

DSP 处理器和通用处理器的比较^{*}

岳虹 沈立 戴葵 王志英

(国防科学技术大学计算机学院 长沙410073)

摘要 随着嵌入式系统的广泛应用,其应用程序的功能变得越来越强大和复杂,从而要求嵌入式处理器系统既能有效支持运算密集型的应用,又能有效支持控制密集型的应用。数字信号处理器(DSPs)能够有效进行运算密集型的实时计算;另一方面,通用微处理器(GPPs)则对控制密集型的应用提供有效的支持。本文从DSP处理器和通用微处理器的功能出发,讨论了两者在指令集、体系结构及存储器结构等方面的异同,同时对两者的性能也进行了评测和比较。结果表明,DSP处理器和通用微处理器都很难同时高效支持运算密集型的应用和控制密集型的应用。将两者体系结构进行融合,研究开发融合型高性能微处理器,是解决该问题的有效途径。

关键词 数字信号处理器,通用微处理器,指令集,体系结构

The Comparison between Digital Signal Processor and General-Purpose Processor

YUE Hong SHEN Li DAI Kui WANG Zhi-Ying

(School of Computer, National University of Defense Technology, Changsha 410073)

Abstract With the wide spreaded applications of embedded system, the power and complexity of functionality implemented in embedded devices is increasing. Thus it is necessary for embedded processor to support both arithmetic-intensive and control-intensive computations. The Digital Signal Processors (DSPs) can effectively support the arithmetic-intensive and real-time computations; on the other hand, General-Purpose Processors (GPPs) are suitable for a variety of applications which are control-intensive. This paper is to discuss the differences between DSPs and GPPs in instruction set, architecture type and memory architecture according to their different functions. Performance evaluation then is done to prove it. Preliminary results indicate that either DSPs or GPPs can effectively support both arithmetic-intensive and control-intensive computations. Combining them in architecture and developing a hybrid-architecture processors is necessary for meeting the requirements of current embedded systems.

Keywords DSP, General-purpose processor, Architecture, Instruction set

1 引言

嵌入式系统常常采用专门的数字信号处理器(Digital Signal Processor, DSPs)实现快速实时的数字信号处理。数字信号处理器能够对实时的运算密集型应用提供有效的支持。

随着嵌入式系统应用的飞速发展,其功能变得越来越强大和复杂。目前的嵌入式应用程序不仅需要进行数字信号处理,也需要进行大量的复杂控制类函数的处理。通用处理器(General-purpose Processors, GPPs)能够有效支持这些非DSP类的控制密集型应用,也适用于嵌入式系统。

但是,DSPs和GPPs指令集和体系结构等各方面的特点,决定了它们只能有效支持各自擅长的应用,无法同时对运算密集型和控制密集型的应用提供有效的支持。本文详细分析了二者的功能和结构特点,并针对一组基准程序比较了它们的性能。比较结果说明,在当前的嵌入式系统中,单纯选择DSPs或者GPPs都不能较好地满足其高性能、低价格、低功耗的需求。将两者的体系结构进行有效的融合,研究开发既能高效支持运算密集型应用,又能高效支持控制密集型应用的融合型高性能处理器,是支持当前越来越复杂的嵌入式应用

的有效途径。

本文第2节从指令集结构、体系结构和存储器结构三方面分析了它们的异同;第3节对其性能进行了评测和比较;最后给出了结论和今后研究的方向。

2 DSPs和GPPs的比较

2.1 功能特点

数字信号处理算法的特征决定了DSP处理器必须具备的功能。数字信号处理需要进行大量的实时运算,大运算量决定了DSPs必须具有多个可并行执行的功能单元,并对常用的DSP操作提供直接的硬件加速支持;强实时性决定了DSPs不能具有太多的动态特征,因此对Cache机制、分支预测机制和中断响应机制的选择或采用,具有一定的限制。在运算上,DSP算法具有程序短小、包含大量简单运算(如MAC运算)和短小循环、数据运算高度重复的特点,因此应在DSPs中设置片内存储器、单周期MAC功能单元和硬件“零开销循环”控制机制,对DSP算法的需求提供必要的支持。在DSP算法处理的过程中,需要进行频繁的数据访问,以满足运算实时、快速的要求。数据地址的计算会随着访问的频繁度而线性

