

Pokročilé architektury procesorů. Paralelismus na úrovni instrukcí (ILP). Superskalární procesory se statickým a dynamickým plánováním provádění instrukcí. Spekulativní provádění instrukcí a podpora přesného přerušení. Procesory VLIW a EPIC. Využití datového paralelismu, SIMD a vektorové instrukce v ISA.

Pokročilé architektury procesorů

Klasifikace počítačových architektur podle Flynna:

- **SISD** (Single Instruction stream Single Data stream) klasická architektura von Neumann bez zavedení paralelismu
- **SIMD** (Single Instruction stream Multiple Data stream) vícenásobné PE; historicky: CM-5, ILLIAC IV; maticové a vektorové počítače; dnes v podobě rozšíření ISA pro práci na krátkých vektorech (MMX, SSE,...), také v GPU -> tj. lokální podpora
- **MISD** (Multiple Instruction stream Single Data stream) pipelining? Ano/ne? ; Uplatnění ve Fault-tolerant systémech – zpracovávání stejného datového proudu, porovnávání výsledků;
- **MIMD** (Multiple Instruction stream Multiple Data stream) vícevláknové, vícejádrové, víceprocesorové, vícepočítačové systémy (Cluster, Grid, Cloud)
 - SPMD (Single Program stream Multiple Data stream)
 - MPMD (Multiple Program stream Multiple Data stream)

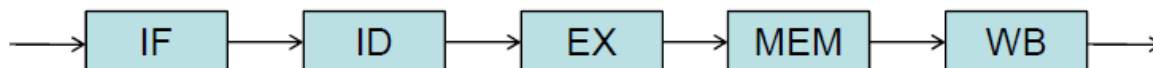
Paralelismus na úrovni instrukcí (ILP = Instruction Level Parallelism)

- Zřetězení (pipelining) – časový paralelismus (sekvenční instr. proud)
- Superskalární vykonávání (v širším smyslu) – prostorový paralelismus

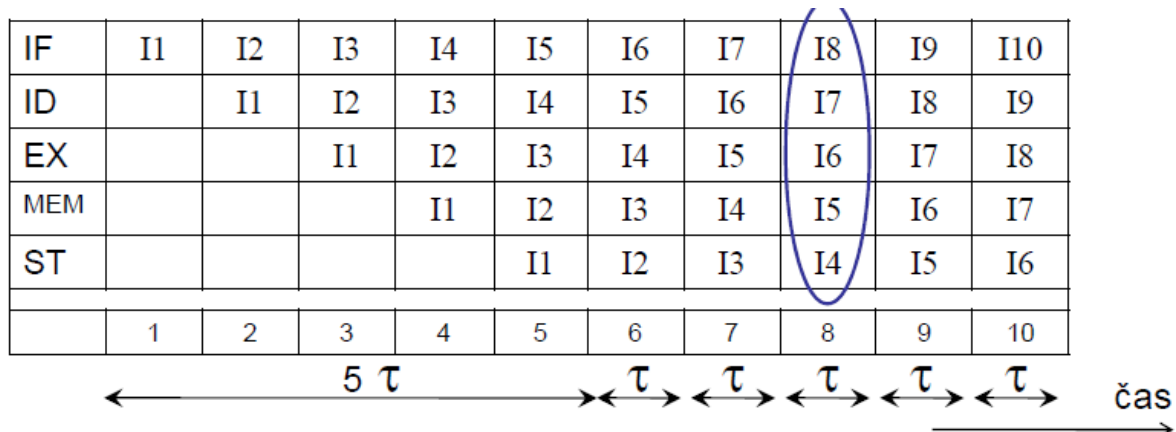
ILP je provádění instrukcí, které se může plně nebo částečně překrývat. ILP je omezeno datovými, jmennými a řídicími závislostmi které jsou v programech neodmyslitelné.

Zřetězení

Předpokládejme, že vykonání instrukce můžeme rozdělit do 5 stupňů.



