MIPS INSTRUCTION SET EXTENSIONS

Tarun Arora
CSE 240A
Goals for ISA extensions

- Provide Digital Media support for embedded processors
- Realize performance improvement
- Keeping it simple and fast
MIPS ISA philosophy

Basic Instructions evolve by super-sets and application-specific extensions permitted

ISA

MIPS V

MDMX

MIPS IV

PRIVATE (e.g., MulAC)

MIPS III

TBA

MIPS II

Applications

Media Processing

Memory: Constrained embedded

Partner Specific apps

Extended FP registers

Performance

Improved effective memory latency, FP computation

Data throughput & SIMD

04-bit
Music Extensions

- MDMX - To accelerate multimedia applications
- MIPS-3D ASE- Instructions added for improving the performance of 3D graphics applications.
- MIPS16e ASE – Provides advanced code density, high level of power efficiency and improves instruction cache hit rate.

- MIPS 32 24KE core family incorporates DSP ASE. This improves signal processing performance by 200%.
- MIPS 32 4KS core has an extension which does cryptography without the need of a coprocessor.

- E.g. ADDR, MULR -> accelerate matrix multiplication
- ADDR.PS fd, fs, ft
## Instructions in the MIPS-3D ASE

<table>
<thead>
<tr>
<th>Type</th>
<th>Mnemonic</th>
<th>Valid Formats</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic</td>
<td>ADDR</td>
<td>PS</td>
<td>Floating point reduction add</td>
</tr>
<tr>
<td></td>
<td>MULR</td>
<td>PS</td>
<td>Floating point reduction multiply</td>
</tr>
<tr>
<td></td>
<td>RECIP1</td>
<td>S, D, PS</td>
<td>Reciprocal first step with a reduced precision result</td>
</tr>
<tr>
<td></td>
<td>RECIP2</td>
<td>S, D, PS</td>
<td>Reciprocal second step (en route to the full precision result)</td>
</tr>
<tr>
<td></td>
<td>RSQRT1</td>
<td>S, D, PS</td>
<td>Reciprocal square-root with a reduced precision result</td>
</tr>
<tr>
<td></td>
<td>RSQRT2</td>
<td>S, D, PS</td>
<td>Reciprocal square-root second step (en route to the full precision result)</td>
</tr>
<tr>
<td>Format conversions</td>
<td>CVT.PS.PW</td>
<td>PW</td>
<td>Converts a pair of 32-bit fixed point integers to paired-single FP format</td>
</tr>
<tr>
<td></td>
<td>CVT.PW.PS</td>
<td>PS</td>
<td>Converts a paired-single FP format to a pair of 32-bit fixed point integers</td>
</tr>
<tr>
<td>Compare</td>
<td>CABS</td>
<td>S, D, PS</td>
<td>Magnitude compare of floating point numbers</td>
</tr>
<tr>
<td>Branch</td>
<td>BC1ANY2F</td>
<td></td>
<td>Branch if either one of the two specified (consecutive) condition codes is False</td>
</tr>
<tr>
<td></td>
<td>BC1ANY2T</td>
<td></td>
<td>Branch if either one of the two specified (consecutive) condition codes is True</td>
</tr>
<tr>
<td></td>
<td>BC1ANY4F</td>
<td></td>
<td>Branch if any one of the four specified (consecutive) condition codes is False</td>
</tr>
<tr>
<td></td>
<td>BC1ANY4T</td>
<td></td>
<td>Branch if any one of the four specified (consecutive) condition codes is True</td>
</tr>
</tbody>
</table>
References