

Procesory s vláknovým paralelismem, Příklady x86 procesorů

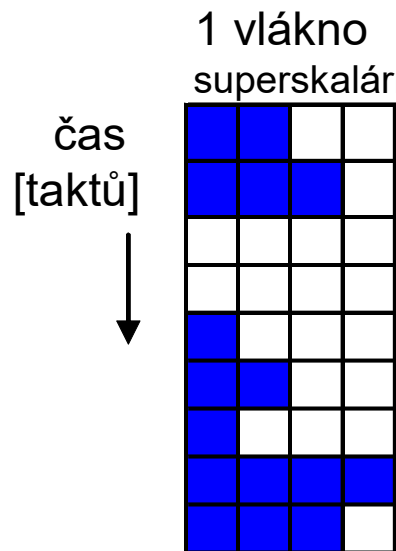
AVS – Architektury výpočetních systémů
Týden 6, 2024/2025

Jirka Jaroš

Vysoké učení technické v Brně, Fakulta informačních technologií
Božetěchova 1/2, 612 66 Brno - Královo Pole
jarosjir@fit.vutbr.cz



- Použitím řady technik v mikroarchitektuře bylo dosaženo dramatické zvýšení výkonnosti CPU při zpracování jednoho kódu (vlákna).
 - Vyšší hod. kmitočet (počet stupňů linky),
 - zvýšení IPC (šířka linky),
 - superskalární zpracování,
 - odstranění konfliktů mezi instrukcemi – OOO,
 - cache L1 – L3 na čipu,
 - prediktory skoků,
 - až 6 instrukcí dokončuje v 1 taktu.
- Pokusy pokračovat v rozvoji těchto technik nevedou již k prokazatelnému zvýšení výkonnosti. Vadí při tom
 - paměťové latence,
 - nízké využití funkčních jednotek (~ 30 %).
- **Řešení:** technologie multithreading a multi-core.



Průměrné **IPC** = **16** / 9 = 1,78

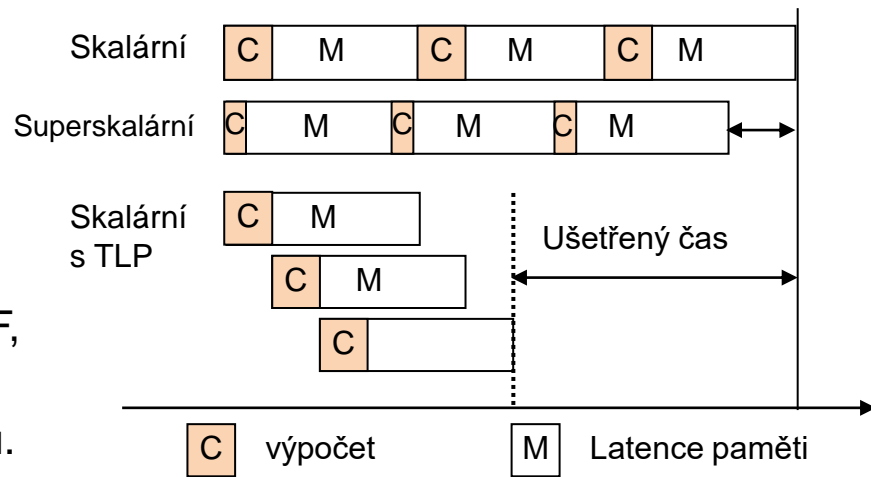
Využití = průměrné IPC / max. IPC
= 1,78 / 4 = 44,5 %.

Běžné **4 cestné** superskalární CPU mají průměrné udržitelné IPC v rozmezí **1,5–2**, tedy využití < 50 %.

