

Zaawansowane architektury procesorów

CISC, RISC i co dalej?

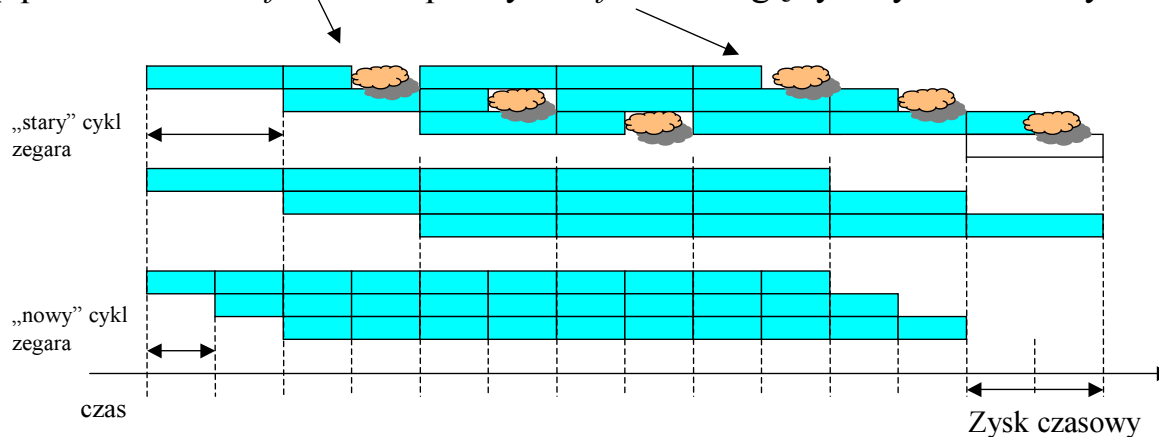
- ▶ Postęp w technologii wytwarzania mikroprocesorów pozwala na upakowanie coraz większej ilości zasobów sprzętowych procesora – odejście od koncepcji „szybkość kosztem uproszczenia listy rozkazów i możliwości”, skoro można mieć jedno i drugie
- ▶ Dodanie do architektury RISC nowych zaawansowanych instrukcji (arytmetyka zmiennoprzecinkowa, multimedia) przy jednoczesnym zwiększaniu efektywności przetwarzania potokowego.
- ▶ Elastyczne podejście do kolejności i jednoczesności wykonywania instrukcji programów.

Co nowego?

- **Superpipelining** –
zwiększenie efektywności przetwarzania potokowego i zwiększenie szybkości zegara bez zmiany technologii
- **Architektura superscalarna** –
możliwość równoległego wykonywania instrukcji
- **Wykonywanie poza kolejnością** (*out of order execution*) –
możliwość wykonywania instrukcji niezgodnie z ich kolejnością w programie.

Superpipelining (przetwarzanie super-potokowe)

Fazy instrukcji różnią się stopniem komplikacji i rzeczywistym czasem wykonania (np. dekodowanie *faza 2* i zapis wyniku *faza 5* mogą być szybsze od innych faz).



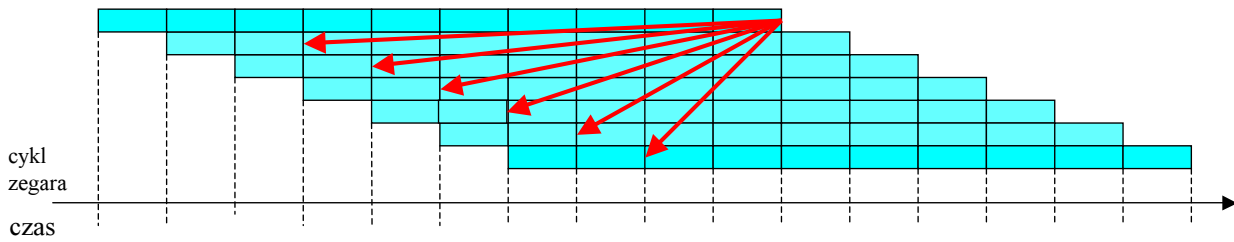
Dzieląc pozostałe fazy na części można wyeliminować bezczynność zasobów komputera podczas mniej skomplikowanych faz.

