

# Processors

Kasper Brink, Nils Jansen, and Cynthia Kop



# Lecturers



Kasper  
Brink

[K.Brink@cs.ru.nl](mailto:K.Brink@cs.ru.nl)



Nils  
Jansen

[n.jansen@science.ru.nl](mailto:n.jansen@science.ru.nl)



Cynthia  
Kop

[C.Kop@cs.ru.nl](mailto:C.Kop@cs.ru.nl)

# Student Assistants

1. Stefan Beneschanscher, [S.Beneschanscher@student.ru.nl](mailto:S.Beneschanscher@student.ru.nl)
2. Jelle Besseling, [J.Besseling@student.ru.nl](mailto:J.Besseling@student.ru.nl)
3. Jen Dusseljee, [jendusseljee@gmail.com](mailto:jendusseljee@gmail.com)
4. Ciske Harsema, [C.Harsema@student.ru.nl](mailto:C.Harsema@student.ru.nl)
5. Luko van der Maas, [L.vanderMaas@student.ru.nl](mailto:L.vanderMaas@student.ru.nl)
6. Jordi Riemens, [jordi.riemens@student.ru.nl](mailto:jordi.riemens@student.ru.nl)
7. Niek Janssen, [niek.janssen@student.ru.nl](mailto:niek.janssen@student.ru.nl)
8. Rick van der Wal, [R.vanderWal@student.ru.nl](mailto:R.vanderWal@student.ru.nl)
9. Rico te Wechel, [R.teWechel@student.ru.nl](mailto:R.teWechel@student.ru.nl)

# Introduction - Organization

Contents and information: **Brightspace**

Enrolling: **OSIRIS**

Book: Andrew S. Tanenbaum: Structured Computer Organisation, sixth edition, Pearson, 2013, or fifth edition, Pearson, 2006.

# Teaching Methods

## Lectures

- purpose of the **lectures** is to present an overview of the contents, **reading materials** are meant to deepen the understanding. The material will appear shortly after the lecture on Brightspace (click on the respective lecture in the content section).

## Exercise Classes

- discussion of (previous) weekly assignments
- further deepen the contents of the lectures
- distribution to groups/student assistants will be done automatically in the next days

## Practical Assignment

- build CPU
- will be introduced after a few weeks
- needs to be handed in by groups of two

# Lecture and Werkcollege

- **Lecture** on Wednesday at 13:30
  - Lecture highlights important contents
  - Read the material in the book
- **Exercise Class** on Friday at 13:30 or 15:30
  - Starts on 16.11.2018
  - Discussion of homework and additional exercises

# Weekly Assignments

The weekly assignments are not mandatory, but we strongly encourage all students to take part.

- The assignments must be submitted in **groups of two**.
- The assignments must be handed to the respective student assistant via email **no later than Tuesdays at 18:00**.
- The grading scheme for the weekly assignments is NSI/O/V/G.
- On Brightspace on the day of the lecture
- No assignment this week!

**This information is not fixed yet, may change!**



# Practical Assignment

- The **practical assignment** will be announced and uploaded in a few weeks.
- Task: Design your own CPU based on a partial description that is given to you
- Graphical simulation tool Hades: <https://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/index.html>
- Needs to be submitted **in groups of two!**

# Evaluation

- The final grade will be the (rounded) average of both partial grades of the exam and the practical assignment, provided you received at least a 5 for each part. Otherwise, the final grade will be the minimum of the two partial grades.
- You can get a bonus point from the weekly assignments, if you have no more than two insufficient grades there

This information is not fixed yet, may change!

## Bonus for Weekly Assignments

- There are 6 assignments
- If you get at least 4 sufficient marks for the assignments, you will get a bonus of one point that will be **added to the lower of your two partial grades.**

# Consequences of Fraud (Cheating)

- no grade for the respective assignment or exam
  - no resit in the same academic year
  - registration in the dossier and no “cum laude” in the diploma
- 
- small and identifiable fraud:  
new or additional assignment
  - serious fraud:  
exclusion from university possible!

# Problem Solutions

- too little time to finish the processor
  - sign up for resit
- evaluation of partial solution
  - always account for the sources of your solution
- cooperation is not going well
  - talk with the student assistants or teachers



# Responsibilities

- Homework
- Reading
- Practical assignment
- Active participation in the lecture



# Goals

Insights in the structure of the **hardware of a computer** and into **the instructions of processors**.

- You know the basic principles of computer organization.
- You can describe the different abstraction levels of the architecture of a modern computer.
- You can explain the relationship between a processor's organization and its instruction set.
- You know some ways to improve the performance of a processor.
- You have some knowledge of circuit design, processor design and assembly programming.

# Overview

- approach: bottom-up
- introduction to computer structure (today)
- CPU structure
- electrical signals
- how is code executed? instruction set architecture, machine code
- assembly as simple programming language

# Lectures

- 1.Von Neumann Architecture/CPU Structure
- 2.Boolean Algebra, Gates and Circuits
- 3.Numbers and Memory
- 4.Execution Cycle, Data Path
- 5.Machine Code
- 6.Assembly Language
- 7.Assembly Programming and Pipeline

This information is not fixed yet, may change!



# CPU Structure: Von-Neumann-Architecture

Nils Jansen



# Machine Code vs. High-level Language

```
I FOX 12:01a 23- 1
A 002000 C2 30 REP #$30
A 002002 18 CLC
A 002003 F8 SED
A 002004 A9 34 12 LDA #$1234
A 002007 69 21 43 ADC #$4321
A 00200A 8F 03 7F 01 STA $017F03
A 00200E D8 CLD
A 00200F E2 30 SEP #$30
A 002011 00 BRK
A 2012

r
PB PC NUmxDIZC .A .X .Y SP DP DB
; 00 E012 00110000 0000 0000 0002 CFFF 0000 00
g 2000
BREAK

PB PC NUmxDIZC .A .X .Y SP DP DB
; 00 2013 00110000 5555 0000 0002 CFFF 0000 00
n 7f03 7f03
>007F03 55 55 00 00 00 00 00 00 00 00 00 00 00 00 00:W .....
```

```
if(top!=self) {
    function calcWidth() {
        var wW = 0;
        if (typeof window.innerWidth == 'number') (
            wW = window.innerWidth;
        ) else if (document.documentElement.clientWidth > 0) (
            wW = document.documentElement.clientWidth;
        ) else if (document.documentElement.clientWidth > 0) (
            wW = document.documentElement.clientWidth;
        ) else if (document.body.clientWidth > 0) (
            wW = document.body.clientWidth;
        )
        if (sH = document.documentElement.scrollHeight || docume
            var wH = window.innerHeight || docume
            sW = !document.all && (sH > wH)
            , 'wid
```

