Choosing the Best Processor for Your Audio DSP Application

Paul Beckmann
DSP Concepts
About Paul Beckmann

1984-1992. M.I.T. Received SB, SM, and PhD in EE specializing in DSP under Prof. Bruce Musicus.
2001-2003 Enuvis Corporation
2003- DSP Concepts
   \textit{Founder and CTO}
   \textit{Providing tools and design services to audio product developers}

Key skills
\textit{DSP algorithms and optimization}
\textit{Audio product design}

Passionate about audio design tools
Outline

Problem statement
Comparing processor architectures
DSP benchmarks
Benchmarking with Audio Weaver
Conclusion
The Audio Product Development Challenge

Complexity of audio products is increasing
Development times are shrinking
Multiple audio formats
Connectivity
Software updates
Have to integrate 3\textsuperscript{rd} party IP
Size and power requirements
Multiple Skill Sets Required

Electrical
Mechanical
Acoustical
User interface

Audio processing software requires multiple skills

Audio DSP: 872
Audio Engineer: 30,915
Embedded Software: 97,975

Audio DSP 59 261 129
Audio Engineer 11
Embedded Software 11
Analog Devices
  SHARC, Blackfin, SigmaDSP
Texas Instruments
  C55, C67x, C66x
ARM
  ARM 9 / 11
  Cortex-M4 / M7
  Cortex-A8 / A9 / A15 / etc.
Intel x86 / x64
And embedded cores
  Tensilica, CEVA, and ARC
System Design Issues

Peripherals
- Serial ports
- Connectivity
- Sample rate converters
- DMA

Memory
- Built in RAM / Flash
- External memory

Size

Power consumption
Connected products require an MCU

- USB
- Wi-Fi / Ethernet
- Etc.

Signal processing needs for multimedia products are growing

- Audio, video, microphones, etc.

IoT products packed with sensors

Can a single chip handle all functions?
The ABC’s of Processor Choice

Awareness – I didn’t know I could use this processor for audio
Benchmarking – Will my application fit?
Cost – What is the overall system cost?
How We’ll Compare Architectures

Define a true DSP
Then consider various processor families

ADI SHARC
ADI Blackfin BF53x / BF70x
ARM Cortex-M4 / M7
ARM Cortex-A8/9/15
A True DSP

- Single cycle multiply - accumulate
- Load and stores in parallel with computation
- Zero-overhead loops
- Circular and bit-reversed addressing

- Accumulator with guard bits
- Fractional and saturating math

The Test: Can you do an FIR filter in roughly 1 cycle/tap?
SHARC Core

- A True DSP
- Floating-point support
- 2-way SIMD
- Large 5 Mbit internal memory
- No cache

- Hardware accelerators for FIR, IIR, and FFT
- 4 stereo sample rate converters
- S/PDIF transceiver
- Lots of I2S and TDM I/O
- External SDRAM interface

Starting at $8.00
Blackfin BF5xx Core

- True DSP
- 32-bit internal registers
- Dual 16-bit MAC
- Data and program cache
- Up to 148 kbytes internal RAM
- USB and Ethernet support

Starting at $2.00
Blackfin BF7xx Core

- True DSP
- 32-bit internal registers
- Dual 32-bit MAC
- Single 16-bit MAC
- Data and program cache
- 136 kbytes internal RAM
- USB support

Starting at $4.00
ARM Cortex-M Series

The Underlying Core
Cortex-M3 released 2004
  Traditional microcontroller
  32-bit native data type. Fixed-point
Cortex-M4 released in 2010
  Digital signal controller
  Adds floating-point and some DSP capabilities
Cortex-M7 announced Sept. 2014
  Further architecture improvements for DSP

Many Licensees
ST Microelectronics, Freescale, NXP, Atmel, Texas Instruments, Analog Devices, Infineon, etc.

