Architecture et Système

Stefan Schwoon

Cours L3, 2014/15, ENS Cachan

Organisation

Lectures: Monday 13:45–15:45 h, C321

Tutorials: Friday 13–16 h, C411

Slides and other course material:

`http://www.lsv.ens-cachan.fr/~schwoon/`

E-mail: `schwoon@lsv.ens-cachan.fr`

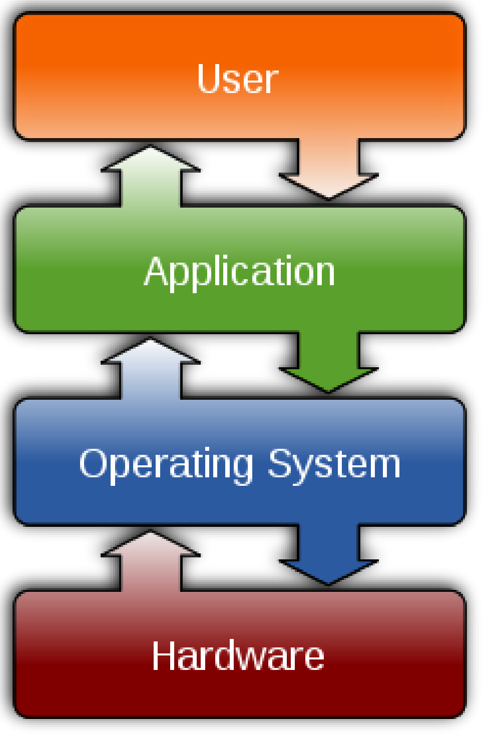
Office: RH-B-112 (LSV)

Subject

How does a computer work? What happens inside?
Hardware and software aspects – architecture / operating systems

In hardware design, one distinguishes architecture (logical aspects) from implementation (physical details).

In software, one distinguishes the operating system (which enable the operation of the computer, control the resources, etc) from applications (which solve the user's problems).



Subject

Goals:

- Better comprehension of computer behaviour
- Acquisition of practical knowledge for everyday tasks, useful in experimentation, implementation, . . .

Topics:

- Physical layer, bit/word-level operations, data representation
- Organisation of a processor, components, peripherals
- Components of an operating system (based on POSIX): processes, file system, memory management, scheduling, . . .
- Assembly language and C

Literature

Architecture:

John P. Hayes, *Computer Architecture and Organization*, McGraw Hill (3rd edition)

Operating systems

Andrew S. Tanenbaum, *Operating systems*, Prentice Hall

Today’s contents

A brief history of computer architecture

Common elements of architecture

Transistors / Logical circuits (“hardware implementation”)

Early computing devices

In a loose sense: abacus or slide rule

- Facilitate calculus, but no automatism - human executes algorithms
- Use of precomputed tables (e.g., for logarithms), calculated by hand

First automatic calculators (17th to 19th century)

- Blaise Pascal / Gottfried Leibniz / Thomas de Colmar
- addition, subtraction / multiplication, division / mass production
- mechanical devices, based on decimal notation

