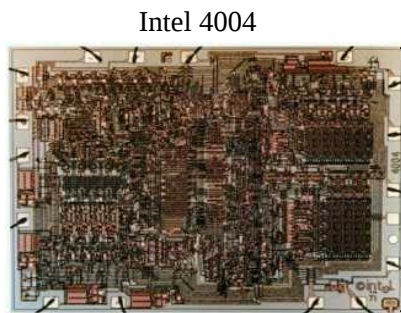
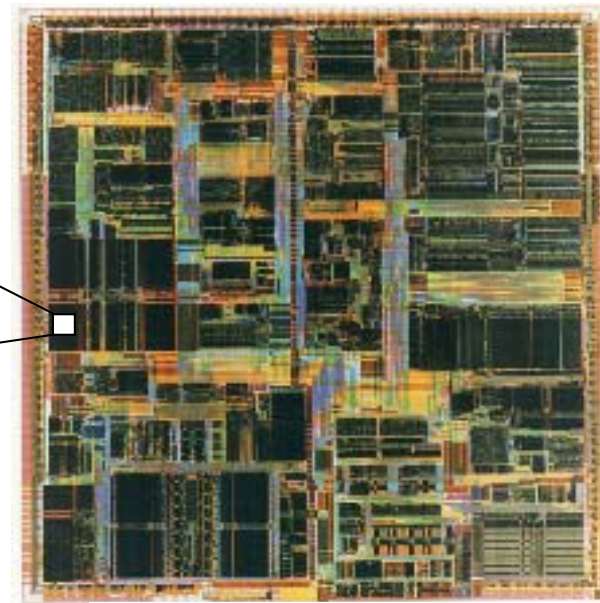


Mikrodatortechnik, 4p/5p



Intel 4004

2300 transistors @ 108 kHz



Intel Pentium

32 million transistors @ 2600MHz

Målsättning med kursen

- **Förstå funktionen hos en processor, samt förstå olika processorarkitekturer**
- **Utifrån en specifikation konstruera ett program i MC68000 assembler**
- **Kunna redogöra funktionen för centrala delar i ett mikrodatorsystem med inriktning på MC68000**
- **Kunna konstruera ett enklare mikrodatorsystem**

Mikrodatorteknik, 4p/5p

– Kurslitteratur

- Rune Körnefors, Mikrodata bit för bit, ISBN 91-44-30862-0, 1996.
- Examination
 - Dugga, inlämningsuppgifter , laborationer
- Föreläsningar (10x2h), Övningar (6x2h), Laborationer (6x4), Flex. mtrl.

– Kursansvarig

- Benny Thörnberg
 - Email: benny.thornberg@miun.se
 - URL: www.itm.miun.se/~bentho

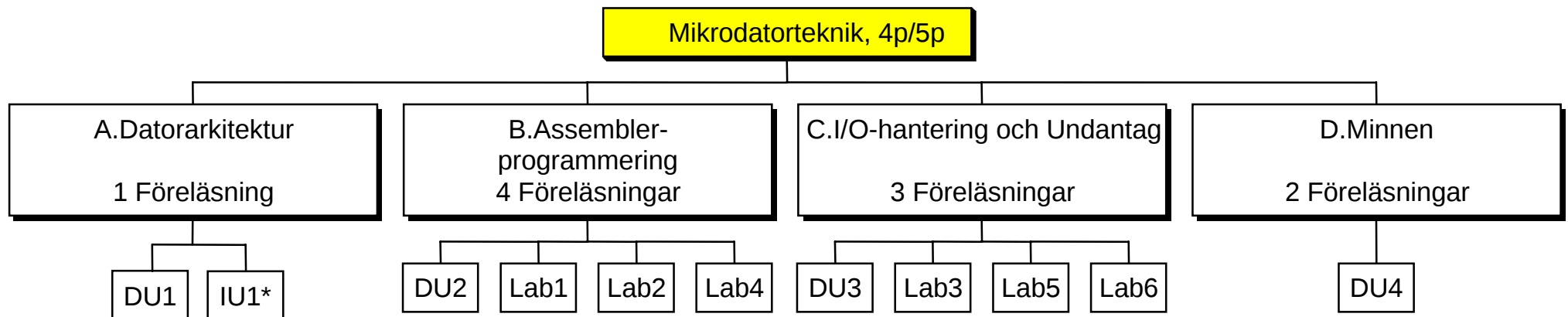
– Labb- och övningslärare

- Benny Thörnberg
 - Email: benny.thornberg@miun.se

– Kurswebb:

- www.itm.miun.se/~elektro/etea20

Organisation



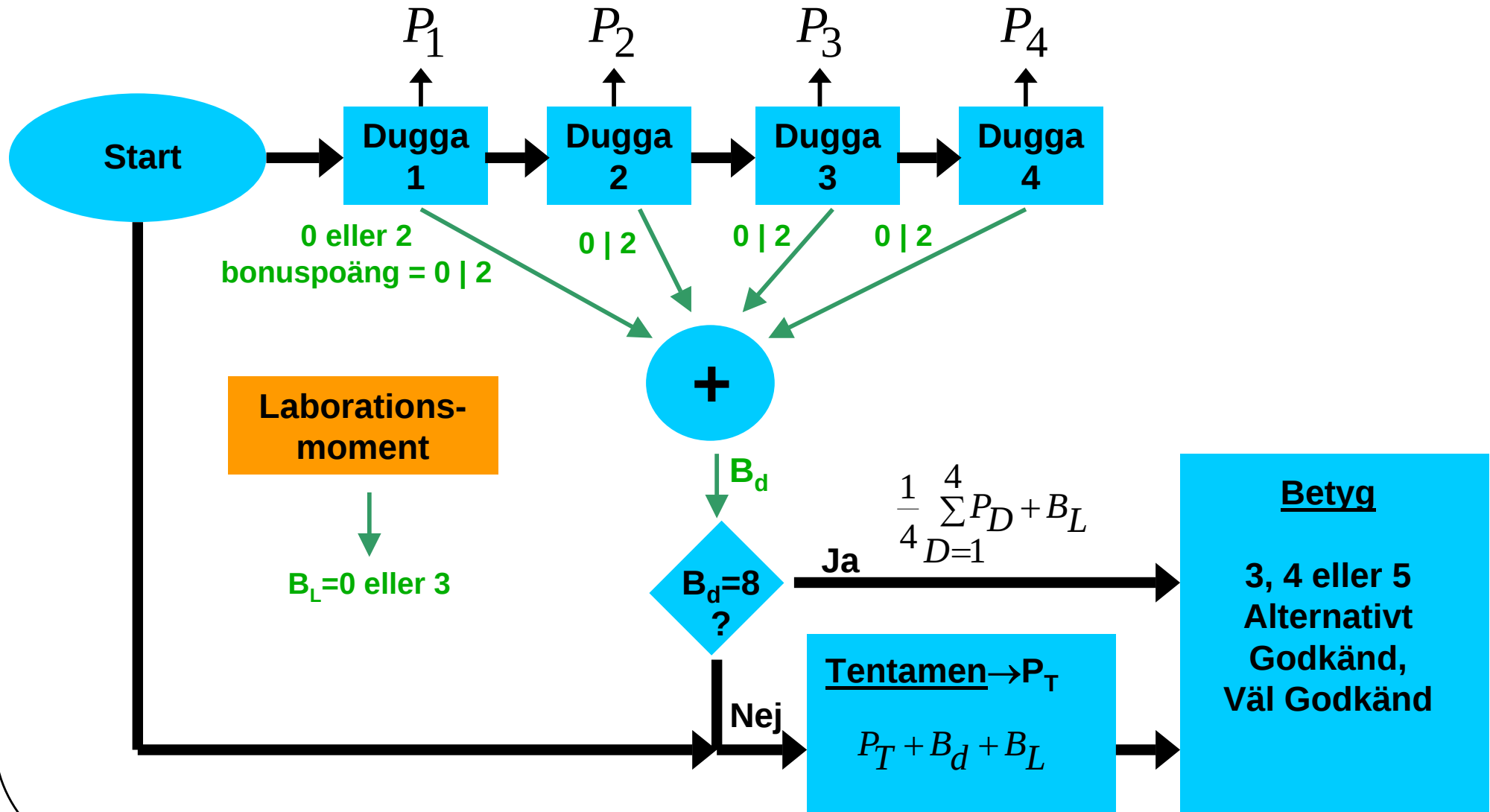
* Gäller endast när kursen ges som 5p

Planering

Vecka	Föreläsning	Övning	Examination	Laboration
45	F1(A)			
46	F2 (B), F3 (B)	Ö1		Lab1
47	F4 (B)	Ö2	DU1	Lab2
48	F5 (B), F6 (C)	Ö3		Lab3
49	F7 (C)		DU2	Lab4
50	F8 (C), F9 (D)	Ö4		Lab5
51	F10 (D)	Ö5	DU3	Lab6
52				
1				
2			DU4	
3			Tentamen	

Examination

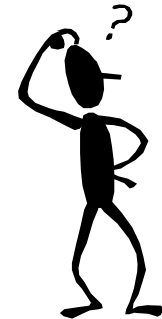
Inlämninguppgift
för 5p-kursen



F1: Introduktion

- **Innehåll:**
 - Motivation av kurs
 - Historik
 - Von Neumann datorn
 - Delarna i ett mikrodatorsystem

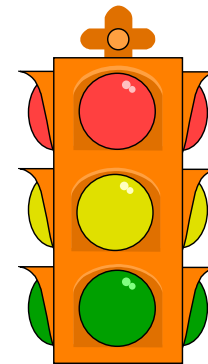
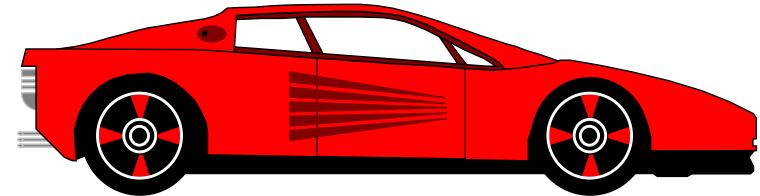
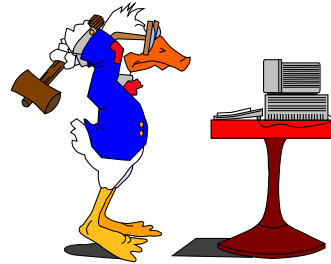
Varför ska vi lära oss mikrodator teknik?



- **Allmänbildning!!!**
 - Vad som skiljer en tekniker och en systemvetare
- **Nödvändig kunskap för att**
 - arbeta med utveckling av mikrodatorsystem.
 - implementera system (telekom, video, etc.), för att kunna optimera måste man förstå arkitekturen.
 - arbete med utveckling av hårdvara
 - Hårdvara används ofta i datorsystem.
 - Många likheter med digital hårdvaruutveckling och processor utveckling.
 - börja jobba på Intel?

Vart hittar vi mikrodatorsystem?

- PC
- Bilen (tändningen, färddator, bromsar, etc...)
- Videospel
- Mikrovågsugn
- TV, video
- Mobiltelefon
- Trafikljus
- Etc...



