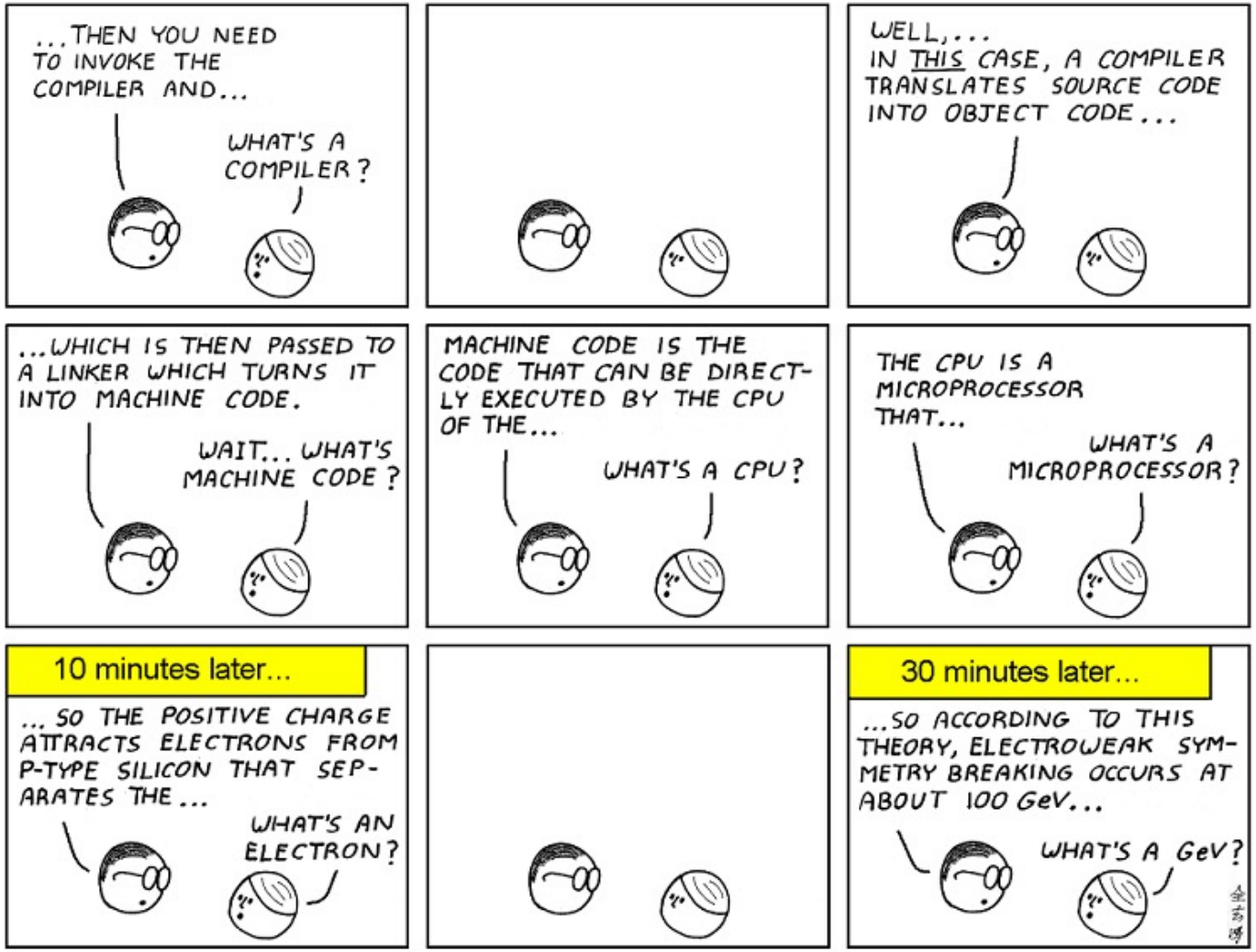
Computer Science & Engineering Department

Calculatoare Numerice

- Cursul 0 -

Facultatea de Automatică și Calculatoare
Universitatea Politehnica București

Comic of the day



http://abstrusegoose.com/98

Sper că sunteți aici din cauza asta

IOCLA

- Cum ajunge un program scris în asamblare să fie executat în logica digitală?
- Ce se întâmplă la mijloc?
- Cum putem să proiectăm un calculator pornind de la porți logice și conexiuni?

PL

"C/ASM" drept model de calcul

Perspectiva programatorului asupra felului în care funcționează un sistem de calcul

Perspectiva Architectului de Sistem:
Cum să proiectez un sistem de calcul care să satisfacă cerințele de design
Alegerile afectează atât inginerul software cât și inginerul hardware

Perspectiva inginerului HW asupra felului în care merge un calculator

Logica digitală drept model de calcul

About me

Dan Ștefan Tudose

- dan.tudose@upb.ro
- Office: ED422
- Office Hours: (almost) anytime on Teams
- https://ocw.cs.pub.ro/courses/iothings/dan.tudose
- Research & teaching:
  - Computer architecture, hardware/software interaction
  - Embedded and Pervasive Computing
  - Wireless Sensor Networks
  - Low Power Computing Architectures, Energy Harvesting
  - Fault tolerance
- Start-ups, Fitbit, Google

Echipa de Asistenți

- Cristina Buciu
- Adina Smeu
- Ștefan Dan Ciocîrlan
- Alexandru Pîrlea
- Bogdan Firuți
- Rareș Petruc
- George-Mircea Grosu

https://ocw.cs.pub.ro/courses/cn1

Important Stuff

Curs: Moodle

Laboratoare: https://ocw.cs.pub.ro/courses/cn1

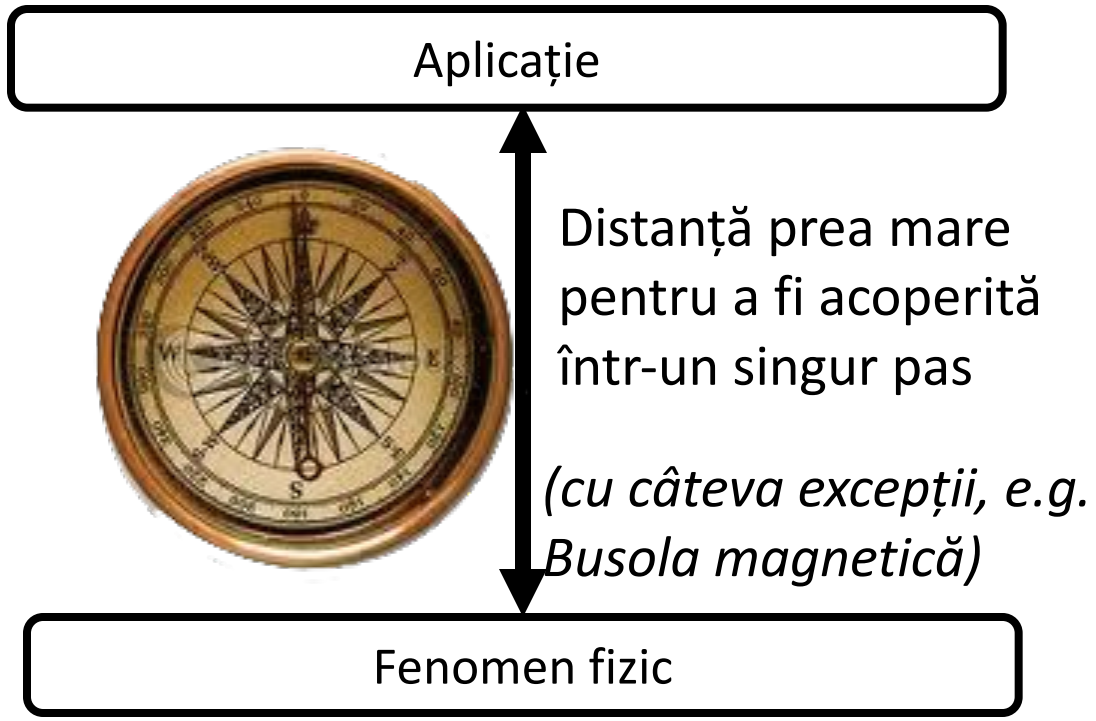
Notare:

- 3 puncte activitatea laborator
- 2 puncte lucrare laborator
- 5 puncte examen final
- BONUS pentru activitate la curs!!!

Cerințe minime pentru a promova:

- minim 6 prezențe la laborator
- minim 1.5 puncte din cele 3 puncte pentru activitatea laborator
- minim 2.5 puncte din cele 5 puncte din examenul final

Ce este o arhitectură de calcul?



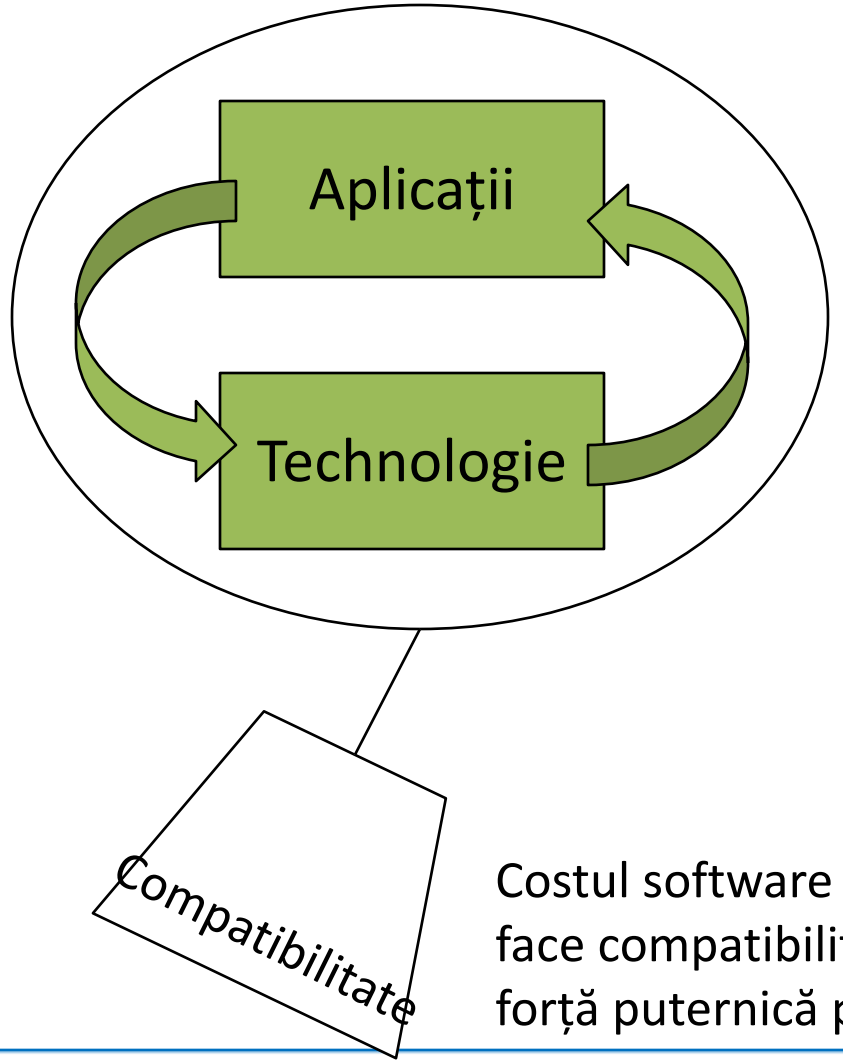
Definiție generală: o arhitectura de calcul presupune design-ul unor niveluri succesive de abstractizare ce ne permit implementarea eficientă a unei aplicații de procesare de informație, în limitele tehnologiilor curente de fabricație.

Niveluri de abstractizare într-un sistem de calcul modern

- Aplicație
- Algoritm
- Limbaj de programare
- Sistem de operare/Mașină virtuală
- Instruction Set Architecture (ISA)
- Microarhitectură
- Porți logice/Transfer de registre
- Circuite electronice
- Dispozitive și componente
- Fenomen fizic

Arhitectura este într-o continuă schimbare

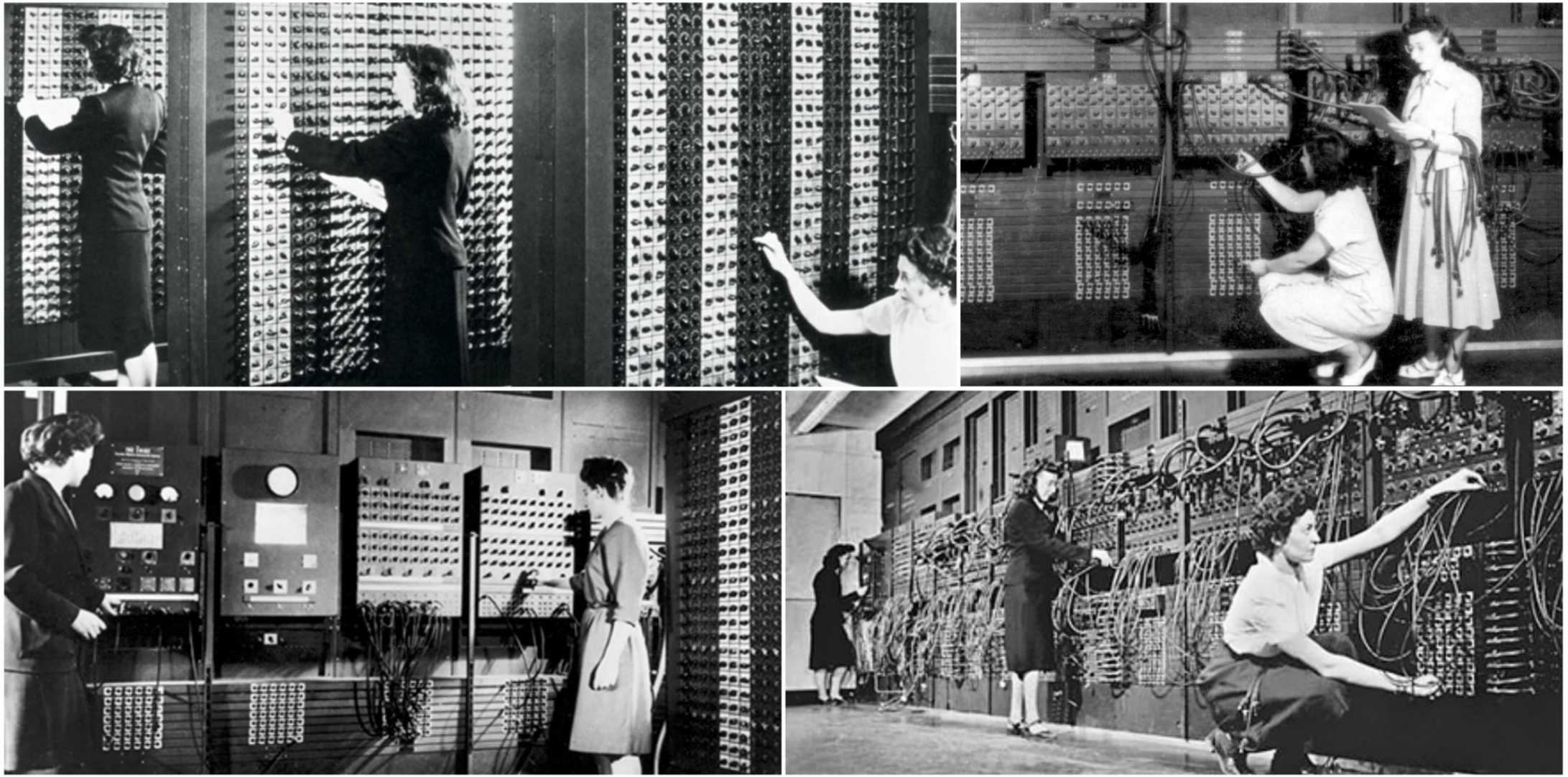
Aplicațiile dau indicii asupra modului în care poate fi îmbunătățită tehnologia de fabricație și produc venit pentru a finanța dezvoltarea.



