

Ένωση Πληροφορικών Ελλάδας

Από τον áβακα στους σύγχρονους
επεξεργαστές: Χτίζοντας τους Η/Υ

Χάρης Γεωργίου (MSc, PhD)

Ένωση Πληροφορικών Ελλάδας

Στόχοι:

- Πρώτος “καθολικός” φορέας εκπροσώπησης πτυχιούχων Πληροφορικής.
- Αρμόδιος φορέας εκπροσώπησης επαγγελματιών Πληροφορικής.
- Αρμόδιος επιστημονικός “συμβουλευτικός” φορέας για το Δημόσιο.
- Αρωγός της Εθνικής Ψηφιακής Στρατηγικής & Παιδείας της χώρας.

<https://www.epe.org.gr>



Τομείς παρέμβασης

Ποιοι είναι οι κύριοι τομείς παρεμβάσεων της ΕΠΕ;

- ① Εθνική Ψηφιακή Στρατηγική & Οικονομία
- ② Εργασιακά (ΤΠΕ), Δημόσιος & ιδιωτικός τομέας
- ③ Παιδεία (Α', Β', Γ')
- ④ Έρευνα & Τεχνολογία
- ⑤ Έργα & υπηρεσίες ΤΠΕ
- ⑥ Ασφάλεια συστημάτων & δεδομένων
- ⑦ Ανοικτά συστήματα & πρότυπα
- ⑧ Χρήση ΕΛ/ΛΑΚ
- ⑨ Πνευματικά δικαιώματα
- ⑩ Κώδικας Δεοντολογίας (ΤΠΕ)
- ⑪ Κοινωνική μέριμνα (ICT4D)





Harris Georgiou (MSc, PhD) – <https://github.com/xgeorgio/info>

- R&D: Associate post-doc researcher and lecturer with the University Athens (NKUA) and University of Piraeus (UniPi)
- Consultant in Medical Imaging, Machine Learning, Data Analytics, Signal Processing, Process Optimization, Dynamic Systems, Complexity & Emergent A.I., Game Theory
- HRTA member since 2009, LEAR / scientific advisor
- HRTA field operator (USAR, scuba diver)
- Wilderness first aid, paediatric (child/infant)
- Humanitarian aid & disaster relief in Ghana, Lesvos, Piraeus
- Support of unaccomp. minors, teacher in community schools
- Streetwork training, psychological first aid & victim support
- 2+4 books, 200+ scientific papers/articles (and 6 marathons)

Επισκόπηση

- Περιεχόμενα:
 - Μέρος I: Αριθμητική και Λογική στις μηχανές
 - Μέρος II: Ψηφιακή τεχνολογία επεξεργαστών
- Σχετικό υλικό:
 - «Computer Architecture - Lecture 1: Introduction and Basics (Fall 2022)», ETH –
<https://www.youtube.com/watch?v=BIpPTqHK-Lc>
 - «Architecture All Access: Modern CPU Architecture Part 1 – Key Concepts», Intel –
<https://www.youtube.com/watch?v=vgPFzblBh7w>
 - Computer History Museum (CHM) – <https://www.youtube.com/@ComputerHistory>
 - Advent of Computing – <https://www.youtube.com/@adventofcomputing4504>
 - «Hello x86: Low-level assembly coding for the 8086», @ApneaCoding –
<https://youtu.be/tF16xTbd42w>
 - «BAM neural network in Arduino», @ApneaCoding – <https://youtu.be/RkM-rpSVD4I>

Μέρος Ι: Αριθμητική και Λογική στις μηχανές

1. Οι απαρχές της Αριθμητικής στην καθημερινότητα.
2. Ο άβακας και άλλα εργαλεία.
3. Μηχανικοί υπολογιστές.
4. Ψηφιακά ηλεκτρονικά.



1		6	
2		7	
3		8	
4		9	
5		10	

In the Paleolithic era, the tally system was discovered on bones called tally sticks. They simply used lines to represent numbers, i.e. |||||.

Carved reindeer antler with tally marks

La Madeleine, France

About 17,000–11,500 years old

Cast

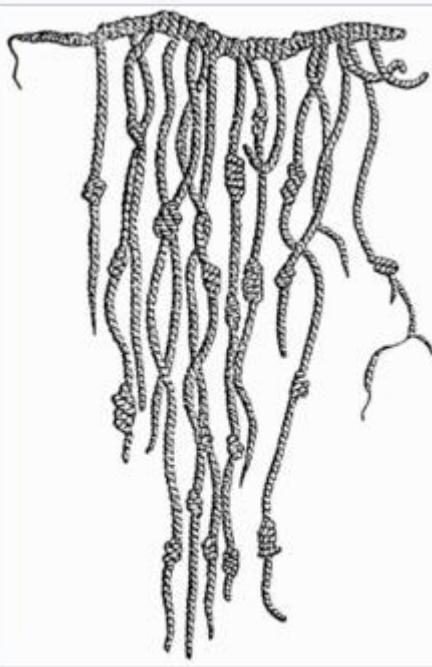


♀ 1	♂ 11	♀♀ 21	♀♀♂ 31	♀♀♀♀ 41	♀♀♀♀♂ 51
♀♀ 2	♀♀♂ 12	♀♀♀♀ 22	♀♀♀♀♂ 32	♀♀♀♀♀♀ 42	♀♀♀♀♀♀♂ 52
♀♀♀♀ 3	♀♀♀♀♂ 13	♀♀♀♀♀♀ 23	♀♀♀♀♀♀♂ 33	♀♀♀♀♀♀♀♀ 43	♀♀♀♀♀♀♀♀♂ 53
♀♀♀♀♀♀ 4	♀♀♀♀♀♀♂ 14	♀♀♀♀♀♀♀♀ 24	♀♀♀♀♀♀♀♀♂ 34	♀♀♀♀♀♀♀♀♀♀ 44	♀♀♀♀♀♀♀♀♀♀♂ 54
♀♀♀♀♀♀♀♀ 5	♀♀♀♀♀♀♀♀♂ 15	♀♀♀♀♀♀♀♀♀♀ 25	♀♀♀♀♀♀♀♀♀♀♂ 35	♀♀♀♀♀♀♀♀♀♀♀♀ 45	♀♀♀♀♀♀♀♀♀♀♀♀♂ 55
♀♀♀♀♀♀♀♀♀♀ 6	♀♀♀♀♀♀♀♀♀♀♂ 16	♀♀♀♀♀♀♀♀♀♀♀♀ 26	♀♀♀♀♀♀♀♀♀♀♀♀♂ 36	♀♀♀♀♀♀♀♀♀♀♀♀♀♀ 46	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♂ 56
♀♀♀♀♀♀♀♀♀♀♀♀ 7	♀♀♀♀♀♀♀♀♀♀♀♀♂ 17	♀♀♀♀♀♀♀♀♀♀♀♀♀♀ 27	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♂ 37	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀ 47	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♂ 57
♀♀♀♀♀♀♀♀♀♀♀♀♀♀ 8	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♂ 18	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀ 28	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♂ 38	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀ 48	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♂ 58
♀♀♀♀♀♀♀♀♀♀♀♀♀♀ 9	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♂ 19	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀ 29	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♂ 39	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀ 49	♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♂ 59
♂ 10	♀♀ 20	♀♀♀♀ 30	♀♀♀♀♀♀ 40	♀♀♀♀♀♀♀♀ 50	

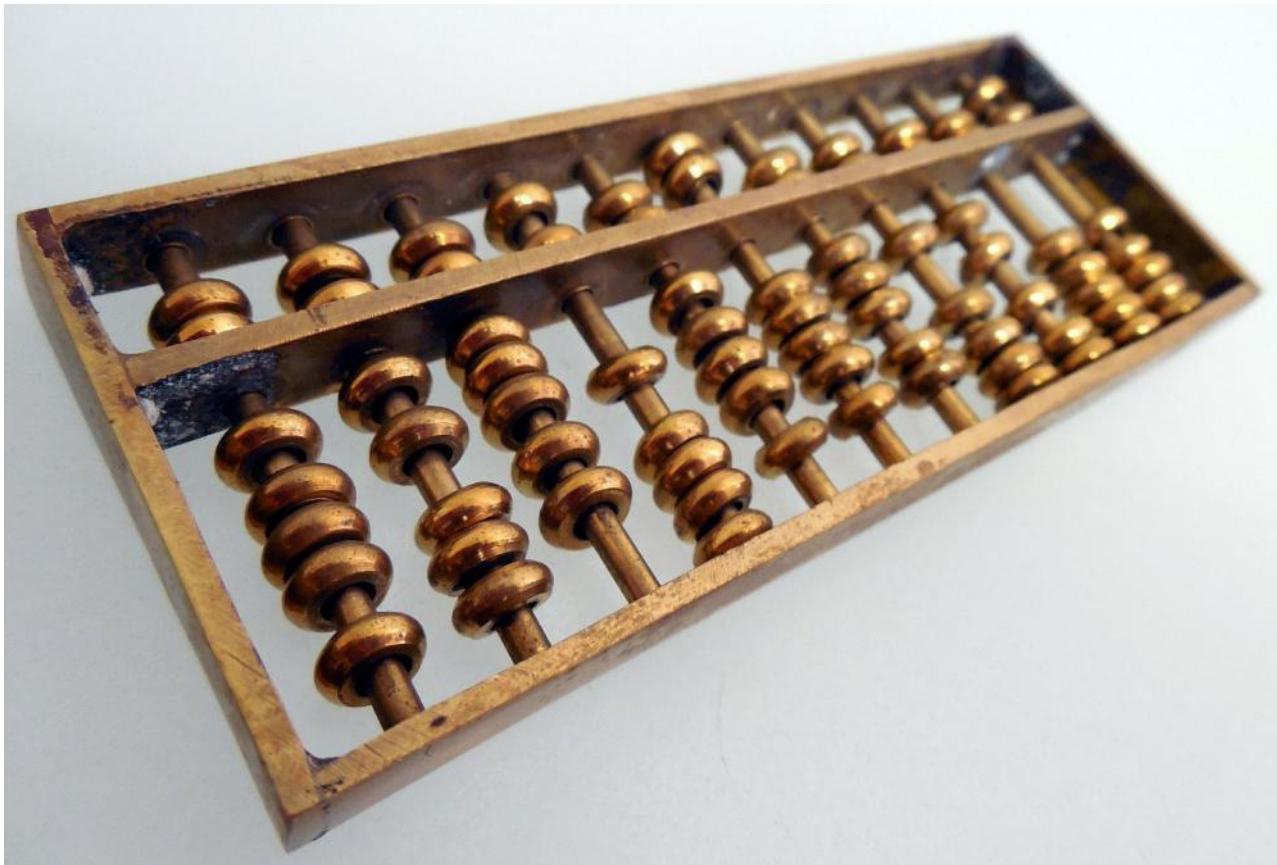
Babylonian cuneiform numerals

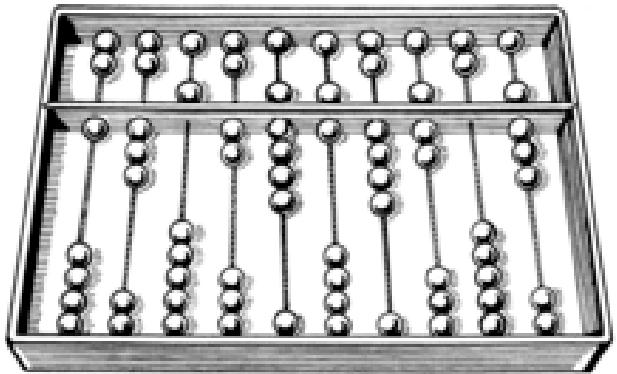


Uruk period: globular envelope with a cluster of accountancy tokens, from Susa. Louvre Museum

