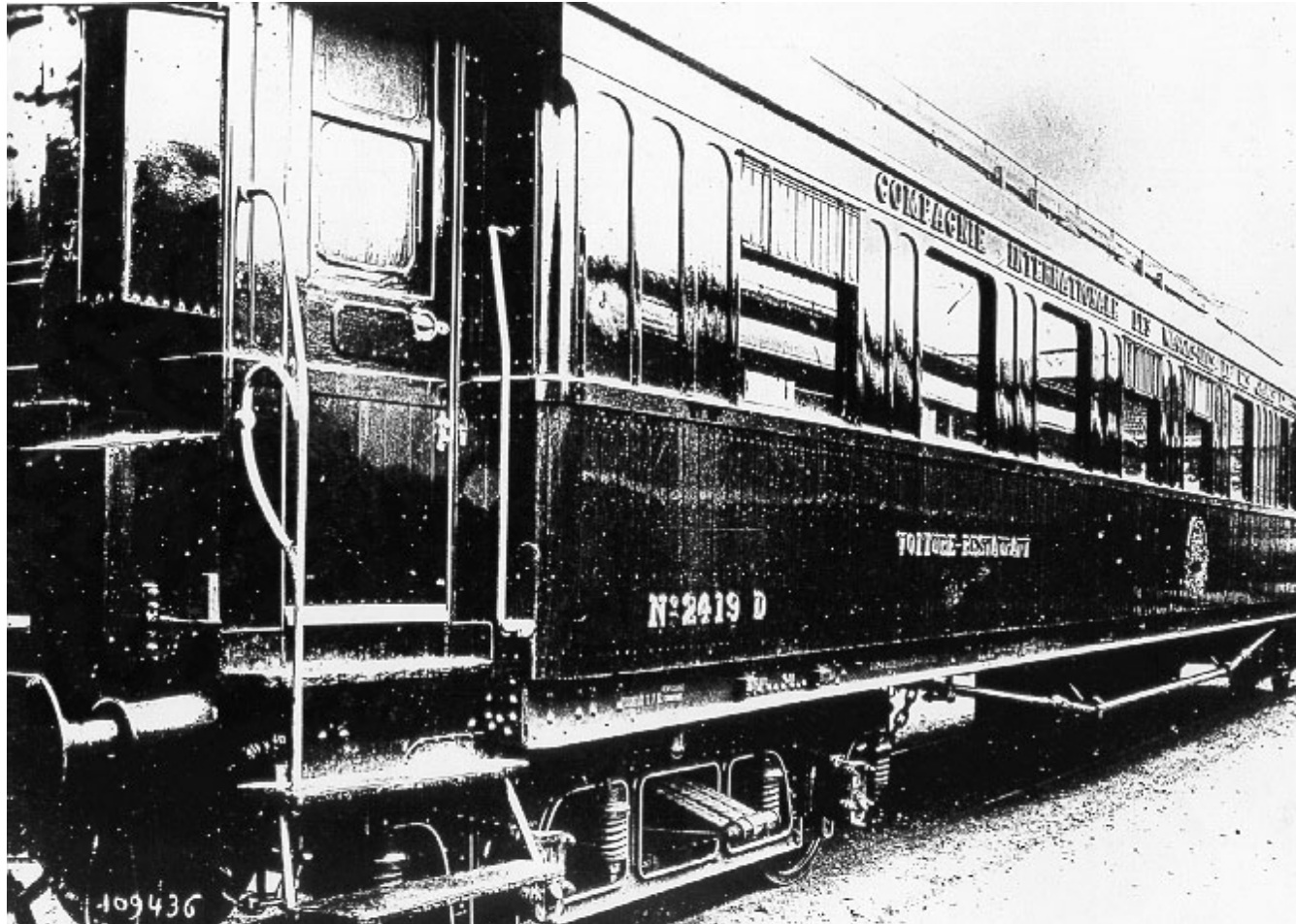


ICT Infrastructuuren

11 november 2013

David N. Jansen

11 november 1918: einde van Wereldoorlog 1



commons.wikimedia.org/wiki/File:Compiegnevagnen.jpg

11 november 1918: einde van Wereldoorlog 1

De HEER zal recht spreken tussen machtige volken,
over grote en verre naties een oordeel vellen.

Dan zullen zij hun zwaarden tot ploegijzers omsmeden
en hun speren tot snoeimessen.

Geen volk zal nog het zwaard trekken tegen een ander volk,
geen mens zal meer weten wat oorlog is.

Micha 4:3

Leerdoelen van ICTI: kennis

- Opbouw en functiewijze van computer
- Basisbouwstenen van hardware
- Functies van besturingssysteem
 - nadruk op: processen
- dit is achtergrondkennis
 - doel: realistische inschatting van mogelijkheden

Leerdoelen van ICTI: vaardigheden

- een eenvoudig assembly-programma schrijven (ca. 20 instructies)
- formele analyse van processen (eenvoudige procesalgebra)

