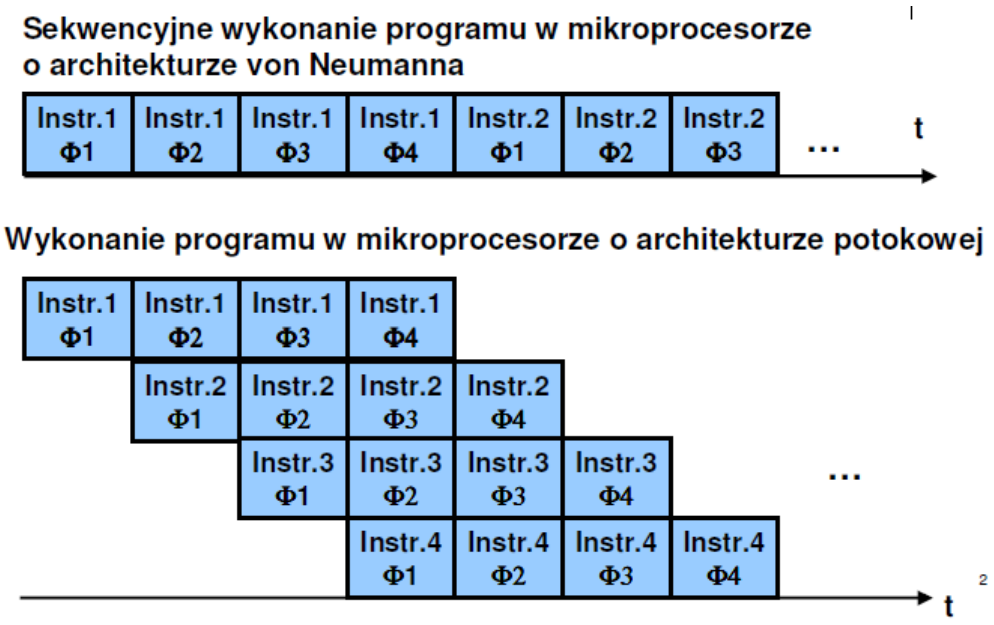
Tendencje rozwojowe w architekturze procesorów

Architektura komputera – sposób organizacji elementów tworzących komputer.

Architektura potokowa.



Cechy architektury potokowej:
- Podział wykonania instrukcji na etapy
- Jednoczesne wykonanie różnych etapów kilku instrukcji
- Przetwarzane dane organizowane są w tzw. potok (dane i rozkazy przepływają przez mikroprocesor)
- Konieczne jest zapamiętanie pośrednich wyników przetwarzania i związany z tym narzut sprzętu

Mikroprocesory o architekturze potokowej posiadają zazwyczaj uproszczona listę instrukcji i wykonują operacje ALU jedynie na wewnętrznych rejestrach roboczych
Zwane są mikroprocesorami RISC (ang. Reduced Instruction Set Computer) w odróżnieniu od mikroprocesorów zgodnych z architektura von Neumanna, zwanych CISC

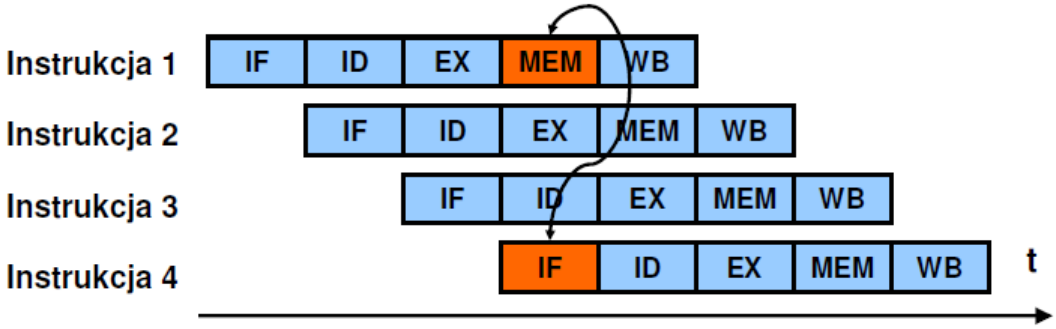
Fazy instrukcji w architekturze potokowej
Wszystkie instrukcje są dzielone na tę samą liczbę faz, np.:
- IF – pobranie kodu instrukcji z pamięci
- ID – dekodowanie instrukcji
- EX – wykonanie instrukcji
- MEM – dostęp do pamięci
- WB – zapisanie wyniku do bufora

Jeśli wykonanie danej instrukcji wymaga mniejszej liczby faz, wstawiane są fazy bezczynne.

Głębokość potoku jest to liczba stopni potoku (Pipeline Stages).

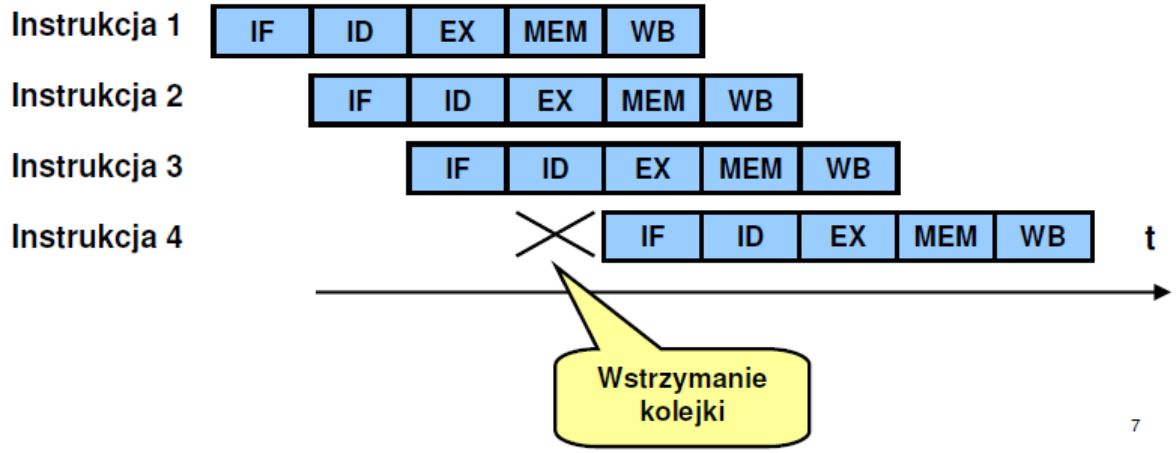
Konflikt zasobów

•W przypadku, gdy dwie instrukcje odwołują sie do tych samych zasobów komputera, występuje tzw. konflikt zasobów.



Konflikt zasobów - rozwiązanie problemu

W takich sytuacjach rozwiązaniem jest wstrzymanie kolejki instrukcji.



