

FPGA 在高速数字信号处理系统中的应用

上网时间：2001 年 04 月 14 日

由于成本、系统功耗和面市时间等原因，许多通讯、视频和图像系统已无法简单地用现有 DSP 处理器来实现，现场可编程门阵列(FPGA)尤其适合于乘法和累加(MAC)等重复性的 DSP 任务。本文从 FPGA 与专用 DSP 器件的运算速度和器件资源的比较入手，介绍 FPGA 在复数乘法、数字滤波器设计和 FFT 等数字信号处理中应用的优越性，值得(中国)从事信号处理的工程师关注。

Chris Dick

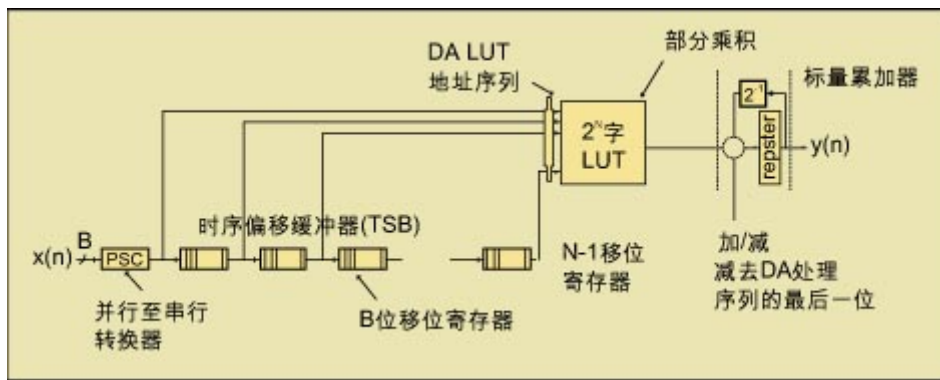
Xilinx 公司

由于在性能、成本、灵活性和功耗等方面的优势，基于 FPGA 的信号处理器已广泛应用于各种信号处理领域。近 50%的 FPGA 产品已进入各种通信和网络设备中，例如无线基站、交换机、路由器和调制解调器等。FPGA 提供了极强的灵活性，可让设计者开发出满足多种标准的产品。例如，万能移动电话能够自动识别 GSM、CDMA、TDMA 或 AMPS 等不同的信号标准，并可自动重配置以适应所识别的协议。FPGA 所固有的灵活性和性能也可让设计者紧跟新标准的变化，并能提供可行的方法来满足不断变化的标准要求。

复数乘法

复数运算可用于多种数字信号处理系统。例如，在通讯系统中复数乘积项常用来将信道转化为基带。在线缆调制解调器和一些无线系统中，接收器采用一种时域自适应量化器来解决信号间由于通讯信道不够理想而引入的干扰问题。量化器采用一种复数运算单元对复数进行处理。用来说明数字信号处理器优越性能的指标之一就是其处理复数运算的能力，尤其是复数乘法。

一个类似 DSP-24(工作频率为 100MHz)的器件在 100ns 内可产生 24×24 位复数乘积(2 个操作数的实部和虚部均为 24 位精度)。复数乘积的一种计算方法需要 4 次实数乘法、1 次加法和 1 次减法。一个满精度的 24×24 实数管线乘法器需占用 348 个逻辑片。将 4 个实数乘法器产生的结果组合起来所需的 2 个 48 位加法/减法器各需要 24 个逻辑片(logic slice)。这些器件将工作在超过 100MHz 的时钟频率。复数乘法器采用一条完全并行的数据通道，由 $4 \times 348 + 2 \times 24 = 1440$ 个逻辑片构成，这相当于 Virtex XCV1000 FPGA 所提供逻辑资源的 12%。计算一个复数乘积所需的时间为 10ns，比 DSP 结构的基准测试快一个数量级。为了获得更高的性能，几个完全并行的复数乘法器可在单个芯片上实现。采用 5 个复数乘法器，假设时钟频率为 100MHz，则计算平均速率为每 2ns 一个复数乘积。这一设计将占用一个 XCV1000 器件约 59%的资源。