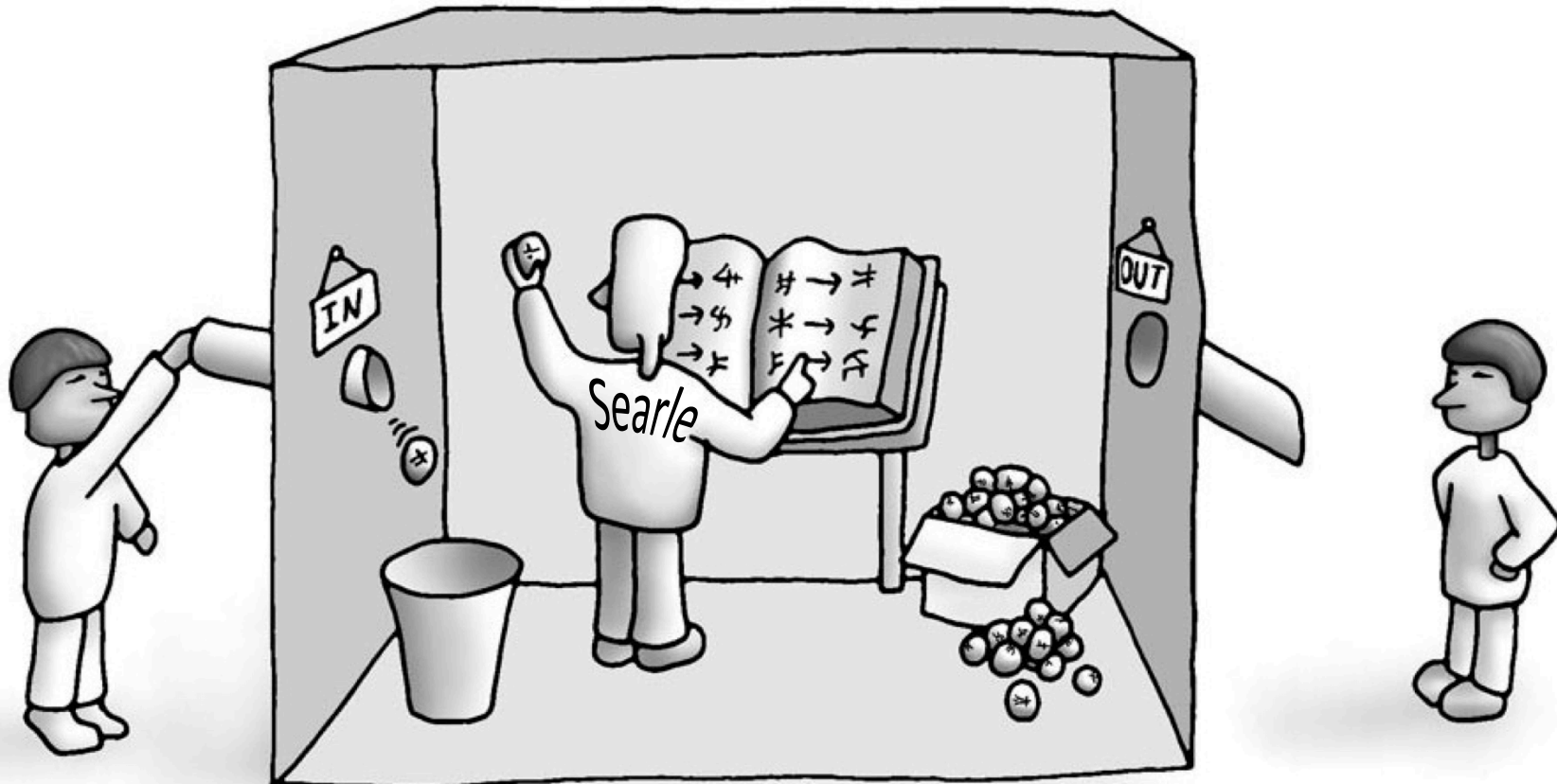
Het college in één zin

**Computers manipuleren
symbolen formeel.**

Computers manipuleren symbolen formeel.

- Symbolen: Cijfers, letters e.d.
 - eigenlijk alleen bitpatronen
 - andere symbolen worden vertaald naar bitpatronen
- **Formeel: De computer let alleen op de vorm en trekt zich niets aan van de betekenis.**
- Manipuleren:
Berekenen, verschuiven, presenteren etc.

Chinese kamer: een formeel systeem



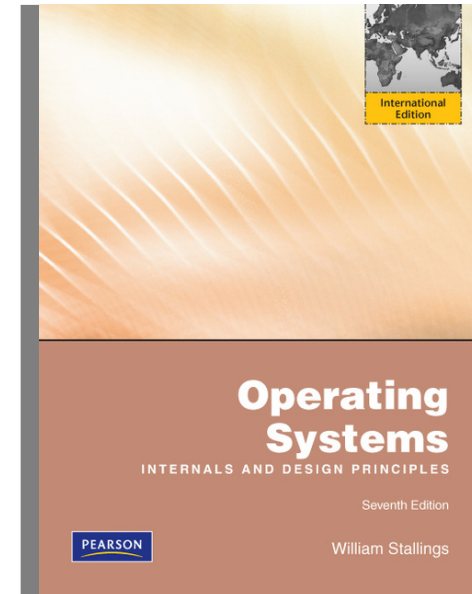
afbeelding: berto-meister.blogspot.nl/2011/12/chinese-room-thought-experiment.html

Searle, John R.: **Minds, brains, and programs.**

Behavioral and Brain Sciences, 3(3)1980, pp. 417–424

Literatuur

- **Operating systems: internals and design principles.** / *William Stallings.* – 7th ed. – Upper Saddle River, NJ: Pearson, 2011.



- **Computer networks** / *Andrew S. Tanenbaum.* – Fourth ed. – Upper Saddle River, NJ: Prentice Hall PTR, 2003.
- **Concurrency: state models & Java programs** / *Jeff Magee; Jeff Kramer.* – Chichester: Wiley, 1999.

Praktische opdrachten

- Twee praktische huiswerkopdrachten
 - schrijf een eenvoudig machinetaal-programma
 - beschrijf een aantal processen in FSP
- individueel of groepswork
- Let op de deadlines en begin vroeg genoeg!
(wanneer deadlines?)

Beoordeling

- Praktische opdrachten: geen cijfer, wel toelatingsvoorwaarde
- schriftelijk tentamen
 - je mag jouw uitwerking van de opdrachten gebruiken (evtl. ontdaan van overdadig commentaar)

Werkhouding

- 2010: 5 van de 15 ICTI-deelnemers leverden kopie van oude uitwerking in
 - tot in details van formattering gelijk
 - houdt vol dat hij zelf heeft nagedacht
 - lijkt wel politiek: fouten verdoezelen ipv erkennen
- Later zijn klanten niet meer zo aardig
 - als je fraudeert
krijg je nooit meer een vervolgoopdracht

