[Von Neumann] Computers

- CPU
- memory (disk drives, DRAM, SRAM, CD)
- input (mouse, keyboard)
- output (display, printer)
- network
- software

We need abstraction ...
Levels of abstraction

Software:
  Application
  Operating system
  Firmware

Instruction set architecture:
  Data type and structures: encodings and machine representation
  Instruction set
  Instruction formats
  Addressing modes and accessing data and instructions

Hardware:
  Instruction set processing
  I/O System
  Digital design
  Circuit design
  Layout
What is “Computer Architecture”

Computer Architecture =

Instruction Set Architecture +

Machine Organization
Instruction Set Architecture

• Organization of Programmable Storage
• Data type and Structures: encodings and machine representation
• Instruction set
• Instruction Formats
• Addressing Modes and Accessing Data and Instructions
• Exception Handling

Computer Organization

• Capabilities and Performance of the Basic Functional Units
• The Way These Units are Interconnected
• Information Flow between components
• Information Flow Control
What is “Computer Architecture”?

- Application
- Operating System
  - Compiler
  - Firmware
- I/O system
- Datapath & Control
- Digital Design
- Circuit Design
- Layout

Instruction Set Architecture
MIPS machine
Example: adding two variables

Software level

C:
  • $A = B + C$

Assembler
  • $B \rightarrow $s1, $C \rightarrow $s2
  • add $t0, $s1, $s2
  • $t0 \rightarrow A$

Machine instruction:

<table>
<thead>
<tr>
<th>op</th>
<th>rs</th>
<th>rt</th>
<th>rd</th>
<th>…</th>
<th>funct</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>17</td>
<td>18</td>
<td>8</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>000000</td>
<td>10001</td>
<td>10010</td>
<td>01000</td>
<td>00000</td>
<td>100000</td>
</tr>
</tbody>
</table>
Example: Datapath and control
Example: from instructions to gates

Register file and ALU

Instruction

Read register 1
Read register 2
Write register
Write data

Read data 1
Read data 2

ALU operation

Zero

RegWrite

ALU result

ALU
Example: from instructions to gates

Inside register file
Example: from instructions to gates

Arithmetic Logic Unit (ALU)
Example: from instructions to gates

ALU: Carry Out logic

\[
\text{CarryOut} = b \cdot \text{CarryIn} + a \cdot \text{CarryIn} + a \cdot b + a \cdot b \cdot \text{CarryIn}
\]

or

\[
\text{CarryOut} = b \cdot \text{CarryIn} + a \cdot \text{CarryIn} + a \cdot b
\]
Example: from instructions to gates

Logic gates

1. **AND gate (c = a . b)**
   - ![AND gate diagram]
   - Table:
     | a | b | c = a . b |
     |---|---|----------|
     | 0 | 0 |    0     |
     | 0 | 1 |    0     |
     | 1 | 0 |    0     |
     | 1 | 1 |    1     |

2. **OR gate (c = a + b)**
   - ![OR gate diagram]
   - Table:
     | a | b | c = a + b |
     |---|---|----------|
     | 0 | 0 |    0     |
     | 0 | 1 |    1     |
     | 1 | 0 |    1     |
     | 1 | 1 |    1     |

3. **Inverter (c = a)**
   - ![Inverter diagram]
   - Table:
     | a | c = a |
     |---|-------|
     | 0 |    1  |
     | 1 |    0  |

4. **Multiplexor**
   - ![Multiplexor diagram]
   - Table:
     | d | c     |
     |---|-------|
     | 0 | a     |
     | 1 | b     |