
- Historical Perspective
- Survey of Computer Architecture Concepts
Goals of the Course

1. Demystify *software and hardware* layers of modern *computing systems*
2. Provide concrete experience
   - Complementary to the study of *computing* science
     
     "Computer science is no more about computers than astronomy is about telescopes"
     Edsger Dijkstra, Turing Award 1972
   - Let’s talk about *telescopes*, little about *astronomy*
3. Introduction to Master courses in *electrical and computer engineering*
4. Top-down pedagogical approach
   (complementary to the bottom-up one in INF559 – computer architecture)
Supercomputer = Computing System
Roadrunner, first PetaFLOPS computer (AMD Opteron and IBM Cell)

Storage/Networking

Data Center

Cloud Computing and Services

Virtualization
Embedded Device = Computing System
Mobile Device = Computing System

Nokia N810
Mobile Device = Computing System

Google Android
Mobile Device = Computing System

I have always wished that my computer would be as easy to use as my telephone...

My wish has come true: I no longer know how to use my telephone.

Prof. Bjarne Stroustrup, father of C++

→ smartphone sales outnumber all PC sales since Q4 2011

- Historical Perspective
- Survey of Computer Architecture Concepts
(Compressed) Historical Perspective

1936: Turing machine
Logical abstraction

http://www.computerhistory.org/timeline
1941–1944: Electronic programmable computers

Zuse Z3 (electromechanical, 1941)
Colossus (electronic, 1943)
Harvard Mark I / IBM ASCC (punch card, 1944)

http://www.computerhistory.org/timeline
1945: ENIAC First Turing-complete computer
John Eckert and J. Presper Mauchly
Digital, but decimal arithmetic
Program = cables and switches

http://www.computerhistory.org/timeline
1964: IBM System/360
Integrated circuits, family of 6 compatible computers and 40 peripherals
OS: millions of lines of assembly code

http://www.computerhistory.org/timeline
1969: UNIX — Ken Thompson and Dennis Ritchie
UNiplexed Information and Computing Service (economical redesign of MULTICS)
Rewritten in C with Brian Kernighan in 1973

http://www.computerhistory.org/timeline
1974: Xerox PARC — Alto
First interactive window system, menus, icons

http://www.computerhistory.org/timeline
1981: IBM – PC and compatibles
Cross-vendor microcomputers, progressively defining the hardware standards
Plagued by the lack of a reasonable operating system until 1996 (Windows NT, Linux), nourished non-sensical backward binary compatibility business

http://www.computerhistory.org/timeline
1983: GNU — Richard Stallman
Free software, free operating system (free like freedom)

http://www.computerhistory.org/timeline
(Compressed) Historical Perspective

1991: Linux — Linus Torvalds
Free OS kernel, originally inspired by Minix (A. Tanenbaum, VU Amsterdam)
Provides a complete OS: GNU/Linux
Runs on low-end mobile phones up to 1024-core multiprocessors (SGI Altix) or a
884736-core cluster (BlueGene/P, 3PFLOPS)

http://www.computerhistory.org/timeline
The (Simplified) UNIX Family

1970
BSD Family
BSD (Berkeley Software Distribution)
SunOS (Stanford University)
NextStep

1980
Bill Joy
Xenix OS
NeXT

1990
Solaris (SUN)
Darwin
GNU Project
Minix

2000
FreeBSD
NetBSD
OpenBSD
Darwin
MacOS X
GNU/Linux
Linus Torvalds
Minix
Andrew Tanenbaum

Unix Time-Sharing System (Bell Labs)
Ken Thompson
Dennis Ritchie (C Language)

System III & V Family
HP-UX
AIX (IBM)
UnixWare (Univel/SCO)
IRIX (SGI)

http://www.osdata.com/kind/history.htm
GNU/Linux

- Free software (open source)
- Robust and modern flavor of UNIX
- Most portable and largest range of supported devices
- Highly compatible with other OSes
- Modular and customizable, excellent code quality
- Lightweight: can be downsized for embedded devices
- Benefits from most OS innovations

- Historical Perspective
- Survey of Computer Architecture Concepts
**Stored Program, von Neumann Architecture**

- IAS Architecture (John von Neumann)
- SSEM “Baby”: 1948, Tom Kilburn, Victoria U. of Manchester
  - First implementation of the *stored program* concept in a real machine
- EDVAC: John Eckert, J. Presper Mauchly and John von Neumann
Pedagogical Example: LC-3 Processor (INF559)