Důvody nízkého využití:

- špatně predikované skoky (zbytečné výpočty)
- čekání na data kvůli výpadkům, zvláště v L3 cache
- čekání na data kvůli instrukčním závislostem
- čekání na volnou funkční jednotku

- Pracovní zátěž serverů (**transakce**) je charakterizována vysokým TLP a nízkým ILP (**databáze, web servery**).
- Možnosti zvýšit výkonnost jednoho vlákna jsou omezené (vlákna dlouho **čekají na vyřízení výpadků mimo čip**)
- Proto je snaha spojit **vysoký TLP aplikací** s **podporou pro více vláken na čipu procesoru**.
- **Vlákno = proud instrukcí**. Kontext: IP + SP, RF, PSW včetně myid a priority.
- Vlákna pracují ve sdíleném adresovém prostoru.
- Vlákna jsou (oproti procesům) „lehká“: rychlejší nebo okamžité přepnutí kontextu, málo nebo žádné kopírování.



1. Časový multithreading (TMT, temporal MT):

- Vlákna **se střídají** na jednom jádru **kvůli** jeho **lepšímu** využití, **vyššímu IPC**.
- Lze skrýt výpadky.
- Procesory TLP = procesory s podporou TMT v jádrech.

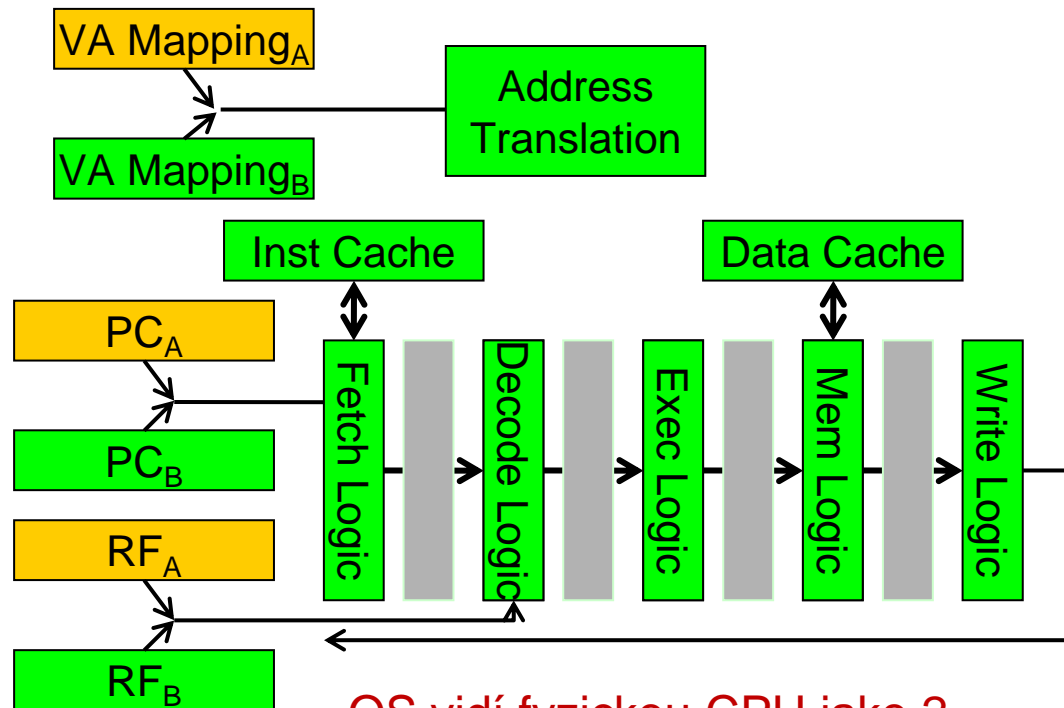
2. Prostorový MT:

- Vlákna **běží paralelně** na více-jádrovém procesoru, 1 vlákno na 1 jádru, **kvůli zvýšení výkonnosti** vzhledem k 1 jádru.

3. Kombinace 1 a 2:

- Čipový MultiProcessing / Multithreading, **CMP/MT** tj. víc vláken na každém jádru.

V každém případě je třeba zajistit co nejrychlejší přepínání vláken. Proto musí mít každé vlákno **svůj HW kontext**.