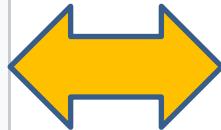


Representation of an Inca quipu





Bi-quinary coded decimal-like
abacus representing
1,352,964,708

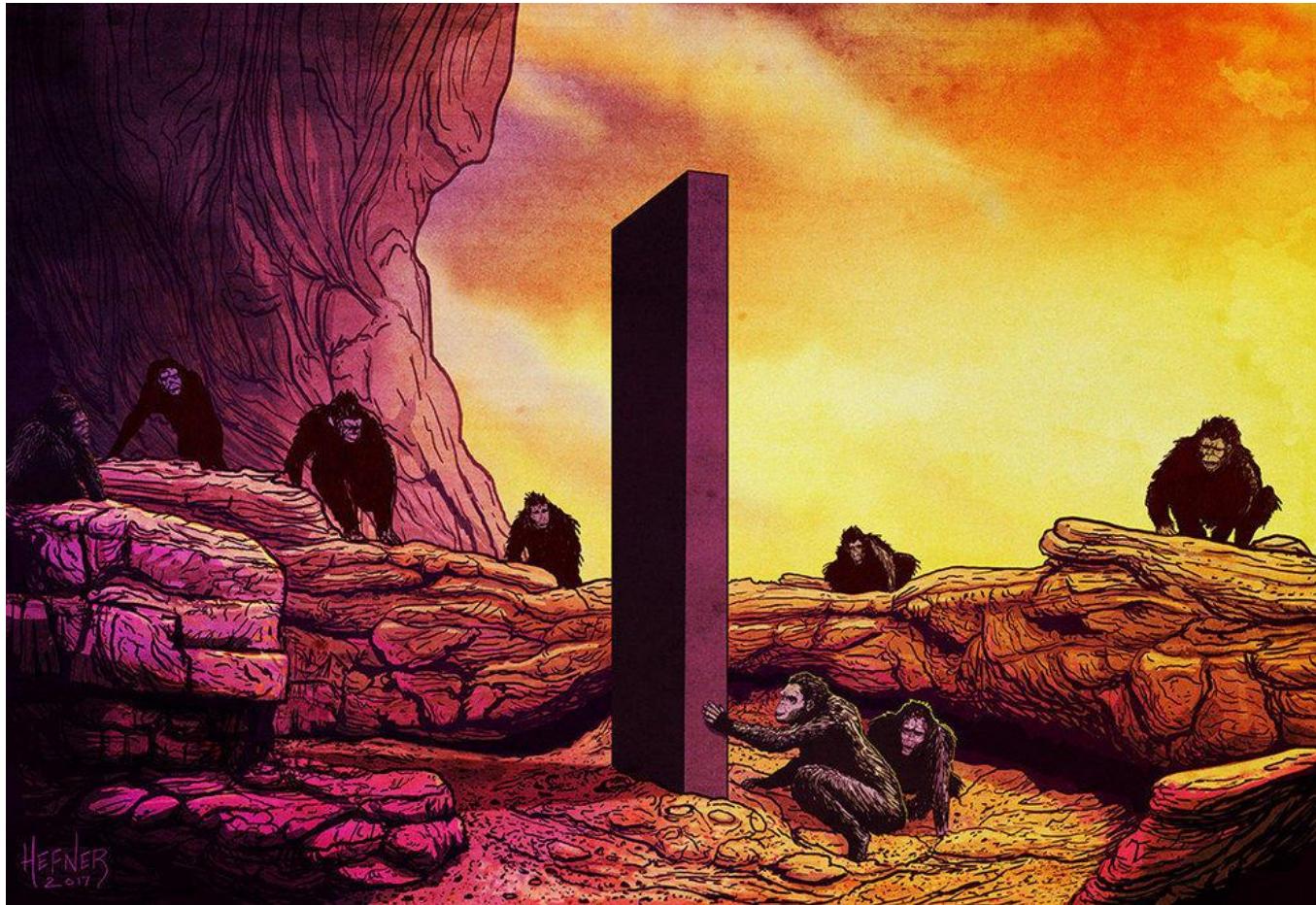


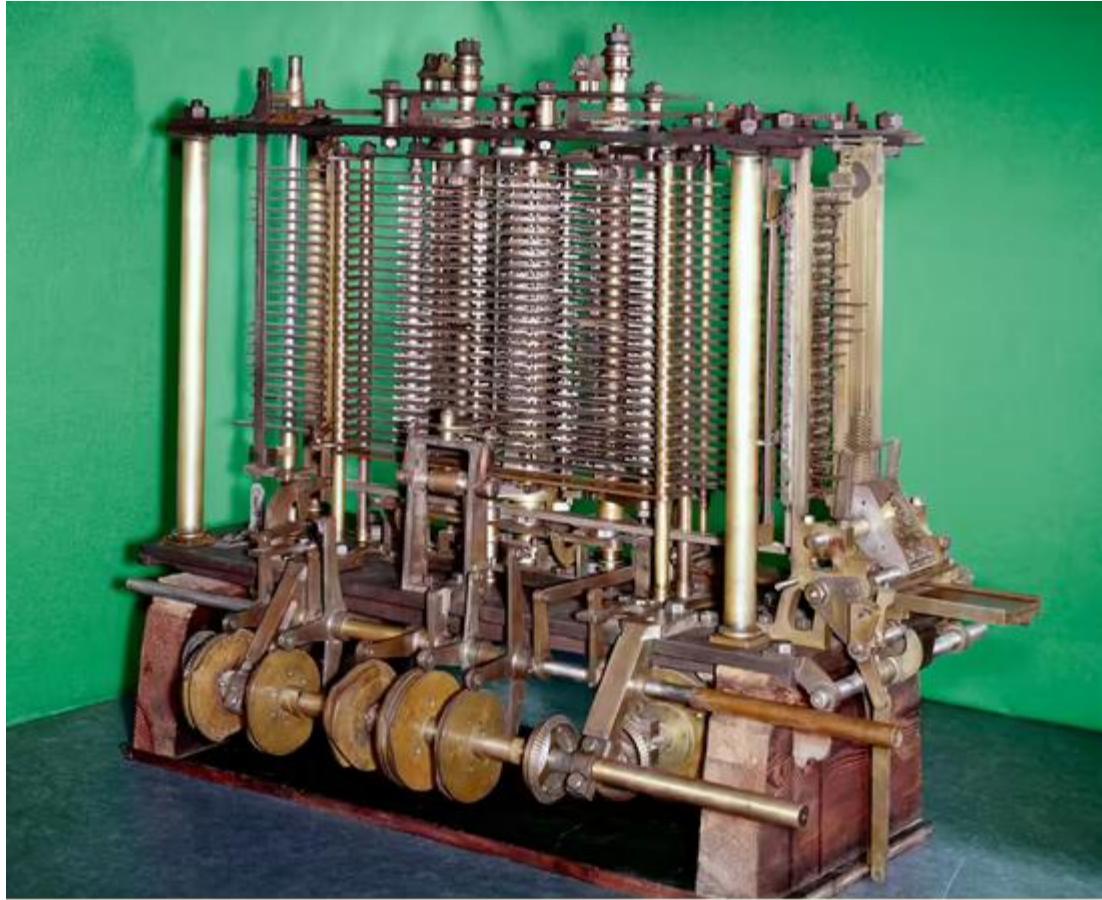
Decimal digit	5	0	4	3	2	1	0
0							
1							
2							
3							
4							
5							
6							
7							
8							
9							

Biquinary
code
example^[1]

Decimal digit	-	-	-	-	-	-	-
0							
1							
2							
3							
4							
5							
6							
7							
8							
9							

Reflected
biquinary
code





This analytical engine, conceived by Charles Babbage in 1834, was designed to calculate any mathematical formula and to have even higher powers of analysis than his original difference engine. This portion of the mill was under construction at the time of his death. SSPL/GETTY IMAGES



Difference Engine The completed portion of Charles Babbage's Difference Engine, 1832.

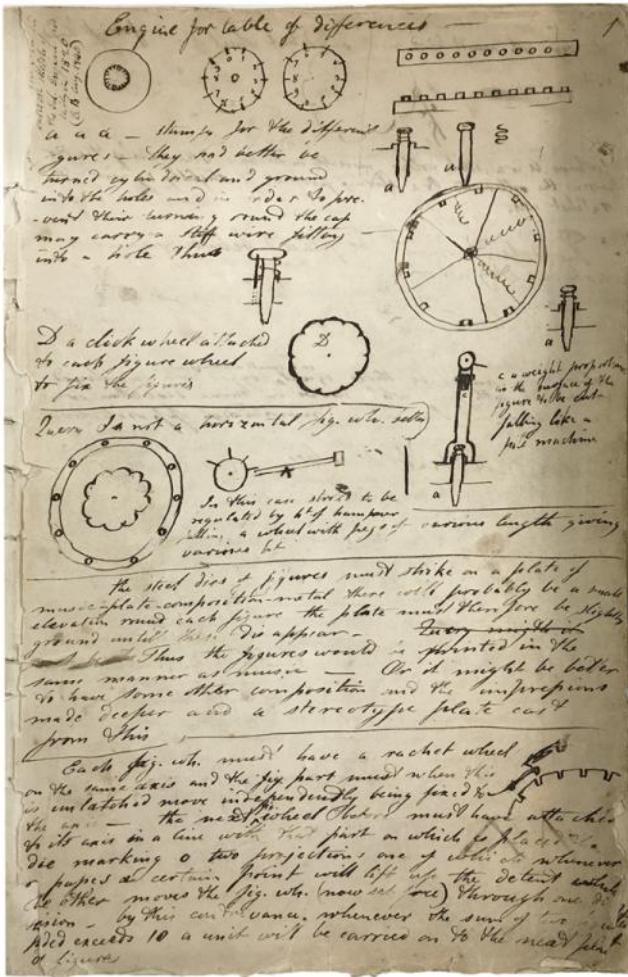


Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 *et seq.*)

Number of Operation.	Nature of Operation.	Variables acted upon.	Variables receiving results.	Indication of change in the value of any Variable.	Data										Working Variables.										Result Variables.			
					δV_2	δV_3	δV_4	δV_5	δV_6	δV_7	δV_8	δV_9	δV_{10}	δV_{11}	δV_{12}	δV_{13}	δV_{14}	δV_{15}	δV_{16}	δV_{17}	δV_{18}	δV_{19}	δV_{20}	δV_{21}	δV_{22}	δV_{23}		
1	\times	$\delta V_2 \times \delta V_3$	$\delta V_4, \delta V_5, \delta V_6$	$\{ \delta V_2 \\ \delta V_3 \\ \delta V_4 \\ \delta V_5 \\ \delta V_6 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
2	$-$	$\delta V_4 - \delta V_1$	δV_2	$\{ \delta V_4 \\ \delta V_1 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
3	$+$	$\delta V_5 + \delta V_1$	δV_2	$\{ \delta V_5 \\ \delta V_1 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
4	$+$	$\delta V_5 + \delta V_4$	δV_{11}	$\{ \delta V_5 \\ \delta V_4 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
5	$+$	$\delta V_{11} + \delta V_2$	δV_{11}	$\{ \delta V_{11} \\ \delta V_2 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
6	$-$	$\delta V_{12} - \delta V_2$	δV_{12}	$\{ \delta V_{12} \\ \delta V_2 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
7	$-$	$\delta V_3 - \delta V_1$	δV_{10}	$\{ \delta V_3 \\ \delta V_1 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
8	$+$	$\delta V_2 + \delta V_7$	δV_7	$\{ \delta V_2 \\ \delta V_7 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
9	$+$	$\delta V_6 + \delta V_7$	δV_{11}	$\{ \delta V_6 \\ \delta V_7 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
10	\times	$\delta V_{21} \times \delta V_{11}$	δV_{12}	$\{ \delta V_{21} \\ \delta V_{11} \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
11	$+$	$\delta V_{12} + \delta V_{13}$	δV_{12}	$\{ \delta V_{12} \\ \delta V_{13} \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
12	$-$	$\delta V_{10} - \delta V_1$	δV_{10}	$\{ \delta V_{10} \\ \delta V_1 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
13	$-$	$\delta V_6 - \delta V_1$	δV_6	$\{ \delta V_6 \\ \delta V_1 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
14	$+$	$\delta V_1 + \delta V_7$	δV_7	$\{ \delta V_1 \\ \delta V_7 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
15	$+$	$\delta V_6 + \delta V_7$	δV_8	$\{ \delta V_6 \\ \delta V_7 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
16	\times	$\delta V_8 \times \delta V_{11}$	δV_{11}	$\{ \delta V_8 \\ \delta V_{11} \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
17	$-$	$\delta V_6 - \delta V_1$	δV_6	$\{ \delta V_6 \\ \delta V_1 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
18	$+$	$\delta V_1 + \delta V_7$	δV_7	$\{ \delta V_1 \\ \delta V_7 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
19	$+$	$\delta V_6 + \delta V_7$	δV_9	$\{ \delta V_6 \\ \delta V_7 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
20	\times	$\delta V_{21} \times \delta V_{11}$	δV_{11}	$\{ \delta V_{21} \\ \delta V_{11} \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
21	\times	$\delta V_{21} \times \delta V_{11}$	δV_{12}	$\{ \delta V_{21} \\ \delta V_{11} \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
22	$+$	$\delta V_{12} + \delta V_{13}$	δV_{12}	$\{ \delta V_{12} \\ \delta V_{13} \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
23	$-$	$\delta V_{10} - \delta V_1$	δV_{10}	$\{ \delta V_{10} \\ \delta V_1 \}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			



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.				
							V_{11}	V_{12}	V_{13}	V_{14}	V_{15}	V_{16}	V_{17}	V_{18}	V_{19}	V_{110}	V_{111}	V_{112}	V_{113}	V_{114}	
1	\times	$V_{12} \times V_{13}$	V_{14}, V_{15}, V_{16}	$\{V_{11} = V_{12}$	$= 2n$	1	2	n	2n	2n	2n							B_1	B_1	B_1	
2	-	$-V_{14} - V_{15}$	V_{12}	$\{V_{11} = V_{12}$	$= 2n - 1$		1											B_2	B_2	B_2	
3	+	$+V_{14} + V_{15}$	V_{12}	$\{V_{11} = V_{12}$	$= 2n + 1$		1											B_3	B_3	B_3	
4	-	$-V_{14} - 2V_{14}$	V_{11}	$\{V_{11} = V_{14}$	$= 2n - 1$													B_4	B_4	B_4	
5	-	$-V_{12} - V_{13}$	V_{11}	$\{V_{11} = V_{12}$	$= 2n - 1$													B_5	B_5	B_5	
6	-	$-V_{12} - 2V_{12}$	V_{12}	$\{V_{11} = V_{12}$	$= -\frac{1}{2} \cdot 2n - 1 = \Lambda_2$													B_6	B_6	B_6	
7	-	$-V_{12} - V_{13}$	V_{13}	$\{V_{11} = V_{12}$	$= n - 1 (= 3)$		1		n												
8	+	$+V_{12} + V_{13}$	V_{12}	$\{V_{11} = V_{12}$	$= 2 + 0 = 2$			2													
9	+	$+V_{12} + V_{13}$	V_{13}	$\{V_{11} = V_{13}$	$= 2n$																
10	\times	$V_{13} \times V_{12}$	V_{12}	$\{V_{11} = V_{12}$	$= B_2 \cdot \frac{2n}{2} = B_2 \Lambda_1$												B_7	B_7	B_7		
11	+	$+V_{12} + V_{13}$	V_{12}	$\{V_{11} = V_{12}$	$= -\frac{1}{2} \cdot 2n - 1 + B_2 \cdot \frac{2n}{2}$																
12	-	$-V_{12} - V_{13}$	V_{10}	$\{V_{11} = V_{12}$	$= n - 2 (= 2)$		1														
13	-	$-V_{14} - V_{15}$	V_{14}	$\{V_{11} = V_{14}$	$= 2n - 1$		1														
14	$+V_{14} + 3V_{17}$	V_{17}		$\{V_{11} = V_{14}$	$= 2 + 1 = 3$		1														
15	$+V_{14} + V_{17}$	V_{14}		$\{V_{11} = V_{17}$	$= \frac{2n - 1}{3}$																
16	\times	$V_{14} \times V_{13}$	V_{13}	$\{V_{11} = V_{13}$	$= \frac{2n - 2n - 1}{3}$																
17	-	$-V_{14} - V_{15}$	V_{15}	$\{V_{11} = V_{15}$	$= 2n - 2$		1														
18	$+V_{14} + 3V_{17}$	V_{17}		$\{V_{11} = V_{17}$	$= 3 + 1 = 4$		1														
19	$+V_{14} - V_{17}$	V_{14}		$\{V_{11} = V_{14}$	$= \frac{2n - 2}{4}$																
20	\times	$V_{14} \times V_{13}$	V_{11}	$\{V_{11} = V_{13}$	$= \frac{2n - 2n - 1}{4}$																
21	\times	$V_{12} \times V_{13}$	V_{12}	$\{V_{11} = V_{12}$	$= B_2 \cdot \frac{2n - 2n - 1}{3} = B_2 \Lambda_2$																
22	$+V_{12} + 2V_{12}$	V_{12}		$\{V_{11} = V_{12}$	$= \Lambda_2 + B_2 \Lambda_1 + B_2 \Lambda_2$																
23	$-V_{12} - V_{13}$	V_{10}		$\{V_{11} = V_{12}$	$= n - 3 (= -1)$		1														
24	$+V_{12} + 3V_{17}$	V_{17}		$\{V_{11} = V_{17}$	$= B_7$																
25	$+V_{12} + V_{13}$	V_{12}		$\{V_{11} = V_{12}$	$= n + 1 - 4 + 1 = 5$		1		n + 1			0	0								

Here follows a repetition of Operations thirteen to twenty-three.

$$B_m = 1 - \sum_{k=0}^{m-1} \binom{m}{k} \frac{B_k}{m-k+1}$$

$$B_0 = 1$$

$$\tan x = \sum_{n=1}^{\infty} \frac{(-1)^{n-1} 2^{2n} (2^{2n} - 1) B_{2n}}{(2n)!} x^{2n-1}, \quad |x| < \frac{\pi}{2}.$$

$$\cot x = \frac{1}{x} \sum_{n=0}^{\infty} \frac{(-1)^n B_{2n} (2x)^{2n}}{(2n)!}, \quad 0 < |x| < \pi.$$