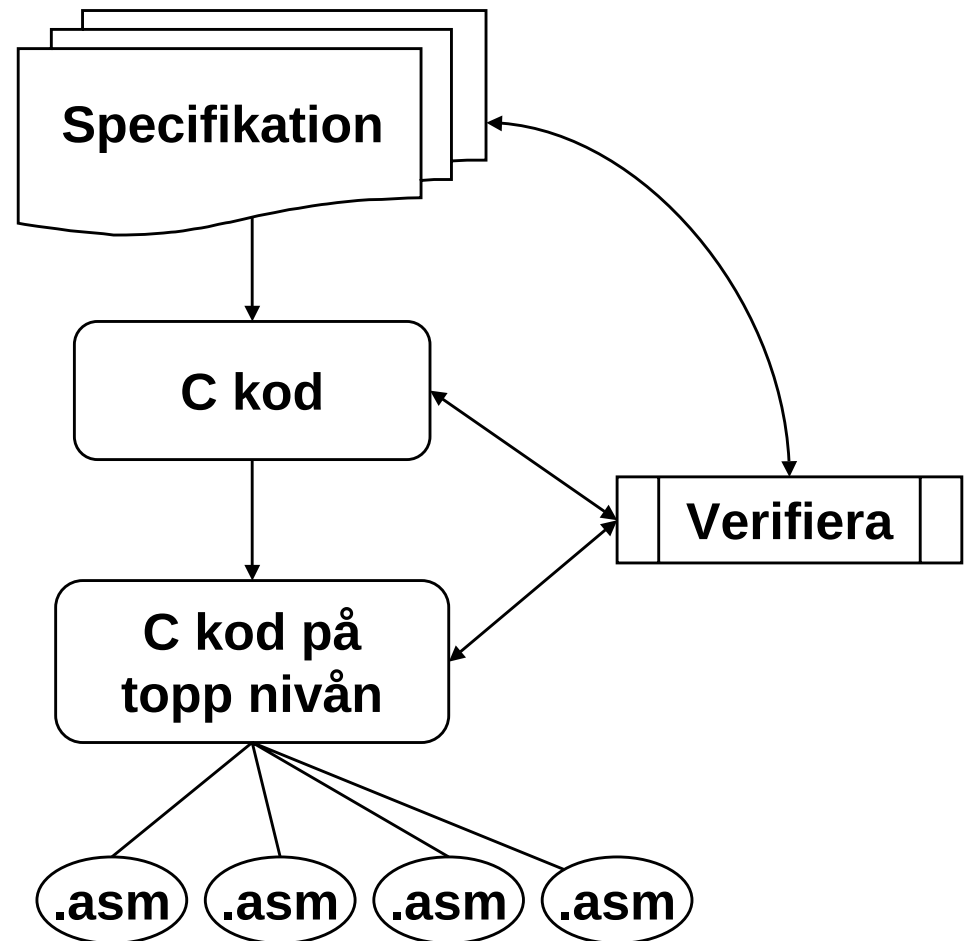
Varför ska vi lära oss assembler?

- **Bakgrundskunskap**
 - För att skriva högkvalitativ kod måste man förstå hur ett program exekveras
 - Ibland måste man programmera i assembler
- **När behöver man programmera assembler?**
 - Inte ofta...
 - Programmering av processorer med speciella arkitekturer
 - T ex DSP processorer
 - Utveckling av realtids/operativsystemkärnor
 - Utveckling av kompilatorer
 - Programmering av objekt, där det är hårda krav på minnes användning och/eller exekverings tid.
 - Det blir svårare att skriva snabbare assemblerkod än vad en kompilator kan, eftersom det blir svårare att förstå arkitekturen och hur instruktioner ska schemaläggas.



Konstruktionsflöde för assembler-programmering i industrin.

- Från en specifikation skrivs en implementering i C kod
 - Denna används för att verifiera funktionen och för att lättare strukturera programmet
- Innehållet i funktionerna ändras sedan till assemblerkod.
- Topp nivån är fortfarande skriven i C-kod (oftast)
 - Underlättar strukturering



Processor- och datorhistoria

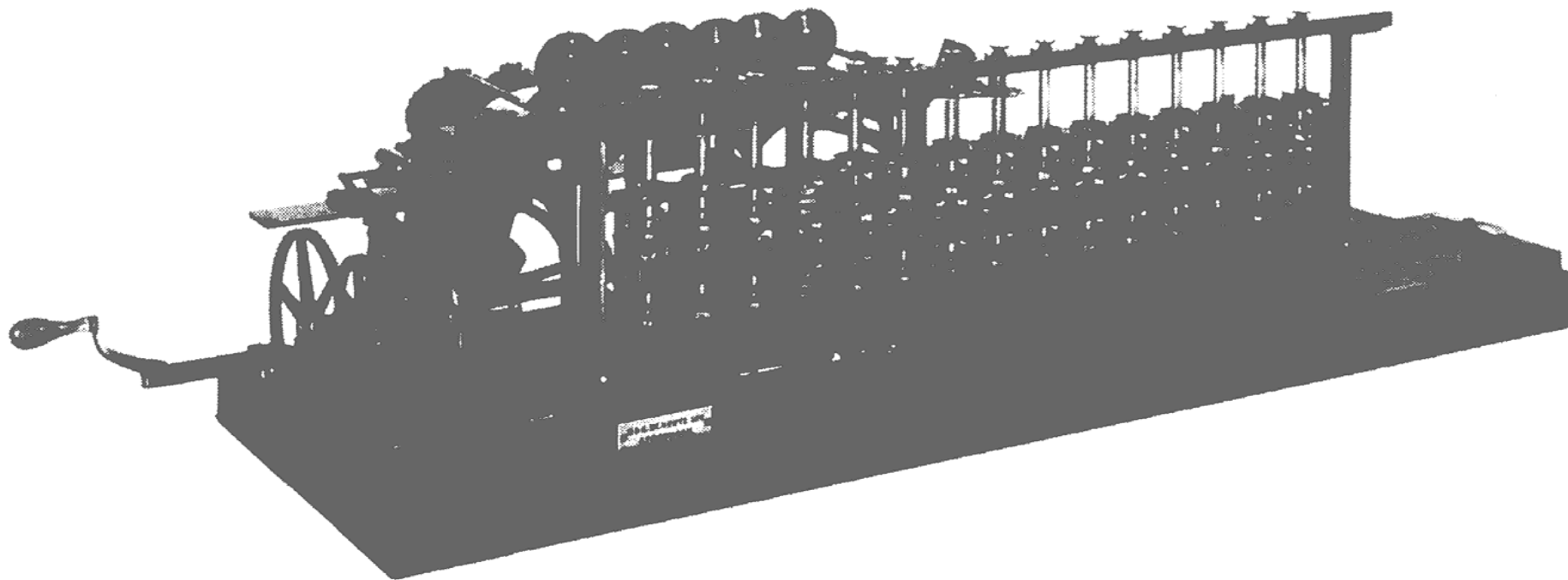
- **Mekaniska datorer, -1944**
- **Elektronrörbaserade datorer, 40-talet**
- **Transistorbaserade datorer, 50-talet**
 - Diskreta transistorer
- **IC-baserade datorer, 60-talet**
 - Diskreta logiska grindar (typ 74xx serien)
- **Datorer med mikroprocessor på ett chip, 70,80-talet**
- **Datorer med superskalära mikroprocessorer, 90-talet**
 - Fler exekveringsenheter på ett chip

