嵌入式系统译丛

嵌入式实时系统的DSP软件开发技术

DSP Software Development Techniques for Embedded and Real-Time Systems

[美] Robert Oshana 著
郑 红 刘振强 王 鹏 译

北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

嵌入式系统译丛

嵌入式实时系统的DSP软件开发技术

DSP Software Development Techniques for Embedded and Real-Time Systems

[美] Robert Oshana 著

郑 红 刘振强 王 鹏 译

北京航空航天大学出版社

译者序

《DSP Software Development Techniques for Embedded and Real - Time System》是一部详细介绍DSP在嵌入式实时系统设计中软件开发方法，以及探讨DSP软件设计技术的专业技术指南。本书的作者Robert Oshana是美国德州仪器(TI)公司DSP系统部软件开发的工程管理人员，具有超过20多年的实时嵌入式软件设计开发经验。他也是Southern Methodist University的兼职教授，教授研究生软件工程以及嵌入式实时系统课程。

本书的特点是从DSP与嵌入式系统的内在联系着手，介绍实时DSP软件开发技术。全书由浅入深、从理论到实践地探讨了嵌入式实时DSP系统软件开发过程的各种问题，使读者对于DSP用于实时嵌入式系统的软件技术形成一个完整的概念：从数字信号处理理论到DSP常用信号处理算法软件开发技术；从嵌入式系统特点到嵌入式DSP系统涉及的硬件结构与软件性能的相互制约关系；从一般开发步骤介绍，到DSP软件设计优化方法实现；不仅突出了DSP软件技术开发难点，也给出了相应的解决方案以及大量实例，并综述了DSP软件技术发展前景。全书的体系结构合理，不同于国内有关DSP方面的图书主要介绍硬件系统及其应用为主的结构体系，而是以软件开发技术为重心，对于DSP软件技术研究及开发具有很好的指导作用。

我们从事嵌入式实时DSP系统应用研究多年，研究中发现软件问题是系统性能提高的关键，这本书也给与我们启发良多，因此，翻译出来与同好共享。

本书的翻译工作主要由北京航空航天大学DSP实验室的教师和研究生共同参与完成，除第一作者外，翻译者包括刘振强、王鹏、杜佳颖、方能辉、陈磊、刘钊江、赵振华、李文庆及李香祯等。正是他们的辛勤努力，才使得本书的翻译工作得以在这么短的时间内完成。在此，对他们的工作深表谢意。

译　者

2010年秋于

北京航空航天大学

致　谢

此书的编写得到了来自家人、朋友和同事在技术和感情上的重要支持。这里不可能将所有帮助支持过我完成整个项目的人全部列出，若有遗漏，深表歉意。

我的编辑工作人员众多。Tiffany Gasbarrini，很高兴与你共事，能与你共同努力是我的荣幸。Carol Lewis，我不会忘记你。感谢 Elsevier 公司的扶持。感谢 Borrego 出版社 Kelly Johnson 的所有努力工作和支持。

感谢 Frank Coyle，作为我在南卫理工会大学的学术和私人顾问，你提供了本书最初的灵感。感谢你所做的一切。

我要感谢这些向我提供了重要资源以及支持这项工作的人：Gene Frantz、Gary Swoboda、Oliver Sohm、Scott Gary、Dennis Kertis、Bob Frankel、Leon Adams、Eric Stotzer、George Mock、Jonathan Humphreys 及 Gerald Watson，来自 TI 技术训练培训组的众多技术作者，以及我在书中引用和参考的优秀应用笔记的无名作者们。同时，特别感谢 Cathy Wicks、Suzette Harris、Lisa Ferrara、Christy Brunton 和 Sarah Gonzales 的支持、奉献和幽默。

