通用处理器多媒体支持功能的分析与研究*)

成 运[1,2] 戴 葵[1] 王志英[1] 岳 虹[1]

(国防科技大学计算机学院 长沙410073)[1] (娄底学院计算机系 娄底417000)[2]

摘 要 大量的多媒体信息处理是现代通用处理器面临的一个现实问题,本文在对通用处理器中的多媒体扩展进行简单的回顾之后,对多媒体的特征进行了分析,得出了现有通用处理器中多媒体扩展的共同特征,然后对现有多媒体扩展指令集进行了分类介绍,研究了在通用处理器中进行多媒体功能扩展的发展特点,最后,指出发展多线程或超线程多媒体处理器将是通用处理器发展的必然趋势。

关键词 处理器,多媒体,子字并行,扩充

Study and Analysis of Supporting Multimedia Process in General-Purpose Processors

CHENG Yun[1,2] DAI Kui[1] WANG Zhi-Ying[1] YUE Hong[1]

(College of Computer, National University of Defense Technology, Changsha 410073)[1]

(Department of Computer, Loudi College, Loudi 417000)[2]

Abstract It is a reality that general processors encounter a mass of multimedia workloads. After giving an overview of the multimedia extensions in general processors, characteristics of multimedia processing are analyzed and common feature is given in this paper. Classes of instructions for multimedia extensions are introduced and the developing features in the multimedia extensions for the general processors are also studied in this paper. In the end the authors point out that it will be a nevitable trend to develop multithreaded or superthreaded multimedia processors for general-purpose processors.

Keywords Processors, Multimedia, Sub-word parallelism, Extensions

1 引言

多媒体信息的大量涌现是现代信息社会的普遍特点,因此,卓越的多媒体处理能力就成了对现代计算机的必然要求之一。图像和视频等多媒体信息的处理往往需要很大的计算量,这在通用处理器性能不高时是很难胜任的,过去,通常是用专用的硬件或者用 DSP 来实现,所以,直到20世纪90年代初期,人们还认为,在通用处理器中进行多媒体信息处理是不可能的。然而,随着多媒体信息的逐渐普及以及通用处理器性能的不断提高,同时,也为了扩大通用处理器的应用范围,增强通用处理器多媒体处理能力就成了通用处理器发展的一种必然趋势。在对通用处理器进行多媒体功能扩展时,要充分考虑到多媒体应用的特点,有些媒体的处理(如音频、视频等)需要有较好的实时处理能力,如对视频的处理,在视频会议中,如果要求每秒显示15帧160×160的图像就需要处理器每秒至少进行10多亿次[1]运算,相比之下,如果要显示更高分辨率的图像就需要有更高的性能。单纯依靠通用处理器中的通用指令是难以胜任多媒体处理这种需要有高度密集计算的任务的,因此,必须在通用处理器中增加新的多媒体功能处理部件以及相应的多媒体处理指令才能完成对多媒体信息的高效处理。

在通用处理器中进行多媒体扩展时,不能像专用媒体处理器那样,设计专门的实现某一特殊媒体算法的硬件电路,所以在通用处理器中进行多媒体扩展指令集的设计时,主要考虑能够普遍并且高效支持各类多媒体处理算法的通用型多媒体处理指令[2],在这些多媒体处理指令的基础上,可以高效地构造出复杂的多媒体处理操作、循环以及各种多媒体处理应用程序。

本文第1部分简单回顾通用处理器中的多媒体扩展;第2部分阐述通用处理器中多媒体扩展功能部件的执行方式;第3部分阐述通用处理器中的多媒体扩展指令系统分析;第4部分阐述在通用处理器中进行多媒体功能扩展的发展趋势;最后是结束语,指出发展多线程或超线程多媒体处理器将是通用处理器发展的必然趋势。

2 通用处理器中多媒体扩展的简单回顾

大多数通用处理器的体系结构都已经进行了多媒体功能扩展。最早推出市场的含多媒体扩展的处理器是1994年1月由 HP 公司推出的 PA-RISC,称其为 MAX-1(Multimedia Acceleration extensions)[3]。1995年,SUN 公司在其 Sparc ISA(Instruction Set Architecture)上扩展了 VIS(Visual Instruction Set)[4],HP 公司在其64位处理器 PA-RISC2.0上扩展了 MAX-2指令[5]。1996年 SGI 公司宣布在其 MIPS 处理器上扩展了 MDMX(Mips Digital Media extension)指令[6],DEC 公司(现已并入 HP 公司)宣布在其 Alpha 上特别为加速 MPEG-2编码而扩展了 MVI(Motion Video Instruction)指令[6],Intel 公司在其 ix86 ISA 上扩展了多媒体处理指令 MMX(Multi-media extension)[7]。1997年 AMD 公司宣布在