Historia - Mekaniska 'datorer'

År	Upphovsman	Maskin	Räknesätt	Innovationer
1642	Pascal	-Pascaline	Add, Sub	Carrybit och negativa tal
1671	Leibniz	-	Add, Sub, Mul, Div	Mekanik för mul & div
1805	Jacquard	Jacquard maskinen	-	Väva mönster enligt hålkortsprogram
1843	Scheutz	Difference Engine	Maclaurin utveckling	Flerstegsoperationer
1834-1871	Babbage	Analytical Engine	Generell beräkning	Programvara, sekvensstyrning
1941	Zuse	Z3	Generell beräkning	Första fungerande generella datorn
1944	Aiken	Mark I	Generell beräkning	

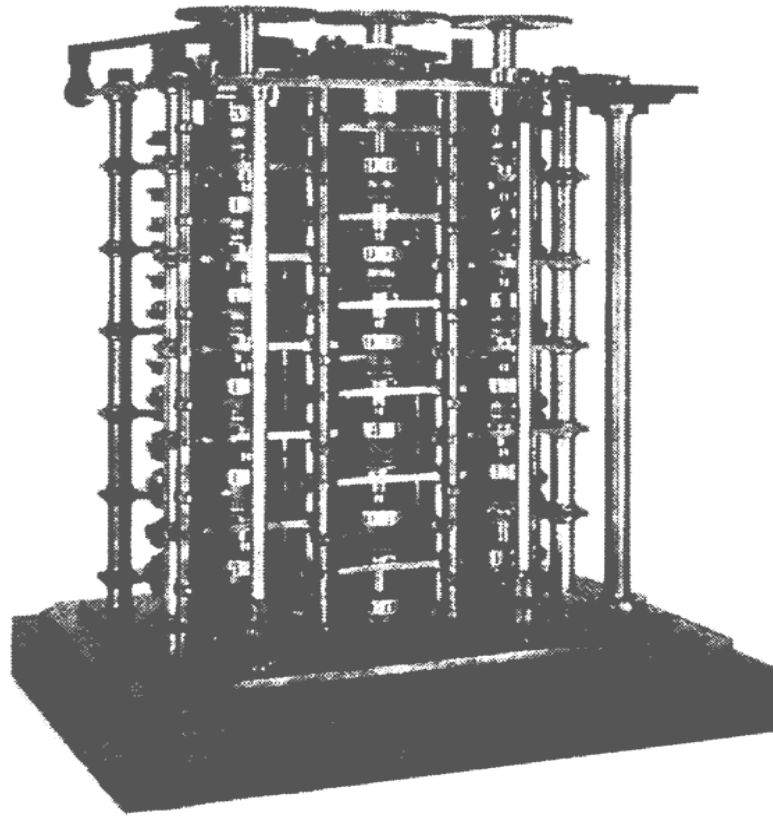


Scheutz, differensmaskin



Figur 2.3. Världens första dator – Scheutz differensmaskin.

The Difference Engine, Babbage

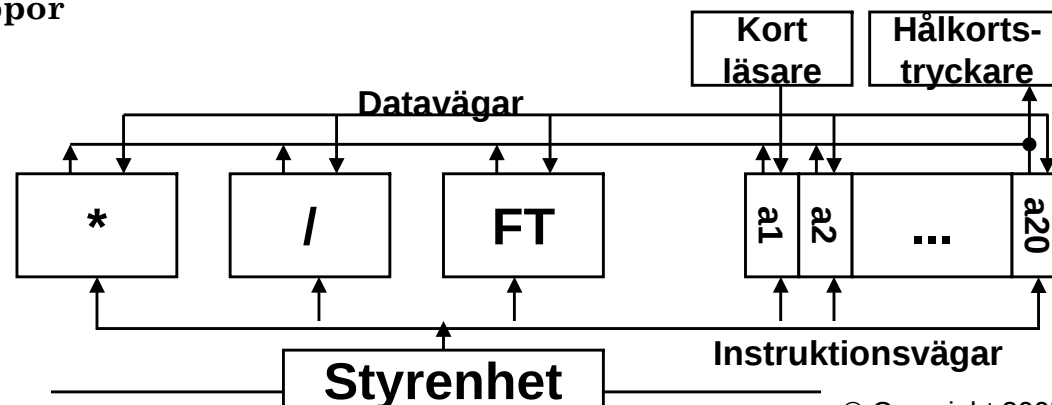


Figur 2.2. The Difference Engine, färdigställd efter Babbages död.