感谢我的经理们，让我有机会在从事这项工作：Greg Delagi、David Peterman、Hasan Khan 和 Ed Morgan，谢谢你们。

感谢所有评审人。我曾努力纳入所有的反馈意见，并将继续接受任何新的反馈意见。感谢那些允许在书中使用其图片的人们，这些图片增加了素材的质量。

我还要感谢我的家人和朋友的支持和理解，这本书也耗费了他们很多时间。Susan、Sam 和 Noah，感谢你们，很高兴有你们陪伴着我。

让我们走进 DSP！

绪论：为什么要用 DSP

为了便于理解可编程数字信号处理器(DSP)的用处，我首先做一个类比，然后解释哪些特殊环境可以采用 DSP。

实际上，DSP 只是一种特殊形式的微处理器。它具有所有与微处理器相同的基本特征和部件：CPU、存储器、指令集、总线等。主要的区别就是对每个部件都进行了略微的改进，以便更有效地执行某些特定的操作，我们将马上讨论这些细节。不过，通常 DSP 具有最优化的硬件结构和指令集，用以实现高速的数字处理应用以及环境中模拟信号的快速实时处理。CPU 也有略微的自定义改动，存储器、指令集、总线等也是如此。

我想用社会做一个类比。比如，每个人都是处理器(有认知能力的处理器)，都能专门做好某些事情：工程设计、看护、财政管理等。我们接受某些(专门的)领域的训练和教育，从而能够高效地执行某些工作。当专门从事某一套任务时，我们消耗较少的能量就能完成这些任务。这与微处理器没有太大的差异。这里有成百上千的微处理器可供选择，每类微处理器都在某些专门领域中有良好表现。DSP 是专门完成高效信号处理的处理器，并且如同我们生活中的专业一样，因为 DSP 专于信号处理，它完成这一工作消耗的能量也比较少。因此，在执行信号处理任务时，DSP 比通用微处理器耗费更少的时间、能量和功率。

当你专门研究一个处理器时，要注意研究频繁使用它的专门领域。能高效完成一些从不需要的事情的东西的制作是没有意义的。专攻那些能带来巨大收益的领域吧！

在继续之前，我要给出一个简捷的摘要——作为一个数字信号处理器必须做什么？要做好的事有两件。首先，它必须擅长数学并能在一秒内完成数百万(确切的说是数十亿)次乘法和加法运算。这对于执行数字信号处理算法来说是必要的。

其次，就是保证实时性。让我们回到现实生活的例子中。最近我带孩子去看电影，到电影院以后，我们不得不排队购票。实质上，我们被置入一个待处理的队列中，在其他看

电影的人之后站成一队。如果队列保持相同的长度并且不会变得越来越长，即相同数目的顾客正在处理中并有顾客加入队列，那么这个队列就是实时的。这个人的队列可能变短或变长一点，但不会无止境的增长。回忆一下飓风丽塔靠岸而导致的休斯顿撤离，那就是一个无限增长的队列！这个队列明显不是实时的，它无限制地增长，并且系统（撤离系统）被认为是失败的。不能实时执行的实时系统是失败的。

如果队列真的很大（意思是，如果我在电影院所在的队伍真的很长），但是不增长，该系统仍可能是行不通的。如果我花了50 min来排队买到票，或者直接在买到电影票前离开（我的孩子肯定会认为这是一个失败），我可能会真的感到沮丧。实时系统也需要注意能导致系统失败的大队列。实时系统能通过以下两种方法之一处理信息（队列）：要么一次处理一个数据元素，要么缓存信息，然后处理“队列”。队列长度不能太长，否则系统会有显著的延迟，并且被认为是非实时的。

如果实时性受到干扰，那么会导致系统中止，并且必须重新启动。进一步讨论有关实时系统的两个方面。第一个概念是，每个采样周期内必须捕捉一条输入数据并发送一条输出数据。第二个概念是延迟时间。即从信号输入系统到系统输出信号之间的时间延迟。

考虑实时系统时，要谨记：过晚产生正确答案是不合适的！如果排队等候之后，我拿到了所要的电影票，并且买票的钱找零数量无误，但是电影已经开始了，那么这个体制仍然是毫无价值的（除非我晚到的时候电影刚开始）。下面再回到我们的讨论中。