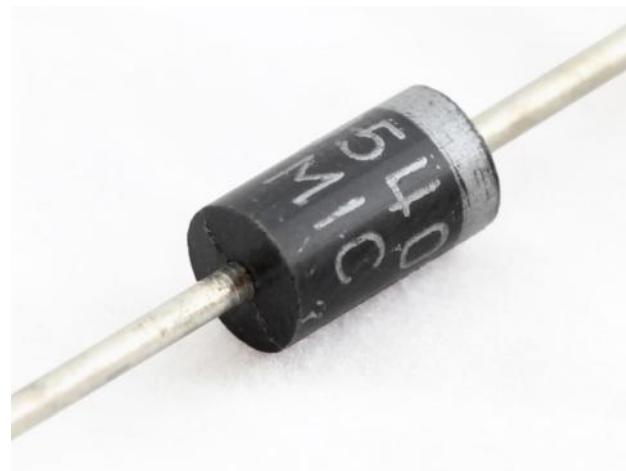
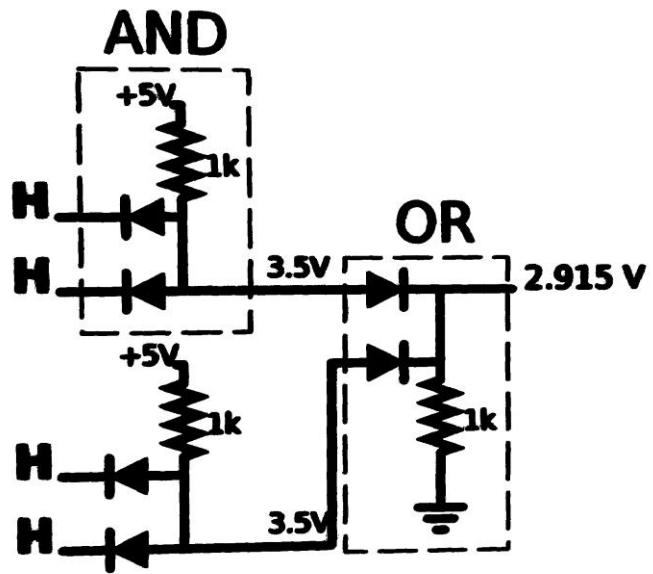
$$\tanh x = \sum_{n=1}^{\infty} \frac{2^{2n} (2^{2n} - 1) B_{2n}}{(2n)!} x^{2n-1}, \quad |x| < \frac{\pi}{2}.$$

$$\coth x = \frac{1}{x} \sum_{n=0}^{\infty} \frac{B_{2n} (2x)^{2n}}{(2n)!}, \quad 0 < |x| < \pi.$$

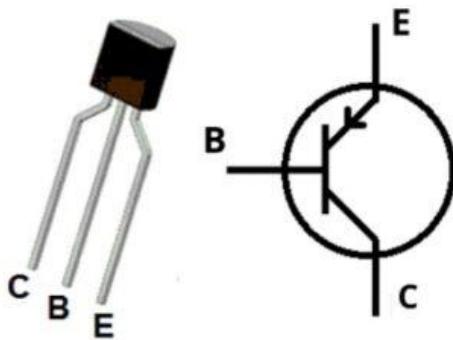
Diagram for the computation by the Engine of the

Number of Operation.	Nature of Operations.	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results.	Data					
						1V_1	1V_2	1V_3	1V_4	1V_5	1V_6
1	X	${}^1V_2 \times {}^1V_3$	${}^2V_4, {}^1V_4, {}^1V_5$	$\left\{ \begin{array}{l} {}^1V_7 = {}^1V_7 \\ {}^1V_8 = {}^1V_3 \end{array} \right.$	$= 2n$	1	2	n			
2	-	${}^1V_4 - {}^1V_1$	${}^2V_4, \dots$	$\left\{ \begin{array}{l} {}^1V_4 = {}^1V_4 \\ {}^1V_5 = {}^1V_4 \end{array} \right.$	$= 2n - 1$		1			$2n - 1$	
3	+	${}^1V_4 + {}^1V_1$	${}^2V_2, \dots$	$\left\{ \begin{array}{l} {}^1V_3 = {}^2V_2 \\ {}^1V_4 = {}^1V_3 \end{array} \right.$	$= 2n + 1$		1				$2n + 1$
4	+	${}^1V_3 - {}^1V_4$	${}^1V_{11}, \dots$	$\left\{ \begin{array}{l} {}^1V_3 = {}^0V_3 \\ {}^1V_4 = {}^0V_4 \end{array} \right.$	$= \frac{2n - 1}{2n + 1}$					0	0
5	+	${}^1V_{11} - {}^1V_7$	${}^2V_{11}, \dots$	$\left\{ \begin{array}{l} {}^1V_{11} = {}^2V_{11} \\ {}^1V_7 = {}^1V_2 \end{array} \right.$	$= \frac{1 - 2n - 1}{2n + 1}$			2			
6	-	${}^1V_{13} - {}^2V_{13}$	${}^1V_{13}, \dots$	$\left\{ \begin{array}{l} {}^1V_{13} = {}^0V_{13} \\ {}^2V_{13} = {}^1V_{13} \end{array} \right.$	$= -\frac{1}{2} \cdot \frac{2n - 1}{2n + 1} = A_3$						
7	-	${}^1V_2 - {}^1V_1$	${}^2V_{10}, \dots$	$\left\{ \begin{array}{l} {}^1V_2 = {}^1V_3 \\ {}^1V_1 = {}^1V_2 \end{array} \right.$	$= n - 1 (= 3)$		1		n		

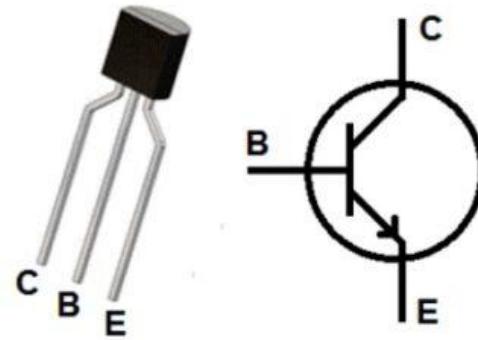
VOLTAGE DROPS ACROSS DIODES WOULD BE A FURTHER CONSTRAINT



PNP Transistor

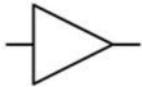


NPN Transistor



TRANSISTOR SYMBOL

Buffer



Input	Output
0	0
1	1

AND



A	B	Output
0	0	0
1	0	0
0	1	0
1	1	1

OR



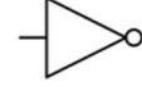
A	B	Output
0	0	0
1	0	1
0	1	1
1	1	1

XOR



A	B	Output
0	0	0
1	0	1
0	1	1
1	1	0

Inverter



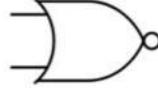
Input	Output
0	1
1	0

NAND



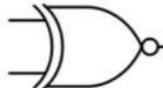
A	B	Output
0	0	1
1	0	1
0	1	1
1	1	0

NOR



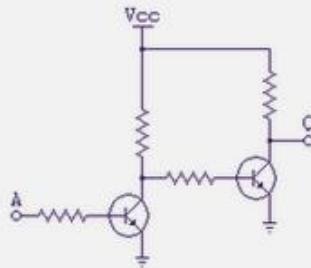
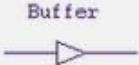
A	B	Output
0	0	1
1	0	0
0	1	0
1	1	0

XNOR

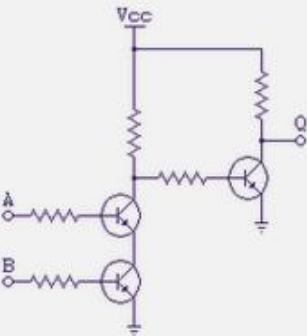
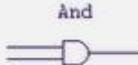


A	B	Output
0	0	1
1	0	0
0	1	0
1	1	1

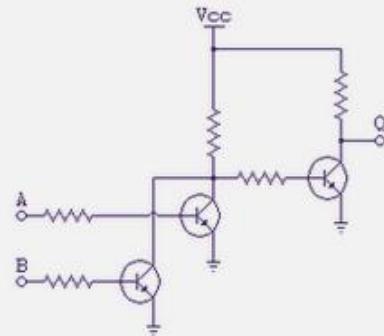
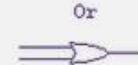
Buffer



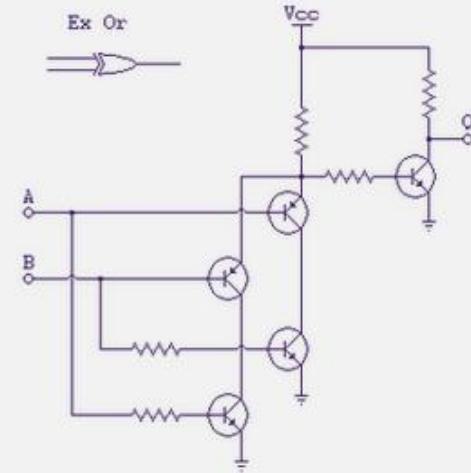
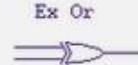
And



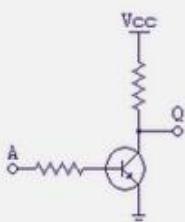
Or



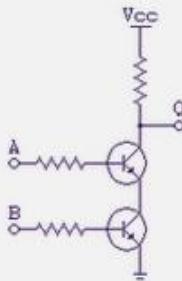
Ex Or



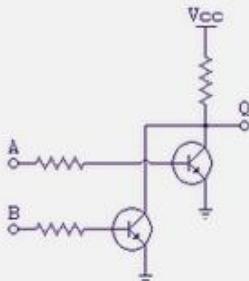
Inverter Not



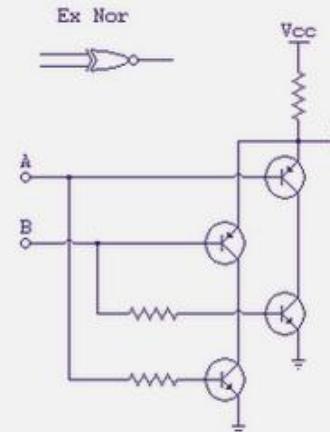
Nand

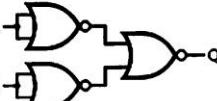
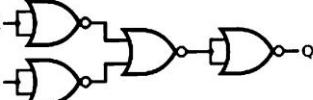
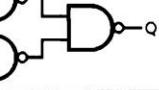
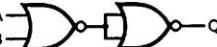
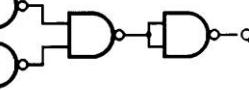
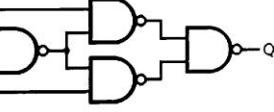
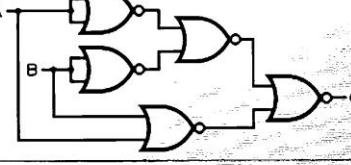
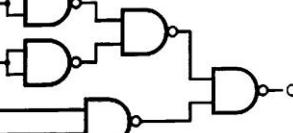
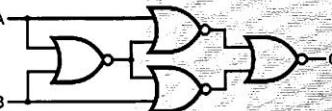


