Computer Systems:
Microprocessors, Memory and Data

http://www-compsci.swan.ac.uk/~csetzer/lectures/compsys/index.html

Dr. Anton Setzer

http://www-compsci.swan.ac.uk/~csetzer/index.html

November/December 2000
Address:
Dr. Anton Setzer
Dept. of Computer Science
University of Wales Swansea
Singleton Park
SA2 8PP
UK
Room  Room 211, Engineering Building
Tel. (01792) 205678 (switchboard) ext. 4518
Fax. (01792) 295651
Email a.g.setzer@swansea.ac.uk
Why this Course?

- **Knowledge** about how a computer really operates.
  A computer *scientist* show know what a computer is – what is behind this black box?
- **Technical Knowledge**
  - Jargon
  - Allows to follow
    * Current discussions.
    * Future developments.
- **Optimization of Programs.**
  - In assembly languages.
  - In higher level languages.
- **Hardware Verification.**
Recommended Literature

• W. Stallings: *Computer organization and architecture.*

• D. Patterson and J. Hennessy: *Computer Organization and Design.*
  Good. Detailed calculations of performance.

• I. Englander: *The architecture of computer hardware and systems software.*
  More elementary.
Other Good Text Books


Advanced Books


Related Topics


Copyright Notice

Most pictures in these slides are copyrighted. They may not be used elsewhere, see the respective books.

All figures from Computer Organization and Design: The Hardware/Software Approach, Second Edition, by David Patterson and John Hennessy, are copyrighted material (COPYRIGHT 1998 MORGAN KAUFMANN PUBLISHERS, INC. ALL RIGHTS RESERVED). Figures may be reproduced only for classroom or personal educational use in conjunction with the book and only when the above copyright line is included. They may not be otherwise reproduced, distributed, or incorporated into other works without the prior written consent of the publisher.
Overview

2. Boolean Logic, Circuits
5. Internal Memory.
6. External Memory.
8. CPU-Instructions Sets, Addressing Modes.
9. CPU Structure.
1. Historic Development

(a) Mechanical Area
(b) The first Generation and the von Neumann Machine.
(c) Later Developments.
(d) Current Computers from inside.
(e) Increase in performance.
(a) The Mechanical Area

- Old tools: Abacus.
  
  Picture omitted for copyright reasons

- Mechanical Computers: Schickhardt (1623), Pascal (1642), Leibnitz (1673).
  
  Picture omitted for copyright reasons
Punch Cards

- Invented 1801 Joseph-Marie Jacquard (France)

- For control of patterns at an automatic loom.
Picture omitted for copyright reasons
Charles Babbage (1791 - 1871)

Picture omitted for copyright reasons

Inventor of the first Computer (≈ 1835):
The **Analytical Machine**.

- Storage, Arithmetic, Logical Unit.
- Punch cards
- Printer
- Instructions contain
  - Arithmetical operations
  - Load/Store
  - Unconditional and **conditional** jumps.
- Addition 1 second, multiplication 1 minute.
- Construction not carried out before 1991.
Picture omitted for copyright reasons
The Mechanical Area (Cont.)

- George Boole: Development of Boolean Logic. (1854).
- Herman Hollerith
  - Construction of a machine based on punch card for Evaluation of the 1890 census in US.
  - Founder of the predecessor of IBM
- Electromechanical computers
  - J. A. Atanasoff (1936-1939) Special purpose computer for solving sets of linear equations.
  - Mark I (1944). First general purpose electro-mechanical computer.
- Turing (1936): Theoretical model of a computer.
(b) The First Generation: Vacuum Tubes

**ENIAC (completed 1946)**

- 18,000 vacuum tubes.
- 5000 additions per second.
- Programmed through rewiring of connections between components.
- Decimal digits represented by 10 vacuum tubes.
- Originally developed for calculating ballistic equations for artillery during World War II.
- Programming through rewiring.
Picture omitted for copyright reasons
The von Neumann Machine (1946)

- Theoretical model of a computer.
- Basis for most computers developed up to now.
- Main principles
  - Stored program-concept. Program in the memory.
  - Main components: Memory, ALU, Control unit, I/O
  - Both programs and data stored in the same memory.
  - Sequential execution.

Picture omitted for copyright reasons
Problems with the von Neumann Machine

- **Von Neumann bottleneck**: Time for data transfer between main memory and CPU is main hindrance for improving speed.

- **Semantic gap** between high level programming languages and implementation on a von Neumann machine.
(c) Later Developments

- **Generation 1 (1945 - 1957)**
  - *Vacuum tubes*, Magnetic drums
  - Machine code, stored programs.
  - *First Computer families* (compatibility) UNIVAC I, UNIVAC II (IBM).
  - Performance: 2 KB memory, 10 KIPS

- **Generation 2 (1958 - 1964)**
  - *Transistors*.
  - High level languages.
  - Floating point arithmetic.
  - Performance: 32 KB memory, 200 KIPS.

