U B

Computer Systems and Architecture

Dr Peter Breuer, School of Computer Science
January 2012

Textbooks

- Computer Organization and Design – The Hardware/Software Interface (Patterson and Hennessy) [Patt] – copies in school library, cheap from Amazon
- Structured Computer Organization (Tannenbaum) [Tann] – cheap from Amazon
- Computer Organization and Architecture (Stallings) [Stall] - cheap from Amazon

Also

- Anything else by Tannenbaum or Stallings
- Computer Architecture – A Quantative Approach (Hennessy and Patterson)

Computer Systems and Architecture

- Module code 06-19340
- Two lectures per week
  - Tuesday 3pm in G34, Mech. Eng. Blg.
  - Friday 2pm in G29, Mech. Eng. Blg.
- Occasional exercise class/extra lecture
  - To Be Determined
- My office hour
  - Monday 2pm-3pm, Room CS F132
- Module is 80% exam, 20% assessed
- You'll find lecture summaries online at **http://www.cs.bham.ac.uk/~ptb/lectures/CSA.html**
- Email me at **ptb@cs.bham.ac.uk**

Module Aims

- Provide an understanding of the

  fundamental concepts

  and principles behind modern computers.
- Introduce the basic

  hardware components

  of modern computers and their **internal design and operation**.
- Develop an appreciation of how

  hardware supports programming languages.
- Provide a basic understanding of

  networking

Practical

- About half-way through term you will be set an exercise:

  $\gg$ Make this CPU go faster! $\ll$
- You will have a Java simulation of a real CPU
  - it's a public open source project: jMIPS on sf.net
- You need to watch out in lectures for tricks you can use
- At the end of term we'll measure your attempts
- Marks will be awarded primarily for speed
  - also for cleverness, sophistication, etc.
  - for strength of your contribution to public good
  - to count for 20% of total

Why should I want to know this stuff

Why wouldn't you!

- A mechanic needs to know how an engine works (and so does a racing driver) in order to understand and improve
- This gives you the 'view from the metal up' of programs
  - if you only know a thing in one way, then you do not know it at all – Marvin Minsky
- A programmer needs to know this in order to
  - do systems programming (drivers, schedulers, . . .)
  - write a compiler
  - optimize code for systems with limited resources
  - understand why your program is slow and how to fix it
  - appreciate deeply how a program operates in the real world

What is Computer Architecture?

- Consists of many levels [Tann 1.1.2]

Level 5	High-level Language	architecture
Level 4	Assembly Language	
Level 3	Operating System	
Level 2	Instruction Set	
Level 1	Micro-architecture	organization
Level 0	Digital Logic	
Level -1	Components	

- Sometimes divided into
  - architecture – what the programmer can see
    - instruction set, bits per byte, interrupt convention
  - organization – what the programmer cannot see
    - components implementing the architecture

This course covers . . .

- Mostly
  - Level 1: Micro-architecture
  - Level 2: Instruction Set Architecture
- Occasionally
  - Level 0: Digital Logic
  - Level 3: Operating System
  - Level 4: Assembly Language
- Other courses
  - Level 5 covered in Year 1,2 Software Workshop
  - Level 1-5 covered in Year 1 Foundations of CS
  - Level 1-4 covered in Year 3 Compilers and Languages
  - Level 3 covered in Year 3 Operating Systems

Some History

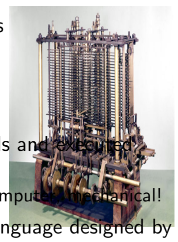
- Prehistoric calculating machines were fingers, stones.
- First civilizations used . . .
  - Africa: necklaces (think 'rosary') and bracelets
  - Peru: knotted strings
  - Greece: Archimedes invented abacus in 287BC
- First mechanical calculating machine: the Antikythera (Greek; used for astronomical calculations)

- Second mechanical calculating machine produced in the 17th century . . .

First Modern Period Calculators

- Blaise Pascal, 19, invented a gear-based machine for addition and subtraction in 1642
- Leibnitz added multiplication and division to the machine in 1672
  - a basic pocket calculator, 300 years ago!
- The Charles Babbage (1792-1871) Difference Engine
  - only addition and subtraction but hard-copy output (punched copper plate)
  - hired the first programmer – Ada Lovelace, daughter of Lord Byron
    - the programming language named after her is 'Ada' (not ADA).

Babbage's Analytical Engine



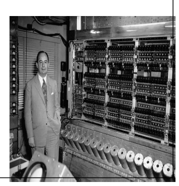
- The Analytical Engine had four parts
  - store (memory)
  - mill (CPU)
  - input (card reader)
  - output (card punch)
- Read instructions from punched cards and executed them.
  - world's first fully programmable computer, mechanical!
- Programmed in a simple assembly language designed by Ada Lovelace.
- Never worked (until modern times) – the gears could not then be manufactured with the required precision and friction would have been too great.

The Very First Electronic Computers

- Fundamentally similar to Babbage's Analytical Engine
- No significant advances until 1930s
  - Konrad Zuse built an electronic calculating machine (Z1) 1936 using telephone relays and punched tape, destroyed in WW2. Z2 in 1939. Z3 1941 was 64-bit. World's first computer startup company.
  - John Atanasoff invented a machine called ABC (Atanasoff-Berry Computer) with binary arithmetic and a capacitor-based memory (DRAM!) in 1939. Never worked properly. The 1973 decision of the patent suit Honeywell v. Sperry Rand named him the inventor of the first automatic electronic digital computer
  - George Stibbitz' Complex Calculator worked in 1940
  - Howard Aiken rebuilt Babbage's Analytical Engine design using electronic relays in 1944 and it (Mark 1) worked

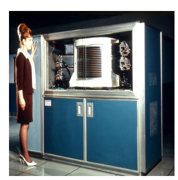
The von Neumann Computer

- John von Neumann was the world's foremost mathematician (his student, John H. Conway, the Prince of Games and Surreal Numbers and inventor of the Game of Life, was my supervisor for a while).
- 1945 EDVAC proposal for storing the program along with the data in memory, not on punched tape.
- 1952 IAS completed at Princeton; the 'stored program' computer.
- The architecture for almost all modern machines follows the IAS design



Birmingham's KDF-9

- 8K 48-bit words of main memory
- $8\mu s$ (125KHz) cycle time
- 2 magnetic tape decks and paper tape I/O - no printers, disks or other I/O devices were thought necessary.