Machines en Machinetaal

ICT Infrastructuren

11 november 2013

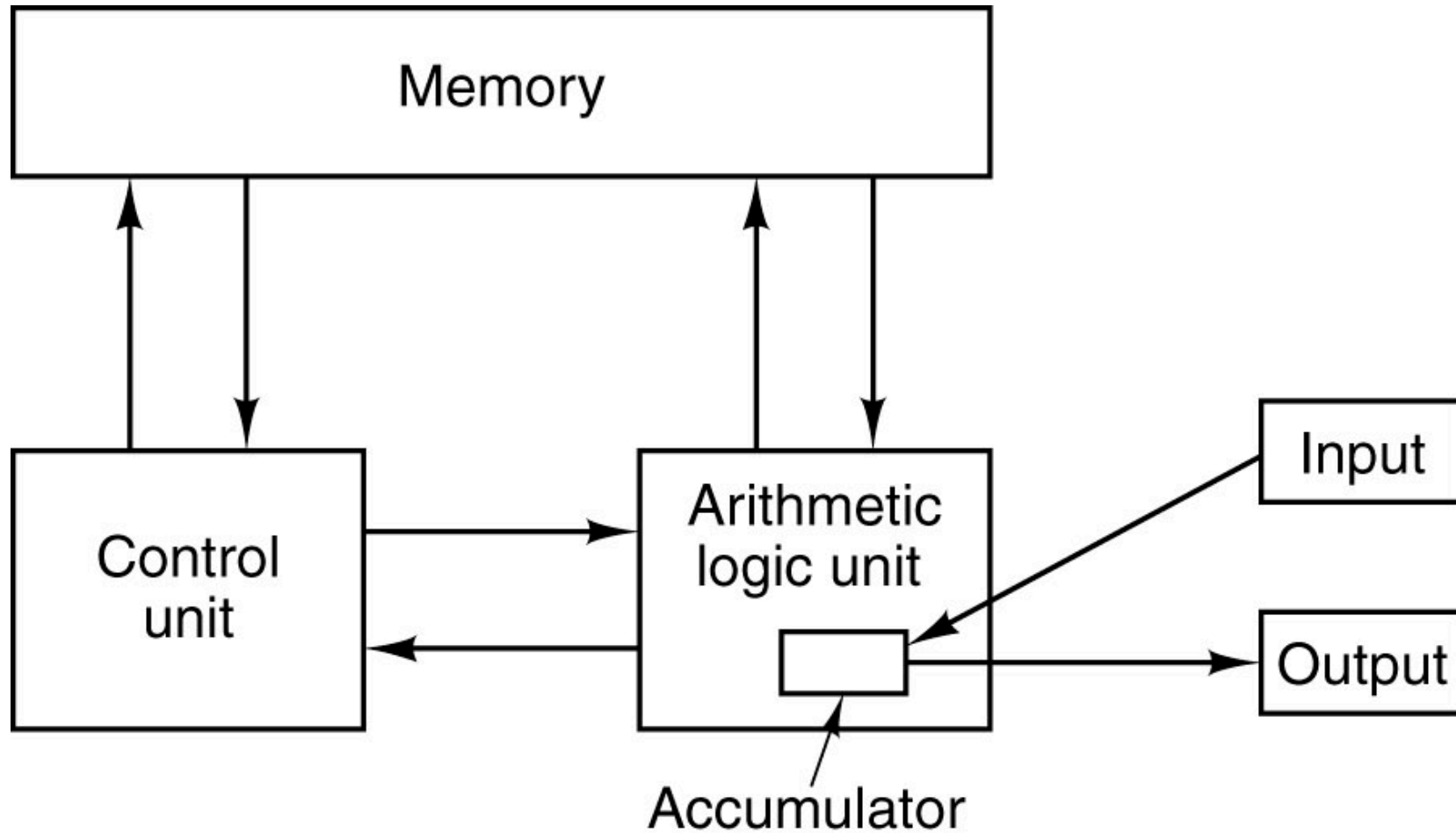
Leerdoel voor vandaag

- Belangrijkste onderdelen van computer-hardware benoemen en beschrijven
- dynamisch gedrag van CPU beschrijven
- control structures in assembly
(eenvoudige programma's schrijven)

Hoofdbestanddelen van een computer

- Welke bestanddelen heeft een computer?
 - besturingseenheid
 - rekeneenheid
 - geheugen (voor programma èn gegevens)
 - in- en uitvoerapparaten
- Dit heet von Neumann-architectuur.

von-Neumann-architectuur .



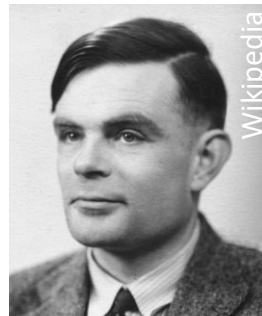
Wiens idee beklift?

- ideeën van een “stored program computer”:

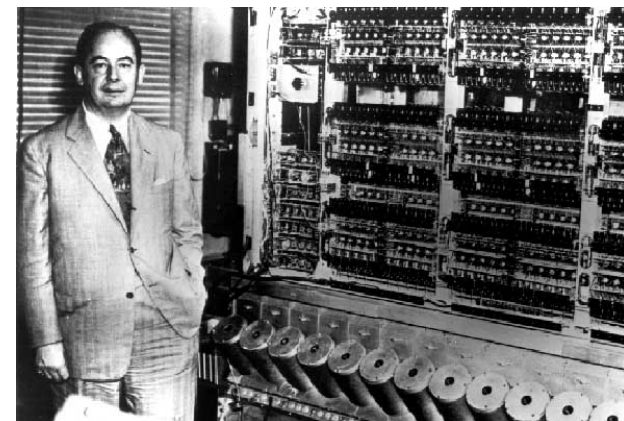
- Konrad Zuse 1936



- Alan Turing 1936



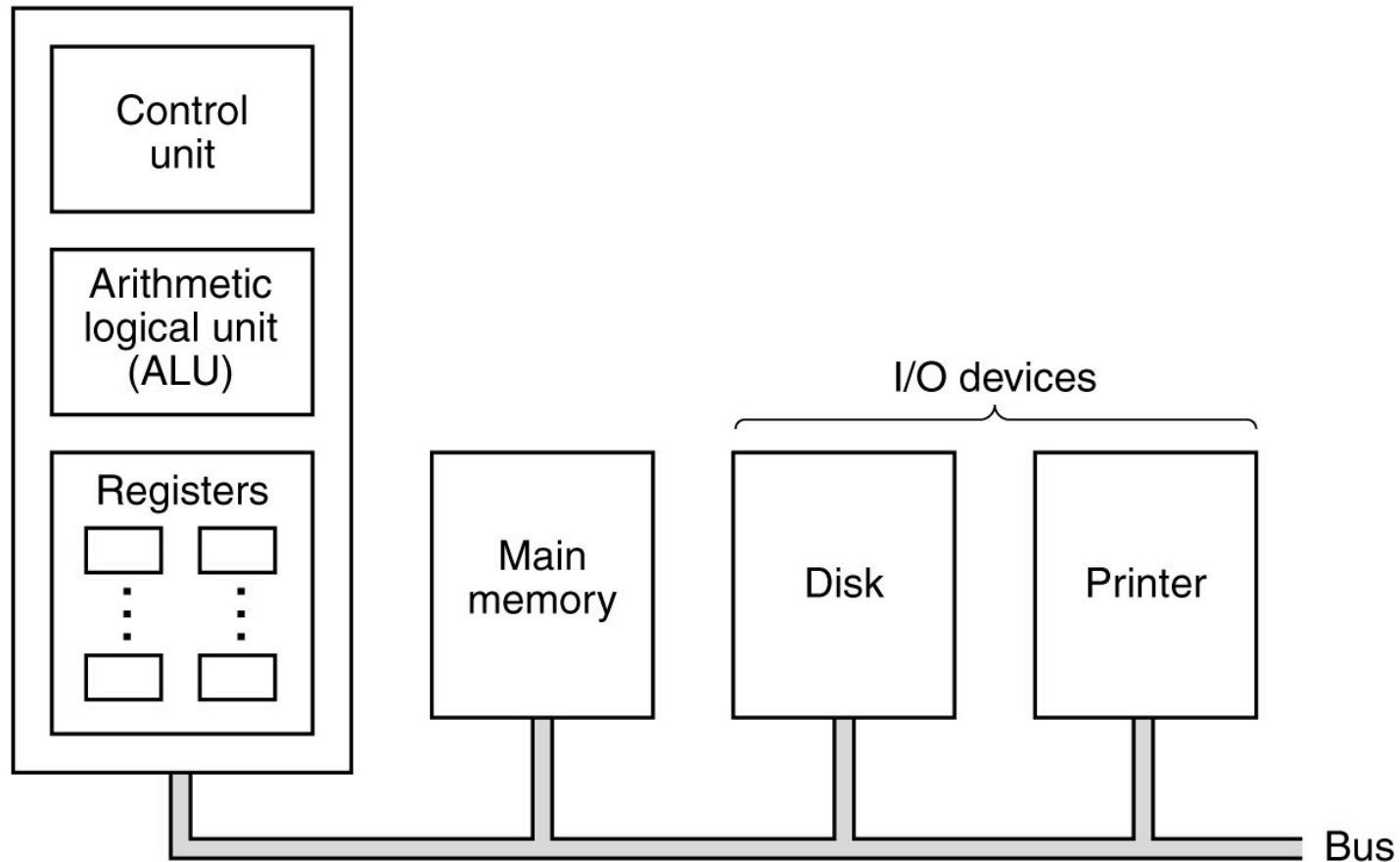
- John von Neumann 1945,
n.a.v. ideeën van
Eckert en Mauchly 1943



<http://www2.lv.psu.edu/OJJ/courses/ist-240/reports/spring2001/fa-cb-bc-kf/1941-1950.html>

„von Neumann“-architectuur .

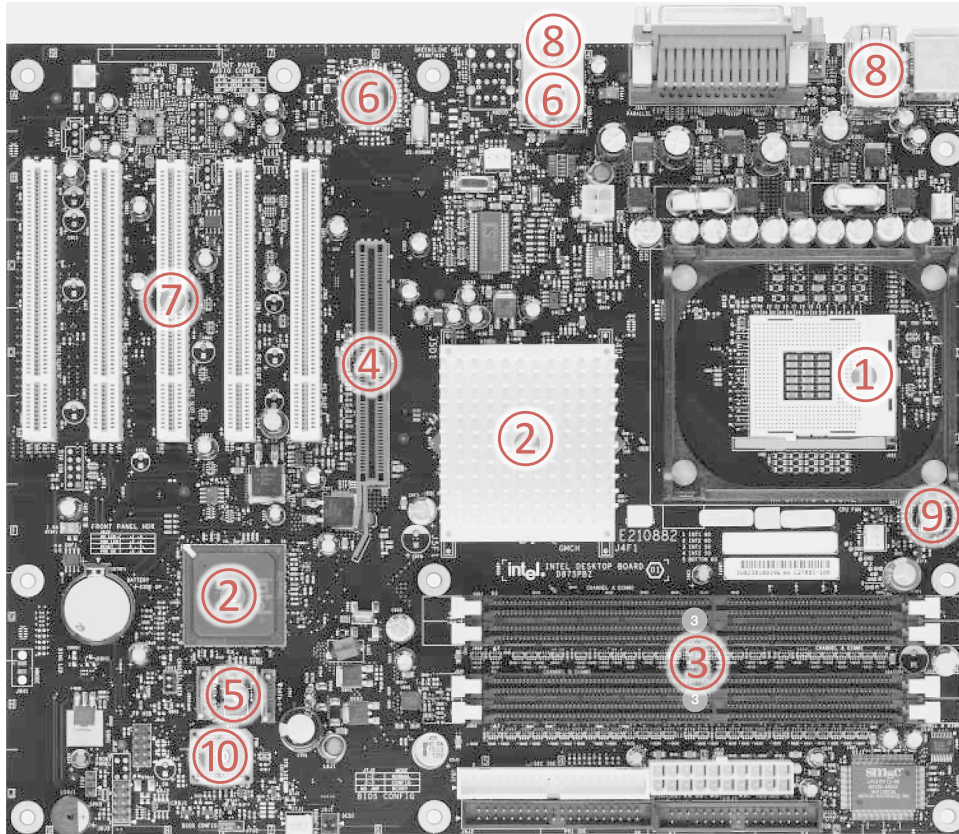
Central processing unit (CPU)



Doelen van de onderdelen

- CPU: besturen èn berekenen
- hoofdgeheugen:
programma en gegevens opslaan
- in-/uitvoer: externe communicatie
- bus: interne communicatie

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

A printed circuit board is at the heart of every personal computer. This figure is a photograph of the Intel D875PBZ board. The photograph is copyrighted by the Intel Corporation, 2003 and is used by permission.