那么DSP能执行哪些“特定的”操作呢？如同它的名字所述，DSP善于信号处理。“信号处理”是什么意思呢？其实，它是在数字域内处理信号的一组算法。这些算法也有对应的模拟算法，但是经证明数字化的处理更加有效。这种趋势已经存在很多年了。信号处理算法是世界上很多应用的基础模块；从手机到MP3播放器、数码像机等。综述这些算法如下表所列。

算　法	公　式
有限冲激响应滤波器	$y(n)=\sum_{k=0}^{M}a_k x(n-k)$
无限冲激响应滤波器	$y(n)=\sum_{k=0}^{M}a_k x(n-k)+\sum_{k=1}^{M}b_k y(n-k)$
卷积	$y(n)=\sum_{k=0}^{N}x(k)h(n-k)$
离散傅里叶变换	$x(k)=\sum_{n=0}^{N-1}x(n)\exp\left[-j\left(\frac{2\pi}{N}\right)nk\right]$
离散余弦变换	$F(u)=\sum_{x=0}^{N-1}c(u)f(x)\cos\left[\frac{\pi}{2N}u(2x+1)\right]$

几乎每个信号处理应用中都用到这些算法中的一个或多个。有限冲激响应(FIR, Finite Impulse Response)滤波器和无限冲激响应(IIR, Infinite Impulse Response)滤波器被用来去除被处理信号中的多余噪声;卷积算法用于寻找信号中的相似点;离散傅里叶变换被用来以更易处理的形式表示信号;而离散余弦变换多用在图像处理中。稍后我们将详细讨论这其中的一些算法,但是对于所有算法需要注意一些事情:它们都具有加法操作。在计算机领域,这相当于大量元素的累加,通过 for 循环完成。鉴于这一特点,DSP 拥有大量的累加器。DSP 还具有专门的硬件来执行 for 循环操作,因而程序员不需要再通过软件完成,那会慢很多。

以上算法还包含两个不同操作数的乘法运算。逻辑上,如果要提高这一操作的速度,会设计一个处理器来适应两个操作数如此快捷的乘法和加法运算。事实上,这就是 DSP 所做的工作,它们被设计用来支持数据组的快速乘法和加法运算,大多数在短短一个周期内即可完成。由于这些算法在 DSP 应用中非常普遍,通过优化处理器可以极大地节省执行时间。

DSP 算法中的固有结构允许它们并行分离操作。正如在现实生活中,如果我能并行地做较多事情,就能在相同时间内完成更多的工作。由此推知,信号处理算法也具有这种特征。因此,可以通过在 DSP 中置入若干互不相关(互不依赖)的执行单元来实现并行机制,并在完成这些算法的时候利用它。

DSP 也必须将某些实际的内容加入以上算法的组合中。就上述的 IIR 滤波器而言,仅仅通过观察该算法,就可以发现将上次输出反馈到当前输出的计算反馈环节。无论你何时处理反馈,始终存在一种内在的稳定问题。就像其他反馈系统一样,IIR 滤波器也能变得不稳定。不小心执行类似 IIR 滤波器的反馈系统,会导致输出振荡,而不是渐近衰减到零(首选的方式)。这个问题混杂于数字世界中,我们必须处理有限字长,这是所有数字系统的一个关键限制。我们可以通过软件中的饱和度检查,或者使用特殊指令来弱化这一问题。在 DSP 中,因为信号处理算法的性质,使用特殊的饱和下溢/上溢指令来有效处理这些情况。

关于这个我可以谈更多,但是你要抓住要点。专业化是 DSP 的全部,这些设备是专门设计来做信号处理的。当处理不以信号处理为中心的算法时,DSP 可能不如别的处理器(这很正常,我也不擅长医学)。所以理解你的应用并选择适当的处理器是很重要的。