Humans can't understand machine language, using high-level language directly on the computer is not efficient

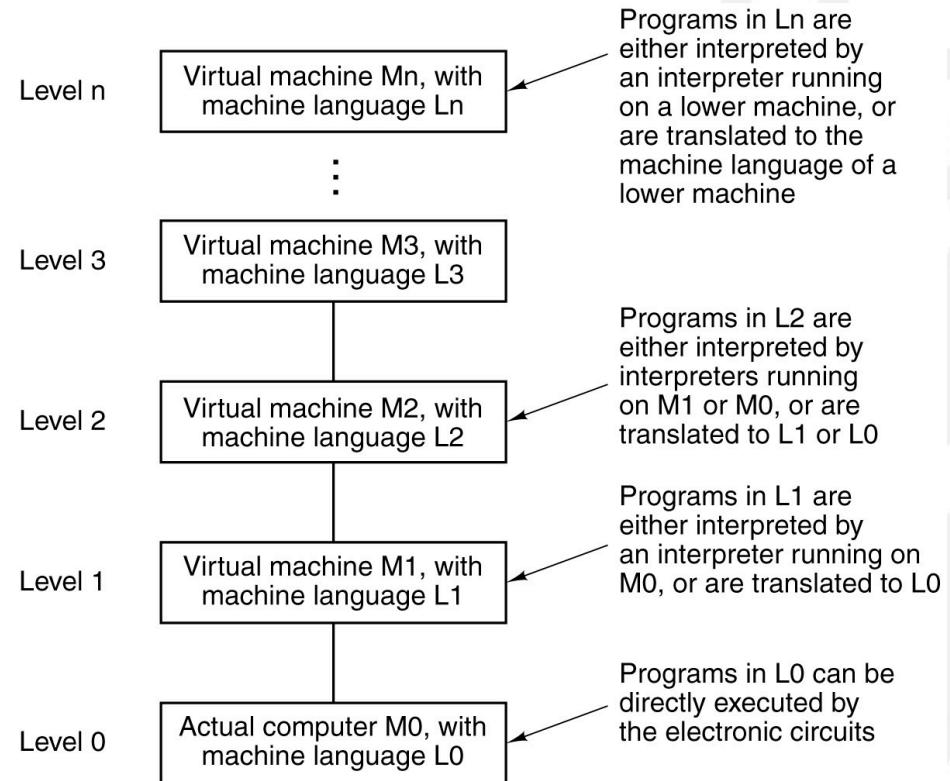


# Languages, Levels, and Virtual Machines

translation vs. interpretation

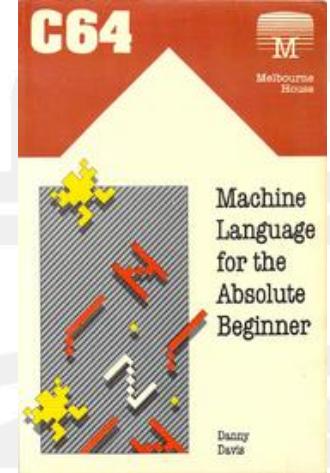
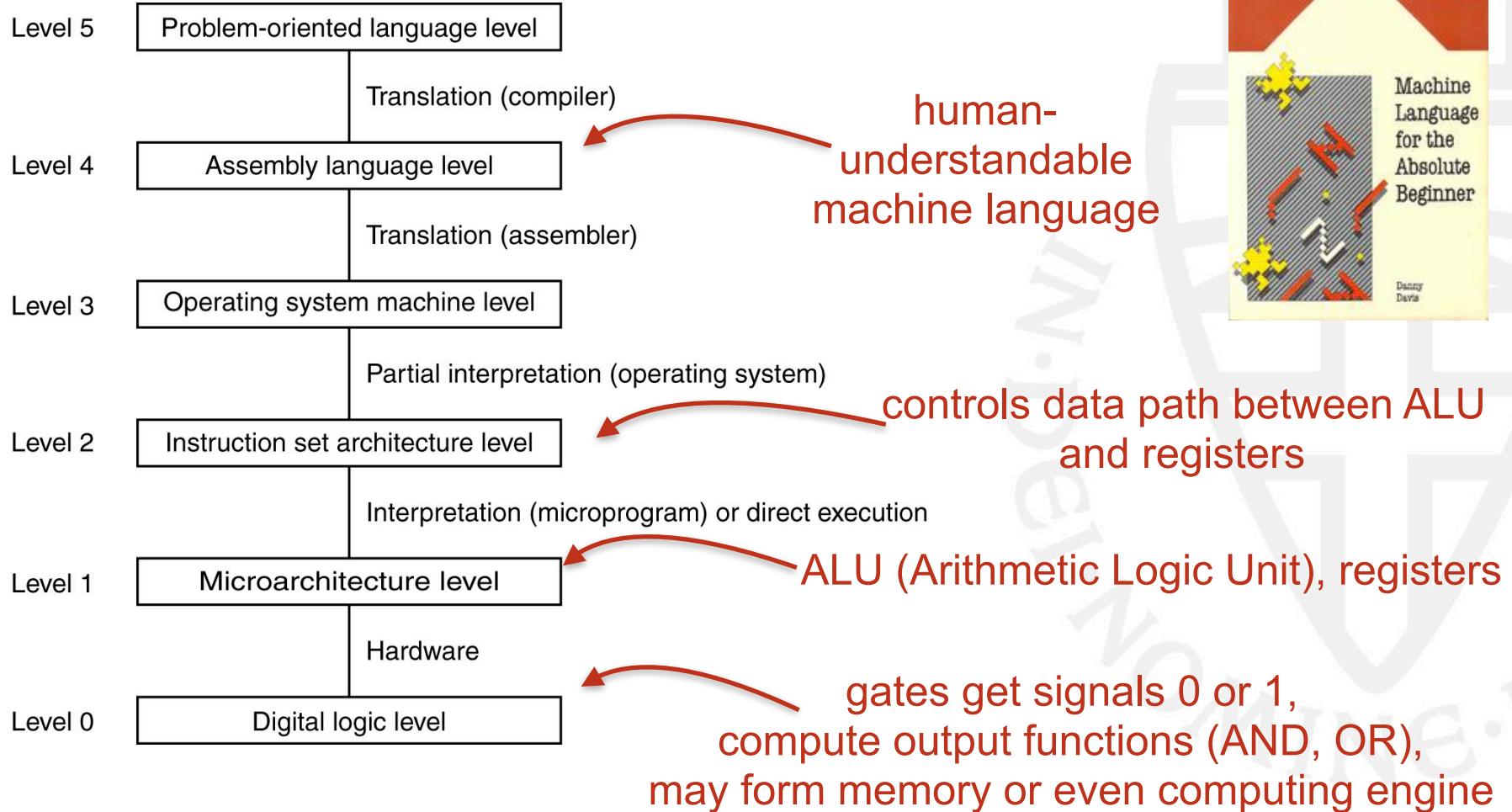
cost-efficiency vs. usability