Historia - 1:a generationen

Elektronrörsdatorer

- 1906 uppfinner Lee de Forest elektronröret
- 1945 ENIAC
 - Electronic Numerical Integrator and Calculator
 - **Vägde 30 ton!!!**
 - 30m², 18000 elektronrör
 - $f_{clk} = 100\text{kHz}$
 - Minnet bestod av 20 st 10-siffriga tal
 - Varje tal representerades av 10st vippor
- 1943, Colossus (Turing)
 - Engelska försvaret
 - Specialiserad maskin inriktat mot att knäcka koder
 - Mindre och snabbare än ENIAC
- 1951, EDVAC
 - (Electronic Descreate Variable Automatic Computer)
 - Baserad på von Neumanns arkitektur



ENIACs arkitektur

Historia - 2:a generationen

Transistordatorer

- 1948 uppfinner Shockley transistorn
- 2:a generationen
 - Transistorer
 - Sekundär minnen
 - Magnetiska trummor
 - Flyttalsenheter
 - Högnivå språk
 - FORTRAN (FORmula TRANslation, 1954)
 - COBOL (Common Business Oriented Language, 1959)
- T ex
 - IBM 7090, IBM 7094
 - 400 sålda



Historia - 3:e generationen

Integrerade kretsar

- **3:e generationen**
 - Integrerade kretsar
 - Mikroprogrammering
- **IBM 360, 1965**
 - Sex olika, men alla var binärkodskompatibla men skillnad i prestanda.
- **Första superdatorn**
 - 1964 CDC (Thornton, Cray)

Historia - 4:e generationen

Mikroprocessorer

- **Aka, VLSI generationen**
- **Första mikroprocessorn, 1971**
 - **Intel 4004**
 - 2300 P-MOS transistorer
 - 4-bitar
 - Utvecklades för att användas i en miniräknare
 - Efterföljdes av i8008(72) och senare i8080(73)
- **Efterföljare till Intel**
 - Motorola 6800, Zilog Z80, Intel 8085
- **Senare mer komplexa mikroprocessorer**
 - i80286, Motorola 68000
 - Samma koncept, men större (fler bitar och snabbare)



