

# Klasifikace a vlastnosti paralelních a distribuovaných architektur

Z FITwiki

## Přehled architektur

Paralelní architektury

- **von Neumann**
  - **SISD** (single instruction, single data)
    - Konvenční
    - VLIW (podle slides)
  - **SIMD** (single instruction, multiple data)
    - Vektorové procesory
    - Asociativní procesory
    - Zřetěžené SIMD
    - Multiple SIMD
    - Same Program Multiple Data
  - **MISD**
    - Zřetěžené procesory
    - Rozdělení úlohy
  - **MIMD**
    - VLIW (podle wikipedie)
    - Zřetěžené MIMD
    - Se společnou sběrnici
    - Se propojovací sítí
    - Se pevnou sítí
- **Data Flow** - není von Neumannovská architektura (nemá program a PC), provádí interpretaci grafu toku dat
- **Redukční**

## Obsah

- 1 Přehled architektur
  - 1.1 Flynnova klasifikace
- 2 Paralelní architektury
  - 2.1 SISD
    - 2.1.1 Very Long Instruction Word (VLIW)
  - 2.2 SIMD
    - 2.2.1 Vektorové procesory
    - 2.2.2 Multiple SIMD (MSIMD)
    - 2.2.3 Same Program, Multiple Data (SPMD)
  - 2.3 MISD
    - 2.3.1 Zřetěžené procesory
    - 2.3.2 Rozdělení úlohy
  - 2.4 MIMD
  - 2.5 Data flow
  - 2.6 Redukční počítač
  - 2.7 Externí odkazy

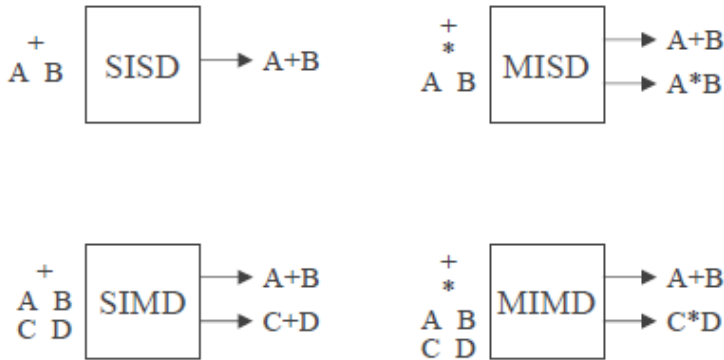
## Flynnova klasifikace

- Flynnova klasifikace popisuje pouze Von-Neumannovské architektury (model architektury počítače využívající společnou paměť pro data i instrukce).

	Single Instruction	Multiple Instruction
Single Data	SISD	MISD
Multiple Data	SIMD	MIMD

- **SISD** - konvenční procesory

- **SIMD** - vektorové procesory
- **MISD** - řetězové procesory
- **MIMD** - více procesorů, který každý pracuje na vlastních datech



## Paralelní architektury

### SISD

#### Very Long Instruction Word (VLIW)

- Podle slides z PRL patří pod SISD podle Wikipedie patří pod MIMD
- Jedna velmi dlouhá instrukce (představuje několik menších instrukcí pro různé procesory)
- Jediný PC, který sekvenčně prochází program
- Tyto procesory mohou pracovat paralelně, pokud je to možné (nejsou kolize, nečeká se na mezivýsledek).

#### Výhody

- jednoduchá implementace
- škálovatelnost

#### Nevýhody

- problémy se skoky
- instrukce v jednom kroku nemohou používat své výsledky navzájem (může vést ke ztrátě paralelismu)
- problémy s operacemi jejichž délka nelze předem vědět (vstup)

#### VLIW a superskalární procesory

- VLIW se nachází někde mezi statickými a dynamickými superskaláry
- **Statické superskalární procesory**

zpracovávají sekvenční program na více procesorech **in-order**, takže paralelnost je jen za souběhu správných instrukcí.

- **Dynamické superskalární procesory**

zpracovávají sekvenční program na více procesorech **out-of-order**, např. spekulativní výpočty za skokem.