*)本课题受国家自然科学基金(60173040)资助。成 运 博士生,副教授,主要研究方向为计算机系统结构、多媒体信息处理与多媒体处理器设计。

其处理器上扩展了3Dnow!指令[8]，1998年 Intel 公司宣布又在其处理器上扩展了 SSE 指令[9]，Motorola 公司宣布在其 Power PC 上扩展了 AltiVec 指令[10]，1999年 MIPS 公司宣布在其 MIPS64处理器上扩展了 MIPS-3D 指令[11]，AMD 公司宣布扩展了增强3Dnow!指令[12]，2000年 Intel 公司宣布扩展了 SSE2指令[13]等等。表1说明了几种通用处理器的多媒体扩展情况[14]。

表1 通用处理器中的多媒体扩展

供应商	多媒体扩展	基本 ISA	宣布或上市时间	指令数	寄存器文件
HP	MAX-1	PA-RISC	1994	9	整数(31×64位)
Sun	VIS	SPARC v9	1995	121	浮点(31×64位)
HP	MAX-2	PA-RISC w /MAX-1	1995	8	整数(32×64位)
MIPS	MDMX	MIPS-V	1996	74	浮点(31×64位)累加器(1×192位)
DEC	MVI	Alpha	1996	13	整数(31×64位)
Intel	MMX	X86	1996	57	浮点(8×64位)
AMD	3DNow!	X86 w/MMX	1997	21	浮点(8×64位)
Intel	SSE	X86 w/MMX	1998	70	8×128位
Motorola	AltiVec	PowerPC	1998	162	32×128位
MIPS	MIPS-3D	MIPS64	1999	23	浮点(32×64位)
AMD	增强3DNow!	X86 w/MMX,3DNow!	1999	24	浮点(8×64位)
Intel	SSE2	X86 w/MMX,SSE	2000	144	8×128位

3 通用处理器中多媒体扩展功能部件的执行方式

根据对大量多媒体处理算法的研究，尤其是多媒体处理算法核心部分的深入研究，人们发现：多媒体信息处理具有以下一些相关特征[15]：1)低精度的数据类型(数据宽度通常低于16位)；2)进行密集计算的短小循环；3)大量的数据级并行性；4)流式数据，数据局部性差；5)要求较大的存储器带宽。

根据多媒体数据的低精度特性、大量的数据级并行性以及通用处理器的字处理宽度都在32位以上(含32位)等基本特点，于是在通用处理器的多媒体扩展中就引入了子字(Sub-word)单元的概念，在扩展的多媒体功能部件中进行子字单元并行(sub-word parallelism)处理[16]。

在子字单元并行处理方式中，一个标准的计算单元或存储单元，也即一个字单元，被分为称做子字的更小单元，对于一个字单元中的多个子字单元可以执行相同的操作，从而提供了一种有效的 SIMD(Single Instruction Multiple Data)处理方式。指令可以对整个字单元进行操作，也可对字单元中的子字单元进行并行操作。子字单元可以有不同的宽度，对于多媒体信息的处理来说，最常用的子字单元宽度为16位。8位宽或者32位宽的子字单元在某些情况下也是有用的，但和16位宽子字单元比起来，8位宽子字单元的精度不如16位子字单元的好，而32位宽子字单元的并行性又不如16位宽子字单元的好。从理论上讲，子字单元可以交迭，也可以不交迭；可以部分或者全部填充整个字；可以用硬件实现，也可用软件实现；可以处理整型数据，也可处理浮点数据。在实践中，为了降低复杂性，子字单元采用不交迭、完全填充且用硬件实现，这样的子字单元并行子集也被称做紧缩并行(Packed Parallelism)[2]。在 MMX、MAX、VIS、MVI 和 MDMX 中，子字单元类型为整数，而在 SSE、3Dnow!、MIPS-3D ASE 和 Altivec 中，子字单元类型为浮点数。