Podział na więcej faz (pogłębienie kolejki instrukcji) oznacza wzrost szybkości zegara bez konieczności zmian technologii wytwarzania mikroprocesora.

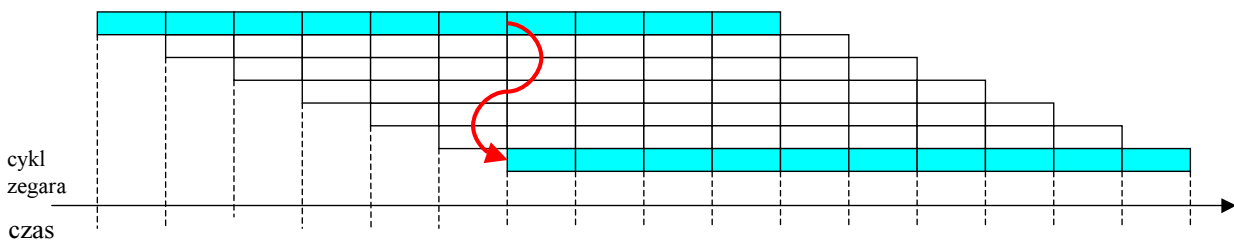
Superpipelining - ograniczenia

Dalsza granulacja nie powoduje wzrostu prędkości, gdyż

- ◆ trudno jest podzielić niektóre fazy na oddzielne operacje,
- ◆ zwiększona liczba faz powoduje wzrost konfliktów danych i zasobów,



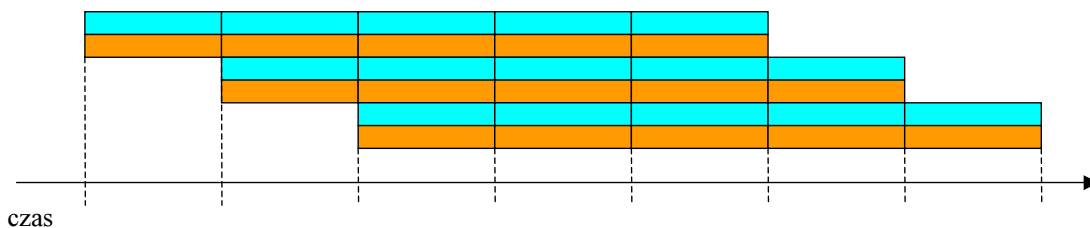
- ◆ opróżnianie kolejki, w wyniku konfliktu sterowania (skoków), wiąże się z unieważnieniem dużej liczby instrukcji



W praktyce głębokość kolejki jest ograniczona do 8-10 faz.

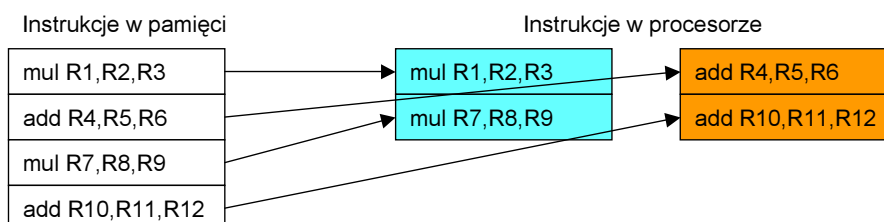
Architektura superskalarana

Podwojenie zasobów procesora dla każdej z faz potoku daje możliwość wykonywania dwóch instrukcji jednocześnie, czyli podwojenie wydajności procesora.