# SIMD

## Výhody

- Jednoduchá implementace
- Menší nároky na paměť
- jednodušší synchronizace a rychlejší komunikace mezi procesory než u MIMD

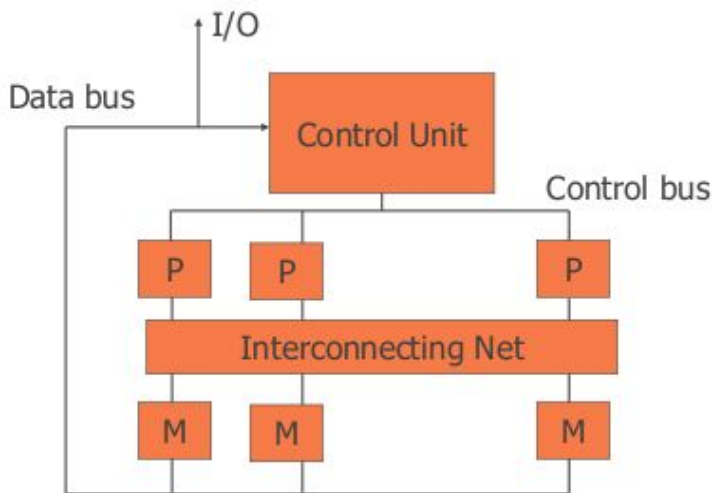
## Nevýhody

- Ne všechny problémy lze datově paralelizovat
- Podmíněné skoky výrazně zpomalují
- Nevhodné pro malý počet procesorů
- vyžadují speciální procesory

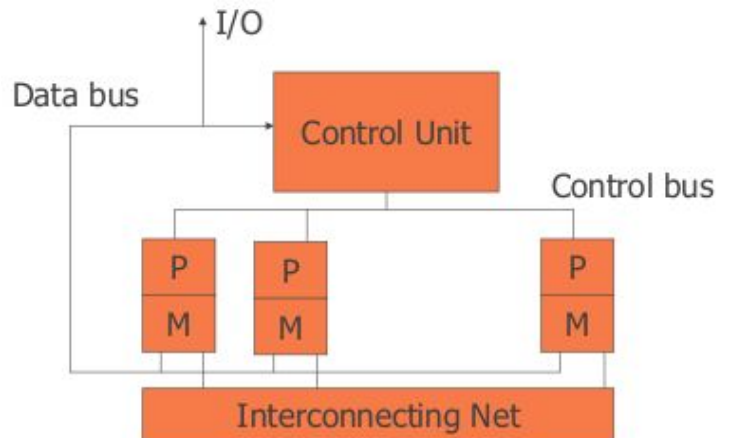
## Vektorové procesory

- paralelně se provádí stejná instrukce na  $n$  procesorech a  $n$  částech vstupních dat
- MMX, SSE

### Se společnou pamětí

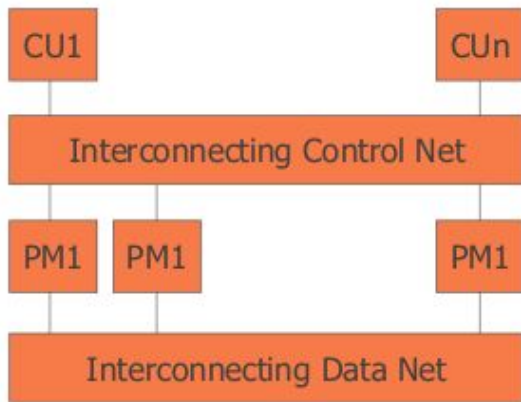


### S oddělenou pamětí



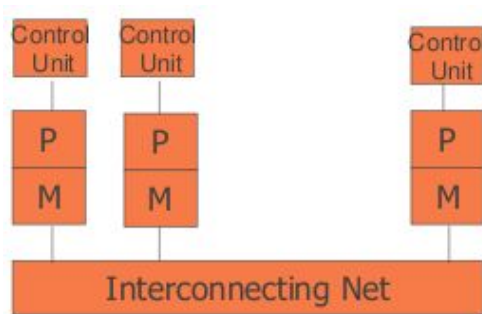
## Multiple SIMD (MSIMD)

- více kontrolních jednotek
- procesory se dynamicky přidělují



## Same Program, Multiple Data (SPMD)

- procesory provádí stejný program, ale zcela nezávisle (žádná synchronizace)



## MISD

### Zřetězené procesory

- pipeline
- lineárně propojené procesory
- řeší proudové úlohy
- data prochází postupně jednotlivými procesory



### Rozdělení úlohy

- data se předají několika procesorům
- každý procesor provede nějakou část výpočtu

Např.: Ověření prvočísła se provede tak, že se číslo předá x procesorům a každý z nich vyzkouší dělení určitou skupinou dělitelů.