这里应该强调的一个问题是 I/O，有这样一条高速数据通道固然不错，但为了充分利用它，所有的乘法器都须始终保持 100% 的利用率。这意味着在每一个时钟来临时都要向这些单元输入新的操作数。

除了具有可实现算法功能的高可配置逻辑结构外，FPGA 还提供了巨大的 I/O 带宽，包括片上和片外数据传输带宽，以及算术单元和存储器等片上部件之间的数据传输带宽。例如，XCV1000 具有 512 个用户 I/O 引脚。这些 I/O 引脚本身是可配置的，并可支持多种信号标准。

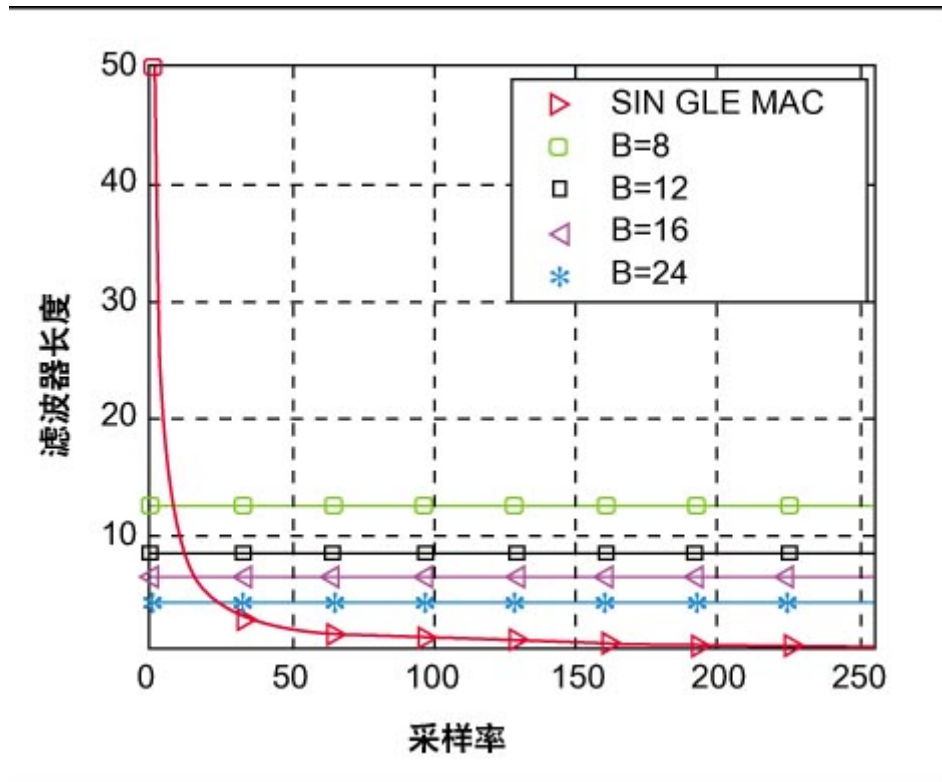
实现复数乘法器的另一种方法是构造一个单元，该单元采用单设定或并行的 24×24 实数乘法器。这种情况下，每一个复数乘法需要 4 个时钟标识，但是 FPGA 的逻辑资源占用率却降到了最低。同样，采用 100MHz 系统时钟，每隔 40ns 可获得一个新的满精度复数乘积，这仍是 DSP 结构基准测试数据的 2.5 倍。这一设定方法需要大约 450 个逻辑片，占一个 XCV1000 器件所有资源的 3.7% (或 XCV300 的 15%)。

构造一条能够精确匹配所需算法和性能要求的数据通道的能力是 FPGA 技术独特的特性之一。而且请注意，由于 FPGA 采用 SRAM 配置存储器，只需简单下载一个新的配置位流，同样的 FPGA 硬件就可适用于多种应用。FPGA 就像是具有极短周转时间的微型硅片加工厂。

先进的软件数字信号处理器在执行乘法相关的应用(包括复数乘法)时十分有效。一个 250MHz 的 DSP 芯片每个时钟周期可进行 2 次 16×16 的复数乘法运算，这相当于每 2ns 产生一个新的乘积。但是，FPGA 的性能更胜一筹。一个 16×16 的复数乘法器(产生全精度 32 位复数结果)占用约 700 个逻辑片，并可支持 160MHz 的时钟频率，这相当于每 6.25ns 产生一个新的复数乘积。将 4 个这样的功能放置在一个芯片上，每 1.56ns 即可产生一个新的乘积。如果需要更高的性能，可采用额外并行方法将计算速率提高到前所未有的水平。