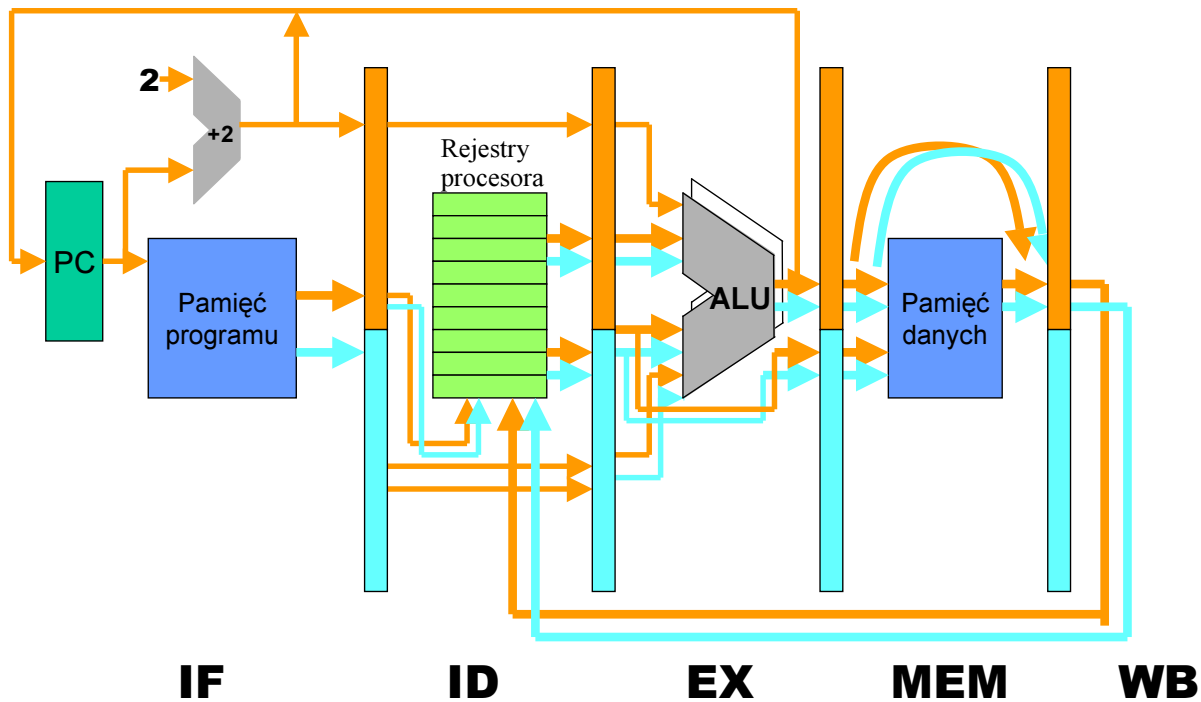


W każdym cyklu zegara procesor wczytuje, dekoduje i wykonuje dwie instrukcje.

Instrukcje muszą być niezależne od siebie, co narzuca konieczność optymalizacji programu pod kątem optymalnego wykorzystania architektury superskalarnej.



Architektura superskalarana



Wszystkie rejestry pośrednie są 2 razy dłuższe, gdyż przechowują dane pośrednie dwóch instrukcji jednocześnie.

Superskalarność - ograniczenia

- W praktyce zwielokrotnianie potoków przetwarzania nie przekracza 6
- Najistotniejszym ograniczeniem klasycznej architektury superskalarnej jest zapewnienie niezależności kolejnych instrukcji w programie.

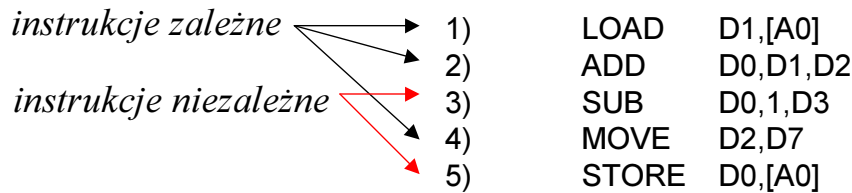
Przy 2 równoległych potokach jest to stosunkowo proste, ale przy 6 jest to możliwe tylko okazjonalnie (procesor wykonuje puste cykle w potoku, dla którego nie można w danym momencie wykonać instrukcji).
- Skoki w programie mogą powodować unieważnienie instrukcji nie tylko z jednej kolejki, ale ze wszystkich kolejek jednocześnie.

Przetwarzanie potokowe z dynamiczną kolejką

(*dynamic pipeline scheduling*)

Tworzenie programów z wbudowaną i stałą niezależnością kolejnych instrukcji jest trudne lub całkiem niemożliwe.

