Paralelní zpracování na úrovni vláken

Břetislav Bakala

Techniky paralelního zpracování

Common name	Issue structure	Hazard detection	Scheduling	Distinguishing characteristic	Examples
Superscalar (static)	Dynamic	Hardware	Static	In-order execution	Mostly in the embedded space: MIPS and ARM, including the ARM Cortex-A8
Superscalar (dynamic)	Dynamic	Hardware	Dynamic	Some out-of-order execution, but no speculation	None at the present
Superscalar (speculative)	Dynamic	Hardware	Dynamic with speculation	Out-of-order execution with speculation	Intel Core i3, i5, i7; AMD Phenom; IBM Power 7
VLIW/LIW Very long instruction word	Static	Primarily software	Static	All hazards determined and indicated by compiler (often implicitly)	Most examples are in signal processing, such as the TI C6x
EPIC Explicitly parallel instruction computing	Primarily static	Primarily software	Mostly static	All hazards determined and indicated explicitly by the compiler	Itanium

Zpracování na úrovni vláken

Vlákno (thread) – abstrakce „toku programu“

Tradiční OS (UNIX, MS DOS):

- jeden proces obsahuje jedno vlákno = jednovláknový proces (singlethreaded process)

Moderní OS umožňují:

- spuštění více současných úloh v rámci jednoho procesu = vícevláknový proces (multi-threaded process).
- po startu procesu běží jediné tzv. primární vlákno
- primární vlákno může vytvářet vlákna sekundární.

Zpracování na úrovni vláken

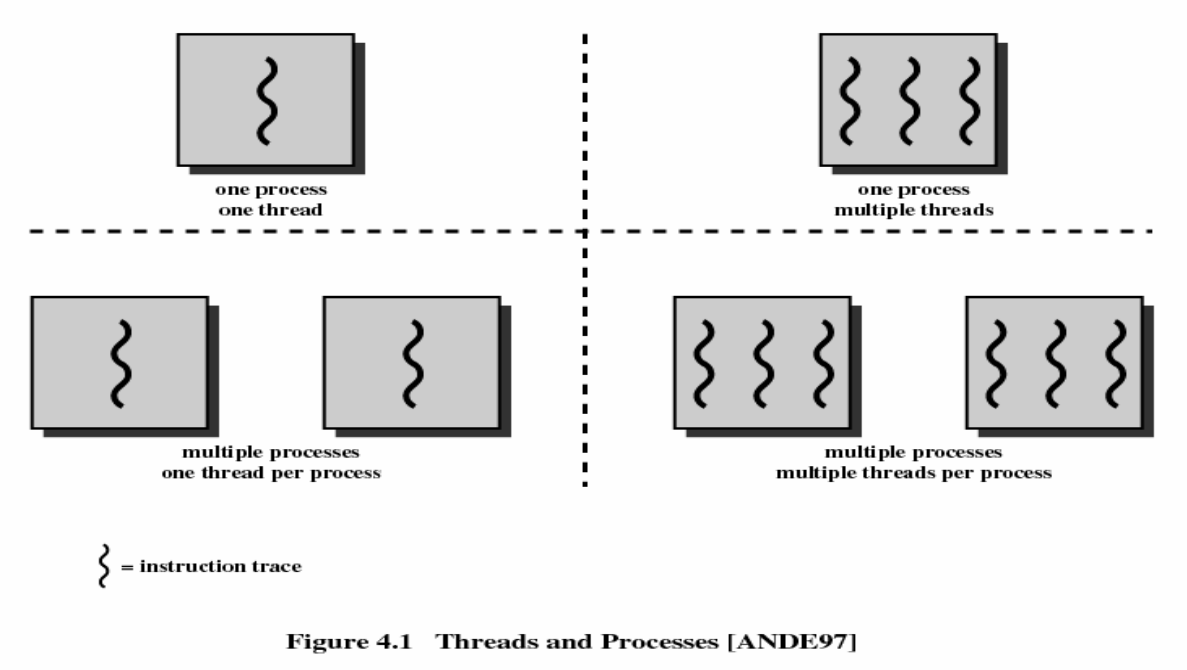


Figure 4.1 Threads and Processes [ANDE97]

Vlastnosti vlákna

Vlákna:

- sdílejí privátní adresový prostor procesu (= mají společný kontext paměti), tzn. mohou přistupovat ke globálním proměnným
- sdílejí systémové prostředky procesu (kontext prostředí)
- každé vlákno má vlastní kontext procesoru a zásobník (včetně lokálních proměnných)

