个较大的寄存器组分成多个小的寄存器组,使得每次寄存器访问只发生在其中的一个小寄存器组,那么将有效减少指令执行过程中的寄存器访问能耗,从而有效降低处理器的整体运行时的能耗。应用(特别是嵌入式应用)的特征为这种分层的思想提供了可能,我们对嵌入式程序核心算法的研究表明,大部分核心算法在运行时对寄存器的平均需求不是很大,通常4个左右就足够了,只是偶尔会需要比较多的寄存器,并且其中有相当一部分变量的生命周期较长,但是访问的频率很低。因此,相比于传统的寄存器文件结构,ET 的寄存器文件设计成两层:超小寄存器文件(TORF, Tiny Operand Register File)和多模式寄存器文件(ERF, Enhanced Register File)。

TORF 是大小为4项的超小型寄存器组,具有一个读端口和一个写端口,每次访问的能耗可以低至110fJ(读),低于访问大小为32项寄存器的能耗的1/8,可以有效地降低指令执行过程中的寄存器访问能耗。TORF 主要用于存放生命周期较短(1~2拍)的变量,用于捕获短生命周期变量的重用,由编译器负责分配。由于TORF的端口数目并没有减少,保证了与功能单元之间的通信带宽,因此对性能影响很小。

ERF 是大小为32项的寄存器组,作为TORF的扩展,用于存放长生命周期的变量(典型的如常量、循环相关变量等),或者用于存放编译时从TORF溢出的变量,从而保证ET执行时的性能。虽然ERF比较大,每次访问的能耗较大,但是由于访问的频率较低,因此仅带来少量的能耗增加。ERF的另一个作用就是对数据进行预取。为了降低能耗,更高效地利用片上存储,数据在ET存储层次之间的移动都是软件显式控制的。ERF实现数据预取也是通过执行指令来实现,每条指令可以加载或保存一个数据块,这样做有两个好处,一是ERF可以进行自动的地址计算,减少了load/store指令的数目,降低了产生数据地址的能耗;二是可以将数据加载指令和运算指令并行执行,隐藏访存的延迟。此外,ERF采用寄存器间接寻址,增加了指令访问数据的灵活性,可以降低编译的难度,增加数据的重用。

## (3) 流水线寄存器(pipeline register)

通过频率的不断提升来获得处理器性能的不不断提升,是被实践证明的行之有效的手段。然而,频率的不断提升也带来了流水线段数的不断加深,使得流水线寄存器的数目显著增加,增加了指令执行的能耗,已有的研究表明,流水线寄存器的能耗占据了执行过程中数据供应能耗的9%<sup>[17]</sup>。为了降低指令执行过程的能耗,ET采用较浅的流水线,以减少流水线寄存器的数目。典型的加法延迟为2拍,逻辑操作的延迟为1拍。

## 1.3 非对称全分布式指令寄存器

在当前大部分可编程处理器中,都采用单周期访问延迟的大容量指令cache来提高性能,但是由于每次cache访问需要进行tag位比较、大规模存储阵列访问,造成每次取指都消耗大量能量,使得指令供应的能耗成为处理器能耗的一个主要部分<sup>[18]</sup>。ET使用较小容量的指令寄存器文件(IRF, Instruction Register File)来降低每次取指令的开销。相比较于小容量的指令cache,比如filter cache,IRF有几个优点:首先,IRF的访问能

耗更低,cache 的每次访问需要进行 tag 标志位的对比,而 IRF 不需要,直接使用地址访问即可;其次,cache 是硬件管理的,有着固定的替换机制,而 IRF 是软件管理的,由编译器决定指令块的加载位置和加载时机,使得 IRF 的利用率更高。最后,IRF 通过软件显式控制,实现了循环缓冲(loop buffer)的功能,可以将循环体的指令在多次重复执行过程中始终保存在 IRF 中,提高了以循环为主的嵌入式应用的指令供应能效。