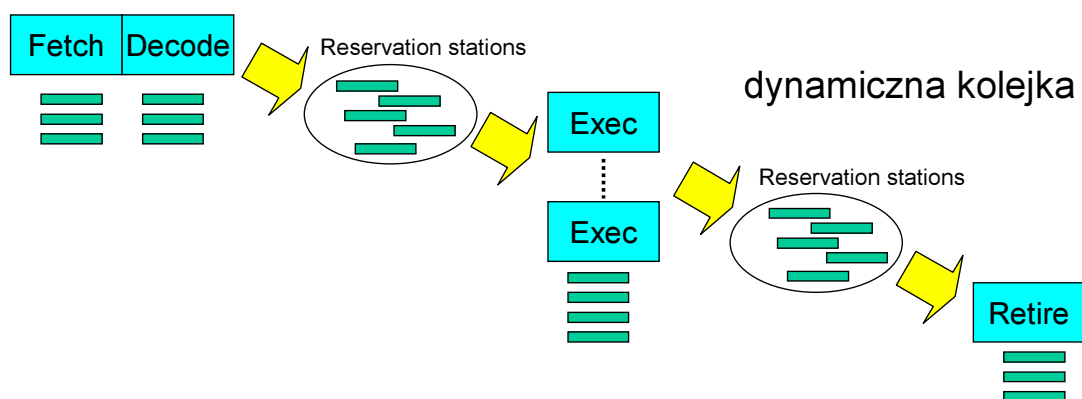
We wszystkie programach istnieje naturalnie znaczna liczba niezależnych instrukcji, ale bez przewidywalnej kolejności.



Dynamiczna kolejka polega na wykonywaniu przez procesor równolegle wielu instrukcji, niekoniecznie kolejnych i kończenie tych, które można, a wstrzymywanie fazy wykonania jedynie dla tych, które wymagają rozwiązania zależności danych.

Idea dynamicznej kolejki

Wielokrotnemu powieleniu ulegają zasoby procesora dla fazy wykonania EX (*procesor jest w stanie szybciej pobierać i dekodować instrukcje niż wykonywać*)



Instrukcje są pobierane i dekodowane w kolejności

Instrukcje i ich dane są zapisywane w rejestrach „postojowych”

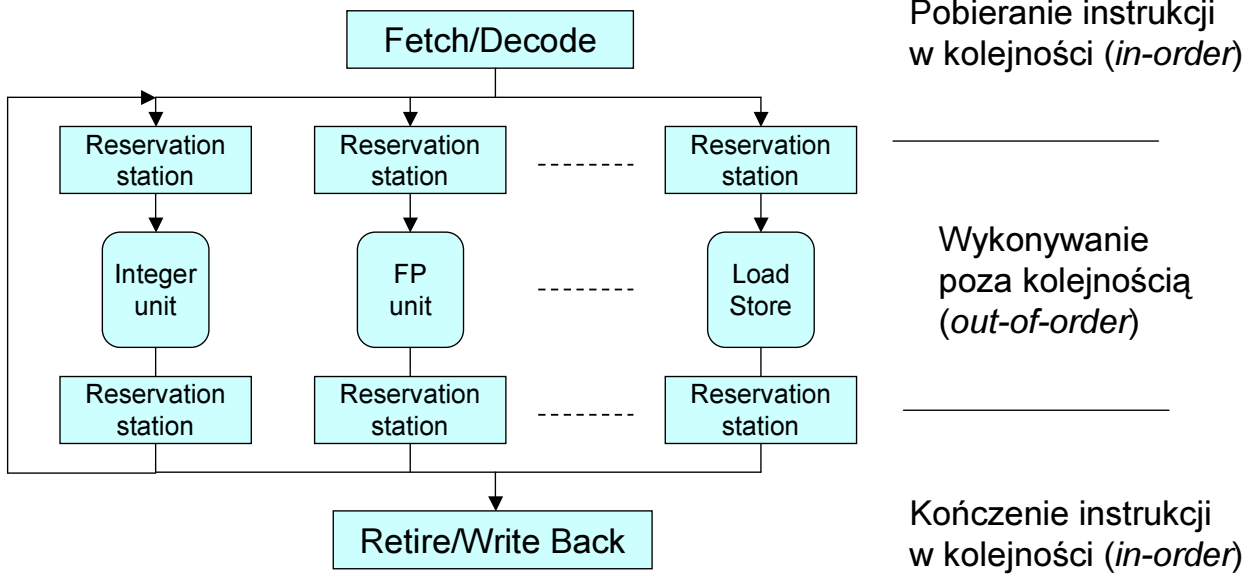
Instrukcje są wykonywanych jednocześnie, bez zachowanie kolejności