随着旨在优化信号处理算法的特殊指令并行执行单元等的应用,没有多余的空间来执行其他一般用途的优化。通用处理器包含了诸如分支预测和前瞻执行之类的优化逻辑,在其他种类的应用中实现了性能的优化。但是这些优化中的某些技术在信号处理应用中不起作用。例如,当应用中有很多分支时,分支预测效果良好,但是 DSP 算法并没有很多分支。很多信号处理代码中包含了执行信号刺激的预定义函数,而不是需要很多分支逻辑的复杂机构。

数字信号处理同样需要软件上的优化。尽管 DSP 中有了各种奇特的硬件优化，但它仍然需要某些重要工具的支持——特别是编译器。编译器是一个优秀的工具，它采用 C 之类的语言并将生成的目标代码映射到指定的微处理器中。优化的编译器要执行一个非常复杂和艰巨的任务——产生充分“利用”DSP 硬件平台的代码。稍后在本书中，我们将讨论更多关于优化编译器的问题。

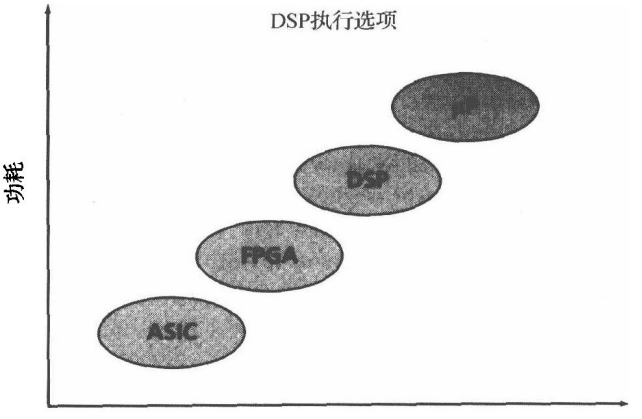
DSP 中没有巫术。事实上，在过去数年中，用来为处理器产生代码的工具已经改善到了这个地步——你可以用 C 和 C++之类的高级语言编写大部分的代码而让编译器去映射和优化这些代码。当然，总有些你可以做的特别的事，而且，为生成最佳的代码，你经常需要给编译器提供某些提示，但是这些真的与其他处理器没有什么区别。事实上，我们将花费几个章节来讨论如何优化 DSP 代码以实现最佳的性能、内存和功耗。

不光是 DSP 中运行的算法种类，DSP 操作的环境也是很重要的。许多（但非全部）DSP 应用需要与现实世界相互作用。这是一个有着很多要素的世界，比如声音、光、温度、运动等。DSP 同其他嵌入式处理器一样，不得不对现实世界作出一定的反应，此类系统其实被称为反应式系统。作为一个反应式系统，它需要响应和控制现实世界，毫无意外，它应是实时的。来自现实世界的数据和信号必须被及时处理。从一种应用到另一种应用的定义是时变的，但它要求我们跟上环境的变化。

由于这种时效性的要求，DSP 以及其他处理器必须对现实世界中的事件迅速地作出响应，迅速地输入输出数据，并迅速地处理数据。我们已经讨论了这一部分的处理。但是不管相信与否，在很多实时应用中，其瓶颈并非数据处理，而是快速地从处理器输入和输出数据。高速 I/O 端口、缓冲串行端口以及其他外围设备等，在 DSP 设计中都是用以尽可能满足现实世界的需求。事实上，由于其处理数据流的速度，DSP 经常被称为数据泵。这是使 DSP 与众不同的又一个特征。

DSP 也在很多嵌入式应用中出现。我将在第 2 章中详细讨论嵌入式系统。然而，嵌入式应用的一个限制因素是资源不足。资源短缺是嵌入式系统的本质特征。我在这里谈到的主要资源是处理器周期、内存、功率和 I/O。后文中将一直如此。不管嵌入式处理器运行速度多么快，适合于芯片的内存多么大等，总会有应用消耗掉所有资源，并寻求更多资源。同样，嵌入式系统具有非常特殊的应用，不像桌上电脑更多地被普通大众应用。