- **Generation 3 (1964 - 1971)**
  - *Integrated circuits*.
  - Semiconductor main memory.
  - Micro-programming
  - Multi-programming
  - Structured programming
  - 1 MB memory, 1 MIPS.
• **Generation 4 (1972 - 1977)**
  – Large scale integration
  – Networks
  – CD
  – Object-oriented languages.
  – Expert systems.
  – 8 MB memory, 10 MIPS.

• **Generation 5 (1978 - present)**
  – Very large scale integration. (VLSI)
  – 256 MB memory, 100 MIPS.
  – World wide web.
(d) Current Computers from Inside
A Motherboard

Picture omitted for copyright reasons

- 1 = Ports
- 2 = ISA slots
- 3 = PCI slots
- 4 = AGP slots
- 5 = CPU slots
- 6 = Chipset (Northbridge)
- 7 = Power connector
- 8 = Memory sockets
- 9 = I/O connectors
- 10 = Battery
- 11 = Chipset (Southbridge)
- 12 = BIOS chip
(e) Increase in Performance
Growth in CPU Transistor Count

Picture omitted for copyright reasons

(Source: Stallings).
Note that the scale is exponential.
Gap between Processor Speed and Memory

Picture omitted for copyright reasons

(Source: Stallings).
Note that the scale is exponential.
Performance Increase of Workstations

Scale: number of times faster than the VAX-11/780
Rate of performance increase is 1.6 years.
(Source: Patterson and Hennessy)
Attachment to Section 1
(Computer Systems Course)

(a) Moore’s Law

(b) Notations for the Inverter (not-gate)

(c) Representation of Boolean functions by example

(a) Moore’s Law

Moore’s law (1965): Number of transistors per Chip doubles every 18 months. In parallel the density of memory, speed of processors and of memory increase with similar exponential speed.
(b) Notations for the Inverter

A negater (not-gate) is denoted by:

If we have a negater followed by an and- or or-gate we write instead of the negater a circle at the corresponding input-line:

Similarly one can contract an and- or or-gate followed by a negater:
(c) Representation of Boolean Functions

Consider as an example the following function (which is randomly chosen):

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>f(x,y,z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ f(x, y, z) = 1 \] for the triples \((0, 0, 0), (0, 1, 0), (1, 1, 1)\) only.
Consider the function $g_1$ which is 1 for the triple $(0,0,0)$ only:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>$g_1(x,y,z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

If we define $t_1$ as

$$\neg x \land \neg y \land \neg z$$

we observe immediately:

$$t_1[x/a, y/b, z/c] = 1$$
iff $a = 0$ and $b = 0$ and $c = 0$
iff $g_1(a, b, c) = 1$

so

$$g_1(a, b, c) = t_1[x/a, y/b, z/b]$$ for all $a, b, c \in \{0, 1\}$ ,
t$_1$ represents $g_1$. 

26d
Consider the function $g_2$ which is 1 for the triple $(0,1,0)$ only:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>$g_2(x,y,z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

If we define $t_2$ as

$$\neg x \land y \land \neg z$$

we observe immediately:

$$t_2[x/a, y/b, z/c] = 1$$

iff $a = 0$ and $b = 1$ and $c = 0$

iff $g_2(a, b, c) = 1$

so

$$g_2(a, b, c) = t_2[x/a, y/b, z/b] \text{ for all } a, b, c \in \{0, 1\}$$

$t_2$ represents $g_2$.  

26e
Consider the function $g_3$ which is 1 for the triple (1, 1, 1) only:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>$g_3(x,y,z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

If we define $t_3$ as

$$x \wedge y \wedge z$$

we observe immediately:

$$t_3[x/a, y/b, z/c] = 1$$

iff $a = 1$ and $b = 1$ and $c = 1$

iff $g_3(a, b, c) = 1$

so

$g_3(a, b, c) = t_3[x/a, y/b, z/b]$ for all $a, b, c \in \{0, 1\}$,

$t_3$ represents $g_3$.  

26f
Consider now the tables for $f$, $g_1$, $g_2$, $g_3$:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>$f(x,y,z)$</th>
<th>$g_1(x,y,z)$</th>
<th>$g_2(x,y,z)$</th>
<th>$g_3(x,y,z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Now for all $a, b, c \in \{0, 1\}$

$$f(a, b, c) = g_1(a, b, c) \lor g_2(a, b, c) \lor g_3(a, b, c)$$

Define $t := t_1 \lor t_2 \lor t_3$.

$$t[x/a, y/b, z/c] = 1$$

iff $t_1[x/a, y/b, z/c] = 1$ or $t_2[x/a, y/b, z/c] = 1$

or $t_3[x/a, y/b, z/c] = 1$

iff $g_1(a, b, c) = 1$ or $g_2(a, b, c) = 1$ or $g_3(a, b, c) = 1$

iff $f(a, b, c) = 1$,

Therefore $f(a, b, c) = t[x/a, y/b, z/b]$ for all $a, b, c$, $t$ represents $f$.  

26g