Îmbunătățirea tehnologiei de calcul face posibilă dezvoltarea de noi aplicații

Costul software development face compatibilitatea să fie o forță puternică pe piață

Sistemele de calcul atunci...



ENIAC, University of Pennsylvania, USA, 1946

Sisteme de calcul contemporane

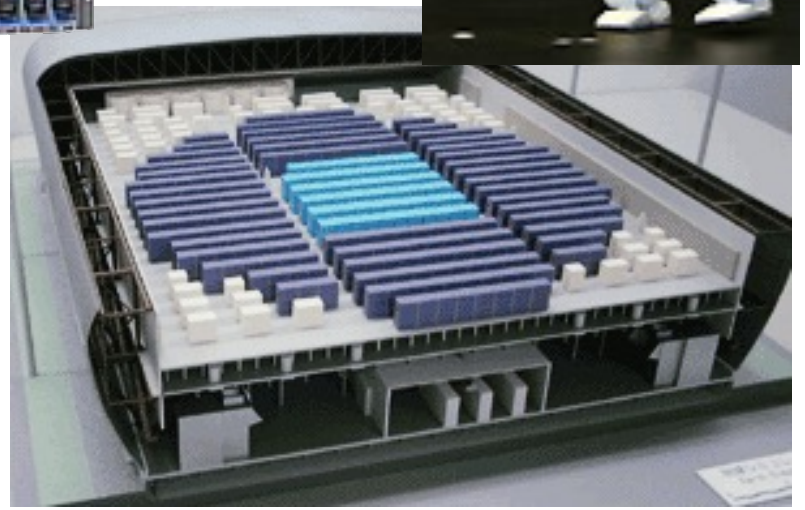


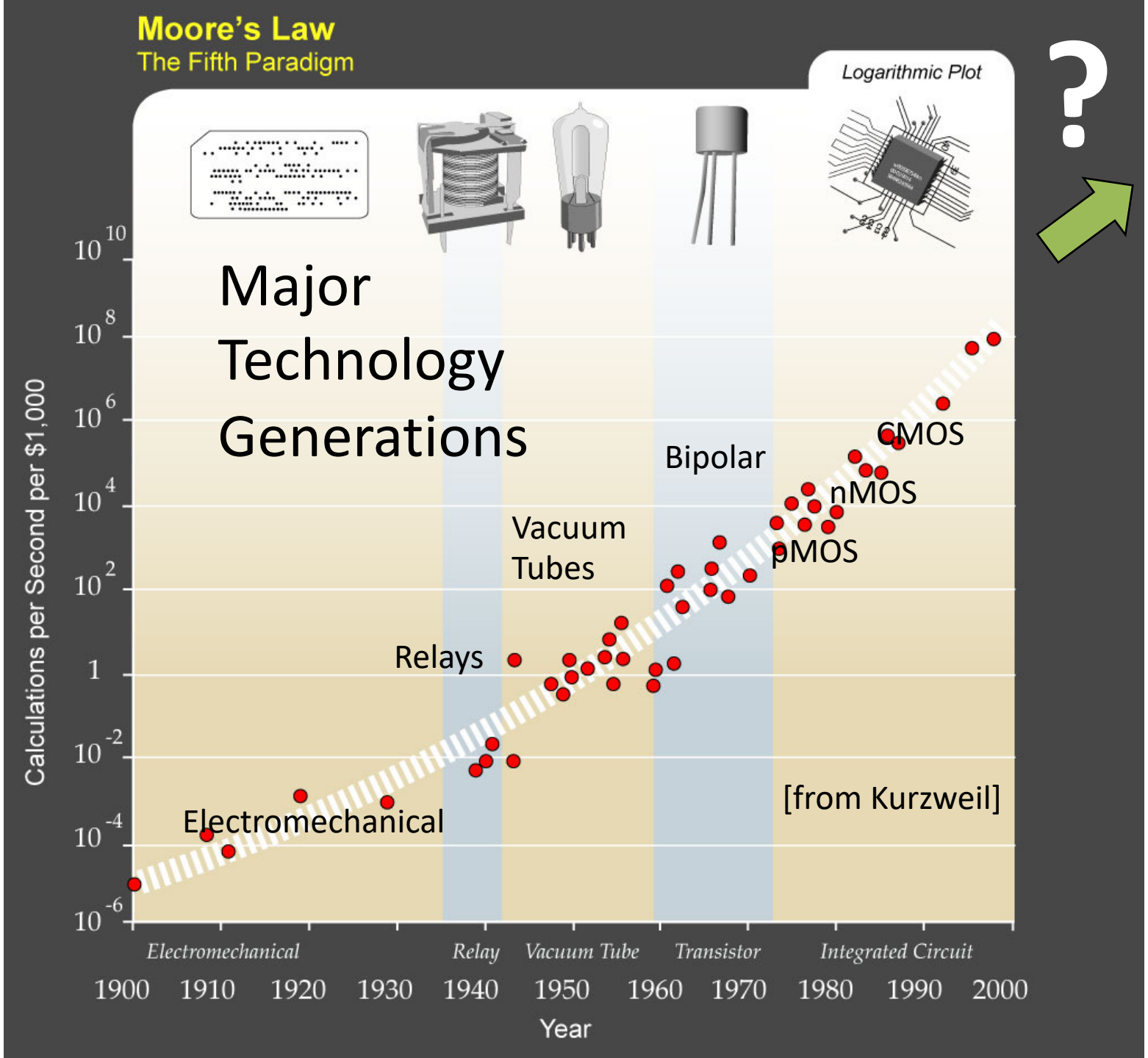








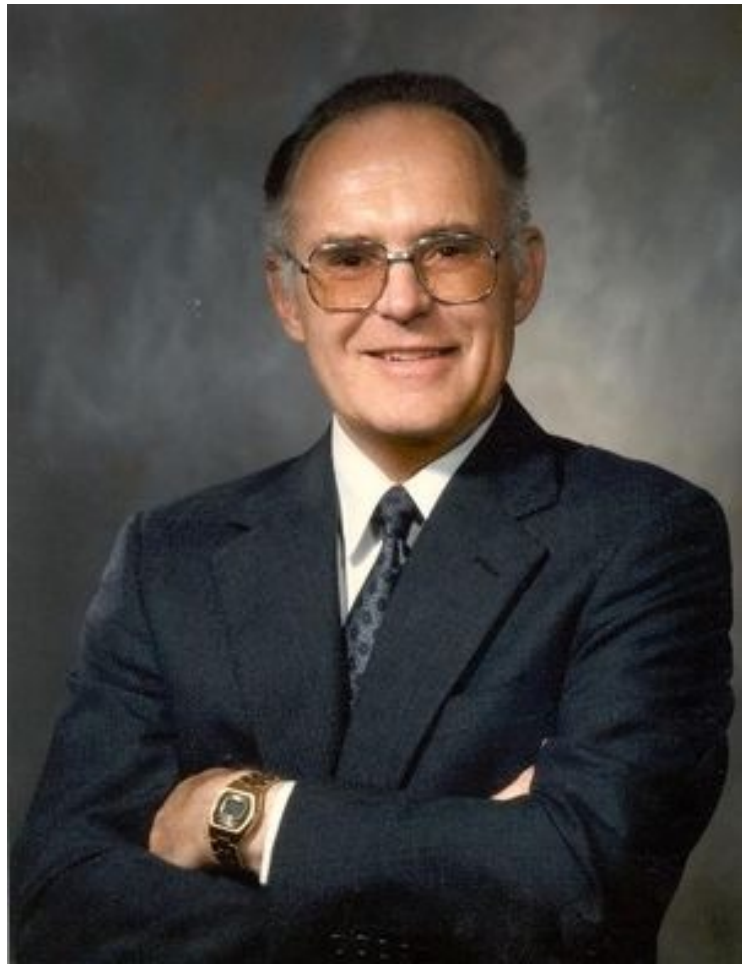



Moore's Law
The Fifth Paradigm
Logarithmic Plot
?
Major Technology Generations
Calculations per Second per $1,000
10^{10}
10^{8}
10^{6}
10^{4}
10^{2}
1
10^{-2}
10^{-4}
10^{-6}
Electromechanical
Relays
Vacuum Tubes
Bipolar
nMOS
CMOS
pMOS
[from Kurzweil]
Electromechanical
Relay
Vacuum Tube
Transistor
Integrated Circuit
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
Year

Factorul decisiv în creșterea complexității

Legea lui Moore (1965)

- Numărul tranzistoarelor de pe 1cm pătrat
  - De două ori mai multe după ~1.5-2 ani
- Tendințe asociate
  - Performanța procesoarelor
    De două ori mai rapide după ~18 luni
  - Capacitatea memoriei
    Se dublează după < 2 ani

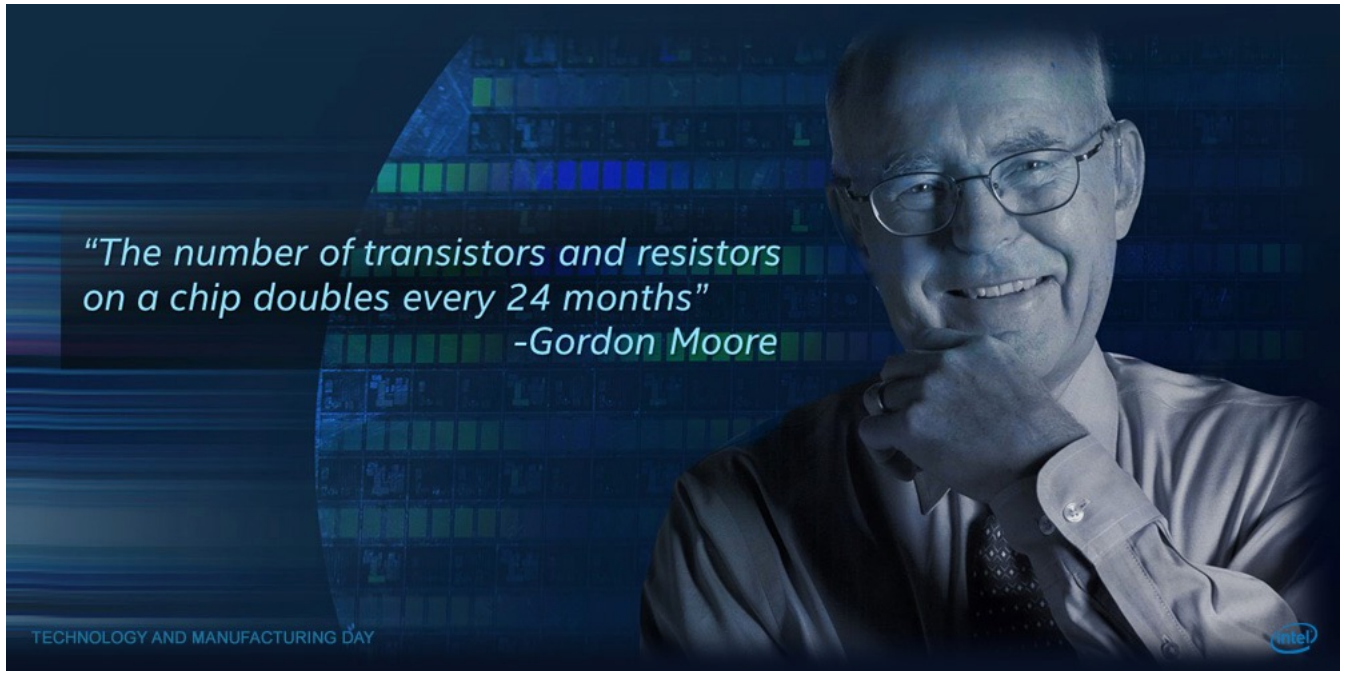


Gordon Moore

Moore's Not-Exactly-Law

Nu este o lege a naturii

- Dar aproximează destul de bine ce s-a întâmplat în industrie în ultimii 45 de ani



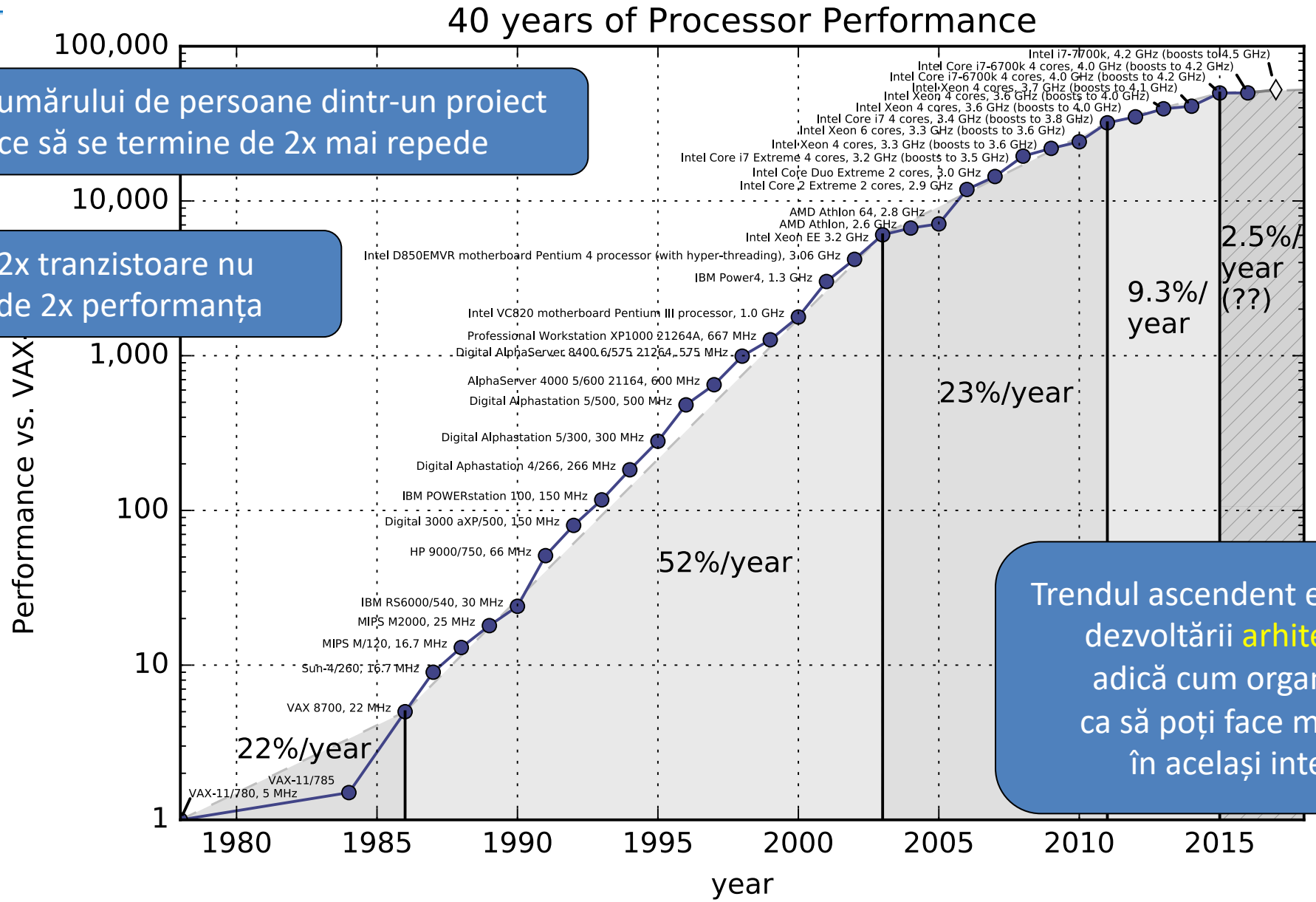
No exponential is forever, but we can delay "forever" (Gordon Moore in 2003)

Mai multe despre legea lui Moore aici

Evoluția Performanței

Dublarea numărului de persoane dintr-un proiect nu-l face să se termine de 2x mai repede

Similar, 2x tranzistoare nu măresc de 2x performanța



Trendul ascendent este posibil datorită dezvoltării arhitecturii de calcul, adică cum organizezi resursele ca să poți face mai mult/eficient în același interval de timp

Sfârșitul legii lui Moore?

