

Komputer EDVAC

czyli von Neumann zrobi to lepiej

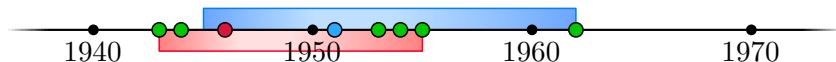
Michał Bukowski Kajetan Dutka

Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski

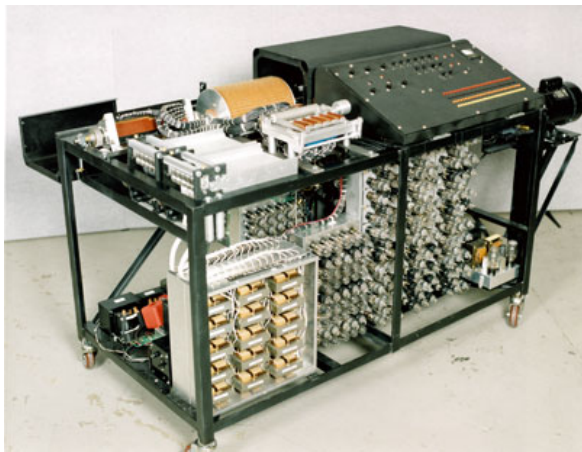
15.12.14 Historia Komputerów

Timeline

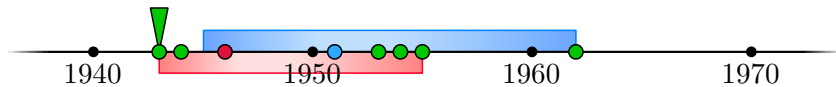
- ▶ Kontekst historyczny dla EDVACa.
- ▶ Kolorowe groszki reprezentują komputery.
- ▶ Wybór tendencyjny i niekompletny, głównie z USA.
- ▶ Kolorowe paski dotyczą ENIACa i EDVACa, trwają od początku budowy do końca działania.



ABC

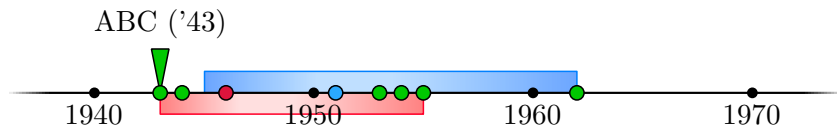


ABC ('43)

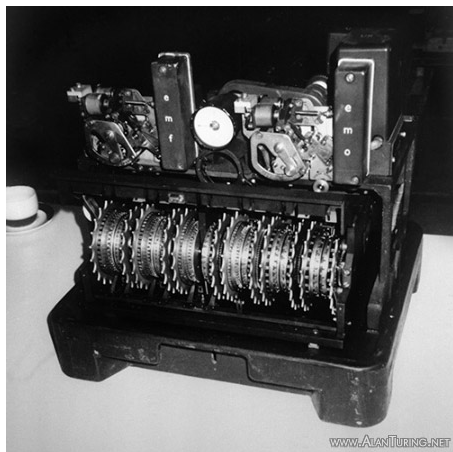


ABC

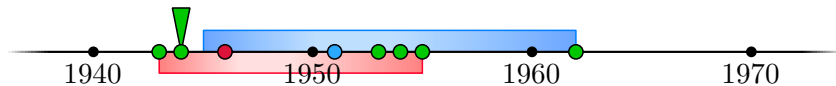
- ▶ vacuum = lampa
- ▶ tube = elektronowa
- ▶ vacuum tube = lampa elektronowa
- ▶ zobacz też: Pchła Szachrajka by Jan Brzechwa



Colossus

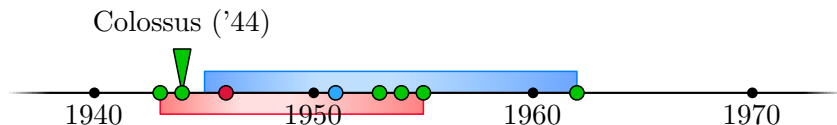


Colossus ('44)

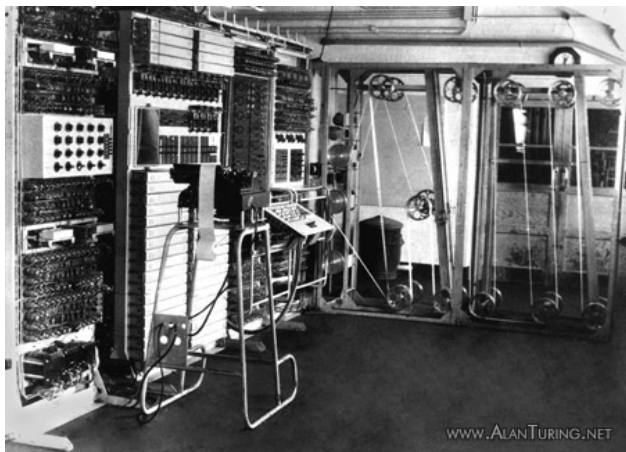


Colossus

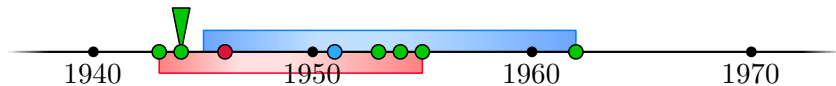
- ▶ Colossus? Wygląda niepozornie.
- ▶ Na poprzednim zdjęciu była Maszyna Lorenza, Tunny.
- ▶ Colossus pomagał w odczytywaniu zaszyfrowanych nią wiadomości.