数字滤波器

FIR 滤波器几乎是所有数字信号处理系统中通用的基本构建模块之一。输出采样通过将输入采样流与滤波系数卷积而得到，这一运算也称作内积(inner-product)或向量点积(vector dot-product)运算。滤波系数定义了网络的频率响应。在要求高阶滤波、高采样率或两者兼有的应用中，所需算法工作量将会十分庞大。计算一个输出值需要 N 次乘法和 N-1 次加法，能够提供这种算法性能的高性能硬件平台对大多数信号处理设计者来讲都非常诱人。



数字滤波器在数字接收器和发送器中的使用方法有很多。多速率滤波器(如十进制编码器和插值器)是数字通讯系统中许多关键功能的核心。例如,在数字主控振荡器中,多相插值器是信号处理链的一部分,该信号处理链先将信号从基带转换到数字 IF,然后通过宽带 DAC 将信号转换为模拟信号并继而由 RF 后端进行处理。在数字接收机中,多速率滤波器可用于数字下行转换器、信道量化器、时序恢复和采集环路中对信号进行数字再采样。

实现实时滤波器的一个常用方法是采用软件可编程的信号处理芯片。ASIC 方案是一种高性能但灵活性欠佳的替代方法。最新的设计方法是利用基于 FPGA 的硬件系统可提供的并行处理能力。

a) 基于 MAC 的方法实现数字滤波器

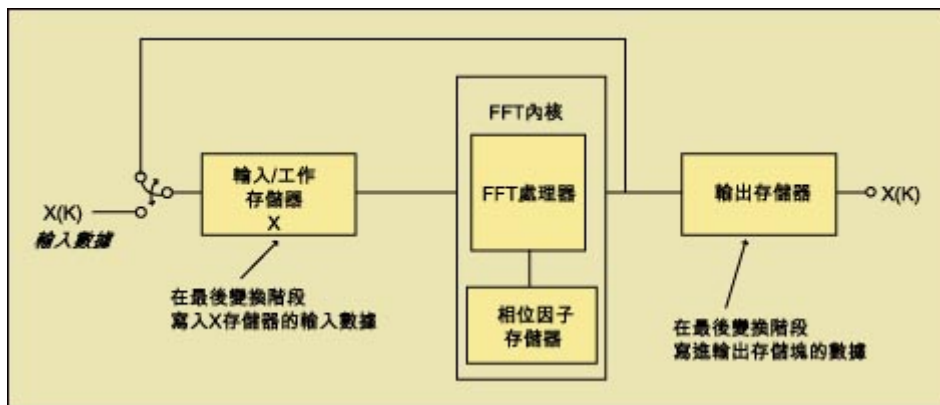
在 FPGA 中实现 FIR 滤波器的方法有很多。最显而易见但并非最佳的方法是模仿用于 ASIC 或指令集 DSP(ISDSP)的技术。这种方法采用一种预置的乘法-累加器(MAC)单元。由于许多信号处理工程师都非常熟悉这些实现滤波器的半导体技术,我们将以此为起点来探讨如何用 FPGA 实现 FIR 滤波器。

内积计算可在 1 个或几个 MAC 单元上分割,这是当前信号处理器(ASIC 和 ISDSP)常用的一种方法。这一方法同样适用于 FPGA 实现数字滤波器的设计,但在 FPGA 环境下,设计者对硅片有完全的控制权,并可确定将多少资源分配给内积引擎。在评估某一特定滤波器实现技术时,设计者需考虑一些必要的因素,如性能(采样率)、功耗、尺寸和成本等。为了给出一个 FPGA FIR 滤波器实现的参考框架,我们将对基于 MAC 的 FPGA FIR 滤波器实现方法与 ASIC 和 ISDSP 方法作一比较。首先考虑由 DSP 结构 DSP-24 信号处理器提供的 ASIC 方案。这一器件可实现多种功能,其中包括滤波。DSP-24 采用 100MHz 的时钟频率及 24