IF – Instruction Fetch, ID – Instr. decode (and Operand Fetch), EX – Execute, MEM – Memory Access, WB – Write Back



Čas vykonání n instrukcí k -stupňové pipeline:

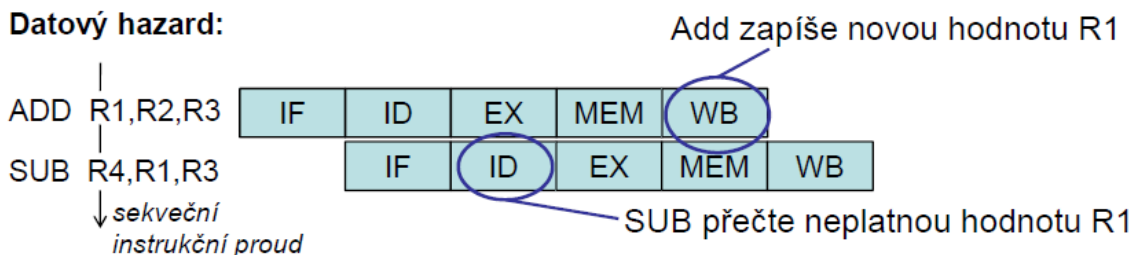
$$T_k = k \cdot \tau + (n - 1) \tau$$

Předpoklad: ideálně vyvážená pipeline

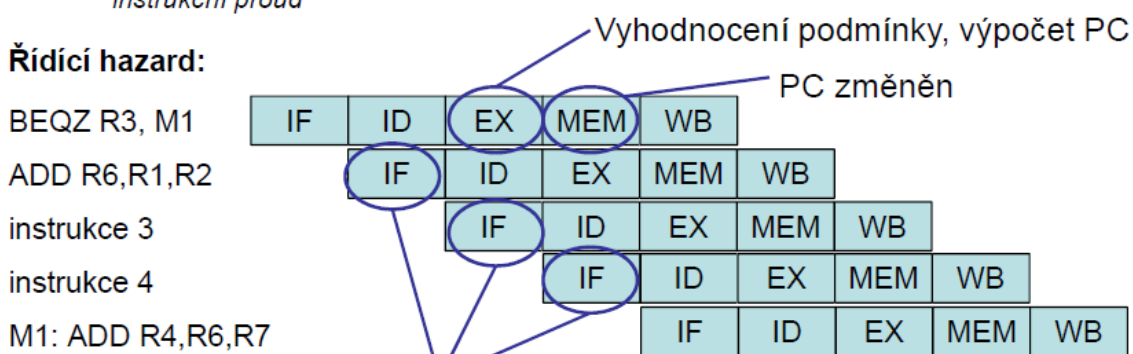
Tau = čas šíření

- Neredukuje čas vykonání individuální instrukce, právě naopak..
- Může způsobovat hazardy:
 - Strukturální (řešeny duplikací)
 - datové (důsledek datových závislostí)
 - řídicí (instrukce měnící PC)

Datový hazard:



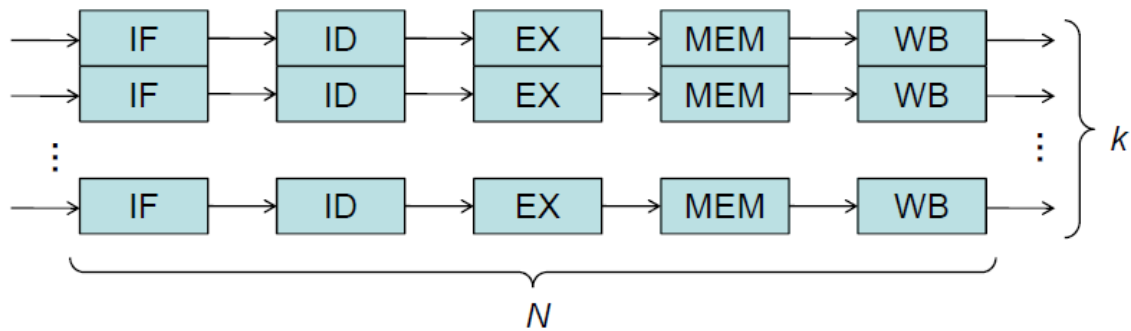
Řídicí hazard:



Měli být přineseny (a následně zpracovávány) tyto instrukce?

