Architektura komputerów

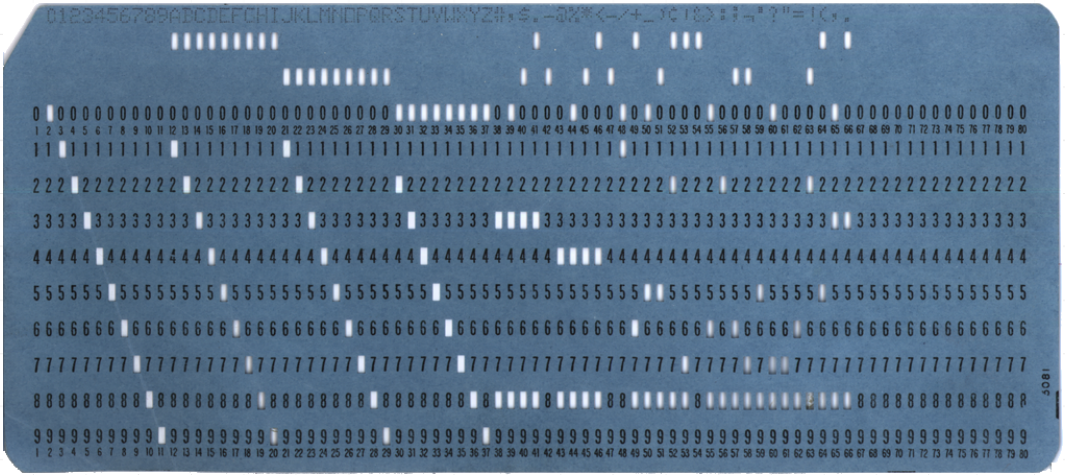
wer. 17 z drobnymi modyfikacjami!

Wojciech Myszka

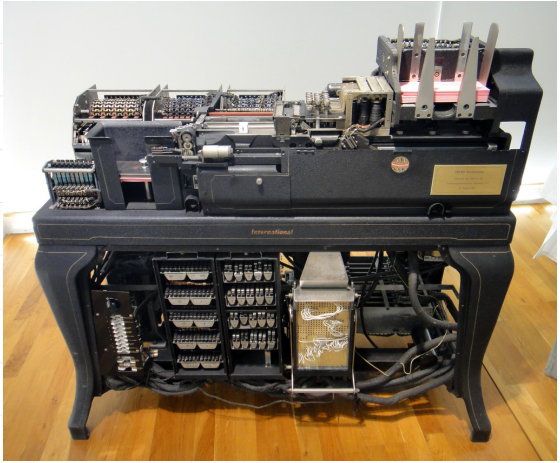
2023-10-30 16:18:48 +0100

Politechnika Wrocławska

Karty perforowane



Kalkulator

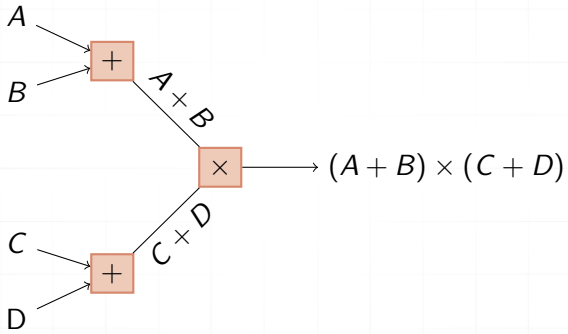


IBM 601, 1931

IBM 601 kalkulator

Maszyna czytała dwie, maksimum ośmiocyfrowe, liczby z karty, mnożyła je przez siebie i wynik dziurkowała na tej samej karcie. Potrafiła również dodawać i odejmować.
Nie mogła wyników drukować. W tym celu musiała „współpracować" z tabulatorem.
Model zainstalowany w laboratorium Eckerta posiadał funkcję interpolacji (specjalnie zaprojektowaną przez IBM). Właściciel (użytkownik) maszyny — Eckert połączył ją z innymi maszynami (Tabulator, Duplikator) za pomocą własnoręcznie zaprojektowanego elektrycznego łącznika tworząc pierwszy „kombajn" do złożonych obliczeń naukowych.
http://www.columbia.edu/cu/computinghistory/601.html

Idea obliczeń

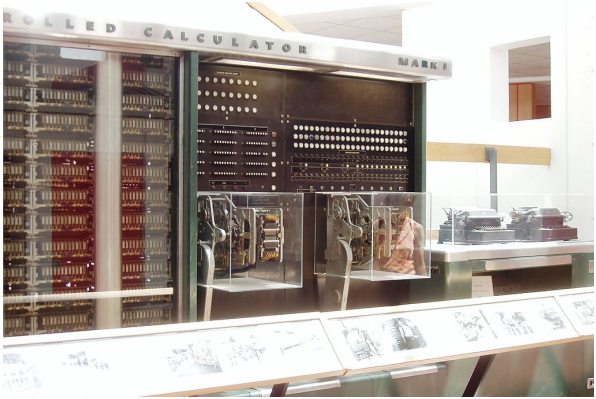


- A, B, C, D — dane: „strumień” (plik) kart perforowanych
- $A + B$, $C + D$ — wyniki częściowe: plik kart perforowanych
- $(A + B)(C + D)$ — wynik, również plik kart perforowanych

Rysunek: Schemat obliczeń z wykorzystaniem maszyn wykonujących proste operacje na kartach perforowanych

Pierwsze elektromechaniczne komputery. . .

