

part1 HiFi2 Overview

[返回目录位置](#)

Cadence HiFi 2/EP DSP（后面简称“HiFi2”或“hifi2”）是一款高度优化的音频处理器。其中 HiFi EP DSP 拓展配置选项拓展了 HiFi 2 DSP ISA，支持**预测硬件预取、32x24 位乘法/累加运算、循环缓冲器加载和存储以及双向移位**。

HiFi 2 DSP ISA 在网页中是一个额外的库，里面有许多函数，其中函数中带 AE 前缀的就是 HiFi 的函数，不带 AE 前缀的就是 ISA 的函数。

AE_ZEROP48	set AE_PR bits to zero	
AE_ZEROQ56	set AE_QR bits to zero	HiFi内部
AE_ZEXT16	Zero Extend 16-bits	
ALL4	All 4 Booleans True	ISA
ALL8	All 8 Booleans True	
AND	Bitwise Logical And	
ANDB	Boolean And	
ANDBC	Boolean And with Complement	

同时我在文档中也看到 ISA 可以带来一些拓展的乘法，比如 32X24-bit 乘法只有拓展了 ISA 才可以使用。

hifi2 主要有两个主要组件组成：一个主要在 24 位数据项上运行的 DSP 子系统，以及一个辅助比特流访问和可变长度（[知霍夫曼](#)）编码和解码的子系统。在后面的图片里面会有展示，两个组件用“slot + 数字”来表示。

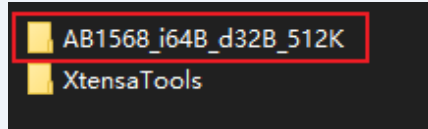
1.1 本用户指南的目的

本指南概述了 hifi2 架构及其指令集。它将通过识别一些通常用于优化算法的技术来帮助程序员使用 hifi2。它提供了通过使用 hifi2 的指令，内部函数，原型和基元来实现改进性能的指南。本指南还可作为 C/C++ 在 C/C++ 软件开发中使用 hifi2 功能的适当方法的 C/C++ 使用参考。本指南还将帮助希望向 hifi2 架构添加其他指令的 Xtensa hifi2 用户。

为了最有效地使用本指南，强烈建议对 Xtensa 软件开发流程有基本的熟悉程度。有关更多信息，请参阅 Xtensa 软件开发工具包用户指南（*Xtensa Software Development Toolkit User's Guide*）。

文档的位置按照目前达发项目安装文档放置在“[C:\msys64\home\Administrator\airoha_sdk_toolchain\xtensa\RI-2021.8-win32](#)”，因为我的 xtensa 依赖于 msys 搭建的 linux 环境，所以安装的位置有所不同。符号 <xtensa_root> 是指用户的 Xtensa 配置的安装目录，按照书中说法我的 <xtensa_root> 就是上面的路径 (C:\msys64\home\Administrator\airoha_sdk_toolchain\xtensa\RI-2021.8-win32\AB1568_i64B_d32B_512K) 。

在我的 <xtensa_root> 里面有两个文件夹，一般看的是红框的那个。



1.2 安装概述

要安装 HiFi 2/EP 配置，请按照 Xtensa 开发工具安装指南 (*Xtensa Development Tools Installation Guide*) 中所述的相同过程进行操作。HiFi 2/EP 包含文件位于以下目录和文件：

```
1 <xtensa_root>/xtensa-elf/arch/include/xtensa/config/defs.h
2 <xtensa_root>/xtensa-elf/arch/include/xtensa/tie/xt_hifi2.h
```

1.3 HiFi2/EP 结构概述

hifi2 是一种 SIMD（单指令/多数据）处理器，可以同时并行处理两个数据项。例如，它允许一个操作并行执行两个 24 位加法，每个加法占用 48 位 AE_PR 寄存器的一半。该架构青睐的数据大小为操作数 24 位和累加器 56 位，提供 8 个额外的保护位。**hifi2 还支持 16 位负载和存储，可拓展为 AE_PR 寄存器中可用的 24 位。**除了双 24X24 位之外，hifi2 还支持双 32X16 位乘法器，但用于 32 位操作数必须来自累加器寄存器，因此 **32X16 位乘法器的效率通常低于 24X24 位**。支持使用 32 位累加器的单个 16X16 位乘法器，从而更好地支持语音算法。HiFi EP 增加了对 32 x 24 位乘法器的支持。