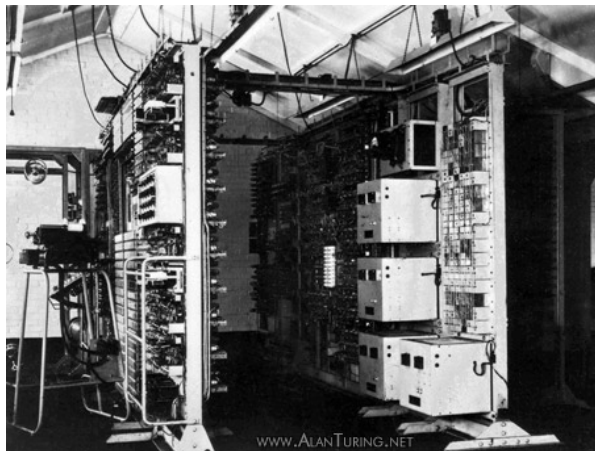
Colossus



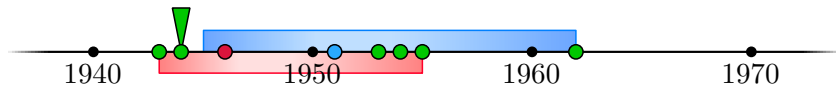
Colossus ('44)



Colossus



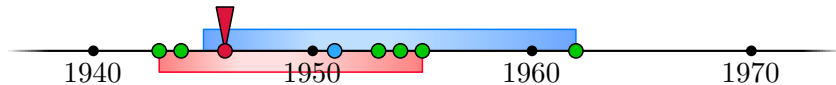
Colossus ('44)



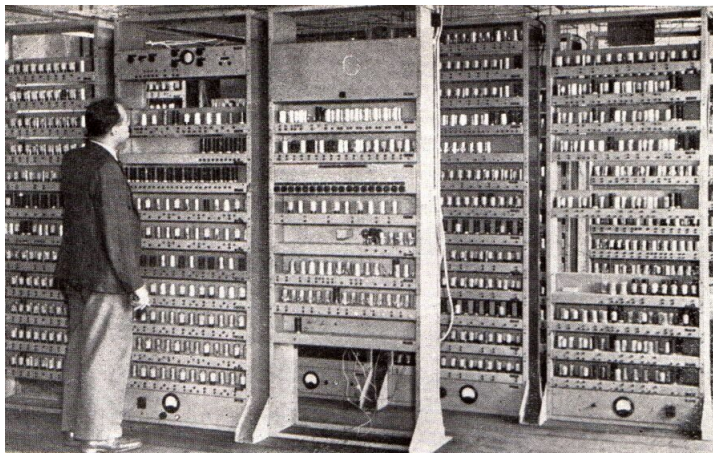
ENIAC



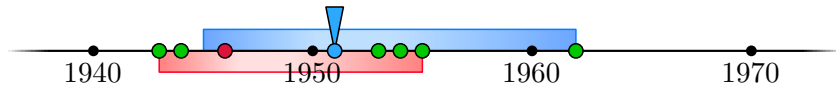
ENIAC ('46)



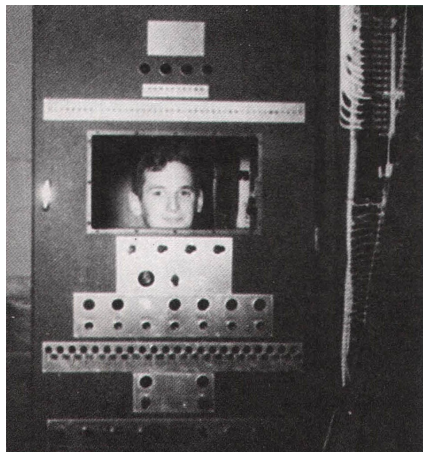
EDVAC



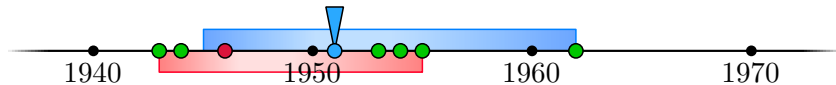
EDVAC ('51)



