Today’s Topic

• CISC & RISC Machines
  – Chapters 1.4 & 1.5 of Leland Beck’s “System Software” book.
Traditional (CISC) Machines

- Complex Instruction Set Computers (CISC)
  - Complicated instruction set
  - Different instruction formats and lengths
  - Many different addressing modes
  - e.g. VAX or PDP-11 from DEC
  - e.g. Intel x86 family

Pentium Pro Architecture (1/5)

- Memory
  - Physical level: byte addresses, word, doubleword
  - Logical level: segments and offsets
  - In some cases, a segment can also be divided into pages
  - The segment/offset address specified by the programmer is translated into a physical address by the x86 MMU (Memory Management Unit)
Pentium Pro Architecture (2/5)

- Registers
  - General-purpose registers:
    - EAX, EBX, ECX, EDX: data manipulation
    - ESI, EDI, EBP, ESP: address
  - Special-purpose registers:
    - EIP: next instruction
    - FLAGS: status word
    - CS: code segment register
  - SS: stack segment register
  - DS, ES, FS, and GS: data segments
  - Floating-point unit (FPU)
  - Registers reserved for system programs

Pentium Pro Architecture (3/5)

- Data Formats
  - Integers:
    - 8-, 16-, 32-bit binary numbers
    - Negative values: 2’s complement
    - FPU can also handle 64-bit signed integers
    - The least significant part of a numeric value is stored at the lowest-numbered address (little-endian)
    - Binary coded decimal (BCD)
      - unpacked: 0000____0000____0000____…...0000____
      - packed: |____|____|____|____|____|____|…..|____|____|
  - Floating-point data formats
    - *Single-precision*: 32 bits=24+7-bit exponent+sign bit
    - *Double-precision*: 64 bits=53+10-bit exponent+sign bit
    - *Extended-precision*: 80 bits=64+15-bit exponent+sign bit
Pentium Pro Architecture (4/5)

• Instruction Formats
  – Prefix (optional) containing flags that modify the operation of instruction
    • specify repetition count, segment register, etc.
  – Opcode (1 or 2 bytes)
  – Operands and addressing modes

• Addressing Modes
  – \( TA = (\text{base register}) + (\text{index register}) \times (\text{scale factor}) + \text{displacement} \)
  – Base register: any general-purpose registers
  – Index register: any general-purpose registers except ESP
  – Scale factor: 1, 2, 4, 8
  – Displacement: 8-, 16-, 32-bit value
  – Eight addressing modes

Pentium Pro Architecture (5/5)

• Instruction Set
  – 400 different machine instructions
    • R-to-R instructions, R-to-M instructions, M-to-M instructions
    • immediate values,
  – Special purpose instructions for high-level programming language
    • entering and leaving procedures,
    • checking subscript values against the bounds of an array

• Input and Output
  – Input is performed by instructions that transfer one byte, word, or doubleword from an I/O register EAX
  – Repetition prefixes allow these instructions to transfer an entire string in a single operation
RISC Machines

• RISC system
  – Instruction
    • Standard, fixed instruction format
    • Single-cycle execution of most instructions
    • Memory access is available only for load and store instruction
    • Other instructions are register-to-register operations
    • A small number of machine instructions, and instruction format
      – A large number of general-purpose registers
      – A small number of addressing modes

RISC Machines

• Three RISC machines
  – SPARC family
  – PowerPC family
  – Cray T3E
UltraSPARC (1/8)

- Sun Microsystems (1995)
- SPARC stands for scalable processor architecture
- SPARC, SuperSPARC, UltraSPARC
  - Memory
  - Registers
  - Data formats
  - Instruction Formats
  - Addressing Modes

UltraSPARC (2/8)

- Byte addresses
  - Two consecutive bytes form halfword
  - Four bytes form a word
  - Eight bytes form doubleword
- Alignment
  - Halfword are stored in memory beginning at byte address that are multiples of 2
  - Words begin at addresses that are multiples of 4
  - Doublewords at addresses that are multiples of 8
- Virtual address space
  - UltraSPARC programs can be written using $2^{64}$ bytes
  - Memory Management Unit
UltraSPARC (3/8)

• Registers
  – ~100 general-purpose registers
  – Any procedure can access only 32 registers (r0~r31)
    • First 8 registers (r0~r8) are global, i.e. they can be accessed by all procedures on the system (r0 is zero)
    • Other 24 registers can be visualized as a window through which part of the register file can be seen
  – Program counter (PC)
    • The address of the next instruction to be executed
  – Condition code registers
  – Other control registers

UltraSPARC (4/8)

• Data Formats
  – Integers are 8-, 16-, 32-, 64-bit binary numbers
  – 2’s complement is used for negative values
  – Support both big-endian and little-endian byte orderings
    • (big-endian means the most significant part of a numeric value is stored at the lowest-numbered address)
  – Three different floating-point data formats
    • Single-precision, 32 bits long (23 + 8 + 1)
    • Double-precision, 64 bits long (52 + 11 + 1)
    • Quad-precision, 78 bits long (63 + 16 + 1)
UltraSPARC (5/8)