- În ultimii 50 de ani Legea lui Moore a fost validă
  - Scalarea tehnologiei a permis îmbunătățiri majore de performanță/consum de energie fără schimbarea modelului software
- În ultimii 10 ani scalarea tehnologiei a încetinit/s-a oprit
  - Legea lui Dennard nu mai e validă (tensiune alimentare ~fixă)
  - Legea lui Moore (cost/tranzistor) nu mai e validă?
  - Nu avem (încă) un înlocuitor competitiv pentru CMOS
  - Eficiența energetică limitează totul
- No “free lunch” for software developers.

  Noi paradigme de programare
  - Sisteme de calcul paralel
  - Sisteme eterogene

Principalele sisteme programabile de azi

- **Mobile (smartphone/tablete)**
  - >1 milliard vândute în fiecare an
  - Piața dominată de procesoare ARM-ISA-compatible, general-purpose, system-on-a-chip (SoC)
  - Plus o adevărată explozie de acceleratoare (radio, image, video, graphics, audio, motion, location, security, etc.)
- **Warehouse-Scale Computers (WSCs)**
  - Sute de mii de nuclee pe warehouse
  - Piața dominată (încă) de procesoare x86-compatible
  - Aplicații specifice, stocare în cloud pe mașini virtuale
  - Folosire accentuată a GPU-urilor, FPGA-urilor sau hardware special pentru accelerarea execuției
- **Embedded computing**
  - Wired/wireless network infrastructure, printers
  - Consumer TV/Music/Games/Automotive/Camera/MP3
  - Internet of Things!

Arhitectura Calculatoarelor: Scurt Istoric

De ce ar trebui să ne preocupe așa ceva?

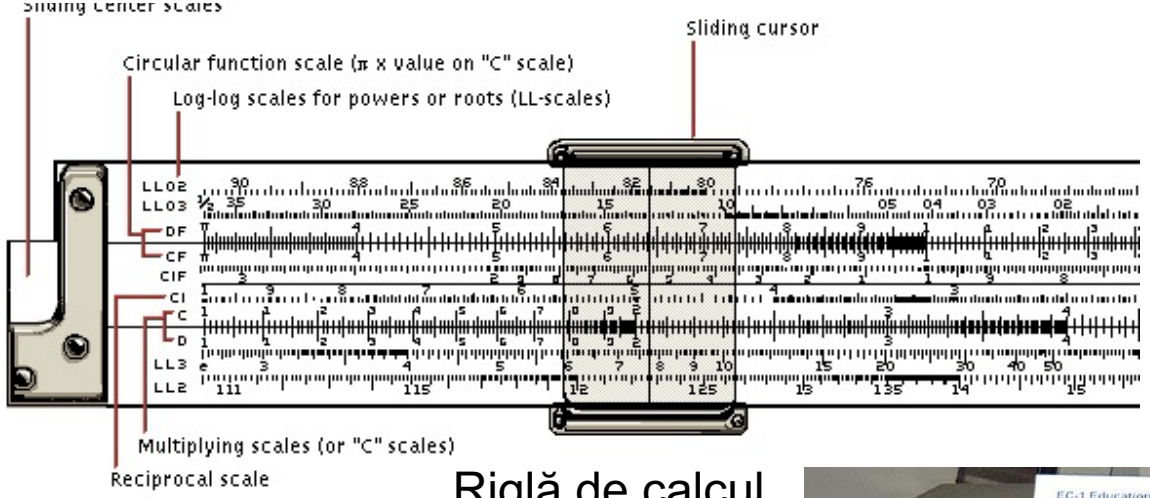
- Ajută în înțelegerea procesului de proiectare a structurilor de calcul și explică de ce au fost luate anumite decizii
- Pentru că tehnologiile viitoare ar putea fi la fel de supuse la constrângeri ca și cele trecute
- Cei care ignoră istoria sunt predispuși să o repete
  - Fiecare greșeală făcută în proiectarea de calculatoare mainframe a fost făcută și în proiectarea minicomputerelor și a microcomputerelor, apoi a telefoanelor smart și a dispozitivelor wearable.

Calculatoare analogice

Reprezintă variabilele prin o serie de cantități fizice (deplasare mecanică, presiune, tensiune etc.) și folosesc fenomene fizice pentru a calcula rezultatul.

[Marsyas, Creative Commons BY-SA 3.0]

Mecanismul Antikythera cca.100îHr



Riglă de calcul

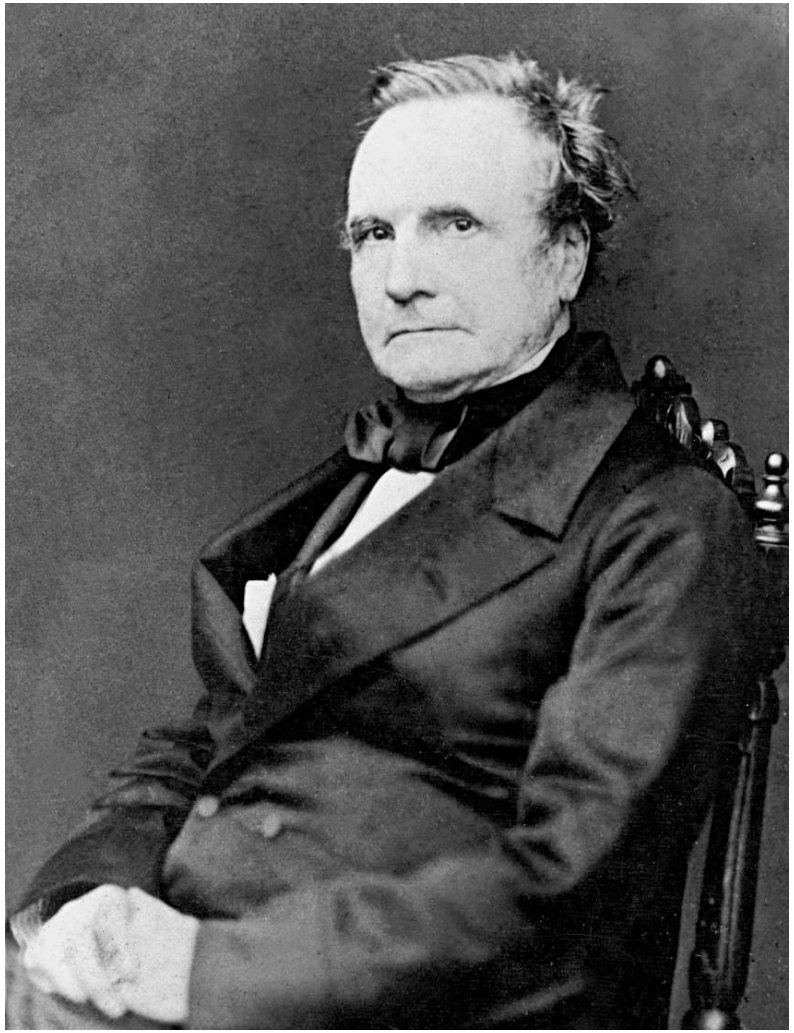
EC1 Electronic Analog Computer cca.1960

Calculatoare numerice (digitale)

- Reprezintă variabilele prin numere codificate folosind valori discrete
  - Valorile discrete dau imunitate la zgomot
- Permit calcule precise și deterministe
  - Aceleași intrări produc întotdeauna aceleași ieșiri
- Nu sunt constrânse de funcții implementabile fizic
- Calculatoarele programabile digitale sunt domeniul de interes al CN (și al CN2, PM, ASC, SI etc.)

Charles Babbage 1791-1871

- Lucasian Professor of Mathematics, Cambridge University, 1828-1839
- Un adevărat polimat, interesat de multe discipline și științe
- Nemulțumit de erorile tabelelor matematice ale vremii, voia să creeze mașini de calcul care să evalueze și să afișeze rezultatul oricărui calcul matematic
- Inspirat de munca calculatorilor umani, ce calculau metodic tabelele aritmetice de mână



x	0	1	2	3	4	5	6	7	8	9	Δ_m	1 2 3	4 5 6	7 8 9
											+	ADD		
10	·0000	0043	0086	0128	0170	0212					42	4 8 13	17 21 25	29 34 38
						0212	0253	0294	0334	0374	40	4 8 12	16 20 24	28 32 36
11	·0414	0453	0492	0531	0569	0607					39	4 8 12	16 19 23	27 31 35
						0607	0645	0682	0719	0755	37	4 7 11	15 19 22	26 30 33
12	·0792	0828	0864	0899	0934	0969					35	4 7 11	14 18 21	25 28 32
						0969	1004	1038	1072	1106	34	3 7 10	14 17 20	24 27 31
13	·1139	1173	1206	1239	1271	1303					33	3 7 10	13 16 20	23 26 30
						1303	1335	1367	1399	1430	32	3 6 10	13 16 19	22 26 29
14	·1461	1492	1523	1553	1584	1614	1644	1673	1703	1732	30	3 6 9	12 15 18	21 24 27
15	·1761	1790	1818	1847	1875	1903	1931	1959	1987	2014	28	3 6 8	11 14 17	20 22 25
16	·2041	2068	2095	2122	2148	2175	2201	2227	2253	2279	26	3 5 8	10 13 16	18 21 23
17	·2304	2330	2355	2380	2405	2430	2455	2480	2504	2529	25	2 5 7	10 12 15	17 20 22
18	·2553	2577	2601	2625	2648	2672	2695	2718	2742	2765	24	2 5 7	10 12 14	17 19 22
19	·2788	2810	2833	2856	2878	2900	2923	2945	2967	2989	22	2 4 7	9 11 13	15 18 20
20	·3010	3032	3054	3075	3096	3118	3139	3160	3181	3201	21	2 4 6	8 11 13	15 17 19
21	·3222	3243	3263	3284	3304	3324	3345	3365	3385	3404	20	2 4 6	8 10 12	14 16 18
22	·3424	3444	3464	3483	3502	3522	3541	3560	3579	3598	19	2 4 6	8 10 11	13 15 17
23	·3617	3636	3655	3674	3692	3711	3729	3747	3766	3784	18	2 4 5	7 9 11	13 14 16
24	·3802	3820	3838	3856	3874	3892	3909	3927	3945	3962	18	2 4 5	7 9 11	13 14 16
25	·3979	3997	4014	4031	4048	4065	4082	4099	4116	4133	17	2 3 5	7 9 10	12 14 15
26	·4150	4166	4183	4200	4216	4232	4249	4265	4281	4298	16	2 3 5	6 8 10	11 13 14
27	·4314	4330	4346	4362	4378	4393	4409	4425	4440	4456	16	2 3 5	6 8 10	11 13 14
28	·4472	4487	4502	4518	4533	4548	4564	4579	4594	4609	15	2 3 5	6 8 9	11 12 14
29	·4624	4639	4654	4669	4683	4698	4713	4728	4742	4757	15	1 3 4	6 7 9	10 12 13
30	·4771	4786	4800	4814	4829	4843	4857	4871	4886	4900	14	1 3 4	6 7 8	10 11 13
31	·4914	4928	4942	4955	4969	4983	4997	5011	5024	5038	14	1 3 4	6 7 8	10 11 13
32	·5051	5065	5079	5092	5105	5119	5132	5145	5159	5172	13	1 3 4	5 7 8	9 10 12
33	·5185	5198	5211	5224	5237	5250	5263	5276	5289	5302	13	1 3 4	5 6 8	9 10 12
34	·5315	5328	5340	5353	5366	5378	5391	5403	5416	5428	13	1 3 4	5 6 8	9 10 12
35	·5441	5453	5465	5478	5490	5502	5514	5527	5539	5551	12	1 2 4	5 6 7	8 10 11
36	·5563	5575	5587	5599	5611	5623	5635	5647	5658	5670	12	1 2 4	5 6 7	8 10 11
37	·5682	5694	5705	5717	5729	5740	5752	5763	5775	5786	12	1 2 4	5 6 7	8 10 11
38	·5798	5809	5821	5832	5843	5855	5866	5877	5888	5899	11	1 2 3	4 6 7	8 9 10
39	·5911	5922	5933	5944	5955	5966	5977	5988	5999	6010	11	1 2 3	4 6 7	8 9 10
40	·6021	6031	6042	6053	6064	6075	6085	6096	6107	6117	11	1 2 3	4 5 7	8 9 10
41	·6128	6138	6149	6160	6170	6180	6191	6201	6212	6222	10	1 2 3	4 5 6	7 8 9
42	·6232	6243	6253	6263	6274	6284	6294	6304	6314	6325	10	1 2 3	4 5 6	7 8 9
43	·6335	6345	6355	6365	6375	6385	6395	6405	6415	6425	10	1 2 3	4 5 6	7 8 9
44	·6435	6444	6454	6464	6474	6484	6493	6503	6513	6522	10	1 2 3	4 5 6	7 8 9
45	·6532	6542	6551	6561	6571	6580	6590	6599	6609	6618	10	1 2 3	4 5 6	7 8 9
46	·6628	6637	6646	6656	6665	6675	6684	6693	6702	6712	9	1 2 3	4 5 5	6 7 8
47	·6721	6730	6739	6749	6758	6767	6776	6785	6794	6803	9	1 2 3	4 5 5	6 7 8
48	·6812	6821	6830	6839	6848	6857	6866	6875	6884	6893	9	1 2 3	4 4 5	6 7 8
49	·6902	6911	6920	6928	6937	6946	6955	6964	6972	6981	9	1 2 3	4 4 5	6 7 8