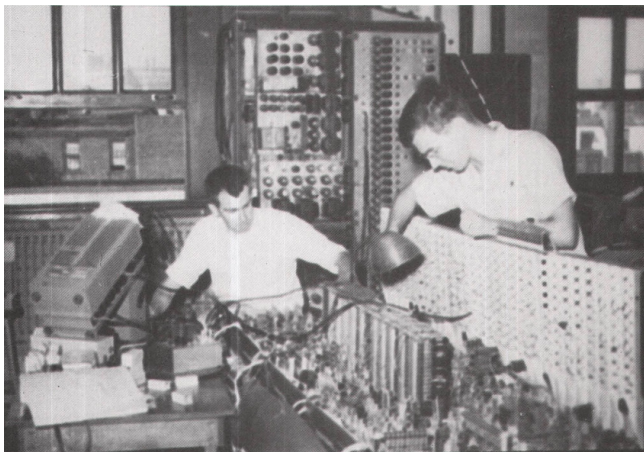
EDVAC



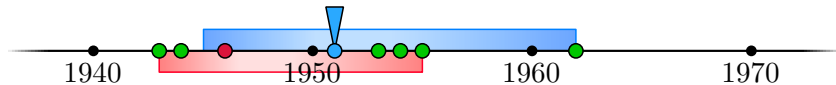
EDVAC ('51)



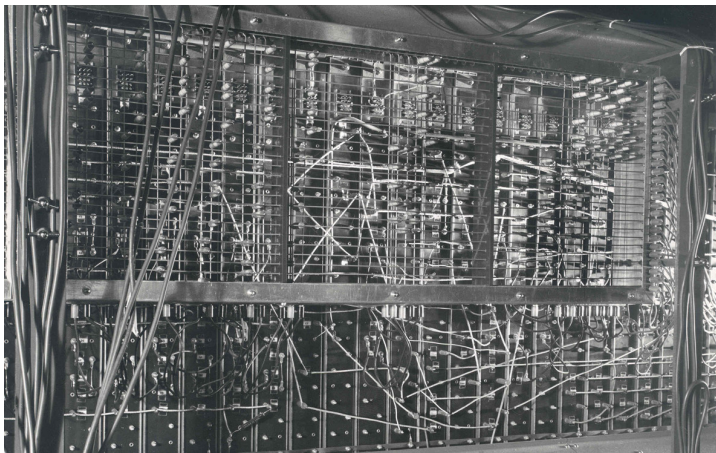
EDVAC



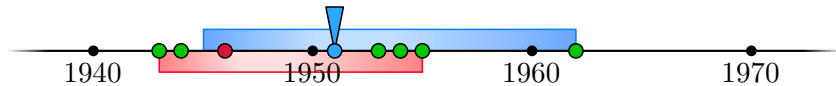
EDVAC ('51)



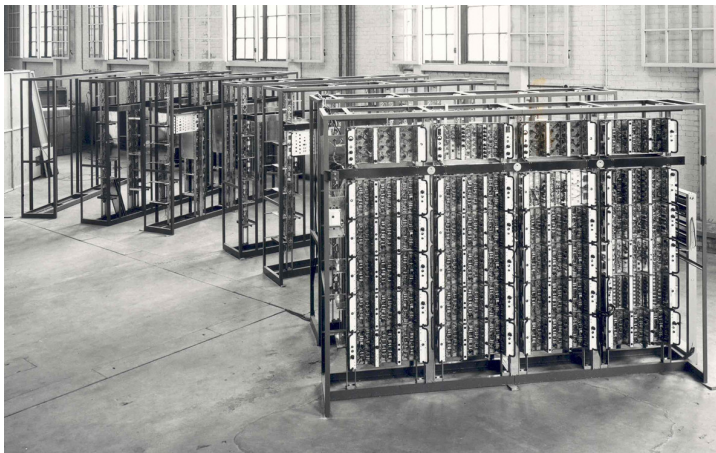
EDVAC



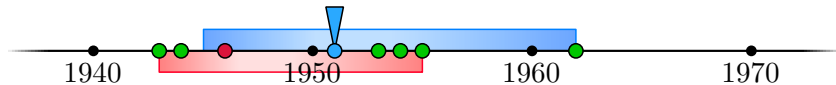
EDVAC ('51)



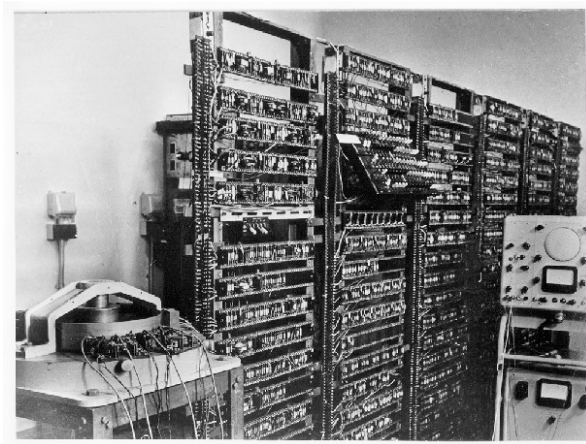
EDVAC



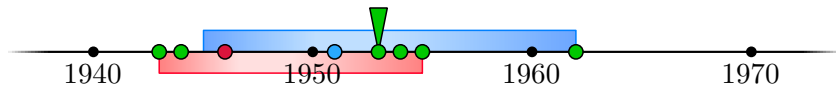
EDVAC ('51)



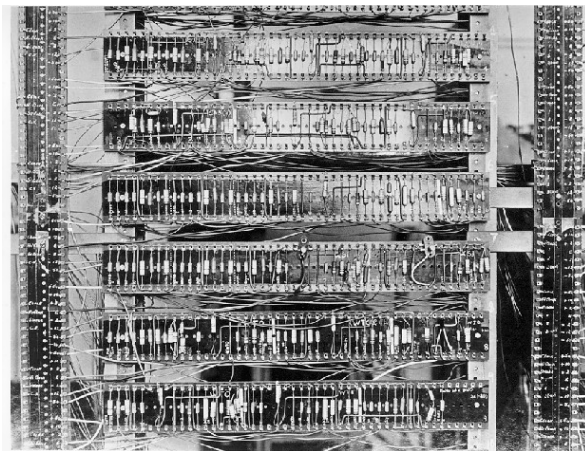
Manchester's etc.