^{*} 本课题受自然科学基金支持(60173040)。岳虹 博士研究生,主要研究方向:计算机体系结构、高性能微处理器设计。沈立 博士,讲师,主要研究方向:计算机体系结构、VLIW体系结构与编译设计。戴葵 博士,副教授,主要从事先进计算机体系结构、微处理器设计技术研究。王志英 博士生导师,教授,主要从事先进计算机体系结构、微处理器设计技术研究。

增长。这样,数据地址计算的速度和数据访问的速度就对 DSPs 的性能产生很大的影响。因此,应在 DSPs 中设计高的存储器带宽、专门的地址计算单元和专用的寻址模式,用以对高效的数据访问提供必要的支持。

和 DSPs 相比,GPPs 具有很强的通用性。GPPs 是为通用计算机的广泛应用而设计的,其运行的程序对运算量和实时性的需求都不像 DSPs 那样严格。GPPs 追求各种应用程序执行性能的平衡,以达到总体性能的提升。因此,能够提高总体性能的各种技术或者机制,GPPs 都会在考虑高性能、低价格、低功耗的前提下,有选择性的采纳,而不会受到应用程序的具体限制。因此,GPPs 对于控制密集型应用可以提供高效的支持。

DSPs 的功能特点决定了 DSPs 在指令集、体系结构、存储器结构以及相应的程序控制机制等方面具有其独特的特点。

2.2 指令集比较

DSPs 指令集的特点在于能够充分利用处理器的硬件资源,并尽量减小 DSP 程序代码所占用的存储空间。在 DSPs 指令集的设计中,通常在一条指令中指定多个并行操作。例如,在 TMS320C50 中,一条 MAC 指令中同时包含了乘法操作、加法操作和存储器取数操作。另外,在设计时也使指令尽量短小,或者一条指令包含非常丰富且专用的功能。通过限制特定操作使用特定寄存器以及使用模式控制位来控制指令操作,可以有限缩短指令长度;而采用 CISC 架构指令,则能在一个时钟周期内完成由多条 RISC 架构指令完成的任务。比如取模指令、取倒数指令。

受这些因素的影响,早期的 DSPs 的指令集非常专用、复杂并且不规则。这必然导致程序员使用的汇编语言复杂程度的增大,从而导致编程及编译的困难。同时,由于高级语言(比如 C 语言)通常并不适合编写 DSP 程序,DSP 编译器也很难将高级语言程序编译成高效的 DSP 汇编代码,因此高级语言的使用也受到了很大的限制。为了克服这些缺陷,在当前高性能的 DSPs 指令集中,已经适当地减小了指令的复杂度,让单条指令功能更加简单,这样虽然增大了代码量,但是指令集已经相对简单规整,从而有效支持高级语言的编程,并且在一定程度上降低了编译器工作的难度。

与 DSPs 相比,GPPs 的指令集的优势在于具有更简洁的结构和更大的通用性,对编译技术也提供了较好的支持。但其程序所占用的存储器空间却比 DSPs 程序占用的空间大。早期的 GPPs 通常采用 CISC 指令集结构,但 CISC 指令集的复杂性和功能的不均衡性带来了体系结构的复杂性,也妨碍了先进技术(如流水技术)的采用。为了克服这些缺点,后期的 GPPs 广泛采用了 RISC 指令集结构。RISC 指令集结构简洁规整,单条指令通常在单个周期内完成单个操作。这样在体系结构复杂性降低的同时也降低了 CPI,提高了性能。无论是 CISC 指令集,还是 RISC 指令集,其指令功能都具有通用性,从而能够有效地支持高级语言的使用,并具有成熟高效的编译技术。

DSPs 和 GPPs 在寻址模式的采用上,也存在着不小的差异。除了使用与 GPPs 相同的一些寻址模式外,DSPs 还采用了一些专门的寻址模式。其中比较具有代表性的寻址方式是环状寻址(Circular Addressing)和位逆寻址(Bit-Reverse Addressing)。环状寻址对于实现数字滤波器延时线很有用;而位逆寻址则适合于 FFT 算法。为了支持这些寻址模式,同时提

高数据地址计算速度,DSPs 设置了专用的地址计算单元,而 GPPs 一般没有设置单独的地址计算单元。

2.3 体系结构比较

DSPs 和 GPPs 在体系结构方面的不同,首先体现在功能单元的采用上。为了满足 DSP 算法快速实时的运算需求,DSPs 除了设置和 GPPs 一样的 ALU、累加器等功能部件外,还设置了乘法单元(乘加单元)、地址产生单元等特有的功能单元。并且常用的功能单元会重复设置多个。同时,设置哈佛总线结构,即分离的程序数据总线,甚至多路数据总线为这些功能单元同时提供数据。这种设置多个功能单元和多路总线的方式,可以充分利用硬件资源,满足单指令单周期多操作的要求,提高处理器的性能。在高性能的 DSPs 中,通常设置6个以上的运算单元。