	No.	log
π =	3·14159	0·49715
e =	2·71828	0·43429

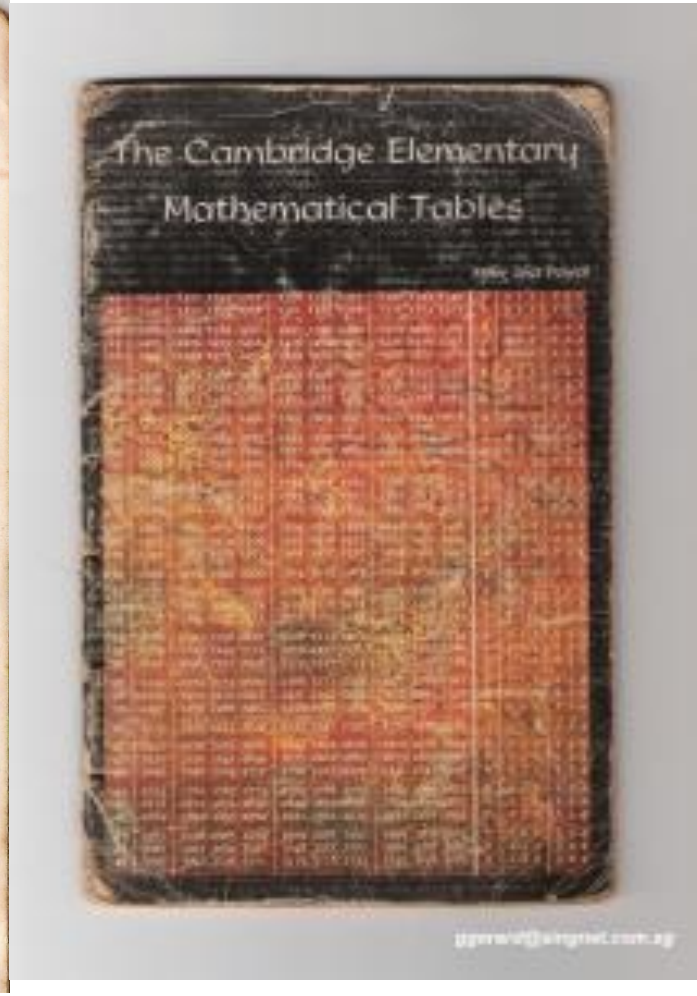
$\ln x = \log_e x = (1/M) \log_{10} x$

$\log x = \log_{10} x = M \log_e x$

	No.	log
$(1/M)$ =	2·30259	0·36222
M =	0·43429	1·63778

p	1	2	3	4	5	6	7	8	9	10
$\log e^p$	0·4343	0·8686	1·3029	1·7372	2·1715	2·6058	3·0401	3·4744	3·9087	4·3429
$\log e^{-p}$	1·5657	1·1314	2·6971	2·2628	3·8285	3·3942	4·9599	4·5256	4·0913	5·6571

x	0	1	2	3	4	5	6	7	8	9	Δ_m	1 2 3	4 5 6	7 8 9
											+	ADD		
50	·6990	6998	7007	7016	7024	7033	7042	7050	7059	7067	9	1 2 3	4 4 5	6 7 8
51	·7076	7084	7093	7101	7110	7118	7126	7135	7143	7152	8	1 2 2	3 4 5	6 6 7
52	·7160	7168	7177	7185	7193	7202	7210	7218	7226	7235	8	1 2 2	3 4 5	6 6 7
53	·7243	7251	7259	7267	7275	7284	7292	7300	7308	7316	8	1 2 2	3 4 5	6 6 7
54	·7324	7332	7340	7348	7356	7364	7372	7380	7388	7396	8	1 2 2	3 4 5	6 6 7
55	·7404	7412	7419	7427	7435	7443	7451	7459	7466	7474	8	1 2 2	3 4 5	6 6 7
56	·7482	7490	7497	7505	7513	7520	7528	7536	7543	7551	8	1 2 2	3 4 5	6 6 7
57	·7559	7566	7574	7582	7589	7597	7604	7612	7619	7627	8	1 2 2	3 4 5	6 6 7
58	·7634	7642	7649	7657	7664	7672	7679	7686	7694	7701	8	1 2 2	3 4 5	6 6 7
59	·7709	7716	7723	7731	7738	7745	7752	7760	7767	7774	7	1 1 2	3 4 4	5 6 6
60	·7782	7789	7796	7803	7810	7818	7825	7832	7839	7846	7	1 1 2	3 4 4	5 6 6
61	·7853	7860	7868	7875	7882	7889	7896	7903	7910	7917	7	1 1 2	3 4 4	5 6 6
62	·7924	7931	7938	7945	7952	7959	7966	7973	7980	7987	7	1 1 2	3 4 4	5 6 6
63	·7993	8000	8007	8014	8021	8028	8035	8041	8048	8055	7	1 1 2	3 4 4	5 6 6
64	·8062	8069	8075	8082	8089	8096	8102	8109	8116	8122	7	1 1 2	3 4 4	5 6 6
65	·8129	8136	8142	8149	8156	8162	8169	8176	8182	8189	7	1 1 2	3 4 4	5 6 6
66	·8195	8202	8209	8215	8222	8228	8235	8241	8248	8254	7	1 1 2	3 4 4	5 6 6
67	·8261	8267	8274	8280	8287	8293	8299	8306	8312	8319	6	1 1 2	2 3 4	4 5 5
68	·8325	8331	8338	8344	8351	8357	8363	8370	8376	8382	6	1 1 2	2 3 4	4 5 5
69	·8388	8395	8401	8407	8414	8420	8426	8432	8439	8445	6	1 1 2	2 3 4	4 5 5
70	·8451	8457	8463	8470	8476	8482	8488	8494	8500	8506	6	1 1 2	2 3 4	4 5 5
71	·8513	8519	8525	8531	8537	8543	8549	8555	8561	8567	6	1 1 2	2 3 4	4 5 5
72	·8573	8579	8585	8591	8597	8603	8609	8615	8621	8627	6	1 1 2	2 3 4	4 5 5
73	·8633	8639	8645	8651	8657	8663	8669	8675	8681	8686	6	1 1 2	2 3 4	4 5 5
74	·8692	8698	8704	8710	8716	8722	8727	8733	8739	8745	6	1 1 2	2 3 4	4 5 5
75	·8751	8756	8762	8768	8774	8779	8785	8791	8797	8802	6	1 1 2	2 3 4	4 5 5
76	·8808	8814	8820	8825	8831	8837	8842	8848	8854	8859	6	1 1 2	2 3 4	4 5 5
77	·8865	8871	8876	8882	8887	8893	8899	8904	8910	8915	6	1 1 2	2 3 4	4 5 5
78	·8921	8927	8932	8938	8943	8949	8954	8960	8965	8971	6	1 1 2	2 3 4	4 5 5
79	·8976	8982	8987	8993	8998	9004	9009	9015	9020	9025	6	1 1 2	2 3 4	4 5 5
80	·9031	9036	9042	9047	9053	9058	9063	9069	9074	9079	5	1 1 2	2 3 3	4 4 5
81	·9085	9090	9096	9101	9106	9112	9117	9122	9128	9133	5	1 1 2	2 3 3	4 4 5
82	·9138	9143	9149	9154	9159	9165	9170	9175	9180	9186	5	1 1 2	2 3 3	4 4 5
83	·9191	9196	9201	9206	9212	9217	9222	9227	9232	9238	5	1 1 2	2 3 3	4 4 5
84	·9243	9248	9253	9258	9263	9269	9274	9279	9284	9289	5	1 1 2	2 3 3	4 4 5
85	·9294	9299	9304	9309	9315	9320	9325	9330	9335	9340	5	1 1 2	2 3 3	4 4 5
86	·9345	9350	9355	9360	9365	9370	9375	9380	9385	9390	5	1 1 2	2 3 3	4 4 5
87	·9395	9400	9405	9410	9415	9420	9425	9430						
88	·9445	9450	9455	9460	9465	9469	9474	9479						
89	·9494	9499	9504	9509	9513	9518	9523	9528						
90	·9542	9547	9552	9557	9562	9566	9571	9576						
91	·9590	9595	9600	9605	9609	9614	9619	9624						
92	·9638	9643	9647	9652	9657	9661	9666	9671						
93	·9685	9689	9694	9699	9703	9708	9713	9717						
94	·9731	9736	9741	9745	9750	9754	9759	9763						
95	·9777	9782	9786	9791	9795	9800	9805	9809						
96	·9823	9827	9832	9836	9841	9845	9850	9854						
97	·9868	9872	9877	9881	9886	9890	9894	9899						
98	·9912	9917	9921	9926	9930	9934	9939	9943						
99	·9956	9961	9965	9969	9974	9978	9983	9987						



degrees	sine	cosine	tangent	degrees	sine	cosine	tangent
1	0.0175	0.9998	0.0175	46	0.7193	0.6947	1.0355
2	0.0349	0.9994	0.0349	47	0.7314	0.6820	1.0724
3	0.0523	0.9986	0.0524	48	0.7431	0.6691	1.1106
4	0.0698	0.9976	0.0699	49	0.7547	0.6561	1.1504
5	0.0872	0.9962	0.0875	50	0.7660	0.6428	1.1918
6	0.1045	0.9945	0.1051	51	0.7771	0.6293	1.2349
7	0.1219	0.9925	0.1228	52	0.7880	0.6157	1.2799
8	0.1392	0.9903	0.1405	53	0.7986	0.6018	1.3270
9	0.1564	0.9877	0.1584	54	0.8090	0.5878	1.3764
10	0.1736	0.9848	0.1763	55	0.8192	0.5736	1.4281

Charles Babbage

- Mașina diferențială 1823
- Mașina analitică 1833
  - Precursorul calculatorului digital modern!

Aplicații

- Tabele matematice - astronomie
- Tabele nautice - marină

Ideea de bază

- Orice funcție continuă poate fi aproximată printr-un polinom --- Weierstrass

Tehnologie

- mecanică – roți dințate, războiul lui Jacquard, calculatoare simple

Mașina diferențială

O mașină pentru calculul tabelelor matematice

Weierstrass:

- Orice funcție continuă poate fi aproximată printr-un polinom
- Orice polinom poate fi calculat folosind tabele *diferențiale*

De exemplu

$f(n) = n^2 + n + 41$

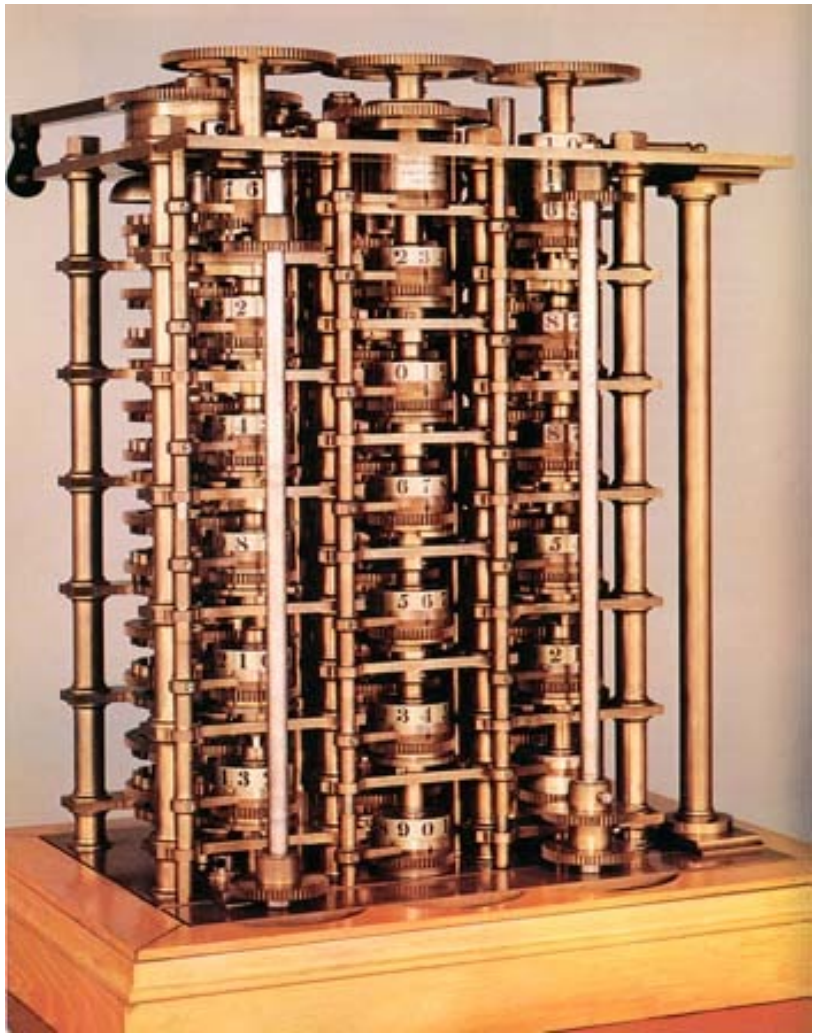
$d1(n) = f(n) - f(n-1) = 2n$

$d2(n) = d1(n) - d1(n-1) = 2$

$f(n) = f(n-1) + d1(n) = f(n-1) + (d1(n-1) + 2)$

Tot ce trebuie să faci este să aduni!

n	0	1	2	3	4
d2(n)			2	2	2
d1(n)		2	4	6	8
f(n)	41	43	47	53	61



Mașina diferențială

1823

- Babbage publică un articol în care descrie mașina diferențială

1834

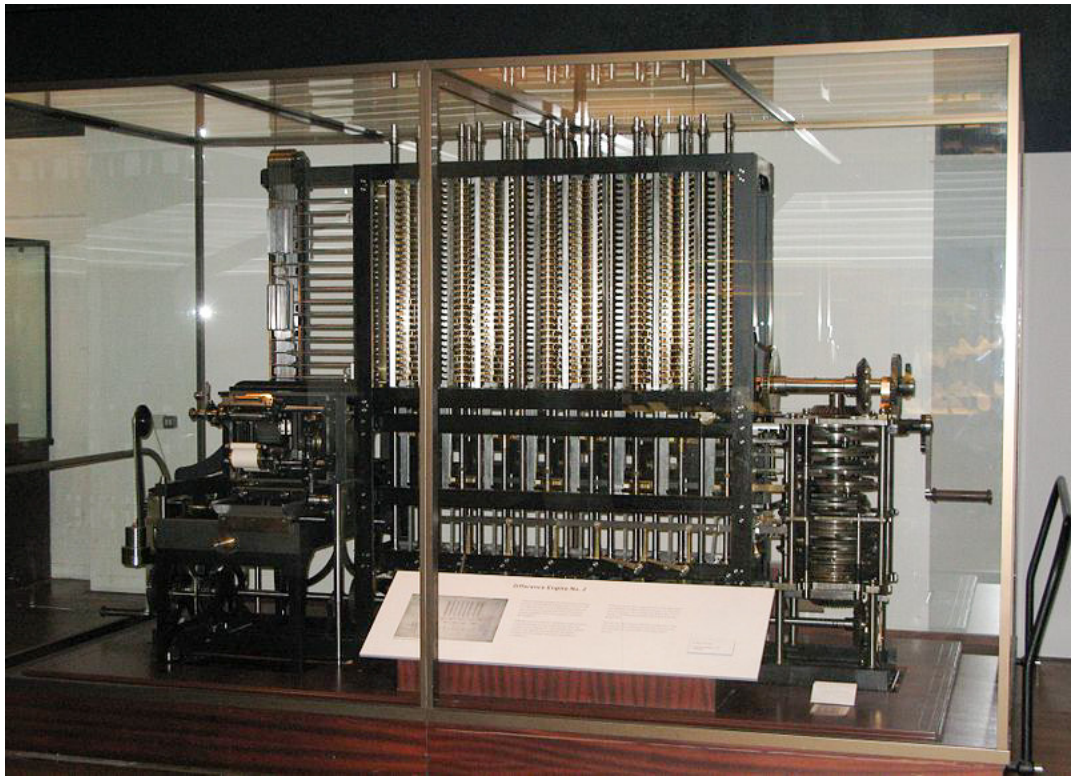
- Articolul este citit de Per Georg Scheutz și fiul lui în Suedia

1842

- Babbage renunță la ideea de-a construi un prototip funcțional; dorește să construiască Mașina Analitică!