Structuur van het RAM

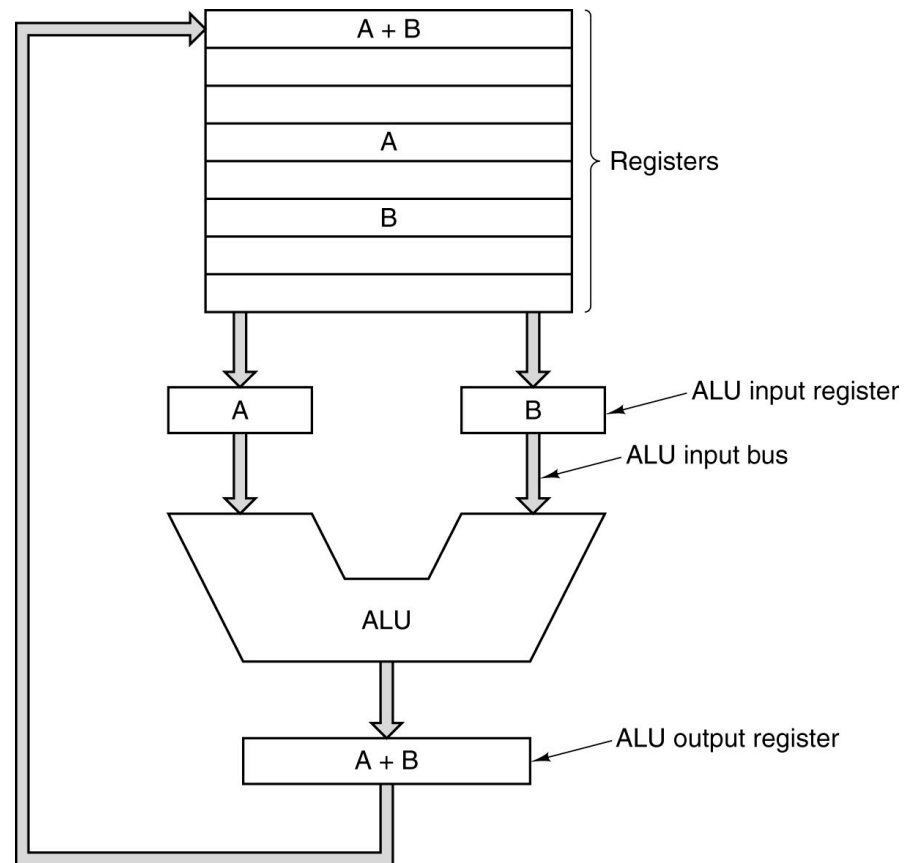
- een soort grote tabel van gegevens
- data is geordend naar adressen
- historisch gegroeid: één adres per byte
- nu vaak: RAM voor 32- of 64-bit data