Other Highlights
Basic audio I/O (I2S)
Low power variants
USB and Ethernet

M4 Starting at $1.50
## Cortex-M Core Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Cortex-M3</th>
<th>Cortex-M4</th>
<th>Cortex-M7</th>
<th>True DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single cycle MAC</td>
<td></td>
<td>Fixed-point only</td>
<td>Fixed and floating-point</td>
<td>Y</td>
</tr>
<tr>
<td>Floating-point</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Fractional and saturating math</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SIMD operations</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Load and store in parallel with math</td>
<td></td>
<td>Y (1)</td>
<td>Y (2)</td>
<td></td>
</tr>
<tr>
<td>Zero overhead loops</td>
<td>Y</td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Accumulator with guard bits</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Circular and bit-reversed addressing</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
</tbody>
</table>
Processing engines for a wide variety of consumer products.
“System on Chip” combining processing, I/O, and graphics.
Excellent connectivity
Power efficient
Strong price pressure and devices available from a variety of vendors
Getting better at audio processing

Starting at $5.00
ARM Cortex-A Architecture

Multicore designs up to 2.5 GHz
   1 → 2 → 4 processors

Family of processors
   A5, A7, A8, A9, A15, etc.

NEON Technology
   SIMD multimedia extensions including
   4 way floating-point
   16 x 128-bit registers
   Provides a significant computational
   boost to audio applications

Versions with audio specific
peripherals starting to appear
A8/A9/A15 Differences

Cortex-A8
- Single core up to 1 GHz
- 13 stage pipeline

Cortex-A9
- Multicore up to 2 GHz
- Out of order execution
- 8 to 11 stage pipeline

Cortex-A15
- Multicore up to 2.5 GHz
- Highly out of order execution
- 17 to 25 stage pipeline
- Twice the memory bandwidth from core to cache
Introducing Benchmarks

FIR Filter
- Equalizers, adaptive filters
- Room correction

Biquad filter
- Audio EQ work horse
- PID loops and motor control

FFT
- Frequency domain processing
Understanding Benchmarks

Considerations
Memory accesses
MACs
Specialized addressing modes used?

Analysis
N-point FIR – Memory intensive
2N+3 memory accesses
N MACs

Biquad Filter Stage – MAC intensive
2 memory accesses
5 MACs

FFT Butterfly – Balanced
10 memory accesses
10 math operations (+ or x)
Write in ASM code
Utilize SIMD whenever possible
Benchmarks using internal DSP Concepts libraries

lcntr=r1, do _sampleLoopEnd until lce;
  f15=f0*f11, r8=dm(i4,m4);
  f8=f3*f5, f15=f8+f15, pm(i12,m12)=r12;
  f10=f0*f6, f2=f8+f15, r0=r3;
  f8=f2*f7, f15=f10+f14, r3=r2;

_sampleLoopEnd:
  f14=f3*f4, f12=f8+f15;
Write in ASM code
Utilize SIMD whenever possible
Internal DSP Concepts libraries and published ADI libraries

```asm
biquad_filter_start:
    a0 += r3.h * r4.h, a1 += r3.h * r4.l(m) || r5 = [fp - 12] || NOP;
a1 += r4.h * r3.l (m) || r6 = [i3++m3] || NOP;
a0 += r5.h * r6.h, a1 += r5.h * r6.l(m)|| r3 = [fp - 16] || NOP;
a1 += r6.h * r5.l (m) || r4 = [i3++m3] || r7 = [p3 + 0];
a0 += r3.h * r4.h, a1 += r3.h * r4.l(m)|| r5 = [fp - 20] ;
a1 += r4.h * r3.l (m) || r6 = [i3++m3] || [p3 + 4] = r7;
a0 += r5.h * r6.h, a1 += r5.h * r6.l(m)|| NOP || [p3 + 0] = r0;
a1 += r6.h * r5.l (m) || r4 = [fp - 4] || r0 = [i0 ++ m0];
    a1 = a1 >>> 15 || r7 = [p3 + 8];
a0 += a1 || [p3 + 12] = r7;
```