Reference: *Introduction to Computing Systems: From Bits and Gates to C and Beyond*, Yale Patt and Sanjay Patel
# Pedagogical Example: LC-3 Processor (INF559)

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD/AND DR, SR1, SR2</td>
<td>0 0/1 0 1</td>
<td>DR, SR1, 0 0 0 SR2</td>
</tr>
<tr>
<td>ADD/AND DR, SR1, imm5</td>
<td>0 0/1 0 1</td>
<td>DR, SR1, 1 (\text{imm5: 5-bit signed immediate})</td>
</tr>
<tr>
<td>NOT DR, SR</td>
<td>1 0 0 1</td>
<td>DR, SR 1 1 1 1 1 1</td>
</tr>
<tr>
<td>LEA DR, label</td>
<td>1 1 1 0</td>
<td>DR 9-bit signed offset</td>
</tr>
<tr>
<td>LD DR, label</td>
<td>0 0 1 0</td>
<td>DR 9-bit signed offset</td>
</tr>
<tr>
<td>ST SR, label</td>
<td>0 0 1 1</td>
<td>SR 9-bit signed offset</td>
</tr>
<tr>
<td>BRnzp label</td>
<td>0 0 0 0 n z p</td>
<td>9-bit signed offset</td>
</tr>
<tr>
<td>JMP/JSR label</td>
<td>0 1 0 0 0/1 0 0</td>
<td>9-bit signed offset</td>
</tr>
<tr>
<td>JMPR/JSRR label</td>
<td>1 1 0 0 0/1 0 0</td>
<td>BaseR 6-bit signed index</td>
</tr>
<tr>
<td>LDR DR, BaseR, label</td>
<td>0 1 1 0</td>
<td>DR BaseR 6-bit unsigned index</td>
</tr>
<tr>
<td>STR SR, BaseR, label</td>
<td>0 1 1 1</td>
<td>SR BaseR 6-bit unsigned index</td>
</tr>
<tr>
<td>RET</td>
<td>1 1 0 1 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>TRAP trapvect8</td>
<td>1 1 1 1 0 0 0 0</td>
<td>trapvect8: 8-bit interrupt vector</td>
</tr>
</tbody>
</table>
Moore’s Law on Si-Based Semiconductors

CPU Transistor Counts 1971-2008 & Moore’s Law

Curve shows ‘Moore’s Law’: transistor count doubling every two years
Evolutions of the von Neumann Architecture

What Can We Do With All These Transistors?

- 60 years of evolution
  - Registers
  - Cache (local memory, memory hierarchy)
  - Instruction pipeline
  - Branch predictor, prefetch, speculative execution
  - Superscalar execution, in-order, out-of-order

Is it the end of the road for the von Neumann architecture?
Evolutions of the von Neumann Architecture

What Can We Do With All These Transistors?

- 60 years of evolution
  - Registers
  - Cache (local memory, memory hierarchy)
  - Instruction pipeline
  - Branch predictor, prefetch, speculative execution
  - Superscalar execution, in-order, out-of-order
  - Multi-thread processor
  - Multi-processors, multi-core, many-core
  - Specialization (hardware accelerators)
  - System-on-chip

Is it the end of the road for the von Neumann architecture?
### Evolutions of the von Neumann Architecture

#### What Can We Do With All These Transistors?

- 60 years of evolution
  - Registers
  - Cache (local memory, memory hierarchy)
  - Instruction pipeline
  - Branch predictor, prefetch, speculative execution
  - Superscalar execution, in-order, out-of-order
  - *Multi-thread processor*
  - *Multi-processors, multi-core, many-core*
  - *Specialization (hardware accelerators)*
  - *System-on-chip*

- Is it the end of the road for the von Neumann architecture?
System-on-Chip

Diagram showing various components and connections within a System-on-Chip (SoC) design. The components include:

- JTAG Scan
- ARM Processor
- Voltage Regulator
- System Controller
  - Advanced Int. Ctrl.
  - Power Mgt. Ctrl.
- PLL
- Osc
- RC Osc
- Reset Ctrl.
- Brownout Detect
- Power On Reset
- Prog. Int. Timer
- Watchdog Timer
- Real Time Timer
- Debug Unit
- PID Ctrl.
- EBI
- SRAM
- Flash
- Peripheral Bridge
- Memory Controller
- Application-Specific Logic
- PIO
- APB
- Ethernet MAC
  - USART0-1
  - SPI
  - Two Wire Interface
  - ADC0-7
- CAN
  - USB Device
  - PWM Ctrl
  - Synchro Serial Ctrl
  - Timer/Counter 0-2