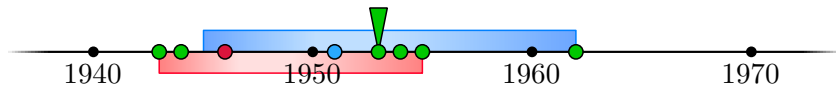
University of Manchester's experimental Transistor Computer ('53)



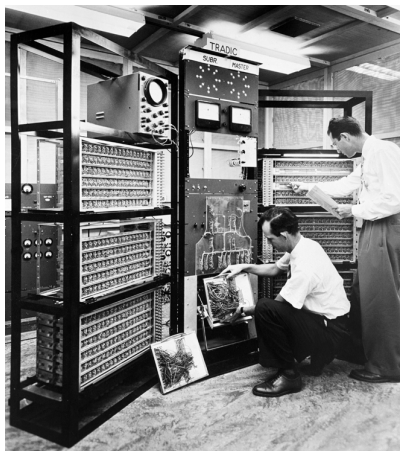
Manchester's etc.



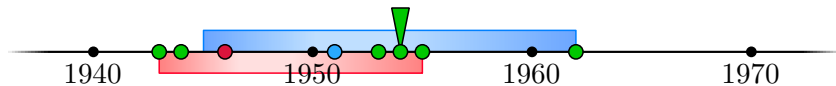
University of Manchester's experimental Transistor Computer ('53)



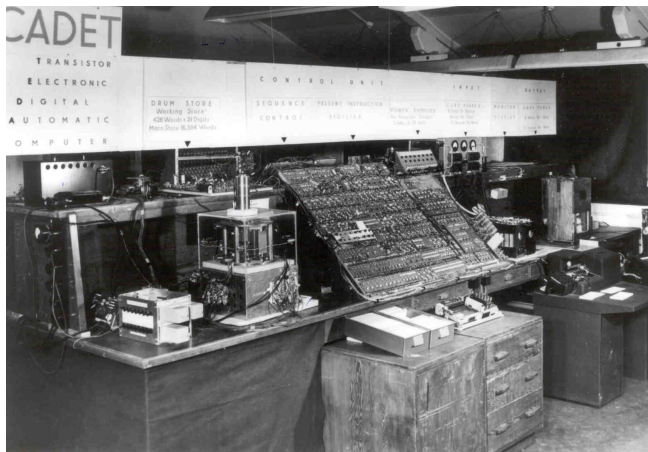
TRADIC



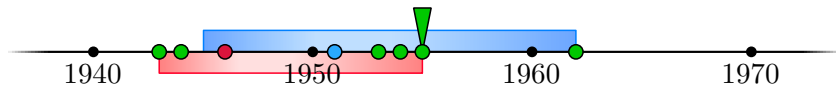
TRADIC ('54)



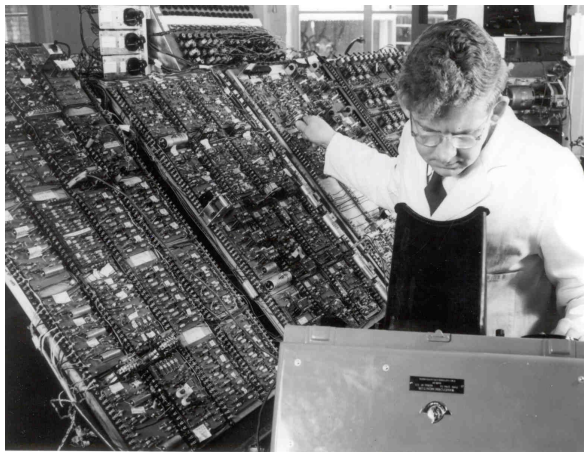
CADET



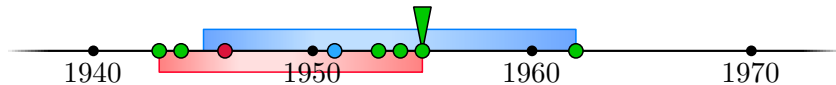
CADET ('55)



CADET



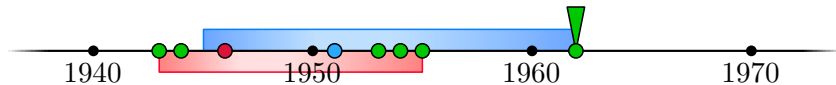
CADET ('55)



BRLESC



BRLESC (1962)