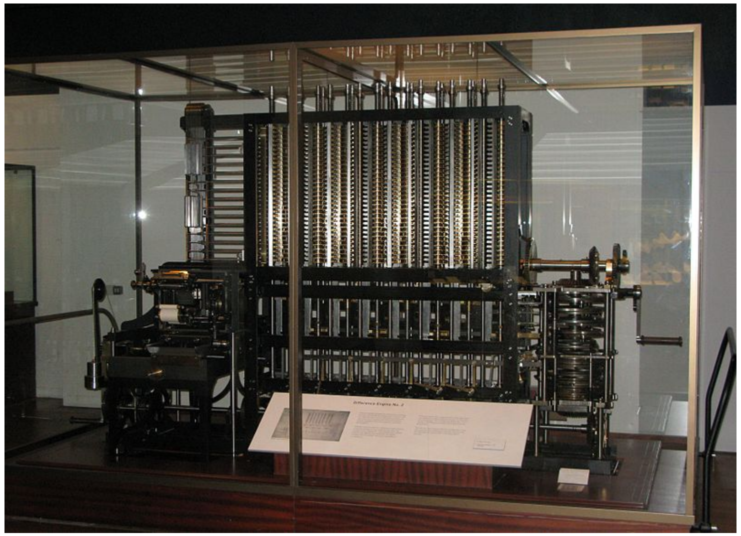
First programmable devides

Difference Engine (1832) by Charles Babbage

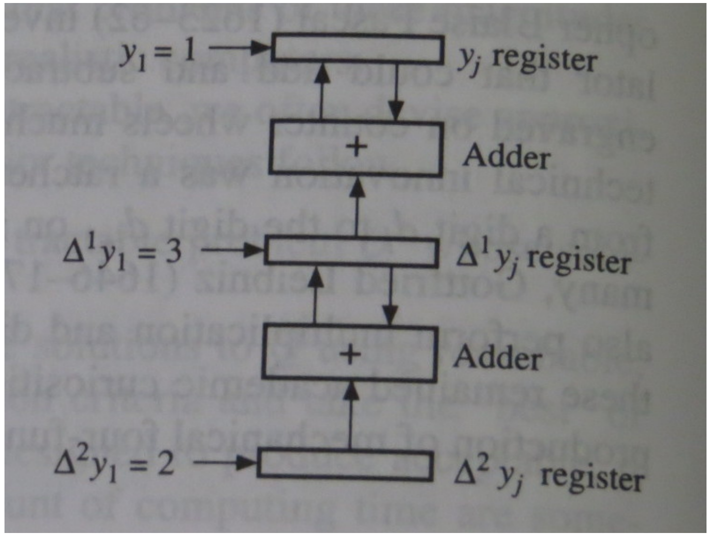
One operation (addition) applied simultaneously to several registers

“Programming” consists in determining the initial values

Can compute polynomials, automatic production of mathematical tables



Programming of Difference Engine for computing series of squares:



Analytical Engine (only partially constructed)

Multiple operations (addition/multiplication/. . .)

Programmable sequence of operations + conditional jump

Late 19th/early 20th century

Further improvements in mechanics and usability

Mostly special-purpose devices:

- *computers* used by engineers, navigators, military, . . .
- other automation (tabulating, production)

Still mostly mechanical devices based on decimal computation

The 1930s

Konrad Zuse's Z1 (1938) and Z3 (1941):

- use of electromechanical relays
- binary calculus
- loops, but no conditional jumps

The Turing Machine

Alan Turing (1936)

Theoretical conception of a universal computing device (*Turing machine*)

(infinite) tape to store data + one internal state

machine can read one symbol from the tape at a time

move left/right + change state/symbols according to a given set of rules

Universal Turing Machine: TM that can execute other turing machines

Undecidability result, concept of “programs as data”

First-generation electronic computers

Mark I (1944) - used for code-breaking in World War II

ENIAC (1946) - first electronic computer

- constructed at University of Pennsylvania
- weight: 30 tons; 18.000 vacuum tubes
- still based on the decimal system (20 ten-digit registers)
- programmable calculation sequences by (re-)plugging cables
- time for performing multiplication: 3 ms
- used by military for computing mathematical tables

The von Neumann area

EDVAC (1951)

working on binary basis

program stored in memory

instructions of the form (a_1, a_2, a_3, a_4, op):
perform *op* on data at a_1 and a_2, store result at a_3, next instruction at a_4.

conditional jump: compare data at a_1, a_2, then jump to a_3 or a_4.