1855

- Scheutz expune mașina la Expoziția Mondială de la Paris
- Poate calcula orice polinom până la gradul 6.
- *Viteză de calcul:* 33 până la 44 numere de 32 de cifre pe minut!



Acum, mașina se află la Smithsonian

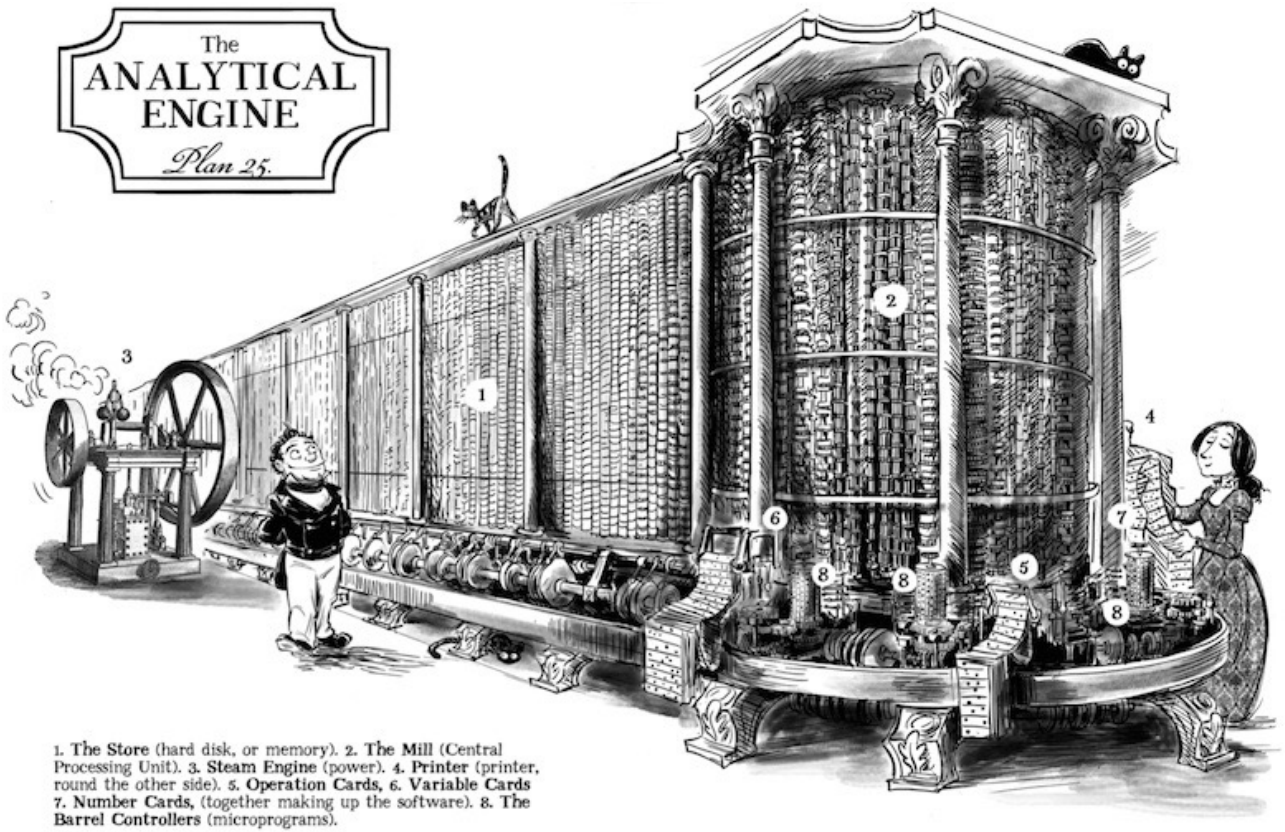
Mașina Analitică

1833: Babbage publică un articol în care-i descrie funcționarea
- *Dezvoltată în timpul unei pauze pe care a făcut-o în dezvoltarea mașinii diferențiale*

Inspirația: Războaiele de țesut *Jacquard*
- Controlate prin cartele perforate
  - Modelul găurilor perforate în cartele dictează modelul țesăturii ⇒ *program*
  - Același set de cartele poate fi folosit cu fire de culori diferite ⇒ *numere*

1871: Babbage moare
- Mașina analitică rămâne neconstruită

Nu este clar dacă mașina analitică ar fi putut fi construită cu tehnologia din vremea lui Babbage



1. **The Store** (hard disk, or memory). 2. **The Mill** (Central Processing Unit). 3. **Steam Engine** (power). 4. **Printer** (printer, round the other side). 5. **Operation Cards**, 6. **Variable Cards** 7. **Number Cards**, (together making up the software). 8. **The Barrel Controllers** (microprograms).

The Thrilling Adventures of Lovelace and Babbage: The (Mostly) True Story of the First Computer – Sydney Padua

Primul software developer

Ada Byron *aka* "Lady Lovelace" 1815-52

- Traduce lecturile lui Luigi Menabrea, care publică notițele lui Babbage în Italia
- Lovelace adaugă contribuții proprii notițelor luate și descrie un program pentru mașina analitică ce poate calcula numere Bernoulli
  - Primul program!
- Își imaginează alte utilizări ale mașinii analitice, în afara calculelor matematice simple
- Era interesată în modelarea creierului

Linear Equation Solver

John Atanasoff, Iowa State University

1930:

- Atanasoff construiește Linear Equation Solver.
- Avea 300 de tuburi!
- Calculator digital binar dedicat
- RAM dinamic(valori binare stocate în condensatoare la care se făcea refresh)

Aplicație:

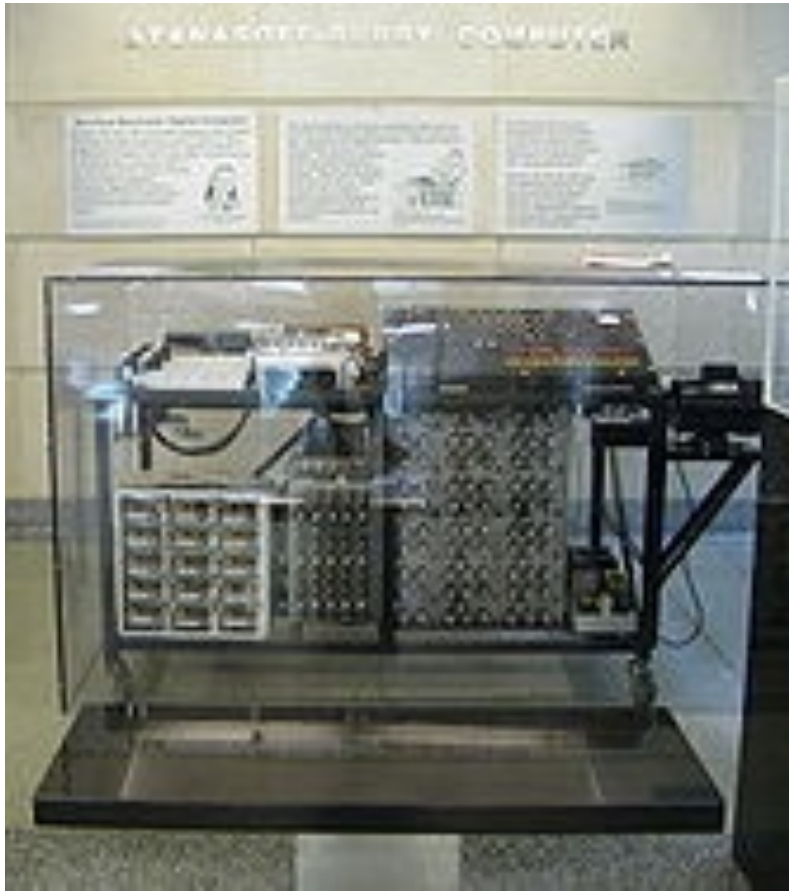
- Rezolvarea ecuațiilor liniare integrale și diferențiale.

Competiția:

- Analizorul Diferențial al lui Vannevar Bush
  --- *calculator analogic*

Tehnologie:

- Tuburi și relee electromagnetice

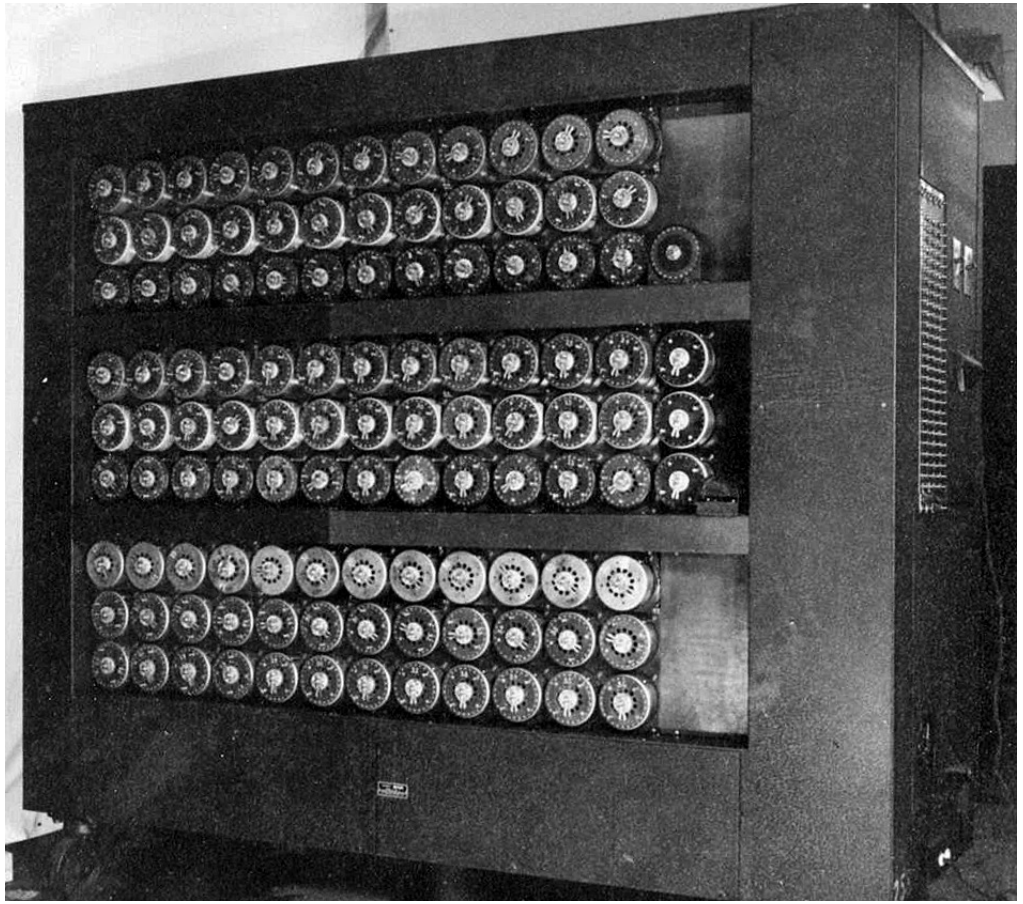


Atanasoff a decis că modul corect de-a face calcule este prin folosirea de numere binare stocate electronic ca tensiuni.

The Bombe

Marian Rejewski, Alan Turing, Gordon Welchman, Bletchley Park 1939

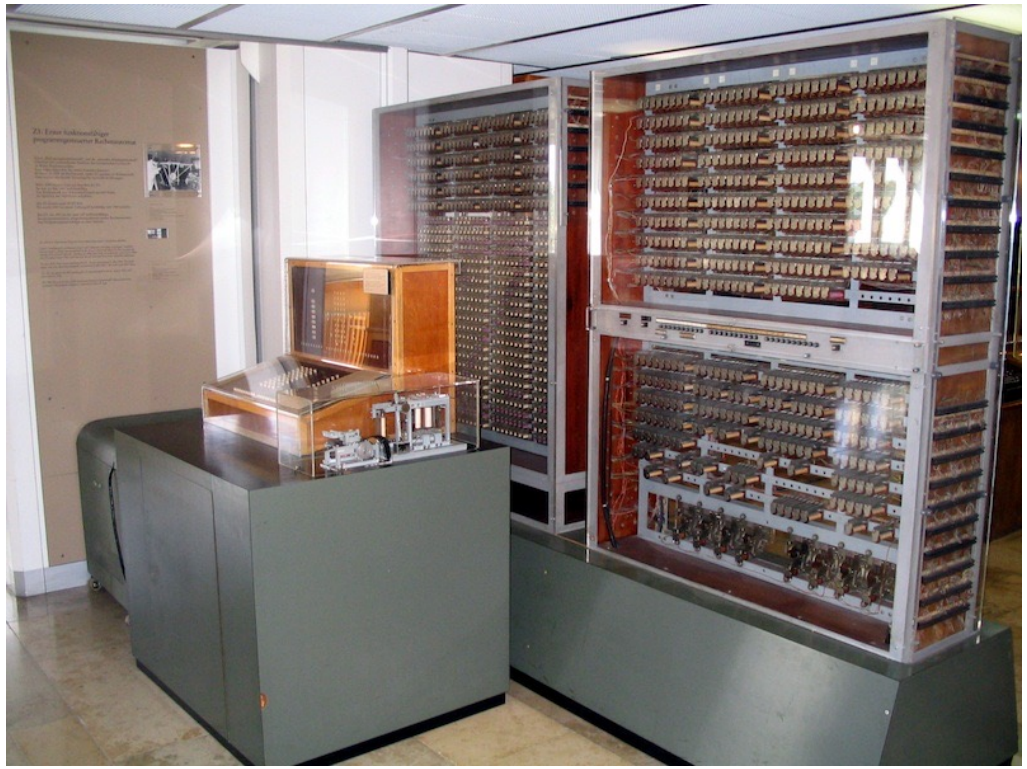
- Mașină de calcul electromecanică pentru descifrarea codurilor Enigma folosite de Germania nazistă
- După un model inițial produs de Biuro Szyfrow (biroul de criptare) polonez