从上面的综述来看，效率最高的还是 24X24 位的操作，只要一个寄存器就可以完成，所以优化的时候优先考虑这种方法。

hifi2 是一种 VLIW 架构，支持并行执行两个操作。DSP 加载和存储、比特流和霍夫曼操作以及核心操作在 VLIW 指令的 slot0 中可用。DSP MAC 和 ALU 操作在 slot1 中可用。

ALU 操作是什么意思？


HiFi 2/EP 支持**缓存或本地存储器**，具有 Xtensa 提供的全部灵活性。配置可以具有其中之一或两者，并且可以对指令和数据做出不同的选择。Cadence 提供的音频包不使用  DMA（直接内存访问）。因此，**大多数客户要么使用缓存，要么使本地内存足够大以覆盖所需的应用程序。**

Figure 1-1 显示了 HiFi 2/EP DSP 添加到 Xtensa LX 处理器的自定义状态、寄存器文件和执行单元。

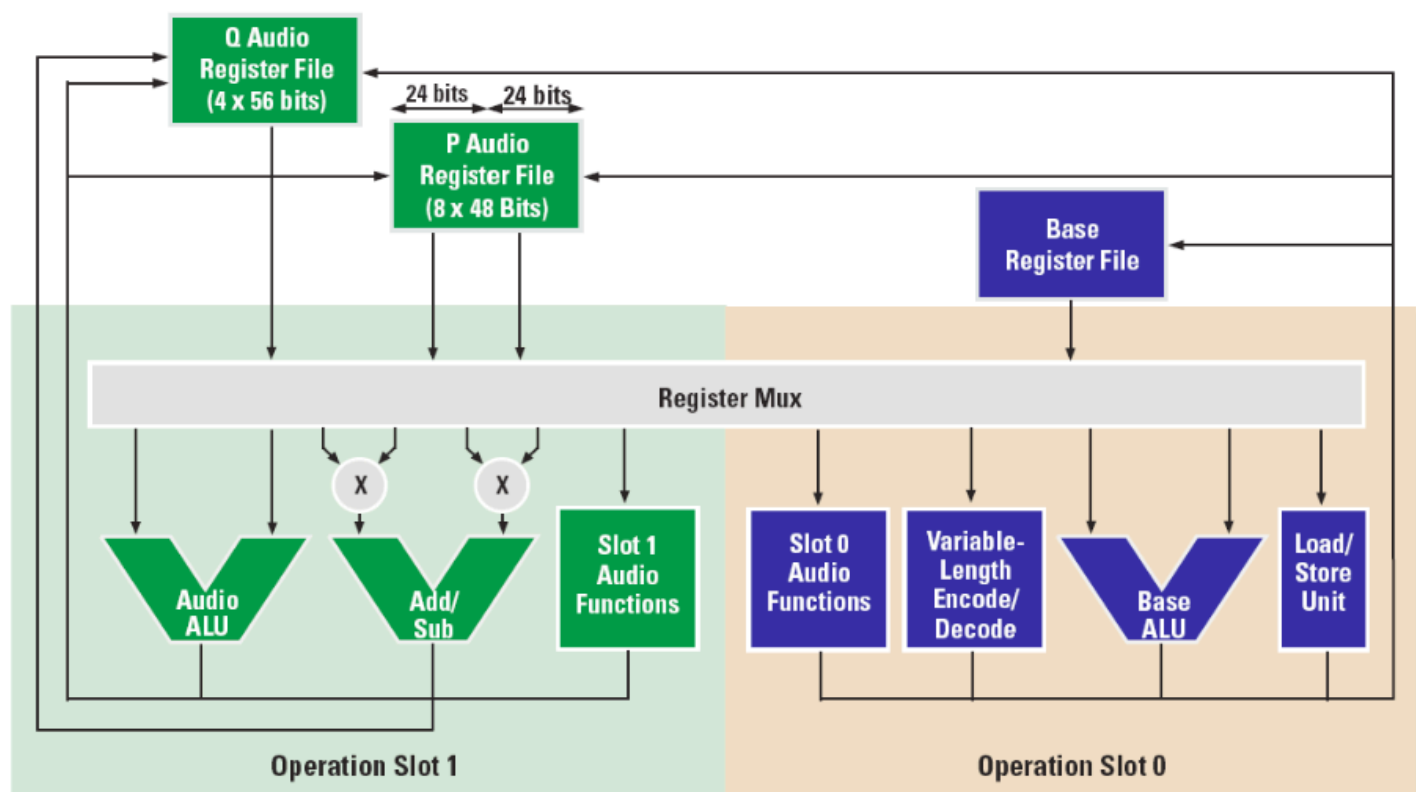


Figure 1-1 HiFi 2/EP DSP Components

以下是 DSP 子系统的主要硬件资源：

- 双乘法/累加单位 (Add/Sub)
- 寄存器文件 AE_PR 以保存一对 24 位数据项 (P Audio)
- 寄存器文件 AE_QR 以保存 56 位累加器值 (Q Audio)
- 算术逻辑单元 (ALU) ， 用于对 AE_PR 值和 AE_QR 值进行操作
- 移位单元以对 AE_PR 值和 AE_QR 值进行操作

算术逻辑单元应该就是 slot1 里面的 ALU，然后移位单元应该就是 slot1 里面的 Audio function？

是的，ALU 就是算法逻辑单元，移位单元应该也属于 ALU。

怎么从图片的角度理解双乘法？

上图两个“X”的部分就是双乘法。

乘法/累加单元还支持将 AE_QR 寄存器的 32 位值乘以 AE_PR 寄存器的 16 位值，并将 48 位结果写入或累加到 AE_QR 寄存器。

EP 版本支持 32X24 位乘法，也是 AE_QR 的 32 位值乘以 AE_QR 的 24 位值，然后将 56 位结果写入或累加到 AE_QR 寄存器中。

