RISC processors

RISC Processors
COMP375 Computer Architecture and Organization
blue slides ©Intel Gautam Doshi

RISC Traits
• Pipelined
• Simple instructions
• Few instructions
• No microcode
• Few addressing modes
• Load/Store architecture
• Sliding register stack
• Delayed branches
• Fast

Current RISC Systems
• **PowerPC** – The processor in the Apple Power Mac. Produced by IBM and Apple.
• **Sparc** – The processor in Sun workstations and servers. Produced by Sun Microsystems. First commercial RISC.
• **Itanium** – In new servers replacing the Intel Pentium. Produced by Intel.

Intel Itanium®
• Intel’s latest RISC system.
• The current processor is the Itanium 2.
• Intel seems to indicate that this is the replacement for the Pentium chip.

Support of Pentium Instructions
• The Itanium can execute both Itanium instructions and Pentium (IA-32) instructions
• There are jump to IA-32/Itanium instructions

Today’s Architecture Challenges
**Sequential Semantics**
• Program = **Sequence** of instructions
• Implied order of instruction execution
• Potential dependence from inst. to inst.
But ...
• High performance needs parallel execution
• Parallel execution needs independent insts.
• Independent insts must be (re)discovered

Sequentiality inherent in traditional archs

Intel Itanium® System Environment
IA-32 Instructions
• Segmentation
• Paging & Interruption Handling in the Intel Itanium Architecture
• Itanium Instructions
**Compiler to Processor Hints**

- Every memory load and store in the Itanium architecture has a 2-bit cache hint field.
- The compiler can provide a hint to indicate if a branch is likely to be taken.
- Templates define which execution units will be used and if dependencies exist.

---

**Predication**

**Traditional Arch**

- `if cond { then } else { }`

**Itanium® Architecture**

- `if cond { then } { then } { else }`

- **Control flow to Data flow**

- Predication removes/reduces branches.
Register Stacks

- Many RISC processors have a large number of registers, not all of which are visible at any one time.
- The mapping of register X to a hardware register changes when a function is called.

Before a Function Call

- Assume the assembly language programmer sees 32 registers.
- Before a function call, arguments and the return address are put in registers R24 to R31.

After a Function Call

- After a function call, the input arguments and the return address are available in registers R8 to R15.
- R16 to R23 are used for local variables.
- R24 to R31 contain arguments to next function

After another Function Call

- After another function call, the input arguments and the return address are again available in registers R8 to R15.
- Return values are also put in R8 to R15 upon function return.
RISC processors

After Function Return

• After the function return, the return values are available in registers R24 to R31.

R0 R7
R8 R15 R16 R23 R24 R31

Itanium Register Stack

• The Itanium uses a sliding register system somewhat similar to the generic description
• General registers 0 through 31 are termed the **static general registers**.
• General registers 32 through 127 are termed the **stacked general registers**.
• A function can specify how many of the stacked general registers the system is to shift.
• GP0 is always zero.

Itanium Registers

APPLICATION REGISTER SET

General Registers Floating-point Registers Predicts
\[\begin{array}{cccccc}
\text{R0} & \text{R7} & \text{R8} & \text{R15} & \text{R16} & \text{R23} \\
\hline
\end{array}\]

Branch Registers Application Registers

Itanium Floating Point

• The Itanium has 128 floating-point registers
• Each register holds an 82-bit floating point value.
• Values are rounded as they are stored as 32 bit floats or 64 bit doubles.

Register Stack Engine (RSE)

• Automatically saves/restores stack registers without software intervention
  ▪ Provides the illusion of infinite physical registers
  ▪ by mapping to a stack of physical registers in memory
  ▪ Overflow: Alloc needs more registers than available
  ▪ Underflow: Return needs to restore frame saved in memory
• RSE may be designed to utilize unused memory bandwidth to perform register spill and fill operations in the background

FP: High performance and ____ high precision

Floating-Point Architecture

• Fused Multiply Add Operation
  ▪ An efficient core computation unit
• Abundant Register resources
  ▪ 128 registers (32 static, 96 rotating)
• High Precision Data computations
  ▪ 82-bit unified internal format for all data types
• Software divide/square-root
  ▪ High throughput achieved via pipelining
Endian

- The Itanium can execute in Big Endian or Little Endian mode.
- Instruction fetches are always Little Endian

Itanium OS Support

- Redhat Linux servers will run on the Itanium. The desktop does not.
- Microsoft Windows Servers will run on the Itanium. Windows XP Professional will not.
- Sun Solaris runs on 64 bit Sparc processors, but not on the Intel Itanium.

PowerPC Registers

<table>
<thead>
<tr>
<th>General-purpose registers</th>
<th>Floating-point registers</th>
<th>Condition register</th>
<th>XER register</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPRI (32/64)</td>
<td>FRR (64)</td>
<td>CCR (32)</td>
<td>XER (32)</td>
</tr>
<tr>
<td>GPRI (32/64)</td>
<td>FRR (64)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPRI (32/64)</td>
<td>FRR (64)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PowerPC Branches

- Every jump instruction has two extra bits
- AA bit
  - 1 (use absolute address)
  - 0 (use relative address)
- LK bit
  - 0 (no link --- branch)
  - 1 (link --- turns branch into a procedure call)