- Hazardy způsobují (mohou způsobovat) pozastavení vykonávání (stall) nebo vyprázdnění pipeline
- Hlubší pipeline (více stupňů) znamená méně hradel v každém stupni a tím pádem možnost zvýšit pracovní frekvenci procesoru.., avšak více stupňů znamená i vyšší režii (nutnost lépe řadit instrukce do pipeline)

Zřetězení + superskalárnost



$$S_{k,N} = \frac{T_1}{T_{k,N}} = \frac{nNk\tau}{k\tau + (n-1)\tau} \quad \lim_{n \rightarrow \infty} S_{k,N} = Nk$$

$S(k,N)$ = čas zrychlení

- Sekvenční instrukční proud
- Datové závislosti jsou dynamicky zjišťovány hardwarem

Podporné techniky ILP pro zvýšení stupně paralelizmu

- Předání výsledků uvnitř pipeline (forwarding)
- Vykonávání mimo pořadí (out-of-order execution)
- Přejmenovávání registrů (register renaming)
- Spekulativní vykonávání (speculative execution)
- Předpovídání větvení (branch prediction)
- VLIW (Very Long Instruction Word) a EPIC – MIMD na nejnižší úrovni

Spekulativní provádění

Spekulace je v této souvislosti znamená spuštění kódu, jehož výsledek se „může hodit“ (může se použít). Potíž – skutečnost může dopadnout jinak. Spekulativní provádění instrukcí je forma optimalizace výkonu. Může mít (negativní) vliv na spotřebu procesoru.

Podmínky při spekulativním provádění

- Podm. A – respektování datových závislostí, instrukce čekají na jejich vyřešení
- Podm. B – respektování řídicích závislostí, skok se neprovede, dokud není známa adresa příští instrukce
- Podm. C – Přerušování je přerušováním přesným.

Při spekulativním provádění se částečně nebo dočasně narušují Podm. A a B. Korektnost provádění programu ale splněno být musí, Podm. C trvá.

Moderní procesory s paralelismem na úrovni instrukcí (ILP) užívají řídicí spekulace u podmíněných skoků (dnešní standard) a datové spekulace (Load/Store spekulace). Load se provede dříve, než je známa adresa předchozích Store (např. Itanium, Power 5, Core 2)

Přesné přerušení

- procesor obslouží přerušení v programovém pořadí
 - To nemusí odpovídat pořadí jejich objevení se v proudu (pipeline)
- stav procesoru při jejich obsluze je sekvenčně konzistentní
 - To znamená: instrukce před zdrojem přerušení musí být dokončeny a instrukce za zdrojem přerušení nesmí stav procesoru a paměti změnit.

VLIW - Very Long Instruction Word (typicky několik instrukcí)

- VLIW architektura umožňuje paralelní zpracování (původních několika instrukcí) jednou novou instrukce.
- Paralelně zpracovatelné instrukce jsou naplánovány předem.
- VLIW je vlastně příkladem třídy MIMD.
- Sémantickou jednotkou pro akceptování přerušení zůstává instrukce (velmi dlouhá, či spíše široká)
- Pevný formát instrukce obsahuje kód několika operací, které se ale mohou provést paralelně.

EPIC - Explicitly Parallel Instruction Computing

- Kořeny této architektury vyrůstají z VLIW
- Představitelem této architektury je **Itanium**
- Jsou v ní implementovány dříve popsané metody jako, spekulace, predikce skoků a přejmenování registrů.
- Od architektury x86 (i x86-64) se dramaticky mliší.
- Je založena na explicitní ILP, o paralelizaci se rozhoduje při překladu.
- Paralelně se provádí v jednom hodinovém taktu až 6 instrukcí.
- Nepotřebuje ale speciální HW pro zajištění odstranění hazardů.