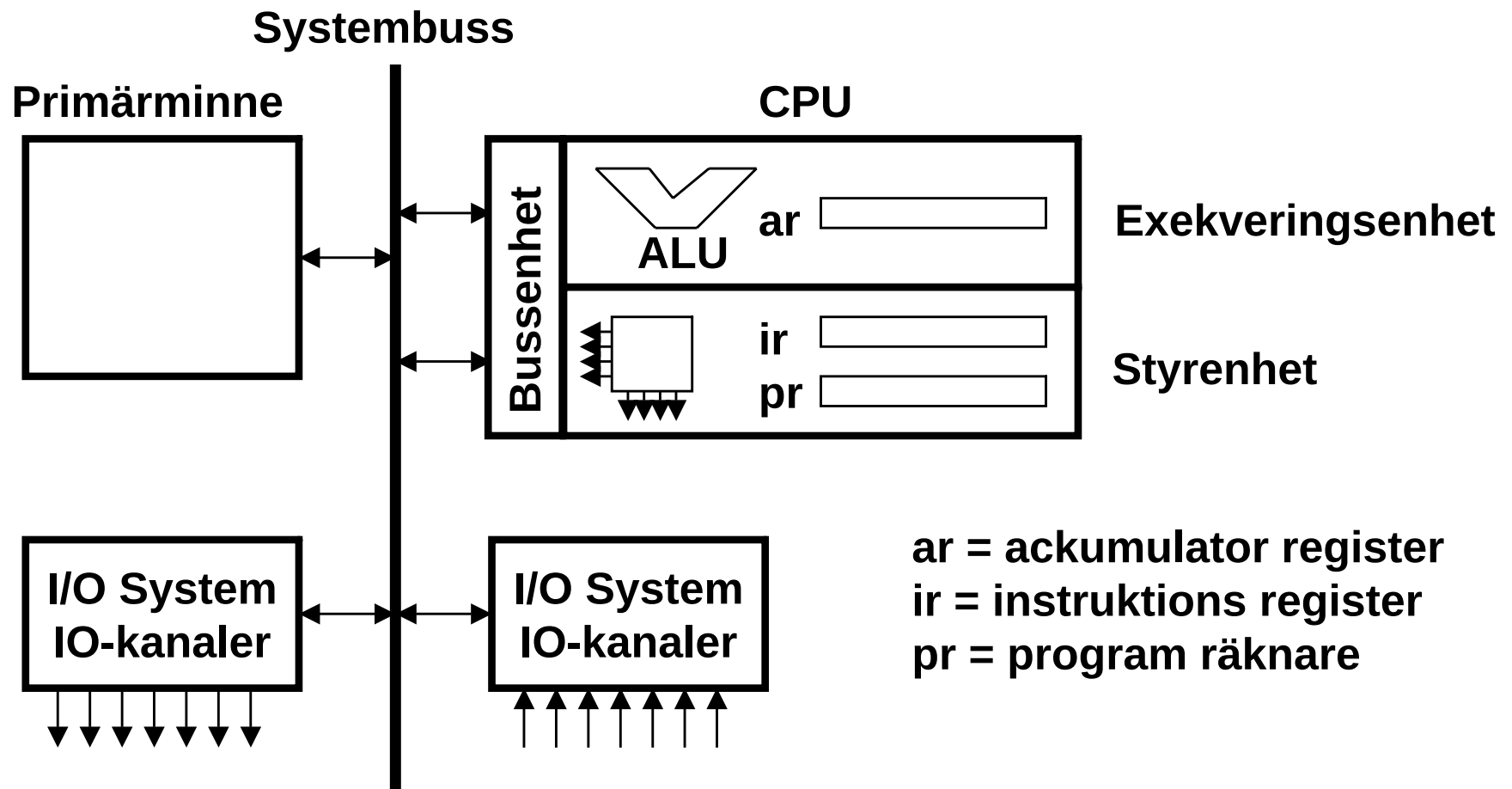
Historia - 5:e generationen

Superskalära processorer

- Pipeline, cache, parallella
- Intel Pentium
 - 2 exekveringsenheter för heltal
 - 1 flyttalsenhet
 - Cache: L1: 16 kbyte
 - 1.8 M transistorer
- Alpha 21384
 - 4 exekveringsenheter för heltal
 - 2 exekveringsenhet för flyttal
 - Cache: L1: 128k, L2: 1.5M
 - 100 M transistorer

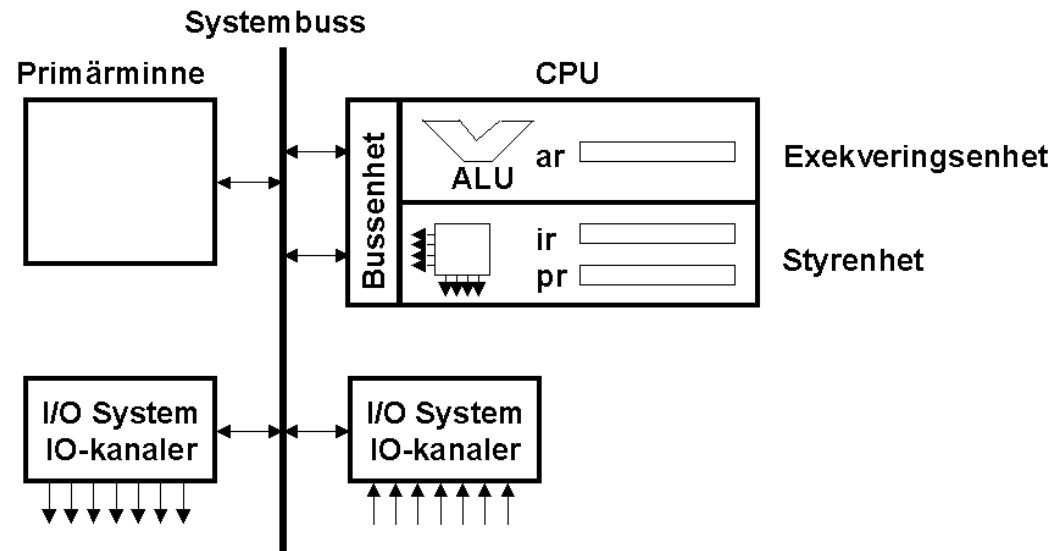


von Neumann arkitekturen



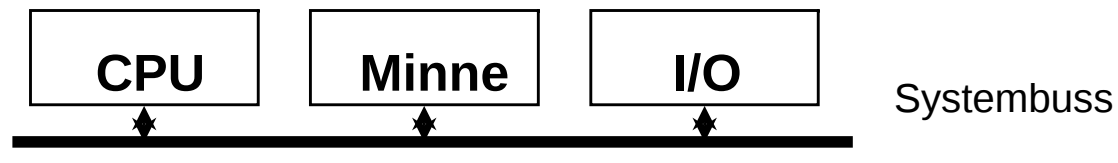
Exekvering av von Neumann datorn

- Utför: $Y=a+b$ (ADD Y, a, b)
 - Adressbuss sätts till **pr**, **ir** laddas med värde från databussen
 - Avkoda instruktionen
 - Ladda **ar** med a
 - Addera **ar** med b
 - Lagra **ar** i Y
- Sekvensen kallas mikroprogram



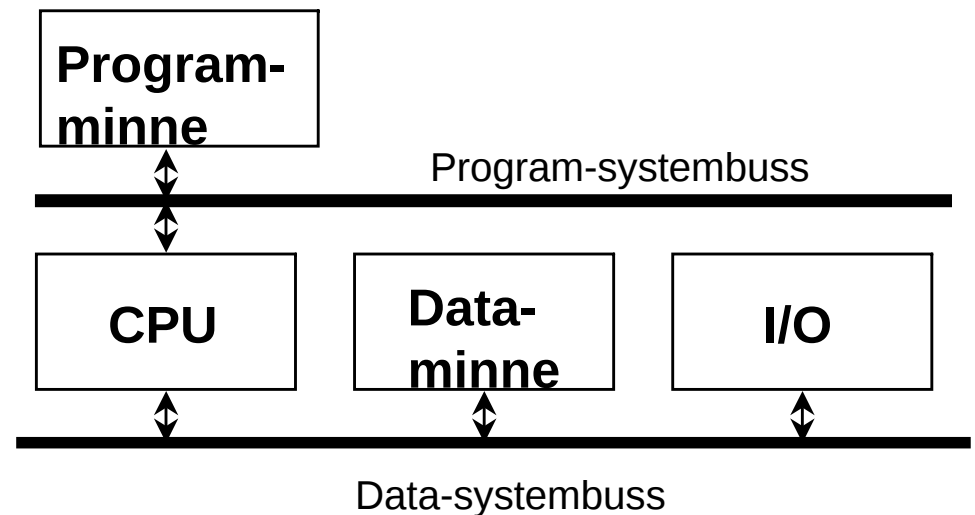
Von Neumann maskinen, forts.

