



HF0010

Introduktionskurs i Datateknik

F1: Vad är en dator?



Introduktionskurs i Datateknik

F1: Vad är en dator?

F2: programmeringsspråk?

L1: Hour of code, alla kan programmera, Diplom!

L2: Visuellt programmering av svårare problem!

F3: Webbprogrammering – JavaScript!

L3: IDE, Web sida & JavaScript!



HTML





Introduktionskurs i Datateknik

ALLTID
ALLTID
ALLTID
ALLTID
ALLTID
ALLTID



Account



Dashboard

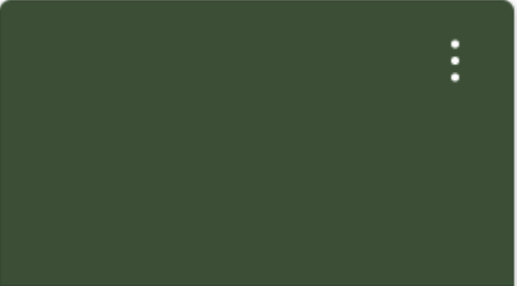


Courses



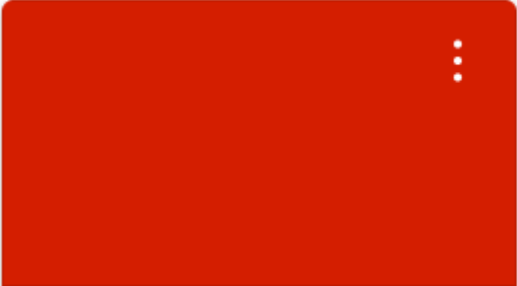
Calendar

Dashboard



Examensarbete data/elektrotekni...
Examensarbete data/elektrote...





HE1033 VT17-1 Kommunikations...
HE1033





HF0010 HT17-1 Introduktionskur...
HF0010

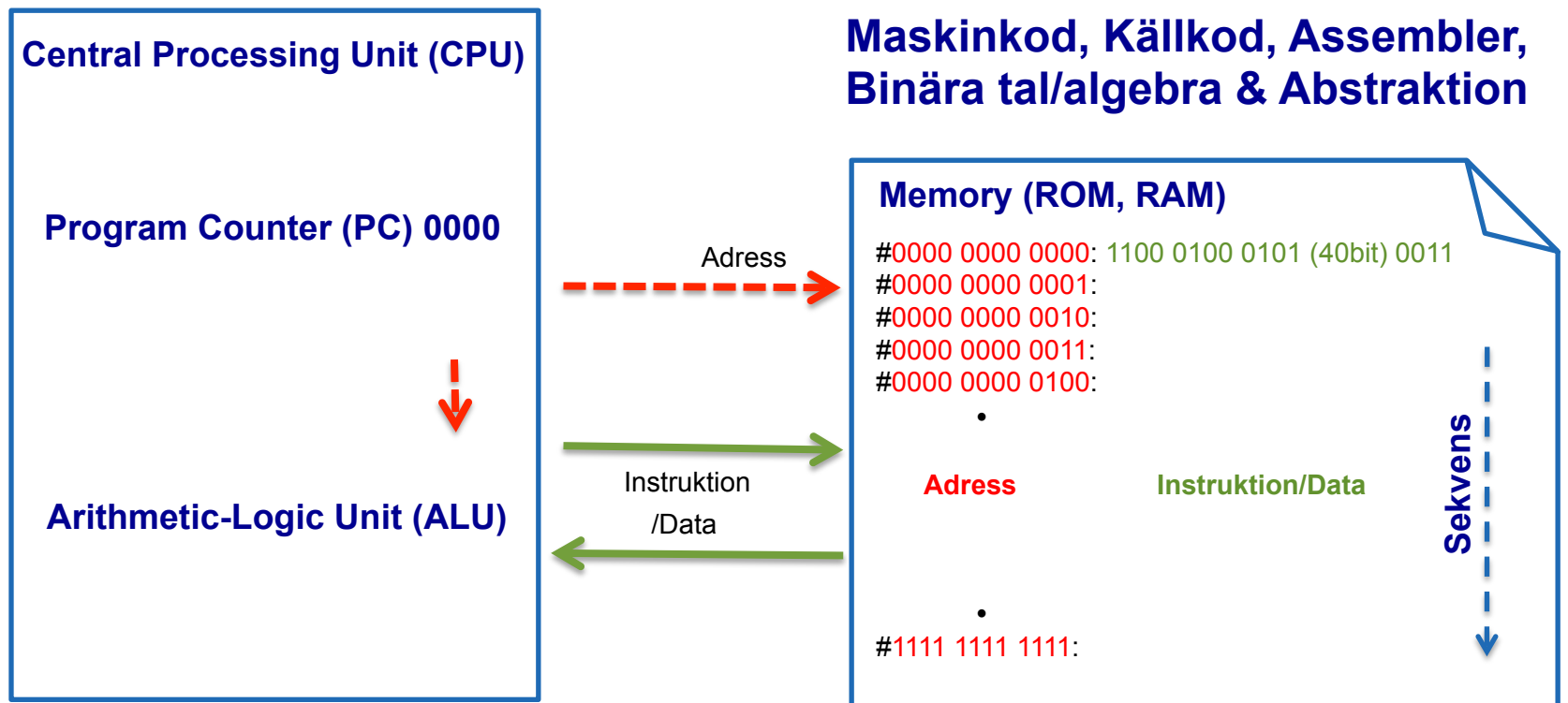
ALLTID
ALLTID
ALLTID
ALLTID

Läs på/följ - kursinfo i Canvas!



Kursmål

Sidor märkta med den här symbolen innehåller begrepp som är extra viktiga för dina fortsatta studier!



Harvard vs von Neumann arkitektur

16 Begrepp du bör känna till efter föreläsning #1!

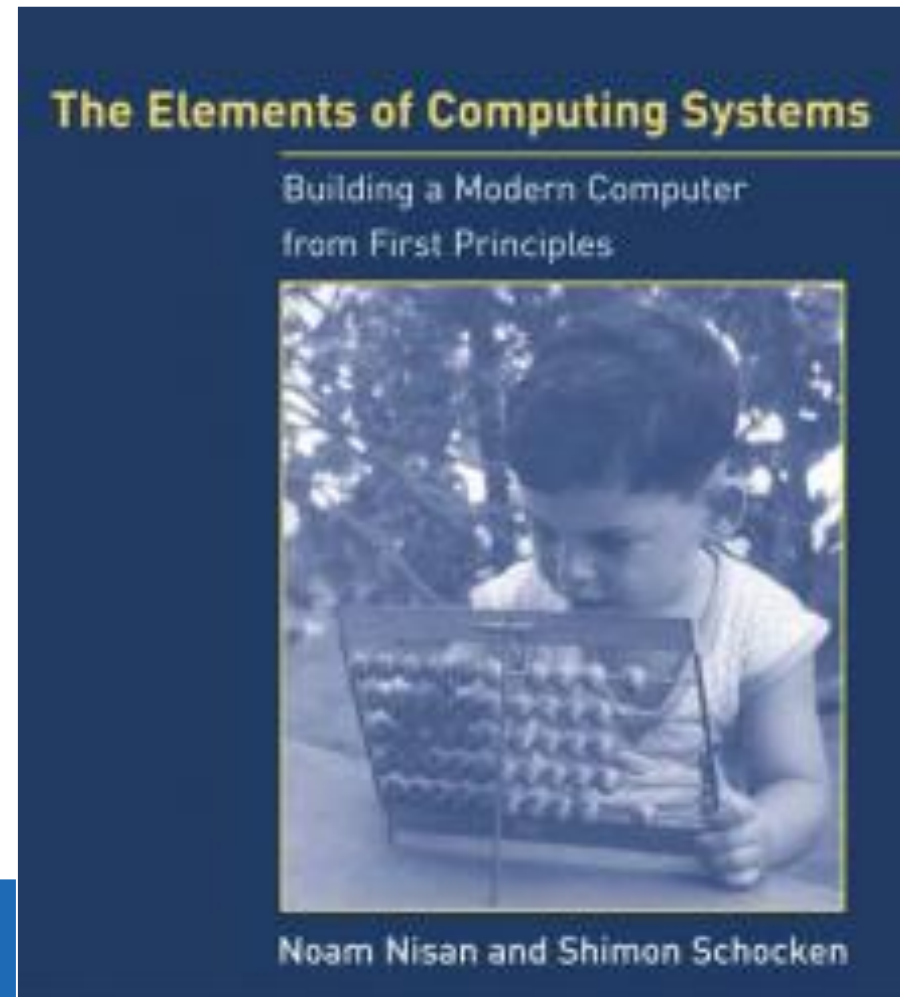


Pedagogik...

Vi börjar från noll...

...och försöker förstå helheten!

Bra, gratis, bok att ha vid sidan om...
<http://www.nand2tetris.org>





Rätt komplex pryl...





Rätt komplex pryl...



Likheter ur användarsynpunkt?!

Rätt komplex pryl...**abstraktion/gränssnitt!**



Bägge styrs med en ratt (men de går lite olika fort)!



Rätt komplex pryl...**abstraktion/gränssnitt!**

SW



PROGRAM
KOMPILATOR
OPERATIVSYSTEM



Källkod

ASSEMBLER

Assemblerkod

Maskinkod



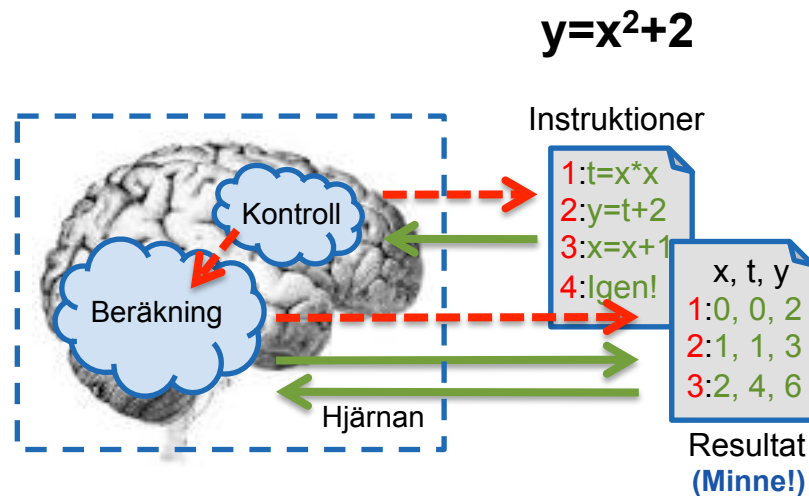
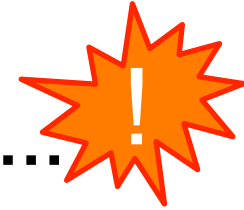
HW



**Hårdvara – vad kan en dator egentligen göra?!
,och hur?!**



Människan som dator, papper & penna...