- lezen en schrijven: altijd op een bepaald adres

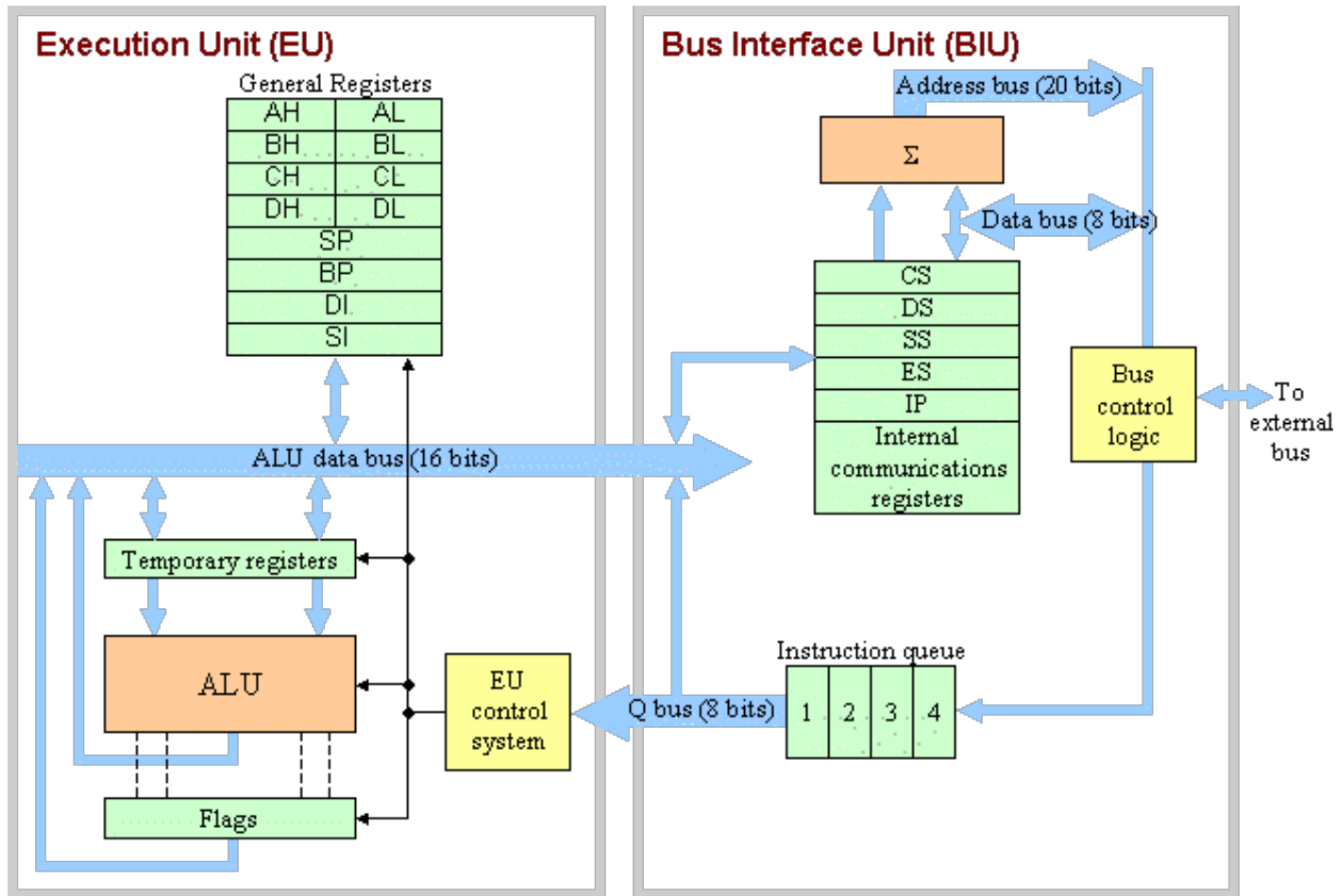
Structuur van de CPU

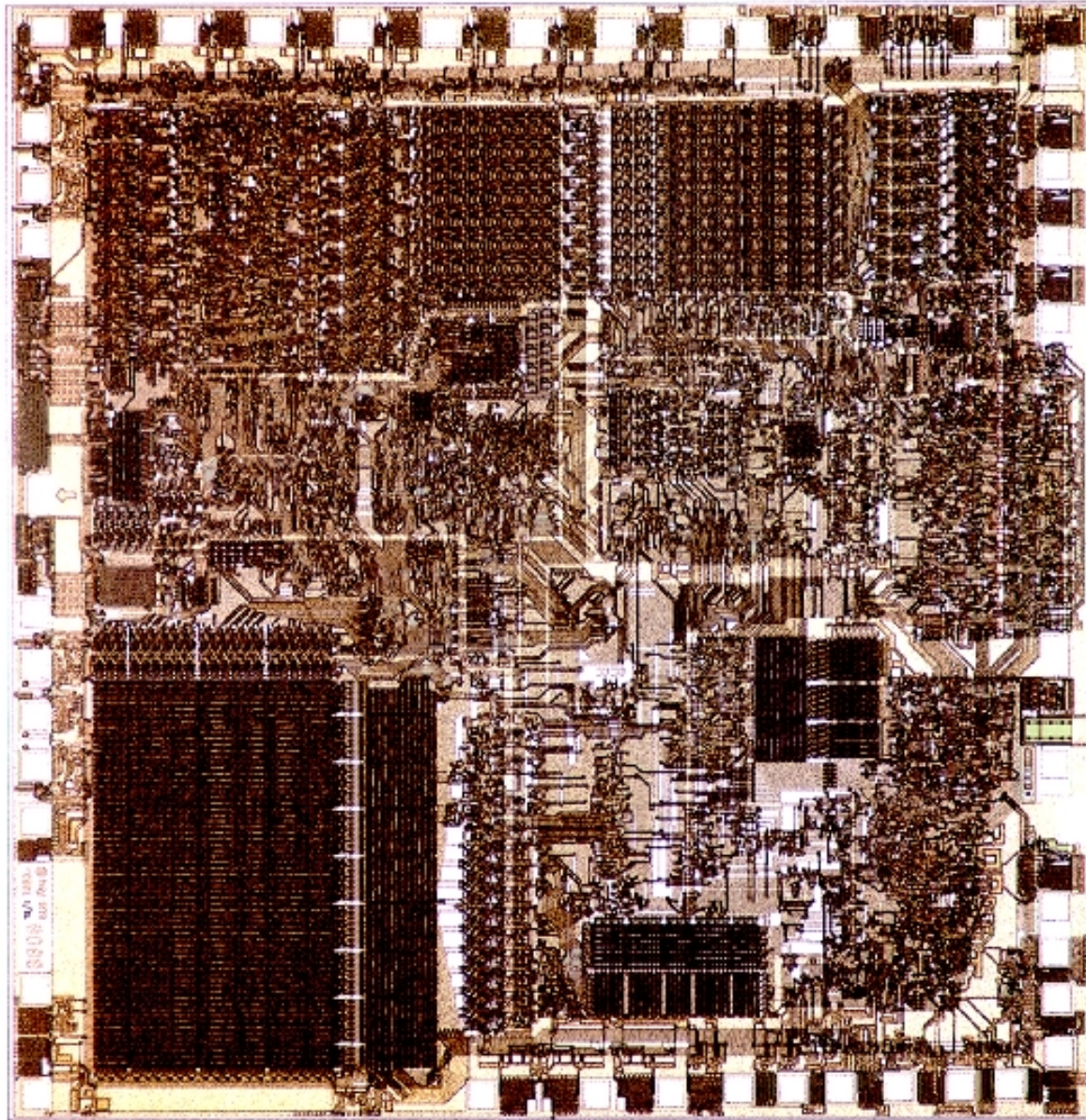
- registers (kleine geheugenstukjes)
 - gegevens- en adresregisters
(voor tussenresultaten van berekeningen)
 - vlaggen (om voorwaarden te testen)
 - instructie/programma-teller
(geeft aan waar het programma staat)
 - interne registers
- arithmetisch-logische eenheid = ALU
- besturingseenheid = control logic

