

Logiczny model komputera

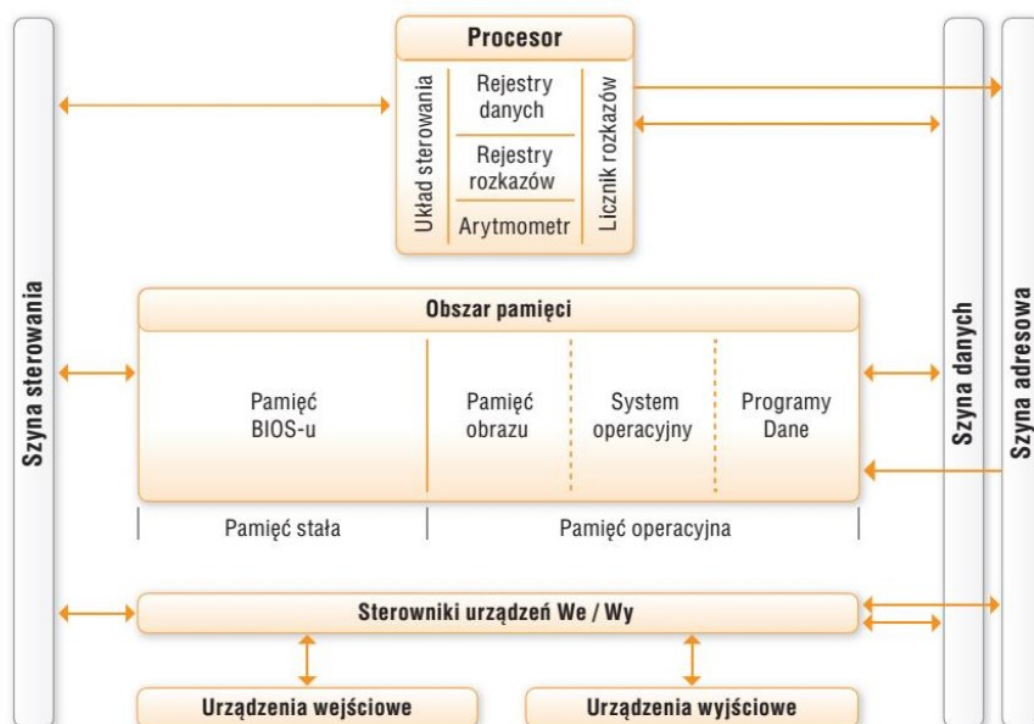
Od wprowadzenia komputerów osobistych na rynek minęło już ponad 30 lat, a od opracowania logicznego modelu komputera przez John von Neumanna - przeszło pół wieku. Ogólne zasady ich działania pozostały jednak niezmiennie i współczesne komputery są wciąż budowane według pierwotnej koncepcji.

Idea maszyny von Neumanna:

Programy i dane posiadają taką samą postać,

Programy i dane są przechowywane w tej samej pamięci.

Zgodnie z tą koncepcją dane mogą być traktowane jako instrukcje programu komputerowego, a instrukcje programu komp. jako dane.



1. Uproszczony model komputera zgodny z ideą von Neumanna

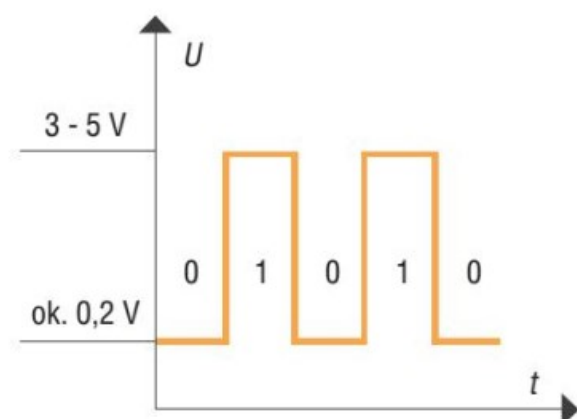
Na czym polega działanie procesora?

Głównym elementem komputera jest procesor (CPU - *Central Processing Unit*). Współczesny procesor to pojedynczy układ scalony, oparty na strukturze krzemowej, zawierającej od kilkudziesięciu milionów do kilkunastu miliardów tranzystorów.

Procesor wykonuje operacje arytmetyczne i logiczne oraz odpowiada za prawidłową współpracę wszystkich modułów funkcjonalnych komputera. Szybkość taktowania współczesnego procesora wynosi kilka gigaherców, zatem w jednej sekundzie taki procesor potrafi wykonać kilka miliardów operacji (w przypadku procesorów wielordzeniowych liczbę tę należy pomnożyć przez liczbę rdzeni procesora). W ich zakres wchodzi m in. operacje arytmetyczne (np. dodawanie, odejmowanie) i logiczne (np. suma logiczna, iloczyn, negacja, porównywanie) oraz operacje przesyłania danych z i do pamięci operacyjnej. Zbiór tych operacji nazywamy **listą rozkazów procesora**. W oparciu o możliwości listy rozkazów konstruowane są programy komputerowe.