1. Papper med instruktioner...
2. Hjärnan tolkar första instruktionen...
3. Hjärnan utför första instruktionen...
4. (Mellan-)resultat dokumenteras!
5. Hjärnan tolkar nästa instruktion...
(Sekvens!)

Krux:

Begränsad beräkningshastighet, 10 sek/operation.
Notorisk förmåga att göra fel (distraherad, trött...)!

...i en ny tid som behövde tabeller!

- 1794 fick G. F. Prony i uppdrag av Franska staten att leda en satsning att beräkna bl.a. Logaritmtabeller för alla naturliga tal mellan 1 och 200.000 med 19 decimaler!
- Ca 100 personer arbetade parvis i 2 år och resultatet blev 17 gigantiska böcker med ca 8 millioner siffror.
- Sen började man med sin och cos...
- (Tabellerna innehöll trots alla ansträngningar ett antal fel...)



TABLE V.
Logarithmes à 19 décimales pour tous les nombres impairs de 1163 à 1501, et pour tous les nombres premiers de 1501 à 10000.

Nota. Cette Table fait suite aux logarithmes à 20 décimales des Tables de Gardiner, édit. d'Avignon. Elle est extraite des grandes Tables du Cadastre, déposées au Bureau des Longitudes, et dont la notice se trouve dans le tome V des Mémoires de l'Institut.

Nomb.	Logarithmes.	Nomb.	Logarithmes.	Nomb.	Logarithmes.
1163	06557 07147 38448 4114	1243	09447 11286 41644 7035	1323	12155 08441 87500 0733
1165	06635 30255 62037 7760	1245	09516 03514 31755 1450	1325	12221 58782 20800 6552
1167	06703 08500 42370 7755	1247	09586 64234 28512 6137	1327	12289 09228 64435 5110
1169	06781 35111 61810 1107	1249	09656 23383 24135 5120	1329	12355 49800 42731 0972
1171	06855 68050 72363 1209	1251	09725 73066 03419 9551	1331	12422 80555 26625 1123
1173	06930 80121 15502 2447	1253	09795 10709 04109 0908	1333	12488 01404 13859 2061
1175	07003 20006 07255 0710	1255	09864 37258 17050 0141	1335	12554 19057 00501 0268
1177	07077 61608 43414 0810	1257	09933 55766 08075 1772	1337	12621 11071 61981 3085
1179	07151 38050 05080 1354	1259	10002 57501 07865 5075	1339	12688 02770 13008 0946
1181	07224 09040 13514 2091	1261	10071 50805 73081 6310	1341	12754 87228 51508 0170
1183	07298 27470 27030 3691	1263	10140 33505 55330 7447	1343	12820 60191 08715 3305
1185	07371 88563 40122 6701	1265	10209 65055 11836 7244	1345	12887 23843 38426 2440
1187	07445 09180 54501 2104	1267	10277 66438 03441 3410	1347	12954 75075 20070 6222
1189	07518 18516 18601 5818	1269	10346 16220 04704 7763	1349	13021 10406 71004 2470
1191	07591 19014 82777 5032	1271	10414 55505 04008 7763	1351	13088 53400 22030 5013
1193	07664 04430 70311 8728	1273	10483 84366 53055 3057	1353	13154 77005 07020 0720
1195	07736 72052 81166 4829	1275	10551 01847 60973 0724	1355	13221 09050 10411 5343
1197	07809 41504 06110 6608	1277	10619 08072 63125 2866	1357	13287 08140 50737 0801
1199	07881 01830 08048 6760	1279	10687 05444 28053 0236	1359	13354 01507 35403 3114
1201	07954 30074 09000 0480	1281	10754 12107 44086 3019	1361	13421 81252 03334 6002
1203	08026 36073 30844 7138	1283	10822 06603 74928 5036	1363	13488 38538 31673 3572
1205	08098 72089 10809 1880	1285	10890 31107 67113 3410	1365	13554 26513 70771 8420
1207	08170 27000 07340 3110	1287	10957 85400 01386 6846	1367	13621 85145 67012 2700
1209	08242 63008 60771 8801	1289	11025 20173 53403 0211	1369	13688 34401 35080 0930
1211	08314 41431 43622 2453	1291	11092 64422 66420 3088	1371	13754 74547 80512 6507
1213	08386 20008 60570 9747	1293	11159 85128 86321 2381	1373	13821 03772 30755 1114
1215	08457 60770 34320 0913	1295	11226 07864 17270 6323	1375	13888 21000 60080 4500
1217	08529 05702 30064 0888	1297	11293 09760 84000 0814	1377	13954 30100 50020 6772
1219	08600 37050 10381 0245	1299	11360 01510 73027 8800	1379	14021 40001 25010 7581
1221	08671 50630 44880 1740	1301	11427 29065 61280 2544	1381	14088 50001 00001 2814
1223	08742 64570 30005 0631	1303	11494 44457 12884 6106	1383	14155 60001 00001 2814
1225	08813 60005 00551 2710	1305	11561 05116 12009 7807	1385	14222 70001 00001 2814
1227	08884 45005 20004 2400	1307	11627 55855 80544 2078	1387	14289 80001 00001 2814
1229	08955 18008 06454 0805	1309	11694 06465 50755 8000	1389	14356 90001 00001 2814
1231	09026 80000 31310 3070	1311	11760 00000 00000 2777	1391	14423 00001 00001 2814
1233	09097 30705 05731 6130	1313	11826 17000 80770 3435	1393	14489 10001 00001 2814
1235	09168 00575 09684 5355	1315	11892 27500 25770 6138	1395	14556 20001 00001 2814
1237	09239 00000 20120 6536	1317	11958 37400 61883 8079	1397	14622 30001 00001 2814
1239	09307 13003 70603 4583	1319	12024 47955 03005 2065	1399	14689 40001 00001 2814
1241	09377 17814 90700 8050	1321	12090 58170 14507 2041	1401	14755 50001 00001 2814

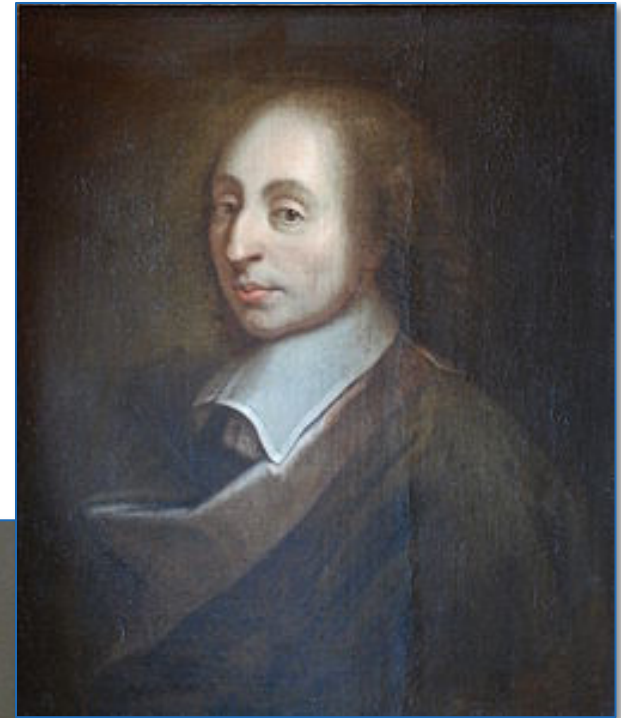
$$\log(x) = x - 1 - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \dots$$



Den mekaniska eran: 1623-1945
Byggstenar I...

Blaise Pascal (1623-1662)

+ -



Basen

10

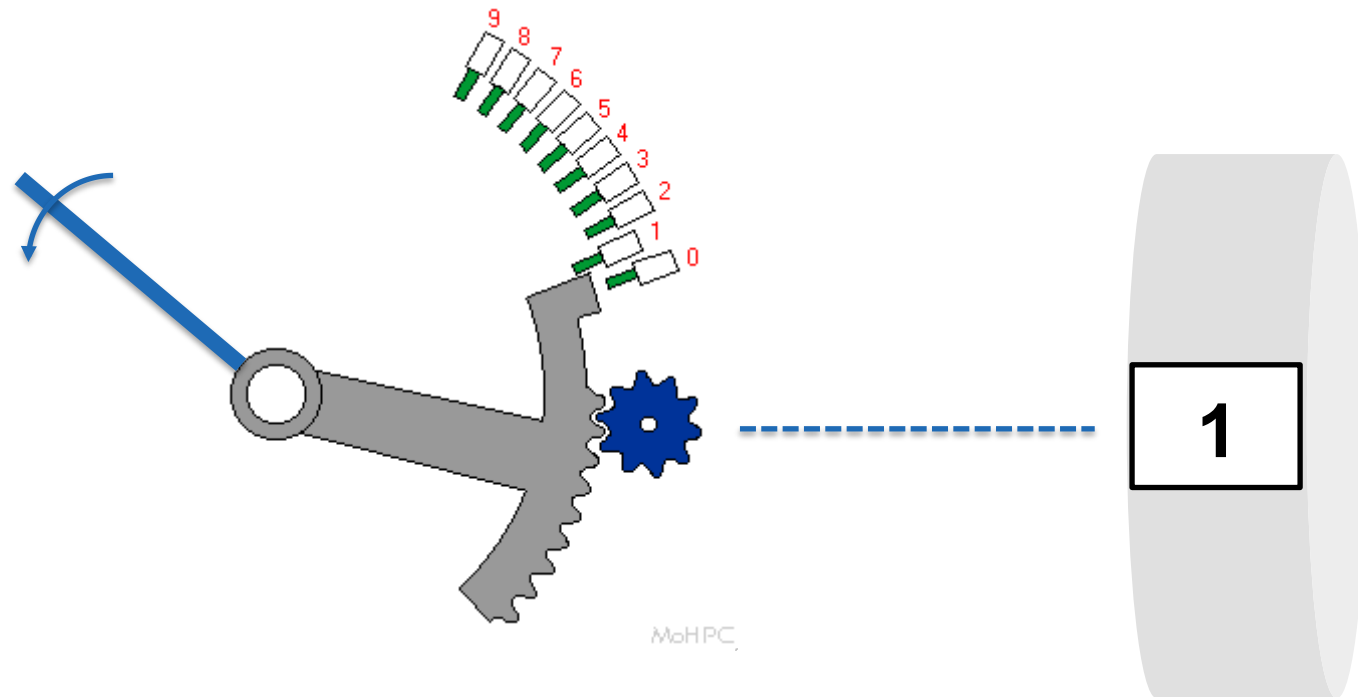
Gottfried Leibniz (1646-1716)

