



AudioDE™

ARM公司推出面向音频处理的高性能低功耗数据引擎

作者:

Mark Wezelenburg

ARM

摘要:

本文介绍了针对便携式音频应用中的嵌入式数字信号处理而经过设计的专用数据引擎——AudioDE™ 音频处理器。AudioDE由ARM公司的OptimoDE™ 架构开发的授权处理器。结构上的特点、高并行度编译为音频应用提供了极小的功耗和面积。配合专门的系统功能与常规的AMBA接口功能(AIKO), AudioDE综合系统易于实现。

简介

AudioDE处理器的设计目标在于满足便携式音频应用的需求:最优的性能、功耗和面积(PPA)。也就是在满足在数字信号处理平台上运行的处理任务的需求情况下,需要尽可能小的处理速度、非常小的核心尺寸和存储器尺寸。除了命名上的不同,AudioDE数据引擎实现了一个标准DSP指令集的微处理器架构,因而可以在整数和小数数据计算方面媲美其他任何DSP结构。

AudioDE处理器在计算资源、寄存器文件、存储器带宽、寻址能力之间取得了完美的平衡。这些特点与多指令可变超长指令字编码控制器一起,使快速、高并行度编码的开发成为可能,从而实现最少的运行频率需求。

AudioDE处理器为双哈佛结构,带有一个单24位算术逻辑单元(ALU)和一个带有48位累加器的单24x24位乘加器(MAC)。这些单元都以并行状态工作,并具有双口数据存储接口,可以同时产生地址信息。所有寻址模式包括反转和求模运算都支持高效率的单周期访问。

ALU具有增强指令以加速在处理压缩媒体文件时典型的比特流访问。这一特性使AudioDE数据引擎可以直接处理DMA驱动的输出数据流,并可以对像诸多标准数据流中的乱序访问自动提供缓冲处理。像存储器这样的外部接口一样,AudioDE处理器在没有数据处理的时候可以自动进入低功耗模式。

一个MP3解码器可以作为高效的AudioDE处理器和OptimoDE工具的完美示例。一个可编程的解码器仅需要:

- 8MHz 运行频率
- 22k字节程序存储器
- 22k字节数据存储器

最终实现的AudioDE处理器(0.13微米CMOS工艺,供电电压1.2伏)仅消耗功率0.8mW,核心逻辑仅有45k电路门。

OptimoDE 技术

AudioDE音频处理器由标准的OptimoDE架构产生出来,使用架构中提供的标准工具和IP结构。这也就意味着,尽管这篇文章中提供了详尽明确的配置,仍然很容易配置出一些其他的数据引擎来适应更多样的应用。

OptimoDE架构体系包括一系列广泛的开发工具套件。这些工具包括了可以用来增强核心功能的结构工具、类似指令仿真器的仿真工具和转为OptimoDE开发的高并行度编译器。这个编译器可以达到任何手工编写的汇编库同样好的编码效率。由于这个工具可以提供一个交互的运行环境,在源码阶段就可针对速度和目标代码尺寸间的权衡提供进一步优化的反馈。

在数据引擎开发的过程中有三个阶段:

- 设计阶段:这一阶段选择或创建微处理器结构。
- 编码阶段:为运行在某一固定架构的数据引擎上的任务编写源代码。
- 实现阶段:以可综合的Verilog例化微处理器架构。这包括对标准和用户定义资源的实现。



一旦微处理器结构变得可以调整，上述阶段中的前两个就变成了交互的过程。所有的配置工具作为专一的架构，从而简化了这一过程。另外，又有最后两个阶段是和应用相关的，并且只有在最后阶段才会产生针对某一固定结构的机器码。图1阐明了这一过程。