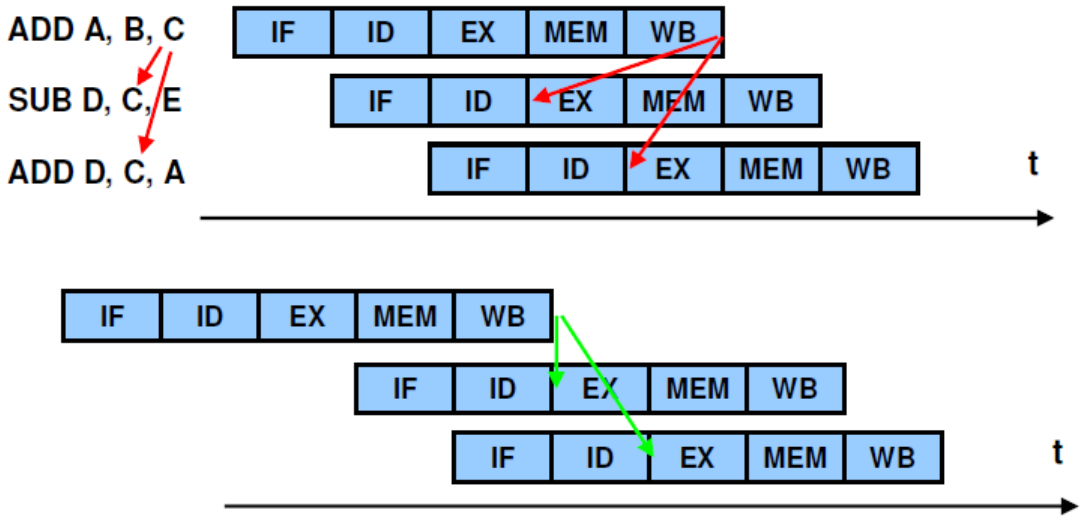
Konflikt zasobów - inne rozwiązanie problemu

- •W procesorze integruje sie tzw. pamięć podręczną (ang. cache).
- •Kod instrukcji jest jednocześnie pobierany do rejestru IR oraz do pamięci Cache.
- •Podczas kolejnego pobrania instrukcji z tego samego adresu, np. w pętli, kod jest pobierany z wewnętrznej pamięci Cache.
- •Daje to możliwość jednoczesnego dostępu do danych w pamięci operacyjnej.

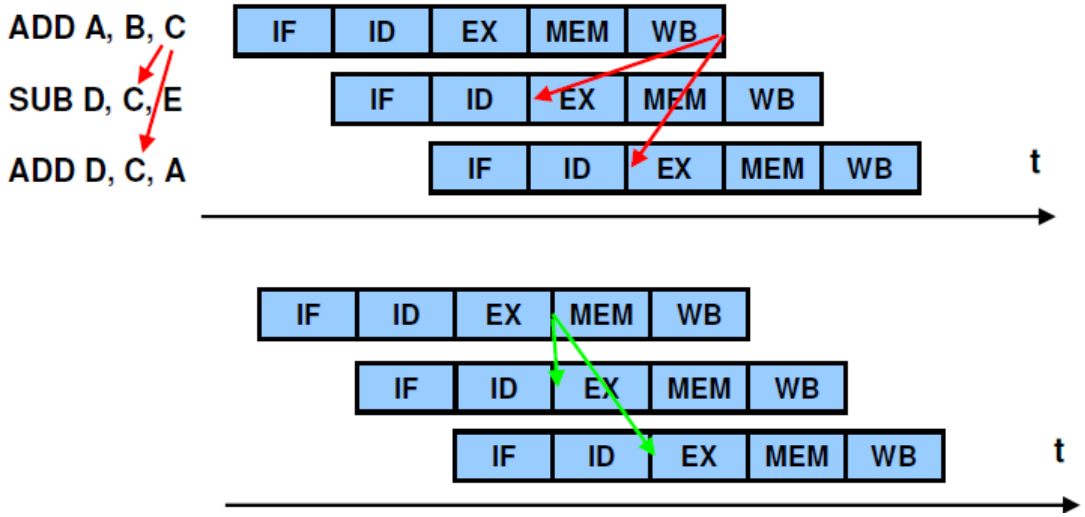
Ze względu na ograniczoną pojemność pamięci Cache wcześniej zapisane instrukcje są usuwane i zastępowane ostatnio pobranymi.

Konflikt danych

W przypadku, gdy argumenty danej instrukcji są wynikiem wykonania poprzedniej, nie są jeszcze dostępne.



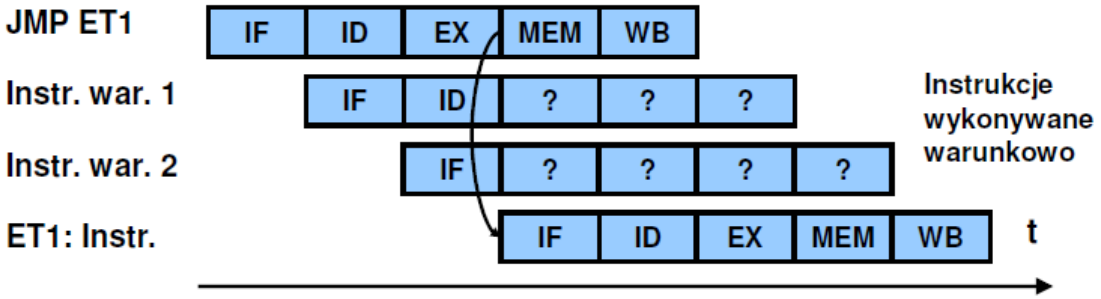
Konflikt danych - forwarding



Forwarding polega na przesyłaniu wyników działania z faz późniejszych do wcześniejszych, bez czekania na ich ostateczny zapis; realizowany jest poprzez sprzętowe porównywanie numerów rejestrów biorących udział w poszczególnych fazach instrukcji.

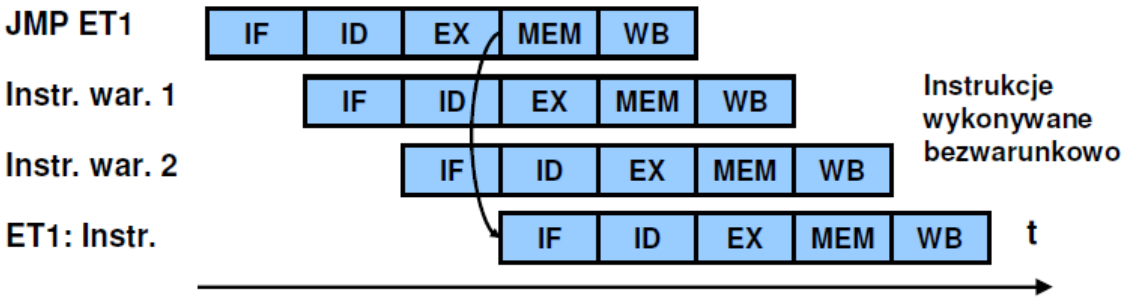
Konflikt sterowania

W przypadku skoku warunkowego nie wiadomo z góry które instrukcje są wykonywane jako następne.



Konflikt sterowania - opóźnienie skoku

Pewna liczba instrukcji po instrukcji skoku jest wykonywana niezależnie od wykonania skoku.



Techniki superskalarne.

Istnieje optymalna głębokość potoku (najczęściej ok. 8 stopni). Jeśli nie można zwiększać dalej liczby stopni ani szybkości taktowania procesora to można zwiększyć wydajność procesora stosując rozwiązanie wielopotokowe. Przykładem procesora o strukturze superskalarnej jest procesor Pentium, wyposażony w dwa równoległe kanały przetwarzające dane typu Integer, potoki całkowitoliczbowe oznaczone U i V oraz jeden zmiennoprzecinkowy FPU (Floating Point Unit).