Zpracování na úrovni vláken

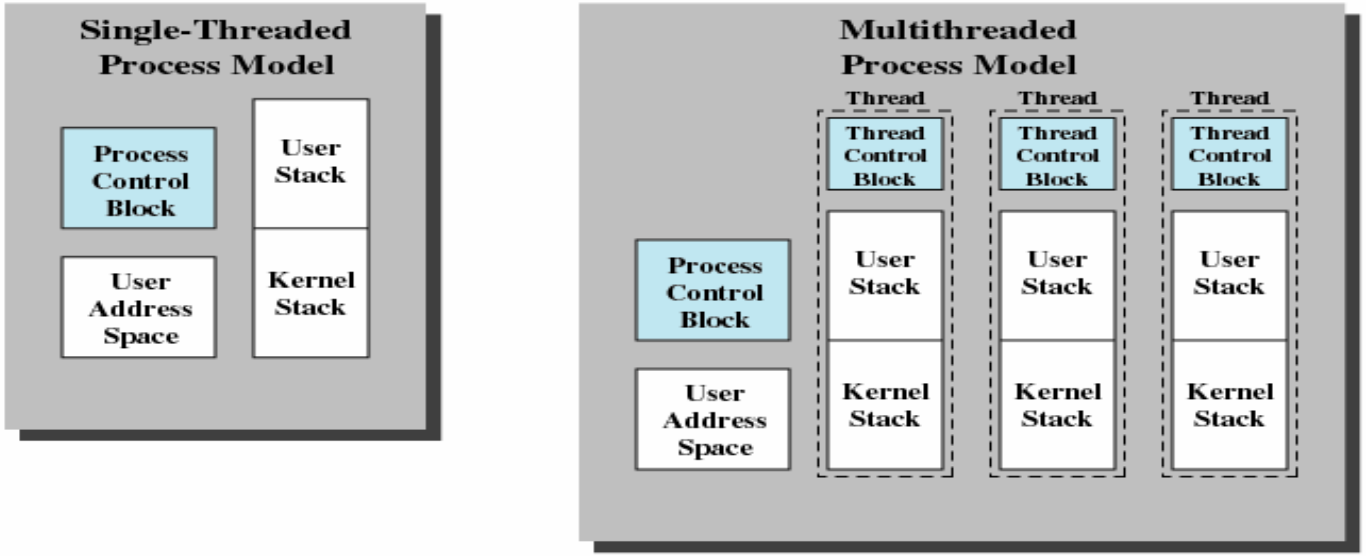


Figure 4.2 Single Threaded and Multithreaded Process Models

Důvody použití vícevláken

- Použití vícevlákna **zvyšuje propustnost jednojádrového procesoru pomocí více podprocesů** pro skrytí zpoždění při zřetězení a v paměti.
- **ILP** může být v některých aplikacích poměrně **omezený nebo obtížný** (využití stejných bloků zřetězení není možné, cache miss, …)
- Když je **procesor zablokován** a čeká na cache miss, **využívání** funkčních jednotek dramaticky **klesá.**
- Vícevlákno umožňuje **více podprocesům sdílet funkční jednotky** jednoho procesoru **překrýváním**.

Výhody vícevláken

- Vícevláknový proces není blokován při zablokování jednoho z vláken
- Paměť a systémové prostředky jsou sdíleny (na rozdíl od procesů), zvyšuje se efektivita
- Přepínání kontextu vláken je výrazně rychlejší než přepínání kontextu u
- procesů
- Efektivní provádění pomalých I/O operací (I/O operace se mohou překrývat s výpočetními operacemi)
- U vícejádrových procesorů běží vlákna skutečně paralelně (NE pseudoparalelně!!!)

Principy vícevláken

- **Obecnější** metodou TLP (Thread-level parallelism) je **vícejádrový** procesor, který má několik nezávislých vláken pracujících naráz a paralelně.
- Vícevlákno **sdílí** většinu jádra procesoru mezi sadou vláken **přepínáním** jednotlivých **stavů** - registry a čítače instrukcí.
- Duplikování stavu vlákna jádra procesoru znamená, že pro **každý podproces** je vytvořen **samostatný soubor registru, samostatný čítač instrukcí a samostatná tabulka stránek**
- paměť může být sdílena prostřednictvím mechanismů **virtuální paměti**
- hardware musí podporovat schopnost změnit relativně rychle jiný podproces; **přepnutí vlákna je mnohem účinnější než přepnutí procesu**, který obvykle vyžaduje stovky až tisíce procesorových cyklů.

Hardwarové přístupy k vícevláknu

- **Jemně strukturované přepínání** mezi vlákny při každém hodinovém cyklu umožňuje **proložení instrukcí**
- Prokládání se často provádí způsobem **round-robin**, přitom **přeskakuje** v té době **zastavené vlákna** – snížení ztrát propustnosti
- Zpomaluje provedení samotných jednotlivých vláken, protože vlákno, které je připraveno provést bez prostojů, bude **zpožděno instrukcemi z jiných vláken**
- **Hrubé přepínání** zjednodušuje přepínání vláken, používá se při **delších blokacích vlákna**, **nezvyšuje propustnost**.