Zuse Z3

Konrad Zuse, Berlin, 1941

- Construit de Konrad Zuse în timpul războiului, folosind 2000 relee
- Aritmetică în virgulă mobilă cu tratarea situațiilor excepționale în hardware (+/- infinity, undefined)
  - 1-bit sign, 7-bit exponent, 14-bit significand
- 64 cuvinte de memorie
- Bandă asamblare cu două etape
  1) fetch&execute 2) writeback
- Fără execuție condițională
- Programat cu benzi de hârtie perforată

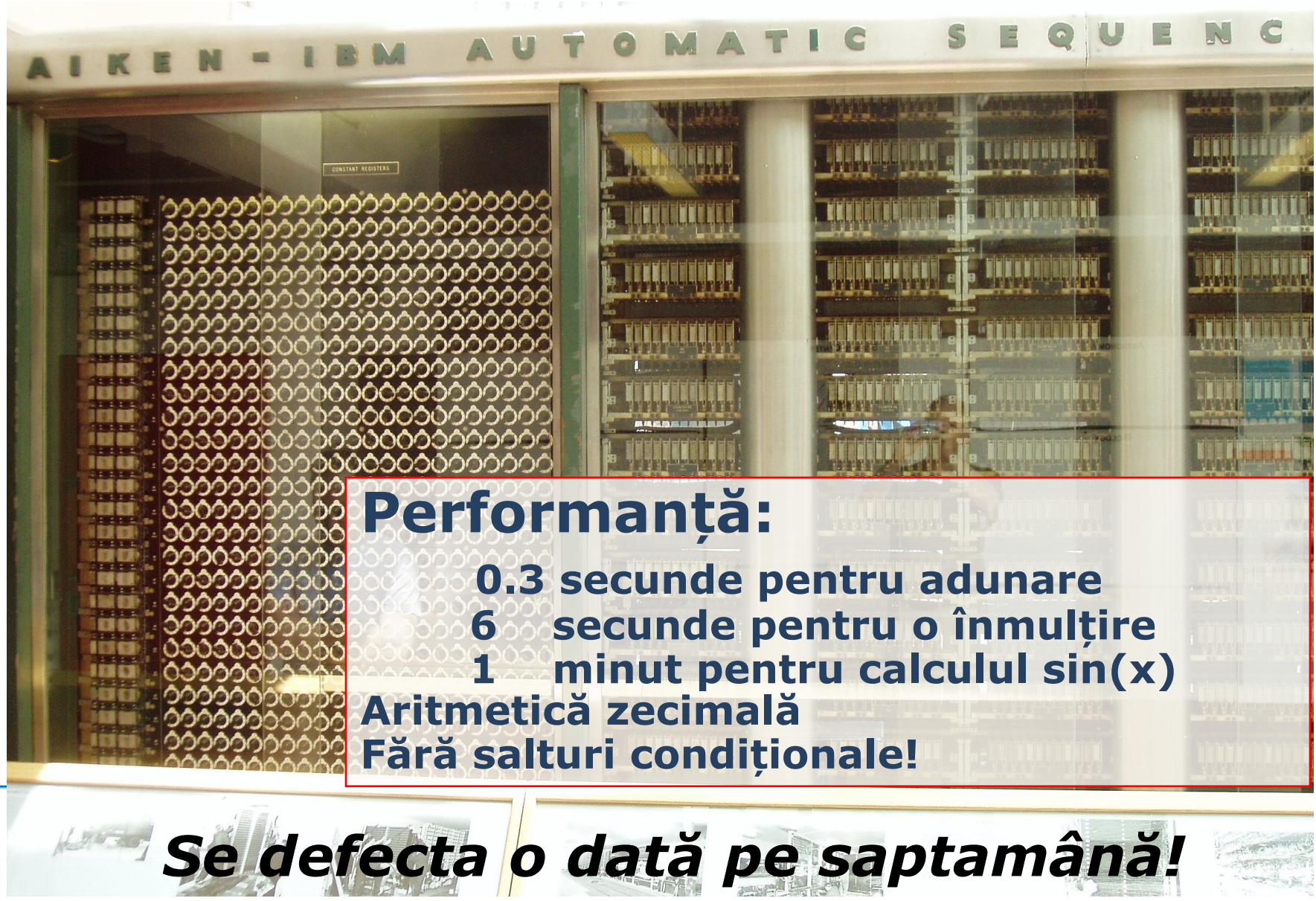


Replica mașinii Zuse Z3 de la Deutsches Museum, Munchen

Harvard Mark I

Construit în 1944 în Laboratoarele IBM Endicott

- Howard Aiken – Profesor de fizică la Harvard
- În mare parte mecanic, dar include și relee și angrenaje acționate electro-magnetic
- Cântarea 5 tone și avea 750,000 de componente
- Ceas pentru sincronizare cu o perioadă de 0.015 secunde (66Hz)
- Codul era executat ciclic prin lipirea benzii perforate pe care era stocat într-o **buclă**



Performanță:

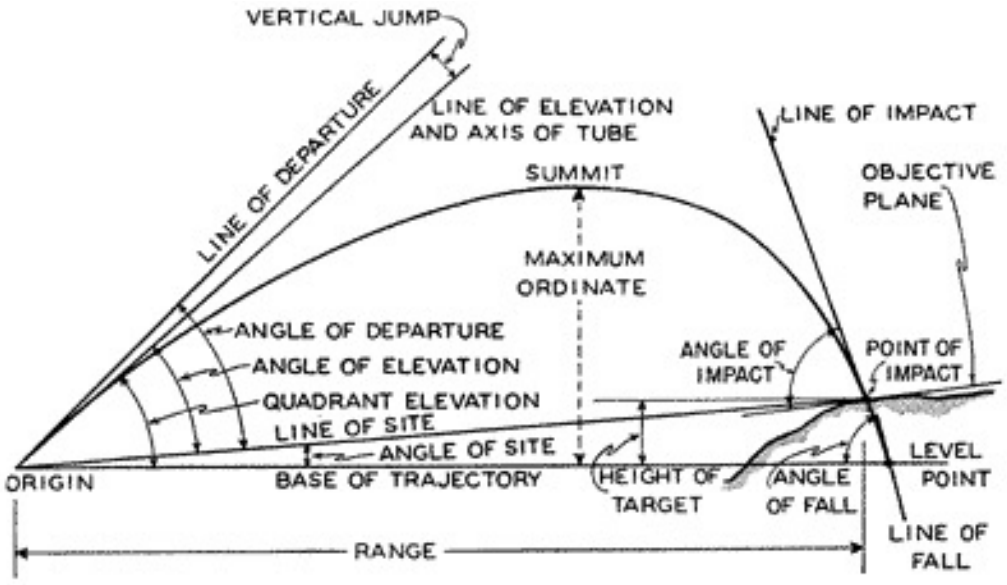
0.3 secunde pentru adunare
6 secunde pentru o înmulțire
1 minut pentru calculul sin(x)
Aritmetică zecimală
Fără salturi condiționale!

Se defecta o dată pe saptamână!

Electronic Numerical Integrator and Computer (ENIAC), 1945

- Inspirați de Atanasoff și Berry, Eckert și Mauchly construiesc ENIAC (1943-45) la University of Pennsylvania
- Primul calculator electronic digital, complet operațional și **general-purpose**!
  - 30 tone, 72 metri pătrați, 200KW
- Performanță
  - Citea 120 cartele pe minut
  - Adunare în 200μs, împărțire în 6ms
  - De 1000x mai rapid decât Mark I
- Nu și foarte fiabil! (record de funcționare continuă: 5 zile)

Proiectat în timpul WW2

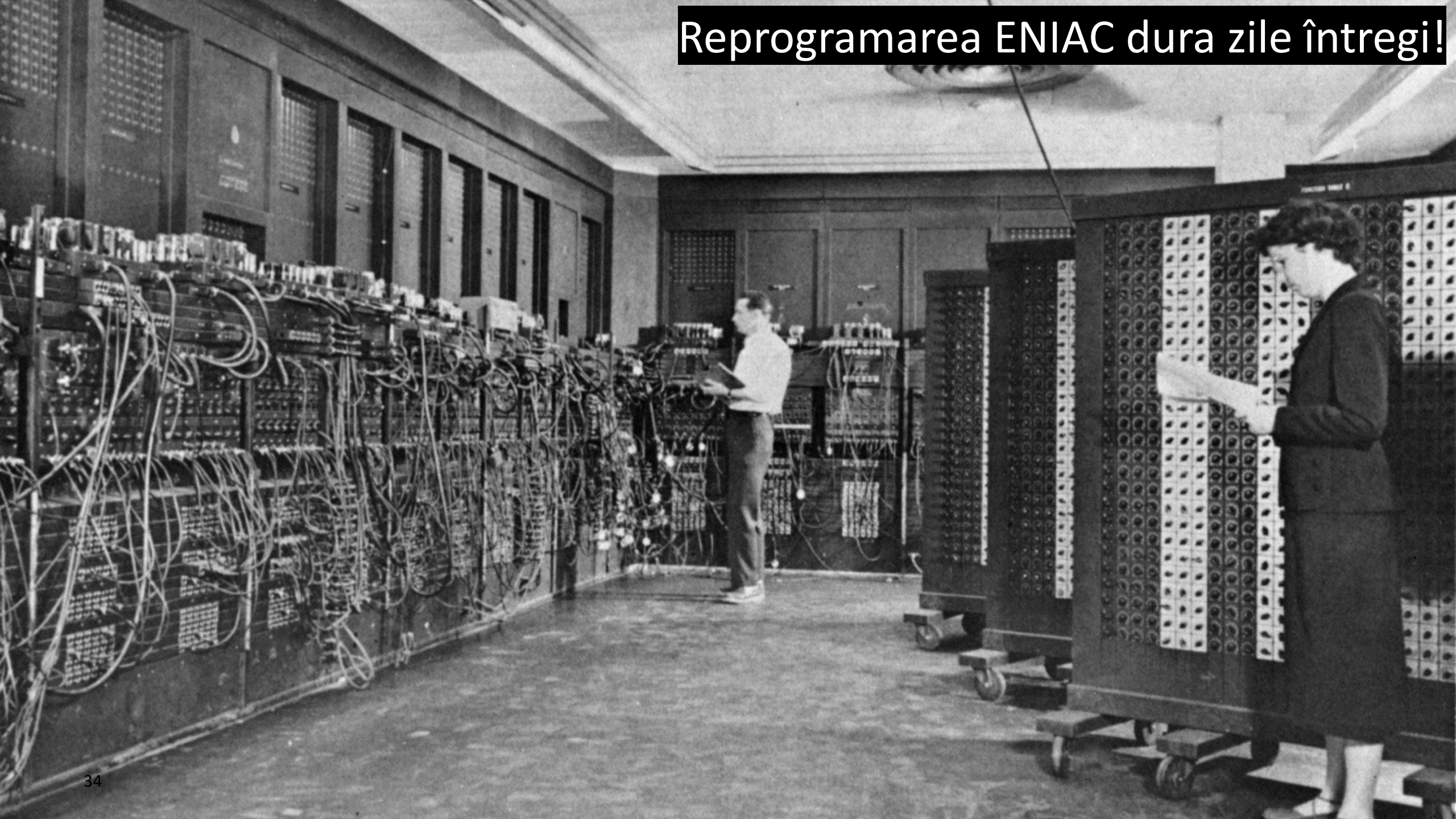


Aplicație: Calcule balistice

unghi = f (locație, vânt spate, vânt lateral, densitate aer, temperatură, greutate obuz, putere propelant, ...)

Calcula în 30s ce un om calcula în 20 ore

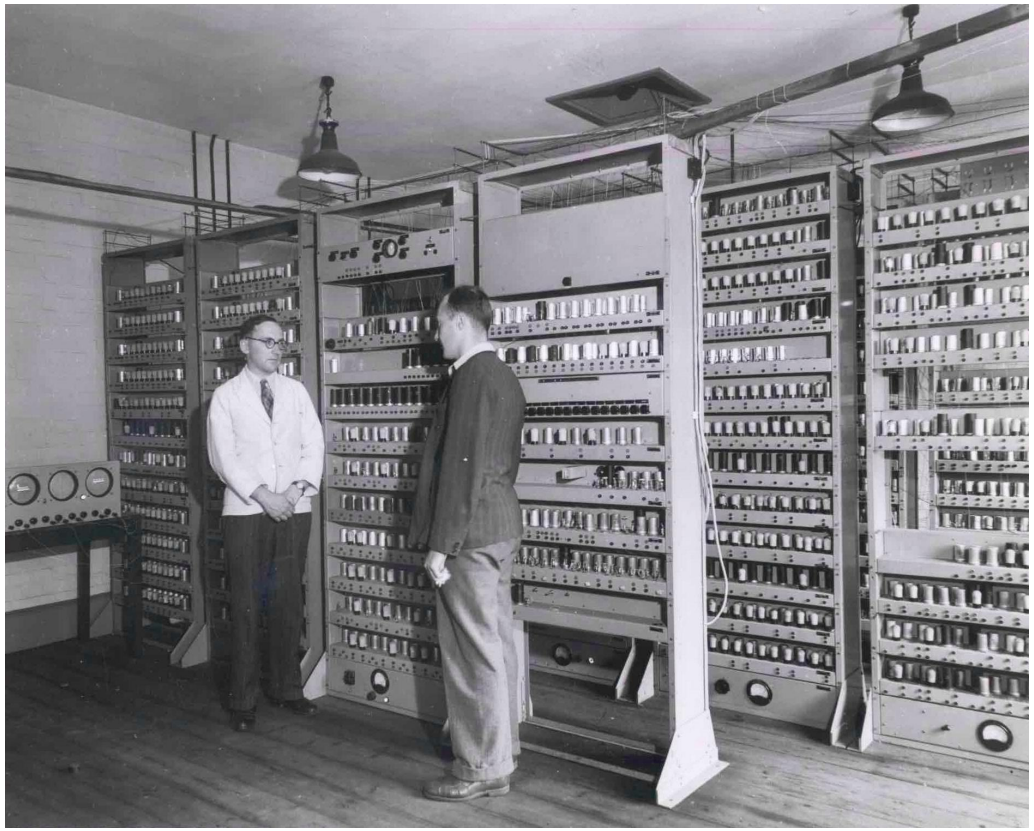
Reprogramarea ENIAC dura zile întregi!