子字单元并行为基于字单元的处理器提供了一种代价较低的小规模的 SIMD 并行处理实现方式，因为只需在一个字宽的整数单元上增加少量的硬件开销就可以分成多个并行的子字单元。比如说，一个64位补码整数加法器在一个周期里可以允许4个16位整数子字单元并行相加，或者一个单一的64位整数相加，这样处理的硬件开销非常小，因为两种情况下的数据路径是相同的：两个64位寄存器读和一个寄存器写，只需要阻塞其中3个16位加法器的边界进位就可以了。同时，子字并行对寄存器文件端口数的要求也比较低：比如一个具有两个 ALU 的超标量微处理器可以支持8路并行操作，但只需要寄存器文件具备6个端口(4个读端口，2个写端口)就可以了，如果是用8个独立的16位功能单元来进行8路并行操作的话，则寄存器文件需要有24个读写端口(每路需2个读端口和一个写端口)。这就说明，子字单元并行组织形式，既适合于用于特殊目的的专用媒体处理器，也适合于已有基于字单元操作数据路径的通用处理器[2]。事实上，目前在通用处理器上进行多媒体扩展时绝大多数都采用了子字单元并行这一处理方式。

4 通用处理器中多媒体扩展指令系统分析

4.1 数据类型转换与处理

有些媒体处理算法，比如视频和图像处理算法中，通常用一个字节的无符号数据来存储媒体信息，但由于8位无符号数据在进行运算时，结果可能会发生溢出，为了避免运算过程中有用数据发生丢失，在进行8位无符号数运算之前，要对参与运算的8位数据进行扩充，把它们从8位提升到16位，因此，数据提升指令在多媒体处理中是非常有用的，一般多媒体扩展中都提供了相应的数据类型提升指令。

SIMD 指令是在多个数据元素上执行相同的操作。因为寄存器里的数据是同等地被处理的，所以在寄存器里或寄存器之间进行有效的数据字节重排序就成了提高多媒体处理部件性能的关键因素。与此相关的指令也可称之为“数据通信”指令。交织(Interleave)或合并(merge)指令(也有的称混合(mixing)或者紧缩(packing)指令)可以把两个源寄存器里的上半部或者下半部里的元素进行交替合并。插入操作允许一个标量值插入到寄存器里的指定位置，而提取操作则允许从寄存器中提取出特定的紧缩元素作为标量来使用。混洗(shuffle)指令可以有几种固定的数据通信模式，如果要增加它的灵活性，则需要再增加一个源寄存器(第三源寄存器)或者使用立即数寻址方式来指定那些特殊的数据通信模式[17]，如 MDMX 的混洗指令采用的就是这种三地址寻址方式。

4.2 算术与逻辑操作

4.2.1 *加法和减法* 和许多通用计算情况一样，很多图像和视频等多媒体信息都是运用整数加法和减法来进行处理的，只是在多媒体处理中数据的精度比较低：16位甚至只有8

位。因此，并行子字单元加(Padd)和并行子字单元减(Psub)就成了给32位或64位通用处理器进行多媒体扩展的头两条指令[2]。

4.2.2 溢出与饱和算法 因为参与并行运算的子字单元的位数比较低，所以在运算过程中更容易产生溢出，在单条Padd或Psub指令的执行中，可能会同时产生多个溢出，因此，如果溢出处理不当的话，就会降低因子字单元并行执行而获得的性能的提高。在多媒体处理中，对溢出的处理可以有两种方法：一种是环绕(Wraround)算法，这是一种传统的运算处理方法。在环绕算法中，上溢或下溢的结果都作截位处理，只截取并返回结果的低位，而忽略进位和借位。另一种方法是饱和(Saturation)算法。在饱和算法中，上溢和下溢的操作结果饱和到数据类型规定的数据范围内，大于数据类型范围的操作结果就饱和为该范围内的最大值，而小于数据类型范围的操作结果就饱和为该范围内的最小值。

一般通用处理器的多媒体扩展体系结构均提供了环绕算法，此外，MAX、MMX、MIPS、Altivec等提供了饱和算法，但像VIS这种没有提供饱和算法的体系结构，为避免意想不到的溢出发生造成潜在的不良后果，需要有相应的溢出处理措施，如用软件的方式来进行显式的溢出检测和处理，但这样做和直接通过饱和算法来处理溢出比起来，会导致性能下降[2]。