Wyniki są zapisywane w rejestrach „postojowych”

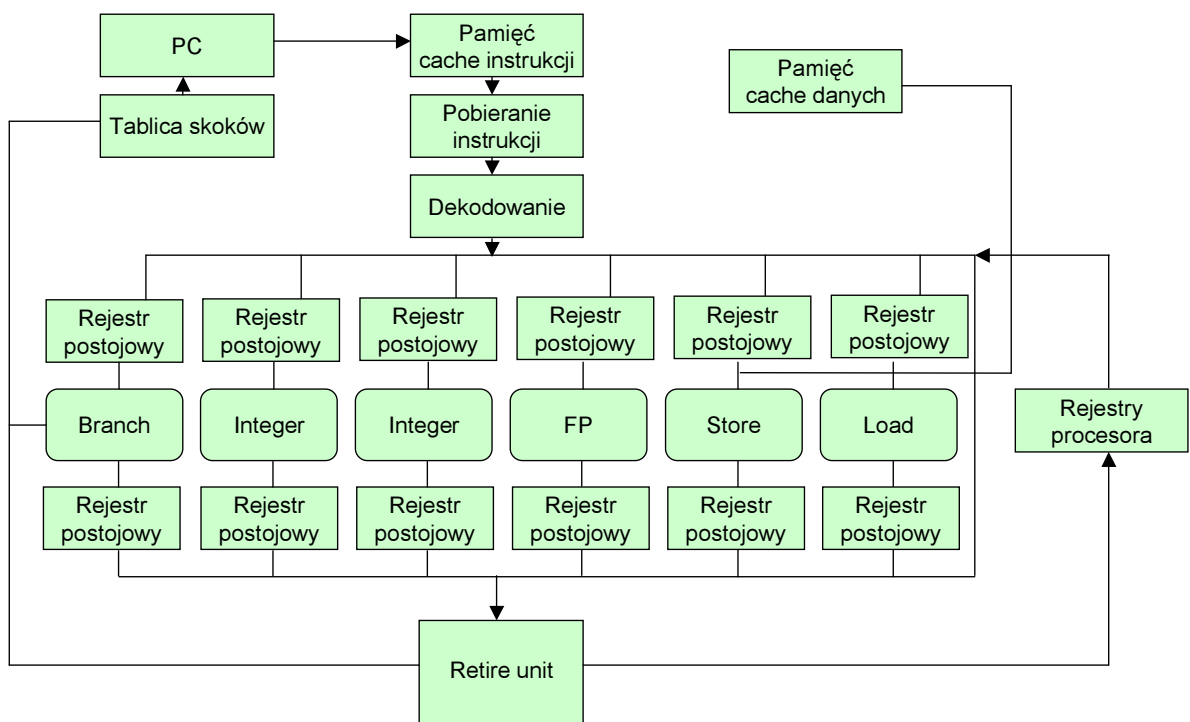
Wyniki są „uznawane” i zapisywane w kolejności zgodnej z programem. Jeśli wykryto konflikt danych, wykonanie instrukcji jest powtarzane.

Wykonywanie instrukcji poza normalną kolejnością nosi nazwę *Out of Order Execution*.

Architektura dla dynamicznej kolejki



Pentium Pro / Power PC



Inne koncepcje

Procesory o długich instrukcjach:

- instrukcja procesora zawiera w sobie dane do kilku niezależnych operacji, z góry przeznaczonych do konkretnej jednostki wykonawczej wewnątrz procesora
- nie jest potrzebne dynamiczne rozwiązywanie kolejności instrukcji w kolejce i upraszcza się konstrukcja procesora
- konieczne jest specjalne tworzenie kodu programów (kompilatory), aby uzyskać niezależność operacji dla jednej długiej instrukcji.

EPIC – *explicitly parallel instruction computing*

VLIW – *very long instruction word*

