Computer Architecture

Virendra Singh
Associate Professor
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

Computer Organization

Lecture 2
Historic Events

- **1623, 1642**: Wilhelm Strickland/Blaise Pascal built a mechanical counter with carry.
- **1823-34**: Charles Babbage designed difference engine.

http://www.youtube.com/watch?v=0anlyVGeWOI&feature=related
Babbage’s Difference Engine

• **Babbage Difference Engine**
  - Hand-cranked mechanical computer.
  - Computed polynomial functions.
  - Designed by **Charles Babbage** in the early to mid 1800s.
    - Arguably the world’s first computer scientist, lived 1791-1871.
  - He wasn’t able to build it because he lost his funding.

  □ His plans survived and this working model was built.
    ■ Includes a working printer!

Historic Events

• **1943-44:** John Mauchly (professor) and J. Presper Eckert (graduate student) built ENIAC at U. Pennsylvania.

• **1944:** Howard Aiken used “separate data and program memories” in MARK I – IV computers – *Harvard Architecture*.

• **1945-52:** John von Neumann proposed a “*stored program computer*” EDVAC (Electronic Discrete Variable Automatic Computer) – *Von Neumann Architecture* – use the same memory for program and data.
Most Influential Document

Theory of Computing

• Alan Turing (1912-1954) gave a model of computing in 1936 – *Turing Machine*.


* The question of decidability, posed by mathematician Hilbert.
History Continues

- **1946-52**: Von Neumann built the IAS computer at the Institute of Advanced Studies, Princeton – *A prototype for most future computers*.
- **1947-50**: Eckert-Mauchly Computer Corp. built UNIVAC I (Universal Automatic Computer), used in the 1950 census.
- **1949**: Maurice Wilkes built EDSAC (Electronic Delay Storage Automatic Calculator), the first stored-program computer.
What was Computing Like?

• A data processing application involved passing decks of punched cards through electromechanical “unit record” machines.

• Repetitive sort, calculate, collate, and tabulate operations ...
  – ... were programmed with hand-wired plugboard control panels.
Plugboard Control Panel

IBM 407 Accounting Machine (1949)
Plugboard Control Panel
Programming a Plugboard

• “Programming” was hand-wiring plugboards.

“Hmm, should I pass this parameter by value or by reference?”
Programming a Plugboard

- Plugboard wiring diagram
  - It doesn’t look too complicated, does it?
Data Processing

- Cards were punched manually at a keypunch machine.
  - Or they were punched automatically by unit-record equipment under program control.
Data Processing

- Cards were re-keyed on a **verifier** to ensure accuracy.
  - Good cards were notched at the top right edge.
  - Bad cards were notched at the top edge above each erroneous column.
Data Processing

- A sorter sorted cards one column at a time.
  - You had to run decks of cards multiple times through a sorter.

- Accounting machines performed arithmetic on card fields and printed reports.
Running a Data Processing Application ...

• ... meant passing decks of cards through a sequence of unit-record machines.
  
  – Each machine was programmed via its plugboard to perform its task for the application.
  – Each machine had little or no memory.
  – The punched cards stored the data records
  – The data records moved as the cards moved.

An entire work culture evolved around punched cards!
Von Neumann Bottleneck

• Von Neumann architecture uses the same memory for instructions (program) and data.
• The time spent in memory accesses can limit the performance. This phenomenon is referred to as von Neumann bottleneck.
• To avoid the bottleneck, later architectures restrict most operands to registers (temporary storage in processor).

John von Neumann (1903-1957)
Second Generation Computers

- 1955 to 1964
- Transistor replaced vacuum tubes
- Magnetic core memories
- Floating-point arithmetic
- High-level languages used: ALGOL, COBOL and FORTRAN
- System software: compilers, subroutine libraries, batch processing
- Example: IBM 7094
Third Generation Computers

• Beyond 1965
• Integrated circuit (IC) technology
• Semiconductor memories
• Memory hierarchy, virtual memories and caches
• Time-sharing
• Parallel processing and pipelining
• Microprogramming
• Examples: IBM 360 and 370, CYBER, ILLIAC IV, DEC PDP and VAX, Amdahl 470
C Programming Language and UNIX Operating System

1972

Now

CADSL
The Current Generation

- Personal computers
- Laptops and Palmtops
- Networking and wireless
- SOC and MEMS technology
- And the future!
  - Biological computing
  - Molecular computing
  - Nanotechnology
  - Optical computing
  - Quantum computing
Running Program on Processor

Processor Performance = \[ \frac{\text{Time}}{\text{Program}} \]

\[ = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Time}}{\text{Cycle}} \]

\text{(code size)} \times \text{(CPI)} \times \text{(cycle time)}

Architecture --> Implementation --> Realization

Compiler Designer \quad Processor Designer \quad Chip Designer
Abstraction and Complexity

• Abstraction helps us manage complexity
• Complex interfaces
  – Specify what to do
  – Hide details of how
• Goal: remove magic

Scope of this course

15 Mar 2013
Computer Organization@IIT Mandi
Computer Architecture

- Exercise in engineering tradeoff analysis
  - Find the fastest/cheapest/power-efficient/etc. solution
  - Optimization problem with 100s of variables

- All the variables are changing
  - At non-uniform rates
  - With inflection points
  - Only one guarantee: Today’s right answer will be wrong tomorrow

- Two high-level effects:
  - Technology push
  - Application Pull
Technology Push

• What do these two intervals have in common?
  – 1776-1999 (224 years)
  – 2000-2001 (2 years)

• Answer: Equal progress in processor speed!
• The power of exponential growth!
• Driven by Moore’s Law
  – Device per chips doubles every 18-24 months
• Computer architects work to turn the additional resources into speed/power savings/functionality!
### Some History

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1939</td>
<td>First digital computer</td>
<td>John Atanasoff (UW PhD ’30)</td>
</tr>
<tr>
<td>1947</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; transistor</td>
<td>Bell Labs</td>
</tr>
<tr>
<td>1958</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; IC</td>
<td>Jack Kilby (MSEE ’50) @TI Winner of 2000 Nobel prize</td>
</tr>
<tr>
<td>1971</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; microprocessor</td>
<td>Intel</td>
</tr>
<tr>
<td>1974</td>
<td>Intel 4004</td>
<td>2300 transistors</td>
</tr>
<tr>
<td>1978</td>
<td>Intel 8086</td>
<td>29K transistors</td>
</tr>
<tr>
<td>1989</td>
<td>Intel 80486</td>
<td>1.M transistors, pipelined</td>
</tr>
<tr>
<td>1995</td>
<td>Intel Pentium Pro</td>
<td>5.5M transistors</td>
</tr>
<tr>
<td>2005</td>
<td>Intel Montecito</td>
<td>1B transistors</td>
</tr>
</tbody>
</table>
Performance Growth

Unmatched by any other industry!
[John Crawford, Intel]

- **Doubling every 18 months (1982-1996): 800x**
  - Cars travel at 44,000 mph and get 16,000 mpg
  - Air travel: LA to NY in 22 seconds (MACH 800)
  - Wheat yield: 80,000 bushels per acre

- **Doubling every 24 months (1971-1996): 9,000x**
  - Cars travel at 600,000 mph, get 150,000 mpg
  - Air travel: LA to NY in 2 seconds (MACH 9,000)
  - Wheat yield: 900,000 bushels per acre
Thank You