specific machine language  
for each machine



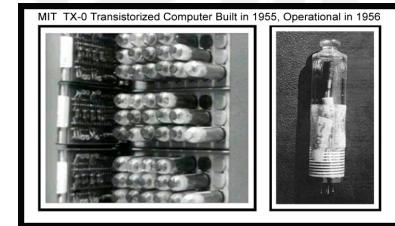
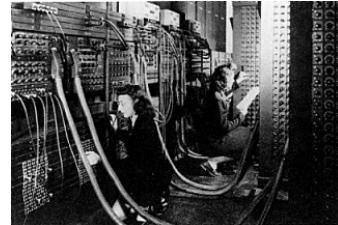
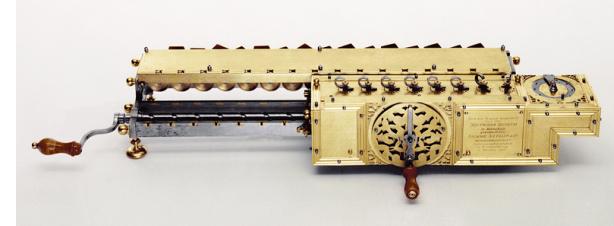
A multilevel machine

# A Six-level Computer

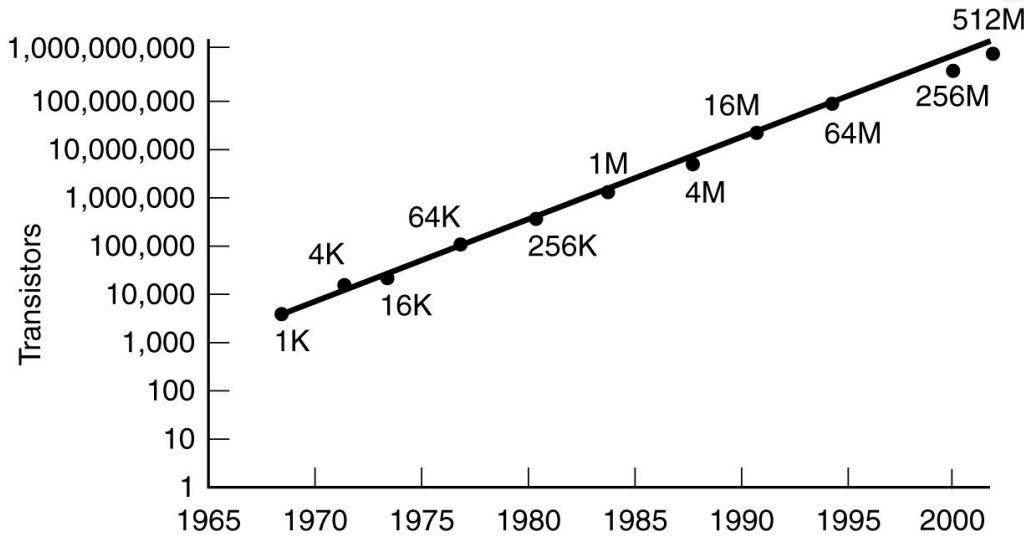


# Computer Generations

- Zeroth Generation  
Mechanical Computers (1642 – 1945)
- First Generation  
Vacuum Tubes (1945 – 1955)
- Second Generation  
Transistors (1955 – 1965)
- Third Generation  
Integrated Circuits (1965 – 1980)
- Fourth Generation  
Very Large Scale Integration (1980 – ?)



## Moore's Law



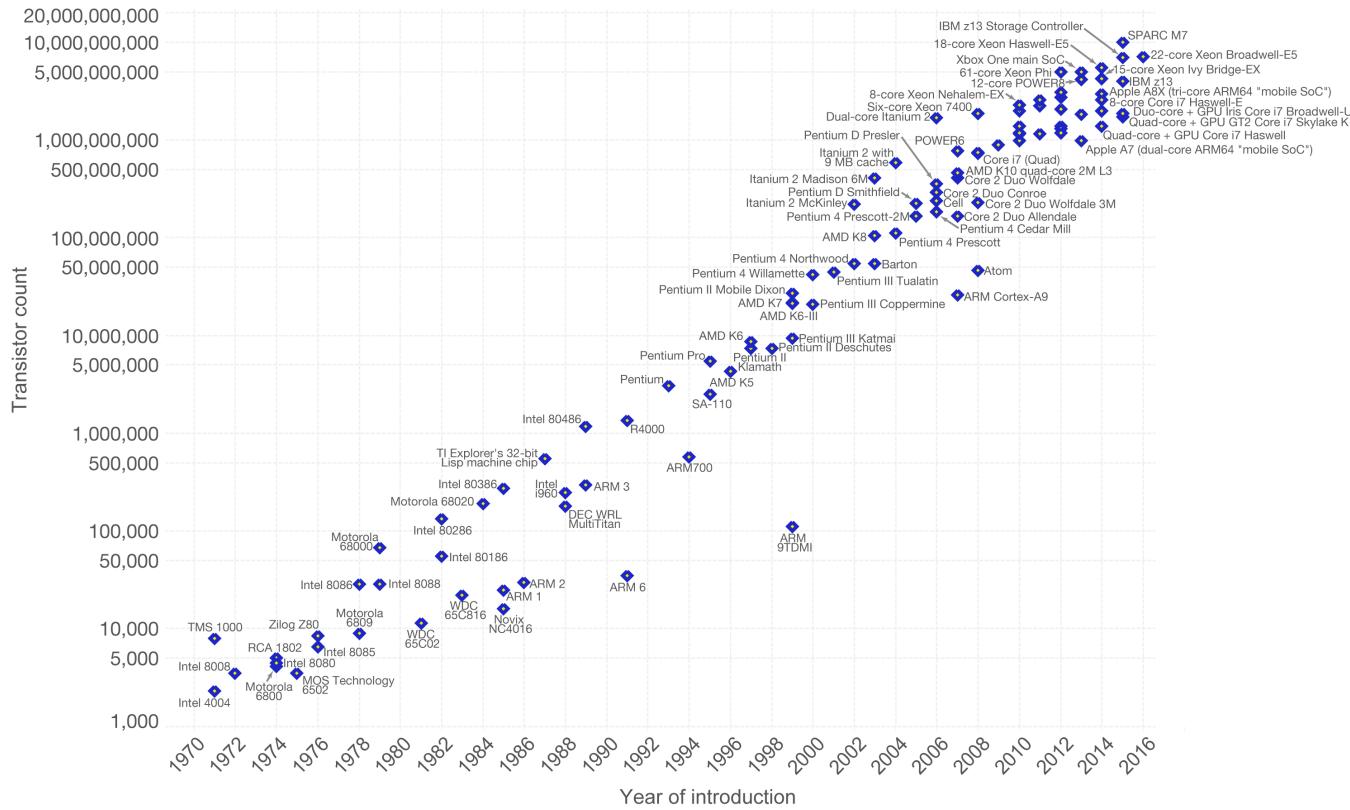
Moore's law predicts a 60-percent annual increase in the number of transistors that can be put on a chip.  
The data points given in this figure are memory sizes, in bits.

