CASE STUDY

Optimizing PPA with RISC-V custom extensions in TWS earbuds

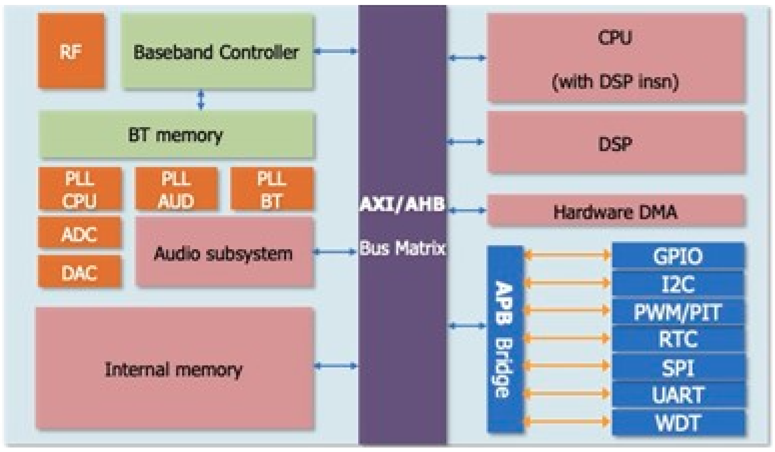
A look at using custom extensions in a RISC-V processor to enable power, performance, and area (PPA) optimized true wireless stereo earbuds SoC design.

John Min

Introduction

A common goal for many [IC] designs is achieving an optimum combination of power, performance, and area (PPA). This article examines the components and design of a currently shipping true wireless stereo (TWS) chip design, indicating the engineering tradeoffs made to achieve the PPA demanded by the final earbuds. In particular, we will also look at the RISC-V ISA (instruction set architecture) extensions that helped to dramatically reduce power while achieving the desired performance at no extra cost.

A TWS chip is mostly a Bluetooth RF transceiver. It comprises a 128-tap finite impulse response (FIR) filter, 128-tap least means square (LMS) adaptive filter, an active noise control (ANC) hybrid function and the computational elements and buses needed to compute these functions. In this design, the first three of the four elements would require the greatest amount of computing resource: The FIR filter will demand 1,693 instruction cycles, the LMS filter another 1,820 cycles, and the ANC function, a whopping 10,404 cycles. The system block diagram illustrates the major components of the TWS design with analog functions on the left and digital on the right. The Bluetooth transceiver (top left) consists of an RF baseband controller and 64kbytes of exchange memory. The transceiver and audio subsystem below is complex because the TWS design has two independently operating speakers and microphones. The phase lock loop (PLL) for each must be synchronized to ensure each is running at the same frequency.



Key components of a TWS design, with analog functions on the left and digital on the right. (Image: Andes Technology)

A further complication is the Bluetooth system operates on 16-bit data in little-endian format. The audio system requires 24-bit data in big-endian format, thus requiring a compute-intensive transformation. The active noise control (ANC) function puts the greatest demand on computing resources. It requires very large numbers of calculations like multiplication and additions. The ANC element interacts with a high-precision DAC and a low-latency ADC. Real-time analog sampling requires one sample in for every one sample out.

In the digital section on the right is a RISC-V CPU with P (DSP/SIMD) extensions and standalone DSP for handling noise control and audio codec computations. In addition, the design contains a small amount of SRAM hardware DMA and AXI/AHB bus matrix. The design comes with the general purpose I/O functions such a consumer device demands: GPIO, I2C, real-time clock, UART among others.