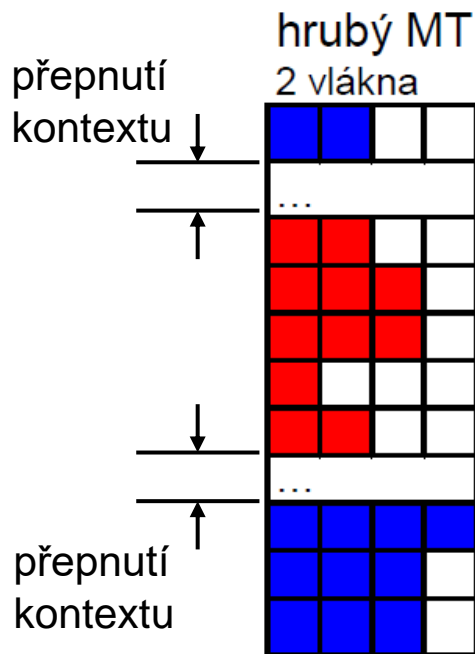
OS vidí fyzickou CPU jako 2
logické CPU!

- Prostředky **sdílené**:
 - CPU
 - všechny cache
 - prediktor skoků
 - funkční jednotky
- Prostředky **replikované**:
 - sady registrů RF
 - čítače programů PC
 - ukazovátka zásobníku
 - TLB
 - RSB, return stack buffer
 - různé řídicí registry
 - řadič přerušení APIC
- Prostředky **nové**: HW pro výběr vlákna

- Velký soubor virtuálních registrů je již k dispozici, stačí
- **Oddělené tabulky RAT** přejmenování registrů pro vlákna
 - Stejný registr každého vlákna je mapován do jiného fyzického registru mapovací tabulkou RAT
- Schopnost dokončit instrukce z několika vláken
 - **Oddělený ROB** pro každé vlákno.
 - Levnější řešení: jen 1 ROB a v 1 taktu dokončují instrukce jen z 1 vlákna
- Je třeba zajistit stejnou **výkonnost s TLP HW jako bez něj i když běží jen 1 vlákno**
- **TMT** se jeví nejslibnější pro maximalizaci výkonnosti procesorů ILP, VLIW a vícejádrových.
- Závisí na aplikaci, IPC může vzrůst až o 30 %.

- OS pokládá jednu fyzickou CPU za několik standardních oddělených logických CPU (LCPU).
- Vše co je potřeba pro využití MT je **podpora symetrického multiprocessingu (SMP) v OS**.
- **Ale pozor!** Nemá-li plánovač v OS povědomí o MT, mohl by špatně rozložit zátěž, např. naplánovat 2 vlákna na 1 jádro (více se zahřívá) a druhé jádro nechat bez práce.
- Plánovač proto musí rozlišovat fyzické CPU a Logické CPU. Všechny moderní OS poběží s aktivovanou technologií MT (Intel HT) a „umí“ detekovat a řešit uvedenou situaci. (Microsoft Windows 7 a vyšší).

Jedno vlákno běží řadu taktů, **k přepnutí kontextu** dochází pouze při události, která **zablokuje linku** na mnoho taktů (např. výpadek v I/D-cache, timeout, špatná predikce).



Co s rozpracovanými instrukcemi v lince:

- **zahodit** (*flush*) a opakovat při další aktivaci, vyžaduje krátkou linku
- **dokončit** paralelně s instrukcemi nového vlákna (délka linky nehraje roli)
- **dokončit**, teprve pak nové.