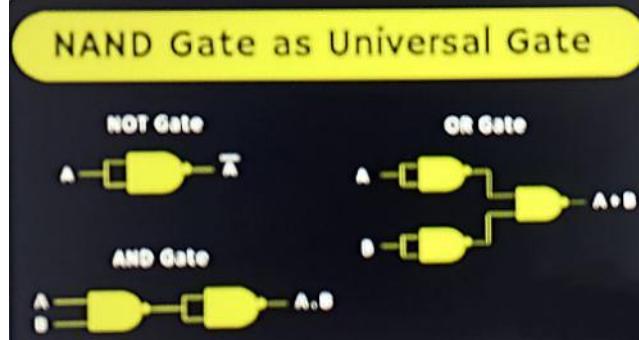
Nor

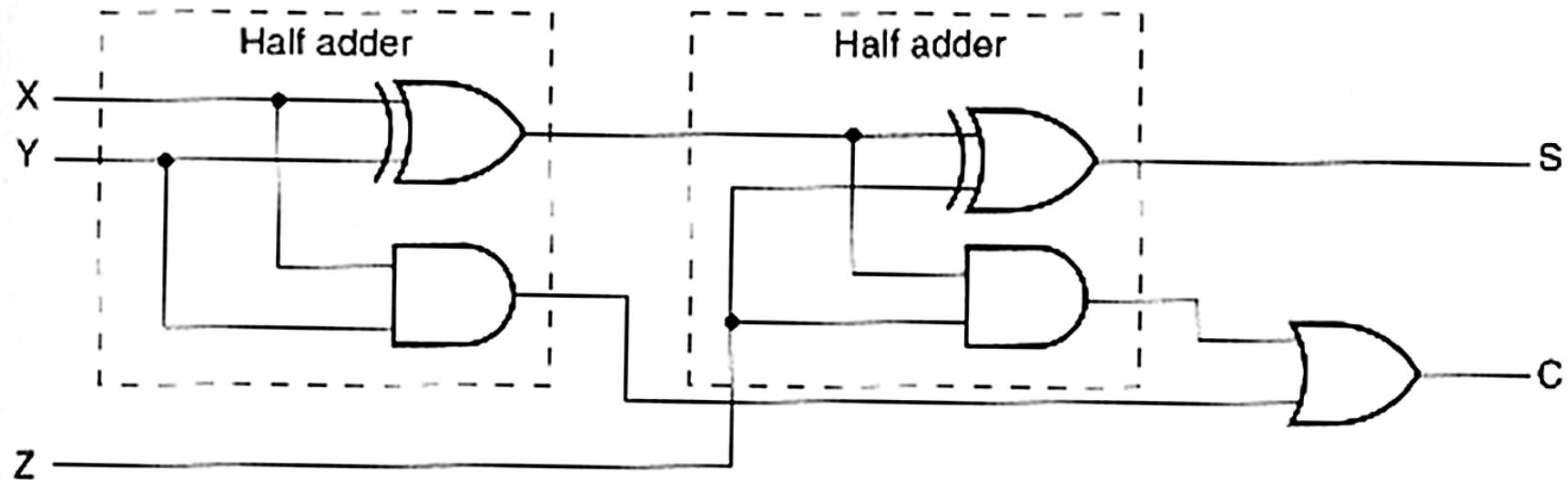


Ex Nor

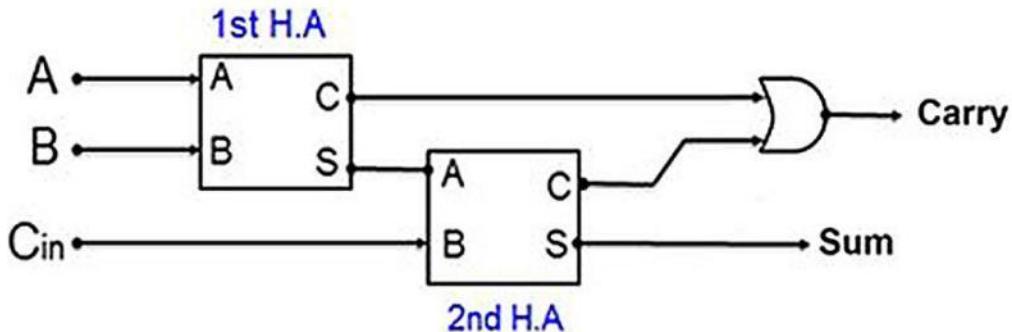


type	NAND construction	NOR construction
NOT		
AND		
NAND		
OR		
NOR		
XOR		
XNOR		



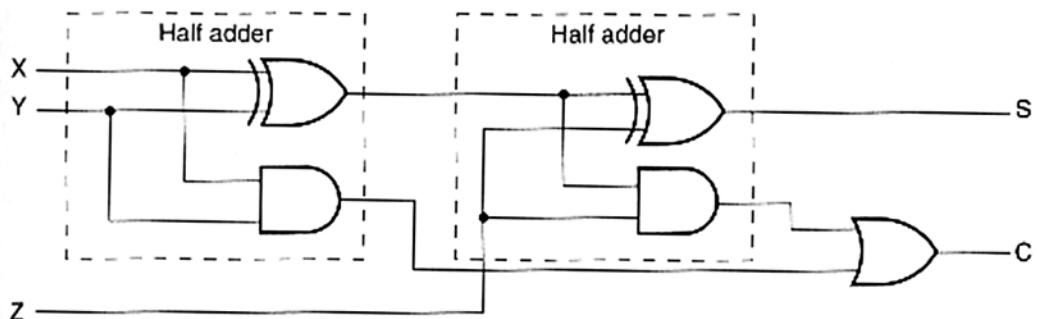


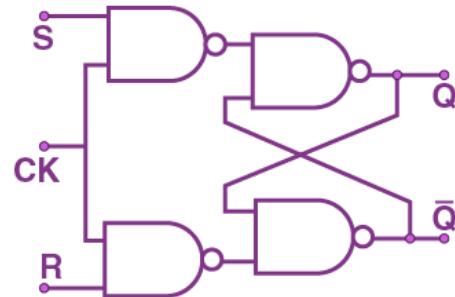
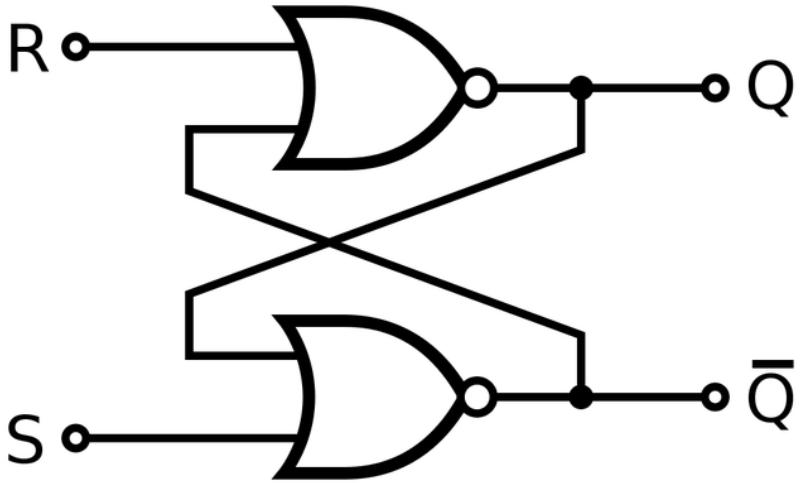
(a)



(b)

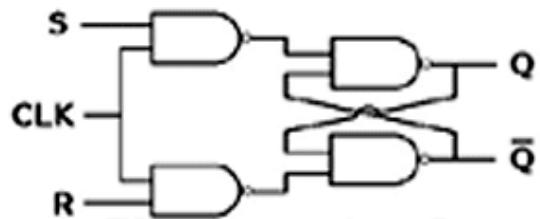
Input			Output	
A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



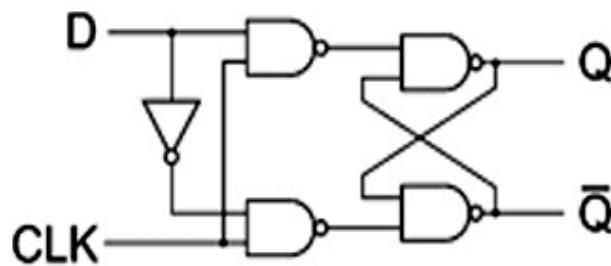
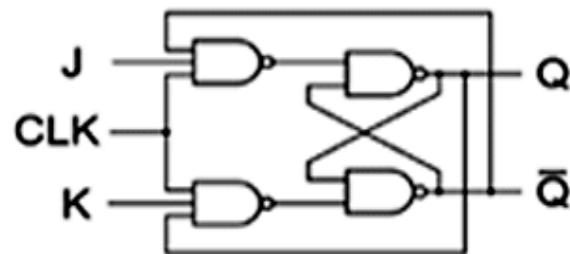


Truth Table

S	R	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-



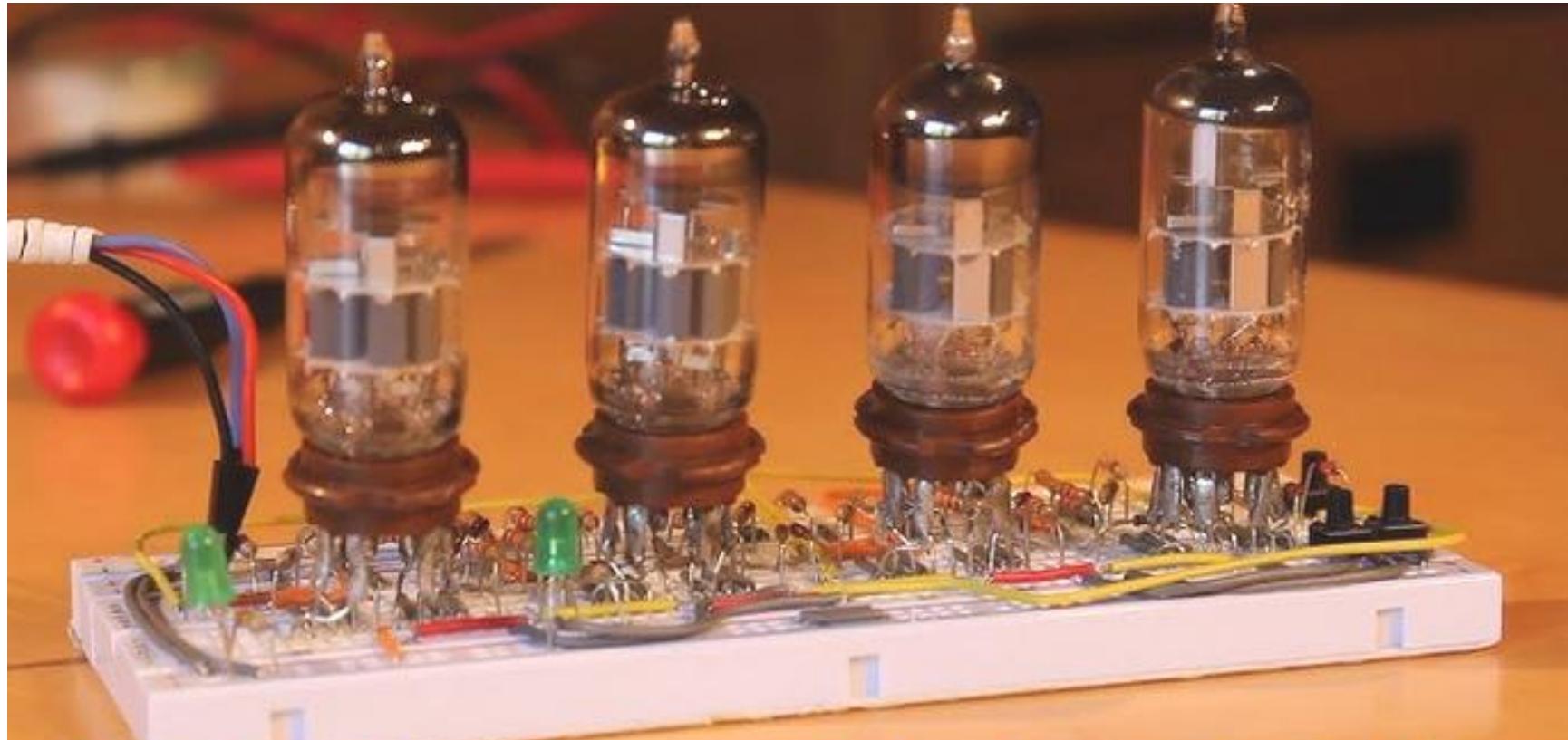
www.Electronicsforu.com



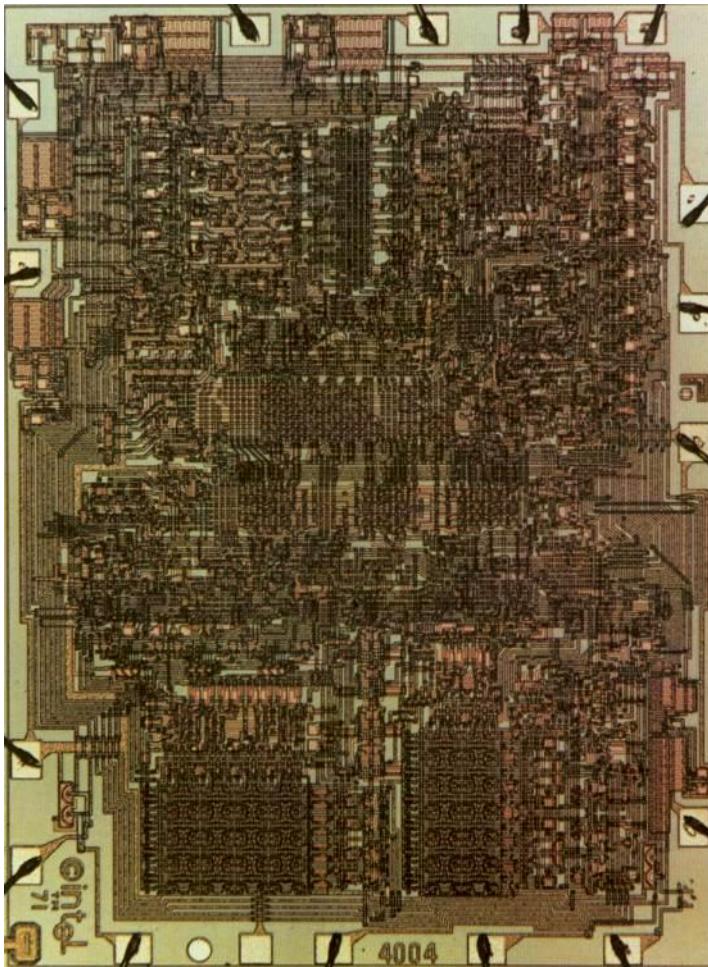
S	R	Q	Q'
0	0	0	1
0	1	0	1
1	0	1	0
1	1	∞	∞

J	K	Q	Q'
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	1
0	1	1	0
1	0	1	1
1	1	1	0

Clock	D	Q	Q'
$\downarrow 0$	0	0	1
$\uparrow 1$	0	0	1
$\downarrow 0$	1	0	1
$\uparrow 1$	1	1	0



Let's Build a Vacuum Tube 1-Bit Full Adder



INTEL 4004

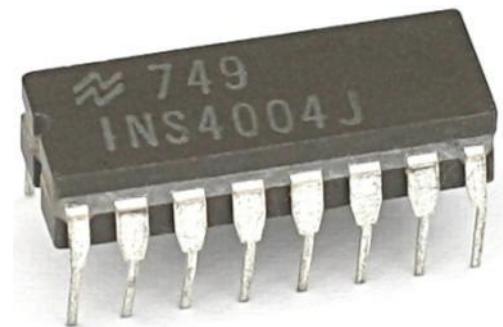
16-pin CPU for desktop calculators

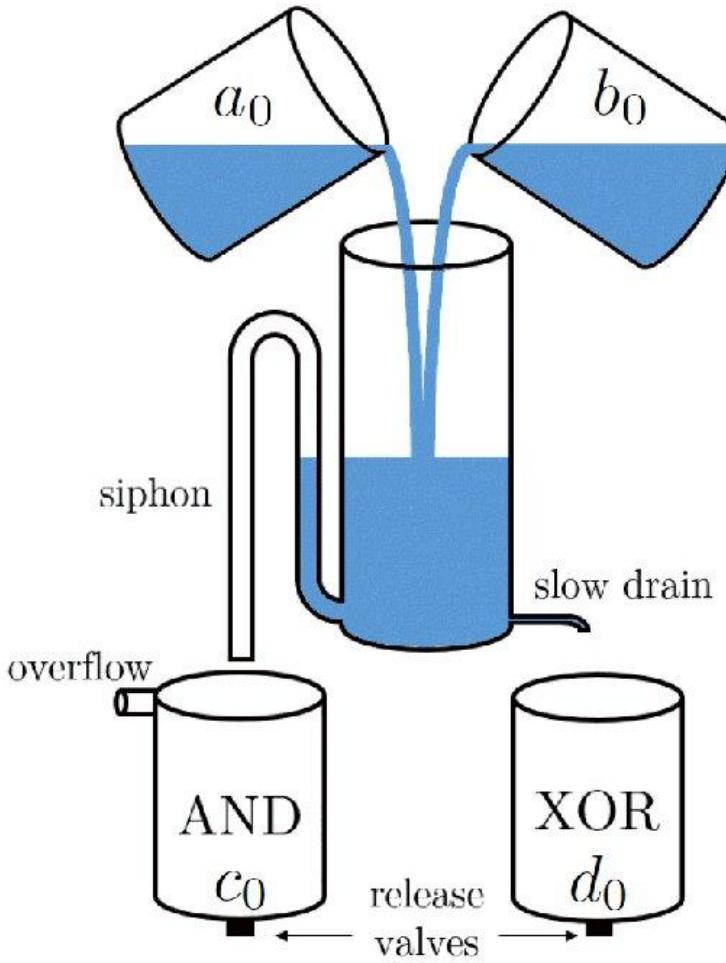
4-bit addressing / instruction set

@ 740 kHz clock

Available: 15-Nov-1971

1st ever single-chip CPU





Watch: <https://www.youtube.com/watch?v=lxXaizglscw>



NASA's Saturn V, first test launch in Nov. 9, 1962, with D-17B flight computer on board.

Μέρος II: Ψηφιακή τεχνολογία επεξεργαστών

1. Ιστορική εξέλιξη – Οι πρώτοι επεξεργαστές.
2. Δεδομένα και κώδικας στην κεντρική μνήμη.
3. Μικρο-ελεγκτές (micro-controllers).
4. Οι σημερινές δυνατότητες.



HISTORY OF COMPUTING

A BRIEF TIMELINE OF THE EVENTS THAT CREATED THE DIGITAL WORLD WE LIVE IN TODAY

Written by Timothy Williamson

The history of computers goes back over 200 years. At first theorised by mathematicians and entrepreneurs, during the 19th century mechanical calculating machines were designed and built to solve increasingly complex number-crunching challenges.

The advancement of technology enabled ever more-complex computers by the early 20th century, and computers became larger and more powerful.