Simultánní multivlákno - SMT

- Implementováno **jemně zrnité vícevlákno na více jader** s dynamickým naplánováním.
- SMT (Simultaneous multithreading) používá paralelismus na úrovni vlákna pro **překrytí** událostí s **dlouhou latencí** v procesoru - zvyšuje využití funkčních jednotek.
- Vyžaduje mnoho **hardwarových mechanismů** potřebných k jeho podpoře.
- Používá velký **soubor virtuálních registrů** a **tabulku přejmenování** na podprocesy se samostatnými **čítači instrukcí.**
- **Dynamické plánování** umožňuje **vykonání několika instrukcí z nezávislých podprocesů** s ohledem na závislost mezi nimi.

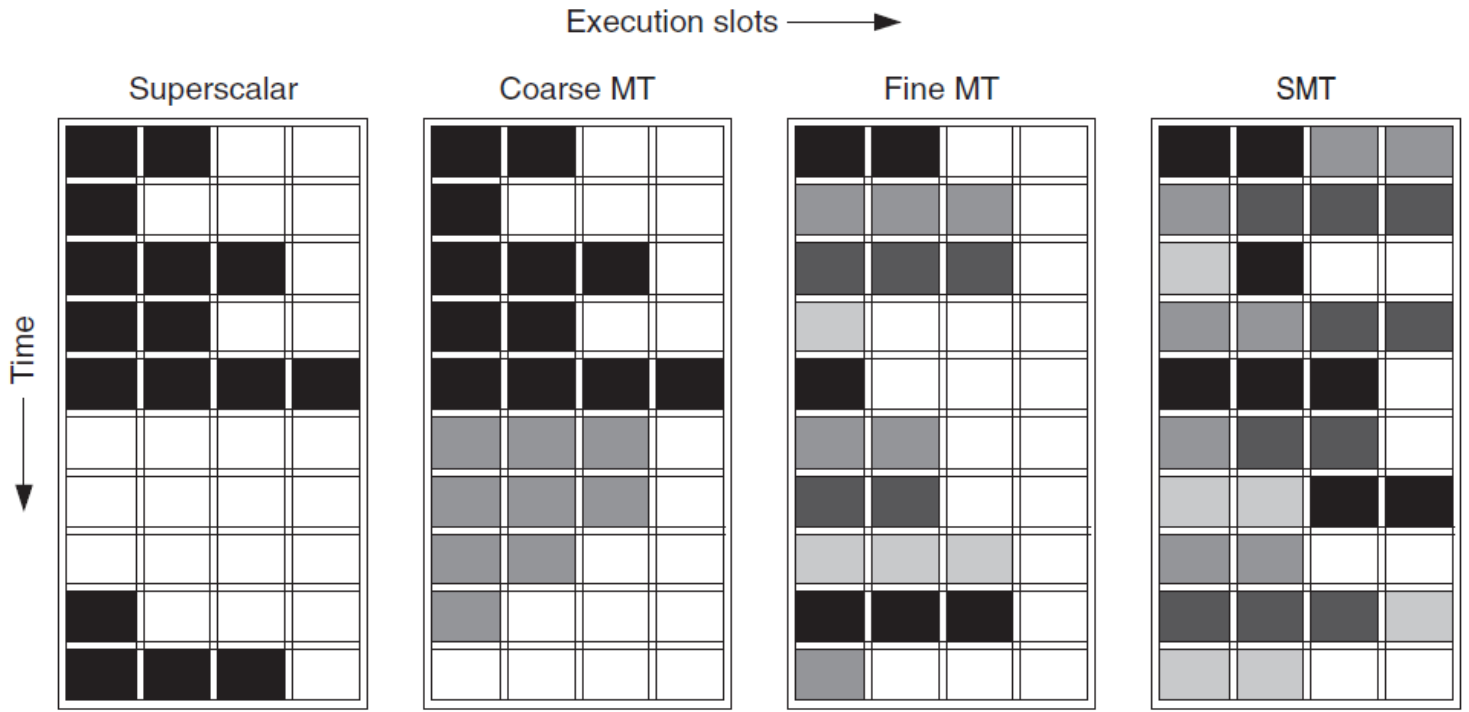
Možnosti využití vláken

Využití zdrojů superskalárního procesoru s použitím vláken:

- **Bez** využití vláken
- Vlákna s **hrubou** strukturou
- Vlákna s **jemnou** strukturou
- Vlákna **SMT** – simultánní multivlákna