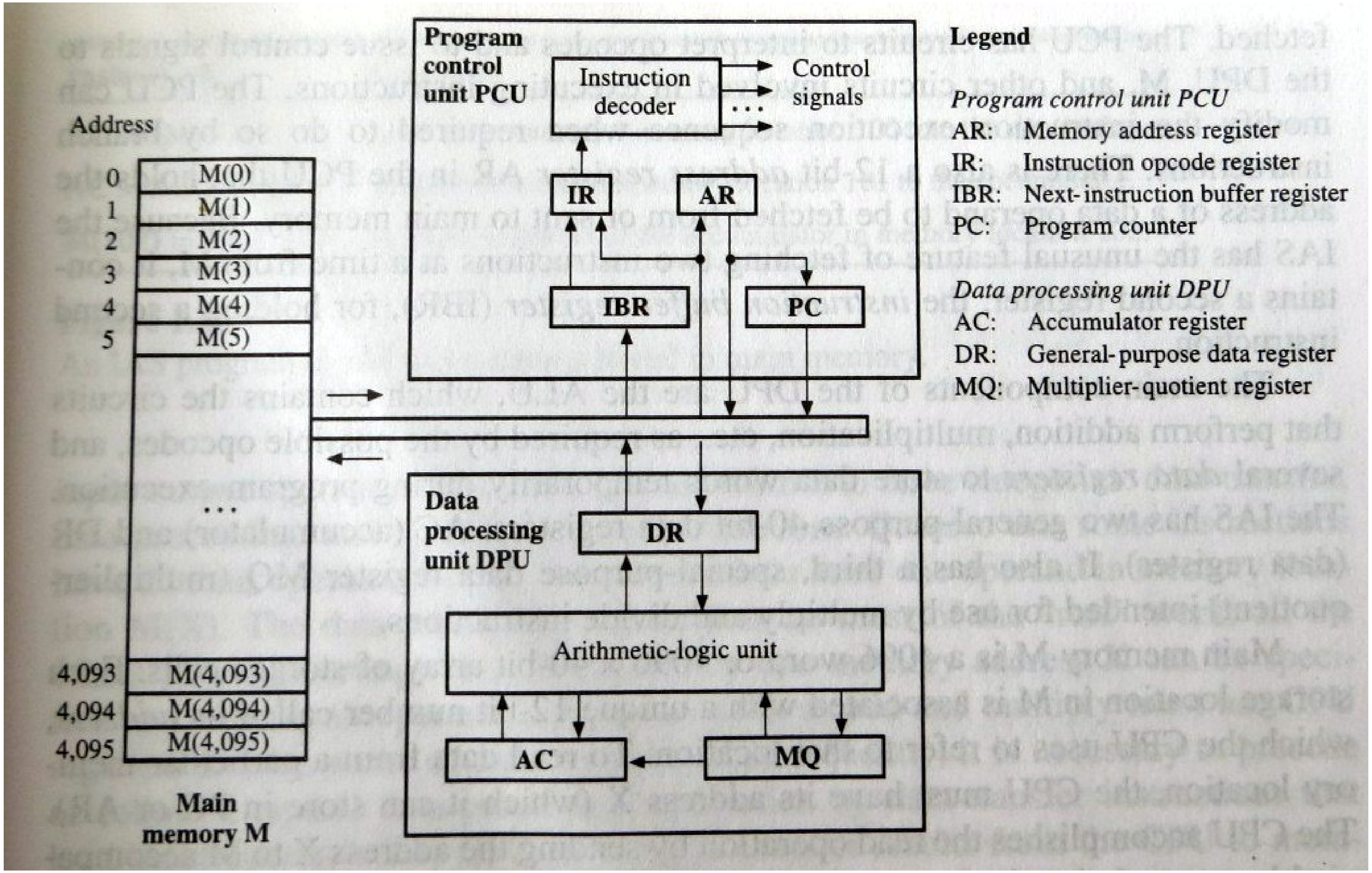
The IAS machine (designed at Princeton University)

memory: $4096 = 2^{12}$ memory cells (‘words’) of 40 bit each

notions of *program-control unit* and *data-processing unit* (CPU and ALU)

considered as the prototype of modern computer architecture

Architecture of the IAS machine



Instruction set of the IAS

A memory word can be interpreted as data or as instructions

Data interpretation: integer or fixed-point between -1/+1

Instruction: one word = 2 instructions with 20 bits

single-address instructions (*op*, *a*) operating on memory and registers:

– op: 8-bit operation code / a = 12-bit address

transfer data between memory and registers/between registers

add/multiply data

control operations (conditional jumps, self-modifying code)

Example IAS program: copying a memory block

0	999	Constant (count N).
1	1	Constant.
2	1000	Constant.
3L	AC := M(2000)	Load A(I) into AC.
3R	AC := AC + M(3000)	Compute A(I) + B(I).
4L	M(4000) := AC	Store sum C(I).
4R	AC := M(0)	Load count N into AC.
5L	AC := AC – M(1)	Decrement count N by one.
5R	**if** AC ≥ 0 **then go to** M(6, 20:39)	Test N and branch to 6R if nonnegative.
6L	**go to** M(6, 0:19)	Halt.
6R	M(0) := AC	Update count N.
7L	AC := AC + M(1)	Increment AC by one.
7R	AC := AC + M(2)	Modify address in 3L.
8L	M(3, 8:19) := AC(28:39)	
8R	AC := AC + M(2)	Modify address in 3R.
9L	M(3, 28:39) := AC(28:39)	
9R	AC := AC + M(2)	Modify address in 4L.
10L	M(4, 8:19) := AC(28:39)	
10R	**go to** M(3, 0:19)	Branch to 3L.

Second-generation computers (mid 50s-60s)

Physical developments:

- replacement of vacuum tubes by transistors
- smaller, cheaper, faster, more reliable

Architectural developments:

- improved instruction sets: indirect addressing, index registers
- floating-point registers and operations
- call/return instructions