# Moore's Law

## Moore's Law – The number of transistors on integrated circuit chips (1971-2016)

Our World  
in Data

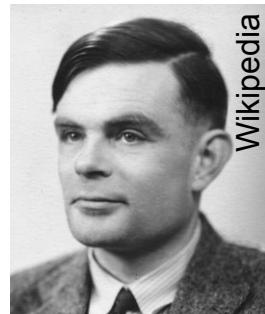
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



# Whose Idea?

Idea of a “stored-program computer”:

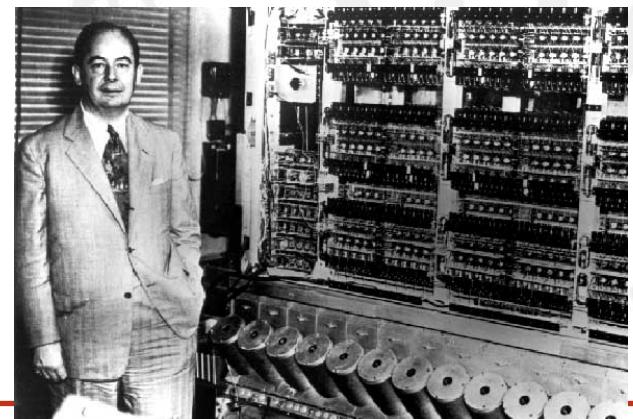
Konrad Zuse 1936 (Z1)



Alan Turing 1936  
(Turing Machine)



John von Neumann 1945 (EDVAC),  
using ideas from  
Eckert and Mauchly 1943



[http://www2.lv.psu.edu/OJJ/courses/ist-240/  
reports/spring2001/fa-cb-bc-kf/1941-1950.html](http://www2.lv.psu.edu/OJJ/courses/ist-240/reports/spring2001/fa-cb-bc-kf/1941-1950.html)

Radboud Universiteit Nijmegen



# Von Neumann machine

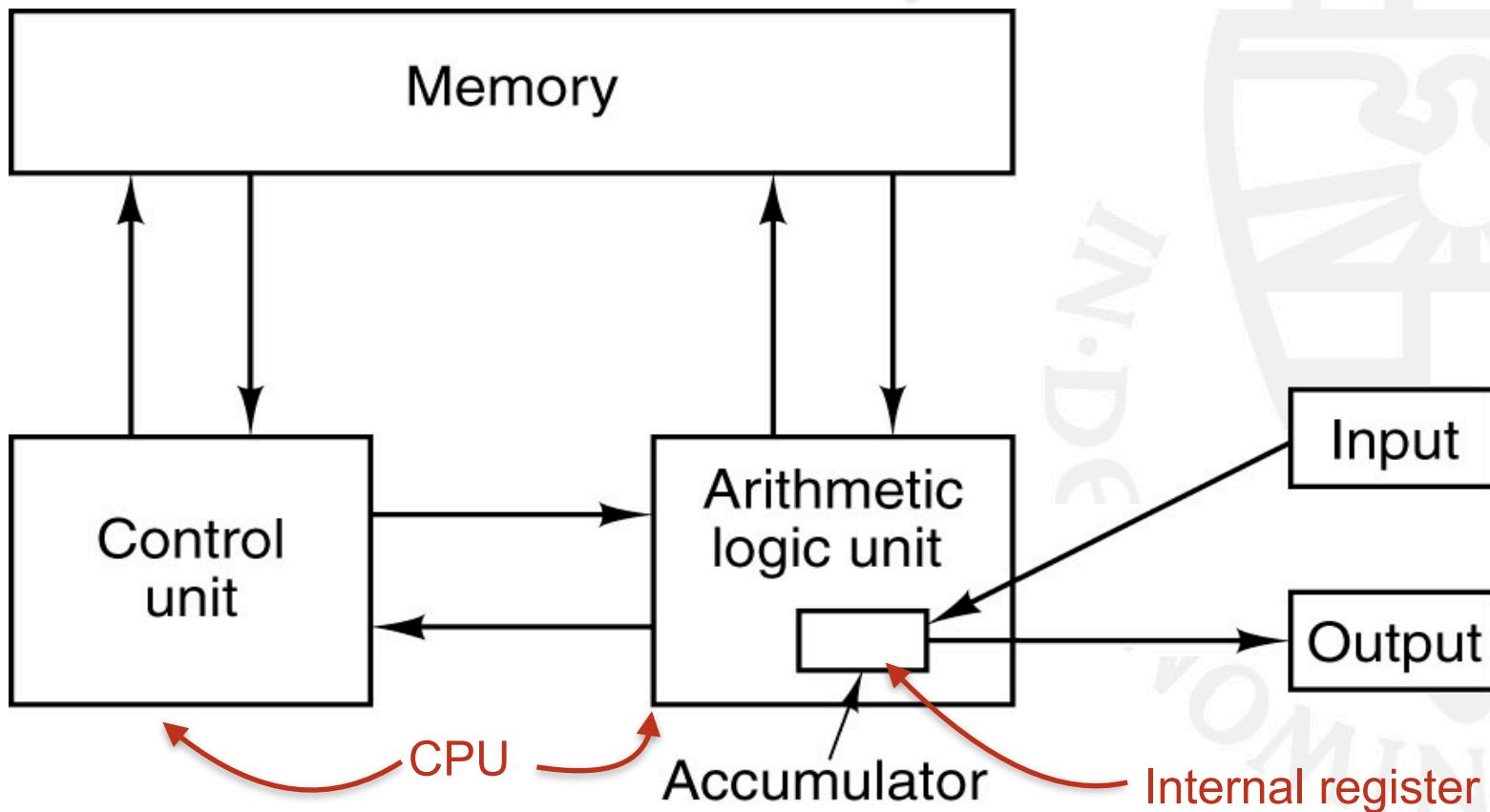
What are the main parts of a computer?