早期采用 RISC 结构的 GPPs,一条指令通常只进行一个操作,所以通常设置单一的 ALU 单元。很多操作,比如乘法操作需要多个时钟周期才能完成。随着生产工艺的发展和技术的进步,高性能的 GPPs 也在芯片内集成多个运算单元,并设置硬件乘法单元或者乘加单元。单流出的 GPPs 采用了冯·诺伊曼总线结构,即统一的程序和数据空间以及共享的程序和数据总线。

DSPs 和 GPPs 在体系结构方面的第二个不同,体现在多指令流出技术的采用上。DSPs 和 GPPs 在发展中都采用了多指令流出的并行技术。多指令流出技术是指通过软件或者硬件的调度,在一个时钟周期内发射多条指令,并分配到多个功能单元中去处理。在调度合理和功能单元充足的基础上,同时流出的指令越多,性能优势越显著。超长指令字(VLIW)结构的采用是一种典型的多指令流出技术。它将指令并行调度的工作交给软件来执行,因此在程序运行的过程中指令序列并不改动。这种执行的确定性非常符合 DSP 应用程序的需求,因此高性能的 DSPs 通常采用 VLIW 的体系结构,每个时钟周期通常流出4~8条指令。比如 TMS320C62xx、SC140 和 TigerSHARC 等处理器。图1给出了基于 TMS320C62xx 处理器的 VLIW 体系结构示意图。

采用 VLIW 结构在获得高性能的同时,硬件复杂度没有得到显著的增加,也没有带来很多的动态行为,对于需要实时确定性运算的 DSPs 来说,具有很大的优势。同时也具有很好的可扩展性,可以根据具体需求增加或者删除并行的功能单元。VLIW 结构也带来了相应的缺陷。因为很多指令调度工作交给软件执行,所以编译的复杂度大大提高,同时也不能保证二进制代码的兼容性。程序存储器空间需求的增大,对存储器的带宽也提出了更高的要求。

相比而言,在多指令流出技术的采用上,GPPs 比 DSPs 具有更大的灵活性。大多数的高性能 GPPs 通常采用 Super-Scalar 的多流出技术,每个时钟周期通常流出2~4条指令。例如 Pentium III 和 PowerPC 74xx 处理器。Superscalar 技术的指令并行调度工作由硬件调度执行,因此在大幅度提高性能的同时保证了二进制代码的兼容性和对代码的透明性,代码量也不会有明显增加。这样就降低了软件的复杂度,同时不可避免地带来了硬件复杂度的增加,并且加入了动态行为。

2.4 存储器结构比较

如前所述,DSPs 为了提高并行性设置了哈佛总线结构,使取指和取数据同步进行。为了和哈佛总线结构相适应,DSPs 设置了同样的存储器结构:分离的程序存储器和(多个)数据存储器,允许 CPU 核同时对它们进行访问。早期的

GPPs 通常采用冯·诺依曼存储器结构,即统一的程序数据存储器空间通过一组总线(一条地址总线和一条数据总线)连接到处理器核。但这种存储器结构不能同步进行指令和数据的访问,因而对性能造成很大的制约。所以高性能的 GPPs 通常也会采用哈佛总线结构和存储器结构。