DSP 子系统的指令是根据分为两组的操作构建的：slot0 集和 slot1 集。在每个执行周期中，可以根据机器代码中表示的静态捆绑，独立执行每个集合中的零个或一个操作。因此，例如，**负载操作可以与乘法/累加操作同时执行，因为负载位于 slot0 中，而乘法/累加操作位于 slot1 中。**

1.4 预取 (EP only)

这个预取指令在之前和原厂的沟通中一般是不动这个的，所以这个当做了解就可以了。

HiFi EP 包括一个预取选项，适用于内存延迟较长的系统。当 HiFi EP 处理器检测到缓存未命中流（数据或指令）时，它可以推测性地提前预取多达**四行缓存**，并将它们放置在靠近处理器的缓冲区中，或者将数据侧选择性地放入 L1 数据缓存中（不支持直接预取到 L1 指令缓存中）。此外，用户可以手动发出预取指令。

默认情况下，硬件预取在 Cadence 提供的重置代码中启用，设置较低。默认情况下，在支持它的配置上，数据预取将放入 L1 数据缓存中。可以使用以下 HAL 调用显式禁用预取或增加其在代码不同部分中的主动性。使用更积极的预取，硬件将在检测到流时更早地预取，并将预取更多行。假设有足够的总线带宽，性能将通过更积极的预取而提高，但系统将需要更多的带宽。预取指令和数据可以单独控制。

```
1 #include <xtensa/hal.h>
2 int xthal_set_cache_prefetch(int mode);
```

返回的值不用于直接使用或解释；但是，它适合传递给对 `xthal_set_cache_prefetch()` 的后续调用。

mode 参数可以是以下参数之一：

- 上一次调用 `xthal_set_cache_prefetch()` 或 `xthal_get_cache_prefetch()` 返回的值。
- 以下常量之一，适用于指令和数据缓存：
 - XTHAL_PREFETCH_ENABLE (启用缓存预取)
 - XTHAL_PREFETCH_DISABLE (禁用缓存预取)
- 两个缓存预取模式常量的按位 OR，一个用于指令缓存：
 - XTHAL_ICACHE_PREFETCH_OFF (禁用指令高速缓存预取)
 - XTHAL_ICACHE_PREFETCH_LOW (启用，低优先级预取)