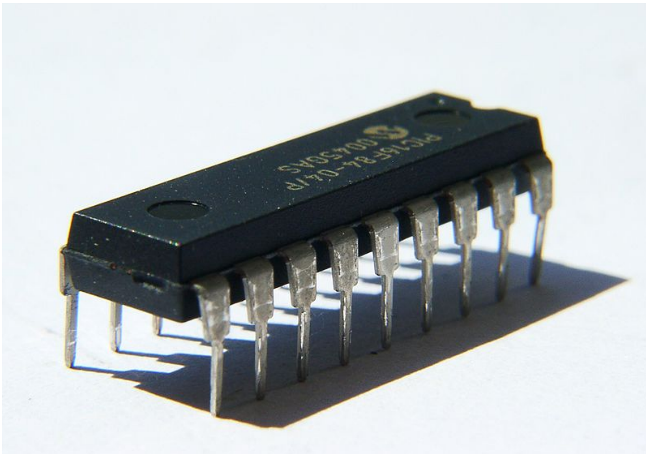
User-interface developments

- Programming languages and compilers
- First simple operating systems

Third-generation computers (1960s-70s)

Arrival of integrated circuits (IC)

- large number of transistors assembled in very small space
- faster operations due to reduced switching time



Small, fast memory (cache) can be integrated on same IC as processor; main memory (slower) on other ICs

Development of specialized devices for direct memory transfer

First multi-user timesharing operating systems:

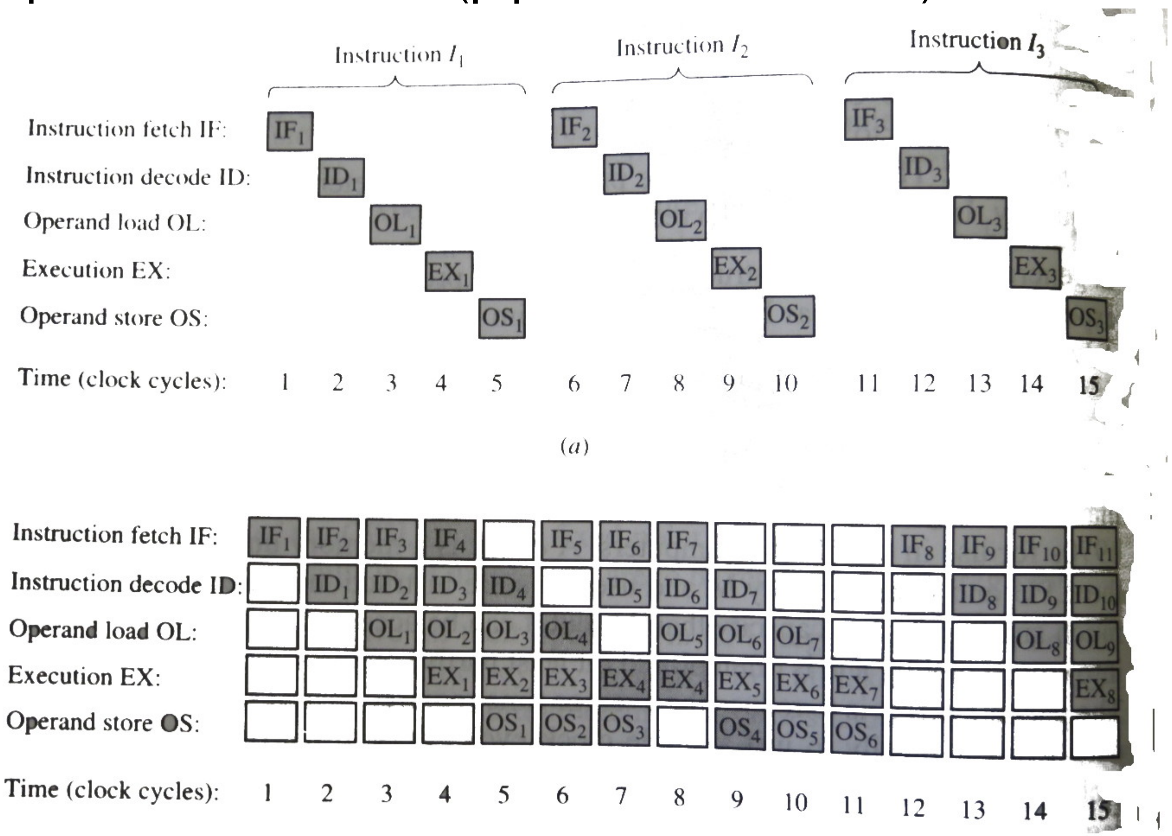
- interactive applications
- interrupt-driven scheduler
- CPU running in “supervisor” or “user” mode
- file systems
- development of Unix in the early 1970s

From the 1970s to today

Ever faster and cheaper circuits

Computers become a mass product (desktops, laptops, smartphones, . . .)

Superscalar computer architecture (pipelined execution)



Concurrency

Concurrency becomes more important:

- specialized components operating in parallel (graphics processor, FPU)
- multi-core CPUs
- computer networks

$\rightarrow$ study of concurrent algorithms