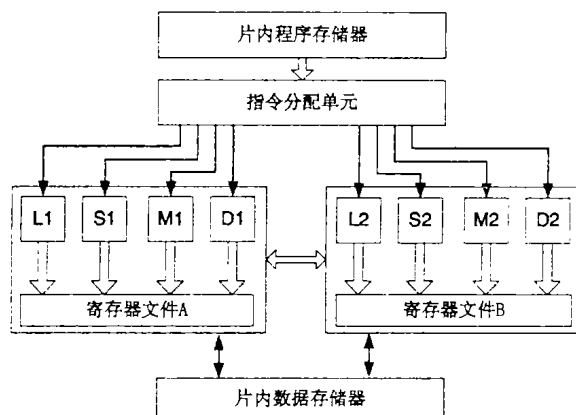


图1 VILW 结构的 DSP 处理器示意图

DSPs 设置了片内存储器。将程序放在片内程序存储器中,可以明显减少程序访问时间,并有效缓解芯片外部总线接口的压力;片内数据存储器不存在和外部存储器的总线竞争问题和访问速度不匹配的问题,因此访问速度快,可以缓解 DSP 处理器的数据访问瓶颈,充分利用 DSP 处理器强大的处理能力。而 GPPs 运行的程序一般都很大,不可能放置于空间有限的片内存储器中。因此 GPPs 通常不设置片内存储器。GPPs 减小处理器速度和存储器访问速度之间差距的典型方法是在 GPPs 内部设置高速缓存(Cache)。根据程序访问的局部性原理,GPPs 使用相应的控制逻辑来决定哪些数据和指令存储在 Cache 中,对程序员来说是透明的。而在使用 DSP 处理器时,程序员要明确地控制哪些数据和指令存储在片内存储器中。程序员在写程序时,必须保证处理器能够有效地使用其双总线。所以 Cache 的动态特征决定了 DSPs 采用 Cache 时,必须加入一些特殊的机制,例如在其中加入循环缓冲区域。

图2给出了 GPPs 和 DSPs 典型的存储器结构。

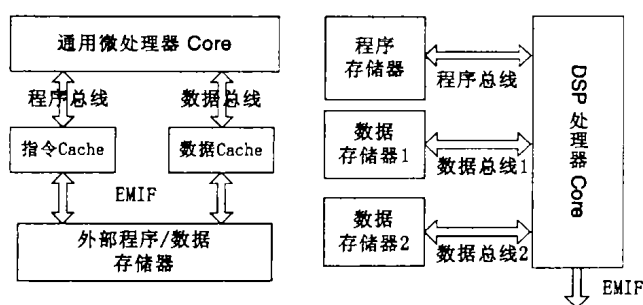


图2 通用微处理器和 DSP 处理器存储器结构比较

3 性能评测与结果

因为大多数的 GPPs 具有相似的指令集,因此对 GPPs 性能进行衡量的主要标准是时钟频率,并且通常由 MIPS、MFLOPS 或 MOPS 来表现。而不同 DSPs 的指令集差别很大,同样的操作由不同的 DSPs 来执行,可能需要不同数量的指令。所以由 MIPS 等作为 DSPs 性能的衡量标准,往往不能

准确反映 DSPs 的实际性能。同时,存储器的访问带宽和所占存储器空间的大小也是评价 DSPs 的重要因素。DSPs 独特的功能和结构特点决定了对 DSPs 性能进行评测时,不能单纯以时钟频率为标准,而应该考虑其每个时钟周期所进行的操作数量以及其对存储器的使用情况,这样才能反映 DSPs 的真实性能。因此,我们在对 DSPs 和 GPPs 的性能进行比较时,应该独立于各自的指令长度和数据类型,而结合各自的时钟频率和体系结构,以程序所需的时间和代码量作为衡量标准,如表1所示。

表1 衡量标准的选择

度量	定义	单位
Run time	程序运行时间	微秒(microsecond)
Code Size	代码量	字节(Byte)

同时,我们选用了四个算法,分别作为 DSP 类程序和控制类程序的代表,对 DSPs 和 GPPs 的性能进行测试。从而说明它们分别对运算密集型和密集型应用的支持程度,如表2所示。

表2 测试程序的选择

名称	类型	描述
FIR Filter	DSP	FIR 滤波器算法
FFT	DSP	64点快速傅立叶变换
Dhrystone	Control	综合性测试程序
Viterbi	Control	对信道内传输控制序列进行纠错和排错

我们选择了这样五种处理器进行性能评测,如表3所示。

表3 处理器的选择

名称	简称	类型	主频(MHz)	多指令流出技术
TMS320C5510	C55	DSPs	160	无
SC140	SC140	DSPs	300	VLIW
TMS320C64x	C64	DSPs	600	VLIW
ARM9	ARM9	GPPs	200	无
Pentium III	P III	GPPs	500	SuperScalar

对于运行时间的评测结果如图3所示。对于代码量的评测结果如图4所示。

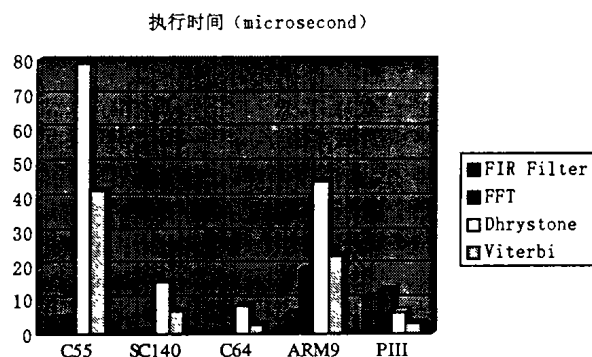


图3 运行时间的评测与比较

从图3所示的评测结果可以看出,在主频和体系结构相当的情况下,对于数字信号处理类的程序,DSPs 的性能优于 GPPs;对于控制类的程序,GPPs 的性能明显优于 DSPs。而在