. . . realizowały podobne idee



John von Neumann

1. Jednym z najważniejszych i największych projektów obliczeniowych lat czterdziestych były obliczenia na potrzeby bomby atomowej.
2. Brał w ich udział John von Neumann, genialny matematyk i fizyk (który zajmował się również mechaniką kwantową, teorią gier, informatyką, analizą funkcjonalną...)

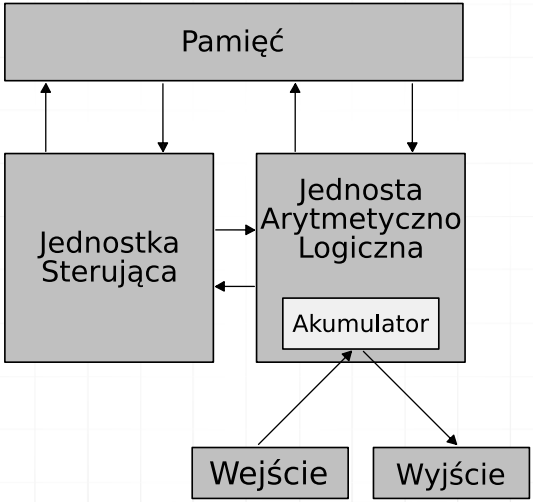
Architektura von Neumanna

Architektura von Neumanna — rodzaj architektury komputera, przedstawionej po raz pierwszy w 1945 roku przez von Neumanna stworzonej wspólnie z Johnem W. Mauchly'ym i Johnem Presperem Eckertem Polega na podziale komputera na trzy podstawowe części:

- procesor (w ramach którego wydzielona bywa część sterująca oraz część arytmetyczno-logiczna);
- pamięć komputera (zawierająca dane i sam program);
- urządzenia wejścia/wyjścia.

Rysunek: Schemat architektury von Neumanna

Architektura von Neumanna



Rysunek: Schemat architektury von Neumanna

Architektura von Neumanna

Rysunek: Schemat architektury von Neumanna

System komputerowy zbudowany w oparciu o architekturę von Neumanna powinien:

- mieć skończoną i funkcjonalnie pełną listę rozkazów;
- mieć możliwość wprowadzenia programu do systemu komputerowego poprzez urządzenia zewnętrzne i jego przechowywanie w pamięci w sposób identyczny jak danych;
- dane i instrukcje w takim systemie powinny być jednakowo dostępne dla procesora;
- informacja jest tam przetwarzana dzięki sekwencyjnemu odczytywaniu instrukcji z pamięci komputera i wykonywaniu tych instrukcji w procesorze.

Architektura von Neumanna

Rysunek: Schemat architektury von Neumanna

Podane warunki pozwalają przełączać system komputerowy z wykonania jednego zadania (programu) na inne bez fizycznej ingerencji w strukturę systemu, a tym samym gwarantują jego uniwersalność.
System komputerowy von Neumanna nie posiada oddzielnych pamięci do przechowywania danych i instrukcji. Instrukcje jak i dane są zakodowane w postaci liczb (binarnych). Bez analizy programu trudno jest określić czy dany obszar pamięci zawiera dane czy instrukcje. Wykonywany program może się sam modyfikować traktując obszar instrukcji jako dane, a po przetworzeniu tych instrukcji — danych — zacząć je wykonywać.