Today, computers are almost unrecognisable from the designs of the 19th century such as Charles Babbage's Analytical Engine – or even from the huge computers of the 20th century that occupied whole rooms, such as the Electronic Numerical Integrator and Computer.

Here's a brief history of computers, from their primitive number-crunching origins to the powerful modern-day machines that run the Internet, run games and stream multimedia.

Timeline:

- 1801: Joseph Marie Jacquard invents a punch-card loom similar to early punch card computers.
- 1821: Swedish inventor Per Georg Scheutz and his son Edward build the world's first printing calculator.
- 1842: Herman Hollerith designs a punched-card system to help calculate the 1890 US Census. The machine saves the government years of calculations and approximately \$5 million.
- 1846: Vannevar Bush invents and builds the Differential Analyzer, the first large-scale automatic general-purpose mechanical analog computer.
- 1877: John Vincent Atanasoff submits a grant proposal to build the first electric-only computer.
- 1930: David Packard and Bill Hewlett found the Hewlett-Packard Company in Palo Alto, California.
- 1941: Atanasoff and his graduate student, Clifford Berry, design the first digital electronic computer in the US, called the Atanasoff-Berry Computer (ABC). This marks the first time a computer is able to store information on its main memory. It is capable of performing one operation every 15 seconds.
- 1945: First Colossus, designed by Tommy Flowers, becomes operational at Bletchley Park, cracking German Lorenz ciphers in hours instead of weeks. At Harvard University, the Harvard Mark I is completed by IBM, a room-sized, relay-based calculator.
- 1946: John Mauchly and J. Presper Eckert design and build the Electronic Numerical Integrator and Calculator (ENIAC), the first automatic, general-purpose, electronic, decimal, digital computer.

1821 THE DIFFERENCE ENGINE
English mathematician Charles Babbage conceives of a steam-driven calculating machine that would be able to compute tables of numbers. Funded by the British government, the project, called the Difference Engine, fails due to the lack of technology at the time.

1848 THE FIRST PROGRAMMER
Ada Lovelace, an English mathematician and the daughter of poet Lord Byron, writes the world's first computer programme. Lovelace does this while translating a paper on Charles Babbage's Analytical Engine from an Italian translation. Her annotations, which she calls "notes", are three times longer than Babbage's original paper. Included is a step-by-step guide to the computation of Bernoulli numbers (a sequence of rational numbers often used in computation) with Babbage's machine, an algorithm that makes her the first computer programmer.

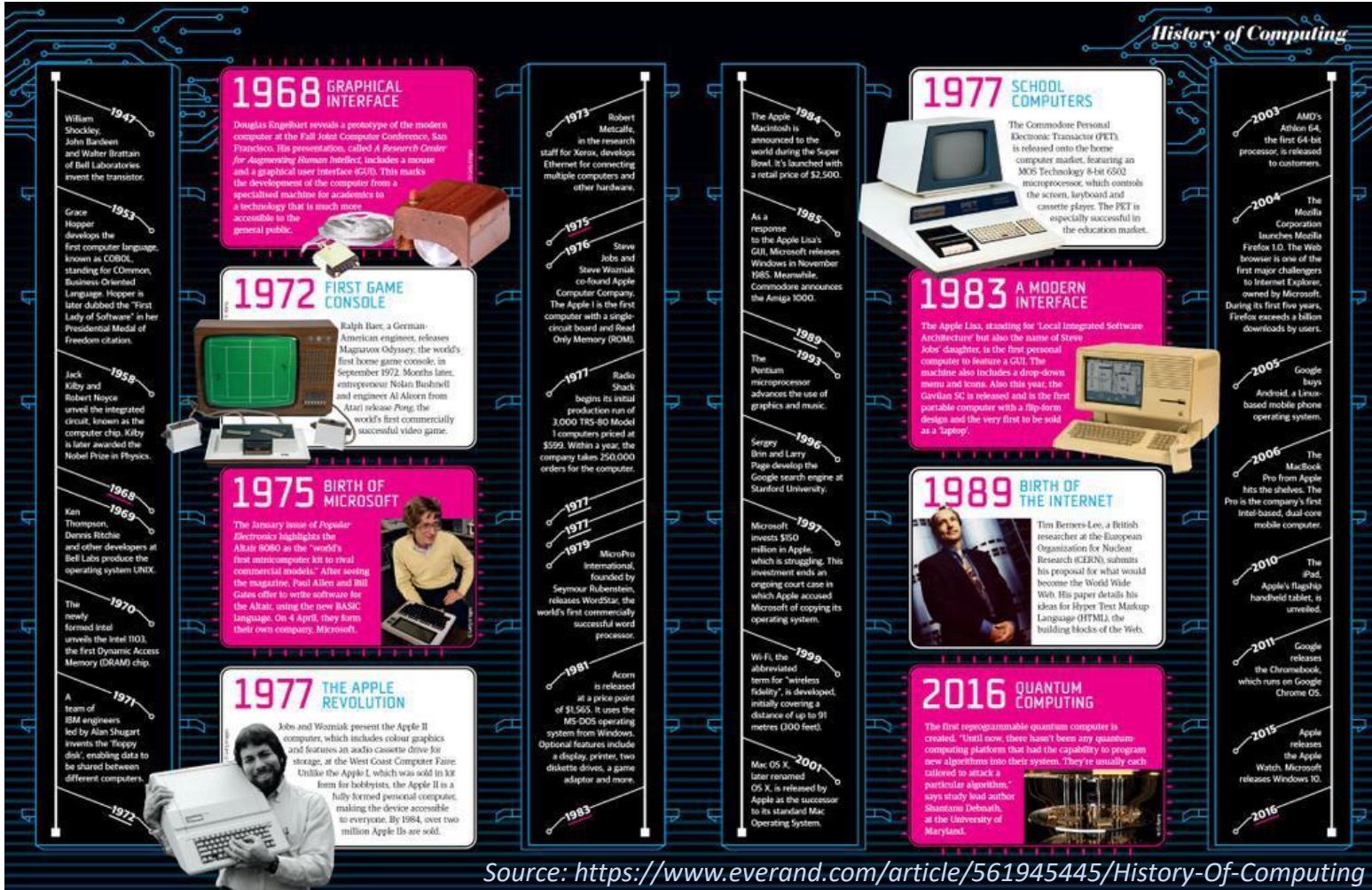
1936 THE TURING MACHINE
Alan Turing, a British scientist and mathematician, presents the principle of a universal machine, later called the Turing machine, in a paper called *On Computable Numbers*. Turing machines are capable of computing anything that is computable. The central concept of the modern computer is based on his idea. Turing is later involved in the development of the Turing-Welchman Bombe, an electro-mechanical device designed to decipher Nazi codes during World War II.

1941 THE Z3 MACHINE
German inventor and engineer Konrad Zuse completes his Z3 machine, the world's earliest digital computer. The machine was destroyed during a bombing raid on Berlin during World War II. Zuse fled the German capital after the defeat of Nazi Germany and later released the world's first commercial digital computer, the Z4, in 1950.

History of Computing

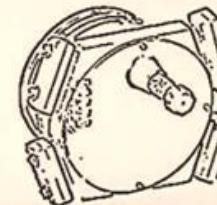
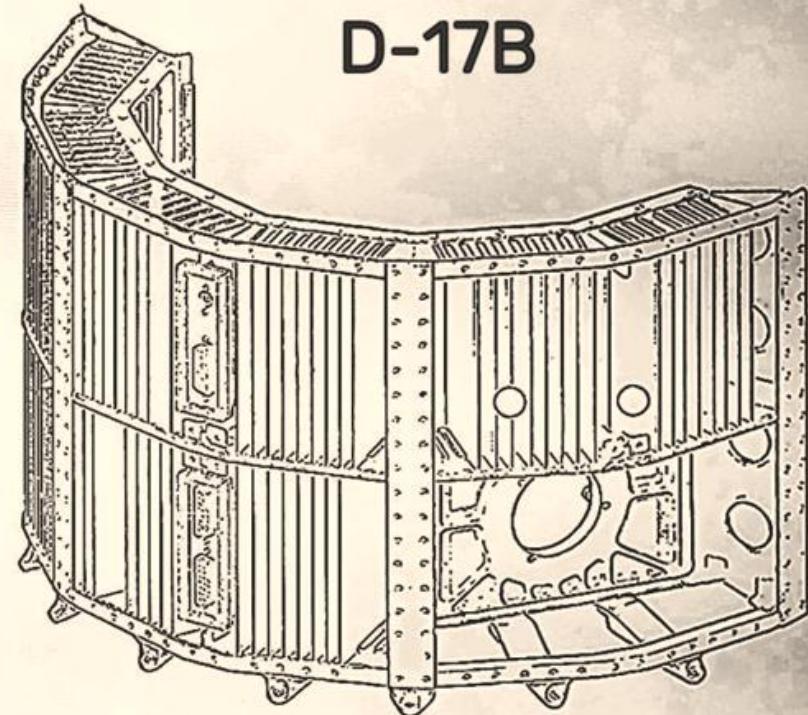
Source: <https://www.everand.com/article/561945445/History-Of-Computing>

History of Computing



Manufacturer	Autonetics
Year	1962
Type	Serial, synchronous
Number system	Binary, fixed point, sign plus 2's complement
Logic levels	False (0 volts), True (-10 volts), negative logic
Data word length	24 bits (full word) 11 bits (split word)
Instruction word length	24 bits
Number of instructions	39
Execution times	
Add	78.125 μ sec
Multiply	1015.625 μ sec
Divide	Software
Clock frequency	345.6 kHz
Addressing	Direct addressing Two-address (unflagged) Three-address (flagged)
Memory	Ferrous-oxide coated disk Non-destructive readout 2727 (24 bit) word capacity 78.125 μ sec cycle time
Input/Output	48 digital lines (input) 26 specialized incremental inputs 28 digital lines (output) 12 analog lines (output) 3 pulse lines (output) 25,600 words/sec maximum I/O transfer rate
Physical characteristics	
Dimensions	20 in. high, 29 in. diameter, 5 in. deep
Power	28 VDC at 25 A
Circuits	DRL and DTL
Weight	62 pounds

D-17B



430⁵

Computers in Spaceflight

The NASA Experience

James E. Tomayko

CONTRACT NASW-3714
MARCH 1988

(NASA-CR-182505) COMPUTERS IN SPACEFLIGHT: X88-10180
THE NASA EXPERIENCE (Wichita State Univ.)
409 p LIMIT USGA

Unclassified
B3/60 0130186



titude hold to go into effect when a restart occurred. This would be potentially dangerous if a restart began with the LEM close to the lunar surface. The solution was to give the crew responsibility to manually fire the engines during a restart if necessary¹⁷⁵.

Software development for the AGS followed a tightly controlled schedule:

1. 12.5 months before launch: NASA delivers the preliminary reference trajectory and mission requirements to TRW.
2. 11 months: Program specification and AGS performance analysis is complete.
3. 10.5 months: NASA conducts the Critical Design Review (CDR).
4. 8 months: The final mission reference trajectory is delivered.
5. 7 months: The equation test results, verification test plan, and preliminary program goes to NASA for approval.
6. 6.5 months: The First Article Configuration Inspection (FACI) conducted.
7. 5 months: The verified program and documentation is delivered to NASA.
8. 4.5 months: NASA conducts the Customer Acceptance Readiness Review (CARR).
9. 3 months: The operational flight trajectory is delivered by NASA to the contractor.
10. 2 months: The final Flight Readiness Review (FRR) is held.
11. 1.5 months: The tape containing the final program is delivered¹⁷⁶.

One method of software verification was quite unique. To simulate motion and thus provide more realistic inputs to the computer, planners used a walk-in van containing the hardware and software. Technicians drove the van around Houston with the programs running inside it¹⁷⁷.

8088

8-Bit Microprocessor CPU
iAPX86 Family
FINAL

DISTINCTIVE CHARACTERISTICS

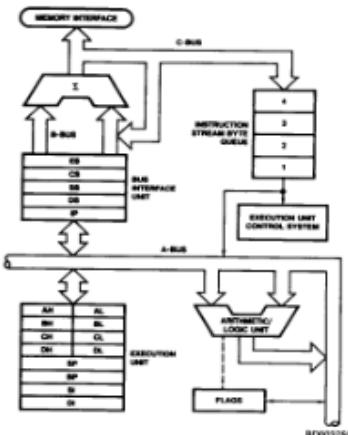
- 8-bit data bus, 16-bit internal architecture
 - Directly addresses 1 Mbyte of memory
 - Software compatible with 8086 CPU
 - Byte, word, and block operations
 - 24 operand addressing modes
 - Powerful instruction set
 - Efficient high level language implementation
 - Three speed options: 8MHz 8088
8MHz 8086-2
10MHz 8086-1

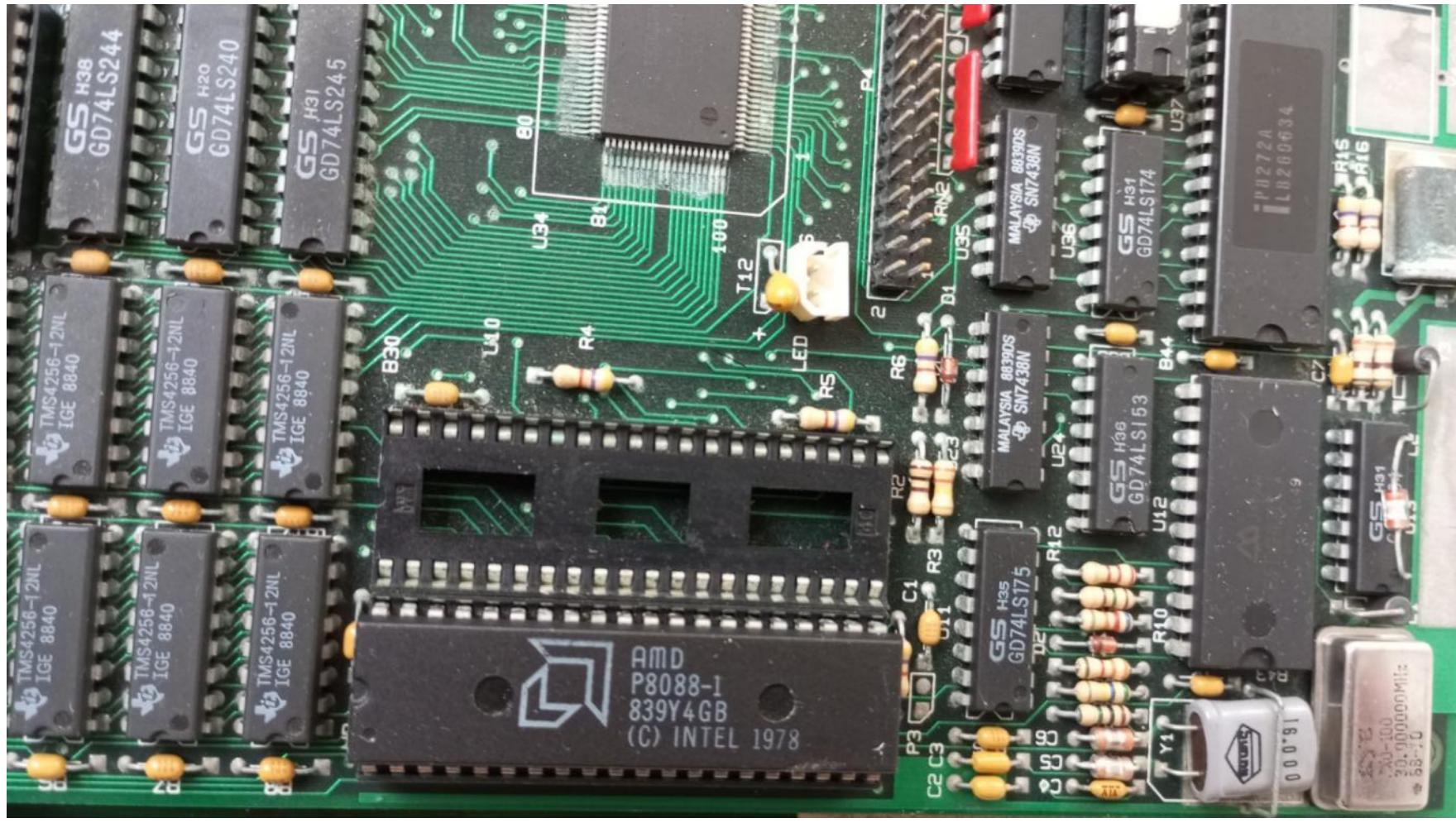
GENERAL DESCRIPTION

The 8086 CPU is an 8-bit processor designed around the 8086 internal structure. Most functions of the 8086 are identical to the equivalent 8086 functions. The pinout is slightly different. The 8086 handles the external bus the same way the 8089 does, but it handles only 8 bits at a time. Sixteen-word bytes are fetched or written in two consecutive bus cycles. Both processors will appear identical to the software engineer, with the exception of execution time.