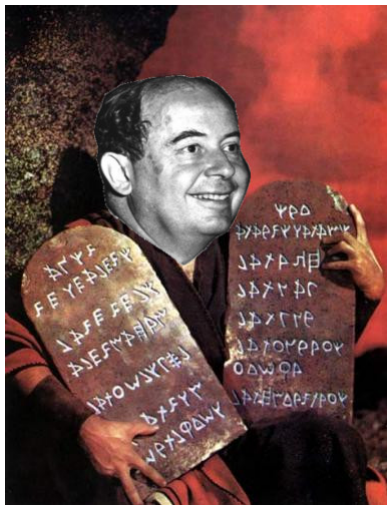
EDVAC

- ▶ Electronical Discrete Variable Automatic Computer.
- ▶ Uruchomiony w sierpniu 1951 roku.
- ▶ Początek EDVACowi dały rozważania Johna Mauchly'ego i Johna Eckerta.
- ▶ Pomysł na konstrukcję jeszcze przed ukończeniem ENIACa (Sierpień 1944).
- ▶ W 1945 dołączył John von Neumann, podsumował całą ideę w pracy "First Draft of a Report on the EDVAC".

Teoria

- ▶ Architektura von Neumanna.
- ▶ Obliczenia w systemie binarnym w U2
(Zuse 1 : von Neumann 1).
- ▶ Fixed Binary Point : liczby z przedziału -1 do 1
(Zuse 2 : von Neumann 1).
- ▶ Obliczenia sekwencyjne.
- ▶ Prostota przede wszystkim.

Pentalog von Neumanna



Jam jest Pan Bóg twój,
Którym cię wywiódł z ziemi
dziesiątej,
Z domu niewoli.

Nie będziesz miał operacji cudzych przede mną...

- I. Urządzenie powinno wspierać conajmniej 4 podstawowe operacje arytmetyczne : $+$, $-$, $*$, \div , $\sqrt{\quad}$ przy pomocy wydzielonych organelli.

Tutaj trudniej znaleźć odpowiednik w dekalogu

II. Powinna istnieć jednostka logiczna zawiadująca wykonywaniem operacji.

Pamiętaj, abyś...

III. Urządzenie musi być w stanie wykonać długi i skomplikowany ciąg operacji (w szczególności obliczeń).
W tym celu powinno umieć zapamiętać znaczną liczbę częściowych wyników.

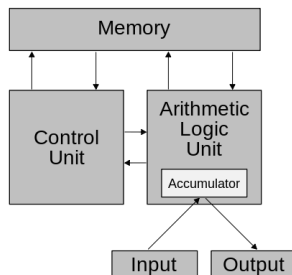
Pożądaj danych bliźniego swego

IV. Urządzenie musi być w stanie przekierowywać dane z zewnętrznego nośnika do swojej pamięci...

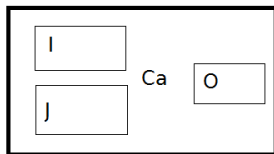
Jak Kuba Bogu, tak Bóg Kubańczykom

V. ...jak i wyliczone dane ze swojej pamięci do zewnętrznego nośnika.

Architektura von Neumanna na przykładzie EDVACa



- ▶ Jednostka Arytmetyczno Logiczna (CA) wykonuje obliczenia.
- ▶ Program trzymany razem z danymi w Pamięci (M).
- ▶ Jednostka Kontrolna (CC) decyduje, który rozkaz wykonać i na jakich danych.
- ▶ Komputer ma Wejście (I) i Wyjście (O) do komunikacji ze światem zewnętrznym.



- ▶ Głównym zadaniem EDVACa były obliczenia numeryczne.
- ▶ Podstawowe operacje arytmetyczne były dostępne jako instrukcje : $+$, $-$, $*$, \div , $\sqrt{\cdot}$.
- ▶ Poza tym dostępne były konwersje db, bd oraz magiczne instrukcje: i, j, s.
- ▶ Bardziej skomplikowane funkcje miały być stabilizowane w pamięci.

- ▶ Otrzymuje rozkazy z pamięci (tej samej, w której są dane numeryczne).
- ▶ Rozkaz wykonania operacji w CA.
- ▶ Rozkaz przesunięcia liczby.
- ▶ Wczytanie następnego rozkazu (implicite chodzi o możliwość skoku).
- ▶ Operacje wejścia/wyjścia.

M

- ▶ Przechowywane są w niej zarówno dane jak i rozkazy.
- ▶ Adres określany przez 2 wartości $\mu\rho$.
 - ▶ μ określona za pomocą 8 bitów.
 - ▶ ρ określona za pomocą 5 bitów.
- ▶ Nie ma bezpośrednich połączeń między komórkami pamięci, transfery między komórkami umożliwia CA.
- ▶ Rozkazy ułożone w kolejnych komórkach pamięci.
- ▶ Cała pamięć ma połączenie do rejestru I.

Klasy rozkazów

- ▶ Instrukcje należą do jednej z 8 klas oznaczonych greckimi literami:
 $\alpha \beta \gamma \delta \epsilon \theta \zeta \eta$
- ▶ W raporcie klasy są przedstawione w tej właśnie kolejności, dociekliwy obserwator widzi jednak, że 3 ostatnie klasy są w złym porządku.
- ▶ Przypadek? Być może.
- ▶ Skoro mamy 8 klas rozkazów to potrzebujemy 3 bitów na określenie, która klasa nas interesuje.