Niestety, strumień rozkazów do wykonania zawiera wiele zależności i punktów rozgałęzień. Ponadto oba pracujące równolegle potoki korzystają z tych samych zasobów procesora np. rejestrów, stąd dalsze zwiększanie liczby potoków nie wpływa znacząco na wzrost wydajności – moc obliczeniowa procesora zużywana jest w większości przez wewnętrzny system komunikacji międzypotokowej.

Stosowany jest układ rozdzielania rozkazów, przemianowywania rejestrów (Register Renaming), wykonywanie nadmiarowe tej samej operacji w obu potokach, przewidywanie skoków (Branch Prediction) lub wykonywani obu dróg (Multiple Paths of Execution).
Speculative Execution – przewidywanie ścieżki programu wymaga rejestrowania historii skoków w tablicy BTB (Branch Target Buffer), która zawiera adres instrukcji, adresy skoków oraz 1 lub 2 bity informacji czy ostatni skok był efektywny.

Możliwości przetwarzania w dwóch potokach znalazły zastosowanie w technologii Hyper-Threading (HT) – możliwości symulowania dwóch procesorów na jednym rdzeniu. Wątek to część programu, która może być wykonywana niezależnie od innych. Każdy z procesorów logicznych ma do dyspozycji własny komplet rejestrów odzwierciedlający w pełni aktualny stan wszystkich układów (Architecture State), lokalny kontroler przerwań (APIC) oraz pewne dodatkowe struktury zarządzające.

Rozszerzenia procesora.

Przetwarzanie szerokiego strumienia danych pochodzenia multimedialnego (obrazy, dźwięki, projekcje wideo) oznaczało konieczność udostępniania nowych typów danych i operujących na nich rozkazów.

MMX (Multimedia Extension)

- dedykowana magistrala AGP do kontrolera grafiki,
- grupowanie danych w większe bloki, na których wykonywano ten sam rozkaz (Single Instruction Multiple Data),
- wewnętrzny stos powrotu,
- 64-bitowe rejestry MM0-MM7;

SSE (Streaming SIMD Extension)

Technika podobna do MMX lecz obejmuje również dane zmiennoprzecinkowe.

SSE2 i **SSE3** – rozszerzenia wprowadzone w Pentium 4 obejmujące nowe typy danych.

VLIW (Very Long Instruction Word)

Umożliwia równoległe przetwarzanie wielu instrukcji, przy czym muszą one być podane w jednym słowie rozkazowym (64-bitowym). Ciężar zrównoleglenia wykonania programu został przerzucony z obwodów procesora na kompilator języka wysokiego poziomu, który sam musi zdecydować które rozkazy mogą być wykonane równolegle.

Architektury równoległe

Podstawowe architektury używanych obecnie systemów komputerowych można podzielić:

1. Komputery z jednym procesorem
2. Komputery równoległe
3. Systemy rozproszone

Taksonomie architektur

Podstawowe typy architektur komputerowych

- Służą do klasyfikacji architektur komputerowych
- podział na kategorie
- określenie własności

Klasyfikacje

Ze względu na rodzaj połączeń procesor-pamięć i sposób ich wykorzystania dzielimy architektury zgodnie z **taksonomią Flynna**:

- SISD (ang. *Single Instruction Single Data*) – skalarne,
- SIMD (ang. *Single Instruction Multiple Data*) – wektorowe (macierzowe),
- MISD (ang. *Multiple Instruction Single Data*) – strumieniowe,
- MIMD (ang. *Multiple Instruction Multiple Data*) – równoległe.

Ze względu na **sposób podziału pracy i dostęp procesora do pamięci** możemy podzielić architektury na:

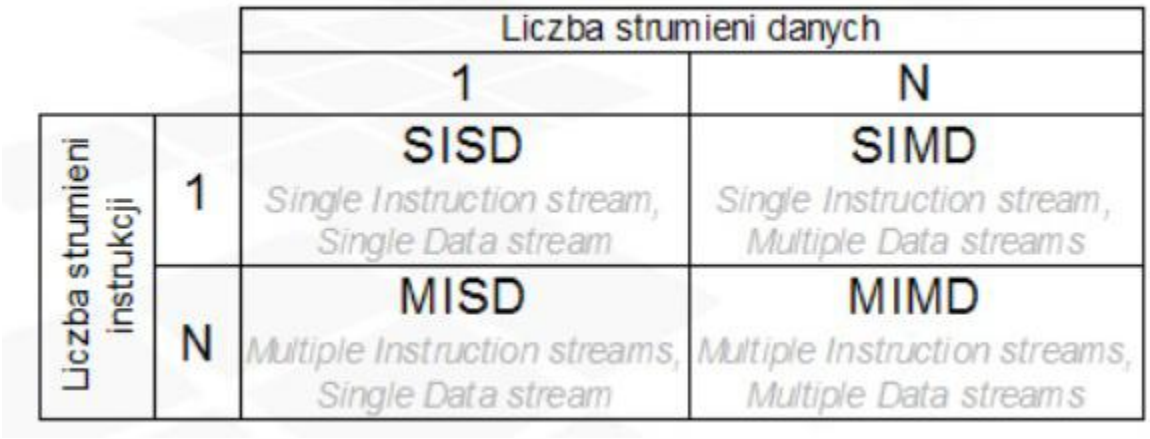
- SMP (ang. *Symmetric Multiprocessing*) – symetryczne,
- ASMP (ang. *Asymmetric Multiprocessing*) – asymetryczne,
- NUMA (ang. *Non-Uniform Memory Access*) – asymetryczne (rozróżniające pamięć lokalną i zdalną),
- AMP (ang. *Asynchronous Multiprocessing*) – asynchroniczne,
- MPP (ang. *Massively Parallel Processors*) – równoległe.

Taksonomia Flynna – 1968. Jest prosta; należy ją znać – ma znaczenie historyczne.
Zaproponowana przez Michaela J. Flynna około 1968 roku
Zakłada, że komputer jest urządzeniem przetwarzającym strumienie danych na podstawie strumieni instrukcji. Klasyfikuje komputery według liczby strumieni instrukcji i danych. Liczba strumieni może wynosić 1 lub więcej

Skrót	Nazwa angielska	Liczba strumieni instrukcji	Liczba strumieni danych
SISD	*Single Instruction Single Data*	1	1
SIMD	*Single Instruction Multiple Data*	1	wiele
MISD	*Multiple Instruction Single Data*	Wiele	1
MIMD	*Multiple Instruction Multiple Data*	Wiele	wiele

Klasyfikacja komputerów według Flynna

Klasyfikacja architektur systemów komputerowych wg Flynna ze względu na zrównoleglenie operacji w jednym cyklu zegara:



SISD (Single Instruction Stream, Single Data stream) – pojedyncza instrukcja, pojedyncza dana; klasyczne komputery jednoprocesorowe.

- pojedynczy strumień instrukcji i danych,
- Przykład uniprocesor von Neumanna,
- najbardziej rozpowszechniony typ architektury.

SIMD (Single Instruction Stream, Multiple Data stream) – pojedyncza instrukcja, wiele danych; komputery wektorowe, macierzowe.