Rozkazy procesory to dane zapisane w postaci zero-jedynkowej, przy czym każdemu rozkazowi odpowiada inna sekwencja danych. Fizyczne rozkazy, tak jak i dane istnieją w komputerze w postaci sygnałów elektrycznych.

Przyjmuje się, że obecność sygnału odpowiada cyfrze 1, a brak sygnału cyfrze 0. Każda z tych wartości (0,1) stanowi 1 bit informacji.



OZNACZENIE	NAZWA
b	bit (0 lub 1)
B	Bajt
1 Bajt	8 bitów
1 Kb (1 kilobit)	1024 b
1 Mb (megabit)	1024 Kb
1 KB (kilobajt)	1024 B
1 MB (megabajt)	1024 KB
1 GB (gigabajt)	1024 MB
1 TB (terabajt)	1024 GB

Bit jako porcja informacji

Żeby komputer mógł wykonywać jakiekolwiek programy i operować danymi, muszą być one dla niego dostępne. Miejscem, w którym są umieszczane, jest **pamięć operacyjna**. Chociaż fizycznie pamięć operacyjna może być zrealizowana w różny sposób (jako pamięć półprzewodnikowa, wirtualna - odwzorowana za pomocą pamięci masowej) - procesor widzi ją jako jednolity obszar pamięciowy.

Pamięć operacyjną możemy sobie wyobrazić jako zbiór ponumerowanych komórek. Numer komórki jest jej adresem. Zawartość komórki stanowią dane.

Danymi mogą być:

- wartości wprowadzone przez użytkownika,
- wartości będące wynikiem działania programu,
- rozkazy procesora,
- adresy innych komórek,
- wartości pochodzące z urządzeń wejścia.

ADRES KOMÓRKI	ZAWARTOŚĆ
0	
1	ROZKAZ PROCESORA
K-1	Wartość (dane)
K+1	Adres innej komórki
...	...
n	

Pamięć RAM

Procesor komunikuje się z pamięcią przez **szyny** (magistrale): adresową i danych. Szyna adresowa jest jednokierunkowa. Za jej pośrednictwem procesor uaktywnia odpowiednie komórki pamięci, umożliwiając tym samym zapis lub odczyt ich zawartości. Szyna danych jest dwukierunkowa. Pozwala na przesyłanie danych do i z procesora.

W jaki sposób procesor dodaje liczby?

Wykonanie programu umieszczonego w pamięci operacyjnej zaczyna się od pobrania pierwszego kodu rozkazu tego programu. Procesor żąda dostępu do niego, wstawiając na szynę adresową adres komórki, w której się znajduje. Pod żądanym adresem uaktywnia się odczyt zawartości komórki, zawartość ta jest przesyłana do procesora, gdzie zostaje umieszczona w rejestrze. Jeżeli jest to kod rozkazu, to zostanie on wykonany.

Przykładem może być operacja dodania dwóch liczb (zmiennych) oraz umieszczenie jej w trzeciej zmiennej:

Założenia: dodali liczby x=4 i y=5

Zadanie	Adres	Co robi
move.1	(ax), d0	Przesłanie liczby z adresu ax(4) do rejestru d0
move.1	(ay), d1	Przesłanie liczby z adresu ay(5) do rejestru d1
add.1	d0, d1	Operacja sumowania i zapisanie wyniku w rejestrze d1
move.1	d1, (az)	Przesłanie liczby z rejestru d1(9) do adresu (az) w pamięci

Przykład 1. Dodawanie liczb przez procesor.

W pamięci operacyjnej programu i dane umieszczane są tylko na czas pracy komputera. Dane, które muszą być pamiętane, gdy komputer nie pracuje, zapisujemy w pamięci masowej.

ZADANIA:

1. Jaka jest główna idea maszyny von Neumanna? Na jakie części dzieli komputer?
2. Jak reprezentowane są rozkazy procesora?
3. W jaki sposób dane są umieszczane w pamięci operacyjnej?
4. Dokonaj porównania architektury Harvardzkiej i von Neumanna
5. Wyszukaj informacje na temat pierwszej udokumentowanej architektury komputera? Czego dokładnie dotyczyła?

Sprawdź jaki rysunek utworzą cyfry 1 gdy zapiszesz rozwinięcia dwójkowe liczb: (41702, 45705, 45801, 43657, 43657, 42633, 42633, 41606)?

* Wskazówka: Polecenie najlepiej wykonać w tabeli np. Excel.
Możesz w rozwinięciach dwójkowych zastąpić jedynki np. literami X, zer nie wpisuj.