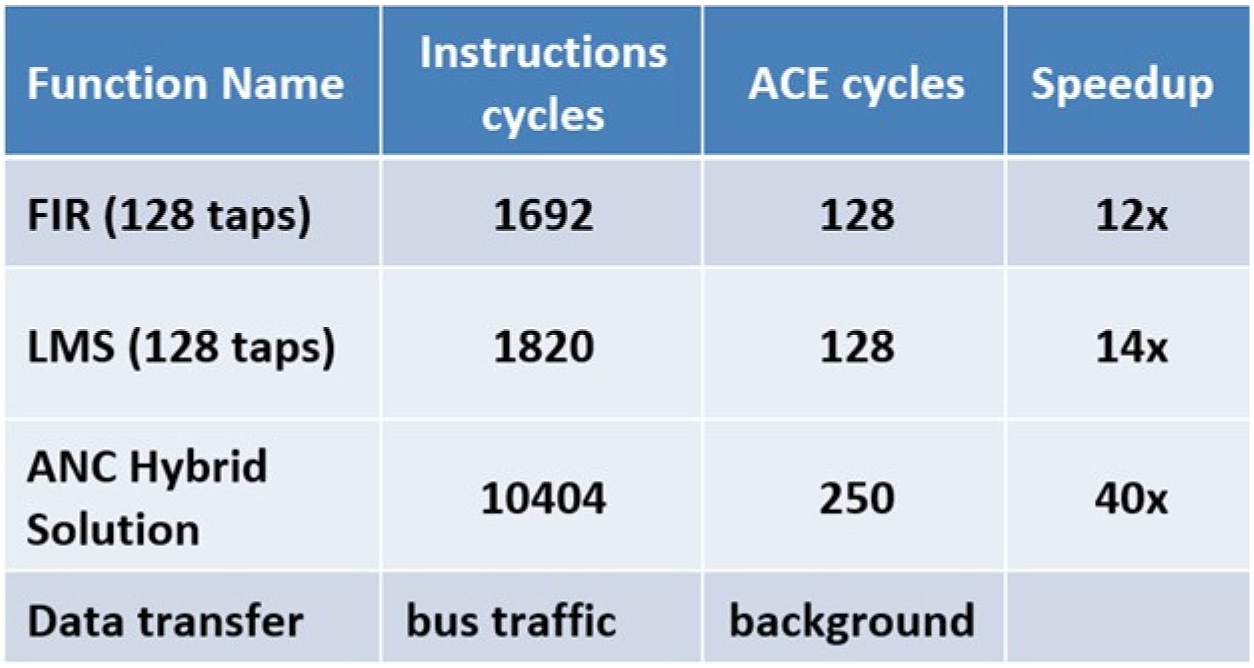
To perform the most complex TWS function, a hybrid active noise control circuit, demands a high tap number filter and, in this example, a 128-tap FIR filter and 128-tap least mean square filter. To process these functions a total of 896 additions, and 896 multiplications are needed for every sample. In this example, the ADC and DAC are running at 192 kHz which is roughly one sample approximately every 5.2 microseconds.

In the block diagram shown, the computing resources must perform the following functions for processing the ADC and DAC operations: a DMA to load the sample, followed by audio codec computation and Bluetooth and housekeeping operations. Next comes the heavy processing needed for the FIR and LMS filters. In addition to these operations, active noise control adds considerable computing requirements as well, primarily large numbers of multiplication and additions.

Custom RISC-V extensions reduce the number of instruction cycles

Recognizing the need for more computing requirements the design team determined to create custom extensions to the RISC-V processor dedicated to handling the FIR and LMS filter, one added instruction for each. A multiplier, adder, and storage were developed to handle the ANC operations. These extensions were created using the Andes Custom Extension (ACE) and CoPilot tools. The tools eliminated the time-consuming tasks of integrating the custom instructions into the design and verification flow, thus minimizing the time the design team needed to implement the modifications.

The results are shown in the table. The 128-tap FIR filter in pure C code will require around 1600 instruction cycles. Adding the custom instruction reduces the number to 128 cycles, a 12-times speedup. Likewise, cycles needed to compute the 128-tap LMS filter drops form 1820 cycles to 128 cycles, a 14-times speedup. The most dramatic acceleration occurs for the ANC function, which drops from 10,404 cycles to 250, a huge 40-times reduction. The CPU cycle counts for the direct memory access function applied to TWS platform will depend on the SoC (system on chip) architecture. Normally the heavier bus traffic it has, the more cycles will be needed. The ACE and CoPilot tools assigned the added instruction background attributes. Thus, the customized instructions run in parallel to ensure optimum execution. As a result, the CPU can be run at a slower 80 MHz clock frequency, thus reducing power for longer battery life.



Function Name	Instructions cycles	ACE cycles	Speedup
FIR (128 taps)	1692	128	12x
LMS (128 taps)	1820	128	14x
ANC Hybrid Solution	10404	250	40x
Data transfer	bus traffic	background	

The table shows speedup enabled by custom extensions (under the ACE cycles column). The most dramatic acceleration occurs for the ANC function, a 40-times reduction. (Image: Andes Technology)

While we can't comment on the actual results of this proprietary design, the data shown in the table is typical of the improvements that adding custom extensions to a design can produce. In addition to the performance boost, design teams using custom extensions also benefit from a circuit that runs at reduced clock speeds to save power and that uses far fewer gates to occupy a smaller silicon footprint.

Another significant benefit of using custom extensions is that competitors cannot easily reverse-engineer the design. Finally, for designers concerned about the added time custom instructions will have on design verification, the experience of the 60 percent of customers that have successfully used the ACE tool testify to this not being a concern. With this particular design, the final chip successfully taped out with no design spin and is currently shipping in the Razer's True Wireless Pro earbuds.

Welcome to the Open Era of Computing!

Join us and see how open technical collaboration along with the support of many RISC-V programs can accelerate your business.

riscv.org /membership/ | info@riscv.org