Schemat komputera

Rysunek: Schemat kmputera nawiazujący do architektury von Neumanna

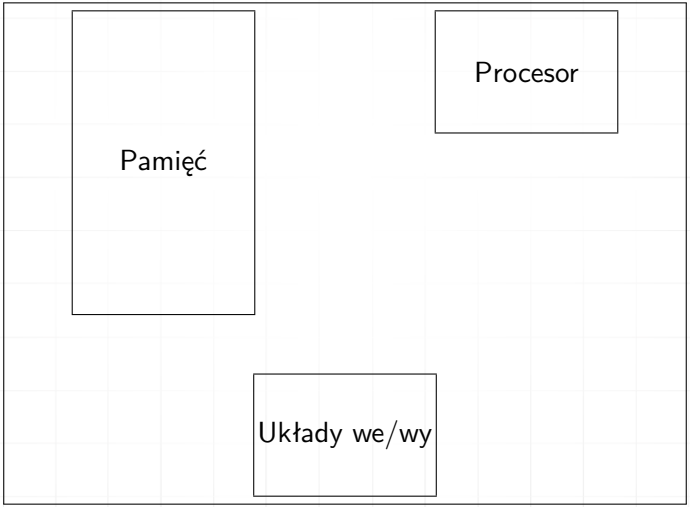
Schemat komputera

Rysunek: Schemat kmputera nawiazujący do architektury von Neumanna

Schemat komputera

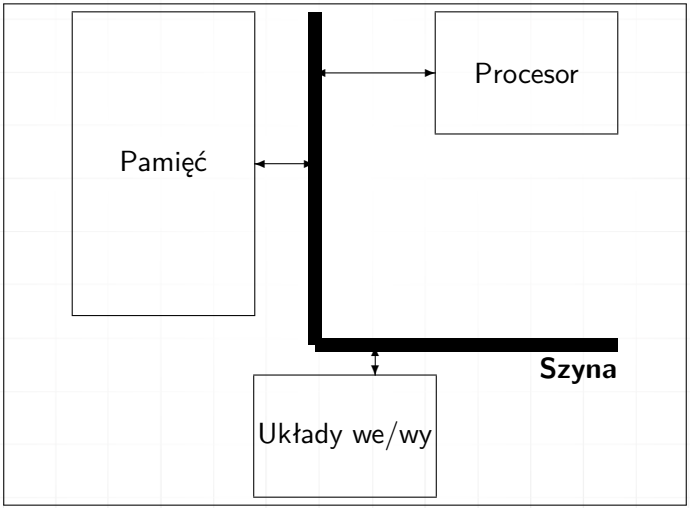
Rysunek: Schemat kmputera nawiazujący do architektury von Neumanna

Schemat komputera



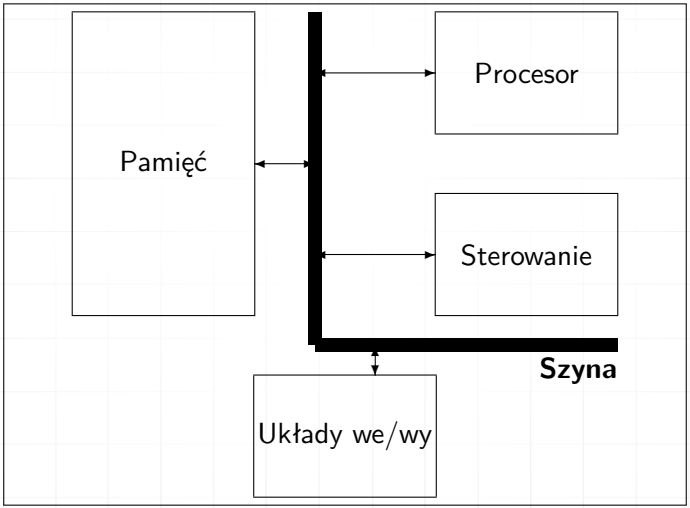
Rysunek: Schemat kmputera nawiazujący do architektury von Neumanna

Schemat komputera



Rysunek: Schemat kmputera nawiazujący do architektury von Neumanna

Schemat komputera



Rysunek: Schemat kmputera nawiazujący do architektury von Neumanna

Schemat komputera I

W modelu tym wyróżniamy:

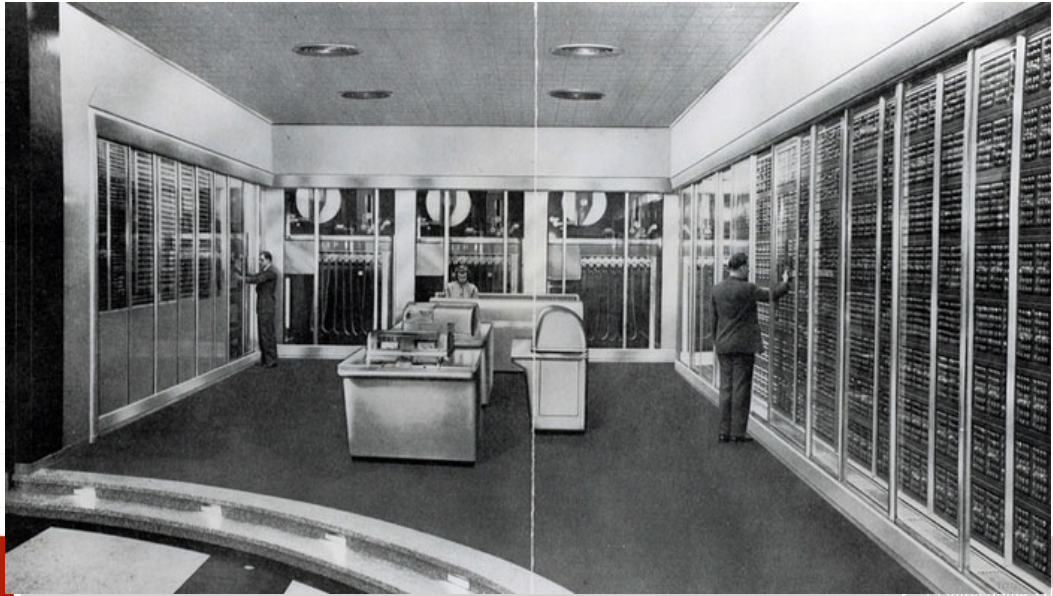
- Procesor.
- Pamięć (i pod tym pojęciem rozumiemy wszystkie rodzaje pamięci: Cache, RAM, ROM, dyski, dyskietki, dyski wymienne — zapisywalne i nie).
- Urządzenie wejścia/wyjścia (wszystkie urządzenia pozwalające na kontakt ze „światem zewnętrznym": klawiatura, mysz, karta graficzna, drukarka, czytniki różnego rodzaju).
- Sterowanie (wszystkie układy elektroniczne zapewniające właściwą komunikację wszystkich wymienionych wyżej urządzeń ze sobą i zapewniający uporządkowany przepływ informacji po szynie/szynach).
- Magistrala (szyna) — to wszystkie „drogi" łączące wymienione wyżej urządzenia.