- control unit
- arithmetic logic unit
- memory
- input and output equipment

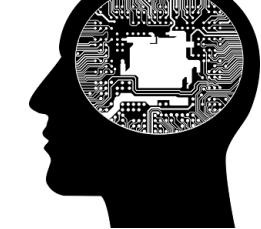
## Von Neumann-architecture

4096 words, 40 bits per word, words may be instructions or (signed) integers

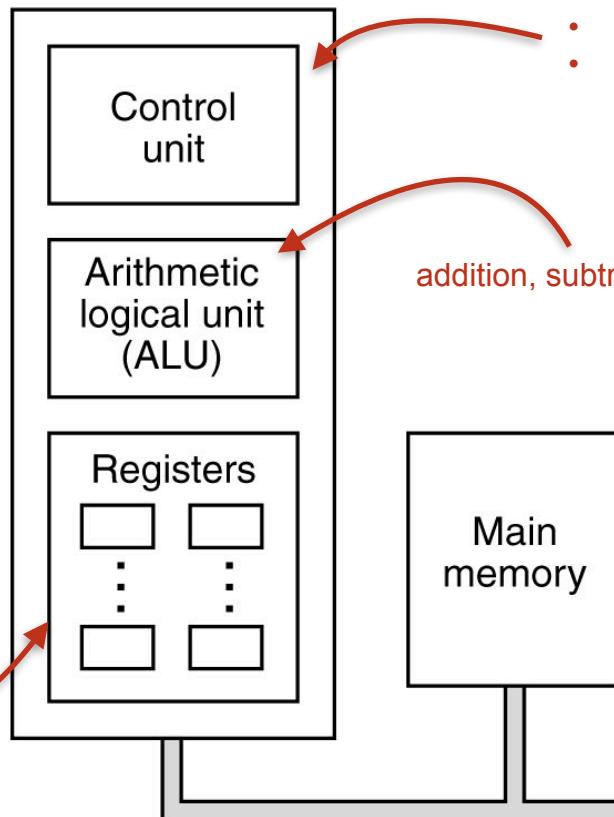


Tanenbaum, Structured Computer Organization, Fifth Edition, © 2006 Pearson Education, Inc. All rights reserved. 0-13-148521-0

# Von Neumann-architecture



Central processing unit (CPU)



- fetch program instructions from main memory
- execute programs

addition, subtraction, other simple operations needed to carry out instructions

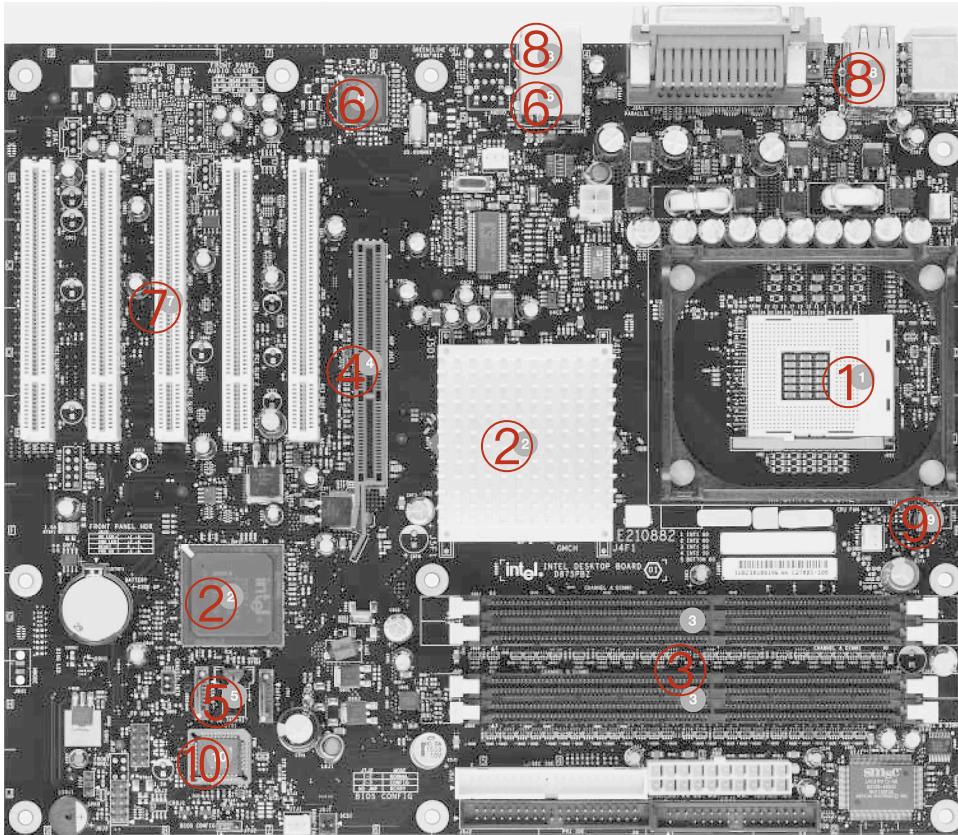
I/O devices

parallel wires,  
modern computers  
have several buses

Bus

Tanenbaum, Structured Computer Organization, Fifth Edition, © 2006 Pearson Education, Inc. All rights reserved. 0-13-148521-0