Klasa rozkazów α

- ▶ Wykonaj jedną z 10 instrukcji CA.
- ▶ Zakładamy, że dane są w rejestrach I oraz J.
- ▶ Do zakodowania potrzebne jest 5 bitów (4 na instrukcję, 1 na magiczną wartość c).

Klasa rozkazów β

- ▶ Załaduj liczbę z pamięci do rejestru I.
- ▶ W rozkazie podane adresy μ i ρ .
- ▶ Do zakodowania potrzeba 13 bitów.

Klasa rozkazów γ

- ▶ Załaduj liczbę, która znajduje się w następnej komórce pamięci, do rejestru I.
- ▶ Szybsze niż β , które jest instrukcją charakteryzującą dostęp swobodny.
- ▶ Do zakodowania potrzebne jest potrzebne 0 bitów!

Klasa rozkazów δ

- ▶ Załaduj liczbę z rejestru O do pamięci.
- ▶ Adresy podobnie jak w γ .
- ▶ Do zakodowania potrzebne jest 13 bitów.

Klasa rozkazów ϵ

- ▶ Załaduj liczbę z rejestru O do następnej komórki pamięci.
- ▶ Instrukcja przeciwna do instrukcji γ .
- ▶ Nadal 0 bitów.

Klasa rozkazów θ

- ▶ Załaduj liczbę z rejestru O do rejestru I.
- ▶ Rozmiar 0 bitów.

Klasa rozkazów ζ

- ▶ Skok do instrukcji pod adresem $\mu\rho$.
- ▶ Rozmiar 13 bitów.

Klasa rozkazów η

- ▶ Operacje wejścia/wyjścia.
- ▶ Rozmiar bliżej nieokreślony.

Optymalizacje

- ▶ Słowo liczy 32 bity, rozkazy maksymalnie 13, czy jest tu pole do popisu?
- ▶ Łączymy instrukcje α oraz δ i ϵ i θ by jednocześnie obliczać i przypisywać wynik.

Koniec niespodzianek

- ▶ Wszyscy dobrze znają architekturę von Neumanna.
- ▶ Można sobie mniej więcej wyobrazić, jak EDVAC działał w środku.
- ▶ Pewnie tak samo, jak dzisiejsze komputery, tylko był wielki i głupi.

Spoiler alert



**KEEP
CALM
AND PREPARE TO
BE
AMAZED!**

PREPARE...

TO BE...

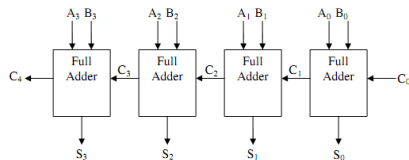


AMAZED!!!

Nieśmiałe kroki w kierunku dodawania

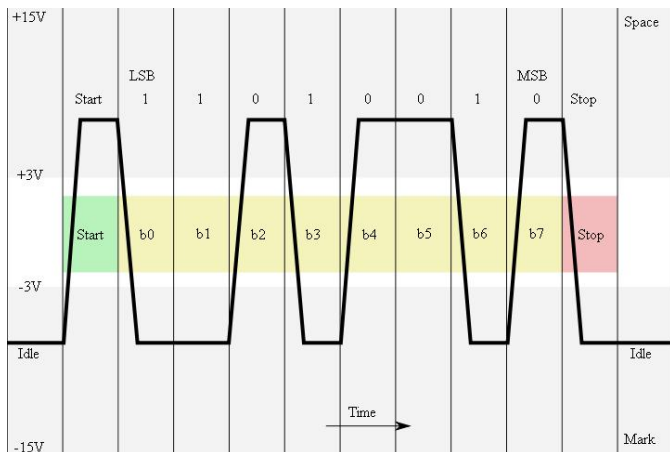
- ▶ Be amazed, ale jeszcze nie teraz.
- ▶ Liczba to ciąg bitów.
- ▶ Można go reprezentować na dwa sposoby.
- ▶ Nie chodzi o to, że jako int albo float, ani o to, że można zamienić rolę zer i jedynek.

Równoległość vs sekwencyjność



- ▶ “place multiple events in a temporal succession rather than in (simultaneous) spatial juxtaposition”

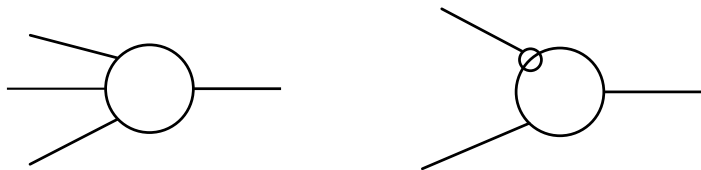
Kodowanie liczb



- ▶ Jedna liczba – jeden kabel, niezależnie od liczby bitów.
- ▶ Uzasadnienie: prostszy sprzęt, bo mniej lamp.
- ▶ Taktowanie rzędu kilku MHz \gg 10Hz nie ogranicza.