- XTHAL_ICACHE_PREFETCH_MEDIUM (启用, 中优先级预取)
- XTHAL_ICACHE_PREFETCH_HIGH (启用, 高优先级预取)
- XTHAL_ICACHE_PREFETCH(n) (显式将 PREFCTL 寄存器的 InstCtl 字段设置为 0..15。有关详细信息, 请参阅 Xtensa 微处理器数据手册(*Xtensa Microprocessor Data Book*)中预取单元选项一章的预取架构添加部分。)
- 一个用于数据缓存:
 - XTHAL_DCACHE_PREFETCH_OFF (禁用数据缓存预取)
 - XTHAL_DCACHE_PREFETCH_LOW (启用, 低优先级预取)
 - XTHAL_DCACHE_PREFETCH_MEDIUM (启用, 中优先级预取)
 - XTHAL_DCACHE_PREFETCH_HIGH (启用, 高优先级预取)
 - XTHAL_DCACHE_PREFETCH(n) (显式将 PREFCTL 寄存器的 DataCtl 字段设置为 0..15。有关详细信息, 请参阅 Xtensa 微处理器数据手册(*Xtensa Microprocessor Data Book*)中预取单元选项一章的预取架构添加部分。)
 - XTHAL_DCACHE_PREFETCH_L1_OFF (预取数据以仅预取缓冲区)
 - XTHAL_DCACHE_PREFETCH_L1 (在支持它的配置上, 直接预取到 L1 数据缓存)

为了便于模拟, 还可以使用 `xt-run --prefetch=0` 标志在模拟器中禁用预取。从模拟命令行禁用预取将覆盖任何 HAL 调用。

1.4.1 软件预取

软件预取可用于数据或指令。它们可以作为硬件预取的补充或代替硬件预取使用。如果禁用了硬件预取, 软件预取仍处于启用状态。

考虑一个执行能量计算的简单示例。您可以选择在循环之前放置一些显式预取指令, 以设定硬件预取程序的种子。否则, 根据模式, 硬件预取可能会延迟预取, 直到第二次未命中之后。

```
C
1  __builtin_prefetch(&ap[0]);
2  __builtin_prefetch(&ap[XCHAL_DCACHE_LINESIZE]);
3  __builtin_prefetch(&ap[2*XCHAL_DCACHE_LINESIZE]);
4  for (i=0; i<n; i++) {
5      sum += ap[i]*ap[i];
6  }
```

你可能还希望将预取指令直接放在循环中。这样，既可以比硬件预取程序更积极地预取，也可以预取除硬件预取程序检测到的 stride-1 引用以外的模式。相反，将预取指令放在循环中会产生指令开销，无论循环是否实际遭受缓存未命中的影响。

通常，鉴于硬件预取程序的有效性，应明智地使用软件预取。仔细比较使用和不使用软件预取之间的性能。

一般来说预取这里都不使用，所以这个可能后面的时候再去探索会好一些。

1.5 HiFi2/EP 指令设置概述

HiFi 2/EP DSP 建立在基准 Xtensa RISC 架构之上，该架构实现了一组丰富的通用指令，针对高效的嵌入式处理进行了优化。HiFi 2/EP 的强大功能来自全面的 DSP 和音频指令集。各种加载/存储操作支持六种不同的寻址模式，支持 16/24 位标量和矢量数据类型以及 32/56 位标量。**矢量数据管理支持选择操作和移位。**

乘法运算包括 24x24 位、32x16 位、16x16 位和 EP 上的 32x24 位。所有乘法运算都会在 AE_QR 累加器寄存器文件的元素中产生更高精度的结果。由于累加器只能保存一个结果，因此**不支持真正的 SIMD 乘法**。相反，HiFi 2 / EP 支持双乘法，其中两个乘法的结果在与乘法相同的指令中相加或相减，然后再添加到累加器中。HiFi 2/EP 支持一组丰富的乘法指令，允许程序员选择从哪里选择乘法的每个操作数（操作数的下半部分或上半部分），是加还是减总和，是实现乘法、乘法-加法还是乘法-减法，以及是实现整数还是小数乘法。

一组比特流和可变长度指令允许高效访问串行比特流，包括霍夫曼编码和解码。

1.6 HiFi2/EP 设计模型

HiFi 2/EP 包含许多数据类型，对应于架构支持的不同精度的标量和矢量版本。**这些数据类型可以像标准 C 数据类型一样加载和存储，并且可以在 C 和 C++ 中使用内置运算符重载进行操作**，如第 3 节中所述。XCC 编译器将对这些隐式操作执行常规优化，例如自动推断使用更新加载和存储指令。为了获得额外的控制，每个 HiFi2/EP 指令都可以**使用具有相同名称的[知内联函数](#)直接访问**，并采用适当的数据类型。程序员可以自由混合和匹配内部和隐式负载、存储和运算符的使用。

1.7 小结