- jeden strumień instrukcji, wiele strumieni danych
- jedna instrukcja powoduje wykonanie tej samej operacji na wielu kompletach danych
- przykład: procesor wektorowy lub macierzowy

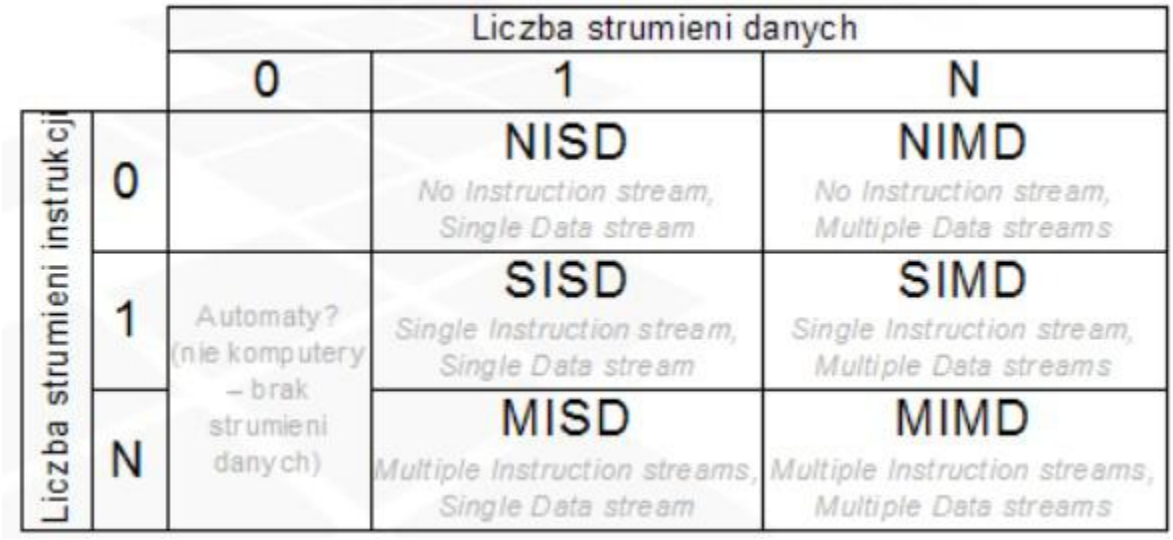
MISD (Multiple Instruction Stream, Single Data stream) – wiele instrukcji, pojedyncza dana (konstrukcje eksperymentalne).

- nie bardzo wiadomo co to jest, trudno wskazać wzorcowego reprezentanta (może procesor. potokowy – procesory graficzne)

MIMD (Multiple Instruction Stream, Multiple Data stream) – wiele instrukcji, wiele danych; różnorodne komputery równoległe

- przykład: wieloprocesor, wielokomputer, czyli maszyny złożone z wielu połączonych uniprocesorów von Neumanna

Rozszerzenie taksonomi Flyna



Urządzenie bez strumieni danych nie jest komputerem - komputer musi przetwarzać dane. W tej części tabeli można by umieścić niektóre automaty.

Urządzenia bez strumieni instrukcji mogą przetwarzać dane - same dane mogą nieść informację o potrzebnym przetwarzaniu. Są to tzw. komputery sterowane przepływem danych (Dataflow). Możemy wyróżnić uniprocesory i wieloprocesoyy Dataflow

Komputery sterowane przepływem danych

- Jednostka przetwarzana – ***token*** zawiera dane oraz „metkę" (***tag***) opisująca zawartość; metka jest odpowiednikiem instrukcji, opisującej co należy zrobić z danymi
- W wyniku przetworzenia uzyskuje się nowy token, zawierający zmodyfikowane dane i zmodyfikowany token
- Przetworzony token może podlegać dalszemu przetwarzaniu
- Maszyny Dataflow nie są obecnie konstruowane, około 1995 NEC produkowała mikroprocesor dataflow.
- Paradygmat przetwarzania sterowanego przepływem danych jest używany do opisu procesów przetwarzania informacji, nie ma nic wspólnego ze sprzętem komputerowym

Procesory macierzowe

Macierze jednostek przetwarzających dane (SIMD), zwane procesorami wektorowymi.

Systemy wieloprocesorowe

- Komputery zawierające więcej niż jeden mikroprocesor.
- Procesory w takim systemie maja podobne możliwości przetwarzania danych.
- Procesory pracują pod kontrola wielozadaniowego systemu operacyjnego.
- Komunikacja między procesorami odbywa sie przez wspólną pamięć operacyjną (ang. shared memory) lub przesyłanie komunikatów.

Systemy wielokomputerowe

Komputery połączone w sieć wspólnie realizujące wybrane zadanie obliczeniowe.

Sieci neuronowe

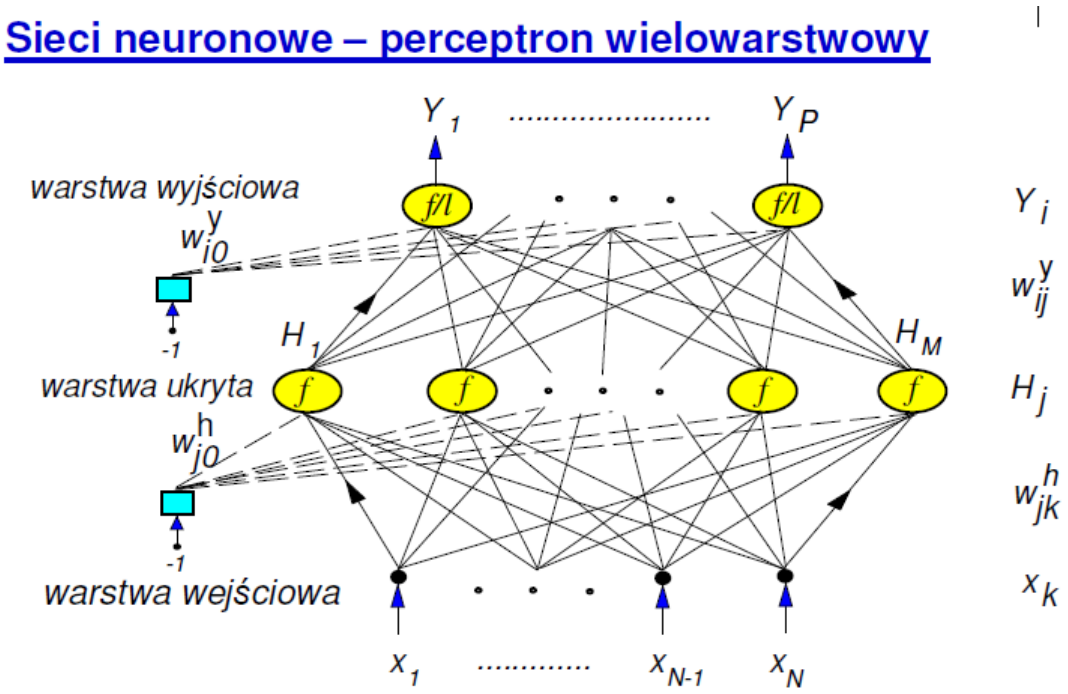
Sieć jednakowych, prostych procesorów – neuronów, połączonych za pomocą tzw. połączeń synaptycznych o regulowanej sile połączenia (wagi).