Zanim przejdziemy dalej zastanówmy się co to jest komputer.

No właśnie...

...co to jest komputer?

???



???

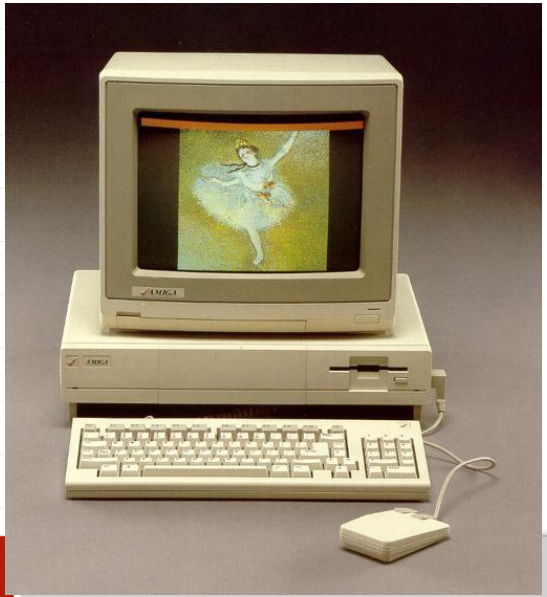
???


nazwa: BEM
procesor: Intel Xeon E5 2670v3
flops:
jeden węzeł: 883.2 GFLOPS
całość: 636 TFLOPS
właściciel: WCSS
system: Linux

???


commodore
PET
2001 Series
personal
computer

???


AMIGA
AMIGA

???


IBM
IBM

???

???


Sinclair ZX81 Personal Computer-
the heart of a system
that grows with you.
Kit:
£49.95
Built:
£69.95
16K-byte RAM
pack for massive
add-on memory.
Available now-
the ZX Printer
for only £49.95
ZX81
New BASIC manual
sinclair
ZX81

???


SAMSUNG

???

???


51°
TUES
WED
THUR
FRI

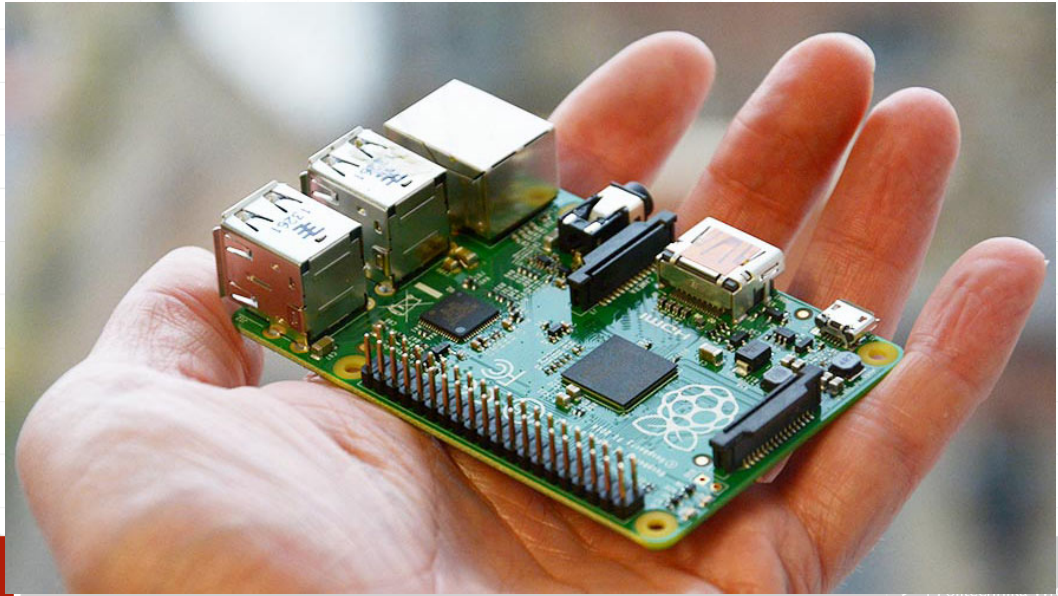
???


11:35
Google
11:35
WED, JULY 16

???


MODECOM
intel
inside

???



???