Electronic Discrete Variable Automatic Computer (EDVAC), 1951

- Unitatea de programare pt. ENIAC era externă
  - Diferite secvențe de operații erau executate independent de rezultatele operațiilor respective
  - Pentru a modifica ordinea execuției era nevoie de intervenția unui operator uman
- Eckert, Mauchly, John von Neumann și alții au proiectat EDVAC pentru a rezolva această problemă
  - Soluția era calculatorul cu program stocat (*stored program computer)*

⇒ *"programul poate fi manipulat precum datele"*



Problema majoră: *Fiabilitatea*!

Mean time between failures (MTBF)

Whirlwind de la MIT, cu un MTBF de 20 min. era probabil cea mai fiabilă mașină de calcul a vremii!

Motivele lipsei de fiabilitate

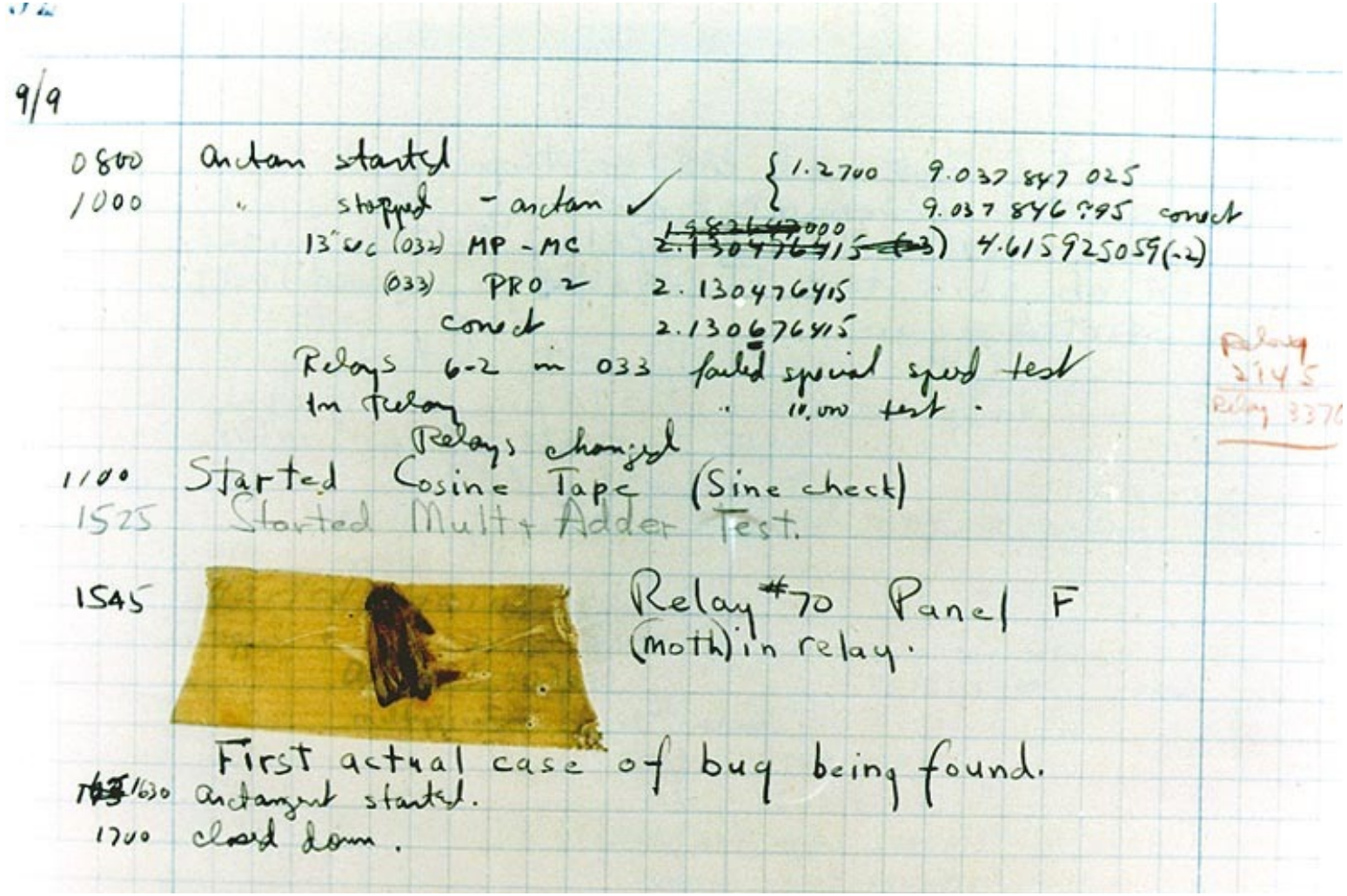
1. Tuburi cu vid
2. Mediul de stocare
   - Linii acustice de întârziere
   - Linii de întârziere cu mercur
   - Tuburi Williams
   - Memorie pe tambur
3. Bugs!

Fiabilitatea mărită odată cu inventarea memoriei cu miez de ferită

J. Forrester 1954 la MIT pentru proiectul Whirlwind

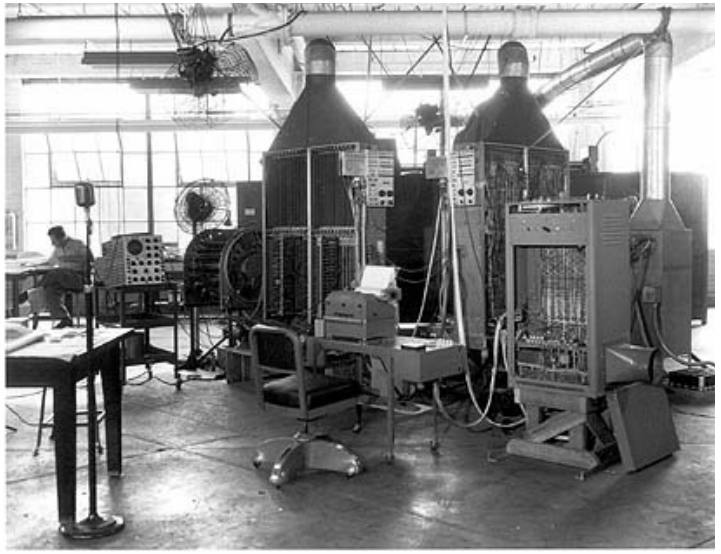
First actual case of bug being found

O pagină din jurnalul calculatorului Harvard Mark II cu o molie moartă ce a fost extrasă din dispozitiv.



BINAC (1949) & UNIVAC (1951)

- Eckert și Mauchly pleacă de la U.Penn după o dispută pe drepturi de autor și formează împreună Eckert-Mauchly Computer Corporation
- Primul calculator comercial din lume a fost BINAC, cu două CPU-uri care se verificau unul pe celălalt
  - BINAC nu a mai funcționat niciodată după livrarea către primul (și ultimul) client
- Al doilea calculator commercial a fost UNIVAC
  - Folosea linii de întârziere cu mercur ca memorie, 1000 cuvinte
  - Folosit in prezicerea rezultatului alegerilor din 1952
  - Vândut în 46 exemplare a câte >$1M fiecare
  - De multe ori, denumit incorect IBM UNIVAC



IBM 701, 1952

Primul calculator comercial, științific al IBM

IBM 701 -- 30 mașini vândute intre 1953-54
foloseau tuburi catodice drept memorie, 72 tuburi, 32x32 biti fiecare

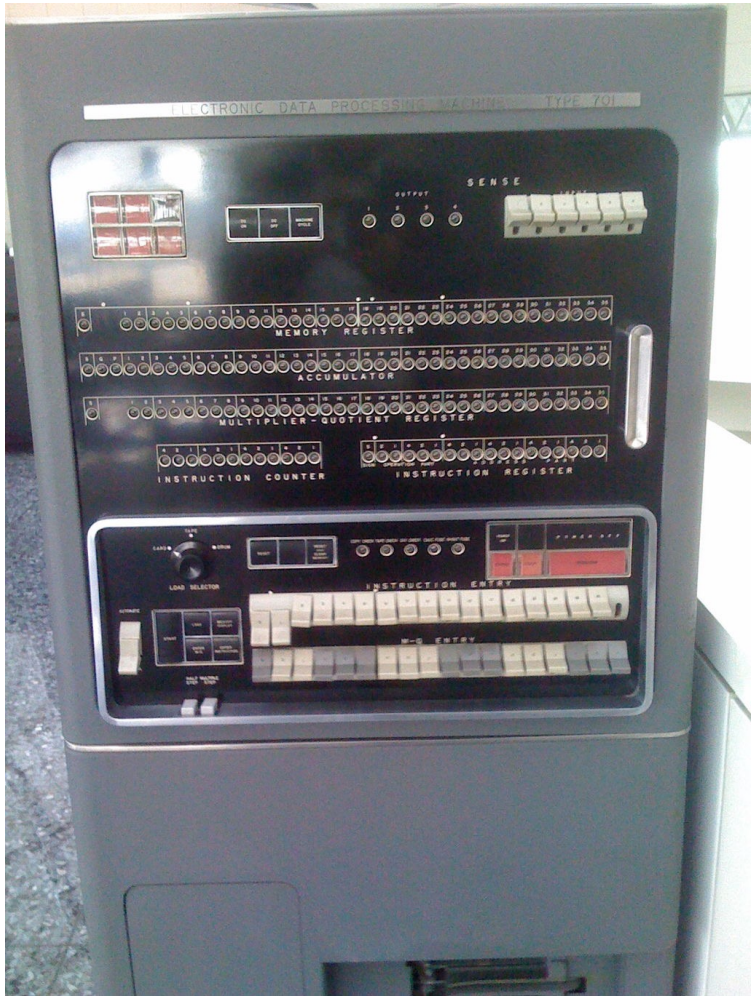
IBM 650 -- versiune mai ieftină bazată pe memorie pe tambur, mai mult de 120 de unități vândute în 1954 și comenzi pentru încă 750!

Citat atribuit greșit lui Thomas Watson Sr (Chairman & CEO IBM):

"I think there is a world market for maybe five computers"

De fapt, TWJr a spus, într-o întalnire a asociaților:

"as a result of our trip [selling the 701]***, on which we expected to get orders for five machines, we came home with orders for 18."***



IBM701, consolă operator

Calculatoarele anilor 50

- Hardware-ul foarte costisitor
- Memorii foarte mici (mii de cuvinte)

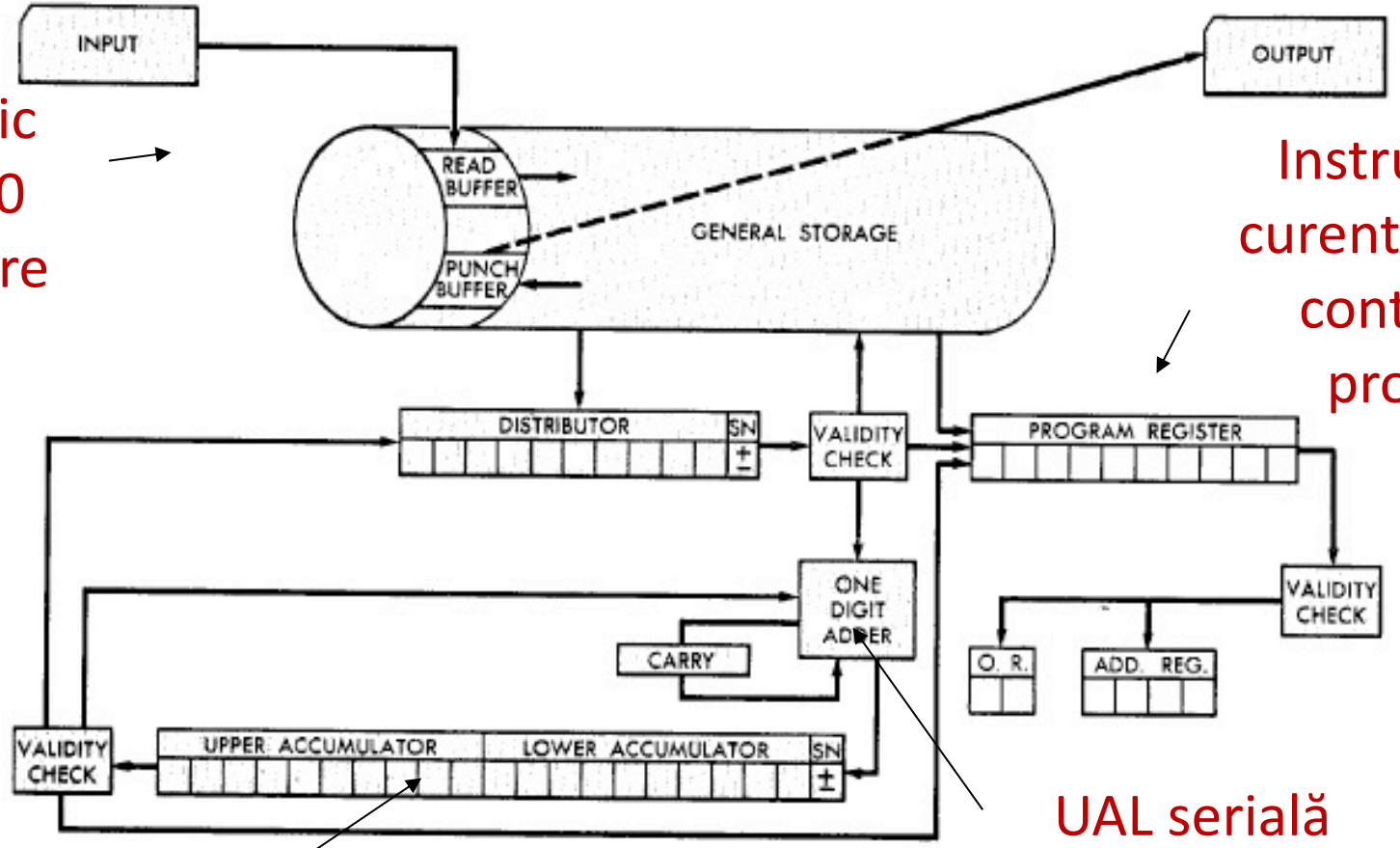
  ⇒ Fără un program de sistem rezident în memorie!
- Accesul la memorie de 10-50x mai lent decât un ciclu de procesor

  ⇒ Timpul de execuție al unei instrucțiuni era dominat de timpul de acces la memorie.
- Capabilitatea de-a proiecta circuite complexe de control pentru execuția unei instrucțiuni era principala problemă și nu viteza de execuție sau decodificare a unei instrucțiuni
- Modul în care un programator vedea mașina de calcul nu era deloc diferit față de un proiectant hardware.