Cechy:

•Silne zrównoleglenie przetwarzania danych

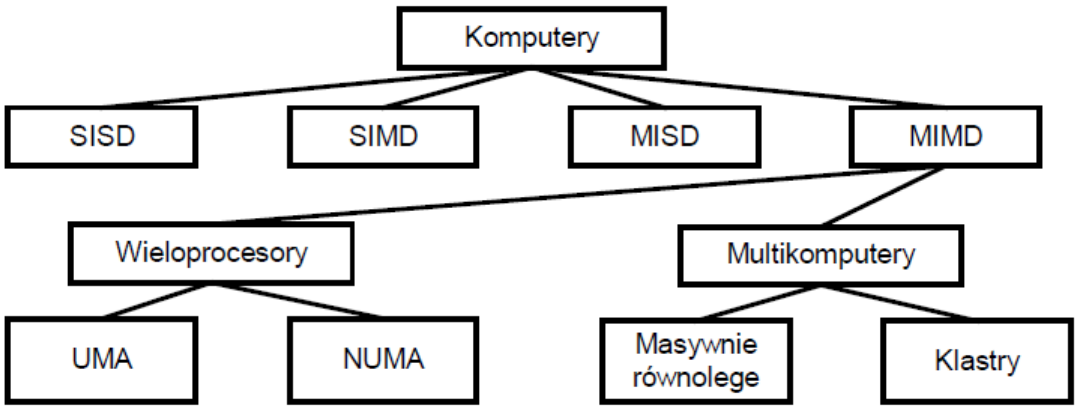
•Uczenie zamiast programowania

• stosowane w zadaniach rozpoznawania wzorów



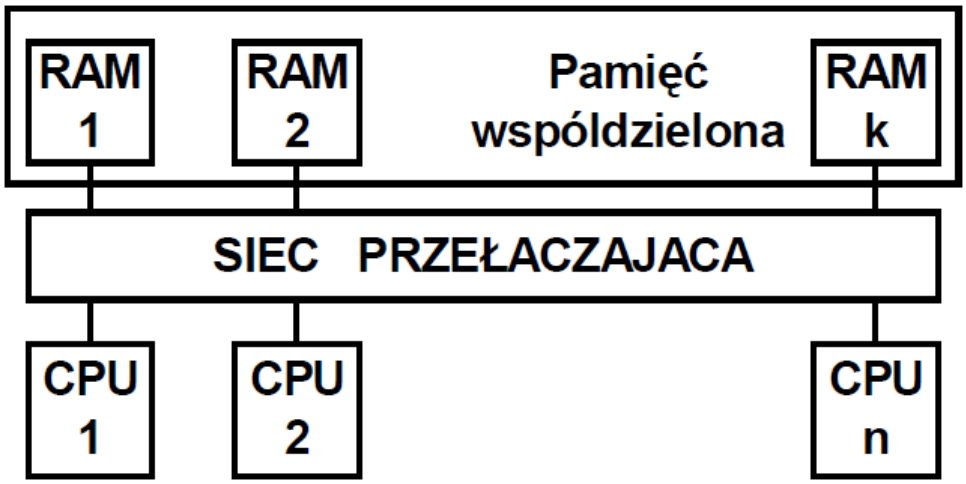
Klasyfikacja komputerów MIMD (wg Tannenbauma)

Podział bierze pod uwagę komunikację pomiędzy procesorami. Może ona być przez:
· wspólną pamięć
· system wejścia wyjścia



Multiprocesory

Multiprocesor składa się z pewnej liczby procesorów i modułów pamięci połączonych siecią przełączającą.



Schemat multiprocesora

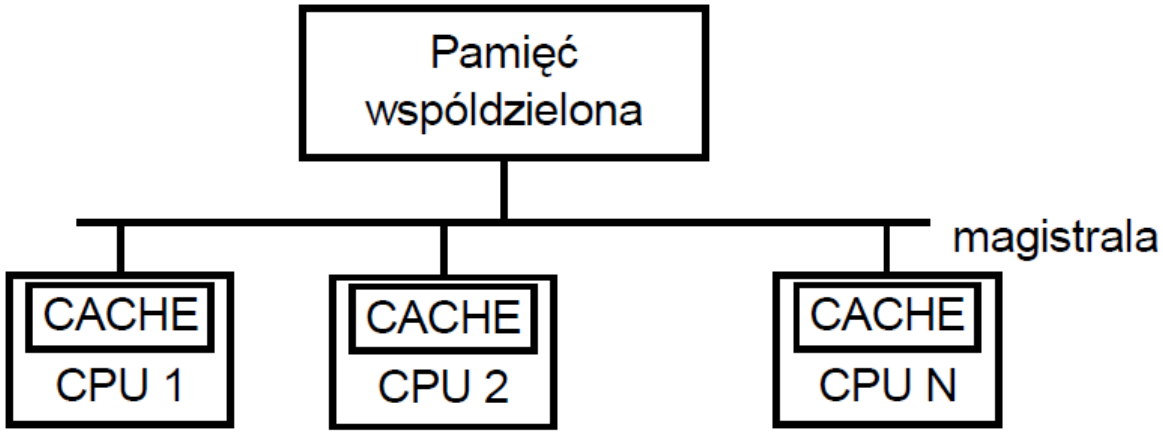
Wszystkie procesory multiprocesora dzielą tę samą przestrzeń adresową.
Na poziomie kodu maszynowego dostęp do pojedynczych komórek pamięci realizowany jest poprzez rozkazy typu LOAD i STORE.

Sieć przełączająca może być zbudowana w różny sposób.

- Przełącznica krzyżowa (ang. *Crossbar Switch*)
- Sieci wielostopniowe (ang. *Multistage Networks*),
- Magistrala.

Przełącznica krzyżowa - rozwiązanie kosztowne i w małym stopniu skalowalne. Dlatego stosuje się je w systemach, gdzie liczba procesorów nie przekracza kilkunastu.

Architektura magistrali - w danej chwili pamięć może być wykorzystywana tylko przez jeden procesor, co powoduje, że dostęp do pamięci staje się wąskim gardłem systemu.

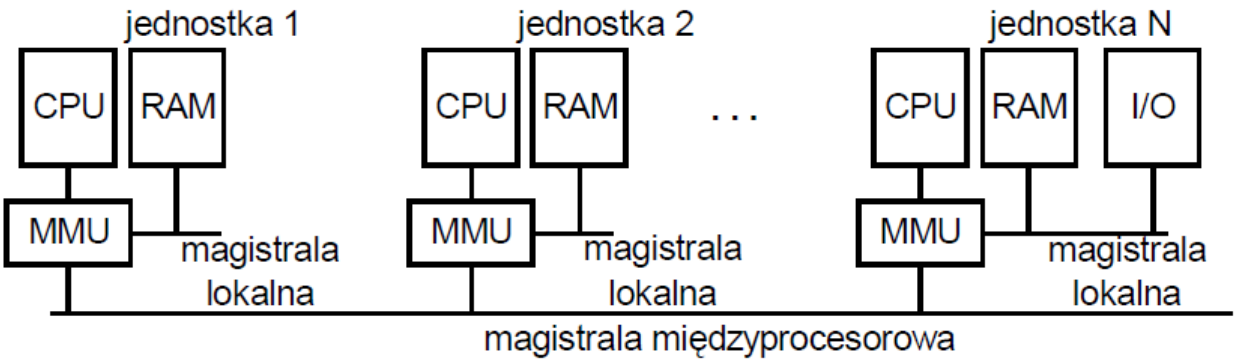


Wieloprocesor z magistralą i pamięciami podręcznymi

Złagodzenie problemu - zastosowanie pamięci podręcznych (ang. *Cache Memory*).
Konsekwencja zastosowania pamięci podręcznych - powstaje problem niespójności pamięci.
W wieloprocesorach zapewnianie spójności pamięci zapewniane jest na poziomie sprzętowym.
Typowe rozwiązanie zrealizowane wg specyfikacji INTEL MP Specification 1.4