* /



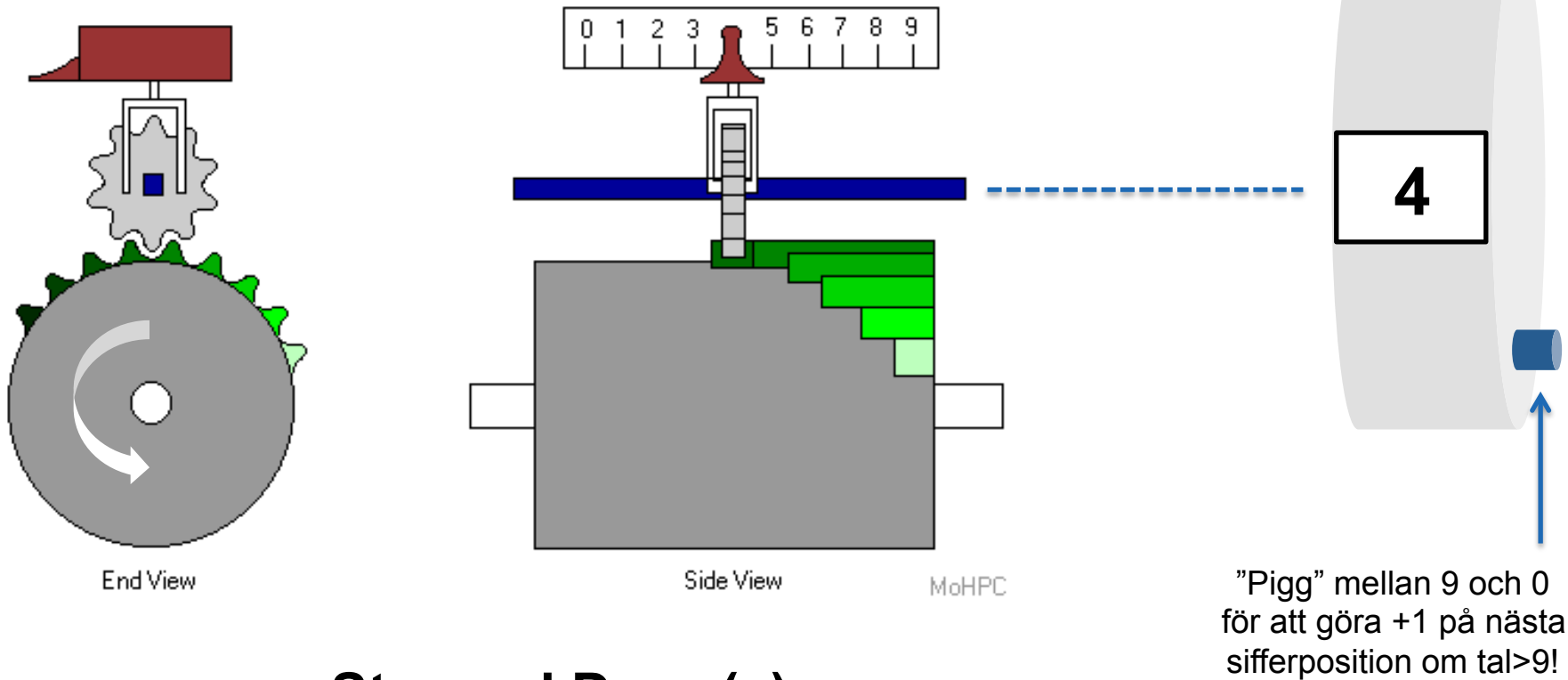
Facit 1954

Byggstenar I...



Rocking Segments

Byggstenar I...



Stepped Drum(s)



Den mekaniska eran: 1623-1945
Byggstenar I...



source: E. Anthes, Historische Bürowelt

Curt Herzstark
(July 26, 1902 - October 27, 1988)



Byggstenar II...



Utvecklades 1801--05



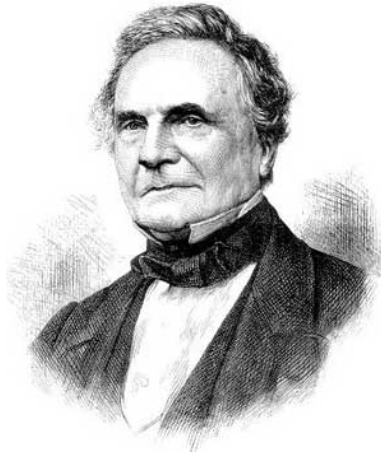
11000 enbart i F 1811!

Joseph Marie Jacquard (Lyon, 1752 - Oullins, 1834)



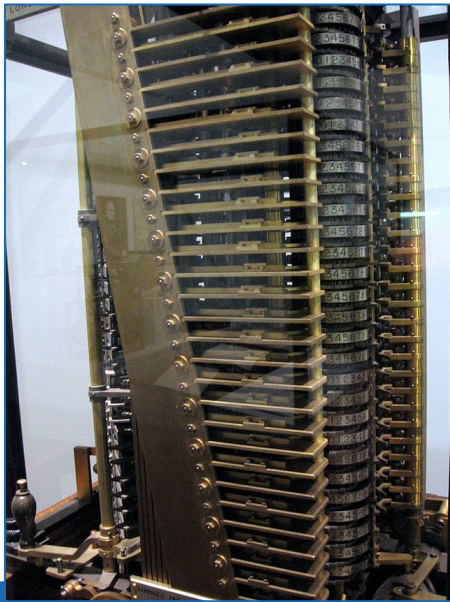
Den mekaniska eran: 1623-1945

Den första datorn...

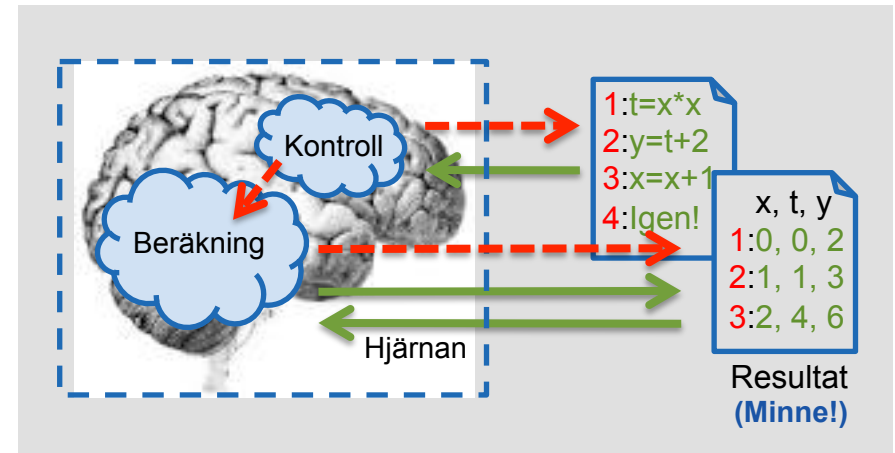


Charles Babbage
1791-1871

The Analytical Engine



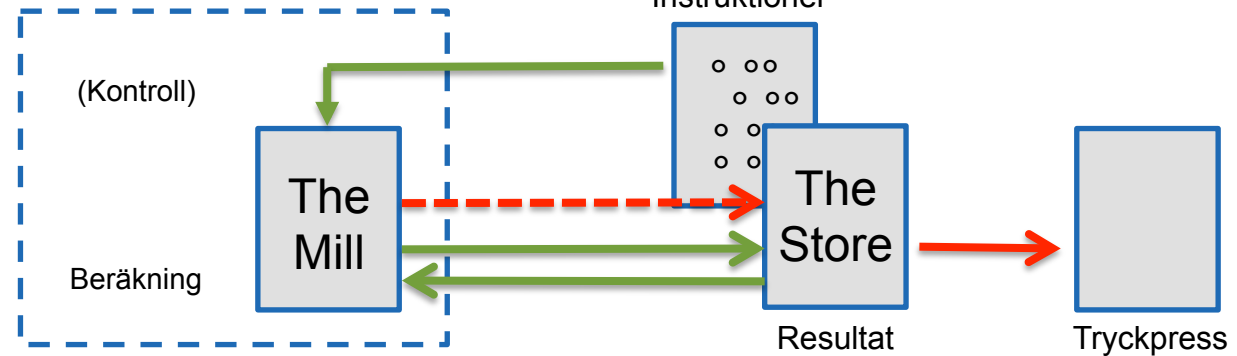
The Mill



Instruktioner

Basen

10



Addition = 1 sekund
Multiplikation = 1 minut

1000 st
50 siffriga tal

Den första datorn...

