ISA - Instruction Set Architecture (IV)

Addressing Modes
- Another factor impacting the ISA is addressing modes.
- An instruction usually has opcode and operands.
- The operation field of an instruction specifies the operation to be performed, and the address field specifies the data to be executed by the operation.
- Operands sometimes refer to GPRs, main memory addresses, or values. Operands are all in binary. To specify how to parse the operands, it’s necessary to have an addressing mode field to indicate the addressing mode.
- The address field are relatively small. To be able to reference a large range of locations in memory, a variety of addressing techniques has been employed.

Immediate

- **Operand value is present in the instruction**
- **Used to define constants or set initial values of variables**
- **The number will be stored in 2's complement**
- **Advantage: no memory reference**
- **Disadvantage: the size of the number is restricted to the size of the address field.** (In most instruction sets, the size of operand field is small compared with the word length.)

Direct
• The address field contains the actual address of the operand
• So, the value of operand is not restricted by the address field. Common in earlier generations of computers, but not common on contemporary architectures
• Requires only one memory reference and no special calculation
• Advantage: simple
• Disadvantage: limited address space (address field is always less than a word)

Indirect

- The address field refers to the address of a word in memory, and that word contains the actual address of the operand
- The word can provide a full-length address solving the disadvantage of direct addressing
- Advantage: large address space (word length N, address space $2^N$ is available)
- Disadvantage: multiple memory references (one to get the address, one to get the value)

- Example:
  - Given the following memory values, assume the LOAD instruction loads the value at a specific address to the accumulator based on the addressing mode. What values do the following instructions load into the accumulator?

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>102</td>
</tr>
<tr>
<td>101</td>
<td>103</td>
</tr>
<tr>
<td>102</td>
<td>104</td>
</tr>
<tr>
<td>103</td>
<td>105</td>
</tr>
</tbody>
</table>

a. LOAD IMMEDIATE 100 (100)
b. LOAD DIRECT 100 (102)
c. LOAD INDIRECT 100 (104)
Register Direct

- Similar to direct addressing. The only difference is that the address field refers to a register that contains the operand.
- To clarify, if the content of R is 5, then register R5 is the intended address, and the operand value is contained in R5.
- Typically, R is 3 to 5 bits, so that a total of 8 to 32 general-purpose registers can be referenced.
- **Advantages** compared with Direct:
  - small address field in the instruction
  - no memory reference (memory reference are time-consuming)
- **Disadvantages** compared with Direct: limited storage space (limited number of registers compared with main memory locations)

Register Indirect

- Analogous to indirect addressing. The only difference is that the address field refers to a register. The value stored in the register is the actual address of the operand in main memory.
- **Advantage** compared with Indirect:
  - one less memory reference, and smaller address field
- **Disadvantage** compared with Indirect:
  - limited storage space