位数据和系数，可在 5ns 内完成一次实数滤波抽头计算，在 10ns 中完成一次复数抽头计算。

多数 MAC 引擎的核心是乘法器。为了支持 24 位输入采样和系数，需要一个 24×24 位精度的乘法器。这一乘法器采用 Virtex FPGA 技术，可用 348 个逻辑片实现，这相当于 XCV300 等低密度器件的 11%。为实现一个完整的 MAC 单元，累加器必须与乘法器级联。利用 FPGA，系统结构师可自由选择这一部件的精度。下面以一个 56 位宽累加器为例说明这一点。累加器需要 28 个逻辑片。除了地址生成和算法单元安排所需的少量控制逻辑外，MAC 单元可用大约 $348+28=376$ 个逻辑片实现。除了算法引擎外，完整的 FIR 滤波器还需要存储滤波器系数和输入采样的历史记录。从 Virtex FPGA 的模块化 RAM 或分布式 RAM 中可选择几种存储资源。

对大型滤波器来讲，模块化存储器可为输入采样和/或滤波器系数提供有效的存储。此外，输入采样和/或系数还可存储在分布式存储器内。总之，分布式存储器是由构成逻辑结构的元素实现的，如上面所提及的 16×1 LUT。一个采用 24 位输入采样和 24 位系数的 16 组滤波器总共需要 24 个逻辑片来实现滤波器存储和系数存储。100MHz 的时钟频率可实现每组 10ns 的乘法-累加速率。实际上，根据速度等级，较高的时钟速率是可以实现的。



尽管单个 MAC 单元可提供 100Mega-MAC(MMAC)的性能，但 FPGA 的可配置特性可让设计者利用许多 DSP 算法中的高度并行处理能力来获得更高的性能。在这种情况下，可在一个滤波器中采用 2MAC 来提供 200MMAC 的性能，这相当于每 5ns 完成一次 MAC 计算。这种双 MAC 设计需要大约 3% 的 XCV1000 FPGA 资源。当然，额外的 MAC 单元也可用来进一步缩短有效 MAC 循环周期。

目前的许多 ISDSP 提供了多个执行单元来执行算法操作。几个 MAC 单元通常能够加速滤波器的计算。每一时钟周期执行的 MAC 数目决定了计算的精度。例如，目前的 ISDSP 在每个时钟周期可完成 2 个 80 位实数 MAC(32 位数据和系数)或 8 个 40 位 MAC(16 位数据和系数)运算。后一个数据相当于 2Giga-MAC(GMAC)在 250MHz 时钟频率时的性能。这一有效使用并行处理能力的方法可增强许多数据计算强度要求高的系统的性能。

并行访问(accessing concurrency)是 FPGA 成功实现所有信号处理功能的关键。但是，与芯片设计者决定并行访问和并行访问能力的 ISDSP 不同，通过 FPGA 信号处理硬件，系统设计者可自由分配芯片资源以开发适当的并行处理能力来满足系统性能要求。假设要用 16 位精度的数据和系数以及 40 位的累加器搭配一个传统的乘法-累加式 2GMAC 内核引擎。一个 16×16 的乘法器需要 168 个逻辑片，并可方便地在 150MHz 频率下工作。一个 40 位的累

加器大约需要 20 个逻辑片，因此 1 个 MAC 单元总共需要 188 个逻辑片。若采用 150MHz 时钟，则需要 14MAC 单元来实现 2GMAC 的性能。每个 MAC 单元的输出通过一个加法器树(adder-tree)连接起来。14 个 MAC 单元和加法器树由大约 2,892 个逻辑片实现，这相当于一块 XCV400 FPGA 资源的 61%。