BUSINESS ELITE
VISA

???


twelve
six
12
06
PM
Tue
Jul 8
Laura Harrison
12
06
PM
Laura Harrison
PM
12
06
Tue,
Jul 8
28
steps

Kalkulator

Wyświetlacz			
1	2	3	$+$
4	5	6	$-$
7	8	9	$*$
0	C	$=$	$\div$

Kalkulator

Wyświetlacz			
1	2	3	$+$
4	5	6	$-$
7	8	9	$*$
0	C	$=$	$\div$

- Wyobraźmy sobie kalkulator. Taki najprostszy, czterodziałaniowy.

Kalkulator

Wyświetlacz			
1	2	3	$+$
4	5	6	$-$
7	8	9	$*$
0	C	$=$	$\div$

- Wyobraźmy sobie kalkulator. Taki najprostszy, czterodziałaniowy.
- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.

Kalkulator

Wyświetlacz			
1	2	3	$+$
4	5	6	$-$
7	8	9	$*$
0	C	$=$	$\div$

- Wyobraźmy sobie kalkulator. Taki najprostszy, czterodziałaniowy.
- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu

Kalkulator

			1
1	2	3	+
4	5	6	−
7	8	9	∗
0	C	=	÷

- Wyobraźmy sobie kalkulator. Taki najprostszy, czterodziałaniowy.
- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu

Kalkulator

			12
1	2	3	+
4	5	6	−
7	8	9	∗
0	C	=	÷

- Wyobraźmy sobie kalkulator. Taki najprostszy, czterodziałaniowy.
- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu

Kalkulator

			123
1	2	3	+
4	5	6	−
7	8	9	*
0	C	=	÷

- Wyobraźmy sobie kalkulator. Taki najprostszy, czterodziałaniowy.
- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu

Kalkulator

			123
1	2	3	+
4	5	6	−
7	8	9	*
0	C	=	÷

- Wyobraźmy sobie kalkulator. Taki najprostszy, czterodziałaniowy.
- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu
- I tu pojawia się podejrzenie, że wprowadzana liczba jest gdzieś zapamiętywana. Jak oderwiemy palec od klawiatury — wartości nie znikają. Pamięć ta jest (jakoś) powiązana z wyświetlaczem (i klawiaturą).

Kalkulator

			123
1	2	3	+
4	5	6	−
7	8	9	*
0	C	=	÷

- Wyobraźmy sobie kalkulator. Taki najprostszy, czterodziałaniowy.
- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu
- I tu pojawia się podejrzenie, że wprowadzana liczba jest gdzieś zapamiętywana. Jak oderwiemy palec od klawiatury — wartości nie znikają. Pamięć ta jest (jakoś) powiązana z wyświetlaczem (i klawiaturą).
- Musimy zmodyfikować nasz schemat.

Kalkulator

Wyświetlacz			
Akumulator			
1	2	3	$+$
4	5	6	$-$
7	8	9	$*$
0	C	$=$	$\div$

- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.

Kalkulator

Wyświetlacz			
Akumulator			
1	2	3	$+$
4	5	6	$-$
7	8	9	$*$
0	C	$=$	$\div$

- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Nie widać pamięci nazwanej tu akumulatorem, ale chyba ona tam jest.

Kalkulator

Wyświetlacz			
Akumulator			
1	2	3	$+$
4	5	6	$-$
7	8	9	$*$
0	C	$=$	$\div$

- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Nie widać pamięci nazwanej tu akumulatorem, ale chyba ona tam jest.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu i w akumulatorze.

Kalkulator

			1
			1
1	2	3	+
4	5	6	−
7	8	9	*
0	C	=	÷

- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Nie widać pamięci nazwanej tu akumulatorem, ale chyba ona tam jest.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu i w akumulatorze.

Kalkulator

			12
			12
1	2	3	+
4	5	6	−
7	8	9	*
0	C	=	÷

- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Nie widać pamięci nazwanej tu akumulatorem, ale chyba ona tam jest.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu i w akumulatorze.

Kalkulator

			123
			123
1	2	3	+
4	5	6	−
7	8	9	*
0	C	=	÷

- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Nie widać pamięci nazwanej tu akumulatorem, ale chyba ona tam jest.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu i w akumulatorze.

Kalkulator

			123
			123
1	2	3	+
4	5	6	−
7	8	9	*
0	C	=	÷

- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Nie widać pamięci nazwanej tu akumulatorem, ale chyba ona tam jest.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu i w akumulatorze.
- Naciskamy klawisz operacji. Niech to będzie +

Kalkulator

			123
			123
1	2	3	+
4	5	6	−
7	8	9	*
0	C	=	÷

- Na pierwszy rzut oka widać klawiaturę (z cyframi i podstawowymi klawiszami operacji) oraz wyświetlacz.
- Nie widać pamięci nazwanej tu akumulatorem, ale chyba ona tam jest.
- Gdy zaczynamy naciskać klawisze cyfr (123) odpowiednie informacje pojawiają się na wyświetlaczu i w akumulatorze.
- Naciskamy klawisz operacji. Niech to będzie +

Kalkulator

			123
			123
1	2	3	+
4	5	6	−
7	8	9	∗
0	C	=	÷

- Na wyświetlaczu nic się nie zmieniło, ale zmieniło się zachowanie kalkulatora: kolejne wprowadzane wartości powodują skasowanie i zastąpienie wyświetlanej liczby na wyświetlaczu.