- Three Instruction Formats
  - 32 bits long
  - The first 2 bits identify which format is being used
  - Format 1: call instruction
  - Format 2: branch instructions
  - Format 3: remaining instructions

UltraSPARC (6/8)

- Addressing Modes
  - Immediate mode
  - Register direct mode
  - Memory addressing

<table>
<thead>
<tr>
<th>Mode</th>
<th>Target address calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC-relative*</td>
<td>(TA = (PC) + \text{displacement} {30 \text{ bits, signed}})</td>
</tr>
<tr>
<td>Register indirect</td>
<td>(TA = (\text{register}) + \text{displacement} {13 \text{ bits, signed}})</td>
</tr>
<tr>
<td>with displacement</td>
<td></td>
</tr>
<tr>
<td>Register indirect indexed</td>
<td>(TA = (\text{register}-1) + (\text{register}-2))</td>
</tr>
</tbody>
</table>

*PC-relative is used only for branch instructions*
UltraSPARC (7/8)

- Instruction Set
  - <100 instructions
  - Pipelined execution
    - While one instruction is being executed, the next one is fetched from memory and decoded
  - Delayed branches
    - The instruction immediately following the branch instruction is actually executed before the branch is taken
  - Special-purpose instructions
    - High-bandwidth block load and store operations
    - Special “atomic” instructions to support multi-processor system

UltraSPARC (8/8)

- Input and Output
  - A range of memory locations is logically replaced by device registers
  - Each I/O device has a unique address, or set of addresses
  - No special I/O instructions are needed
PowerPC Architecture (1/8)

- POWER stands for Performance Optimization with Enhanced RISC
- History
  - IBM (1990) introduced POWER in 1990 with RS/6000
  - IBM, Apple, and Motorola formed an alliance to develop PowerPC in 1991
  - The first products were delivered near the end of 1993
  - Recent implementations include PowerPC 601, 603, 604

PowerPC Architecture (2/8)

- Memory
  - Halfword, word, doubleword, quadword
  - May instructions may execute more efficiently if operands are aligned at a starting address that is a multiple of their length
  - Virtual space $2^{64}$ bytes
  - Fixed-length segments, 256 MB
  - Fixed-length pages, 4KB
  - MMU: virtual address $\rightarrow$ physical address
PowerPC Architecture (3/8)

• Registers
  – 32 general-purpose registers, GPR0~GPR31
  – FPU
  – Condition code register reflects the result of certain operations, and can be used as a mechanism for testing and branching
  – Link Register (LR) and Count Register (CR) are used by some branch instructions
  – Machine Status Register (MSR)

PowerPC Architecture (4/8)

• Data Formats
  – Integers are 8-, 16-, 32-, 64-bit binary numbers
  – 2’s complement is used for negative values
  – Support both big-endian (default) and little-endian byte orderings
  – Three different floating-point data formats
    • single-precision, 32 bits long (23 + 8 + 1)
    • double-precision, 64 bits long (52 + 11 + 1)
  – Characters are stored using 8-bit ASCII codes
PowerPC Architecture (5/8)

- Seven Instruction Formats
  - 32 bits long
  - The first 6 bits identify specify the opcode
  - Some instruction have an additional extended opcode
  - The complexity is greater than SPARC
  - Fixed-length makes decoding faster and simple than VAX and x86

PowerPC Architecture (6/8)

- Addressing Modes
  - Immediate mode, register direct mode
  - Memory addressing

<table>
<thead>
<tr>
<th>Mode</th>
<th>Target address calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register indirect</td>
<td>( TA = \text{register} )</td>
</tr>
<tr>
<td>Register indirect with indexed</td>
<td>( TA = \text{register-1} + \text{register-2} )</td>
</tr>
<tr>
<td>Register indirect with immediate indexed</td>
<td>( TA = \text{register} + \text{displacement} ) {16 bits, signed}</td>
</tr>
</tbody>
</table>

- Branch instruction

<table>
<thead>
<tr>
<th>Mode</th>
<th>Target address calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute</td>
<td>( TA = \text{actual address} )</td>
</tr>
<tr>
<td>Relative</td>
<td>( TA = \text{current instruction address} + \text{displacement} ) {25 bits, signed}</td>
</tr>
<tr>
<td>Link Register</td>
<td>( TA = \text{(LR)} )</td>
</tr>
<tr>
<td>Count Register</td>
<td>( TA = \text{(CR)} )</td>
</tr>
</tbody>
</table>
PowerPC Architecture (7/8)

• Instruction Set
  – 200 machine instructions
    • More complex than most RISC machines
    • e.g. floating-point “multiply and add” instructions that take three input operands
    • e.g. load and store instructions may automatically update the index register to contain the just-computed target address
  – Pipelined execution
    • More sophisticated than SPARC
  – Branch prediction

PowerPC Architecture (8/8)

• Input and Output
  – Two different modes
    • Direct-store segment: map virtual address space to an external address space
    • Normal virtual memory access