Maszyny NUMA

W powyższych konstrukcjach czas dostępu do pamięci dla każdego procesora nie zależy od lokalizacji komórki pamięci. Maszyny tego typu nazywają się maszynami o jednolitym czasie dostępu do pamięci UMA (ang. *Uniform Memory Access*).
W praktyce maszyny UMA nie mają więcej niż kilkadziesiąt procesorów. Gdy potrzeby są większe, z wymagania na jednolity czas dostępu należy zrezygnować, co prowadzi do koncepcji maszyn o niejednolitym czasie dostępu do pamięci NUMA (ang. *Non Uniform Memory Access*).
W maszynach NUMA każdy procesor łączy się z własną pamięć lokalną za pośrednictwem specjalnego układu zarządzania pamięcią zwanego MMU (ang. *Memory Management Unit*).



Architektura maszyny NUMA

Jednostka MMU analizuje wystawiane przez procesor zlecenie dostępu do pamięci. Gdy żądanie dotyczy adresu spoza pamięci lokalnej, kierowane jest ono poprzez magistralę międzyprocesorową do jednostki MMU odległej tej maszyny, w której pamięci lokalnej zawarta jest potrzebna komórka pamięci. Odległa jednostka MMU realizuje żądaną operacją dostępu i przesyła wynik do jednostki lokalnej.
Operacja realizowana jest poprzez sprzęt.
Dostęp do nielokalnej jednostki pamięci będzie trwał dłużej niż do pamięci lokalnej i jest od 10 do 100 razy większy.

Najważniejsze cechy maszyny o architekturze NUMA są następujące:
· Wspólna dla wszystkich procesorów przestrzeń adresowa
· Czas dostępu do komórki pamięci zależny od jej lokalizacji.
· Dostęp do zdalnych komórek pamięci za pomocą instrukcji LOAD i STORE kodu maszynowego.

Maszyny NUMA implementują praktycznie ideę rozproszonej pamięci dzielonej DSM (ang. *Distributed Shared Memory*).

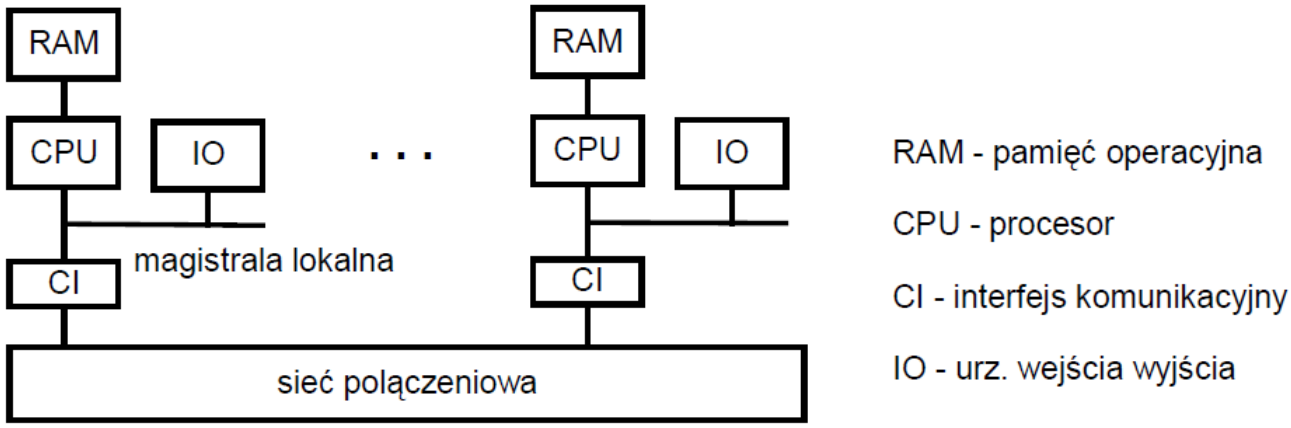
Przykład - CRAY T3E
· Zawiera do 2048 węzłów.
· Każdy węzeł składa się z 64 bitowego procesora DEC ALPHA 375 MHz, pamięci lokalnej 128 MB, jednostki MMU i routera posiadającego 6 połączeń do innych węzłów.
· Sieć połączeń ma postać trójwymiarowego torusa.

Z punktu widzenia programisty najważniejsze cechy maszyny o architekturze wieloprocesora są następujące:
· Możliwość prawdziwie równoległego wykonywania wielu strumieni instrukcji.
· Obecność wspólnej dla wszystkich procesorów przestrzeni adresowej.

Narzędzia programowania:
Model wątków operujących na wspólnym obszarze pamięci. Wątki komunikują się przez wspólną pamięć a wzajemne wykluczanie zapewnione jest przez monitory, muteksy czy semafory.
· Biblioteka wątków Pthread - POSIX
· Specyfikacja OpenMP

Multikomputery

Bariera rozwoju multiprocesorów - koszt realizacji pamięci dzielonej.
Metodą przezwyciężenia tej bariery jest częściowa lub całkowita rezygnacja z pamięci dzielonej co doprowadziło do skonstruowania multikomputerów.
Multikomputer składa się z procesorów posiadających własne pamięci lokalne i komunikujących się ze sobą przez wyspecjalizowane sieci połączeniowe.



Struktura Multikomputera

Komunikacja międzyprocesowa – komunikaty.

· **Send(adres,bufor,ile)** - wysłanie bufora z danymi do procesu docelowego.
· **Receive(adres,bufor,ile)** przekazanie bufora danych procesowi odbierającemu.

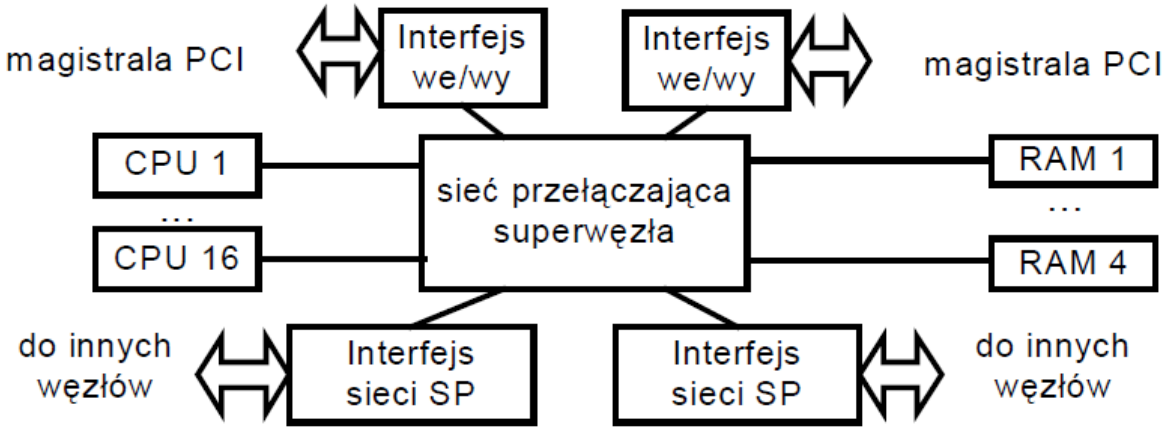
Przesłanie komunikatu pomiędzy procesorami realizowane jest za pomocą interfejsów komunikacyjnych i sieci połączeniowej.