Kalkulator

			123
			123
1	2	3	+
4	5	6	−
7	8	9	*
0	C	=	÷

- Na wyświetlaczu nic się nie zmieniło, ale zmieniło się zachowanie kalkulatora: kolejne wprowadzane wartości powodują skasowanie i zastąpienie wyświetlanej liczby na wyświetlaczu.
- Ale pierwsza wprowadzona wartość nie „ginie”. Jest gdzieś zapamiętana i będzie użyta w operacji (dodawania).

Kalkulator

			123
			123
1	2	3	+
4	5	6	−
7	8	9	*
0	C	=	÷

- Na wyświetlaczu nic się nie zmieniło, ale zmieniło się zachowanie kalkulatora: kolejne wprowadzane wartości powodują skasowanie i zastąpienie wyświetlanej liczby na wyświetlaczu.
- Ale pierwsza wprowadzona wartość nie „ginie”. Jest gdzieś zapamiętana i będzie użyta w operacji (dodawania).
- Potrzebna jest modyfikacja — musimy dodać kolejną pamięć. Akumulator wykorzystywany będzie podczas wprowadzania danych z klawiatury i do wyświetlania wyników. Liczba tam zawarta zawsze będzie jednym z argumentów operacji dwuargumentowych.

Kalkulator

			123
			123
1	2	3	+
4	5	6	−
7	8	9	∗
0	C	=	÷

- Ale pierwsza wprowadzona wartość nie „ginie”. Jest gdzieś zapamiętana i będzie użyta w operacji (dodawania).
- Potrzebna jest modyfikacja — musimy dodać kolejną pamięć. Akumulator wykorzystywany będzie podczas wprowadzania danych z klawiatury i do wyświetlania wyników. Liczba tam zawarta zawsze będzie jednym z argumentów operacji dwuargumentowych.
- Dodatkowa pamięć przechowywać będzie drugi z argumentów.

Kalkulator

Wyświetlacz			
Akumulator			
1	2	3	$+$
4	5	6	$-$
7	8	9	$*$
0	C	$=$	$\div$
Pamięć			

Teraz widać już wszystkie (widoczne i nie) elementy składowe kalkulatora.

Kalkulator

Wyświetlacz			
Akumulator			
1	2	3	$+$
4	5	6	$-$
7	8	9	$*$
0	C	$=$	$\div$
Pamięć			

Operacje

Kalkulator

			123
			123
1	2	3	+
4	5	6	−
7	8	9	∗
0	C	=	÷
			Pamięć

Operacje
123

Kalkulator

			123
			123
1	2	3	+
4	5	6	−
7	8	9	∗
0	C	=	÷
			123

Operacje
123
+

Kalkulator

			55
			55
1	2	3	+
4	5	6	−
7	8	9	∗
0	C	=	÷
			123

Operacje
123
+
55

Kalkulator

			178
			178
1	2	3	+
4	5	6	−
7	8	9	∗
0	C	=	÷
			123

Operacje
123
+
55
+

Kalkulator

			22
			22
1	2	3	+
4	5	6	−
7	8	9	*
0	C	=	÷
			178

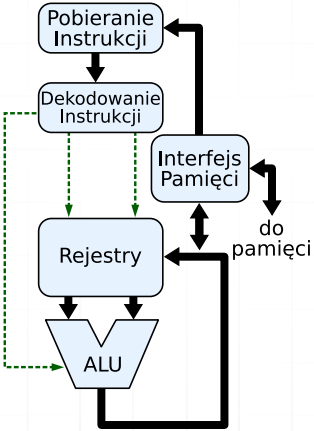
Operacje
123
+
55
+
22

Kalkulator

			200
			200
1	2	3	+
4	5	6	−
7	8	9	∗
0	C	=	÷
			178

Operacje
123
+
55
+
22
=

Procesor



Rysunek: Schemat działania procesora

Podstawowe operacje

Instrukcje arytmetyczne

- `Ładuj` <*adres pamięci*> przepisuje zawartość pamięci o wskazanym adresie do rejestru.
- `Zapisz` <*adres pamięci*> przepisuje zawartość akumulatora do pamięci
- `Ładuj` <*liczba*> zapisuje liczbę do rejestru
- `Dodaj` <*adres pamięci*> do zawartości akumulatora dodaje zawartość komórki o wskazanym adresie (możemy tez założyć, że w podobny sposób potrafi policzyć różnicę, iloczyn i iloraz, choć, w rzeczywistości, nie musi to być prawdą). Wykonanie każdej operacji zmieniającej zawartość rejestru powoduje ustawienie wskaźników (zero, przepełnienie, ujemne).