在这一点上，我们应该理解，除了能高效专门执行信号处理，DSP 与任何其他可编程处理器是一样的。那么现在惟一的问题应该是：为什么要编程呢？我不能将所有信号处理用硬件完成么？事实上你能。伴随着在灵活性、成本、功耗和其他几个参数上的相应权衡取舍，DSP 执行技术的内容相当广泛。下面这个图总结了决定是选择可编程性还是固定功能的两个主要权衡因素：灵活性和功耗。



专用集成电路(ASIC)是只执行操作的硬件。这些设备被编程以执行一个固定的功能或一组功能。作为一个只有硬件的解决方案,ASIC 不会受到来自类似可编程冯·诺伊曼体系的某些限制,例如指令和数据的加载及存储等。与可编程解决方案相比,这些设备的运行是非常快的,但是它们不够灵活。在某种程度上,构建 ASIC 与构建任何其他微处理器是相似的。它的设计过程相当复杂,所以你必须确定将要设计成 ASIC 的算法是有效的并且暂时无需改变。你无法轻易地重新编译以修改缺陷或改成一个新的无线标准(事实上,你可以,但是那将花费大量的金钱和时间)。如果你有一个稳定的、定义清楚的功能需要很快速地运行,ASIC 是可行的。

现场可编程门阵列(FPGA)是一种折中的选择。在一定程度上,你可以对它们进行编程和现场重新编程。这些设备不如真正的可编程解决方案灵活,但是它们比 ASIC 要灵活。由于 FPGA 是硬件,它们拥有与其他基于硬件的解决方案相似的性能优势。FPGA 可以被"调整"以实现精确的算法,这是很好的性能。与 ASIC 不同,FPGA 并非真正的专用电路。把 FPGA 想象为一个大型的门的海洋,在那里你可以打开和关闭不同的门来执行你的功能。最后,你实现了应用,但是周围闲置着很多门,有点像待行的汽车。这占用了额外的空间和成本,因此你需要作出权衡:成本、物理面积、开发成本和性能都符合你的期待吗?

DSP 和 μP(微处理器):在这里我们已经讨论过它们的不同了,因此不再重复。个人而言,我喜欢走灵活路线:可编程的。开发信号处理系统时,我会犯很多错误,因为那是很复杂的技术!因此,当我需要修改错误、执行附加优化以增强性能或减少功耗(我们同样将会在本书中讨论更多相关内容)或改成下一种标准时,我乐于知道自己拥有灵活性来作出改变。整个信号处理领域在不断增长并且变化的如此之快——证据就是标准的不断发展和改变——我宁愿进行快速而便宜的升级和改变,而这只能由可编程解决方案提供。

一般的答案，一如既往，找到平衡点。实际上，许多信号处理解决方案分割横跨多个不同的处理单元。算法流的某些部分——那些有较高概率在未来发生改变的——映射到可编程 DSP。在可预计的将来能保持相当稳定的信号处理功能则被映射到硬件门电路（ASIC、FPGA 或者是其他硬件加速）。信号处理系统中控制输入输出、用户界面以及系统中心的整体管理的那些部分可以映射到更通用的处理器中。复杂的信号处理系统需要合适的处理单元的联合以实现正确的系统性能/成本/功耗的权衡。本书也会在此问题上花费更多的时间。

信号处理时代已经到来，它无处不在。无论何时你得到一个信号，想要更多了解它，以某种方式传递它，把它变得更好或更坏，都需要处理它。数字化只是一个使它可以工作在某些种类计算机上的过程。如果是一个嵌入式应用，你就必须用尽可能少的资源来完成它。什么都会耗费金钱、周期、内存和功率，所以什么都必须节约。这是嵌入式计算的特性：特定应用，为掌上工作量身定造，尽量减少成本，尽可能提高效率。这是 1982 年我开始从事这个行业的方式，而今天应用了同样的技术和处理，标准当然也改变了，当时需要超级计算机的计算问题今天可在嵌入式设备上解决！