(下转第206页)

参考文献

- Berners-Lee T. A roadmap to the Semantic Web. <http://www.w3.org/DesignIssues/Semantic.html> Sept. 1998
- Kosala R, Blockeel H. Web Mining Research: A Survey. ACM-SIGKDD, July 2000
- Zaiane O R, et al. Multimediaminer: a system prototype for multimedia data mining. In: Proc. ACM SIGMOD Intl. conf. on Management of Data, 1998. 581~583
- Chakrabarti S, et al. Mining the link structure of the world wide web. IEEE Computer, 1999, 32(8): 60~67
- Cooley R, Mobasher B, Srivastava J. Web mining: Information and pattern discovery on the world wide web. In: proc. of the 9th IEEE Intl. Conf. on Tools with Artificial Intelligence (ICTAI'97), 1997
- Bray T, et al. Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation. Oct. 2000. <http://www.w3.org/TR/2000/REC-xml-20001006>
- Milo T, Zohar S. Using schema matching to simplify heterogeneous data translation. In: proc. 24th Int Conf on Very Large Data Bases, 1998. 122~133
- Andersen E S, Valente D M. the art of simulation and the Lsd system, ETIC Session, Strasbourg, 2003
- <http://www.nalasoftware.com/DOCS/Xmapper/overview.cfm>
- Decker S, et al. The Semantic Web: The Roles of XML and RDF. IEEE INTERNET COMPUTING 1089-7801/00/\$10.00 (c) 2000 IEEE, 2000, 4 (5): 63~74
- Gruber T R. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 1993, 5: 199~220
- Borst W N. Construction of Engineering Ontologies for Knowledge Sharing and Reuse. [PhD thesis]. University of Twente, Enschede, 1997

- 13 Studer R, Benjamins V R, Fensel D. Knowledge Engineering, Principles and Methods. Data and Knowledge Engineering, 1998, 25(1-2): 161~197
- 14 Grand B L, Soto M, Dodds D. XML Topic Maps and Semantic Web Mining
- 15 Hotho A, Maedche A, Staab S. Ontology-based text clustering. In: Proc. of the IJCAI-2001 Workshop "Text Learning: Beyond Supervision", August, Seattle, USA, 2001
- 16 Chakrabarti S, et al. Automatic resource compilation by analyzing hyperlink structure and associated text. In: Proc. of the 7th World-wide web conf. (WWW7), 1998, 30(1-7): 65~74
- 17 Chakrabarti S, van den Berg M, Dom B. Focused crawling: A new approach to topic-specific web resource discovery. In: Proc. of the 8th World-wide web conf. (WWW8), 31 (11-16), Toronto, May 1999. 1623~1640
- 18 Maedche A, et al. Ontology-focused crawling of documents and relational metadata. In: Proc. of the Eleventh Intl. World Wide Web Conf. WWW-2002, Hawaii, 2002
- 19 Ehrig M. Alexander Maedche Ontology-Focused Crawling of Web Documents
- 20 Spiliopoulou M, Pohle C. Data mining for measuring and improving the success of web sites. Data Mining and Knowledge Discovery, 2001, 5
- 21 Parent S, Mobasher B, Lytinen S. An adaptive agent for web exploration based on concept hierarchies. In: Proc. of the 9th Intl. Conf. on Human Computer Interaction, New Orleans, LA, 2001
- 22 Hotho A, Maedche A, Staab S, Studer R. SEAL-II - the soft spot between richly structured and unstructured knowledge. Journal of Universal Computer Science (J. UCS), 2001, 7(7): 566~590

(上接第168页)

DSPs 间或在 GPPs 间,则以主频的高低决定性能的优劣。GPPs 在 DSP 类程序上的劣势稍低,是因为 GPPs 为了满足嵌入式领域的发展需求,而在指令集或者硬件结构上进行了增强以对 DSP 算法提供支持。主频高的 DSPs 性能比主频低的 GPPs 性能稍好,说明所采用的体系结构和主频的高低对性能的优劣也起着一定的作用。