+ - * /
Load
Store



Ada King,
Countess of
Lovelace,
1840

Directive Card	Operation Card	Action
1st	...	Places a on column 1 of Store
2nd	...	" b " 2 "
3rd	...	" c " 3 "
4th	...	" d " 4 "
5th	...	Brings a from Store to Mill
6th	...	" b " "
...	1	Multiplies a and $b = p$
7th	...	Takes p to column 5 of Store where it is kept for use and record
8th	...	Brings p into Mill
9th	...	Brings c into Mill
...	2	Adds p and $c = q$
10th	...	Takes q to column 6 of Store
11th	...	Brings d into Mill
12th	...	" q " "
...	3	Multiplies $d \times q = p_2$
13th	...	Takes p_2 to column 7 of Store
14th	...	Takes p_2 to printing or stereo-moulding apparatus

Beräkna $(ab+c)d$

Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 *et seq.*)

Number of Operation.	Nature of Operation.	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results.	Data.												Working Variables.			Result Variables.							
						1V ₁	1V ₂	1V ₃	0V ₄	0V ₅	0V ₆	0V ₇	0V ₈	0V ₉	0V ₁₀	0V ₁₁	0V ₁₂	0V ₁₃	1V ₂₁	1V ₂₂	1V ₂₃	0V ₂₄						
						1	2	n																				
1	×	1V ₂ × 1V ₃	1V ₄ , 1V ₅ , 1V ₆	$\left\{ \begin{matrix} 1V_2 = 1V_2 \\ 1V_3 = 1V_3 \end{matrix} \right\}$	= 2n	...	2	n	2n	2n	2n																	
2	-	1V ₄ - 1V ₁	2V ₄	$\left\{ \begin{matrix} 1V_2 = 2V_4 \\ 1V_3 = 1V_3 \end{matrix} \right\}$	= 2n - 1	1	...	...	2n - 1																			
3	+	1V ₅ + 1V ₁	2V ₅	$\left\{ \begin{matrix} 1V_2 = 2V_5 \\ 1V_3 = 1V_3 \end{matrix} \right\}$	= 2n + 1	1	...	...	2n + 1																			
4	+	2V ₅ + 2V ₄	1V ₁₁	$\left\{ \begin{matrix} 2V_5 = 0V_4 \\ 2V_4 = 0V_4 \end{matrix} \right\}$	= $\frac{2n-1}{2} + 1$	...	...	...	0	0																		
5	+	1V ₁₁ + 1V ₂	2V ₁₁	$\left\{ \begin{matrix} 1V_{11} = 2V_{11} \\ 1V_2 = 1V_2 \end{matrix} \right\}$	= $\frac{1}{2} \cdot \frac{2n-1}{2n+1}$	...	2	...																				
6	-	0V ₁₃ - 2V ₁₁	1V ₁₃	$\left\{ \begin{matrix} 2V_{11} = 0V_{13} \\ 0V_{13} = 1V_{13} \end{matrix} \right\}$	= $-\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_0$	...	...	...																				
7	-	1V ₉ - 1V ₁	1V ₁₀	$\left\{ \begin{matrix} 1V_9 = 1V_1 \\ 1V_1 = 1V_1 \end{matrix} \right\}$	= n - 1 (= 3)	1	...	n							n - 1													
8	+	1V ₂ + 0V ₇	1V ₇	$\left\{ \begin{matrix} 1V_2 = 1V_2 \\ 0V_7 = 1V_7 \end{matrix} \right\}$	= 2 + 0 = 2	...	2	...				2																
9	+	1V ₆ + 1V ₇	3V ₁₁	$\left\{ \begin{matrix} 1V_6 = 1V_6 \\ 0V_{11} = 3V_{11} \end{matrix} \right\}$	= $\frac{2n}{2} = A_1$	...	...	...			2n	2																
10	×	1V ₂₁ × 3V ₁₁	1V ₁₂	$\left\{ \begin{matrix} 1V_{21} = 1V_{12} \\ 3V_{11} = 3V_{11} \end{matrix} \right\}$	= B ₁ · $\frac{2n}{2} = B_1 A_1$	...	...	...																				
11	+	1V ₁₂ + 1V ₁₃	2V ₁₃	$\left\{ \begin{matrix} 1V_{12} = 0V_{13} \\ 1V_{13} = 2V_{13} \end{matrix} \right\}$	= $-\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_1 \cdot \frac{2n}{2}$	...	...	...																				
12	-	1V ₁₀ - 1V ₁	2V ₁₀	$\left\{ \begin{matrix} 1V_{10} = 2V_{10} \\ 1V_1 = 1V_1 \end{matrix} \right\}$	= n - 2 (= 2)	1	...	...							n - 2													
13	-	1V ₆ - 1V ₁	2V ₆	$\left\{ \begin{matrix} 1V_6 = 2V_6 \\ 1V_1 = 1V_1 \end{matrix} \right\}$	= 2n - 1	1	...	...				2n - 1																
14	+	1V ₁ + 1V ₇	2V ₇	$\left\{ \begin{matrix} 1V_1 = 1V_1 \\ 1V_7 = 2V_7 \end{matrix} \right\}$	= 2 + 1 = 3	1	...	...				3																
15	+	2V ₆ + 2V ₇	1V ₈	$\left\{ \begin{matrix} 2V_6 = 2V_6 \\ 2V_7 = 2V_7 \end{matrix} \right\}$	= $\frac{2n-1}{3}$	...	...	...			2n - 1	3	$\frac{2n-1}{3}$															
16	×	1V ₈ × 3V ₁₁	4V ₁₁	$\left\{ \begin{matrix} 1V_8 = 0V_8 \\ 3V_{11} = 4V_{11} \end{matrix} \right\}$	= $\frac{2n}{2} \cdot \frac{2n-1}{3}$	...	...	...					0															
17	-	2V ₆ - 1V ₁	3V ₆	$\left\{ \begin{matrix} 2V_6 = 3V_6 \\ 1V_1 = 1V_1 \end{matrix} \right\}$	= 2n - 2	1	...	...			2n - 2																	
18	+	1V ₁ + 2V ₇	3V ₇	$\left\{ \begin{matrix} 1V_1 = 1V_1 \\ 2V_7 = 3V_7 \end{matrix} \right\}$	= 3 + 1 = 4	1	...	...				4																
19	+	3V ₆ + 3V ₇	1V ₉	$\left\{ \begin{matrix} 3V_6 = 3V_6 \\ 3V_7 = 3V_7 \end{matrix} \right\}$	= $\frac{2n-2}{4}$	...	...	...			2n - 2	4	$\frac{2n-2}{4}$															
20	×	1V ₉ × 4V ₁₁	5V ₁₁	$\left\{ \begin{matrix} 1V_9 = 0V_9 \\ 4V_{11} = 5V_{11} \end{matrix} \right\}$	= $\frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{4} = A_3$	...	...	...					0															
21	×	1V ₂₂ × 5V ₁₁	0V ₁₂	$\left\{ \begin{matrix} 1V_{22} = 1V_{12} \\ 0V_{12} = 2V_{12} \end{matrix} \right\}$	= B ₃ · $\frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{4} = B_3 A_3$	...	...	...																				
22	+	2V ₁₂ + 2V ₁₃	0V ₁₃	$\left\{ \begin{matrix} 2V_{12} = 0V_{13} \\ 2V_{13} = 2V_{13} \end{matrix} \right\}$	= A ₀ + B ₁ A ₁ + B ₃ A ₃	...	...	...																				
23	-	2V ₁₀ - 1V ₁	3V ₁₀	$\left\{ \begin{matrix} 2V_{10} = 3V_{10} \\ 1V_1 = 1V_1 \end{matrix} \right\}$	= n - 3 (= 1)	1	...	...							n - 3													
Here follows a repetition of Operations thirteen to twenty-three.																												
24	+	0V ₁₃ + 0V ₂₄	1V ₂₄	$\left\{ \begin{matrix} 0V_{13} = 0V_{24} \\ 0V_{24} = 1V_{24} \end{matrix} \right\}$	= B ₇	...	...	...																				
25	+	1V ₁ + 1V ₃	1V ₃	$\left\{ \begin{matrix} 1V_1 = 1V_1 \\ 1V_3 = 1V_3 \end{matrix} \right\}$	= n + 1 = 4 + 1 = 5	1	...	n + 1				0																

$$S_m(n) = \sum_{k=1}^n k^m = 1^m + 2^m + \cdots + n^m.$$

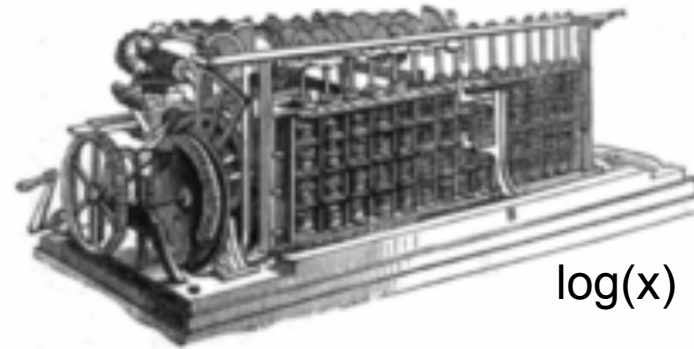


Den mekaniska eran: 1623-1945

Sverige!!!



Pehr (Per) Georg Scheutz (1785 –1873)

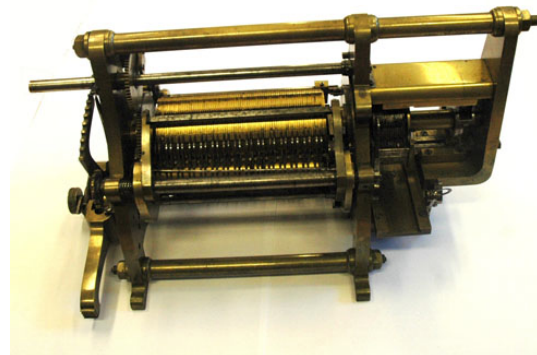


$\log(x)$

-53 -55F, -59GB & -60USA



Martin Wiberg (1826 – 1905)

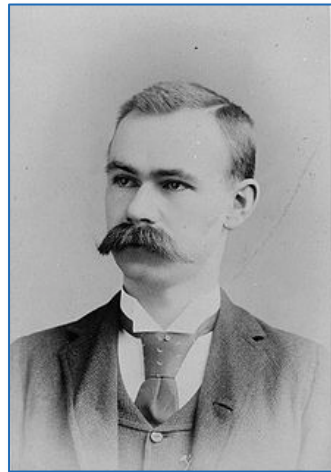


-75 $\log(x)$

(En 110 år lång parentes)

Folkräkning i USA 1880 – Tog 8 år att genomföra...