整个设计过程结束的时候，由DesignDE配置工具生成具有固定特性的微处理架构。同时指令集仿真模型也被创建。

修改或扩展。接下来设计者定义数据通路资源之间的连接。同时提供一个带有功能单元的标准资源库。

用户自定义的资源也可以在库管理器



Figure 2: The OptimoDE interactive design process

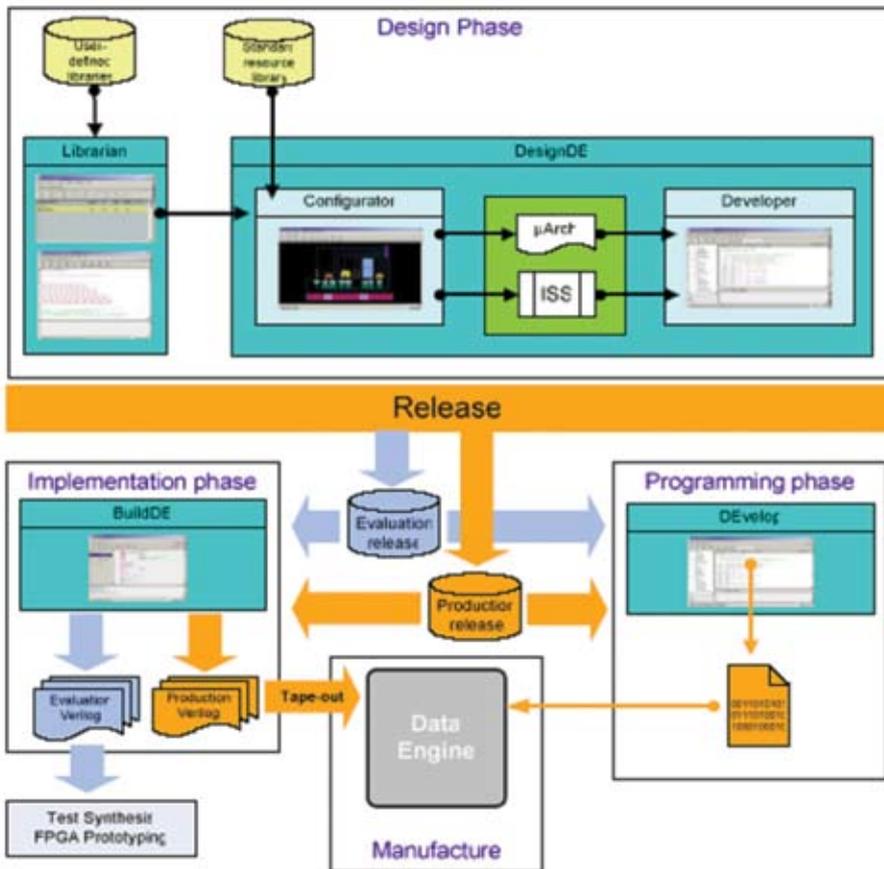


Figure 1: The OptimoDE tools and design flow

这一过程包括了编程和相关的代码片断分析，这与接下来描述的编码阶段一致，但同时也包含了改进微处理结构的目的。

DesignDE包括多个组件。配置工具输入预先配置的微处理结构并且很容易

中实现，库管理器中包含了以C、C++或SystemC语言描述的组件，并产生相应的Verilog代码。这些Verilog代码都是功能正确并且可综合的。

编码阶段最终要对相应的微处理结构代码给出应用源代码和相应的开发编译器

源代码的编译以交互的方式进行。开发环境可以给出关于运算时间和资源使用的静态详细统计。

进一步的动态统计也可在作为微处理结构的补充生成的指令集仿真模型上展开。实现阶段由BuildDE工具例化微处理结构，并生成可综合的Verilog。因为代码开发在设计阶段完成就可以开展，实现阶段就可以和编码阶段并行开发，从而减少了总体的开发时间。

面对特定应用的设计流程

AudioDE音频处理器是使用OptimoDE工具针对特殊应用所配置的处理器的。这一部分描述了AudioDE处理器的设计过程，以此为例，我们可以了解使用OptimoDE架构的方法与作用。

以上图2说明了设计过程中的各个步骤。整个过程的目的在于在核心成本与效率之间找到平衡，同时消除重复设计的风险。

算法与应用类型分析：以分析现存的诸多音频应用和普通数字信号处理应用为起点。以下几点被认为与音频算法具有很大的相关性：

- 声音需求倾向24位数据通路以减少中间量化的开销；



- 音频解码算法基于一个增强的累加滤波组；
- 可变长数据以比特流的形式存储和读取。

更多对于微处理结构性能期望的观察现在已经由特殊指令集或数据并行处理的方法得到解决。面向应用的资源和指令集：对于AudioDE处理器，这种方法的一个应用是标准ALU对比特流输出访问的增强特性。对这种架构的分析表明由于提供针对这种目的的特殊指令，使得指令数目得到明显的减少。

这个调整阶段定义了一个支持高吞吐计算的数据通路，同时保持面积和功耗上的高效率。因而，AudioDE系统架构被设计成仅包含单个乘加器和累加器。尽管这限制了操作的并行度，累加器和乘加器仍然可以被安排成可100%装载算法的结构，这样对于数据的移动同样提供了高水平的并行度。