Birmingham's First Upgrade

- Machine envy set in in 1968 (the KDF-9 was c. 1965) . . .
- Then the machine was upgraded with a 4M word disk weighing over a ton, a punched card i/o reader and punch, and a high-speed printer.

Birmingham's First Mini

- The KDF-9 was later replaced by a PDP-8 minicomputer
  - We later had a small PDP at school – I forget which. I wrote my first program for it (it calculated pi to arbitrary precision).

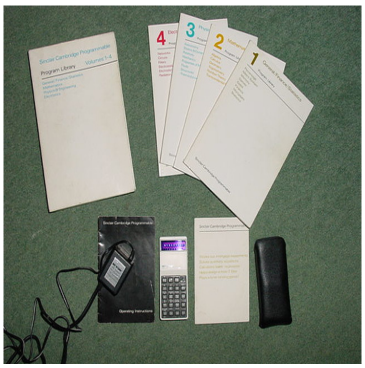
The Integrated Circuit Third Generation Revolution

- 2G computers used thousands of discrete transistors
- Jack Kilby and others invented the integrated circuit (IC) which put many transistors on a single microchip
- This led to cheaper computers and was responsible for the success of the minicomputer trend in the 1970s
- The IBM 360 became a workhorse for both scientific and business computing (I loaded magtapes on one).
- The 360 allowed 'multiprogramming'; it could run several programs at once, and could emulate other machines
  - us machine-room operators kept the machine loaded with a spool of pending jobs, loaded from punched cards.
  - when a compiler was needed, it asked us for the magtape!
  - it ran other jobs while waiting for the tape to be loaded.

Very Large Scale Integration

- IC-based minis were still (large) cupboard sized!
- VLSI technology pushed the number of components per chip up from 100s or 1000s to 10,000s and eventually 10,000,000s!
- This sparked a home compter revolution of microcomputers (Zilog Z80, Intel 8080, MosTech 6502) with the power of minis but in a shoebox-size package.
- Early home computers cost one or two hundred pounds (6 weeks student rent) and used the TV as monitor.
- Mainly used for word-processing, spreadsheets, and games.

The 1973 18-Byte Sinclair Programmable Calculator



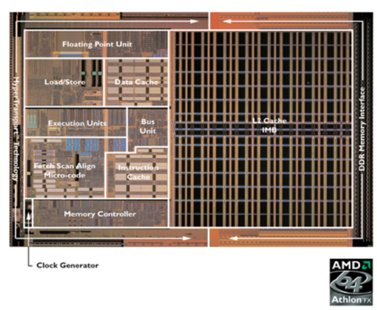
The IBM PC arrived in 1981

- Intel 8088 CPU (8 bit)
- MS-DOS operating system
- Windows came much later (after Apple introduced the Lisa with a GUI, then the Mac).
- IBM made design public so others could build add-ons.
- Clones were sold at lower cost.
- Microsoft and Intel eventually gained control of the market created by IBM and the result is the modern PC industry!

Modern VLSI Microprocessors

- The progression is as follows ...
- The 2G machines of the '60s had a few 1000s of transistors and filled a room (think 'circuit boards').
- Early VLSI chips had a few 10,000s of transistors.
- The modern AMD Athlon FX has 105,000,000 transistors on one microchip area 1.4cm sq.
  - includes 1MB on-chip cache RAM
  - about as much as the KDF-9's 1 ton disk drive in 1968.
  - Transistors are now 0.1 microns in width and shrinking.

AMD Athlon FX



Scales ...

- Modern processors are about 1cm $\times$ 1cm in size.
- Their transistors are about 0.1μm across, so 100,000 fit on an edge of a processor chip (with only 100,000,000 transistors on chip in total there is still 90% of space to spare for traces and insulation and other components).
- If the chip were the size of Birmingham (10km by 10km), a transistor would be
  - the size of a sheet of A4 paper or less and about 1m apart.
- The original transistors were about the same size as microchips are now – high power output transistors for speaker amps were and are still about a quarter-inch across or more.

Moore's Law

- "Transistor density doubles every 18-24 months".
- Broadly true over the last 35 years.
- Means transistor size (die technology) decreases by 30% every 18-24 months, 15-20% every year.
- CPU speed is approximately inversely proportional to the length of the circuit paths so CPU speeds increased by 10% or more every year for a long time.
  - nowadays technological limits tend to mean that we are getting more computers per chip (dual/quad core ...) instead
- Eventually physics means we will need something else.
  - quantum effects interfere with reliability at small scales
- Quantum computing is possible, but very very far off.
  - more likely first to move to photonics from electronics.

Next Lecture

- The next lecture will be about:
  - Digital logic and the smallest hardware components in modern computers.
  - These units will eventually be used to put together the registers, multiplexers, etc. that make up the important parts of a functioning CPU.
  - You should concentrate on preparing yourself for how digital circuits work by reading sections from your course texts (or looking up 'digital logic' and 'logic gate' on Wikipedia!).