# Personal Computer



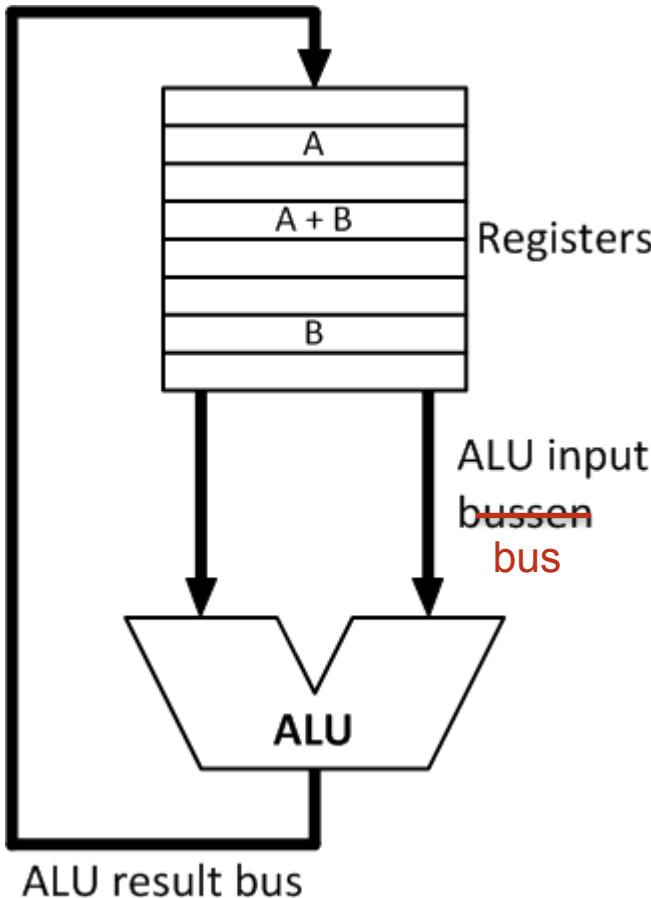
- ① Pentium 4 socket
- ② 875P Support chip
- ③ Memory sockets
- ④ AGP connector
- ⑤ Disk interface
- ⑥ Gigabit Ethernet
- ⑦ Five PCI slots
- ⑧ USB 2.0 ports
- ⑨ Cooling technology
- ⑩ BIOS

This figure is a photograph of the Intel D875PBZ board.

The photograph is copyrighted by the Intel Corporation, 2003 and is used by permission.

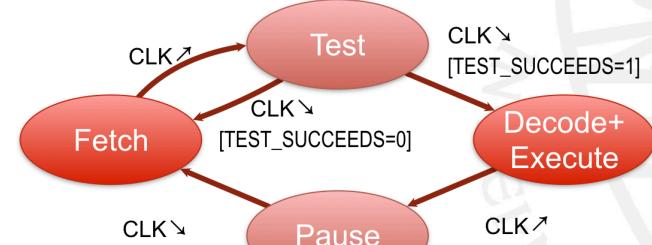
Tanenbaum, Structured Computer Organization, Fifth Edition, © 2006 Pearson Education, Inc. All rights reserved. 0-13-148521-0

# CPU Organization: ALU



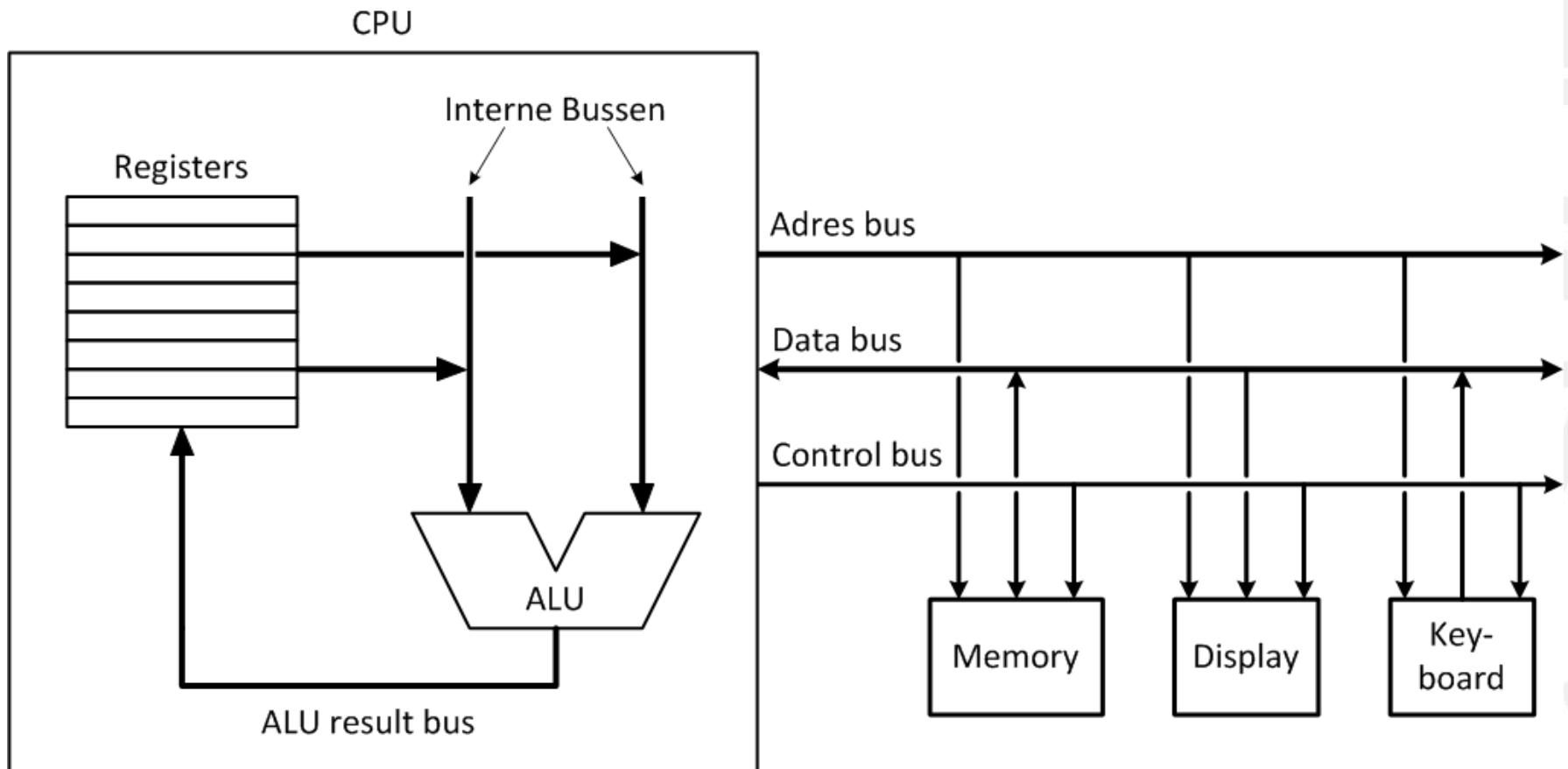
## Data path cycle

- Registers feed into ALU input registers (A and B)
- ALU performs addition, subtraction, and other simple operations
- Output is stored in register