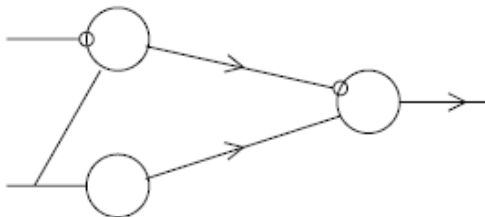
Zamiast bramek logicznych: E-elementy

- ▶ Uogólniona lampa elektronowa, podobna do neuronu.
- ▶ Kilka wejść i kilka wyjść.
- ▶ Wejścia wzbudzające lub hamujące.
- ▶ Domyślna granica aktywacji to 1.



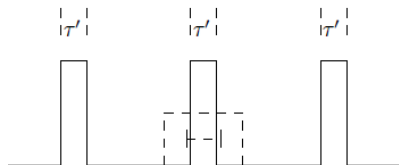
Układ z E-elementów

- ▶ Przykład: układ o granicy aktywacji równej 2.
- ▶ Można myśleć, że to bramka UND, ale próby tłumaczenia kolejnych układów na bramki logiczne spełzną na niczym.
- ▶ Każda strzałka to opóźnienie o jeden cykl.



Taktowanie

- ▶ Charakterystyka sygnału na obrazku.
- ▶ Sygnał (chyba) stanowi zasilanie lamp i jest doprowadzony do każdej oddzielnie.
- ▶ A może tylko wejście jest taktowane?
- ▶ “In all this we are following the procedure of W.J. MacCulloch and W. Pitts”
- ▶ “the stimulus is completely renewed and synchronized after each step”



CLOCK PULSE



tolerance limits
for the open
gate period

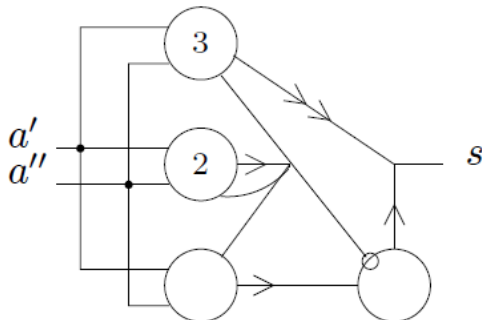
W poszukiwaniu prawdy

- ▶ W.S. McCulloch and W. Pitts “A logical calculus of the ideas immanent in nervous activity”, Bull. Math. Biophysics, Vol.5 (1943)
- ▶ Artykuł wygląda mniej więcej tak:

$$\begin{aligned} & (E m, n) (p) n(i, \psi) (Ej) : . (x) m : \psi(x) = 0 . \mathbf{v.} \psi(x) = 1 : \\ & \phi_i \varepsilon \sigma(\psi, n t + p) : \rightarrow : (w) m (x) t - 1 . \phi_i(n(t + 1) \\ & + p, n x + p, w) = \phi_j(n t + p, n x + p, w) : . \\ & (u, v) (w) m . \phi_i(n(u + 1) + p, n u + p, w) \\ & = \phi_i(n(v + 1) + p, n v + p, w). \end{aligned}$$

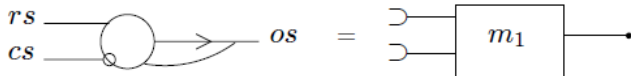
Dodawanie

- ▶ (2) generuje przeniesienie.
- ▶ Wszystkie elementy z lewej warstwy mają 3 wejścia, w tym przeniesienie z poprzedniego cyklu.



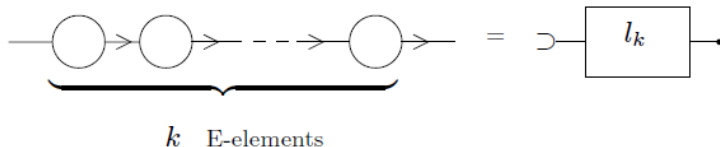
Pamięć 1-bitowa

- ▶ Pojedynczy E-element ze sprzężeniem zwrotnym opóźnionym o jeden cykl.
- ▶ rs - receiving stimulus
- ▶ cs - clearing stimulus



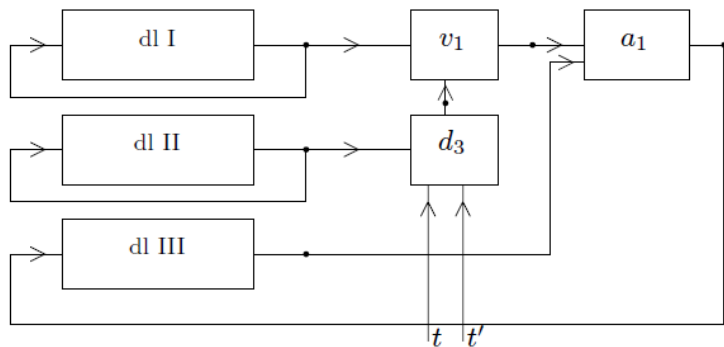
Opóźniacz k-bitowy

- ▶ Ścieżka z k E-elementów pozwala opóźnić ciąg zer i jedynek o k cykli.
- ▶ Można go uzupełnić o sprzężenie zwrotne i elementy do zapisu/zerowania.
- ▶ Daje się zbudować z samych kabli i dwóch E-elementów na końcach.