Ny folkräkning 1890, drastiskt större befolkning...



Herman Hollerith
(1860 –1929)

1	1	3	0	2	4	10	On	S	A	C	E	a	c	e	g		EB	SB	Ch	Sy	U	Sh	Hk	Br	Rm
2	2	4	1	3	E	15	Off	IS	B	D	F	b	d	f	h		SY	X	Fp	Cn	R	X	Al	Cg	Kg
3	0	0	0	0	W	20		0	0	0	0	0	0	0	0	0	●	0	0	0	0	0	0	0	0
A	1	1	1	1	0	25	A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B	2	2	2	2	5	30	B	2	2	●	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
C	3	3	3	3	0	3	C	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
D	4	4	4	4	1	4	D	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
E	5	5	5	5	2	C	E	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
F	6	6	6	6	A	D	F	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
G	7	7	7	7	B	E	G	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
H	8	8	8	8	a	F	H	8	8	8	8	8	8	8	8	8	●	8	8	8	8	8	8	8	8
I	9	9	9	9	b	I	I	9	9	9	9	9	9	9	9	9	●	9	9	9	9	9	9	9	9



1ÅR!





Den mekaniska eran: 1623-1945

Harvard Mark I 1939-44, -59

Basen

10

4,5 ton

2.4*0.6*16m

3.500 Reläer

2.225 Räkneverk

1.464 Brytare

72st, 23 siffriga mekaniska ackumulatorer

800 km kabel

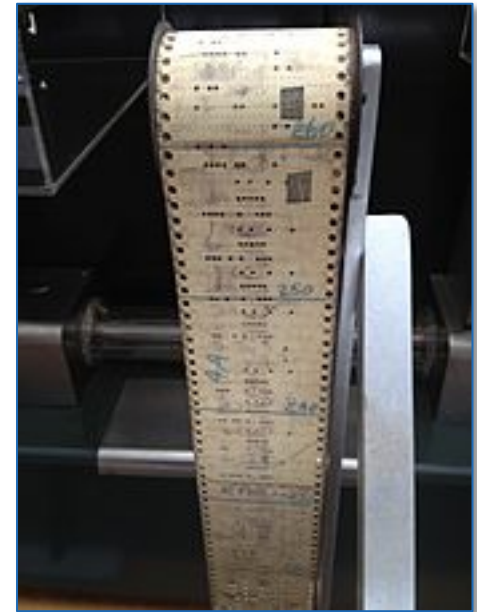
3.000.000 Lödpunkter

Mekaniken drevs av en 4kW elmotor!

MTBF "bra"

>10 instruktioner/sek (+/-)

Felfri!



Howard Aiken (1900 – 1973)

(Atombomben)



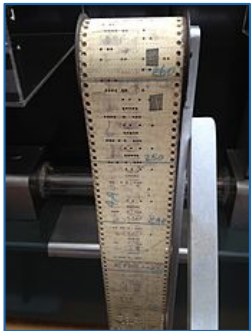
Den mekaniska eran: 1623-1945

Harvard Mark I 1939-44, -59



Med MARK I uppstod "på riktigt" yrket programmerare men också en otroligt viktig princip: **Abstraktion**. Programmeraren behövde inte förstå hur den otroligt komplicerade maskinen var byggd, utan bara vad den kunde utföra, i form av dess instruktionsrepertoar!

Instruktionsrepertoaren på den här nivån kallas **maskinkod** och är ytterst **unik för varje familj av datorer**. En maskininstruktion är alltid ett unikt mönster av vanligtvis 4 till 64 bitar (MARK I: 24 bitar)



En instruktion



Abstraktion används överallt inom data- och telekommunikation för att dölja bort komplexitet genom att i stället erbjuda **enkla och väl beskrivna gränssnitt!**

Kvinnan på bilden är Grace Hopper, även känd som "Amazing Grace", hon var tillsammans med Richard Milton Bloch och Robert Campbell MARK I:s första programmerare.



Harvard Mark I 1939-44, -59

Från Grace loggbok 450909...

92

9/9

0800 Antan started
 1000 " stopped - antan ✓ { 1.2700 9.037 847 025
 1300 (032) MP-MC 2.130476415 9.037 846 995 concl
 (032) PRO. 2 2.130476415
 concl 2.130676415
 Relays 6-2 in 033 failed special speed test
 in relay " 11.00 test.
 Relays changed
 1100 Started Cosine Tape (Sine check)
 1525 Started Multi-Adder Test.
 1545  Relay #70 Panel F
 (moth) in relay.
 First actual case of bug being found.
 1630 Antan started.
 1700 closed down.

Photo # NH 96566-KN First Computer "Bug", 1945

...när hon plockat bort den döda insekten...

...satt hon upp en lapp att maskinen var sanerad (debugged)!



Den mekaniska eran: 1623-1945

Så här långt...

Väsen

Homo
Sapiens

1834
Babbage
Analytical
Engine

1944
Aiken
Harvard
Mark I

Hårdvara



Kugghjul
& reläer **10**

Programmering



Hållremsa

Programhantering

NEJ

NEJ

(Att byta program
innebar att någon
för hand bytte det
band av hålkort
som var önskat
program!)

NEJ

Data

0.1 inst/s
Felben.

1 inst/s
"Felfri"
(Slitage)

10 inst/s
Felfri

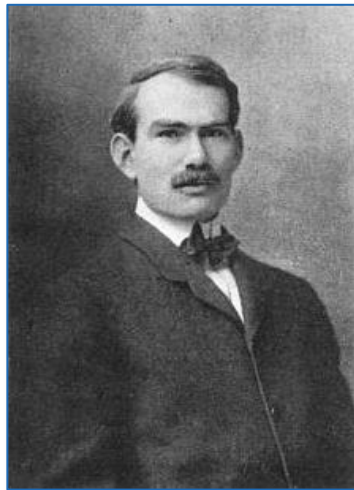


Den första generationen: 1946-1954

Krux & Innovationer...

Problem med de mekaniska "datorerna":

- Beräkningshastigheten begränsas av mekaniska krav.
- Mekaniken är inte helt tillförlitlig.



Lee de Forest (1873 – 1961)



Triod
1906



Den första generationen: 1946-1954

ENIAC 1943-46--55

Basen

10

30 ton

2.4*0.9*30m

17.468 Rör

7.200 Dioder

1.500 Reläer

70.000 Resistanser

10.000 Kapacitanser

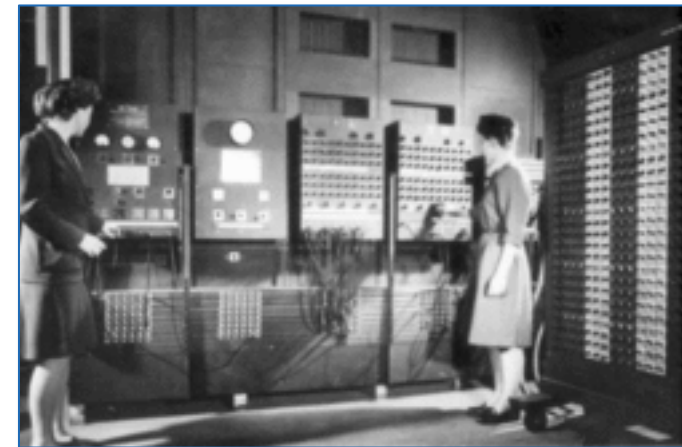
5.000.000 Lödpunkter

150.000 Watt!

MTBF 6-60 timmar

1000 instruktioner/sek

Felfri!



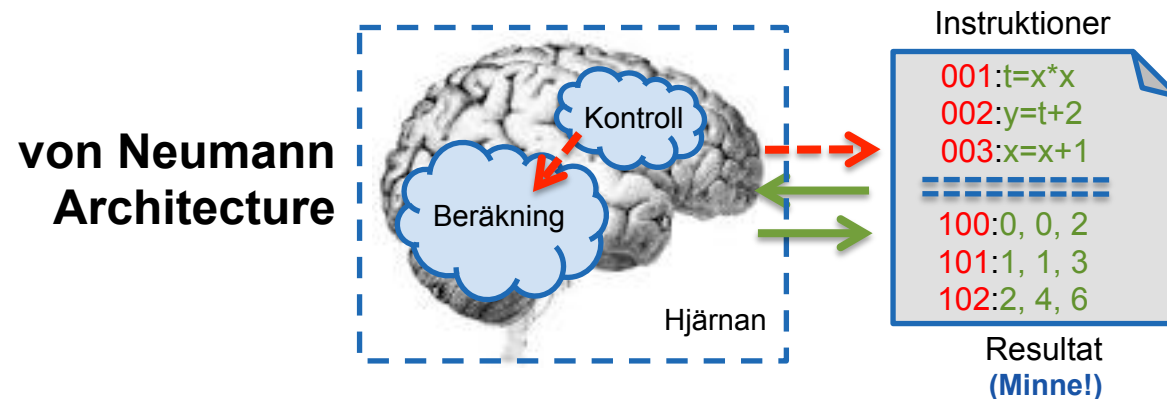
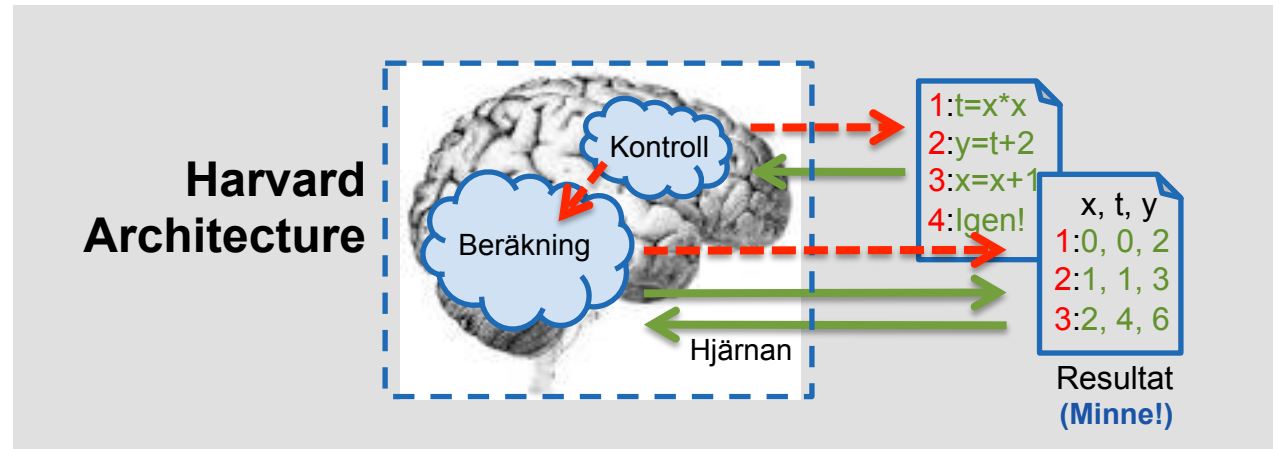
(Artilleritabeller, Vätebomben)