Rodzaje sieci połączeniowej i interfejsów:
· Urządzenia specjalizowane – **komputery masywnie równoległe MPP** (ang. *Massive Paralell Processors*).
· Urządzenia typowe (Gigabit Ethernet, Fast Ethernet) – **Klastry**

W obydwu rodzajach konstrukcji stosuje się podobne procesory i pamięci a różnica tkwi w sieci połączeniowej.

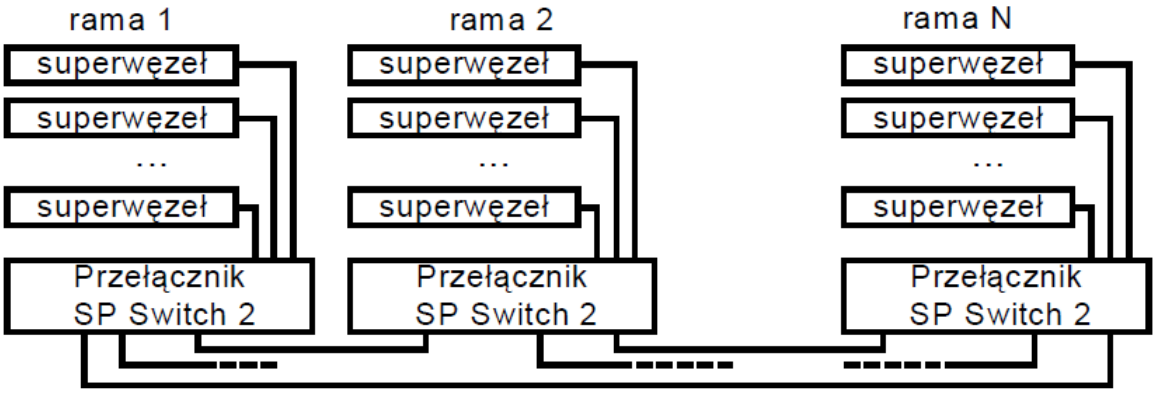
Przykład klastra
„Układ” - zainstalowany we Wrocławskim Centrum Sieciowo Superkomputerowym. Klaster ten zawiera 30 komputerów połączonych siecią Fast Ethernet. Każdy z komputerów zawiera 2 procesory Intel Xeon i 3 GB pamięci RAM.

Przykład komputera MMP
IBM RS/6000 SP
· Komputer składa się z superwęzłów połączonych specjalnymi przełącznikami SP Switch 2.
· W skład superwęzła wchodzi 16 procesorów Power PC, 4 moduły pamięci RAM, 2 interfejsy do przełącznika SP Switch 2 i dwa interfejsy wejścia wyjścia.



Rys. 1-6 Schemat superwęzła komputera IBM RS6000/SP

· Sieć składa się z wielu połączonych przełączników.
· Pojedynczy przełącznik umożliwia połączenie do 16 jednostek.
· Przepustowość przełącznika wynosi 700 MB/sek przy opóźnieniu 17 ms.
· Przełączniki można łączyć w zestawy umożliwiające połączenie do 512 jednostek.
· Sieć przełączników ma topologię drzewa.
· Superwęzły umieszczane są w stojakach w których umieszczane są także przełączniki co tworzy ramę (ang. *Frame*).
· Komputer składa się z ram połączonych siecią .



Rys. 1-7 Struktura superkomputera IBM RS/6000 SP

Struktura superkomputera IBM RS/6000 SP

· Superwęzeł jest mutliprocesorem
· Cały komputer multikomputerem składającym się z multiprocesorów
· Komputer ma strukturę hierarchiczną.

Rozwiązanie jest w znacznym stopniu skalowalne i umożliwia skonfigurowanie komputera z 8192 procesorami.

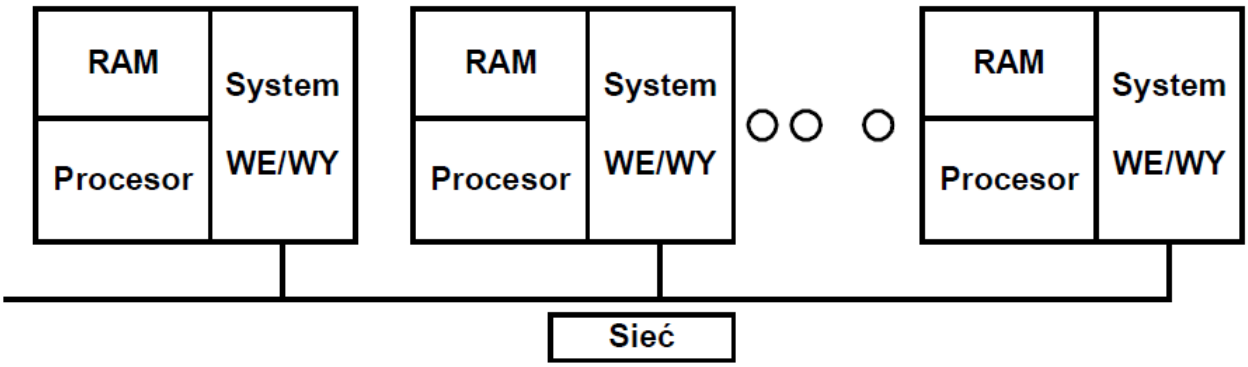
Architektura komputerów ma znaczny wpływ na sposób ich programowania.

Typ	Dostęp do pamięci komp. zdalnego	Skalowalność	Programowanie
Multiprocesor	Instrukcje LOAD i STORE	Trudna	Procesy komunikujące się przez pamięć dzieloną
Multikomputer	Komunikaty	Łatwa	Procesy przesyłające komunikaty

Porównanie multiprocesorów i multikomputerów

Systemy rozproszone

System rozproszony składa się z wielu niezależnych komputerów połączonych za pomocą sieci. Pracujące na nich aplikacje komunikują się poprzez sieć.



Rys. 1-8 Struktura systemu rozproszonego

Do programowania systemów rozproszonych wykorzystywany jest model procesów komunikujących się poprzez wymianę komunikatów (model komunikujących się procesów).