Hard Drive IBM Data Processing Division capacitate 5MB, 1956

IBM 650 (1953-4)

Tambur magnetic (1,000 sau 2,000 cuvinte de 10 cifre zecimale)



Instrucțiunea curentă (inclusiv contorul de program)

UAL serială

Registru acumulator de 20 de cifre

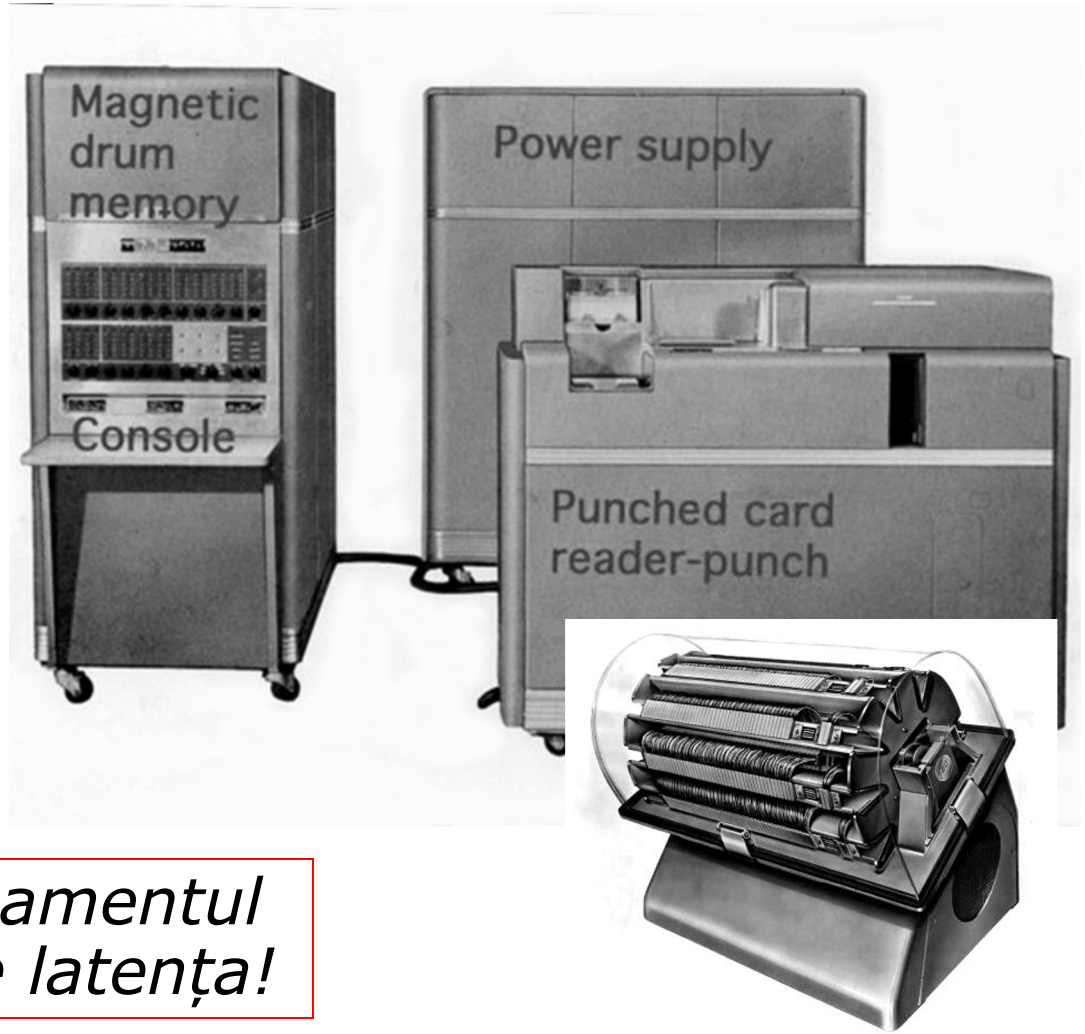
[Din 650 Manual, © IBM]

IBM 650 – perspectiva programatorului

O mașină cu tambur magnetic și 44 de instrucțiuni

Instrucțiune: 60 1234 1009

"Încarcă conținutul locației 1234 în registrul *distribuție*; încarcă-l de asemenea în *acumulatorul superior*; setează *acumulatorul inferior* la zero; apoi mergi la locația 1009 pentru următoarea instrucțiune"



Programatorii pricepuți optimizau amplasamentul instrucțiunilor pe tambur pentru a reduce latența!

Probleme de compatibilitate

În anii 60', *IBM avea 4 familii incompatibile de calculatoare!*

701	→	7094
650	→	7074
702	→	7080
1401	→	7010

Fiecare sistem avea propriul

- Set de instrucțiuni
- Sistem I/O și elemente de stocare:
  bandă magnetică, memorii pe tambur și unități de disc
- asambloare, compilatoare, biblioteci,...
- nișă de piață
  afaceri, cercetare, timp real, ...

⇒ *IBM 360*

IBM360, 1965

- Primul calculator produs în masă
- Sistem low-end cu stocare pe tambur magnetic și UAL serial
- Aproape 2000 unități produse

[Cushing Memorial Library and Archives, Texas A&M, Creative Commons Attribution 2.0 Generic]

IBM 360: Idei de design

Amdahl, Blaauw and Brooks, 1964

- Design-ul trebuie să fie făcut a.î. să permită extinderea și dezvoltarea de mașini succesoare
- O metodă generală de-a conecta dispozitive de I/O
- Performanța totală – rezultate pe lună în locul biți pe microsecundă
- Mașina trebuie să fie capabilă să se auto-monitorizeze, fără intervenția operatorului uman.
- *Fault checking* construit în hardware și metode de-a localiza un defect, pentru a reduce timpii de nefuncționare
- Sisteme simplu de asamblat cu dispozitive I/O, memorii, procesoare redundante, pentru a implmenta toleranta la defecte
- Operații în virgulă mobilă cu precizie mare

IBM 360: O mașină *General-Purpose Register (GPR)*

- Starea procesorului
  - 16 registre General-Purpose de 32 de biti
    - *Pot fi folosite pentru adresare indexată și bazată*
    - *Registrul 0 avea proprietăți speciale*
  - 4 registre în virgulă mobilă de 64 de biți
  - Program Status Word (PSW)
    - *PC, Flag-uri condiționale, Flag-uri de control*
- O mașină de 32 de biți cu adresare pe 24 de biți
  - Nici o instrucțiune nu putea să conțină o adresă pe 24 de biți!
- Format diferit pentru date
  - 8-bit bytes, 16-bit half-words, 32-bit words, 64-bit double-words

Din cauza IBM 360 un byte are 8 biți în ziua de azi!

IBM360 Model 50

IBM 360: Implementări inițiale

	Model 30	*. . .*	*Model 70*
Storage	8K - 64 KB		256K - 512 KB
Datapath	8-bit		64-bit
Circuit Delay	30 nsec/level		5 nsec/level
Local Store	Main Store		Transistor Registers
Control Store	Read only 1~sec		Conventional circuits

Arhitectura setului de instrucțiuni (ISA) IBM 360 ascundea complet diferențele tehnologice dintre diferitele modele.

Primul ISA proiectat să aibă o interfață portabilă hardware-software!

IBM360 este încă prezent în ziua de azi, cu mici modificări!

IBM 360: 56 ani mai târziu…

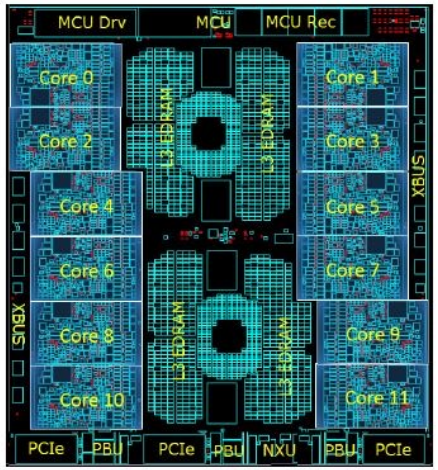
[z15, 2020, 14nm, 4 niveluri de cache, ~10 miliarde tranzistoare]

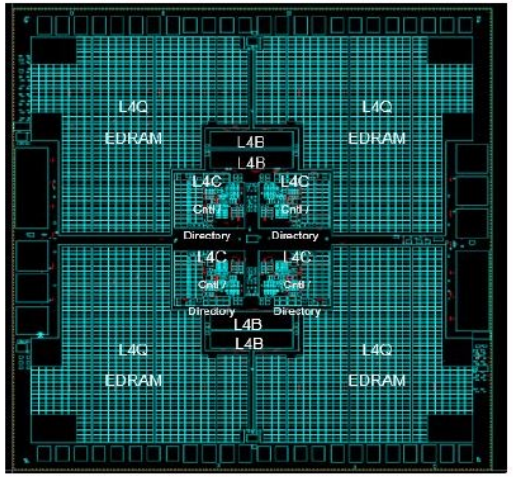
IBM Z

IBM

IBM z15 – designed for massive scale commercial workloads

- Processor Chip w/ L3 cache + System Control Chip w/ L4 cache
- 14nm SOI technology, 5.2GHz water cooled enterprise server
  - CP: 9.2 billion transistors, 14.5 miles of wire
  - SC: 12.2 billion transistors, 13.5 miles of wire
- Up to 240 physical cores in 5-drawer shared-memory SMP
- 190 configurable customer CPUs, plus IO assist and firmware CPUs
- 14% single thread speedup & 25% capacity growth vs z14
- Micro-architectural and architectural enhancements for wide variety of workloads

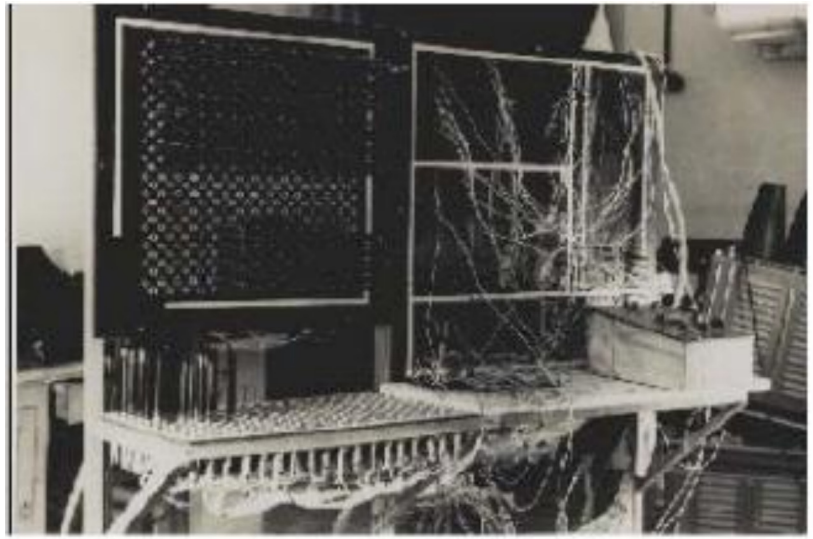


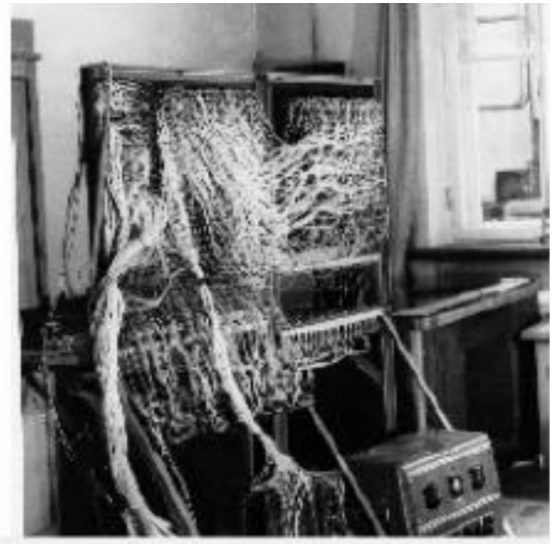


Calculatoare proiectate de membrii Departamentului Calculatoare @ UPB

Electrointegrator pentru soluționarea unor probleme de câmp.

Adrian Petrescu (1963)





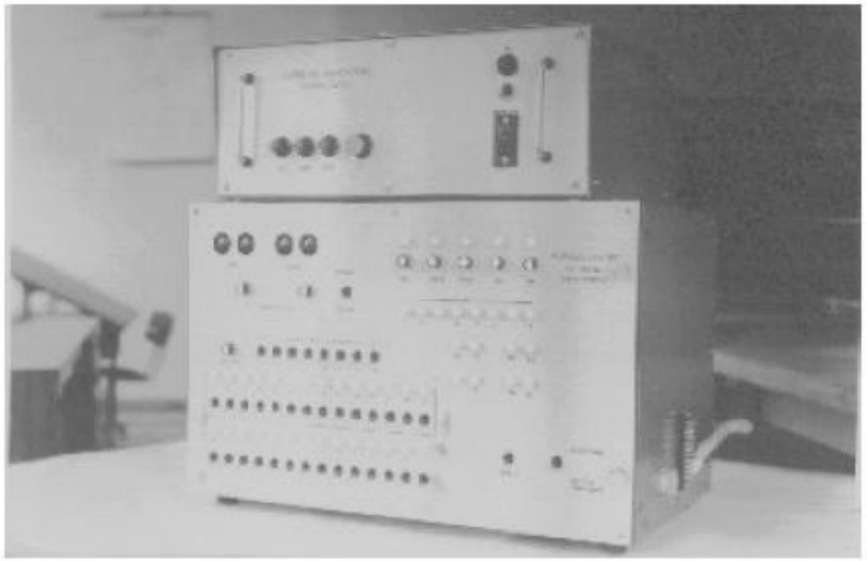
MAC-1 Calculator Analogic cu 30 de amplificatoare operaționale.

Adrian Petrescu, Petre Dimo, Ivan Sipos (1965)

Calculatoare proiectate de membrii Departamentului Calculatoare @ UPB

Microcalculatorul MC-1

Adrian Petrescu, Nicolae Țăpuș, Trandafir Moisa (1972)



Microcalculatorul FELIX MC-8

Adrian Petrescu, Nicolae Țăpuș, Trandafir Moisa (1975)

Calculatoare proiectate de membrii Departamentului Calculatoare @ UPB

Microcalculatorul FELIX M-18

Adrian Petrescu, Nicolae Țăpuș, Trandafir Moisa (1978)

Microcalculatorul aMIC

Adrian Petrescu, Francisc Iacob (1984)

Calculatoare proiectate de membrii Departamentului Calculatoare @ UPB

Microcalculatorul FELIX HC-85

Adrian Petrescu, Francisc Iacob (1985)

Microcalculatorul FELIX-PC

Adrian Petrescu, Nicolae Țăpuș, Trandafir Moisa, Irina Athanasiu (1985)

În concluzie …

- Arhitectura Calculatoarelor >> ISA și RTL
- CN se bazează pe interacțiunea dintre hardware și software și design-ul nivelelor de abstractizare
- Arhitectura de calcul este modelată de tehnologie și aplicații
  - Istoria dispozitivelor de calcul ne poate da indicii despre viitorul tehnologiei
- Computer Science este la un punct de trecere între calculul secvențial și cel paralel
  - Menținerea creșterii de performanță necesită numeroase inovații, inclusiv în domeniul arhitecturii de calcul.

Acknowledgements

- Aceste slide-uri contin materiale dezvoltate de:
  - Arvind (MIT)
  - Krste Asanovic (MIT/UCB)
  - Joel Emer (Intel/MIT)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)
  - David Patterson (UCB)

- MIT material derived from course 6.823
- UCB material derived from course CS252