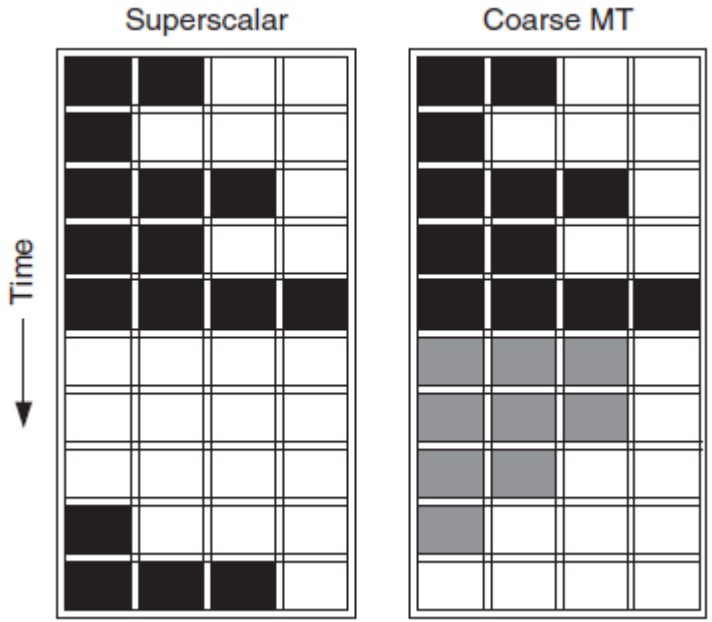
V superskalární architektuře **bez použití vláken** je použití problémových slotů omezeno **vlastnostmi ILP**. Vzhledem k délce **cache miss** L2 a L3 mohou zůstat prostředky procesoru **nečinné**.

Porovnání technik zpracování



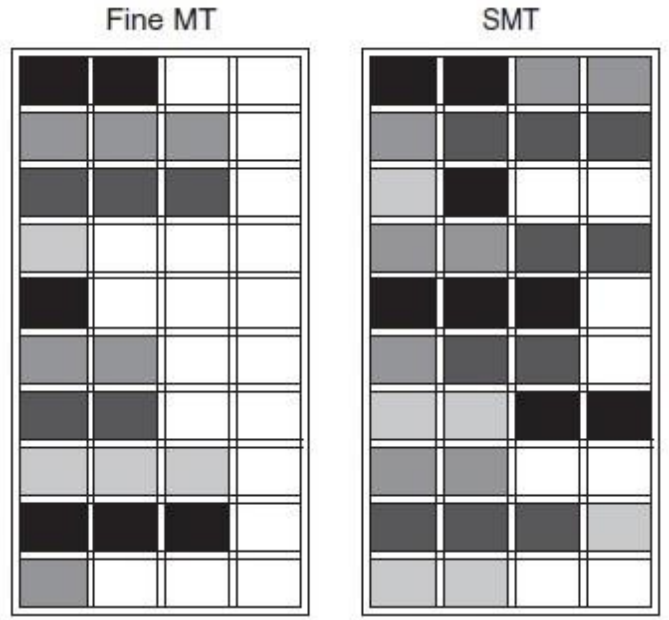
Superskalární, hrubá MT technika

- Dlouhé výpadky u superskalární architektury jsou u techniky hrubých vláken částečně skryty přepnutím na jiný podproces.
- Přepnutí na jiný podproces může způsobit také výpadek díky startovacímu cyklu.



Jemné a simultánní vícevlákno

- Prokládání vláken **v každém hodinovém cyklu** - **vylučuje** celý **prázdný slot**
- **Neprojevuje se delší zpoždění** v podprocesu - přepíná se na jiný podproces každý hodinový cyklus
- Vlákno zpracovává pouze připravené instrukce, cyklus je obsazený/volný – **vhodné u jednojádra**, SMT nejde použít
- **SMT – jemné vícevlákno na vícejádrových procesorech s dynamickým plánováním**