Przykładowe narzędzia programowania:
- Komunikacja niskiego poziomu – gniazdka TCP/IP
- Zdalne wywoływanie procedur – SUN RPC
- Obiekty rozproszone – Java RMI, CORBA
- Systemy operacyjne wspierające systemy rozproszone – QNX6 Neutrino

Maszyna von Neumanna
Architektura von Neumanna jest określona przez zestaw cech. Model maszyny von Neumanna wprowadza specyficzny mechanizm dostępu do pamięci – poprzez adres.
Z takiej organizacji pamięci i z faktu przechowywania w niej programu wynika z kolei obecność rejestru licznika instrukcji

- Instrukcje tworzące program są przechowywane w pamięci w taki sam sposób jak dane
- Pamięć składa się z pewnej liczby ponumerowanych komórek
- dostęp do pamięci następuje poprzez podanie przez procesor numeru komórki
- numer komórki jest nazywany adresem

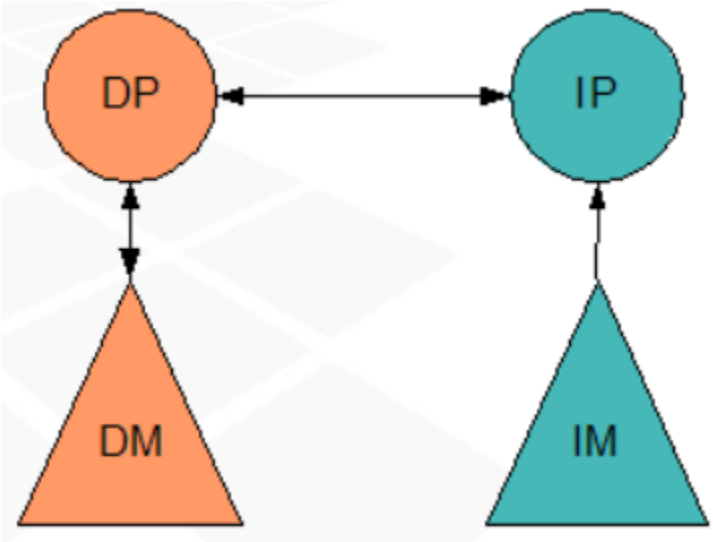
Z powyższych postulatów wynika w praktyce, że:
- zazwyczaj komputer będzie pobierał kolejne instrukcje programu z kolejnych komórek pamięci
- komórki te będą wybierane przez zwiększający się adres, który powinien być przechowywany i inkrementowany w procesorze
- adres jest przechowywany w specjalnym rejestrze PC (ang. Program Counter)

Warianty organizacji hierarchii pamięci w realizacji maszyny von Neumanna

Dwa warianty architektury von Neumanna różnią się sposobem przechowywania instrukcji i danych.
- **Harvard** – oddzielne hierarchie pamięci programu i danych
- **Princeton** – wspólna hierarchia pamięci programu i danych

Architektura Harvard jest niekiedy uważana za architekturę nie spełniającą postulatów von Neumanna wobec faktu oddzielnego przechowywania instrukcji i danych.

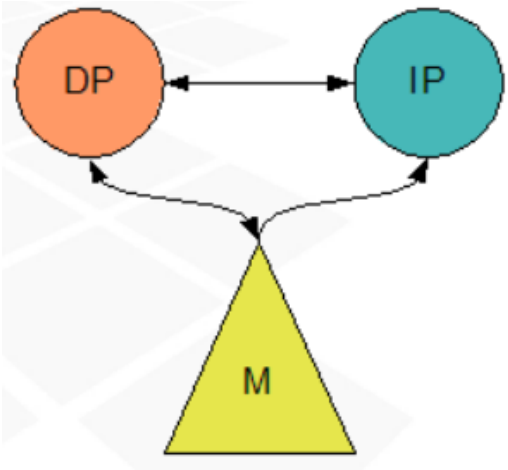
Architektura Harwardzka



Architektura harwardzka - cechy
- Realizacja maszyny von Neumanna z oddzielnymi hierarchiami pamięci programu i danych
- często uznawana za architekturę nie vonneumannowską ze względu na dyskusyjność postulatu o jednakowym składowaniu instrukcji i danych
- Wysoka wydajność dzięki możliwości równoczesnego pobierania instrukcji i operacji na hierarchii pamięci danych

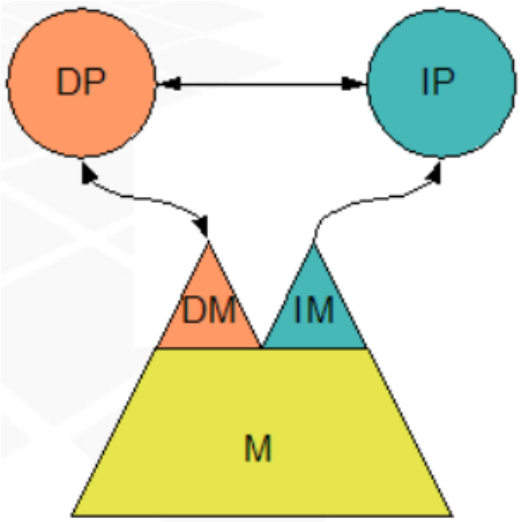
● Brak możliwości zapisu instrukcji do hierarchii pamięci instrukcji
● brak możliwości programowania
● dopuszczalne tylko w zastosowaniach wbudowanych

Architektura Princeton



Architektura Princeton - cechy
● Wzorcowa realizacja maszyny von Neumanna ze wspólna hierarchią pamięci instrukcji i danych
● Wspólna hierarchia wyklucza równoczesne pobieranie instrukcji i danych
● tzw. von Neumann bottleneck
● Nieograniczone możliwości modyfikacji programu
● obiekt zapisany przez procesor danych do hierarchii pamięci jako dane może być następnie pobrany przez procesor instrukcji jako instrukcja
● możliwość programowania – potrzeba w komputerach uniwersalnych
● program może sam sobie modyfikować (automodyfikacja) - cecha nie zawsze pożądana

Architektura Harvard-Princeton



Hierarchia pamięci w architekturze Harvard-Princeton charakteryzuje się częściowym rozdzieleniem hierarchii pamięci. Co najmniej jeden poziom kieszeni jest oddzielny dla hierarchii pamięci instrukcji i danych.
Architektura Harvard-Princeton łączy zalety architektury Harvard (wydajność) i Princeton (programowalność).
Programowalność nie jest tu jednak dokładnie taka sama, jak w architekturze Princeton.

Architektura Harvard-Princeton - cechy

● Realizacja maszyny von Neumanna z oddzielnymi górnymi warstwami hierarchii pamięci i wspólnymi warstwami dolnymi.
● Przynajmniej jeden poziom kieszeni jest oddzielny dla procesorów instrukcji i danych
● Większość odwołań do hierarchii pamięci jest realizowanych w górnych warstwach
● szybkie działanie dzięki równoległości dostępów jak w architekturze Harvard
● Wspólne dolne warstwy hierarchii umożliwiają zapis programu
● programowalność – niezbędna w komputerach uniwersalnych

Harvard-Princeton – modyfikacja programu
● Program użytkowy nie ma pełnej kontroli nad położeniem obiektów w hierarchii pamięci
● brak możliwości automodyfikacji
● Kontrolę taka może mieć system operacyjny
● jeden proces może modyfikować drugi
● możliwość ładowania programu np. z pliku
● Architektura Harvard-Princeton zaspokaja potrzeby programowalności bez narażania bezpieczeństwa
● automodyfikacja jest niebezpieczna
● Większość współczesnych komputerów ma architekturę Harvard-Princeton w tym wszystkie współczesne komputery PC