Två nya koncept >>>

1: Stored program concept



John von Neumann
(1903 –1957)



Instruktioner och data i samma adressrymd!



2: Binära talsystemet

Decimala talsystemet...

Positionssystem med 10 symboler (0..9)

10^3	10^2	10^1	10^0	
Tusental	Hundratal	Tiotal	Ental	
5	4	2	3	$=5*1000+4*100+2*10+3*1=5423_{10}$

Binära talsystemet...

Positionssystem med 2 symboler (0..1)

2^3	2^2	2^1	2^0	
Åttatal	Fyrtal	Tvåtal	Ental	
1	0	1	0	$=1*8+0*4+1*2+0*1=10_{10}$



2: Binära talsystemet...

$$7 + 6 = ?$$

$$7_{10} = 4 + 2 + 1 = 1\ 1\ 1_2$$

$$6_{10} = 4 + 2 + 0 = 1\ 1\ 0_2$$

$$\begin{array}{r} 111 \\ +110 \\ \hline ??? \end{array}$$

Ok, vi kan skriva tal binärt, men går det att räkna binärt?!



2: Binära talsystemet...

$$7 + 6 = ?$$

$$7_{10} = 4 + 2 + 1 = 1\ 1\ 1_2$$

$$6_{10} = 4 + 2 + 0 = 1\ 1\ 0_2$$

111	111
<u>+110</u>	<u>+110</u>
???	??1

$$0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = ?$$



2: Binära talsystemet...

$$7 + 6 = ?$$

$$7_{10} = 4 + 2 + 1 = 1\ 1\ 1_2$$

$$6_{10} = 4 + 2 + 0 = 1\ 1\ 0_2$$

111	111	<u>1</u> 111
+110	+110	+110
???	??1	?01

1
283
+334
617

$0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, $1 + 1 = (2) = 10 = 0 + "1 \text{ i minne}"$!



2: Binära talsystemet...

$$7 + 6 = ?$$

$$7_{10} = 4 + 2 + 1 = 1\ 1\ 1_2$$

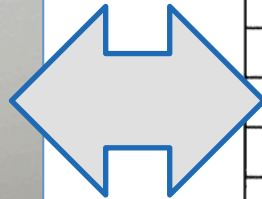
$$6_{10} = 4 + 2 + 0 = 1\ 1\ 0_2$$

**There are only 10 types
of people in the world:
Those who understand binary
and those who don't**

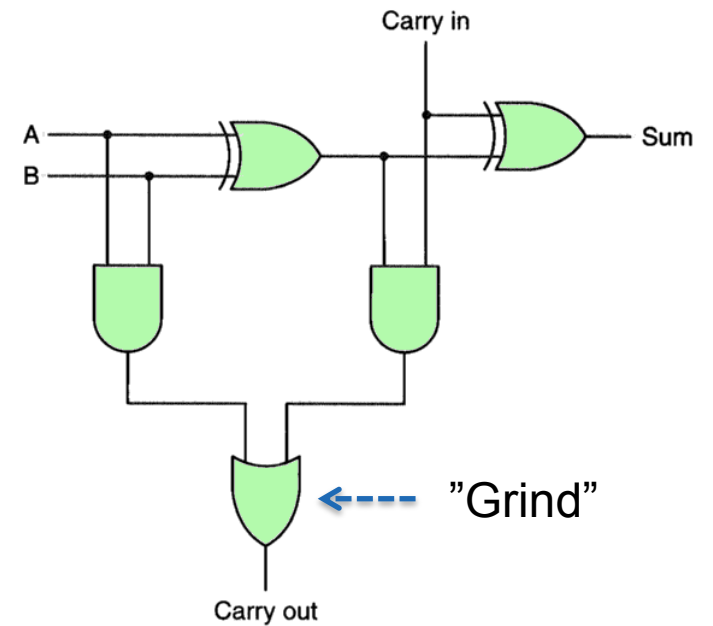
		<u>1</u>	<u>11</u>	
111	111	111	111	
<u>+110</u>	<u>+110</u>	<u>+110</u>	<u>+110</u>	
???	??1	?01	1101	$= 8 + 4 + 0 + 1 = 13_{10}$

$1 + 1 + 1 = (3) = 11 = 1 + \text{"1 i minne"}!$ Wow, det går att räkna!

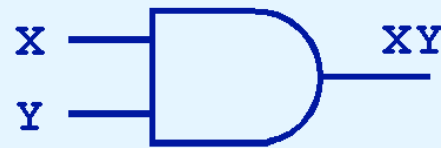
2: Adderare...



A	B	Carry in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

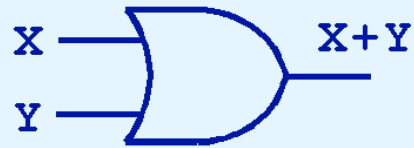


2: Grindar realiserade med rör...



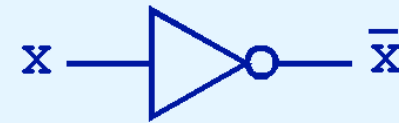
X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1



X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1



NOT X

X	$\bar{X}$
0	1
1	0

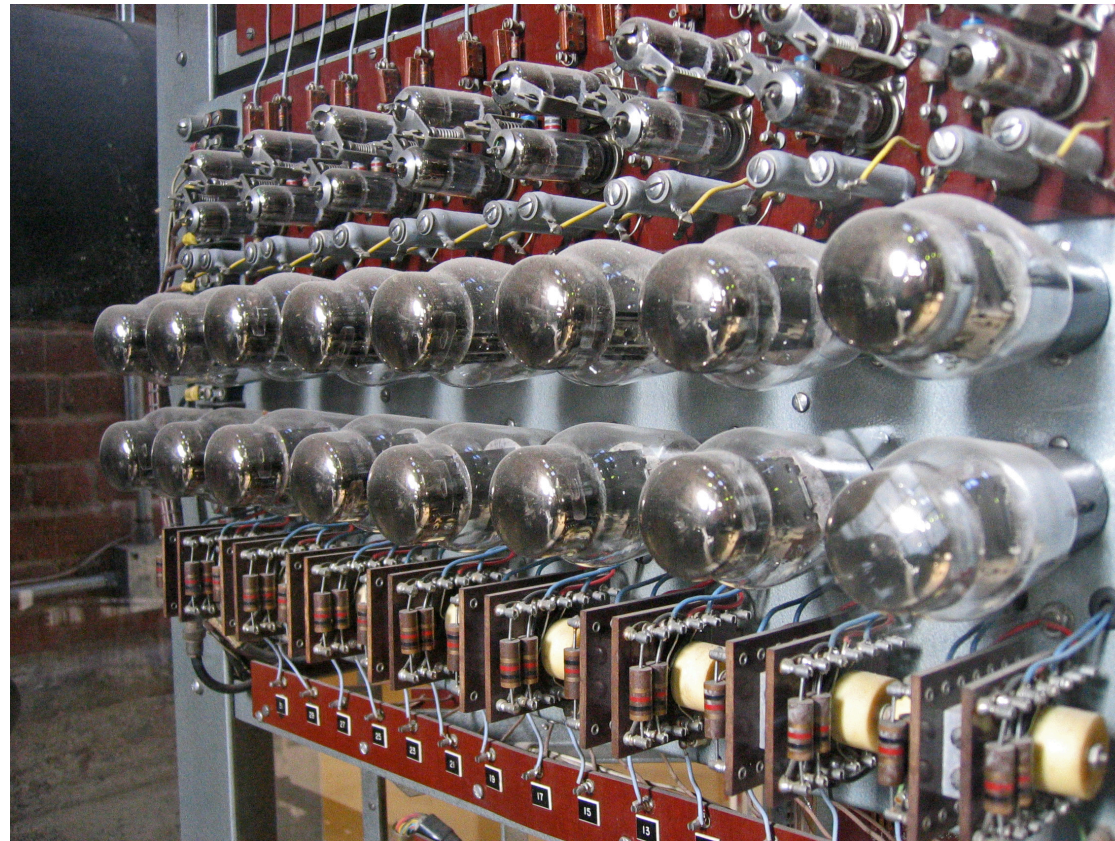
IBM
NOT
Gate



2: "Allt" kunde nu byggas av rör...

Minne

Register
Processor
Aritmetik



(En enda grind kostade
i början ca 5000SEK!)

...i stället för kugghjul!



...men observera:

Programmet skulle fortfarande kunna vara detsamma!