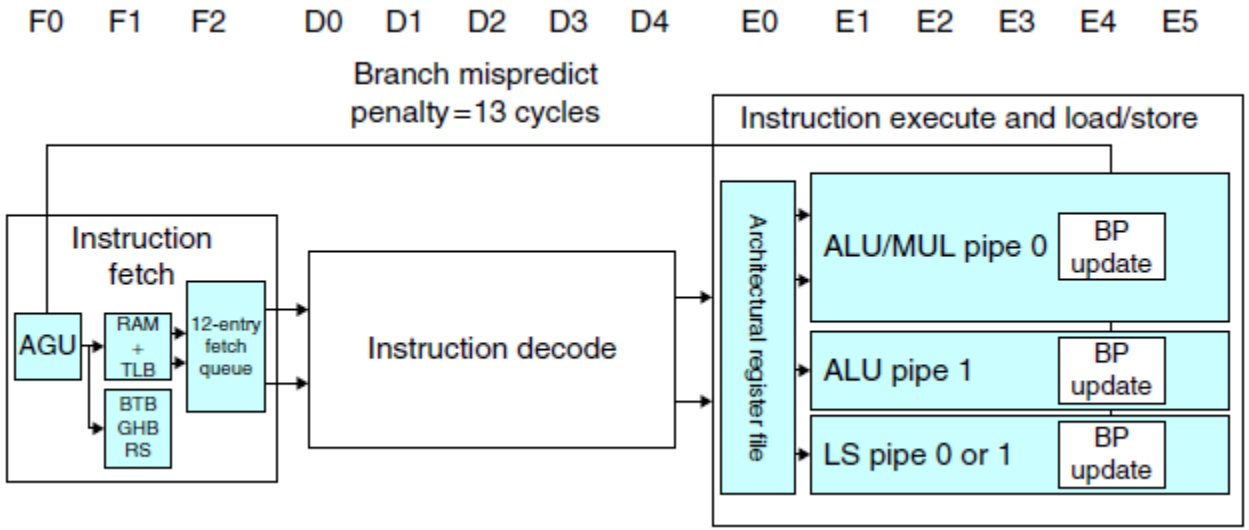
ARM Cortex-A8

Staticky plánovaná superskalární architektura s dynamickým provedením detekce, umožňující provádět až dvě instrukce během jednoho hodinového cyklu

- L1 Data cache = 32 KB. 4-Way, 64 B/line. L1 Instruction cache = 32 KB.
- L2 cache = 256-512 KB. 64 B/line.
- Data TLB size = 32 items, Instruction TLB size = 32 items.
- In-order, dual-issue, superscalar microprocessor core
- 13-stage main integer pipeline
- BTB (Branch Target Buffer): 512 entries
- GHB (Global history Buffer): 4096 * 2bit (accessed via 10-bit Global branch history and 4 low bits of ProgramCounter).
- Return Stack: 8 entries
- 2 x ALU, MUL, LS

Příklad zřetězené struktury ARM

- 3 cykly načtení
- 4 cykly dekódování
- 5 stupňů zřetězení celočíselného zpracování
- 13 cyklů případné misspredict penalty
- Jednotka načítání instrukce udržuje 12 položek fronty instrukcí

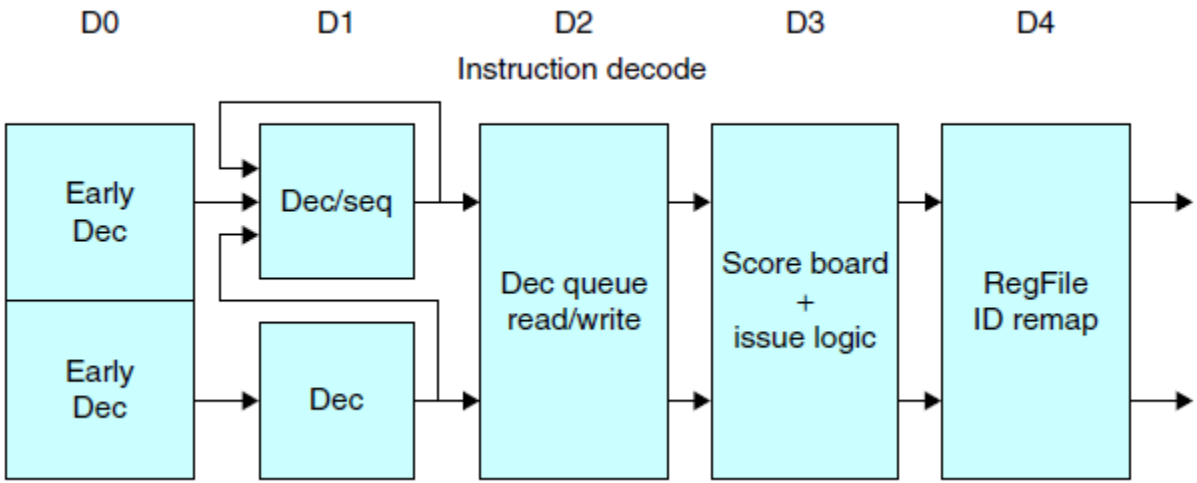


ARM Cortex-A8

ARM Cortex-A8

- Jednotka načítání udržuje spekulativní a ne-spekulativní návratový zásobník. Spekulativní zásobník se čte a aktualizuje okamžitě na základě informací vrácených z přístupu BTB, zatímco nespekulativníí zásobník není aktualizován, dokud není známo, že větev není na konci zřetězení spekulativní. Pokud je větev chybně předvídaná, pak je stav spekulativního zásobníku přepsán stavem v nespekulantním zásobníku.