ET 的 IRF 是一个非对称的全分布式结构,如图 3 所示。

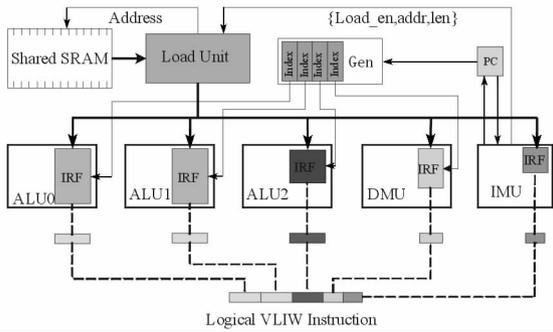


图 3 非对称全分布式指令寄存器结构

Fig. 3 Asymmetrical distributed instruction register file

IRF 具有以下特点:首先,它是一个全分布式的,每个功能单元都与一个较窄的 IRF 绑定。程序执行时,指令流出单元不需要通过很长的路径将指令发射到功能单元,而仅仅发射指令到本地 IRF 中的位置信息(index),使得可以从本地取指执行,从而节省指令发射能耗;其次,IRF 是非对称的,与不同功能单元绑定的 IRF 具有不同的容量。为了降低功耗,ET 小核中的不同发射槽用于完成不同的功能,典型的组成为 2 个加法单元、1 个乘法单元、DMU 和 IMU 单元,这些功能单元具有不同利用率,比如 DMU 和 IMU 单元执行的指

令数就比运算单元的少。因此根据应用的特征设置合适的 IRF 大小,可以有效地降低每次访问 IRF 的能耗。在 ET 中,ALU0 和 ALU1 设置为 32 项,ALU2 (乘法单元)设置为 16 项,DMU 为 16 项,IMU 为 8 项。

基于 IRF 的指令执行过程完全由编译器显式控制。编译器根据 IRF 的大小和程序特征自动将程序划分成指令块,然后指定指令块在 IRF 中的位置,由于编译器知道指令块的生命周期,因此可以将指令块的加载指令恰到时机地插入到正常的控制指令序列(在 IMU 中执行)中。

## 2 评测

为了验证 ET 的有效性,本文基于 TSMC 65nm 工艺,利用 EDA 工具对 ET 处理器进行了初步的评估,评估过程如图 4 所示,所有流程都是基于标准单元,也没有使用 clock gating、多值电压、power gating 等优化手段。最终得到的电路工作在 1.0V、100MHz 情况下,此时,ET 一个小核的峰值性能为 300MOPS,面积为 0.345mm<sup>2</sup> (含 8KB 的 SM)。

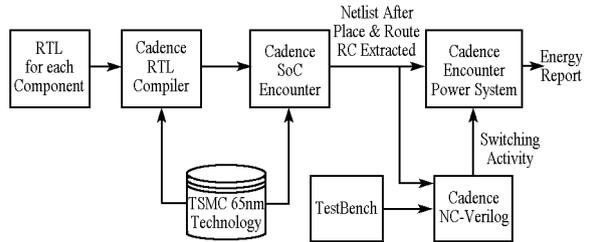


图 4 评估流程

Fig. 4 Flow used for power estimation

测试程序全部来自典型嵌入式应用的核心算法,如表 1 所示,采用 C 语言编写,使用普通的 VLIW 编译器编译,再对代码进行手工调整,以符合 ET 小核的执行机制。

表 1 测试程序

Tab. 1 Benchmark

核心算法	描述
fdct	8 × 8 行列离散余弦变换,广泛用于图像正交变换。
iir	无限冲击响应滤波器,用于声音合成。
correlate	对两个序列检测器相关性。广泛用于图像处理、信号检测和识别及脉冲编码调制等。
rgb2yuv	将 RGB 的图像转化为 YUV 的图像。用于图像编码前的预处理。
idct	2 D 8 × 8 离散反余弦变换。
fir	32 - tap 有限冲击响应滤波器。
quant_intra	量化,用于图像压缩。
me_full_search	全搜索运动估计,用于图像压缩编码。

续表

核心算法	描 述
err_diff	基于错误扩散的抖动(dither)方法,可以消除像素点之间的明显差异,用于打印机。
sad_8×8	残差计算,计算输入与参考的差异,广泛用于图像压缩编码。
conv3×3	3×3 卷积,表示系统对输入的响应,用于图形深度提取,低通或高通滤波。
FFT_r2	基 2FFT,广泛用于一维信号处理。

