Organization of Sony’s Emotion Engine ASIC
Why Emotion Engine?

- Competitors using generic parts
  - Performance benefits to be had for custom design
- Original PlayStation: no vector processing or floating point support
  - Geometry issues
- “Photorealism” at the core of design strategy
  - Sony PS9 ad – virtual reality
  - Requires massive floating-point power
- Focus on multimedia
  - DVD decoding, PSX DVR
Sony EE: Quick Facts

- Mfd. by Toshiba on 250nm CMOS process
- CPU core
- Two vector coprocessors
- MPEG-2 decoder
- Dual-channel memory controller
- Built-in graphics interface
- Linked together using I/O bus and point to point links
  - Components assisted by 10-channel DMA controller
Emotion Engine CPU Core

- Based on MIPS R5900 core
  - Confirms to MIPS III ISA, with parts of MIPS IV
  - Integer SIMD extensions: MMI (107 instructions)
- 32 128-bit registers, R0 – R31
  - Upper 64 bits only available to MMI instructions
- Six-stage, dual-issue pipeline
  - Branch prediction: 64-entry BTB with BHT in cache
  - Mispredict penalty: three cycles
- Split L1 cache: 16k instruction, 8k data
- 16k scratchpad RAM
Emotion Engine FPU

- Runs inline with CPU as a coprocessor (COP1)
- No floating-point SIMD on the FPU
- Consists of one FMAC and one FDIV
  - FMAC takes one cycle, FDIV takes seven
  - Instruction decode stage stalls pipeline to prevent structural hazards when more than one is queued
- Possible to overcome limitation using VU0 in coprocessor mode (COP2)
  - Adds one extra FMAC and one extra FDIV
EE Vector Units (VU0/1)

- 128-bit VLIW architecture (like Itanium)
  - Identical in both
- 32 128-bit general purpose registers
  - Full 128 bits are accessible to all ops, unlike CPU
  - All operations are SIMD (128-bit ops not possible)
- Single-issue, two operations per instruction
  - Upper operation: FMAC
  - Lower operation: Any operation
- Operate in parallel with the CPU
- Vertex operation (19 FMACs, 1 FDIV): 7 cycles
Vector unit 0 (VU0)

- Point-to-point link with CPU
  - Shares 16k SPRAM
- Split instruction and data cache, 4k each
- Can run in one of two modes
  - Coprocessor mode (COP2): Runs inline with the CPU as a second floating point unit
  - Parallel mode: Runs independently of the CPU; program stream assisted by DMA controller
- Four FMAC units and one FDIV unit
  - Extra units only available in parallel mode
Vector unit 1 (VU1)

- Point-to-point link with graphics interface
- Split instruction and data cache, 16k each
  - Data cache shared with GPU
- Runs only in parallel mode, assisted by DMA
- Elementary Functional Unit
  - $n$–stage pipeline
  - Adds extra FMAC and FDIV unit (total five FMAC, two FDIV)
  - Exponents, logarithms, etc.
- Can be reset with new microcode independently of the rest of the chip
Image processing unit (IPU)

- Architecturally similar to a DSP
- Decodes JPEG stills and MPEG-2 macroblocks
  - I-frames, P-frames, B-frames
  - FIFO input/output queues
  - Hardware deinterlace, IDCT, etc.
  - Support for MPEG-2 peculiarities, e.g. variable-length decoding
- Can optionally be used for texture compression
On-chip I/O

- On-chip I/O bus: 128 bits wide at half clock
  - Makes bus transfer from 64-bit CPU easier
  - Peak throughput @ 300MHz core clock: 2.4GB/s

- Memory controller
  - Dual-channel DDR design (actually RDRAM)
  - 2x 16-bit channels at 400MHz each = 3.2GB/s
    - Onboard buffering in memory controller

- Ten DMA channels to offload data from CPU

- Point-to-point links
  - CPU <-> VU0
  - GIF <-> VU1
<table>
<thead>
<tr>
<th>I</th>
<th>Q</th>
<th>R</th>
<th>A</th>
<th>D</th>
<th>W</th>
<th>M</th>
<th>T</th>
<th>T</th>
<th>X</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>Z</th>
<th>S</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Q</td>
<td>R</td>
<td>A</td>
<td>D</td>
<td>W</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>X</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>Z</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>I</td>
<td>Q</td>
<td>R</td>
<td>A</td>
<td>D</td>
<td>W</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>D</td>
<td>1</td>
<td>D</td>
<td>2</td>
<td>D</td>
<td>3</td>
<td>D</td>
</tr>
<tr>
<td>I</td>
<td>Q</td>
<td>R</td>
<td>A</td>
<td>D</td>
<td>W</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>D</td>
<td>1</td>
<td>D</td>
<td>2</td>
<td>D</td>
<td>3</td>
<td>D</td>
</tr>
<tr>
<td>I</td>
<td>Q</td>
<td>R</td>
<td>T</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>S</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td>2</td>
<td>...</td>
</tr>
<tr>
<td>I</td>
<td>Q</td>
<td>R</td>
<td>T</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>S</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td>2</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>D</td>
<td>1</td>
<td>D</td>
<td>2</td>
<td>D</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>D</td>
<td>1</td>
<td>D</td>
<td>2</td>
<td>D</td>
<td>3</td>
</tr>
</tbody>
</table>

Pipelines: Dual-issue integer pipeline, FPU pipeline, VLIW FMAC pipeline, VLIW FDIV/FSQRT pipeline, VLIW EFU pipeline, VLIW FRQSQT pipeline
Comparison

- **EE benchmark @ 300MHz (PS2)**
  - Peak 6.2 GFLOPs
  - 66 million triangle transformations per second

- **Contemporary high-end CPU: 500MHz PIII**
  - Peak 2 GFLOPs
  - Four million triangle transformations per second

- **Contemporary game console: Sega Dreamcast**
  - Peak 1.4 GFLOPs
  - Four million triangle transformations per second

- **Contemporary GPU: 3dfx Voodoo3 (R4)**
  - Peak 2.6 GFLOPs
  - 44 million triangle transformations per second
Why is EE–style design uncommon?

- “Schizophrenic” architecture
  - Some parts SIMD, some parts VLIW, etc.
- Very difficult to write an efficient compiler
  - GCC port does not emit code for vector units
  - VLIW issues mirrored by Itanium
- Design paradigms shifted
  - GPGPU – task–level parallelism on lots of cores
  - Multi–core CPUs
    - Easier to target identical architectures
    - PS3 CPU features eight parallel SPEs