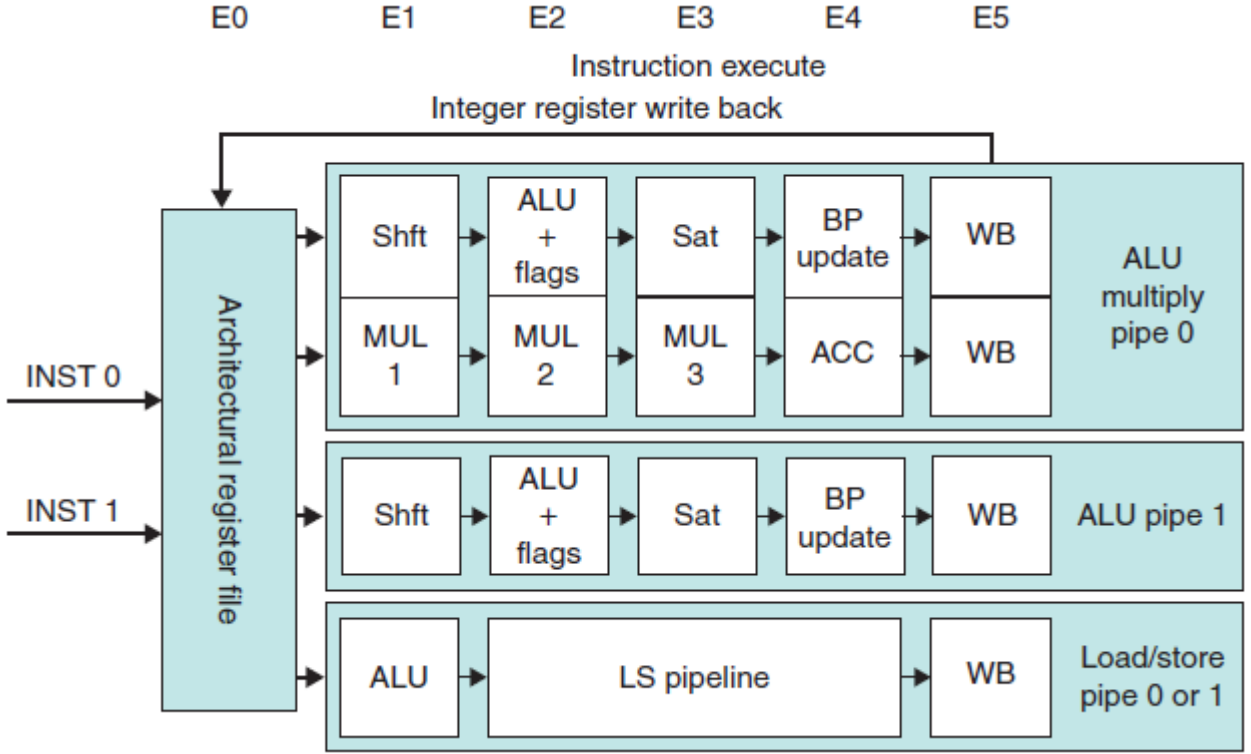
Zřetězení dekódování instrukce



- Až dvě instrukce na takt mohou být zpracovávány použitím in-order mechanismu.
- Používá se tabulková struktura ke sledování kdy lze provádět instrukci
- Dvojice závislých instrukcí jsou serializovány na výsledkové tabulce, dokud průběh cesty neřeší jejich závislost

Zřetězení provedení instrukce

- Pouze jedna instrukce může procházet zřetězením Load/Store
- využívá jednoduchý dvoustupňový staticky plánovaný superskalar, který umožňuje přiměřeně vysokou frekvenci s malou spotřebou.