The 8086 is made with N-channel silicon gate technology and is packaged in a 40-pin Plastic CERDIP or Plastic Leaded Chip Carriers.

BLOCK DIAGRAM





INSTRUCTION SET SUMMARY (continued)

ARITHMETIC

ADD = Add

Reg/memory with register to either

Immediate to register / memory

Immediate to accumulator

7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
0 0 0 0 0 d w	mod reg r/m		
1 0 0 0 0 0 s w	mod 0 0 0 r/m	data	data if s:w = 01
0 0 0 0 0 1 0 w	data	data if w = 1	

ADC = Add with carry:

Reg/memory with register to either

Immediate to register/memory

Immediate to accumulator

0 0 0 1 0 0 d w	mod reg r/m		
1 0 0 0 0 0 s w	mod 0 1 0 r/m	data	data if s:w = 01
0 0 0 1 0 1 0 w	data	data if w = 1	

INC = Increment:

Register/memory

Register

AAA = ASCII adjust for add

DAA = Decimal adjust for add

1 1 1 1 1 1 1 w	mod 0 0 0 r/m		
0 1 0 0 0 reg			
0 0 1 1 0 1 1 1			
0 0 1 0 0 1 1 1			

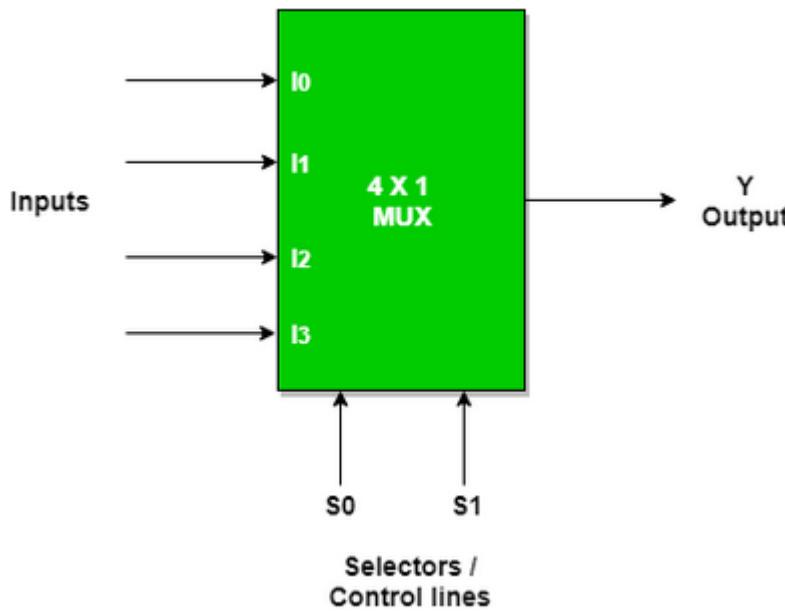
SUB = Subtract:

Reg/memory and register to either

Immediate from register/memory

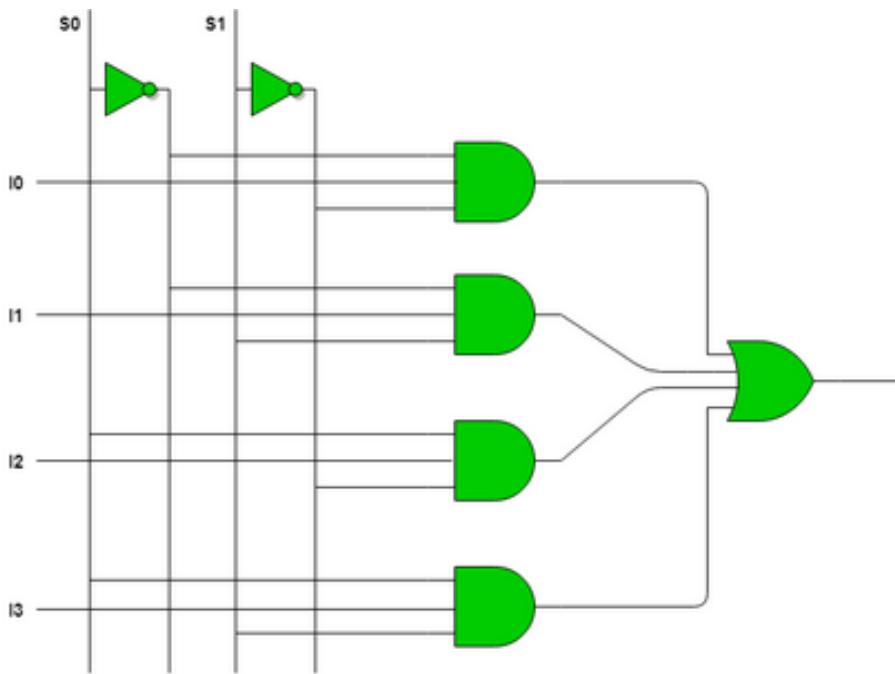
Immediate from accumulator

0 0 1 0 1 0 d w	mod reg r/m		
1 0 0 0 0 0 s w	mod 1 0 1 r/m	data	data if s:w = 01
0 0 1 0 1 1 0 w	data	data if w = 1	



S0	S1	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

- When $S1S0=00$, the input I0 is selected.
- When $S1S0=01$, the input I1 is selected.
- When $S1S0=10$, the input I2 is selected.
- When $S1S0=11$, the input I3 is selected.



MCS-51

The **Intel MCS-51** (commonly termed **8051**) is a single chip microcontroller (MCU) series developed by Intel in 1980 for use in embedded systems. The architect of the Intel MCS-51 instruction set was John H. Wharton.^{[1][2]} Intel's original versions were popular in the 1980s and early 1990s, and enhanced binary compatible derivatives remain popular today. It is a complex instruction set computer, but also has some of the features of RISC architectures, such as a large register set and register windows, and has separate memory spaces for program instructions and data.

Intel's original MCS-51 family was developed using N-type metal–oxide–semiconductor (NMOS) technology, like its predecessor Intel MCS-48, but later versions, identified by a letter C in their name (e.g., 80C51) use complementary metal–oxide–semiconductor (CMOS)

Intel 8051



Intel P8051 microcontroller

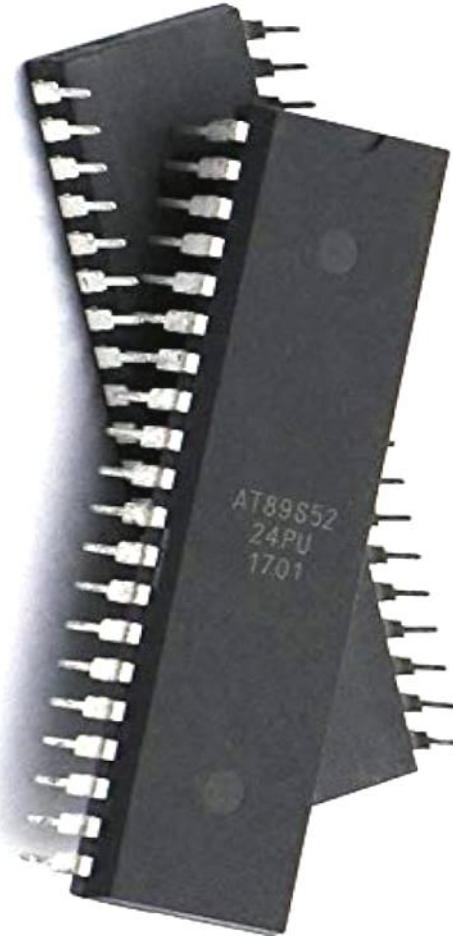
History

Predecessor(s) Intel MCS-48

Successor(s) Intel MCS-151

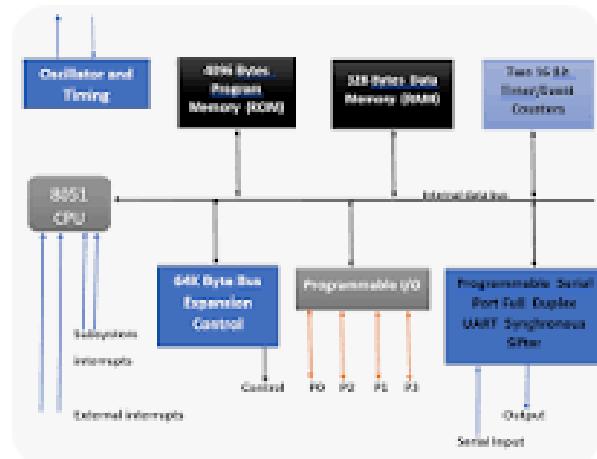
ATMEL8051 Microcontroller

Parameter	Value
Flash (Kbytes):	8 Kbytes
Max. Operating Frequency	24 MHz
CPU:	8051-12C
Max I/O Pins	32
UART:	1
SRAM (Kbytes):	0.25
Operating Voltage (Vcc):	4.0 to 5.5
Timers:	3
ISP:	SPI



The Key features of the 8051 Microcontroller –

- 4 KB on-chip ROM (Program memory).
- 128 bytes on-chip RAM (Data memory).
- The 8-bit data bus (bidirectional).
- 16-bit address bus (unidirectional).
- Two 16-bit timers.
- Instruction cycle of 1 microsecond with 12 MHz crystal.
- Four 8-bit input/output ports.
- 128 user-defined flags.

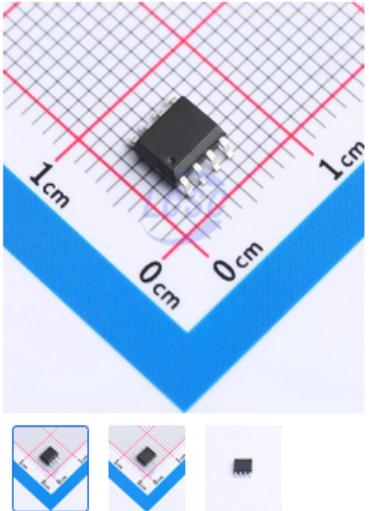


ATME8051 Microcontroller

6. Write an 8051 C program to toggle bits of P1 ports continuously with a 250 ms.

```
#include <reg51.h>
void MSDelay(unsigned int); //delay routine definition
void main()
{
    while (1) //repeat forever
    {
        P1=0x55;
        MSDelay(250);
        P1=0xAA;
        MSDelay(250);
    }
}

// delay routine implementation
void MSDelay(unsigned int itime)
{
    unsigned int i, j;
    for (i = 0; i < itime; i++)
        for (j = 0; j < 1275; j++);
}
```



Images are for reference only

Add to Favourites

Recommended For You

PADAUK Tech PMS150C-S08



Manufacturer [PADAUK Tech](#)

Mfr. Part # PMS150C-S08

LCSC Part # C129127

Package SOP-8

Customer #

Datasheet [PADAUK Tech PMS150C-S08](#)

EasyEDA [EasyEDA Model](#)

ECCN -

Description 1KB 64Byte OTP 6.2V~5.5V RISC SOP-8 Microcontroller Units (MCUs/MPUs/SOCs)
ROHS

In Stock: 3330

WHS-HK: 0

WHS-ZH: 3330

WHS-JS: 0

Stock Notification [?](#)

[Inquiry for more quantity](#)

Minimum : 10 Multiples : 10

- 10 +

[Add to Cart](#)

[Save to Project](#)

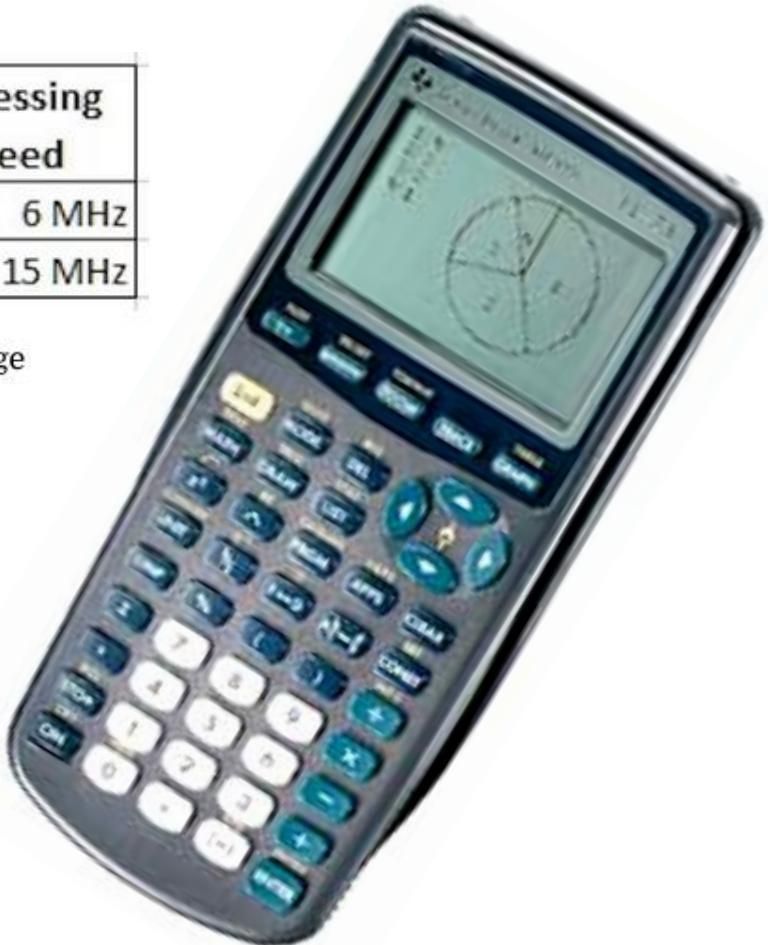
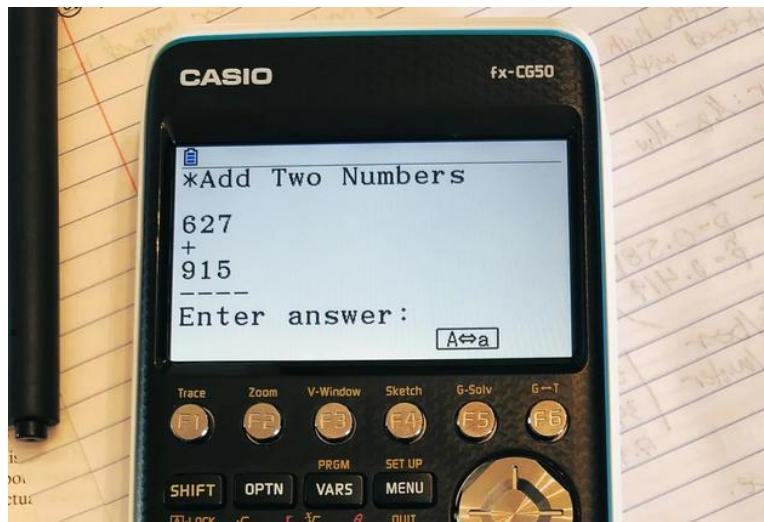
Can Ship in 1-2 Business Days