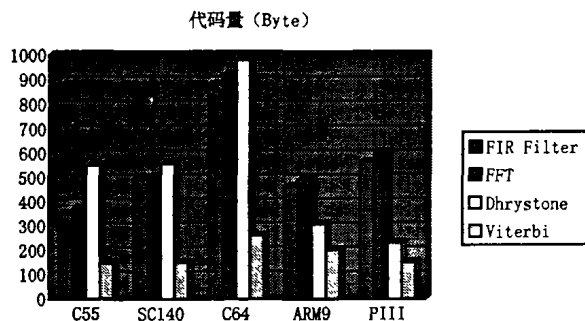


图4 代码量的评测与比较

从图4代码量的测试中可以看出,处理器的结构对于代码量的大小起着决定性的作用。VLIW 结构的采用,增加了指令长度,从而增大了代码量。Superscalar 结构的采用也适度增加了代码量。代码量的大小与编译器的选择有关。从图中可以看出,对于 DSP 类程序, DSPs 的代码量小于 GPPs; 对于控制类程序, GPPs 的代码量小于 DSPs。这是因为针对 DSPs 的编译器在代码优化和调度时,其算法倾向于最大限度地使指令并行,并最大限度地利用 DSPs 的硬件资源。控制类程序不具备 DSP 算法的特点,编译器在对代码进行分析时,其优化算法和调度算法无法发挥最大效能,从而不能较好地对代码进行优化和调度,从而代码量较大。对于 GPPs 亦然。

结束语 综上所述,由于 GPPs 和 DSPs 面向的应用不同,所具有的功能不同,从而导致了 GPPs 和 DSPs 从指令集结构、体系结构到存储器结构都具有显著的不同。DSPs 在对运算密集型应用的支持上,从性能到代码量的大小,都具有明显的优势,但是对控制密集型应用的支持则比较有限;而 GPPs 则对控制密集型应用提供良好的支持,对于运算密集

型应用的支持不如 DSPs。

随着应用的推动和处理器体系结构研究的发展, GPPs 开始考虑在有效支持控制密集型应用的同时,也能够有效支持运算密集型的应用。高性能通用微处理器开始借用 DSP 处理器的结构优点,并在硬件上通过协处理器或者增加 DSP 功能单元的方式对 DSP 功能提供支持。但由于 GPPs 缺乏实时可预测性,存储器带宽有限、优化 DSP 代码困难,有限的 DSP 工具支持,高功耗等问题,因此单纯将 GPPs 应用于 DSP 领域,功能和性能都不能满足要求。而 DSP 处理器也不能单纯地局限于数字信号的处理,而也应具有高效的控制能力。但其独特的功能和结构,对非 DSP 任务进行支持的第三方工具的缺乏决定了它对控制密集型应用的支持非常有限。将通用微处理器的体系结构和 DSP 处理器体系结构进行有效的融合,生成融合型的高性能微处理器,成为嵌入式处理器发展的主流趋势。在嵌入式应用领域中,将高性能嵌入式通用微处理器和 DSP 处理器进行融合,形成专用的融合型嵌入式微处理器,能够同时有效支持控制密集型的应用和运算密集型的应用,具有广泛的应用前景。

参考文献

- BDTI, Microprocessors vs. DSPs: Fundamentals and Distinctions, 2004
- Patterson D A, Hennessy J L. Computer Architecture: A Quantitative Approach. 2nd ed. San Francisco: Morgan Kaufman Publish, 1996
- 张晨曦, 王志英, 等. 计算机体系结构. 高等教育出版社, 2000
- Cuadrado D L, et al. A platform-based comparison between a digital signal processor and a general-purpose processor from an embedded systems perspective. Embedded Systems Group, CPK, Aalborg University, 2002
- Frederisen A, Christiansen R, Bier J, Koch P. An Evaluation of Compiler-Processor Interaction for DSP Applications. In: Proc. 34th IEEE Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA, USA, Oct. 2000
- 任丽香, 马淑芬, 李方慧. TMS320C6000 系列 DSP 处理器的原理和应用. 电子工业出版社, 2000
- 钟文政, 柯鸿禧. DSP TMS320C50 原理与应用. 中国水利水电出版社, 2003
- Frantz G. Digital signal processor trends, Texas Instruments, IEEE Micro, 2000, 20(6): 52~59
- Eyre J, Bier J. The Evolution of DSP Processors, BDTI a BDTI White Paper, 2000. 1~9
- <http://www.DSPsolution.com/html.intro/intro-dsp.htm>
- <http://www.embed.com.cn/forum/show.asp?boardID=12&announceID=3147>