将这一方法加以扩展以构造更具吸引力的 FPGA DSP 引擎可获得更强的并行处理能力。由此可见，采用可编程逻辑简单地模仿 ISDSP 设计是解决并行处理问题的最佳方法。

b) 采用 DA 算法的 FIR 滤波器

有很多方法可计算内积，其中一种方法称为分布式算法(DA)。DA 滤波器的通用模型如图 1 所示。基于分布式算法的计算需要一系列的查表、加法和减法运算。所有这些功能都可用 FPGA 实现。DA 滤波器的一个有趣特性是滤波器吞吐能力不再与滤波器长度有关，而是取决于输入采样精度。图 2 中的线性曲线可说明这一点。如图所示，在 DA FIR 滤波器结构中，对于给定的输入采样精度 B，采样速率保持恒定，与滤波器抽头数无关。例如，若 B=12 且系统时钟为 100MHz，滤波器的采样速率在滤波器长度为 10、20、100、200、...时均为 8.333MHz。若 N=200，这一有效计算速率为每 0.6us 完成 1 个 MAC 或每秒 1.7GMAC。

对于 24 位输入采样，采样速率为 4.1666MHz。在许多系统中，滤波器的系数集是对称的，对称性可用来减少滤波器设计对逻辑片的需求。此时，滤波器速率略微有些减小。对 24 位输入采样和 100MHz 时钟，采样速率将为 4MHz。对于一个 200 抽头的滤波器，仍然会获得每秒 800MMAC 的计算速率。

c) 用并行分布式算法提高乘法速度

基于 DA 的计算最显然而直接的形式是位串行(bit-serial)，对基本算法的扩展可消除这一潜在的吞吐能力限制。串行处理数据(即一次 1 位)可得到适中的计算速率。若输入变量长度为 B 位，则需要 B 个时钟周期来完成一次内积计算。提高速度可用其他几种方法获得。一种方法是将输入字段分割成 L 个子字段，然后并行处理这些子字段。这一方法需要 L 倍的存储查找表，从而导致存储需求和成本的直线上升。通过将输入变量分解为一位子字段可获得最大速度。通过这种分解，每一时钟周期就可计算出一个新的输出采样。

d) 为什么用 FPGA 实现 DA 滤波?

DA 滤波器的一些结构特性使它非常适合采用 Virtex FPGA 技术实现。这些 FPGA 中的功能发生器(FG)可用做移位寄存器，这一功能正是实现保存滤波器输入采样记录的位串延迟线或 TSB 所需要的。TSB 的抽头点可作为 DA 查找表(LUT)的地址输入。功能发生器还可配置为 RAM 和 ROM，这一能力可用来有效创建甚高速 DA 查找表。

在分布式存储器内实现 DA LUT 的另一个方法是利用新一代 FPGA 中的存储模块。以构建一个 70 抽头的线性相位 FIR 滤波器为例，只有 35 个专用的滤波器参数，一个简单的 LUT 方法就需要存储大量的部分乘积项。如果将 35 个地址线划分为 4 组 8 个地址和 1 组 3 个地址，就比较容易管理。这时只需要 4 个 256 输入和 1 个 8 输入的 LUT。这 5 个表可存储在片上存储模块中，再利用 1 个简单的加法器树将它们的输出连接起来。

e) FPGA 多速率滤波器

高效的单速率 FPGA FIR 滤波器有助于构造大量的多速率滤波器。图 3 和图 4 所示的多相插值器和多相十进制编码器就可采用上述 FPGA 滤波器的构建模块实现。

实现多相十进制编码器的一种方法是采用分布式算法。每一个多相分段都可作为一个独立的 DA 过程来实现。所有的分段并行运行，以产生一个可支持特高输入采样率的下采样滤波器。例如，考虑一个 8 至 1 的复数十进制滤波器(实数输入采样，复数系数)，其输入采样率为 80MHz，输出速率为 10MHz。滤波器的每一分段均具有 32 抽头，并支持 10 位输入采样和 12 位系数。每个多相子滤波器都有 320MMAC 的计算速率，复数滤波器的净计算性能为 5.12GMAC。

除了这一结构提供的性能外，另一个需要考虑的重要问题是逻辑片要求。32 抽头滤波器中的任意一个都需要 126 个 Virtex 逻辑片，整个滤波器将占用 2,200 个 Virtex 逻辑片。一个 XCV400 FPGA 具有 4,800 个逻辑片，因此可容纳 2 个这样的十进制编码结构，并可提供每秒 100 亿个 MAC 的性能。该例中假定系统时钟速率为 100MHz，当前的 FPGA 技术很容易支持这一速率，实际上还可为某些算法功能提供 200MHz 时钟频率的支持。最后需要注意的一点是滤波器系数的对称性尚未利用。如果可能，将可得到更为紧凑的设计。