Podstawowe operacje

Instrukcje na bitach

- `Neguj` zmienia znak liczby w akumulatorze
- `And` <*adres pamięci*> iloczyn logiczny (bit po bicie dwu słów)
- `Or` <*adres pamięci*>
- `Xor` <*adres pamięci*> — różnica symetryczna
- `Przesun_w_lewo`
- `Przesun_w_prawo`
- `Przesun_cyklicznie_w_lewo`
- `Przesun_cyklicznie_w_prawo`

Podstawowe operacje

Instrukcje sterujące

- `Skocz` <*adres pamięci*> bezwarunkowe przekazanie sterowanie do adresu
- `Skocz_jezeli_zero` <*adres pamięci*>
- `Skocz_jezeli_ujemne` <*adres pamięci*>
- `Skocz_jesli_nadmiar` <*adres pamięci*>
- `Skocz_do_podprogramu` <*adres pamięci*> bardzo podobne do instrukcji zwykłego skoku, ale dodatkowo zapisuje aktualny stan procesora w specjalnie do tego przeznaczonej pamięci

Asembler

Bardzo proste działanie:
`A=B+C`

Asembler

Bardzo proste działanie:
`A=B+C`
W komórce o adresie `A` ma być umieszczony wynik dodawania zawartości komórek o adresie `B` i `C`.

Asembler

Bardzo proste działanie:
`A=B+C`
W komórce o adresie `A` ma być umieszczony wynik dodawania zawartości komórek o adresie `B` i `C`.
Realizacja komputerowa:

```
Ładuj B
Dodaj C
Zapisz A
```

Asembler

Bardziej skomplikowany przykład

$$Z = \frac{[(A + B)(C + D)]}{W}$$

Asembler

Bardziej skomplikowany przykład

$$Z = \frac{[(A + B)(C + D)]}{W}$$

$$T1 = A + B$$

Asembler

Bardziej skomplikowany przykład

$$Z = \frac{[(A + B)(C + D)]}{W}$$

$$T1 = A + B$$
$$T2 = C + D$$

Asembler

Bardziej skomplikowany przykład

$$Z = \frac{[(A + B)(C + D)]}{W}$$

$$T1 = A + B$$
$$T2 = C + D$$
$$T3 = T1 * T2$$

Asembler

Bardziej skomplikowany przykład

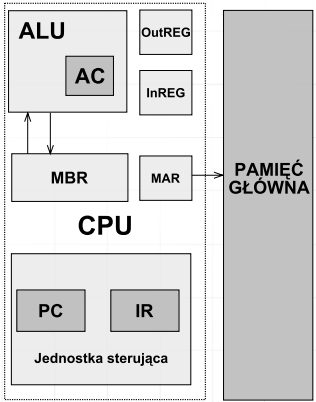
$$Z = \frac{[(A + B)(C + D)]}{W}$$

$$T1 = A + B$$
$$T2 = C + D$$
$$T3 = T1 * T2$$
$$Z = T3/W$$

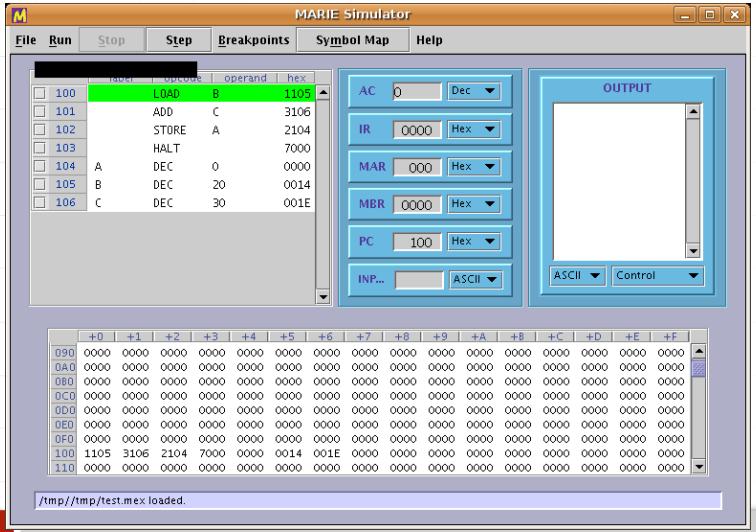
Ten rozdział jest nieobowiązkowy.

MARIE — A Machine Architecture that is Really Intuitive and Easy

- notacja dwójkowa, zapis w kodzie dopełnieniowym
- przechowywanie programu, stała długość słowa
- adresowanie słowne
- 4K pamięci głównej (12 bitów na każdy adres)
- 16-bitowe dane (16-bitowe słowa)
- 16-bitowe rozkazy (4-bitowy kod operacji + 12-bitowy adres)
- 16-bitowy akumulator (AC)
- 16-bitowy rejestr rozkazów (IR)
- 16-bitowy rejestr bufora pamięci (MBR)
- 12-bitowy licznik rozkazów (PC)
- 12-bitowy rejestr adresów pamięci (MAR)
- 8-o bitowy rejestr wejściowy (InREG)
- 8-o bitowy rejestr wyjściowy (OutREG)



Symulator MARIE



Zadanie domowe

Prosty program dla komputera MARIE (na przykład mnożenie dwu liczb)...