4.2.3 派生加法 通过重用可拆分的补码加法器可以派生出许多有用的并行加法指令，比如说求两个数的平均，通常的做法是先把这两个数加起来，然后再除以2，也可以不用除法，即在完成两个数的加法之后，再对所得结果进行一位右移，这一位的右移，可以把结果操作数中已经溢出的最重要的那一位移入最后的结果中，而把原结果中最不重要的最低位移出，通过把加法和移位这两个操作组合起来放在一条指令里，不仅仅是可以提高机器的性能，更重要的是可以避免组合操作发生溢出。

另一类派生加法是并行移位加指令，这类指令所做的操作是先对其中一个操作数进行一位或几位的移位操作后，再与另一个操作数相加，这类操作在整数乘法操作中一个操作数为常数时非常有用。此类派生加法在PA-RISC中得到了很好的应用。

还有一类有用的派生加法是求平均值指令，这类指令在视频编码的运动估计中非常有用。MAX最先引入了这一指令，但其后的VIS、MMX等并没有提供这样的指令，Intel也只是从SSE开始才提供这样的指令。

4.2.4 乘法和除法 在不同的多媒体指令集扩展中，乘法的实现方式会有很大的不同。乘法所面临的实际问题是：一个整数乘法器所占据的芯片面积通常是一个整数加法器的3－4倍，延时通常是加法器的3倍，而所得结果比原操作数都要长。

在有些多媒体指令扩展里，如VIS、MMX等都设计了新的16位整数乘法硬件电路。为了减少芯片的面积，VIS把16位的子字单元并行乘法转化为16×8的乘法，因此，要完成一个16×16的乘法需要2条16×8的乘法指令和一条32位的加法指令。

在MMX中，为了减少芯片面积，只实现了一个16×16的整数乘法器，并行子字单元乘法指令中子字单元的乘法是通过迭代使用16×16整数乘法器来实现的。16×16乘法完成后，可以取低16位也可取高16位作为乘法的结果。MMX还提供了并行子字单元乘加指令，这条指令可以先执行4个16×16的乘法，然后把前两个和后两个乘法结果加起来，得到两个32位的结果。

在有些多媒体扩展，如MAX中，并没有增加16位乘法硬件电路，这主要是考虑到在多媒体应用中媒体计算一般有两类：一类是精度超过16位的需要全功能乘法部件来执行的情况，另一类是精度低于16位，并且常常是乘以某个常数的情况。所谓全功能乘法，不仅意味着是两个变量相乘，而且还需要有乘加功能。声音和图形变换处理属于第一类，而图像、视频的处理属于第二类。对于第一类多媒体计算来说，MAX采取用其浮点处理部件的单精度来实现的方案，对于第二类多媒体计算来说，可以通过并行移位加指令来实现[2]。

既然图形的缩放和转换通常要用除法，有的多媒体扩展如SSE直接提供了除法指令，而有的多媒体扩展则提供了计算倒数($1/x$)和平方根($\sqrt{x}$)的指令，如所有的浮点SIMD多媒体扩展(如AMD的3Dnow!，Intel的SSE和Motorola的Altivec)都提供了计算$1/x$和$\sqrt{x}$近似值的指令[17]。

4.2.5 移位操作 绝大多数多媒体扩展如Intel、MMX等都提供了紧缩移位操作指令，包括算术/逻辑左移、逻辑右移与算术右移三种移位情况，子字单元的大小取值最多有：8、16、32、64位四种情况，一般机型均没有提供全部四种取值的移位情况，多的提供了其中的三种，少的则只提供了其中的一种，如HP就只提供了子字单元为16位的移位指令，而Motorola则提供了8、16、32位三种子字单元的移位指令。

4.2.6 位逻辑操作 除MVI与MAX外的多媒体扩展都提供了紧缩位逻辑操作指令，紧缩位逻辑操作与非紧缩位逻辑操作在实现上没有本质上的区别，因为进行位逻辑操作的子字单元不存在进位或借位的问题。