CPU Organization



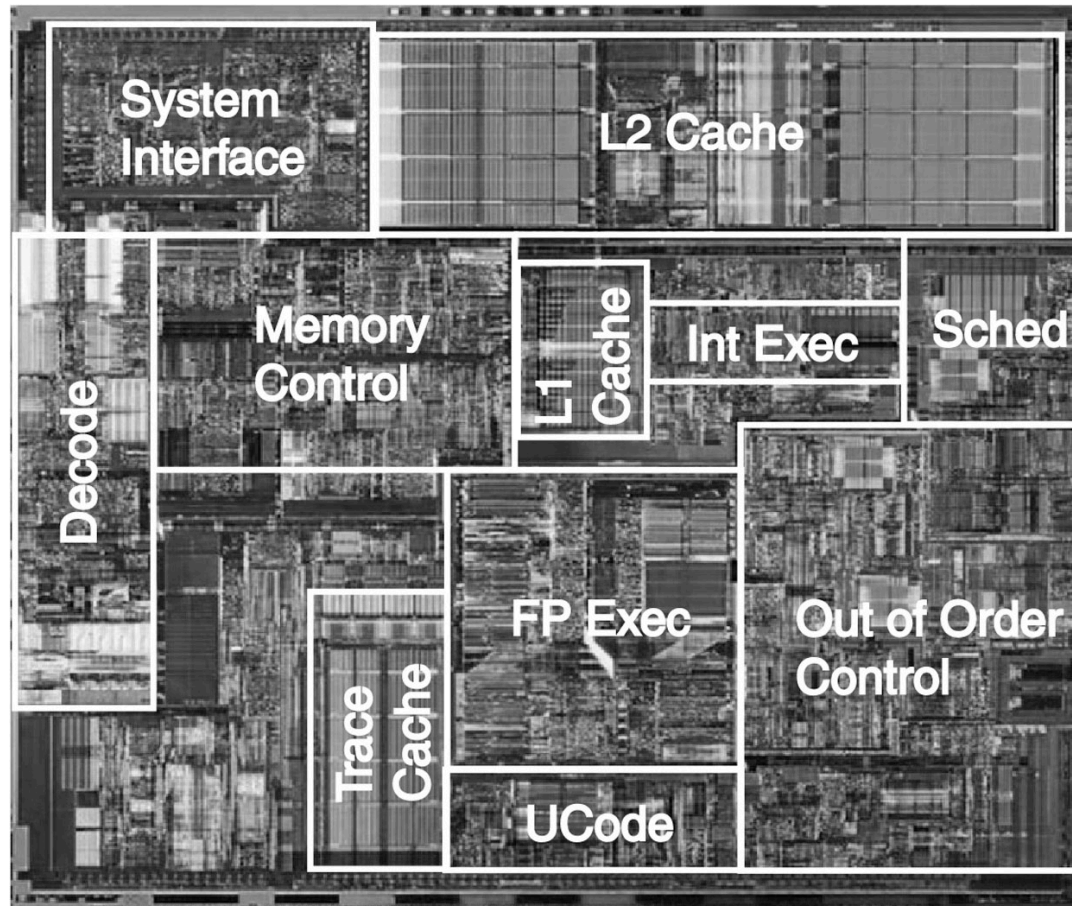
The data path of a typical Von Neumann machine.





http://www.icknowledge.com/trends/8086_8088B1.jpg

Intel Computer Family (2)



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

Gedrag van de CPU

- programma executeren
 - ① instructie lezen (adres in programmateller)
 - ② programmateller ophogen
 - ③ instructie decoderen
 - ④ instructie uitvoeren
- in huidige processoren: parallel = pipelined

Programmeertalen

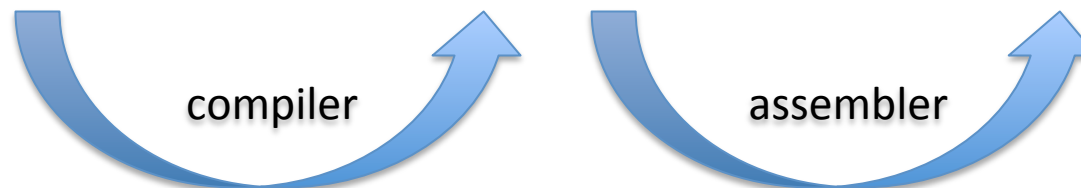
- Machinetaal
- Assembly
- Hogere programmeertaal



gemakkelijk te onthouden

Vertaalstappen

hogere taal	assembly	machinetaal
k = (*i)[2] + 3;	MOV EBX, [i] MOV EAX, [EBX+8] ADD EAX, 3 MOV [k], EAX	891E0C000000 894708 0503000000 8B0610000000



- Compiler:
hogere taal → assembly of
hogere taal → machinetaal

Wat doet een assembler?

- mnemonische namen vertalen
- adressen berekenen
- (vaak) macros
- pseudoinstructies



vaste afkortingen van
een paar instructies



constanten opslaan,
macrodefinities

Formaat van assembly-instructies

- per regel één instructie of pseudoinstructie
 - (label)
 - (pseudo)instructienaam
 - operanden
 - (commentaar)

Labels

- namen voor adressen in code of data
 - code: waarheen de processor kan springen

```
JMP      error
```

```
...
```

```
...
```

```
error:  ...
```

- data: globale variabelen

```
MOV      [length], ECX
```

```
...
```

```
...
```

```
length: .data      177
```


Pseudoinstructies

- regieaanwijzingen aan de assembler
 - waar moet het programma opgeslagen worden?
`.org 1024`
 - datadefinitie, b.v.
`hellostring: .data "Hello, world!", 0`
 - macrodefinities
- lijken vaak erg op instructies

Soorten instructies

- gegevenstransport
- berekenen
- programmaflow

Gegevenstransport

- `A10D000000 MOV EAX, 13`
zet de inhoud van register EAX op 13
- `87F3 XCHG EBX, ESI`
verwissel de inhoud van registers EBX en ESI

Berekenen

- bereken en schrijf het resultaat in register
- vlaggen (zero, sign, carry, overflow) zetten: afhankelijk van resultaat

- **00E3** **ADD** **BL, AH**
tel AH bij BL op
- **F6E1** **MUL** **CL**
vermenigvuldig AL met CL en sla het resultaat in EAX op
- **09D2** **OR** **EDX, EDX**
bereken bit-or van EDX met zichzelf, sla resultaat in EDX op

accumulator =
impliciet register

EDX blijft gelijk,
vlaggen veranderen

Programmaflow

- normaal:
instructies staan op volgorde in het geheugen
- programmaflow-instructies geven afwijkende volgorde aan

normaal in assembly:
programmeur geeft een label op,
assembler berekent het adres.

