Status update of RISC-V P extension task group

Chuan-Hua Chang
Senior Director
Andes Technology Corp.

2019 RISC-V Workshop Zurich
RISC-V DSP (P) Extension TG

- P extension task group charter
  - Define and ratify Packed-SIMD DSP extension instructions operating on XLEN-bit integer registers for embedded RISC-V processors.
  - Define compiler intrinsic functions that can be directly used in high-level programming languages.

- Chair: Chuan-Hua Chang, Andes Technology
- Co-chair: Eric Flamand, Greenwaves Technology
RISC-V DSP (P) Extension Proposal

- DSP instruction set proposal based on AndeStar™ V3 DSP ISA.
  - Use RV32 and RV64 XLEN-bit GPRs.
  - Support saturation and rounding.
  - Support fixed-point and integer data types.
  - SIMD-instructions with 8b, 16b, 32b element size.
  - Partial/Non-SIMD DSP instructions operating on 8-bit, 16-bit, 32-bit and 64-bit data types.
  - 64-bit signed/unsigned addition & subtraction (RV32)
  - 64-bit addition with 16b/32b multiplications
    - E.g., $64 = 64 + 16 \times 16 + 16 \times 16$
    - E.g., $64 = 64 + 32 \times 32$
GPR vs Separate Register

- GPR-based SIMD is a more efficient, low power DSP solution for embedded systems running applications in various domains such as audio/speech decoding and processing, IoT sensor data processing, wearable fitness devices, etc.

- It addresses the need for high performance generic code processing, as well as digital signal processing.
16-Bit SIMD Instructions

- Min
- Max
- Clip
- ABS
- Signed
- Unsigned

OP1  OP2

-  +  ×

<<  >>
8-Bit SIMD Instructions

- Min
- Max
- Unpack
- ABS
- Compare Signed
- Unsigned
Dual 16x16 & 32-Bit Add/Sub (RV32)

32b

Q31 Sat

±

16b 16b

±

32b 32b

16b 16b

6x Baseline operation

Taking RISC-V® Mainstream
Dual 16x16 & 64-Bit Add/Sub (RV32)

2.6x Q15 dot prod
Quad 16x16 & 64-Bit Add/Sub (RV64)

3.6x Q15 dot prod
Quad 8x8 & 32-Bit Add (RV32)

3.4x Q7 dot prod
Dual Quad 8x8 & 32-Bit Add (RV64)

7.7x Q7 dot prod
64-bit Data Type

- Use pairs of GPRs on RV32.
- Use a GPR on RV64.
- Needed for compiler to generate DSP instructions automatically.
- The 64-bit operand type is an interface specification. An implementation can still implement 2R1W register file with multi-cycle reads/writes to support the 64-bit type on RV32.
# DSP Library Speedup

## RV32

<table>
<thead>
<tr>
<th>Speedup</th>
<th>Basic</th>
<th>Complex</th>
<th>Controller</th>
<th>Filtering</th>
<th>Matrix</th>
<th>Statistics</th>
<th>Transform</th>
<th>Utils</th>
<th>Average /Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>P / Base (RV32)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>2.4</td>
<td>1.62</td>
<td>1.84</td>
<td>2.26</td>
<td>1.62</td>
<td>2.44</td>
<td>1.29</td>
<td>1.08</td>
<td><strong>1.82</strong></td>
</tr>
<tr>
<td>MAX</td>
<td>5.16</td>
<td>4.09</td>
<td>2.13</td>
<td>4.11</td>
<td>2.75</td>
<td>4.39</td>
<td>1.78</td>
<td>1.43</td>
<td>5.16</td>
</tr>
</tbody>
</table>