## MIMD

- multiprocesory (komunikují sdílenou pamětí)

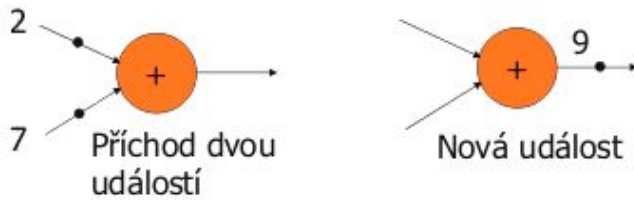
- multicomputery a distribuované systémy (zasílání zpráv)

## Data flow

- nemá program a PC (Program Counter)
- program převeden na graf toku dat, který se pak interpretuje
- Do uzlu vstupují operandy a vystupuje výsledek.
- Paralelismus spočívá v několika operačních jednotkách, které vybírají události a pouští je grafem.
- Potenciální problém: pokud se na jedné hraně objeví dvě události, mohlo by dojít k přehození pořadí

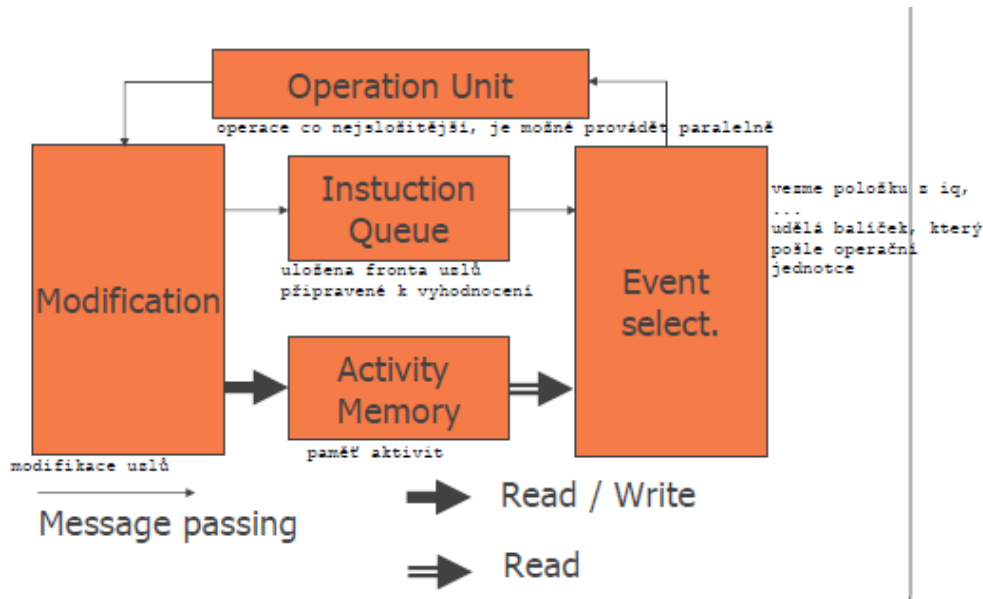
### Graf toku dat

- orientovaný graf
- uzly - operace
- hrany - datové závislosti mezi instrukcemi



### Dataflow processor

- Graf toku dat je uložen v paměti aktivit



## Redukční počítač

- bere výraz a nahrazuje jeho části výsledkem dané operace (např.  $2 \cdot 3$  na 6) - možnost provádět paralelně
- Program se převede na strom a po zadání hodnot se strom redukuje až na jeden uzel - výsledek.

## Externí odkazy

- (en) Flynova klasifikace ([http://en.wikipedia.org/wiki/Flynn's\\_taxonomy](http://en.wikipedia.org/wiki/Flynn's_taxonomy))
- (cs) von Neumannova architektura ([http://cs.wikipedia.org/wiki/Von\\_Neumannova\\_architektura](http://cs.wikipedia.org/wiki/Von_Neumannova_architektura))
- (en) Dataflow architektura ([http://en.wikipedia.org/wiki/Dataflow\\_architecture](http://en.wikipedia.org/wiki/Dataflow_architecture))

Citováno z „[http://wiki.fituska.eu/index.php?](http://wiki.fituska.eu/index.php?title=Klasifikace_a_vlastnosti_paraleln%C3%ADch_a_distribuvan%C3%BDch_architektur&oldid=10034)

[title=Klasifikace\\_a\\_vlastnosti\\_paraleln%C3%ADch\\_a\\_distribuvan%C3%BDch\\_architektur&oldid=10034](http://wiki.fituska.eu/index.php?title=Klasifikace_a_vlastnosti_paraleln%C3%ADch_a_distribuvan%C3%BDch_architektur&oldid=10034)“

Kategorie: Státnice 2011 | Paralelní a distribuované algoritmy

---

- Stránka byla naposledy editována 16. 6. 2012 v 07:57.