// And more
Cortex-M4 / M7 Optimization

Write in C code
Loop unrolled
Heavy register reuse to minimize data accesses
Different code for M4 and M7
Part of the CMSIS DSP library provided by ARM

```c
while(sample > 0u) {
    /* y[n] = b0 * x[n] + d1 */
    /* d1 = b1 * x[n] + a1 * y[n] + d2 */
    /* d2 = b2 * x[n] + a2 * y[n] */

    /* Read the first 2 inputs. 2 cycles */
    Xn1 = pIn[0];
    Xn2 = pIn[1];

    /* Sample 1. 5 cycles */
    Xn3 = pIn[2];
    acc1 = b0 * Xn1 + d1;
    Xn4 = pIn[3];
    d1 = b1 * Xn1 + d2;
    Xn5 = pIn[4];
    d2 = b2 * Xn1;

    // and on and on
```
Cortex-A Optimization

Write in C code with intrinsics
Loop unrolled
Heavy register reuse to minimize data accesses
Same code (mostly) used for all Cortex-A processors
Internal DSP Concepts libraries

while (blockSize >= 16)
{
    w6 = vld1_dup_f32(src);
    src += srcInc;
    y6 = vmul_f32(b0, w6);
    w1 = vmla_f32(w1, a1, w0);
    w2 = vmla_f32(w2, a2, w0);
    y1 = vmla_f32(y1, c1, w0);
    y2 = vmla_f32(y2, c2, w0);
    vst1_lane_f32(dst, y0, 0);
    dst += dstInc;

    w7 = vld1_dup_f32(src);
    src += srcInc;
    y7 = vmul_f32(b0, w7);
    w2 = vmla_f32(w2, a1, w1);
    w3 = vmla_f32(w3, a2, w1);
    y2 = vmla_f32(y2, c1, w1);
    y3 = vmla_f32(y3, c2, w1);
    vst1_lane_f32(dst, y1, 0);
    dst += dstInc;

    // and on and on
Ease of Optimization

Cortex-M > SHARC > Cortex-A > BF70x > BF5xx

(easiest) (hardest)
FIR Benchmarks

256 sample block size
Clock cycles are shown
Floating-point for all except Blackfin (Q31)
Measured using Audio Weaver

<table>
<thead>
<tr>
<th>Num Taps</th>
<th>Cortex-M4</th>
<th>Cortex-A8</th>
<th>Cortex-A9</th>
<th>Cortex-A15</th>
<th>Blackfin 5xx</th>
<th>Blackfin 70x</th>
<th>SHARC 21489</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6743</td>
<td>6467</td>
<td>6315</td>
<td>2673</td>
<td></td>
<td></td>
<td>1954</td>
</tr>
<tr>
<td>10</td>
<td>9871</td>
<td>9793</td>
<td>9245</td>
<td>5142</td>
<td></td>
<td></td>
<td>2473</td>
</tr>
<tr>
<td>20</td>
<td>15650</td>
<td>13598</td>
<td>14338</td>
<td>5031</td>
<td>27404</td>
<td>14456</td>
<td>3777</td>
</tr>
<tr>
<td>50</td>
<td>35801</td>
<td>29310</td>
<td>32799</td>
<td>10267</td>
<td>27404</td>
<td>14456</td>
<td>7677</td>
</tr>
<tr>
<td>100</td>
<td>67833</td>
<td>53913</td>
<td>62145</td>
<td>15525</td>
<td></td>
<td></td>
<td>14210</td>
</tr>
</tbody>
</table>
FIR Analysis

Cycles per sample per tap

<table>
<thead>
<tr>
<th>Cortex-M4</th>
<th>Cortex-A8</th>
<th>Cortex-A9</th>
<th>Cortex-A15</th>
<th>Blackfin 5xx</th>
<th>Blackfin 70x</th>
<th>SHARC 21489</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.65</td>
<td>2.11</td>
<td>2.43</td>
<td>0.61</td>
<td>2.14</td>
<td>1.13</td>
<td>0.56</td>
</tr>
</tbody>
</table>
# Biquad Benchmarks