4.3 条件执行

比较和条件操作的高效执行对SIMD处理所带来的性能提升起着至关重要的作用。

同时比较多个操作数无疑会产生多个比较结果，每一个比较结果在概念上是一个逻辑值真或假。在子字单元并行里有效利用比较结果信息的方法有两种：第一种是采用屏蔽的方法，第二种是采用位方法。在第一种方法里，与通用处理器体系结构的标准比较指令是为后续的分支或者条件转移等提供条件码不同，SIMD处理会对各个项比较出来的逻辑结果建立对应的数值以便能对其进行进一步的操作：产生一个和子字单元一样大的屏蔽字，屏蔽字的内容由对应项的逻辑比较结果复制而来(即屏蔽字的内容要么是全0，要么是全1)。采用屏蔽字方法可以消除条件执行语句，因为我们可以利用屏蔽字只选择字中的部分元素参与运算并获得结果，而这种处理方式的实现所需付出的代价仅仅是适度的数据路径增长，因此它是一种非常高效的比较结果处理方式。第二种方法是把比较的结果紧缩并把它们存储到一个通用寄存器里。这种方式往往用来进行条件存储，也即一个字单元里的内容将按照存储在寄存器里的位图进行部分子字单元存储[18]。在条件执行中，除VIS采用位处理方式外，其余的多媒体扩展均采用了屏蔽处理方式。

还有一类特殊的比较形式是最大值(max)和最小值(min)的计算。因为很多多媒体算法中都涉及到了求最大值和最小值的操作，所以现有通用处理器的多媒体扩展里大多都提供了类似功能的指令。

4.4 存储器访问和预取

在对多媒体数据进行处理时，多媒体数据在存储器中是否恰好对齐对SIMD处理性能的提高有很大的影响，因为如果多媒体数据在存储器中是对齐的，那么对存储器的一次访问就可以取到子字单元并行方式所支持的全部子字单元项；如果多媒体数据在存储器中没有对齐，要取得一个字单元中

的全部子字单元项就需要对存储器进行多次访问，从而导致SIMD处理性能的降低。VIS、MIPS-V等提供了处理数据未对齐的指令。

特殊情况的存储器访问包括对寄存器组的块装载和块存储以及对寄存器内容进行部分存储。通用处理器的多媒体扩展中一般没有设置块装载与块存储的块操作方式，只有VIS设计了一次可以装载/存储8个寄存器(64字节)的块装载/存储指令，并且提供了把3维数组地址转换为块线性地址的指令。部分存储是指对字寄存器中的内容，以子字为单位，有选择地进行存储，具体哪些子字单元被选择进行存储则由预先计算得到的位图来控制(参看条件执行部分)。

正如我们所预见的一样，不适当的存储器层次结构会把SIMD处理所带来的性能提升给掩盖了。我们知道，通用处理器中的存储器层次结构是根据处理器速度和存储器速度之间存在的差别来进行设计的，在对通用处理器进行多媒体扩展时一般不应改变系统的存储器层次结构，然而，现有的存储器层次结构对多媒体数据的处理并不一定合适，而多媒体数据的处理又具有很强的预见性，所以，在一些多媒体扩展指令集里如AMD的3Dnow!、Intel的SSE等提供了预取指令，以使得对存储器的访问延时能降到最低。

4.5 特殊指令

在很多媒体处理算法中，一般不需要用到很复杂的操作，但在一些特殊的媒体处理算法中，某些复杂的操作则可能成为影响算法性能的瓶颈，比如说在MPEG视频压缩算法中对两个宏块中的像素绝对差值之和的计算，为了提高通用处理器对运动视频压缩编码的性能，VIS、SSE、增强3DNow!等提供了相应的计算多个绝对差值之和的指令。

UltraSparc的VIS提供了矩阵指令以支持3维矩阵的计算，它可以产生装载一串以3维数组形式存储在内存的数据地址。

5 在通用处理器中进行多媒体功能扩展的发展趋势

由于我们生活在一个丰富多彩的多媒体世界里，强大的多媒体处理能力是通用处理器的必然追求，因此在通用处理器的设计中必须充分考虑如何高效支持多媒体信息处理，从已有的通用处理器中进行的多媒体功能扩展我们可以看到这样一些发展趋势：

·因为多媒体处理能力成了对通用处理器的必然要求，所以通用处理器中相应的多媒体处理指令不是可有可无的，而是必不可少的，而且不再被称为多媒体扩展指令，而是直接称其为多媒体处理指令或多媒体指令[19]。