- von Neumann arkitekturen har bara en minnesarea (en systembuss)
 - kod exekveras sekventiellt
 - Instruktion i von Neumann arkitekturen utförs enligt följande
 - läs operationsord, avkoda, läs operand, *utför op.*, skriv resultat
 - Enkel arkitektur för datorkonstruktörer, programmerare, kompilatorkonstruktörer
 - Ineffektiv, bara 20% av exekveringstiden används för verklig exekvering (ALU operationer)
 - Mest använda mikroprocessorarkitekturen (dock modifierad)
- Vad kan göras för att effektivisera denna arkitektur
 - Prefetch, cache, pipelining, branch prediction, DMA



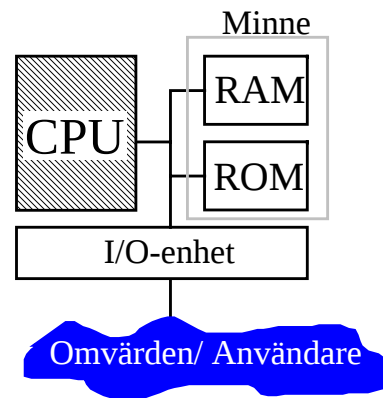
Harvard arkitekturen

- **Två bussar**
 - En för instruktioner
 - En för data
 - Dvs: Data och instruktioner kan laddas samtidigt, vilket ökar effektiviteten.
- **Används internt i moderna mikroprocessorer**



Datorsystemets olika delar

- **Vad innehåller ett datorsystem**
 - Processor (CPU, Central Processing Unit)
 - Minne
 - Perferienheter, I/O
 - Systembuss, för att kommunicera med yttre enheter
- **Har vi ett chip som innehåller allt detta kallas detta ofta för en 'Micro Controller'**

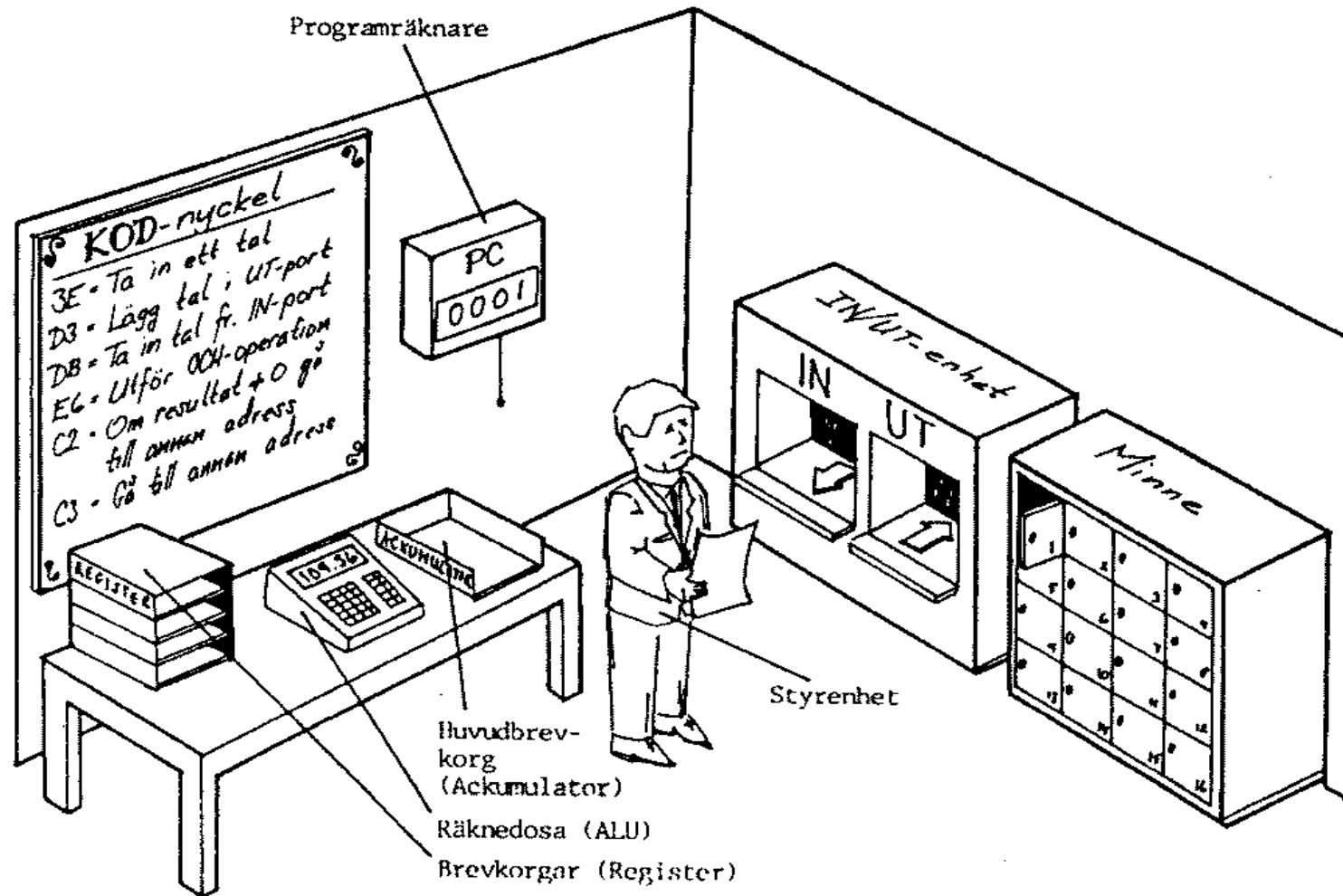


Mikroprocessorn

- **Datorns hjärna**
- **En mikroprocessor klassificeras i termer av**
 - Klockfrekvens (1 MHz - 2000 MHz)
 - Antal bitar som processorn kan bearbeta (8,16,32,64-bit)
 - Dataräckvidd (Adresseringsområde) (64k - 4G)
 - MIPS (Mega Instructions per Second)
 - Benchmarkprestanda (SpecINT, SpecFP)
- **Utför en serie instruktioner, dvs ett program.**
 - Instruktionerna finns lagrade i ett minne
 - Matematiska operationer
 - Logiska operationer
 - Kontrollinstruktioner



Processorn



Så här kan vi tänka oss "kontoret" hos processorn.