我们首先对 ET 一个小核的能耗、性能进行了详细评估,结果如图 5 所示。从图 5(a)、(b)可以看出,在 100MHz 情况下,ET 小核的平均功耗为 3.6mW 左右,实际的运算性能平均达到峰值性能的 60% 以上,对于某些特别规整的密集计算,如卷积(conv3×3)和量化(quant\_intra),则可达到 80%,整体的操作能效比达到 52GOPS/W 以上(见图 5(c)),是 Ti C64x(3GOPS/W, 130nm)的 18 倍。

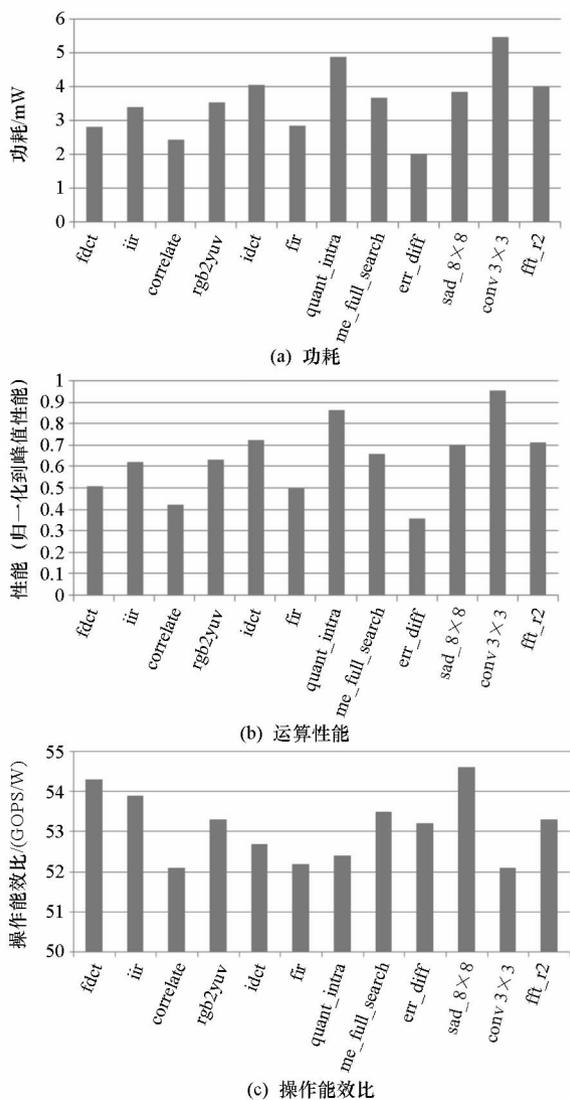


图 5 ET 小核的功耗和性能

Fig. 5 Power and performance of micro thread processor

如果将工艺推进到 40nm (相比于 65nm 工艺,电压降为 0.9V,器件尺寸变为原来的 60%,在处理器规模和频率不变的情况下,根据 Dennard 定律<sup>[19]</sup>,面积和能耗都可以降为原来的一半),那么可以在 13mm×13mm 的单芯片内放置 700 个这样的小核(芯片利用率为 75%左右),并且将频率提高到 500MHz,从而获得超过 1.05TOPS 的单芯片峰值性能,同时可以将芯片整体能耗控制在 10W 左右,获得超过 100GOPS/W 的操作能效比。如果再采取定制电路设计、多电压设计、power gating、clock gating 等多种优化手段,可以进一步提高 ET 的操作能效比,满足未来 4G 通信、高清视频处理应用的需求。

### 3 结论

VLSI 工艺的发展使得单芯片可以集成上亿甚至数十亿只晶体管,然而将如此多的晶体管转化为实际的运算能力需要解决很多难题,其中最大的挑战是在提升处理器性能的同时保证芯片的能耗,对于嵌入式计算,能耗问题则显得更为突出。本文面向高性能嵌入式应用,提出一种能耗有效的高性能嵌入式处理器结构——ET(Embedded Tera-scale Computing)。ET 采用大量的轻量级处理器核(称为小核)来搭建目标处理器,每个小核都是一个基于显式数据和指令管理的 VLIW 核,能单独执行一个线程。在传统处理器中,数据和指令的供应能耗占据了处理器功耗的 70%,ET 则针对此进行了优化,通过引入层次化数据寄存器文件、非对称全分布式指令寄存器、软件显式管理等机制来降低数据和指令供应的能耗。前期的评测结果显示,在 40nm 工艺下,ET 可以获得超过 1TOPS 的性能和 100GOPS/W 能效比。