·多媒体处理能力全面增强，除了要高效支持整数子字单元并行处理外，还要高效支持浮点并行处理[6~13,19]，多媒体指令在整个机器指令系统中占的份额越来越大。

·数据并行处理能力不断增强，整数SIMD并行处理的宽度会由64位提高到128位，同时，把SIMD与传统的向量处理机制相结合也会大大增强对多媒体数据的并行处理能力[20]。

·除了充分考虑多媒体数据的并行处理外，还要充分考虑多媒体任务的并行处理[15,21]，从而开发出多线程或超线程通用多媒体处理器。

结束语 本文首先对通用处理器中的多媒体扩展进行了回顾，接着分析了多媒体处理器的特点，指出了在现有通用处理器中进行多媒体扩展的共同特征是引入了小规模的SIMD并行处理模式，然后从指令集的角度对现有通用处理器的多媒体扩展指令系统进行了分析，最后分析了在现有通用处理器中进行多媒体功能扩展的发展特点，指出发展多线程或超线程多媒体处理器将是通用处理器发展的必然趋势。

参考文献

1 NOMADIK(TM)Open Multimedia Platform for Next-generation Mobile Devices. http://eu.st.com/stonline/books/ascii/docs/9306.htm

2 Lee R B. Multimedia extensions for general-purpose processors. In: Proc. IEEE Workship on Signal Processing Systems -Design and Implementation(SIPS 97),97:9~23

3 Lee R. Accelerating Multimedia with Enhanced Microprocessors. IEEE Micro,1995,15(2):22~32

4 Trembley M,O'Connor M,Narayanan V,He L. VIS Speeds New Media Processing. IEEE Micro,1996,16(4):10~20

5 Lee R. Subword Parallelism with MAX2. IEEE Micro,1996,16(4):51~59

6 Ninth Annual Microprocessor Forum,San Jose,California,1996

7 Peleg A,Weiser U. MMX Technology Extension to the Intel Architecture. IEEE Micro,1996,16(4):42~50

8 MIPS Technologies Inc. MIPS Extensions for Digital Media with 3D,whitepaper,March 1997. http://www.mips.com/

9 Thakkar S,Huff T. Internet Streaming SIMD Extensions. Computer,1999,32(12):26~34

10 Tiler J,et al. AltiVec™:Bringing vector technology to the PowerPC™ processor family. IEEE Int. Conf. on Performance,Computing and Communications,1999. 437~444

11 MIPS Technologies,Inc. Information Backgrounder R1.0. MIPS-3D technology enables low-cost 3D graphics acceleration for embedded MIPS RISC architecture-based systems. http://www.embeddedinsight.com/pdf/MIPS3Dback.pdf

12 Enhanced 3DNow!™ Technology for the AMD Athlon™ Processor. http://www3pub.amd.com/products/cpg/athlon/3dnow_wp.html

13 The Intel® Pentium® 4 processor product overview. http://www.intel.com/design/Pentium4/prodbref/#streaming

14 Slingerland N T,Smith A J. Multimedia Instruction sets for general purpose microprocessors:A Survey. University of California at Berkeley:[Technical Report No. UCB/CSD-00-1124]. Dec. 2000

15 Corbal J,et al. DLP+TLP processors for the next generation of media workloads. In:Proc. 7th Intl. Symp. on HPCA,2001

16 Ferretti M. Multimedia extensions in super-pipelined micro-architecture. A new case for SIMD processing?In:Proc. of Fifth IEEE intl. Workshop on Computer Architectures for Machine Perception,2000. 249~258

17 Slingerland N T,Smith A J. Performance analysis of instruction set architecture extensions for multimedia. In:Proc. of the 3rd Workshop on Media and Stream Processors,Austin,Texas,Dec. 2001

18 Ferretti M. Multi-media extensions in super-pipelined micro-architectures. A new case for SIMD processing? In:Proc. of Fifth IEEE Intl. Workshop on Computer Architectures for Machine Perception,2000. 249~258

19 Lee R B,Fiskiran M,Bubshait A. Multimedia Instructions in IA-64. In:Proc. of the 2001 IEEE Intl. Conf. on Multimedia and Expro (ICME 2001),Aug. 2001. 281~284

20 Corba J,et al. Exploiting a new level of DLP in multimedia applications. In:Proc. Intl. Symp. on Microarchitecture,Nov. 1999

21 Oehring H,Sigmund U,Ungerer T. MPEG-2 video decompression on simultaneous multithreadied multimedia processors. PACT'99,Oct. 1999