AudioDE硬件结构

AudioDE总体硬件结构如图3所示：

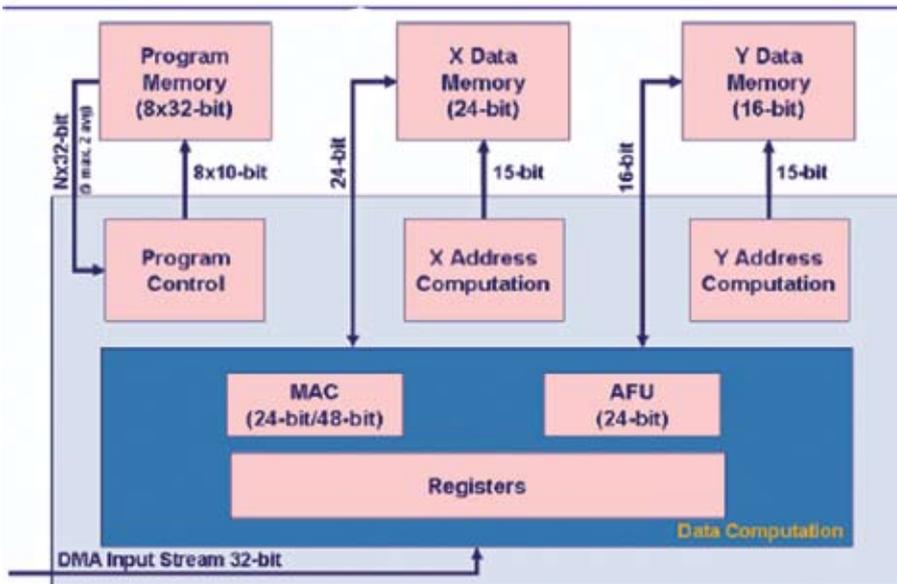


Figure 3: The AudioDE hardware architecture

这一特点非常的有效，因为它不仅降低了运算主频，同时通过给流媒体提供一个统一的接口而减少了结构的复杂度。通常AudioDE处理器具有非常丰富的指令集架构，同时具有适合信号处理的浮点算法、支持标准C的整数算法以及前导零地址计算的增强指令。

系统结构的调整：这个微处理结构的特点在于强调并行运行连续的前导零计算和乘加类型运算的处理能力。性能指示器包括了传统的DSP基准算法和音频应用中常见的滤波算法。

AudioDE音频处理器具有一个低跳转延迟，支持多指令集和可变超长编码指令字(VL-LIW)的控制器。OptimoDE编译器可见所有程序的数据和指令流水线，对于复杂的循环控制开发工具会小心处理，从而减少了硬件的复杂度。

AudioDE功能单元为不含流水线的单周期执行。由于OptimoDE编译器可以控制所有的数据通路行为，所以计算单元可以不通过改变控制器来达到流水，从而拥有更高的运行频率。

AudioDE的数据通路包括一个24位算术逻辑单元(ALU)和一个24位乘加器。乘加器可以执行单精度乘法或单周期的双精度加法。分数乘法的输出支持自动对齐。由于饱和运算单元是整数运算单元的一部分，所有指令既可以是饱和运算也可以是不饱和运算。另外，累加器和乘加器都支持隐性或显性的乘法舍入指令、右移和类型转换。

数据通路包含有一个大容量高带宽的寄存器文件。它的容量适于做FFT和DCT类似算法中以4为基的快速运算。为了减少AudioDE核心结构的复杂度，以及简化编译器的任务，算术数据通路与下面描述的地址产生数据通路被设计成相对独立的。为了保持简单的单周期乘加运算，提供了两个独立的数据存储器端口，以及第三个单宽度的程序存储器接口。

为了满足成本敏感的嵌入式应用对内存尽可能小的需求，X方向和Y方向的内存被设计成非对称的。只有X方向内存是24位宽度以满足典型音频处理需求，而Y方向内存被优化成16位宽，这对整数或系数来说已经足够了。虽然只有X方向内存具有24位宽度，大的数据通路寄存器文件保证了不会有额外的延迟发生。AudioDE处理器的存储器位于数据通路之外，可以完全根据成本优化或系统集成的需求设计。每个数据存储器都有它自己的地址产生器，可以寻址15位宽的地址空间。需要注意的是这个地址空间对于单任务是固定的，整体的地址空间可以通过子系统存储器管理功能实现。

地址产生器在所有模式下能够进行单周期操作，包括自动取模地址和可编程的位反转地址，后者对于FFT算法尤其重要。操作寄存器文件的主要和次要



指针个数都经过优化以减少功耗，面积和控制复杂度。数据流接口是AudioDE中特有的多媒体应用特性，无论从编码或综合的角度都能够提高性能和减少复杂度。这个接口使得AudioDE处理器可以被当作一个DMA的从设备进行连接，并且无需与主系统作进一步的交互而直接处理比特流数据。