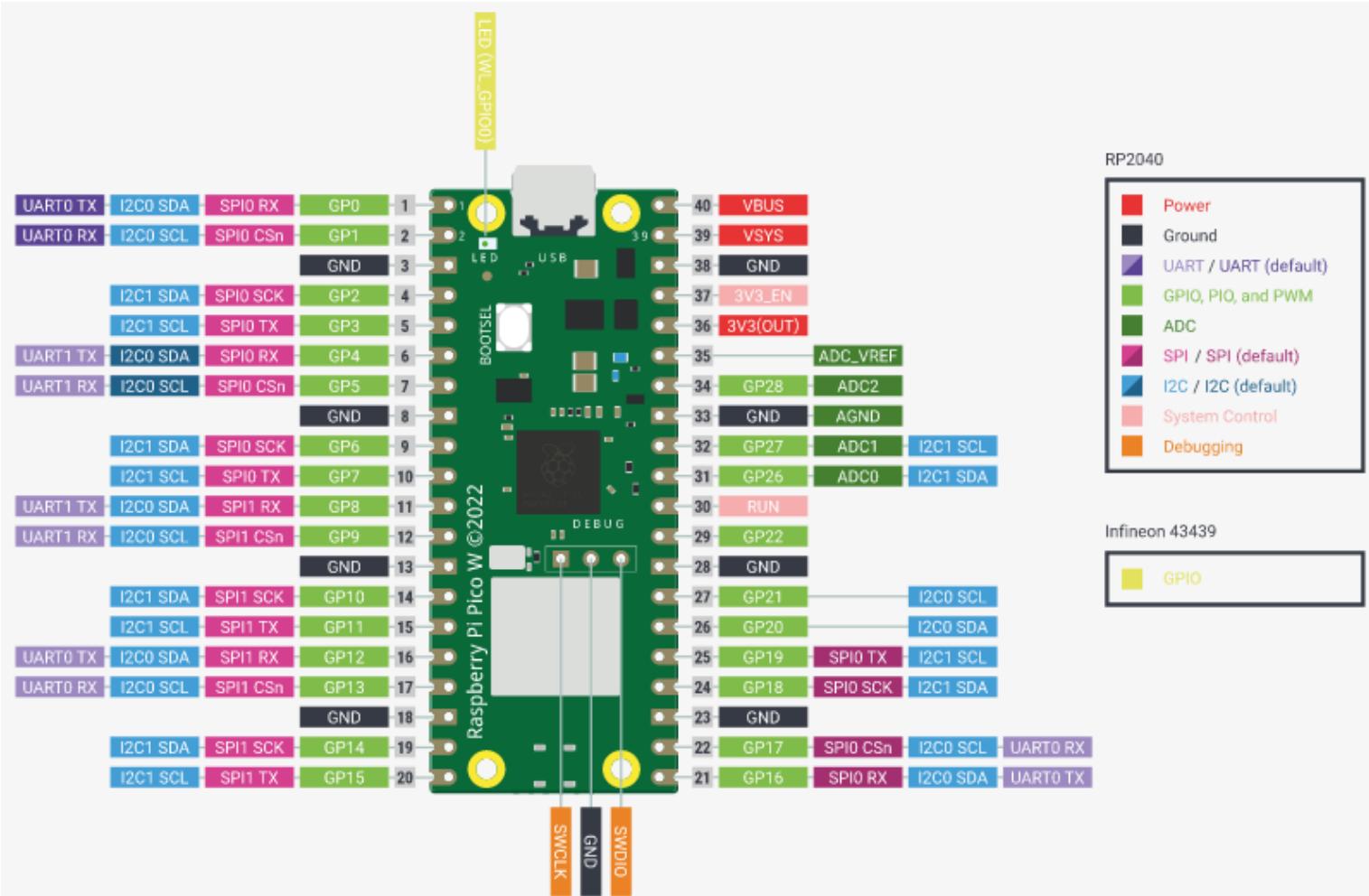
Unit Price \$ 0.0404

Ext. Price \$ 0.40

Calculator	Year Released	ROM	RAM	Processing Speed
TI-73	1998	256KB	512KB	6 MHz
TI-84 Plus	2004	128KB	1MB	15 MHz

Texas Instruments: TI73 and TI-84 Calculator Specifications. Own Image





PIO Instruction Set

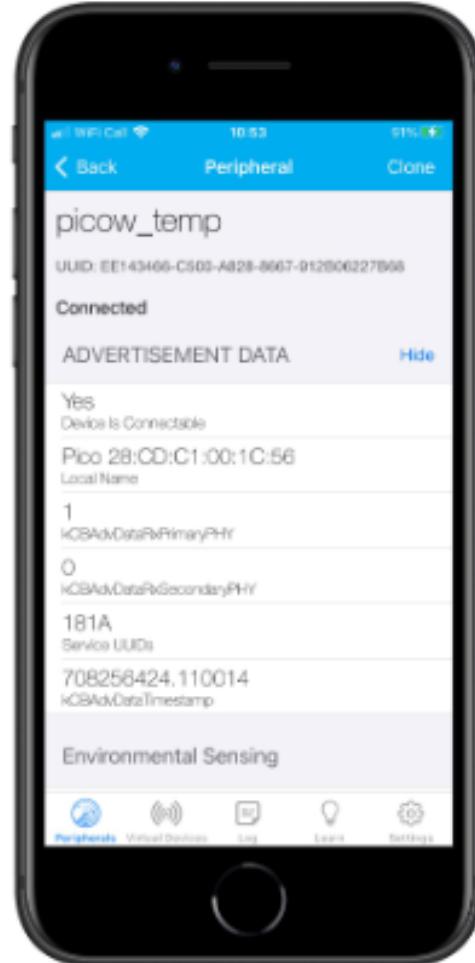
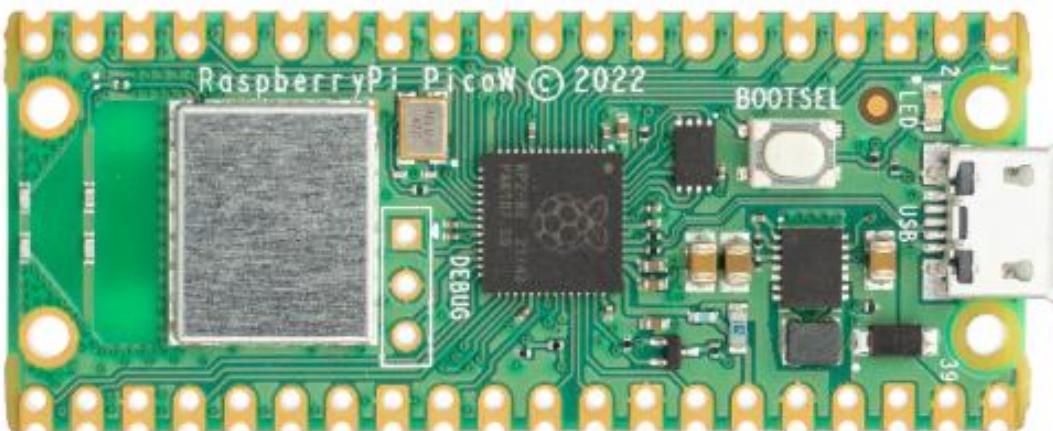
Mnemonic	Short Description
JMP	Set program counter to specified address if specified condition is true. Otherwise, no operation.
WAIT	Stall until some condition is met.
IN	Shift specified number of bits from specified source into the Input Shift Register.
OUT	Shift specified amount of bits out of the Output Shift Register and write those bits to specified destination.
PUSH	Push the contents of the Input Shift Register into the RX FIFO as a single 32-bit word. Clear Input Shift Register to all-zeroes.
PULL	Load a 32-bit word from the TX FIFO into the Output Shift Register.
MOV	Copy data from the specified source to the specified destination.
IRQ	Set or clear the IRQ flag selected by the specified index argument.
SET	Write specified immediate value to specified destination.

The PIOs' instruction set consists of just nine different instructions that are highly specialized on bit manipulation.

```
pull
mov x, ~osr
pull
mov y, osr
jmp test           ; this loop is equivalent to the following C code:
incr:             ; while (y--)
    jmp x-- test   ;     x--;
test:              ; This has the effect of subtracting y from x, eventually.
    jmp y-- incr
    mov isr, ~x
    push
```

Connecting to the Internet with Raspberry Pi Pico W

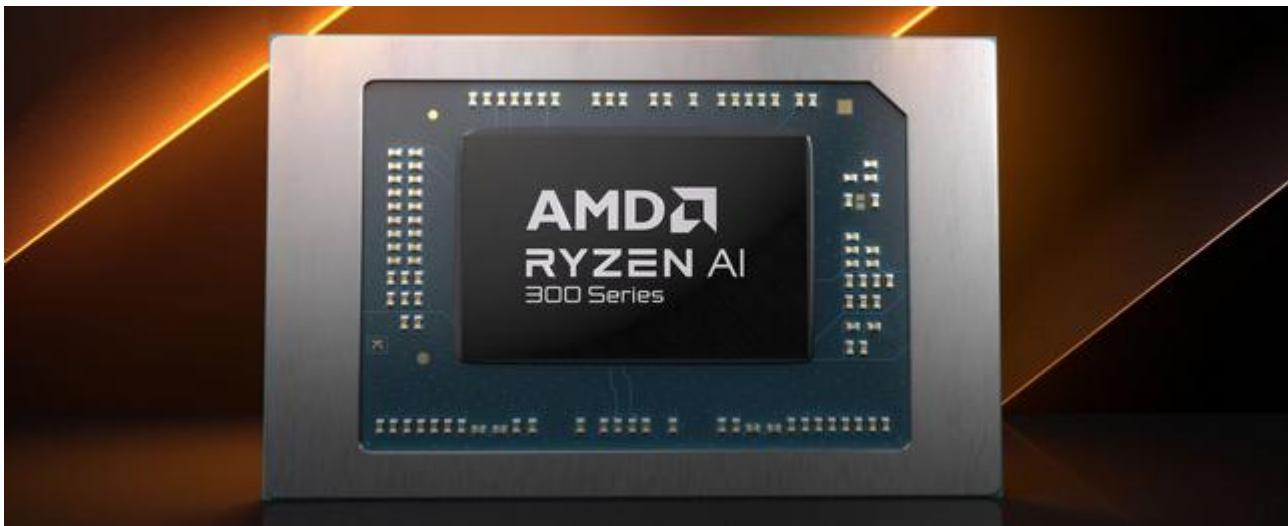
Getting Raspberry Pi Pico W online
with C/C++ or MicroPython





KEY SPECIFICATIONS OF INITIAL RELEASES:

CPU Model	Cores (P+E)	Max Turbo Frequency	L3 Cache	Integrated Graphics	TDP
Core Ultra 9 285K	8+16	5.7 GHz	36MB	Xe-LPG	125W
Core Ultra 7 265K	8+12	5.4 GHz	30MB	Xe-LPG	125W
Core Ultra 7 265KF	8+12	5.4 GHz	30MB	None	125W
Core Ultra 5 245K	6+8	5.3 GHz	24MB	Xe-LPG	125W
Core Ultra 5 245KF	6+8	5.3 GHz	24MB	None	125W

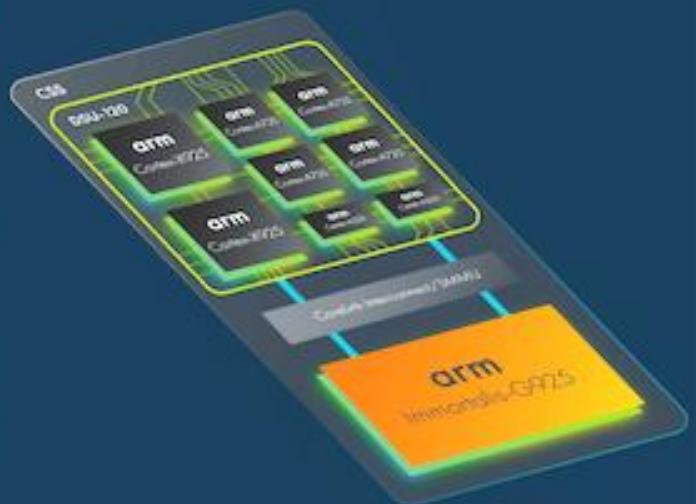


Model	Cores / Threads	Boost ⁹ / Base Frequency	Total Cache	Graphics Model	cTDP	NPU
AMD Ryzen™ AI 9 HX 370	12 / 24	5.1 GHz / 2.0 GHz	36MB	AMD Radeon™ 890M Graphics	15-54W	Yes (50 TOPs)
AMD Ryzen™ AI 9 365	10 / 20	5.0 GHz / 2.0GHz	34MB	AMD Radeon™ 880M Graphics	15-54W	Yes (50 TOPs)

Evolving Arm Reference Platform Configuration

Scaling for performance and efficiency

	T5323 reference platform* (gen 1)	CSS for Client reference platform* (Arm internal)
CPU Cluster	1x Cortex-X4 5x Cortex-A720 2x Cortex-A520 DSU-120 with 8MB L3	2x Cortex-X925 4x Cortex-A725 2x Cortex-A520 DSU-120 with 16MB L3
GPU	Immortalis-G720 12 cores, 2MB L2	Immortalis-G925 14 cores, 4MB L2
Interconnect	Arm CoreLink interconnect 8MB SLC	Arm CoreLink interconnect 16MB SLC
Memory system	4x16b LPDDR5X-8533	4x16b LPDDR5X-8533
Process node	4nm	3nm
Android	Android 13	Android 13



*Representative benchmarking configuration built for FPGA validation. Partner configurations may vary.