本书将触及这些以及更多涉及数字信号处理的领域。这里有很多东西需要讨论，而我将采用实践而非理论的途径来描述这些做好 DSP 所需面临的挑战和过程。

光盘内容

CCStudio 开发工具 120 天的完全免费试用版以及“DSP 入门基本指导”。还有关于样本算法，或者是 CCStudio 集成开发环境的丰富功能，探讨的基准测试程序。更多关于 TI DSP 的信息，请访问 www. ti. com/dsp。

内 容 简 介

本书详细介绍了DSP在嵌入式实时系统设计中的软件开发方法，是探讨DSP软件设计技术的专业技术指南。内容包括：数字信号处理技术、嵌入式实时系统与DSP的内在关联性、DSP嵌入式系统基本开发步骤、DSP硬件结构及DSP软件性能与其硬件结构的关系、DSP软件设计的优化方法和技术、DSP软件设计的实时操作技术、DSP系统的测试和调试方法、多CPU片上系统开发中嵌入式DSP软件设计技术等。随书附光盘一张，内含书中大量应用实例的代码。

本书适合对DSP软件技术开发有兴趣的本科生、研究生、研发人员阅读。

图书在版编目(CIP)数据

嵌入式实时系统的DSP软件开发技术 /（美）奥沙那(Oshana，R.）著；郑红，刘振强，王鹏译. －北京：北京航空航天大学出版社，2011.1

书名原文：DSP Software Development Techniques for Embedded and Real-Time Systems

ISBN 978－7－81124－521－9

Ⅰ.①嵌… Ⅱ.①奥… ②郑… ③刘… ④王… Ⅲ.①微型计算机—系统设计②数字信号—信息处理系统—系统设计 Ⅳ.①TP360.21②TN911.72

中国版本图书馆CIP数据核字(2010)第251333号

嵌入式实时系统的DSP软件开发技术

DSP Software Development Techniques for Embedded and Real-Time Systems

［美］Robert Oshana 著

郑 红 刘振强 王 鹏 译

责任编辑 王福秋

*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(100191) 发行部电话：010－82317024 传真：010－82328026

http://www.buaapress.com.cn E-mail：bhpress@263.net

北京市松源印刷有限公司印装 各地书店经销

*

开本：787 mm×960 mm 1/16 印张：30.25 字数：678千字

2011年1月第1版 2011年1月第1次印刷 印数：4 000册

ISBN 978－7－81124－521－9 定价：69.00元(含光盘1张)

目　录

第1章

数字信号处理概论

1.1 什么是数字信号处理?

数字信号处理(DSP)是一种信号和数据的处理方法,可实现对信号的增强或调整,或者通过分析信号以测定特殊的信息内容。它包含对现实世界中信号的处理,这些信号被转换为数字序列,然后进行数学处理,以剥离出某些信息或者将信号转换为某些更合适的形式。

"数字信号处理"中的"数字"要求在处理中使用离散的信号(更易于处理的数字形式)来表现数据,换句话说就是信号被数字化了。这种表现方法意味着对一种或多种信号特性以某些形式进行量化,包括时间特性。

这只是数字数据的一种,其他种类还包括 ASCII 码制的数字和字母。

"数字信号处理"中的"信号"是一种可变参量,因为它流经电子电路,这个参量又被视为一种信息。信号通常[①]以一条不断变化的信息的形式出现在模拟世界。现实世界中的信号包括:气温、流量、声音、光、湿度、压力、速度、体积及位置。

这类信号本质上是一个变化于无穷数值(理论上)之间的电压值,它代表了物理量的变化模式。信号的另一个例子是正弦波,人类的语音和传统的电视摄像机中的信号都是这种波形,一个信号就是一个可以检测的物理量。在这些信号的基础上我们可以进行信息的传输。