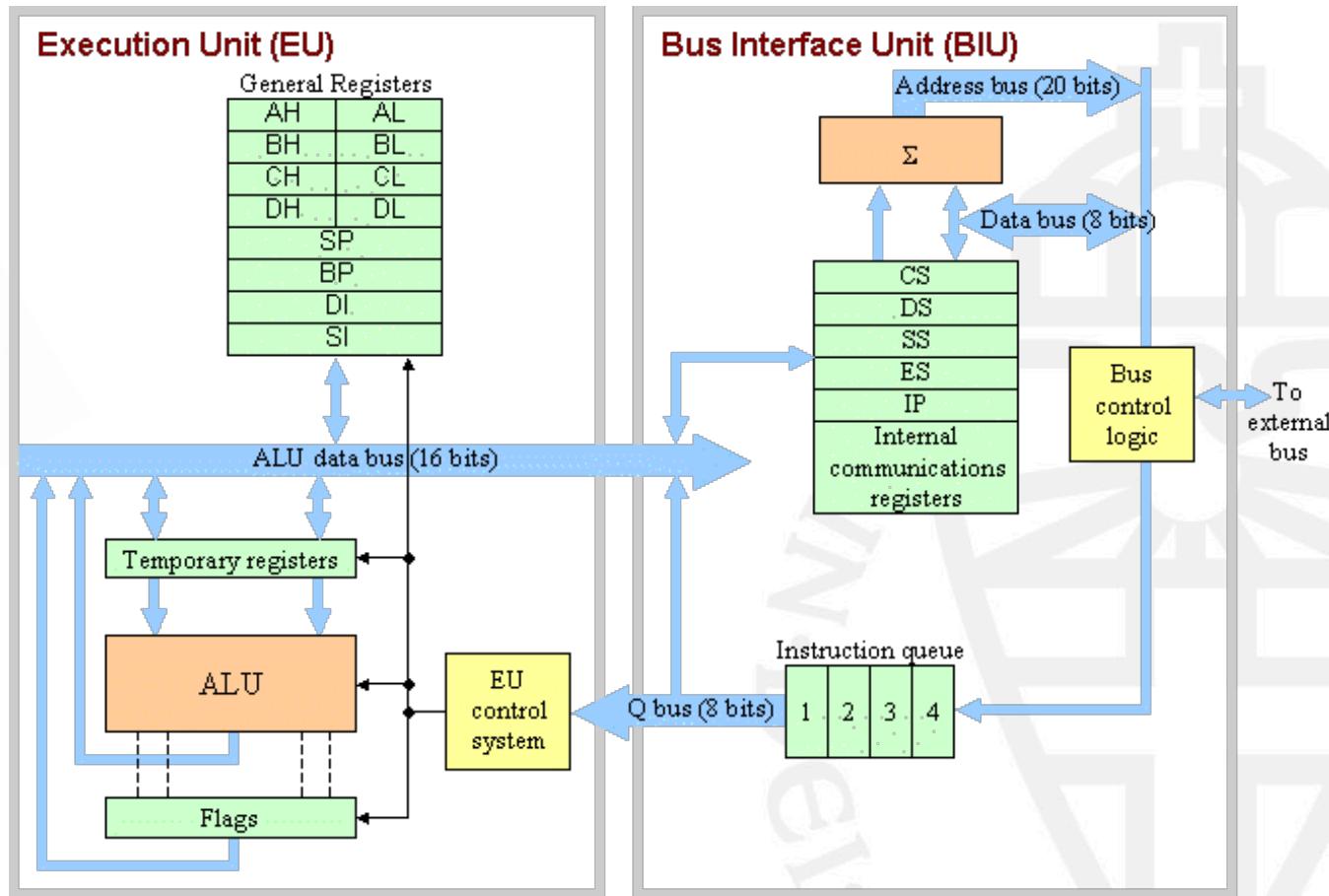


The **data path** of a typical Von Neumann machine.

