Instruction Design
(Architecture)
ISA
to the detailed specifics of a machine. Any use of the word architecture will be generic and cover all three aspects.

Key Structural Concepts

Typical instruction format:

```
| opcode | operands |
```

We can classify each instruction in a three-dimensional space as shown below:
Chapter 3. Instruction Set Design
**Typical Instruction Operations**

<table>
<thead>
<tr>
<th>Category</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Movement</td>
<td>Load (from memory)</td>
</tr>
<tr>
<td></td>
<td>Store (to memory)</td>
</tr>
<tr>
<td></td>
<td>memory-to-memory move</td>
</tr>
<tr>
<td></td>
<td>register-to-register move</td>
</tr>
<tr>
<td></td>
<td>input (from I/O device)</td>
</tr>
<tr>
<td></td>
<td>output (to I/O device)</td>
</tr>
<tr>
<td></td>
<td>push, pop (to/from stack)</td>
</tr>
<tr>
<td>Arithmetic</td>
<td>integer (binary + decimal) or FP</td>
</tr>
<tr>
<td></td>
<td>Add, Subtract, Multiply, Divide</td>
</tr>
<tr>
<td>Logical</td>
<td>not, and, or, set, clear</td>
</tr>
<tr>
<td>Shift</td>
<td>shift left/right, rotate left/right</td>
</tr>
<tr>
<td>Branch</td>
<td>unconditional, conditional</td>
</tr>
<tr>
<td>Subroutine Linkage</td>
<td>call, return</td>
</tr>
<tr>
<td>Interrupt</td>
<td>trap, return</td>
</tr>
<tr>
<td>Synchronization</td>
<td>test &amp; set (atomic r-m-w)</td>
</tr>
<tr>
<td>String</td>
<td>search, translate</td>
</tr>
</tbody>
</table>
MIPS: Instruction Classes

- Load and Store
- Arithmetic/Logic/Shift (including register-register moves)
- Control Transfers (branches, subroutines)
- R/W Control Registers (privileged)
- FP operate
- Coprocessor operate

Separated Load/Store and Reg/Reg Architecture Critical Ingredient of RISC architectures

All instructions exactly 32-bits long

Exactly two formats:
  three register operands OR one register & one memory operand
Classification by the number of Operands

- 0 - Operand Machines - Stack Machines (HP3000)

- 1 - Operand Machines - Accumulator machines (one operand is always implicitly assumed to be in the accumulator ACC)

- 2 - Operand Machines: \( R1 \leftarrow R1 \times R2 \) (One of the source registers is also a destination register - Target)

- 3 - Operand Machines: \( R1 \leftarrow R2 \times R3 \) (Destination / Target Register is explicitly specified - easier on the Compiler)
Classification by the Location of the Operands

- R - M: Register to Memory Instructions
  most of the machines: IBM 360/370 Architecture

- M - M: Memory to Memory Instructions (Architectures)
  VAX - 11 Instructions etc. (some S/360)

- R - R: register to Register Machines (Load/Store)
  (RISC Machines)
Arithmetic and Logic Instruction:

IF    ID

OAG:  Difference between R-R instructions
      M-R Instructions
      M-M Instructions

R - R:

\[
\begin{array}{cccc}
\text{OP} & \text{RT} & \text{RS1} & \text{RS2} \\
\text{OP} & \text{RT/RS1} & \text{RS2} \\
\end{array}
\]

R - M:

\[
\begin{array}{cccc}
\text{OP} & \text{RT} & \text{RS1} & \text{Memory Address} \\
\end{array}
\]

M - M:

\[
\begin{array}{cccc}
\text{OP} & \text{Memory Add1} & \text{Memory Add2} & \text{Memory Add3} \\
\end{array}
\]
INSTRUCTION SETS: ADDRESSING MODES AND FORMATS

Memory Reference Instructions

<table>
<thead>
<tr>
<th>Op Code</th>
<th>D/I</th>
<th>Z/C</th>
<th>Displacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Input/Output Instructions

<table>
<thead>
<tr>
<th>1 1 0</th>
<th>Device</th>
<th>Op Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>11</td>
</tr>
</tbody>
</table>

Register Reference Instructions

Group 1 Microinstructions

<table>
<thead>
<tr>
<th>1 1 1 0</th>
<th>CLA</th>
<th>CLL</th>
<th>CMA</th>
<th>CML</th>
<th>RAR</th>
<th>RAL</th>
<th>BSW</th>
<th>IAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>

Group 2 Microinstructions

<table>
<thead>
<tr>
<th>1 1 1 1</th>
<th>CLA</th>
<th>SMA</th>
<th>SZA</th>
<th>SNL</th>
<th>RSS</th>
<th>OSR</th>
<th>HLT</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>

Group 3 Microinstructions

<table>
<thead>
<tr>
<th>1 1 1 1</th>
<th>CLA</th>
<th>MQA</th>
<th>0</th>
<th>MQL</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>

Mnemonics

CLA = CClear Accumulator
CLL = CClear Link
CMA = CoMplement Accumulator
CML = CoMplement Link
RAR = Rotate Accumulator Right
RAL = Rotate Accumulator Left
BSW = Byte Swap
IAC = Increment Accumulator
SMA = Skip on Minus Accumulator
SZA = Skip on Zero Accumulator
SNL = Skip on Nonzero Link
RSS = Reverse Skip Sense
OSR = Or with Switch Register
HLT = HaLT
MQA = Multiplier Quotient into Accumulator
MQL = Multiplier Quotient Load

FIGURE 9-2. PDP-8 instruction formats.
FIGURE 9-4. Instruction formats used on the PDP-11. The numbers indicate the field lengths.

Source and dest each contain a 3-bit addressing mode field and a 3-bit register number;
FP is 1 of the floating point registers 0, 1, 2, or 3;
R is 1 of the general registers;
CC is the condition code field.
FIGURE 9-5. IBM S/370 instruction formats.