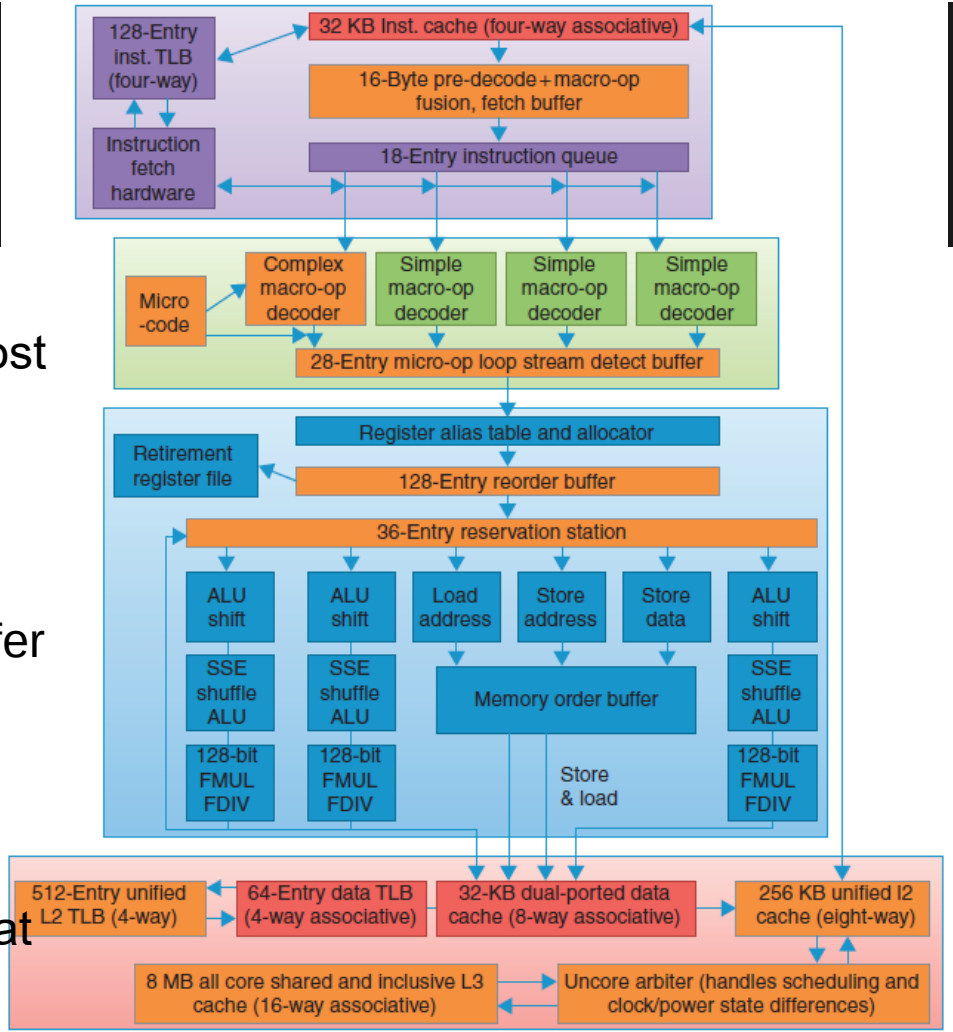
Výkonová rizika zřetězení v A8

- Ideální hodnota CPI má z důvodu dvojité struktury zřetězení hodnotu 0,5.
- Prostoje ve zřetězení mohou mít tyto důvody:

1. Funkční hazard – dvě sousední instrukce nárokují současně stejný blok zřetězení, role překladače, pokud k tomu dojde, provede se jenom jedna instrukce v cyklu.
2. Datový hazard – mohou zastavit obě instrukce – první čeká, druhá je zablokována.
3. Kontrolní hazard – špatně odhadnuté větvení.

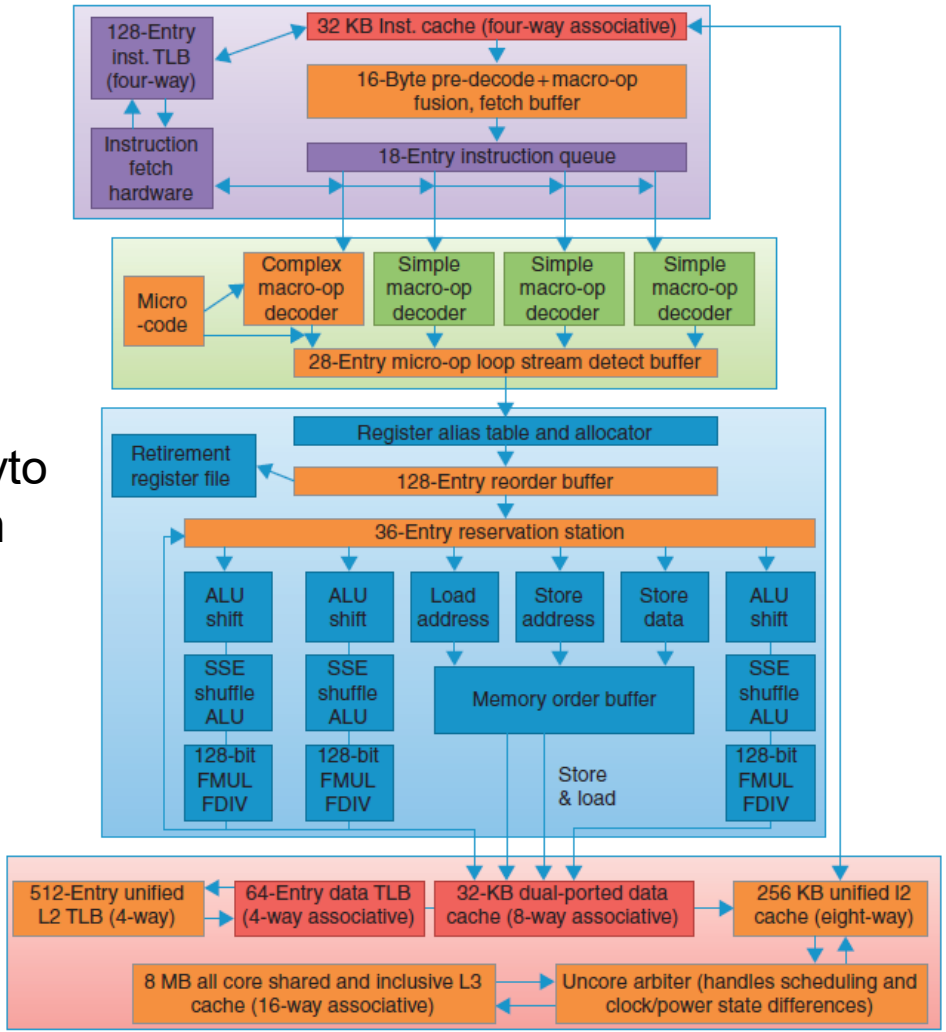
Intel Core i7

- Instruction fetch – víceúrovňový BTB -rovnováha rychlost/přesnost
- Macro-op fusion – porovnání a větvení v jedné operaci
- Predecoder - instrukce 1 – 17 B
- Mikroinstrukce lze zřetězit
- 28 položkový micro-op loop buffer nalezne smyčku a použije pro vykonání
- Microfusion kombinuje dvojice instrukcí LS/ALU do rezervační stanice – pořád možno zpracovat nezávisle