- `E9FE0F` `JMP` `label2`
spring naar label2 (tel 0FFE bij de instructiepointer op)
- `75EC` `JNZ` `label3`
als de ZERO-vlag 0 is, spring naar label3 (tel FFEC op)

Voorwaarden testen

- Processor zet vlaggen om aan te geven of laatste resultaat >0 , $=0$ of <0 was
- Sommige instructies testen vlaggen, b.v.
JZ = jump if zero
= verander programmateller
als laatste resultaat 0 was

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	ADD Eb, Gb	ADD Ev, Gv	ADD Gb, Eb	ADD Gv, Ev	ADD AL, b	ADD AX, Iv	PUSH ES	POP ES	OR Eb, Gb	OR Ev, Gv	OR Gb, Eb	OR Gv, Ev	OR AL, Ib	OR AX, Iv	PUSH CS	--
1	ADC Eb, Gb	ADC Ev, Gv	ADC Gb, Eb	ADC Gv, Ev	ADC AL, Ib	ADC AX, Iv	PUSH SS	POP SS	SBB Eb, Gb	SBB Ev, Gv	SBB Gb, Eb	SBB Gv, Ev	SBB AL, Ib	SBB AX, Iv	PUSH DS	POP DS
2	AND Eb, Gb	AND Ev, Gv	AND Gb, Eb	AND Gv, Ev	AND AL, Ib	AND AX, Iv	ES:	DAA	SUB Eb, Gb	SUB Ev, Gv	SUB Gb, Eb	SUB Gv, Ev	SUB AL, Ib	SUB AX, Iv	CS:	DAS
3	XOR Eb, Gb	XOR Ev, Gv	XOR Gb, Eb	XOR Gv, Ev	XOR AL, Ib	XOR AX, Iv	SS:	AAA	CMP Eb, Gb	CMP Ev, Gv	CMP Gb, Eb	CMP Gv, Ev	CMP AL, Ib	CMP AX, Iv	DS:	AAS
4	INC AX	INC CX	INC DX	INC BX	INC SP	INC BP	INC SI	INC DI	DEC AX	DEC CX	DEC DX	DEC BX	DEC SP	DEC BP	DEC SI	DEC DI
5	PUSH AX	PUSH CX	PUSH DX	PUSH BX	PUSH SP	PUSH BP	PUSH SI	PUSH DI	POP AX	POP CX	POP DX	POP BX	POP SP	POP BP	POP SI	POP DI
6	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
7	JO Jb	JNO Jb	JB Jb	JNB Jb	JZ Jb	JNZ Jb	JBE Jb	JA Jb	JS Jb	JNS Jb	JPE Jb	JPO Jb	JL Jb	JGE Jb	JLE Jb	JG Jb
8	GRP1 Eb, Ib	GRP1 Ev, Iv	GRP1 Eb, Ib	GRP1 Ev, Ib	TEST Gb, Eb	TEST Gv, Ev	XCHG Gb, Eb	XCHG Gv, Ev	MOV Eb, Gb	MOV Ev, Gv	MOV Gb, Eb	MOV Gv, Ev	MOV Ew, Sw	LEA Gv, M	MOV Sw, Ew	POP Ev
9	NOP	XCHG CX, AX	XCHG DX, AX	XCHG BX, AX	XCHG SP, AX	XCHG BP, AX	XCHG SI, AX	XCHG DI, AX	CBW	CWD	CALL Ap	WAIT	PUSHF	POPF	SAHF	LAHF
A	MOV AL, Ob	MOV AX, Ov	MOV Ob, AL	MOV Ov, AX	MOVSb	MOVSw	CMPSb	CMPSw	TEST AL, Ib	TEST AX, Iv	STOSb	STOSw	LODSb	LODSw	SCASb	SCASw
B	MOV AL, Ib	MOV CL, Ib	MOV DL, Ib	MOV BL, Ib	MOV AH, Ib	MOV CH, Ib	MOV DH, Ib	MOV BH, Ib	MOV AX, Iv	MOV CX, Iv	MOV DX, Iv	MOV BX, Iv	MOV SP, Iv	MOV BP, Iv	MOV SI, Iv	MOV DI, Iv
C	--	--	RET Iw	RET	LES Gv, Mp	LDS Gv, Mp	MOV Eb, Ib	MOV Ev, Iv	--	--	RETF Iw	RETF	INT 3	INT Ib	INTO	IRET
D	GRP2 Eb, 1	GRP2 Ev, 1	GRP2 Eb, CL	GRP2 Ev, CL	AAM IO	AAD IO	--	XLAT	--	--	--	--	--	--	--	--
E	LOOPNZ Jb	LOOPZ Jb	LOOP Jb	JCXZ Jb	IN AL, Ib	IN AX, Ib	OUT Ib, AL	OUT Ib, AX	CALL Jv	JMP Jv	JMP Ap	JMP Jb	IN AL, DX	IN AX, DX	OUT DX, AL	OUT DX, AX
F	LOCK	--	REP NZ	REP Z	HLT	CMC	GRP3a Eb	GRP3b Ev	CLC	STC	CLI	STI	CLD	STD	GRP4 Eb	GRP5 Ev

Signed- en unsigned-getallen

- Bitpatronen kunnen zijn:
 - getallen $0 \dots 2^{32}-1$ unsigned
 - getallen $-2^{31} \dots 2^{31}-1$ signed
- afhankelijk daarvan verschillende tests
 - b.v. spring als het resultaat ≤ 0 is (“less or equal”):
 - `JLE` = invoergetallen waren signed
 - `JLEU` = invoergetallen waren unsigned
 - evenzo: `JL/JLU` (“less”), `JG/JGU` (“greater”),
`JGE/JGEU` (“greater or equal”)

if – then – else

Java	simpel Java	assembly (8086)
<pre>if (i == 3) { j = 1; } else { k = 2; }</pre>	<pre>temp = i; temp -= 3; if (temp != 0) goto else; j = 1; goto endif; else: k = 2; endif:</pre>	<pre>MOV EAX, [i] CMP EAX, 3 JNZ else MOV [j], 1 JMP endif else: MOV [k], 2 endif:</pre>

- Assembly/machinetaal kent alleen heel simpele stappen

while

- Probeer een while-loop om te schrijven naar simpel Java.

Java	simpel Java	assembly (8086)
<pre>while (i < 5) { ...; }</pre>	<pre>while: temp = i; temp -= 5; if (temp ≥ 0) goto endwhile; ...; goto while; endwhile:</pre>	<pre>while: MOV EAX,[i] CMP EAX, 5 JGE endwhile ... JMP while endwhile:</pre>

Samenvatting

- Belangrijkste onderdelen van computerhardware benoemen en beschrijven
- dynamisch gedrag van CPU beschrijven
- control structures in assembly
(eenvoudige programma's schrijven)