Maskinkoden döljer bort hur den underliggande maskinen är konstruerad, i detta fall är den baserad på elektriska rör i stället för kugghjul så instruktionerna utförs ca 1000ggr snabbare!

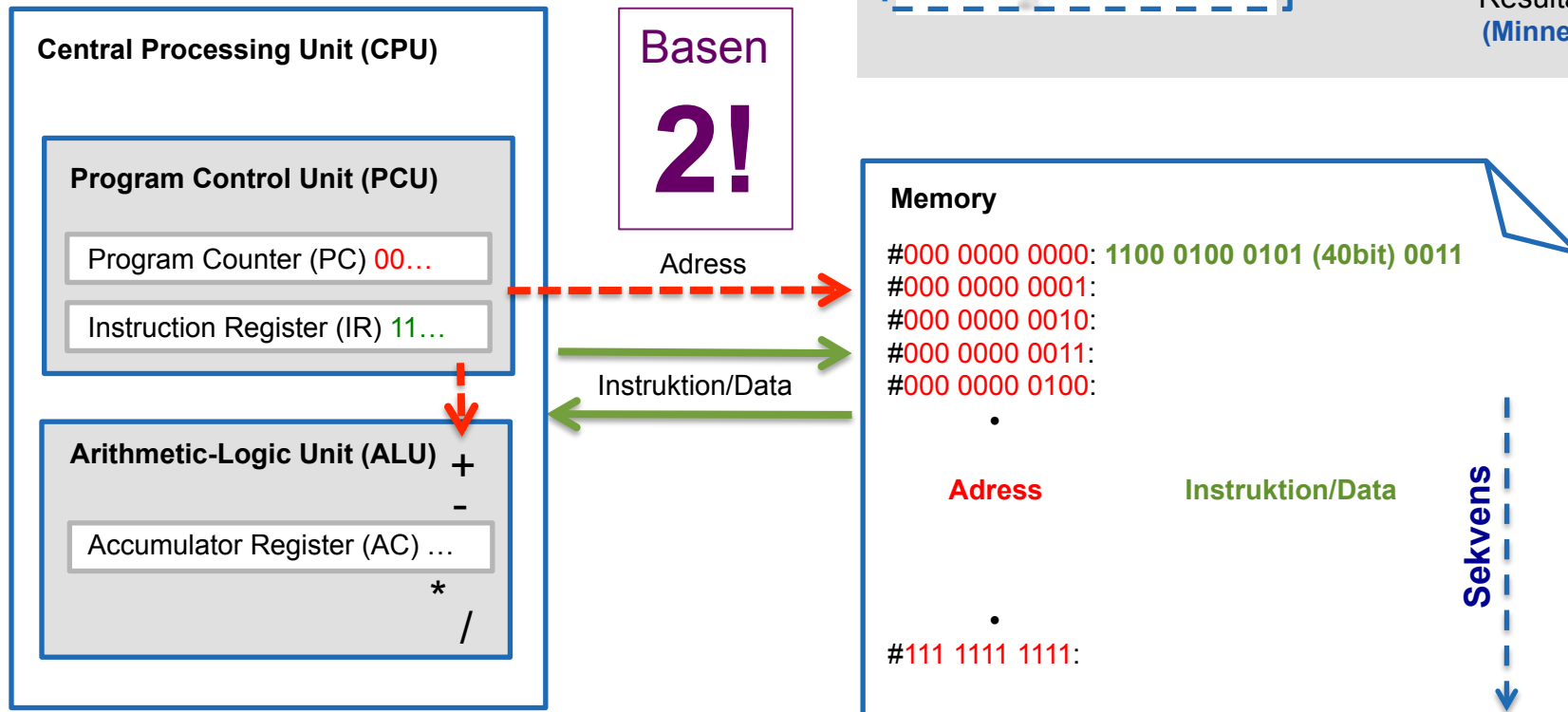
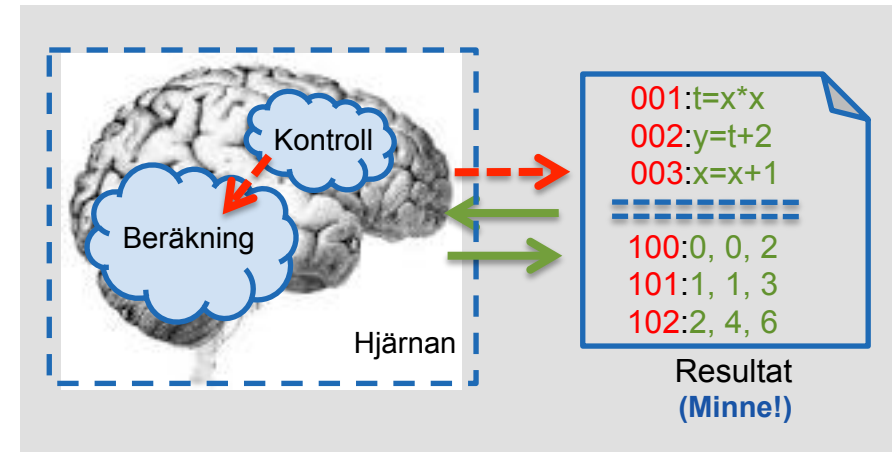
Detta tack vara att maskinkoden erbjuder en **abstrakt** bild av den underliggande datorn

Instruktionerna är fortfarande + - * / samt Load och Store! Utförda på samma sätt som om du skulle ha utfört beräkningarna!





-46..49--61
EDVAC & IAS
1K,44Bit 4K,40Bit



CPU, ALU, PC, Adress, basen 2/16, IR, AC, Sekvens



Gränssnitt...

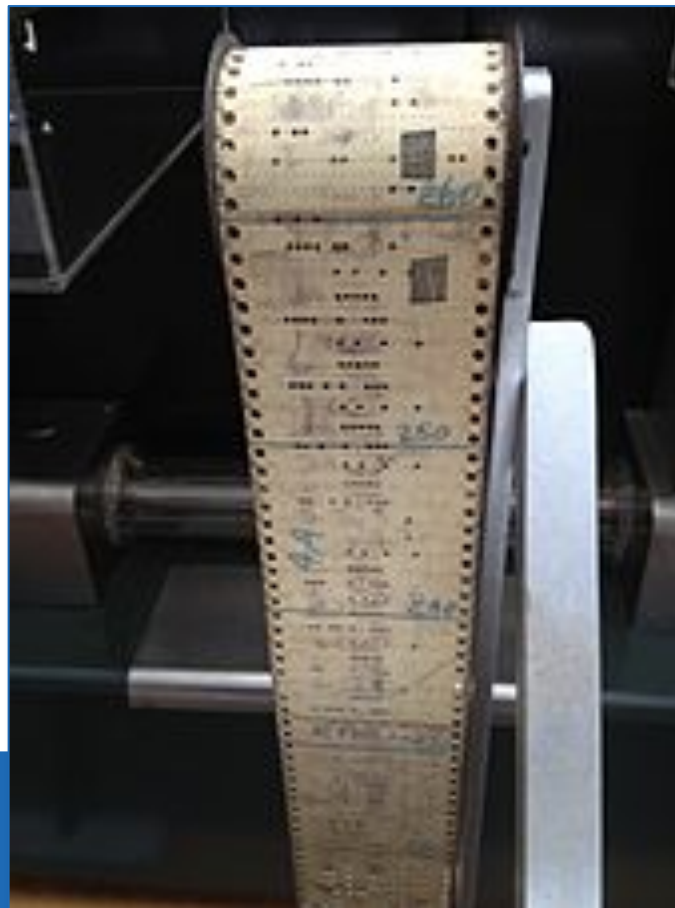


"A = A + 1"

1001 0001

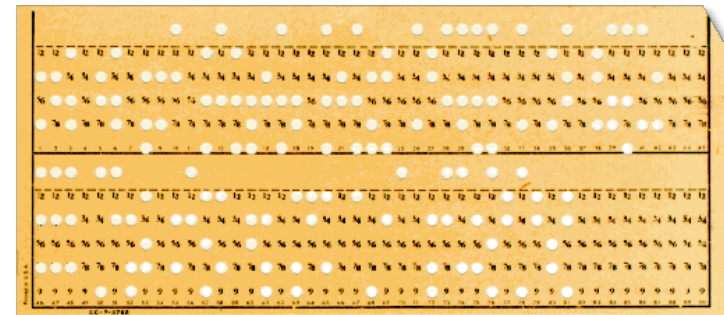


1001 0001 0000 0000 0000 0001



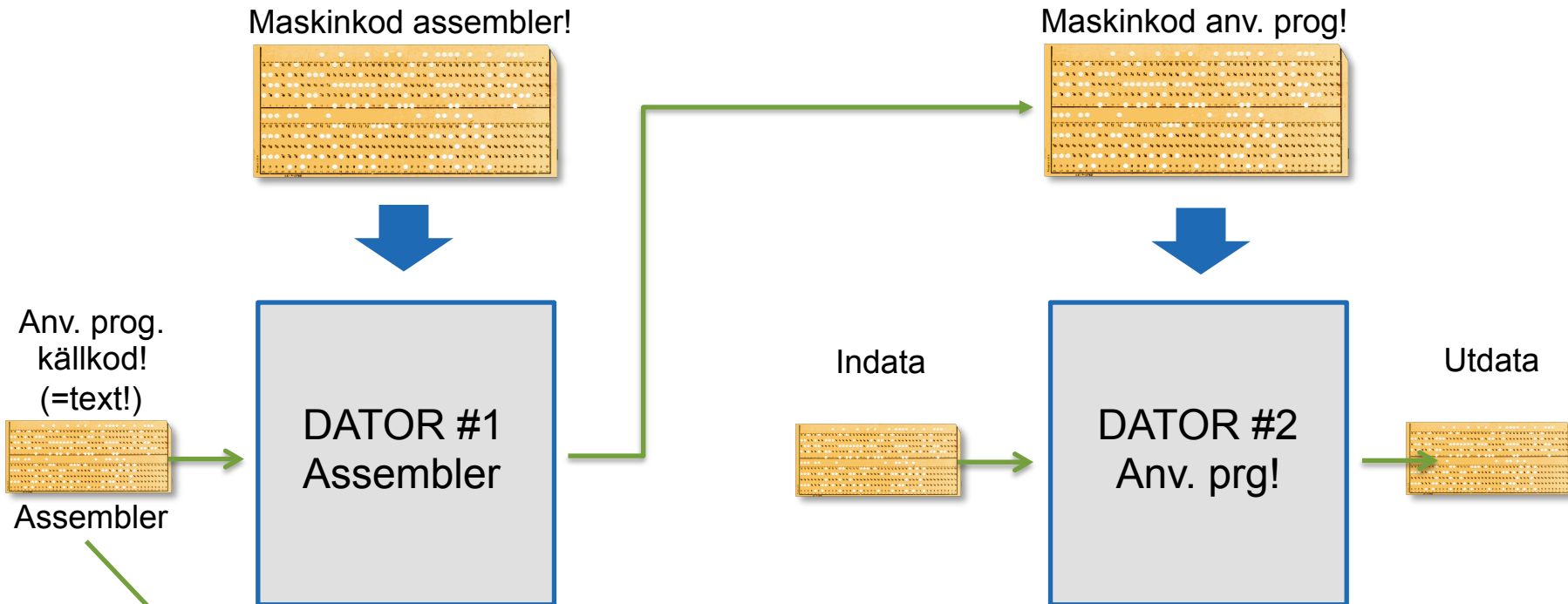
1001 0001 0000 0000 0001

(Maskinkod)