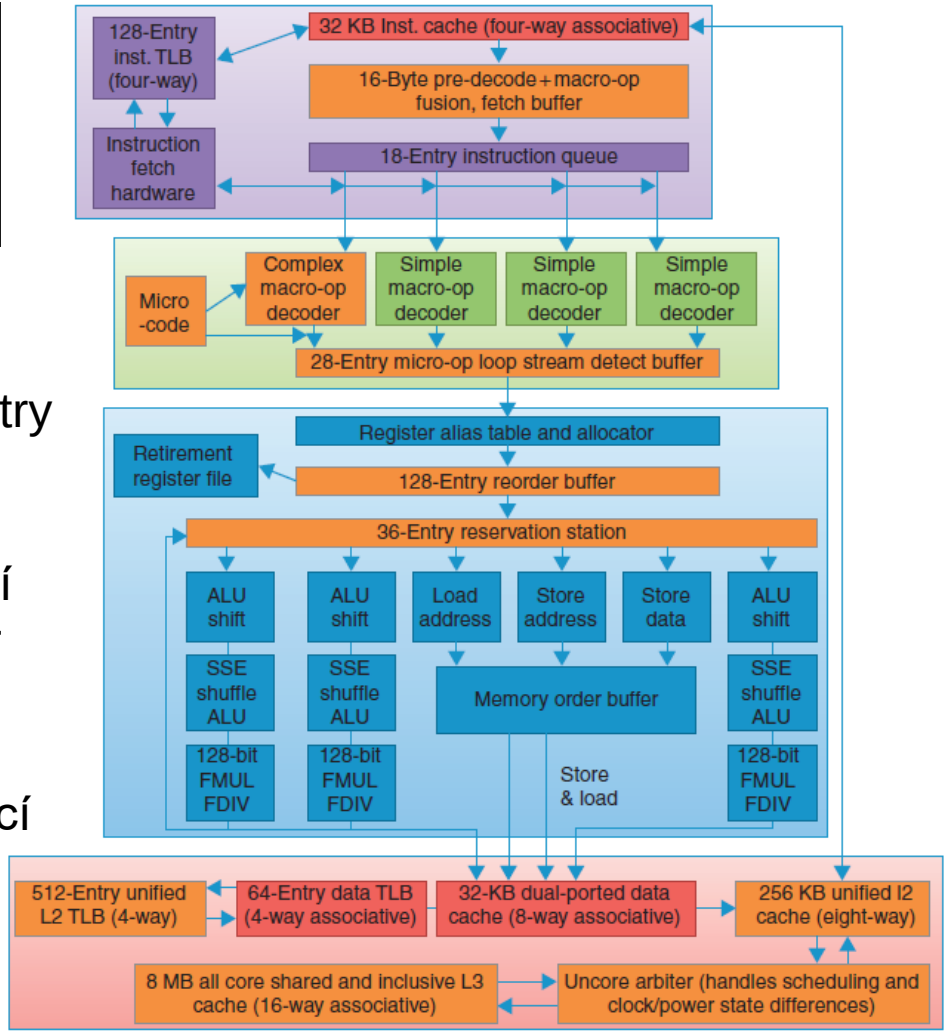
Intel Core i7

- Vyhledá umístění registru v tabulkách registrů, přejmenuje registr, seřadí záznamy v 128 Entry recorder buffer a odešle tyto hodnoty do 36-Entry reservation station.



Intel Core i7

- Mikrooperace jsou prováděny funkčními bloky (až 6/clock), výsledky uloženy zpět do 36-Entry reservation station a do register retirement unit – aktualizují stav registru, pokud instrukce již není spekulativní a položka v reorder buffer je označena jako úplná
- Po označení úplnosti provedení instrukce jsou provedeny čekající zápisy do retirement registru a instrukce je vyjmuta z reorder buffer



Výkonnost Intel Core i7

- Mikrooperace jsou prováděny funkčními bloky (až 6/clock), výsledky uloženy zpět do 36-Entry reservation station a do register retirement unit – aktualizují stav registru, pokud instrukce již není spekulativní a položka v reorder buffer je označena jako úplná
- Po označení úplnosti provedení instrukce jsou provedeny čekající zápisy do retirement registru a instrukce je vyjmuta z reorder buffer