当一个信号把某个物理量的变化描述为单变量的函数时,它被称作是一维的(1 - D)。音频或者语音信号就是一维的,因为它表现了空气压力连续变化的时间函数。

最后,"数字信号处理"中的"处理"是指通过相对于硬件电路的软件程序对数据进行处理的过程。数字信号处理器是一个通过源程序来操作信号以执行对现实(模拟)信号的处理功能的设备或系统。我们能够相对容易地改变软件程序以调节信号处理的行为,这是一种优势,要改变模拟电路就困难得多了。

由于数字信号处理器与外界信号相互影响,DSP 系统必须对外界环境反应灵敏。换言

① 用"通常"来描述是因为某些信号本身已经是离散的形式。例如开关信号,它可直接由开或关的状态量来表示。

之，DSP 必须与环境变化保持一致，这就是我们很快就要谈到的“实时”处理的概念。

1.2 数字信号处理简史

第一批数字信号处理解决方案是采用中等规模集成 TTL① 硅芯片，上百片这种芯片被用来组成多级算术逻辑单元和独立乘法器。这些早期的系统体积大、价格昂贵并且需要高压供电。

第一个单芯片的 DSP 出现于 1982 年，是由 TI 公司出品的 TMS32010 DSP。没过多久 NEC 也推出了 uPD7720。这些处理器运算速度达到 5 MIPS②。早期的单芯片只有很小的 RAM，售价约 600③ 美元。这种单芯片方案有效减少了整个系统芯片的数量，在减少制造的复杂度和成本的同时，降低了系统功耗，提高了系统的稳定性。这类 DSP 的制造多数采用了 NMOS④ 技术。

随着 DSP 设备市场的持续增长，厂商开始增加更多的集成内容，比如片内 RAM、ROM 和 EPROM。高级的寻址功能，包括 FFT 位反向寻址和循环缓冲寻址，被开发出来（这是两种常见的 DSP 中心寻址模式，后面会做更多细节上的介绍）。增加了串行口以实现高速数据传输。这些第二代产品其他功能结构上的增强包括：定时器、直接存储器存取（DMA）控制器、包含映像寄存器的中断系统以及集成的数模和模数转换器。

浮点型 DSP 出现于 1988 年。DSP32 是 AT&T 公司推出的浮点型 DSP 产品，同期 TI 公司也推出了 TMS320C30。它们更易于编程并且具有自动缩放等特点。由于浮点型结构的硅面积较大，浮点型 DSP 的成本比传统的定点型处理器要高，同时功耗较高而且处理速度偏低。

20 世纪 90 年代初，支持并行处理的 DSP 开始出现，支持高级通信功能的单处理器 DSP（例如 TI 的 TMS320C40）面世。多个处理单元被设计集成到同一集成电路中（例如 TMS320C80）。

如今出现许多高级 DSP 结构类型，本书将讲述其中的几种。高级的结构包括多功能单元、甚长指令字结构和快速处理专门任务（例如手机中的回波消除）的专用功能单元。

---

① Transistor－Transistor Logic，晶体管—晶体管逻辑（电路），输出从两个晶体管引出的普通逻辑电路。第一个采用 TTL 的半导体是由 TI 公司于 1965 年开发的。

② 单位 MIPS（Millions of Instructions Per Second，百万条指令每秒）是计算机性能的常用量度标准，也是一台较大型计算机能完成的工作量的标准。历史上许多年间，由 MIPS 每美元度量的计算机成本每年减少一半（摩尔定律）。

③ 今天一个类似的器件售价低于 2 美元。

④ Negative－channel Metal－Oxide Semiconductor（N 沟道金属氧化物半导体）的缩写。这种半导体反向偏置，从而通过电子流动实现晶体管的开关。相对地，PMOS（Positive－channel MOS）通过空穴的流动起作用。NMOS 比 PMOS 速度快，但是它的生产成本也比较贵。