其数据通路的API提供了对可变长比特流访问自动数据对齐的功能，以及对恒比特率（CBR, Constant Bit-Rate）音频编码结构中常见的乱序传输提供数据流缓冲和重绕的功能。

功耗控制特性

OptimoDE数据引擎中常用的低功耗设计方法是通过尽可能发掘所有可用的并行机制以降低所需的运行频率。这样导致了非常密集的代码和对资源非常高的使用，同样也减少了资源不必要的开关触发。另外对于循环体，这样的代码密度也很适合缓存或其他内核以外的系统存储管理的优化手段。

除了这些结构优点之外，AudioDE处理器还拥有另外的功耗控制技术。其中首要的是采用了ARM Artisan物理IP部门提供的特殊的低功耗库用于硅片映射：

- AudioDE处理器完全符合业界标准的门控时钟插入方法。
- AudioDE处理器针对数据流接口和外部存储器的使用进行特殊设计。一旦数据无效，AudioDE处理器自动进入低功耗等待模式。
- OptimoDE编译器对整个指令和数据流水线完全可见。所以对于数据何时产生或何时输入完全可控，这样避免了流水线上不可预料的停滞和额外的逻辑引入。

系统集成

虽然AudioDE处理器本身只是数据引

擎，它是为高性价比系统的集成所特别设计。出于这个原因，除了核心之外，还有几个组件来提供系统中已经以模块形式存在的功能，比如DMA控制器。

同样的优化自由度也提供给系统设计者以满足成本敏感的系统存储器设计需求。以下略述了两个集成系统的范例。这两个范例都支持标准的OptimoDE组件，比如专为OptimoDE提供的AMBA集成工具（AIKO）。

最优成本方案

本系统着重解决功耗与面积的优化，而不偏重应用的灵活性。假设这是可作为低功耗便携式基于闪存的音频播放器产品。在这种情况下，AudioDE处理器与专有存储器集成在一起。因为存储器尺寸和拓扑结构可以自由选择以达到最优的结果，这种配置从而可以将功耗降至最低。另外，数据流接口的存在完全免除了系统中所需的快速交互。AudioDE可以像一个高效率的智能外设一样操作。AIKO接口保证了存储器一直可以被系统访问。

灵活的存储器共享结构

这种系统旨在为应用开发提供最大的灵活性，最小的存储器功耗。这种应用的一个范例是支持多种编码的音频播放器。这种情况下，AudioDE所有的存储器接口被连接到一个可变的单层共享存储仲裁器，像其他系统中的资源一样，比如主总线和DMA控制器。这种特殊的AudioDE子系统还具有特别的存储管理功能以实现存储和任务地址空间的动态分配。

MP3应用示例

为了建立一个性能参考试验，针对AudioDE音频数据处理引擎编辑一个符合ISO/IEC标准的MP3解码结构。这个架

构分为两部分，首先是比特流解码器，然后是带有反量化功能的立体声处理和一个大的两阶合成滤波器组。

第一部分的解码器利用了AudioDE丰富的比特流访问API。因此，产生的代码非常的简单、紧凑和高效率。第二部分的算法充分利用了AudioDE处理器对基数算法的强大处理能力。两部分都使用C语言编写源码。

这个架构从320kbps的比特流解码48kHz采样率的立体声信号只需要运行在8MHz的频率下。这个解码架构使用22KByte的程序存储器和22KByte的数据存储器。结合上面提到的功耗控制方法，使用ARM的低功耗物理库，在0.13um-CMOS工艺下，实现核心大小45k门，每MHz功耗仅为0.1mW。当解码上述音频码流时，AudioDE核心功耗尽在0.8mW以下，同时经过优化的子系统总功耗应低于2mW。

结论

AudioDE音频数据处理引擎是一个非常紧凑，并且功耗效率极高的处理器。它源于OptimoDE架构和提供的标准配置和开发工具。基于AudioDE处理器，实现了一个可执行标准MP3基准架构的设计，相对于其他可选择的解决方案，门级数和资源减少了一半，同时处理能力增加了一倍。



References

1. ISO/IEC, "MPEG1 11172-3: Audio coding", 2000(e).
2. ISO/IEC, "MPEG2 13818-3: Audio coding", 2000(e).
3. ISO/IEC, "MPEG2 13818-7: Advanced audio coding", 2000(e).
4. ISO/IEC, "MPEG2 14496-3: Audio", 2000(e).