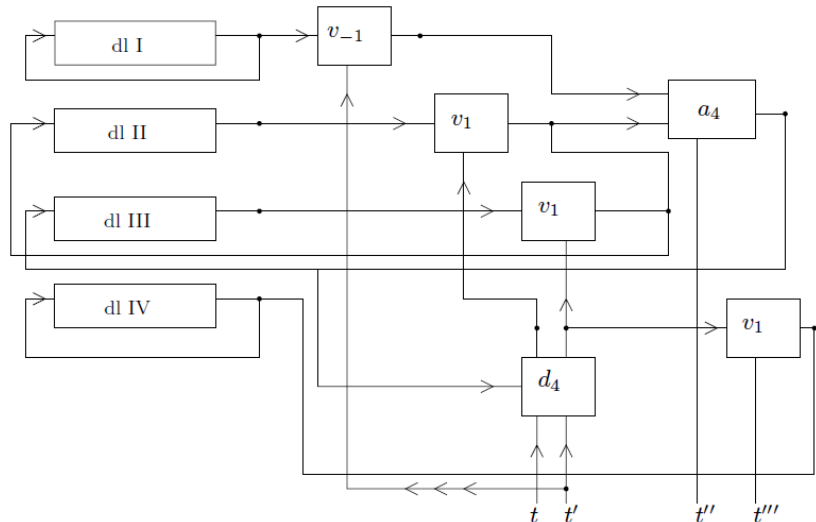


Układ mnożący

- ▶ Argumenty w dl I i dl II.
- ▶ Wynik buduje się w dl III.
- ▶ d_3 trzyma jeden bit przez całą fazę dodawania.
- ▶ v_1 przepuszcza lub blokuje liczbę z dl I.
- ▶ a_1 to układ dodający.



Dzielenie: analogicznie



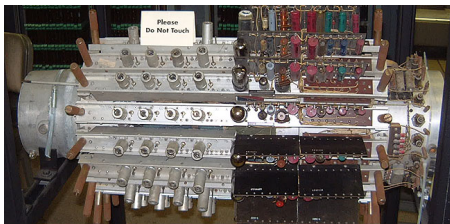
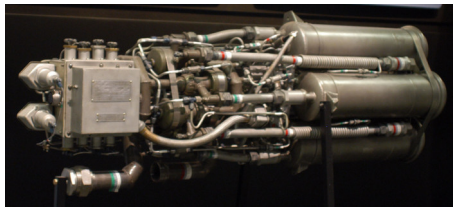
- Czytając schemat trzeba brać pod uwagę upływ czasu.

Pamięć operacyjna

- ▶ CA wystarczą opóźniacze, ale co z M?
- ▶ Pamięć operacyjna z lamp elektronowych to pomysł równie dobry, jak RAM z rejestrów.
- ▶ Rozwiązanie: pamięć rtęciowa, dynamiczna i cykliczna.
- ▶ Wszystkie epitety dotyczą jednej rzeczy.

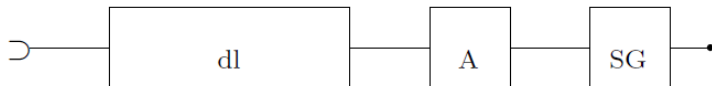
Pamięć rtęciowa

- ▶ Jedno z poniższych zdjęć przedstawia pamięć rtęciową.



- ▶ Na drugim jest silnik raketowy. Czy potrafisz je odróżnić?

Schemat rury z bitami



A: AMPLIFICATION

SG: SWITCHING & GATING

- ▶ Dl ma z lewej strony głośnik, a z prawej słuchawkę.
- ▶ W środku mieści się kilka tysięcy bitów.
- ▶ SG to kontroler pamięci, może czytać i pisać.
- ▶ Całość trzeba jeszcze zapętlić.

Ile to miejsca, tak na chłopski rozum?

- ▶ 256 rur, każda po 1024 bity, $\frac{2^8 \cdot 2^{10}}{8} = 32\text{KB}$.
- ▶ Każda liczba ma 30 bitów znaczących, bit znaku i flagę “czy to jest instrukcja”.
- ▶ W pamięci mieszczą się 8192 liczby.

To ile tych tam lampek w sumie potrzeba?

- ▶ Procesor miał składać się raptem z kilkuset lamp elektronowych.
- ▶ Skok do 6000 spowodowały wzmacniacze i kontrolery pamięci.

Wymiary

ENIAC:

- ▶ Waga 27 ton, powierzchnia 157 m², pobór prądu 150 kW

EDVAC:

- ▶ Waga 8 ton, powierzchnia 45 m², pobór prądu 56 kW

Bebechy i wydajność

ENIAC:

- ▶ W ciągu sekundy 5000 dodawań, 357 mnożeń, 38 dzielení, 3 pierwiastki
- ▶ Pamięć w postaci akumulatorów, później rozszerzony według założeń von Neumanna
- ▶ 17000 lamp elektronowych

EDVAC:

- ▶ W ciągu sekundy 1160 dodawań, 340 mnożeń (zależne od miejsca w pamięci)
- ▶ 32 KB pamięci
- ▶ 6000 lamp elektronowych