<table>
<thead>
<tr>
<th>Num Stages</th>
<th>Cortex-M4</th>
<th>Cortex-A8</th>
<th>Cortex-A9</th>
<th>Cortex-A15</th>
<th>Blackfin BF5xx</th>
<th>Blackfin BF70x</th>
<th>SHARC 21489</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4480</td>
<td>4867</td>
<td>4326</td>
<td>2439</td>
<td>4650</td>
<td>3338</td>
<td>1455</td>
</tr>
<tr>
<td>4</td>
<td>16700</td>
<td>16712</td>
<td>17750</td>
<td>9040</td>
<td>5405</td>
<td></td>
<td>5405</td>
</tr>
<tr>
<td>8</td>
<td>32900</td>
<td>33354</td>
<td>32933</td>
<td>17825</td>
<td></td>
<td>10650</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>49100</td>
<td>49274</td>
<td>50243</td>
<td>26664</td>
<td></td>
<td></td>
<td>15958</td>
</tr>
</tbody>
</table>

256 sample block size  
Clock cycles are shown  
Measured using Audio Weaver  
Mono channel processing

Blackfin notes  
Uses 32x32+64 math  
Additional overhead for shifting of data
## Biquad Analysis

### Cycles per sample per stage

<table>
<thead>
<tr>
<th></th>
<th>Cortex-M4</th>
<th>Cortex-A8</th>
<th>Cortex-A9</th>
<th>Cortex-A15</th>
<th>Blackfin BF5xx</th>
<th>Blackfin BF70x</th>
<th>SHARC 21489</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15.98</td>
<td>16.04</td>
<td>16.36</td>
<td>8.68</td>
<td>18.16</td>
<td>13.04</td>
<td>5.19</td>
</tr>
</tbody>
</table>
Faster Biquads

SHARC has 2-way SIMD and can process 2 channels in parallel. NEON has 4-way SIMD and can process 4 channels in parallel (but we don’t have this function).

For NEON, we have a “Biquad Cascade Delay” function which implements a cascade by mono Biquad filters with a delay between stages. This allows NEON parallelization.

<table>
<thead>
<tr>
<th></th>
<th>Cortex-M4</th>
<th>Cortex-A8</th>
<th>Cortex-A9</th>
<th>Cortex-A15</th>
<th>SHARC 21489</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles per sample per stage</td>
<td>15.98</td>
<td>14.36</td>
<td>9.49</td>
<td>6.01</td>
<td>2.64</td>
</tr>
</tbody>
</table>
## FFT Benchmarks

<table>
<thead>
<tr>
<th>Length</th>
<th>Cortex-M4</th>
<th>Cortex-A8</th>
<th>Cortex-A9</th>
<th>Cortex-A15</th>
<th>Blackfin BF5xx</th>
<th>Blackfin BF70x</th>
<th>SHARC 21489</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>3709</td>
<td>3773</td>
<td>3358</td>
<td>2264</td>
<td>2200</td>
<td>1526</td>
<td>783</td>
</tr>
<tr>
<td>128</td>
<td>9811</td>
<td>6384</td>
<td>5682</td>
<td>3830</td>
<td>5249</td>
<td>3431</td>
<td>1334</td>
</tr>
<tr>
<td>256</td>
<td>21575</td>
<td>11114</td>
<td>9891</td>
<td>6668</td>
<td>11744</td>
<td>7611</td>
<td>2542</td>
</tr>
<tr>
<td>512</td>
<td>37813</td>
<td>21852</td>
<td>19448</td>
<td>13111</td>
<td>27385</td>
<td>17084</td>
<td>5189</td>
</tr>
<tr>
<td>1024</td>
<td>96630</td>
<td>50738</td>
<td>45157</td>
<td>30443</td>
<td>60216</td>
<td>37568</td>
<td>10972</td>
</tr>
</tbody>
</table>

Complex transform
No bit reversal
FFT cycle count is proportional to $K \times N \times \log_2(N)$

K factor shown below
Smaller is better

<table>
<thead>
<tr>
<th>Cortex-M4</th>
<th>Cortex-A8</th>
<th>Cortex-A9</th>
<th>Cortex-A15</th>
<th>Blackfin BF5xx</th>
<th>Blackfin BF70x</th>
<th>SHARC 21489</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.44</td>
<td>4.95</td>
<td>4.41</td>
<td>2.97</td>
<td>5.88</td>
<td>3.67</td>
<td>1.07</td>
</tr>
</tbody>
</table>
Introducing the Cortex-M7