ET 的工作才刚刚开始,由于时间的关系,目前仅完成小核的设计和初步评估,未来的工作包括对 ET 采用的各项技术的详细评估、开发 ET 的编译器、搭建完整的 ET 处理器。

## 参考文献:

- [1] Silven O, Jyrkka K. Observations on Power-efficiency Trends in Mobile Communication Devices [J]. EURASIP Journal of Embedded Systems, 2007(1):17.
- [2] Woh M, Lin Y, Seo S, et al. The Next Generation Challenge for Software Defined Radio [C]//Proceedings of the 7th International Conference on Systems, Architectures, Modeling, and Simulation, 2007:343-354.
- [3] Rattner J R. Tera-scale Computing: a Parallel Path to the Future [EB/OL]. [2007-05-10]. <http://software.intel.com>.
- [4] Kogge P, Bergman K, Borkar S, et al. ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems [R]. DARPA IPTO, 2008. 9.
- [5] Halfhill T R. MIPS Threads the Needle [R]. Microprocessor Report, 2006,20(2):1-8.
- [6] NVIDIA Inc. NVIDIA's Next Generation CUDA Compute Architecture: Fermi (White Paper) [R/OL]. [2009-06-12]. <http://www.nvidia.com>, 2009.
- [7] Tiler Inc. Tile-gx Processor Family Product Brief [R/OL]. [2009-05-20]. <http://www.Tiler.com>.
- [8] Pham D, Asano S, Bolliger M, et al. The Design and Implementation of a First-Generation CELL Processor [C]//Proceedings of IEEE International Solid-State Circuits Conference, 2005:184-185.
- [9] Khailany B, Williams T, Lin J, et al. A Programmable 512GOPS Stream Processor for Signal, Image, and Video Processing [C]//Proceedings of IEEE International Solid-state Circuits Conference, 2007:272-273.
- [10] Hsu S, Venkatraman V, Mathew S, et al. A 2GHz 13.6mW 12 × 9b Multiplier for Energy Efficient FFT Accelerators [C]//Proceedings of the 31st European Solid-state Circuits Conference, 2005:199-202.
- [11] Dally W J, Balfour J, Schaffer D B, et al. Efficient Embedded Computing [J]. IEEE Computer, 2008:27-32.
- [12] Cattoor F, Raghavan P, Lambrechts A, et al. Ultra-low Energy Domain-specific Instruction-set Processors [M]. Springer Science Publisher, 2009.
- [13] Woh M, Seo S, Mahlke S, et al. AnySP: Anytime Anywhere Anyway Signal Processing [C]//Proceedings of the 36th Annual International Symposium on Computer Architecture (ISCA), Austin, Texas, USA, 2009:20-24.
- [14] Hameed R, Qadeer W, Wachs M, et al. Understanding Sources of Inefficiency in General-purpose Chips [C]//Proceedings of the 37th Annual International Symposium on Computer Architecture (ISCA), 2010:37-47.
- [15] Wu N, Yang Q M, Wen M, et al. Tiled Multi-core Stream Architecture [J]. Transactions on High-performance Embedded Architectures and Compilers (HiPEAC), 2009.
- [16] Rixner S, Dally W J, Khailany B, et al. Register Organization for Media Processing [C]//Proceedings of the 6th International Symposium on High-performance Computer Architecture (HPCA), 2000:375-386.
- [17] Balfour J, Dally W J, Schaffer D B, et al. An Energy-efficient Processor Architecture for Embedded Systems [J]. IEEE Computer Architecture Letters, 2008, 7(1):29-33.
- [18] Segar S. Low Power Design Techniques for Microprocessors [C]//Proceedings of International Solid State Circuits Conference, 2001.
- [19] Dennard R H, Gaensslen F H, Yu H N, et al. Design of Ion-implanted MOSFETS with Very Small Physical Dimensions [J]. IEEE Journal of Solid State Circuits, 1974, 9(5):256-268.