# Internal and External Buses



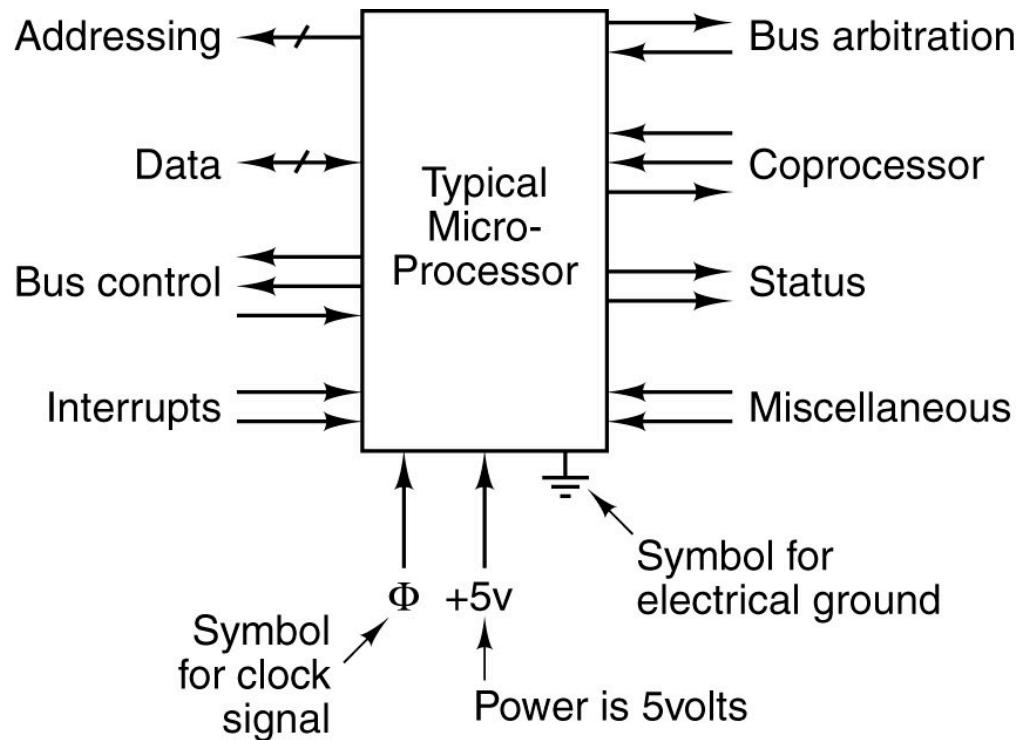
# Organization of the 8088-CPU



- Static registers (groups of D Flip-Flops) used to hold or transfer binary data
- Logic gate circuits designed to perform arithmetic or logical functions
- Logic gate circuits designed to provide internal control to processor
- Internal data busses used to pass information between components

<http://faculty.etsu.edu/tarnoff/ntes2150/uproc/blck8088.gif>

# Connection of a CPU Chip

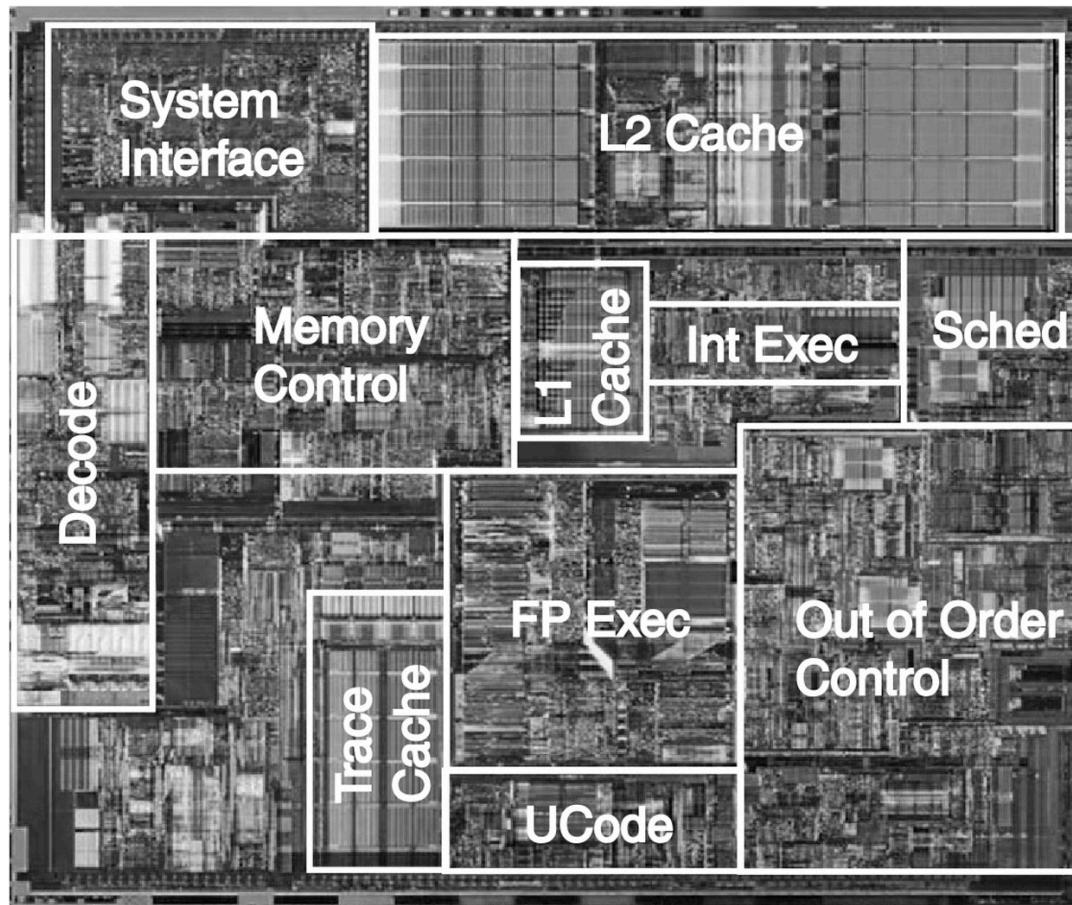


The logical pinout of a generic CPU.

The arrows indicate input signals and output signals.

The short diagonal lines indicate that multiple pins are used.

# Intel Computer Family (2)

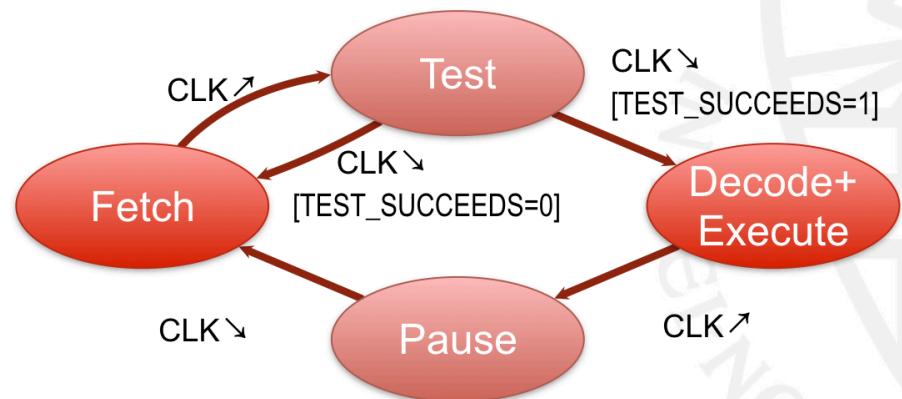


The Pentium 4 chip. The photograph is copyrighted by the Intel Corporation, 2003 and is used by permission.

# CPU Behavior

program execution

1. read instruction (address: program counter) (fetch)
2. increase program counter
3. decode instruction (decode)
4. execute instruction



# Summary

What do we expect from you:

- do your homework
- do your practical assignment
- do not cheat
- take part in the exam
- actively participate in the lectures and werkcolleges

Discussed today:

- Von-Neumann architecture
- Structure of CPU