How many transistors are in a 3nm chip?

The 3nm process allows **250 million transistors per square nanometre** while the 2nm technology will enable over 310 million transistors in the same area. Higher transistor counts lower power consumption while enabling better performance with the same power usage. Aug 23, 2022

Area	
1	= 1e-12
Square nanometre	Square millimeter

*"A human hair is approximately **80,000-100,000 nanometers** (0.1 mm) wide."*

GEFORCE RTX 4070 Ti SUPER 16GB G6X | 256b Interface | More Cores

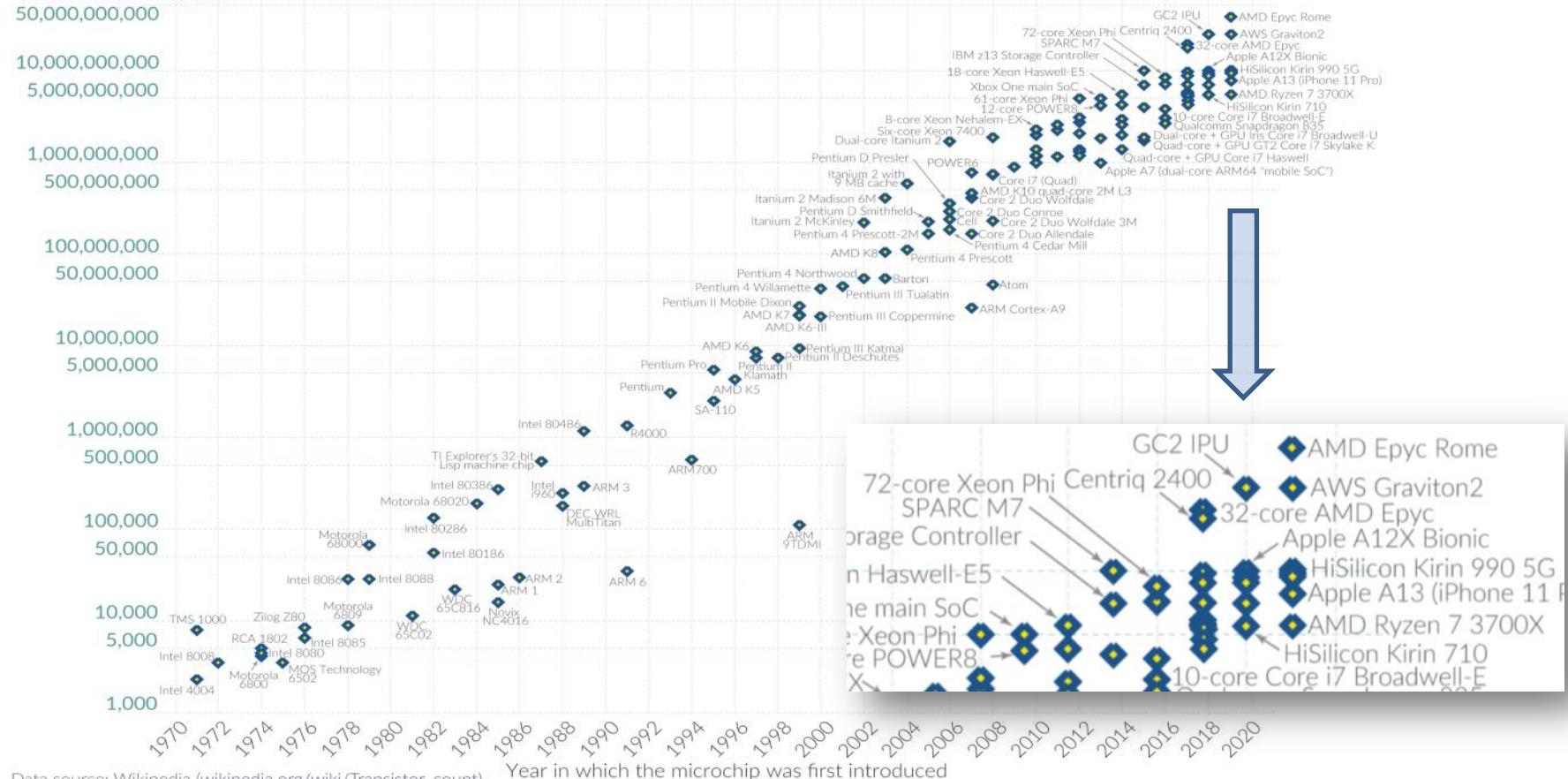
44 Shader TFLOPS | 102 RT TFLOPS | 706 AI TOPS
1440p 144Hz+
2.5X RTX 3070 Ti
\$799 | Jan 24th

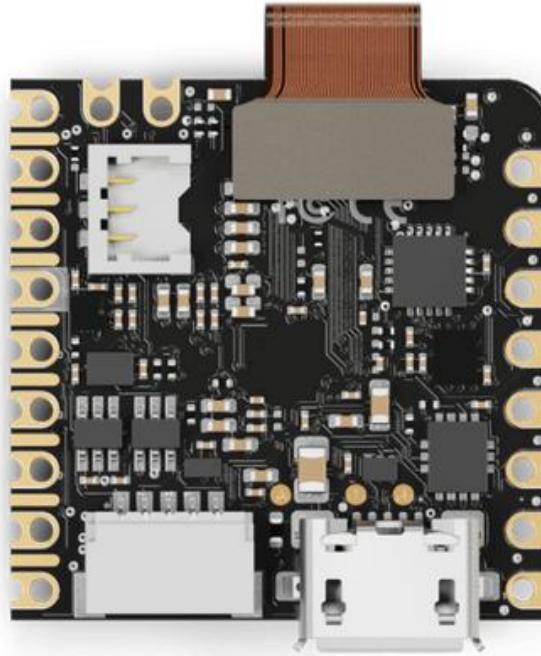


Moore's Law: The number of transistors on microchips doubles every two years

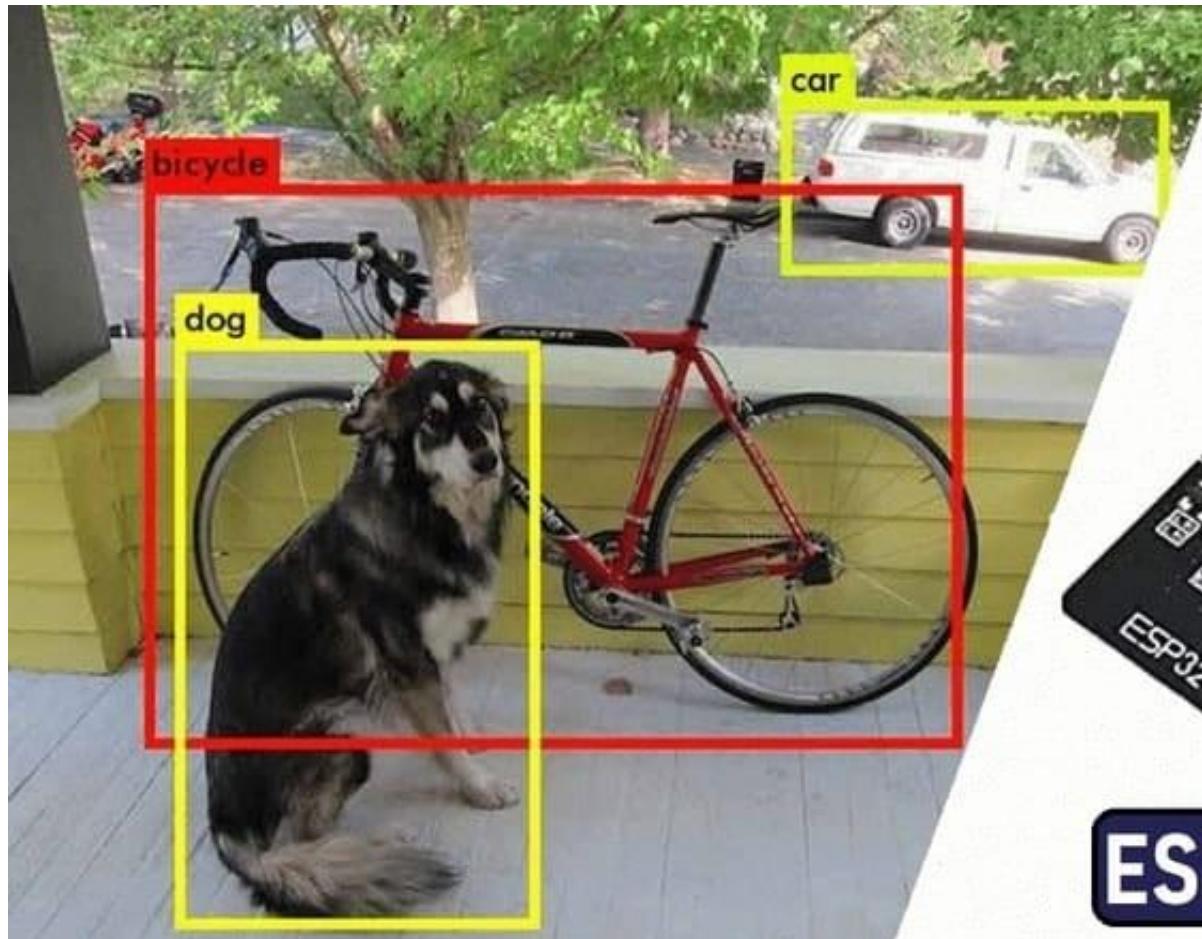
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Transistor count





"Arduino Nicla Vision, the new low-power solution that combines machine vision and edge computing"



Object Detection

Tracking



ESP32- CAM

Σύνοψη

- Περιεχόμενα:
 - Μέρος I: Αριθμητική και Λογική στις μηχανές
 - Μέρος II: Ψηφιακή τεχνολογία επεξεργαστών
- Σχετικό υλικό:
 - «Computer Architecture - Lecture 1: Introduction and Basics (Fall 2022)», ETH –
<https://www.youtube.com/watch?v=BIpPTqHK-Lc>
 - «Architecture All Access: Modern CPU Architecture Part 1 – Key Concepts», Intel –
<https://www.youtube.com/watch?v=vgPFzblBh7w>
 - Computer History Museum (CHM) – <https://www.youtube.com/@ComputerHistory>
 - Advent of Computing – <https://www.youtube.com/@adventofcomputing4504>
 - «Hello x86: Low-level assembly coding for the 8086», @ApneaCoding –
<https://youtu.be/tF16xTbd42w>
 - «BAM neural network in Arduino», @ApneaCoding – <https://youtu.be/RkM-rpSVD4I>

```

MOVE 1 TO DATA-C(N-T).
ADD 1 TO N-CHANGED.
GO TO LOOP-SCAN.

SELECT-CLZ.
ADD DATA-X(N-T) TO SUM2-X.
ADD DATA-Y(N-T) TO SUM2-Y.
ADD 1 TO N-CLZ.
IF DATA-C(N-T) EQUAL 2 GO TO LOOP-SCAN.
MOVE 2 TO DATA-C(N-T).
ADD 1 TO N-CHANGED.

```

```

LOOP- 91    id : Integer := 0; -- target ID (counter)
         det : Integer := 0; -- detection slots in sequence
         pwr : Integer := 0; -- rel. power of detection
         pwr0 : Integer := delimit; -- rel. power baseline (adapt
         disp : Boolean := False; -- target reporting (flag
 92
 93
 94
 95
 96 begin
 97     -- process the FOV slots --
 98     for p in 1..(seekerData'Length)-1 loop
 99         -- rel. power is current detection 'step'
100         pwr := abs(seekerData(p+1)-seekerData(p));
101         if pwr >= delimit then
102             -- detection valid, continue analysis
103             if pwr > pwr0 then
104                 -- strong new 'step' from baseline (new target)
105                 pwr0 := pwr; -- update the baseline
106                 det := 0; -- reset the run-length
107                 disp := False; -- enable target reporting
108
109         end if;
110         det := 1;
111         if
112             i = 1
113             then
114                 d := 1;
115                 p := 1;
116                 d := 0;
117
118         end if;
119         lcd.setCursor(0,1);
120         lcd.print(" 1.0.0");
121         delay(PRINT_DELAY);
122
123         lcd.clear();
124         lcd.setCursor(0,0);
125         lcd.print("Training...");
126
127         train_weights();
128         delay(PRINT_DELAY); // just for display pause
129
130         lcd.setCursor(0,1);
131         lcd.print("> Finished!");
132         delay(1000);
133     end loop;
134
135
136 // continuous loop routine
137 void loop( void )
138 {
139     display_weights();
140     display_input_tests();
141     display_output_tests();
142 }
143

```

STDIN
Input for the program (Optional)

Output:

```

x86_64-linux-gnu-gcc-9 -c HelloWorld.adb
HelloWorld.adb:1:01: compilation unit expected
x86_64-linux-gnu-gnatmake-9: "HelloWorld.adb" com

```



- Hamming (7,4) error correction codes in **R**
- Kmeans clustering in **COBOL**
- Bi-directional Associative Memory (BAM) in **Arduino/C**
- Linear Regression in **SQL, Matlab**
- ...

YouTube:

@ApneaCoding



<https://www.youtube.com/@apneacoding>

<https://www.facebook.com/apneacoding>

Github:

@xgeorgio



<https://github.com/xgeorgio>

<http://apneacoding.blogspot.com>

Ένας ψηφιακός κόσμος γεμάτος γνώση για όλους

Σύμφωνα με το Καταστατικό της Ένωσης Πληροφορικών Ελλάδας, ένας από τους βασικούς σκοπούς της λειτουργίας της είναι η προώθηση της γνώσης και χρήσης των πληροφορικών αγαθών από το κοινωνικό σύνολο και η εξάλειψη της τεχνοφοβίας και του "αναλφαβητισμού" στην Πληροφορική.



<https://courses.epe.org.gr>

Σχετικά με τα ανοικτά μαθήματα της Ένωσης Πληροφορικών Ελλάδας:

- ✓ Τα μαθήματα πραγματοποιούνται εξ ολοκλήρου διαδικτυακά, ζωντανά μέσω της πλατφόρμας Zoom.
- ✓ Η συμμετοχή σε όλα τα μαθήματα είναι ελεύθερη για οποιονδήποτε από οποιδήποτε στην Ελλάδα ή στο εξωτερικό.
- ✓ Δεν υπάρχει οικονομικό κόστος ή άλλες προϋποθέσεις συμμετοχής.
- ✓ Οι Εισιγητές είναι μέλη της Ένωσης Πληροφορικών Ελλάδας και πραγματοποιούν τα μαθήματα εθελοντικά.
- ✓ Τα μαθήματα μαγνητοσκοπούνται και παραμένουν διαθέσιμα για σύγχρονη παρακολούθηση στο Αρχείο Μαθημάτων.
- ✓ Η εκπαίδευση που παρέχεται μέσω των ανοικτών διαδικτυακών μαθημάτων είναι άτυπη και δεν παρέχονται βεβαιώσεις παρακολούθησης στους συμμετέχοντες.



Ερωτήσεις



Χάρης Γεωργίου (MSc,PhD)

<https://www.linkedin.com/in/xgeorgio>

<https://methodd.substack.com>