## RV64

<table>
<thead>
<tr>
<th>Speedup</th>
<th>Basic</th>
<th>Complex</th>
<th>Controller</th>
<th>Filtering</th>
<th>Matrix</th>
<th>Statistics</th>
<th>Transform</th>
<th>Utils</th>
<th>Average /Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>P / Base (RV64)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>4.73</td>
<td>1.92</td>
<td>1.31</td>
<td>2.41</td>
<td>3.04</td>
<td>4.14</td>
<td>1.28</td>
<td>1.19</td>
<td><strong>2.5</strong></td>
</tr>
<tr>
<td>MAX</td>
<td>10.81</td>
<td>4.14</td>
<td>1.59</td>
<td>5.04</td>
<td>6.83</td>
<td>8.51</td>
<td>1.67</td>
<td>2.72</td>
<td>10.81</td>
</tr>
</tbody>
</table>
## Helix MP3 Decoder

**Compiler only, no hand optimization**

<table>
<thead>
<tr>
<th>GCC Compiler</th>
<th>Decode (MCPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile with RV32IMC ISA</td>
<td>20.78</td>
</tr>
<tr>
<td>Compile with RV32IMC + P ISA</td>
<td>10.65</td>
</tr>
<tr>
<td>Speedup</td>
<td>1.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GCC Compiler</th>
<th>Decode (MCPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile with RV64IMC ISA</td>
<td>14.97</td>
</tr>
<tr>
<td>Compile with RV64IMC + P ISA</td>
<td>11.61</td>
</tr>
<tr>
<td>Speedup</td>
<td>1.29</td>
</tr>
</tbody>
</table>

* MCPS: Millions of Cycles Per Second

Taking RISC-V® Mainstream
### Voice Codec (P vs Baseline)

<table>
<thead>
<tr>
<th></th>
<th>RV32</th>
<th>RV32+P</th>
<th>Speedup</th>
<th>RV64</th>
<th>RV64+P</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMRWB En</td>
<td>244.65</td>
<td>66.45</td>
<td>3.68</td>
<td>249.72</td>
<td>70.23</td>
<td>3.56</td>
</tr>
<tr>
<td>AMRWB De</td>
<td>82.92</td>
<td>22.24</td>
<td>3.73</td>
<td>83.87</td>
<td>23.52</td>
<td>3.57</td>
</tr>
<tr>
<td>G729 En</td>
<td>56.39</td>
<td>15.75</td>
<td>3.58</td>
<td>59.42</td>
<td>19.25</td>
<td>3.09</td>
</tr>
<tr>
<td>G729 De</td>
<td>15.43</td>
<td>5.17</td>
<td>2.98</td>
<td>16.46</td>
<td>6.66</td>
<td>2.47</td>
</tr>
<tr>
<td>AMRNB En</td>
<td>25.96</td>
<td>22.72</td>
<td>1.14</td>
<td>26.47</td>
<td>22.72</td>
<td>1.17</td>
</tr>
<tr>
<td>AMRNB De</td>
<td>4.94</td>
<td>4.41</td>
<td>1.12</td>
<td>5.14</td>
<td>4.41</td>
<td>1.17</td>
</tr>
</tbody>
</table>

* MCPS: Millions of Cycles Per Second
# Neural Network (P vs Baseline)

<table>
<thead>
<tr>
<th></th>
<th>RV32</th>
<th>RV32+P</th>
<th>Speedup</th>
<th>RV64</th>
<th>RV64+P</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>54,002,847</td>
<td>9,765,828</td>
<td>5.53</td>
<td>66,726,795</td>
<td>6,077,522</td>
<td>10.98</td>
</tr>
<tr>
<td>GRU</td>
<td>31,545</td>
<td>16,981</td>
<td>1.86</td>
<td>32,892</td>
<td>11,144</td>
<td>2.95</td>
</tr>
</tbody>
</table>

* Cycles of Execution
P Task Group Progress

- Created P extension instruction proposal spreadsheet for TG members to review.

- Benchmarking on DSP library functions for the usefulness of these instructions.
  - About ~100 instructions are used in DSP library and audio/speech codec optimizations.

- Released detailed instruction operation specification.

- Released toolchain and simulator binaries for TG members to evaluate the use of these instructions.
Thank you