Literatura dodatkowa:

MARIE.
https://pl.wikipedia.org/wiki/MARIE.
Only in Polish.

Student resources – essentials of computer organization and architecture, second edition.
http://samples.jbpub.com/9781284123036/9781284136852_FMxx_Print_Final.pdf.
Only Table of Contents and Preface.

Kuo-pao Yang.
MARIE: An introduction to a simple computer.
https://www2.southeastern.edu/Academics/Faculty/kyang/2013/Spring/CMPS375/ClassNotes/CMPS375ClassNotesChap04.pdf, 2013.

Linda Null and Julia Lobur.
The essentials of computer organization and architecture.
Jones and Bartlett Publishers, Sudbury, Mass., 2006.

Linda Null and Julia Lobur.
MARIE: an introduction to a simple computer.
In *The essentials of computer organization and architecture*. 2006.
http://samples.jbpub.com/9781449600068/00068_CH04_Null3e.pdf.

Linda Null and Julia Lobur.
A guide to the MARIE machine simulator environment, 2010.
https://cs.msutexas.edu/~simpson/wordpress/wp-content/uploads/2012/12/MarieGuide.pdf.

Odwrotna Notacja Polska

Popatrzmy na działanie:

$$3 + 7 \times 5$$

Ile wynosi wynik?

Odwrotna Notacja Polska

Popatrzmy na działanie:

$$3 + 7 \times 5$$

Ile wynosi wynik?

50 czy 38?

Odwrotna Notacja Polska

Popatrzmy na działanie:

$$3 + 7 \times 5$$

Ile wynosi wynik?

50 czy 38?

A który jest poprawny?

Odwrotna Notacja Polska

Popatrzmy na działanie:

$$3 + 7 \times 5$$

Ile wynosi wynik?

50 czy 38?

A który jest poprawny?

Czemu tak łatwo znaleźć kalkulator który liczy „źle”?

„Ważność” działań arytmetycznych

1. potęgowanie
2. mnożenie i dzielenie
3. dodawanie i odejmowanie

Nawiasy mogą ją zmieniać!

„Ważność” działań arytmetycznych

1. potęgowanie
2. mnożenie i dzielenie
3. dodawanie i odejmowanie

Nawiasy mogą ją zmieniać!

A gdzie jest zmiana znaku?

Czy jest możliwy zapis jednoznaczny?

Polski logik, Łukasiewicz, wprowadził notację „przedrostkową". Zamiast $z = x + y$ zaproponował zapis:

$$+xy$$

Czy jest możliwy zapis jednoznaczny?

Polski logik, Łukasiewicz, wprowadził notację „przedrostkową". Zamiast $z = x + y$ zaproponował zapis:

$$+xy$$

Zwracam uwagę że jest on bardzo podobny do zapisu funkcji dwu zmiennych:

$$z = f(x, y)$$

Funkcja **suma** jest też dwuargumentowa:

$$z = +(x, y)$$

Zapis polski

Działanie $3 + 7 \times 5$ oznaczające $3 + (7 \times 5)$ zapisujemy:

$$\underbrace{+\ \underbrace{\times 7\ 5}_{35}\ 3}_{38}$$

Odwrotny zapis polski

Utarło się używanie innego zapisu: najpierw podaje się argumenty działania, później samo działanie:

$$xy+$$

Stąd nazwa: „Odwrotna notacja polska".
Nasze działanie zapisujemy tak:

$$7\ 5 \times 3+$$

a to bardziej skomplikowane tak:

$$A\ B\ +\ C\ D\ + \times W\ /$$

Odwrotna Notacja Polska — stos

Praktyczna realizacja działania

$$A\ B\ +\ C\ D\ +\times W\ /$$

wymaga stosu. I dodatkowych operacji w języku wewnętrznym:

- `Zapisz_na_stos` przepisuje zawartość akumulatora na stos.
- `Pobierz_ze_stosu` pobiera ze stosu wartość i przepisuje ją do akumulatora

Odwrotna Notacja Polska — stos

Praktyczna realizacja działania

$$A\ B\ +\ C\ D\ +\times W\ /$$

wymaga stosu. I dodatkowych operacji w języku wewnętrznym:

- `Zapisz_na_stos` przepisuje zawartość akumulatora na stos.
- `Pobierz_ze_stosu` pobiera ze stosu wartość i przepisuje ją do akumulatora

Stos czasami jest nazywany pamięcią LIFO (Last In First Out)...

Odwrotna Notacja Polska — stos

Praktyczna realizacja działania

$$A\ B\ +\ C\ D\ +\times W\ /$$

wymaga stosu. I dodatkowych operacji w języku wewnętrznym:

- `Zapisz_na_stos` przepisuje zawartość akumulatora na stos.
- `Pobierz_ze_stosu` pobiera ze stosu wartość i przepisuje ją do akumulatora

Stos czasami jest nazywany pamięcią LIFO (Last In First Out)...

Kolejka jest nazywana pamięcią FIFO (First In First Out).

Idea stosu