...rätt tröttsamt & felbenäget...

Symbolic programming, assembler...



**Alla maskininstruktioner döps till ett kort namn.
Alla typer av konstanter kan ges förklarande namn.**





Exempel: Microchip assembler (IoT)

```
#include "p18F4520.inc"

CONFIG OSC = HS           ; Oscillator Selection bits
CONFIG PWRT = OFF         ; Power-up Timer Enable bit
CONFIG WDT = OFF          ; Watchdog Timer Enable bit
CONFIG PBAEN = OFF        ; PORTB A/D Enable bit
CONFIG LVP = OFF          ; Single-Supply ICSP Enable bit

    org 0x000000
PORST GOTO INIT

    org 0x000020
;*****
;*** LCD toolbox 1.0 AC *****[ BLOCKING VERSION ]*****
;*** Needs FSR1 PORTD (short wr), RB1 EN & RB3 RS *****
;*** (RS = 1 DATA, RS = 0 INSTRUCTION) *****
;*** 0xFF=\n, 0xFE=\cmd, 0xFD=%v *****
;*** Dependesis: ED2SHORT, EDLONG *****
;*****
ELINIT CLRFB TRISD           ; PDX out: LCD data bus.
        BCF  TRISB,RB1       ; PB1 out: LCD enable.
        BCF  TRISB,RB3       ; PB3 out: LCD reg. select.
        BCF  PORTB,RB1       ; Ensure enable low.
        RETURN

ELWRCMD BCF  PORTB,RB3       ; Inst write in progress...
        MOVWF PORTD          ; ...the inst is...
        BSF  PORTB,RB1       ; ...enable wr...
        BCF  PORTB,RB1       ; ...wr done!
        CALL EDLONG          ; Enforce min delay
        RETURN
```

Notera förhållandet 1:1

CLRFB = CLeaR File
0 1 1 0 1 0 1 0
TRISD = File 0xF95
1 0 0 1 0 1 0 1

HE1028 Mikrodator teknik

Address	00	02	04	06	08	0A	0C	0E
0000	EF4D	F000	0000	FFFF	FFFF	FFFF	FFFF	FFFF
0010	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
0020	6A95	9293	9693	9281	0012	9681	6E83	8281
0030	9281	EC47	F000	0012	8681	6E83	8281	9281
0040	EC43	F000	0012	EC1C	F000	0009	50F5	4AE8



Utvik: RAM/ROM & basen 16

Hur vet vi vad det binära talet i en minnescell betyder?

0000 0000
0000 0001
0000 0010
0000 0011
0000 0100

*

0001 0000
0001 0001
0001 0010
0001 0011
0001 0100

1110 1111
0100 1101
1111 0000
0000 0000
1111 1111

*

0000 0000
1111 1111
1111 1111
0100 0001
0100 0010



Utvik: RAM/ROM & basen 16

Hur vet vi vad det binära talet i en minnescell betyder?

0000 0000
0000 0001
0000 0010
0000 0011
0000 0100

*

0001 0000
0001 0001
0001 0010
0001 0011
0001 0100

1110 1111
0100 1101
1111 0000
0000 0000
1111 1111

*

0000 0000
1111 1111
1111 1111
0100 0001
0100 0010

Instruktion: GOTO 32₁₀

Konstant: 0

Konstant: 255

Konstant: -1 (2-komplement)

Bokstav: 'A'

Bokstav: 'B'

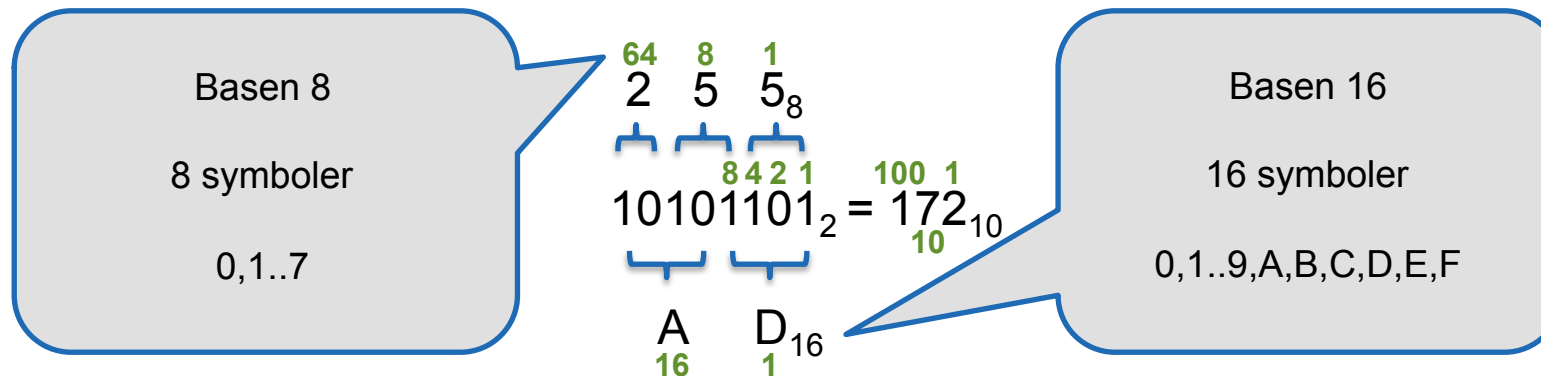
Attans, det måste vi göra själva!
(Hur lagras ett stort tal?)



Utvik: RAM/ROM & basen 16

Alla datorminnen fungerar på så sätt att det går att ange en valfri adress...
...och går det bara att läsa innehållet så är det ett Read Only Memory (**ROM**)
...går det att välja mellan att läsa eller skriva så är det ett Read Write Memory (RWM)...
av historiska skäl så kallas RWM i dag alltid Random Access Memory (**RAM**)!

Det är otroligt tröttsamt att skriva stora binära tal...
...därför användes basen 8, och numer basen 16 i stället!





Dec	Bin	Hex	Oct
0	0000 0000	00	000
1	0000 0001	01	001
2	0000 0010	02	002
3	0000 0011	03	003
4	0000 0100	04	004
5	0000 0101	05	005
6	0000 0110	06	006
7	0000 0111	07	007
8	0000 1000	08	010
9	0000 1001	09	011
10	0000 1010	0A	012
11	0000 1011	0B	013
12	0000 1100	0C	014
13	0000 1101	0D	015
14	0000 1110	0E	016
15	0000 1111	0F	017
16	0001 0000	10	020

10/1

8/4/2/1

16/1

64/8/1



Utvik: RAM/ROM & basen 16

NÄR ÄR $\frac{12}{7} = 2$ SANT?



Utvik: RAM/ROM & basen 16

NÄR ÄR $\frac{12}{7} = 2$ SANT?

I basen 12! $1 \cdot 12\text{-tal} + 2 \cdot 1\text{-tal} = 14$, $14/7 = 2!!!$



Så här långt...

Väsen

Homo
Sapiens

1834

Babbage
Analytical
Engine

1944

Aiken
Harvard
Mark I

-46 ENIAC
-49 EDVAC
-51 IAS

Hårdvara



Kugghjul
& reläer 10

Kugghjul
& reläer 10

Rör! 10
Rör! 2!
Rör! 2!

Programmering



Maskinkod
/Hållremsa

Maskinkod
/Hållremsa

Maskinkod
Assembler
Assembler

Programhantering

NEJ

NEJ

NEJ

NEJ

NEJ

NEJ

Data

0.1 inst/s
Felben.

1 inst/s
"Felfri"
(Slitage)

10 inst/s
Felfri

1000Hz/20
1000Hz/1k
1000Hz/4k



Vad vi lärt oss så här långt...

SW



PROGRAM
KOMPILATOR
OPERATIVSYSTEM

Källkod

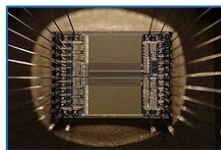
Assemblerkod

ASSEMBLER

Maskinkod



HW



**Hårdvara – vad kan en dator egentligen göra?!
,och hur?!**



Rev history

150206 0.1 AC	Skapad.
150305 0.5 AC	Utkast #1, remiss NB.
150429 0.8 AC	Utkast #2, remissrunda.
150507 0.9 AC	Återkoppling inkluderad, nytt labbintro sist.
150817 1.0 AC	Inkluderat övning.
160608 1.1 AC	Utökat avsnittet om talsystem.
160614 1.2 AC	Inkluderat abstraktion, flyttat modern historia till föreläsning #2.
160701 2.0 AC	Finputs HT16.
170726 2.1 AC	Mindre förbättringar HT17 (binära tal).