Announced by ARM on 9/24/14
Main new feature is improved DSP performance
Achieved through
  Superscalar architecture
  Faster MAC
  Better memory bandwidth
  Clock speed increase to 400 MHz
Can only share preliminary information

Chart shows per cycle speed improvements for the M7 vs the M4
# Normalized Per Cycle Benchmarks

<table>
<thead>
<tr>
<th></th>
<th>Cortex-M4</th>
<th>Cortex-M7</th>
<th>Cortex-A8</th>
<th>Cortex-A9</th>
<th>Cortex-A15</th>
<th>Blackfin 5xx</th>
<th>Blackfin 70x</th>
<th>SHARC 21489</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIR</td>
<td>0.21</td>
<td>0.33</td>
<td>0.26</td>
<td>0.23</td>
<td>0.92</td>
<td>0.26</td>
<td>0.49</td>
<td>1.00</td>
</tr>
<tr>
<td>Biquad</td>
<td>0.16</td>
<td>0.28</td>
<td>0.18</td>
<td>0.28</td>
<td>0.44</td>
<td>0.15</td>
<td>0.20</td>
<td>1.00</td>
</tr>
<tr>
<td>FFT</td>
<td>0.11</td>
<td>0.17</td>
<td>0.22</td>
<td>0.24</td>
<td>0.36</td>
<td>0.18</td>
<td>0.29</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Performance normalized relative to SHARC
Higher numbers are better
With Processor Speed Differences

<table>
<thead>
<tr>
<th></th>
<th>Cortex-M4</th>
<th>Cortex-M7</th>
<th>Cortex-A8</th>
<th>Cortex-A9</th>
<th>Cortex-A15</th>
<th>Blackfin 5xx</th>
<th>Blackfin 70x</th>
<th>SHARC 21489</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIR</td>
<td>0.09</td>
<td>0.29</td>
<td>0.59</td>
<td>0.51</td>
<td>3.05</td>
<td>0.40</td>
<td>0.44</td>
<td>1.00</td>
</tr>
<tr>
<td>Biquad</td>
<td>0.07</td>
<td>0.25</td>
<td>0.41</td>
<td>0.62</td>
<td>1.46</td>
<td>0.23</td>
<td>0.18</td>
<td>1.00</td>
</tr>
<tr>
<td>FFT</td>
<td>0.05</td>
<td>0.15</td>
<td>0.48</td>
<td>0.54</td>
<td>1.20</td>
<td>0.28</td>
<td>0.26</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Takes into account maximum clock speeds

Cortex-M4: 204 MHz
Cortex-M7: 400 MHz
Cortex-A8: 1 GHz
Cortex-A9: 1 GHz
Cortex-A15: 1.5 GHz
Blackfin 53x: 700 MHz
Blackfin BF70x: 400 MHz
SHARC: 450 MHz

Bigger is better
Other Considerations

Complicating Factors
Non-deterministic / data dependent behavior
Long pipelines & processor stalls
Multiple threads
External memory and caches
Operating systems
Fixed vs floating-point

My Rules of Thumb
SHARC bare metal (no OS) - up to 95%
Cortex-M4 bare metal (no OS) – up to 90%
Blackfin bare metal (no OS) – up to 85%
ARM Cortex-A8/9/15 with cache and Linux – up to 65%
Benchmarking Real World Systems with Audio Weaver
Audio Weaver - Proprietary Design Tools

Complete audio processing solution
Large library of optimized modules
Graphical editor
Real-time tuning
Regression testing
Multirate processing
MIPs and memory profiling
Advanced features using MATLAB

Supports
ARM Cortex-M4 / M7
ARM Cortex-A8 / A9 / A15
ADI SHARC and Blackfin
TI C67x
Example: Loudspeaker Processing
Cortex-M7 results not available but we estimate M7 CPU load as 40 to 42 MHz.