多相插值器可采用分布式算法或并行 MAC 方法在 FPGA 中实现。不同的方法适合处理不同的问题。例如，在数字接收机中，载波和信号的同步，以及自动增益控制都需要许多控制回路。在时序恢复回路中，需要插值器来调整信号采样的相位，并驱动其自身的跟踪回路。尽管插值器需要支持许多相位，而在任一时刻通常只有很少的分段部分在运行。这时，传统的预置 MAC 方法就比较适用。所有滤波器的完整系数集都存储在模块化或分布式存储器内，而所需的系数集(对应于一个多相分支)则被导入 1 个或几个 MAC 中，以便进行内积计算。

除了上述结构外，实现多速率滤波器的方法还有很多。例如，由 Hogenauer 首先提出的级连积分梳状滤波法(CIC, cascaded-integrator-comb)就特别适合用 FPGA 实现，因为它依赖于加法器、寄存器和减法器来实现算法，而 FPGA 在实现这些功能方面极为高效。

傅立叶变换

与 FIR 滤波器一样，离散傅立叶变换(DFT)也是许多信号处理系统中常用的方法之一。DFT 通常采用快速傅立叶变换(FFT)算法来实现。在数字通讯系统中，基于 FFT 的算法可用来提取传输信道的特征，以便最有效地利用可用带宽资源。在正交频分多址(OFDM)调制解调器中，FFT 既可作为发送部分的调制器，又可作为接收部分的解调器。

在大约 66us 的时间内，一个先进的软件 DSP 可对 16 位精度复数采样执行一次 1024 点的复数 FFT。如果有必要，通常还需要增加时间(约 9us)来完成位反向置换(bit-reversal permutation)。采用基数为 4 的标准 Cooley-Tukey 算法，并且仅使用适度的并行处理能力，同样的 FFT 就可在 41us 内由 Virtex FPGA 实现，包括位反向置换在内。

在比较 FPGA DSP 和其它 DSP 时，一个极为重要的系统参数是输入/输出(I/O)带宽。比较不同 DSP 硬件方案的原始算法性能很重要，但 I/O 子系统也同样重要。除了一些专用引脚外，FPGA 上几乎所有的引脚均可供用户使用，这使得 FPGA 信号处理方案具有非常高性能

的 I/O 带宽。图 5 所示为上述 FPGA FFT 方案。大量的 I/O 引脚和多块存储器可让信号处理工程师在设计中获得优越的并行处理性能。在上述 FFT 范例中，独立的存储块可作为输入/工作存储区和结果缓存区，这使得 I/O 可与 FFT 计算同时进行。因此，41us 的执行时间包括了所有的输入、输出和计算操作。

上述 1024 点 FFT 占用大约 1,900 个逻辑片，这相当于 XCV300 等中等密度器件资源的 66%。FFT 算法可充分利用并行处理能力，而且目前的 FPGA 提供了 I/O 带宽、片上存储器和逻辑等多种资源，从而使得单片 FFT 引擎的高度并行处理成为现实。例如，利用标准的基数为 4 的 Cooley-Tukey FFT 算法的直接划分，一个计算单元可分配到每个蝶形列(column of butterfly)中。除了相邻蝶形列间的存储器缓存外，这一方法共需 5 个蝶形引擎(butterfly engine)。一个 1024 点 FFT 可在 1024 个时钟周期内完成。若采用 100MHz 的时钟，变换时间为 10us。如果需要，还可进一步提高并行处理能力，以获得小于 10us 执行时间的单片 FFT 方案。

本文结论

尽管 FPGA DSP 系统代表了信号处理领域的一个重要分支，将 FPGA 引入 DSP 硬件的设计却在呈指数增长。这一爆炸性增长的原因是，所有主要的 FPGA 供应商和第三方 IP 设计商都在提供越来越多的 FPGA IP 内核。利用这些资源，系统设计者就能将精力集中在设计上，而无需关心滤波器和转换等底层模块的细节。