"Två" typer av processorer

- **CISC (Complex Instruction Set Computer)**
 - Många instruktioner för att underlätta för assembler-programmeraren och kompilator konstruktören
 - Nackdelen är att kompilatorn använder ett fåtal instruktioner och konstruktionen av en komplex processor blir långsam
 - Det krävs mycket information för att beskriva en instruktion
 - T.ex.. MC68040 med 200 instruktioner och 18 adresseringsmoder
- **RISC (Reduced Instruction Set Computer)**
 - Enklare instruktioner (som används av kompilatorn)
 - Ger snabbare implementering
 - Endast ett kodord för att beskriva en instruktion
 - T.ex.. Alpha, ARM, och MIPS processorn



Processorns systembus

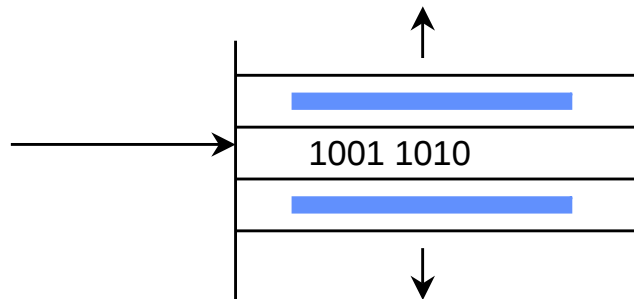
- **Databuss**

- Kommunikationskanal för att flytta data till och från CPU och yttre enheter

- **Adressbuss**

- Används för att peka ut vilka viken minnesposition/IO-port som ska läsas eller skrivas

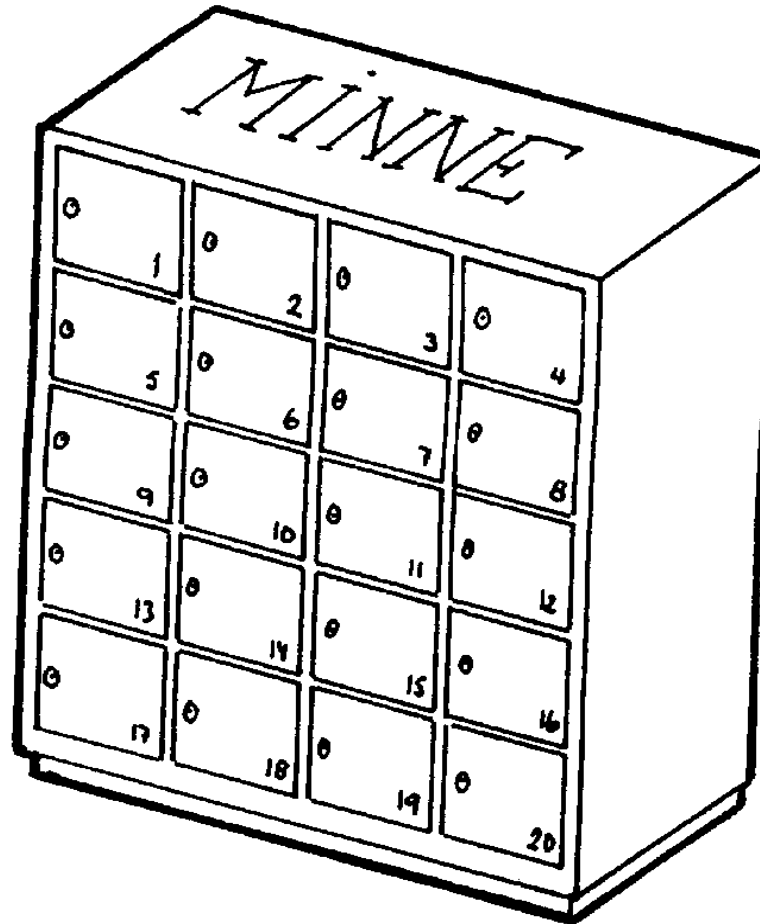
1000 0000 1000 1010



- **Styrsignaler**

- Används för att signalera när en dataöverföring startar och slutar.
- Signalerar om det är en läs eller en skrivoperation

Minne



Minnet kan betraktas som en mängd postfack.

Minnestyper

- **RAM**

- **Statiskt (SRAM, Static Random Access Memory)**

- Lagrad data finns kvar tills spänningen försvinner

- **Dynamiskt (DRAM, Dynamic Random Access Memory)**

- Lagrad data försvinner efter ett tag, man måste påminna minnet med jämna mellanrum.

- **ROM(Read Only Memory)**

- **Data lagras en gång, och finns kvar till domedagen.**

- **Många typer av ROM**

- ROM
 - PROM
 - EPROM
 - EEPROM



I/O - kretsar

- **Kommunikation kretsar**
 - RS 232, Ethernet, etc.
- **AD/DA omvandlare**
 - ADC, Omvandlar analoga signaler till digitala signaler.
 - DAC, Omvandlar digitala signaler till analoga signaler.
- **I/O enheter**
 - Tangentbord
 - Display/Monitor
 - Sensorer

The memory wall