### Loudspeaker Profile

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Module Type</th>
<th>M4 MIPS</th>
<th>SHARC MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS_toFloat</td>
<td>Fract32ToFloat</td>
<td>0.84</td>
<td>0.28</td>
</tr>
<tr>
<td>ToneBass1</td>
<td>SecondOrderFilterSmoothed</td>
<td>2.16</td>
<td>0.63</td>
</tr>
<tr>
<td>ToneTreble1</td>
<td>SecondOrderFilterSmoothed</td>
<td>2.13</td>
<td>0.63</td>
</tr>
<tr>
<td>Volume1</td>
<td>VolumeControl</td>
<td>2.14</td>
<td>0.73</td>
</tr>
<tr>
<td>XoverNway1</td>
<td>LRXoverN2Order2</td>
<td>11.6</td>
<td>3</td>
</tr>
<tr>
<td>OneBass</td>
<td>Adder</td>
<td>1.51</td>
<td>0.78</td>
</tr>
<tr>
<td>BassScale1</td>
<td>Scaler</td>
<td>0.44</td>
<td>0.25</td>
</tr>
<tr>
<td>BassEQ</td>
<td>SecondOrderFilterSmoothedCascade</td>
<td>4.11</td>
<td>2.48</td>
</tr>
<tr>
<td>BassLimit</td>
<td>Subsystem</td>
<td>10.14</td>
<td>7.18</td>
</tr>
<tr>
<td>BassMulti</td>
<td>Interleave</td>
<td>0.72</td>
<td>0.57</td>
</tr>
<tr>
<td>TweeterEQ</td>
<td>SecondOrderFilterSmoothedCascade</td>
<td>8.28</td>
<td>2.57</td>
</tr>
<tr>
<td>TweetLimit</td>
<td>Subsystem</td>
<td>11.96</td>
<td>8.16</td>
</tr>
<tr>
<td>BassAdder</td>
<td>Adder</td>
<td>0.8</td>
<td>0.39</td>
</tr>
<tr>
<td>MultiplexorFade1</td>
<td>MultiplexorFade</td>
<td>0.65</td>
<td>0.46</td>
</tr>
<tr>
<td>MuteSmoothed1</td>
<td>MuteSmoothed</td>
<td>1.49</td>
<td>0.36</td>
</tr>
<tr>
<td>SYS_toFract</td>
<td>FloatToFract32</td>
<td>2.17</td>
<td>0.37</td>
</tr>
<tr>
<td>Meter1</td>
<td>Meter</td>
<td>2.03</td>
<td>1.02</td>
</tr>
</tbody>
</table>

**Totals**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M4 Total</td>
<td>63.16</td>
</tr>
<tr>
<td>SHARC Total</td>
<td>29.86</td>
</tr>
</tbody>
</table>
Example High-End Automotive System

12 Channel High End System
Stereo entertainment content
8 announcement channels
Spectrum analyzer
Graphic equalizer
Speed dependent equalizer
Perceptual volume control
2 to 7.1 channel upmix

Announcement channel duckers
> 100 Biquads for speaker equalization
Separate compressor/limiter per channel
Time delays
Test signal generation for factory tests

443 individual audio modules from 57 different module classes
Automotive Profile

SHARC (21489. 400 MHz)
\[ 265 \text{ MHz} = 66.3\% \]

Cortex-A9 (TI OMAP 4430. 1 GHz)
\[ 914 \text{ MHz} = 92\% \]

Cortex-A15 (TI OMAP 5432. 1.5 GHz)
\[ 649 \text{ MIPs} = 43\% \]
Audio Weaver Business Model

**Pricing**
Free for prototyping, evaluation, and benchmarking
Utilize one of our supported evaluation boards
Pay to obtain processor specific libraries ($1k/per processor family)
Pay unit royalties when you ship products

**Eval Boards**
EZ-KITs from ADI (SHARC & BF)
SHARC audio hardware from Danville Signal
STM32F407 Discovery board (M4)
Multiple Linux-based eval boards for Cortex-A (Panda, BeagleBone, etc)
Increasing the Pace of Innovation

By creating an ecosystem for audio product development

Chris Anderson – ARM TechCon Keynote on 10/3/14

From difficult to easy
From expensive to cheap
From closed to open

Audio DSP
872

Audio Engineer
30,915

Embedded Software
97,975

59
11
129
Conclusion

There are many more choices now for audio processors.
Benchmarks are useful but only take you so far.
To minimize risk you need to prototype your processing and run on the actual device.
Thank You!

pbeckmann@dsppconcepts.com