- **Předpoklad (konzervativní):**
 - kmitočet CPU 1 GHz, výpadek v L1 I-cache 20 ns
 - bez výpadků 1 instr/takt, výpadek 1x za 100 instrukcí
- **Bez MT** je doba vykonání 100 instrukcí + 1 výpadku
 $100 + 20 \text{ taktů} \rightarrow \text{IPC} = 100 / 120 = 0,83.$
- **S hrubým MT (přepnutí kontextu 3 takty)** je doba vykonání
100 instrukcí + 2 přepnutí = $100 + 6 \text{ taktů}$ (pokud jde kód
rozdělit na 2 vlákna)
 $\text{IPC} = 100 / 106 = 0,94$ (zlepšení o $0,94 / 0,83 = 1,14$
tj. o 14%)

- **Kdy dojde k přepnutí vlákna:**

- kromě výpadku v aktuálním vlákně musí existovat jiné připravené vlákno **nebo**
- je připraveno vlákno s vyšší prioritou než je priorita aktuálního vlákna **nebo**
- existuje vlákno, které posledních n cyklů neprovedlo žádnou instrukci (eliminace hladovění).

- **Vlastnosti hrubého MT:**

- vhodný jen pro dlouhá blokování (např. výpadky LLC)
- důležitá je co nejmenší doba přepnutí kontextu
- spravedlivost u vláken s různou četností výpadků: vlákno s malým počtem výpadků je třeba **omezit časovým kvantem**.

- **Poznámka:** připravené vlákno čekající ve smyčce (např. na kritickou sekci) nemá smysl přepínat na běh. Lze je odstavit spec. instrukcemi (viz dále) nebo mu snížit prioritu.

- **1 vlákno:**

R – průměrný počet taktů při zpracování instrukcí mezi paměťovými operacemi

L – latence (paměťové) operace [taktů]

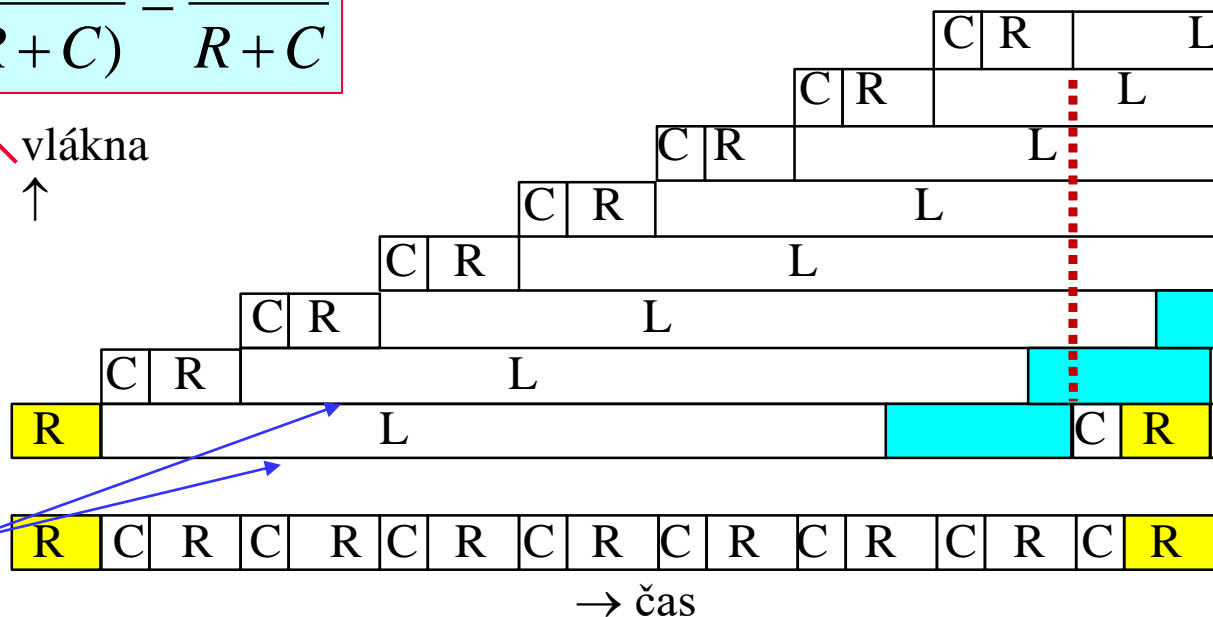
$$\text{účinnost } E_1 = \frac{R}{R + L} = \frac{1}{1 + L / R}$$

- **N vláken:** jedno přepnutí kontextu stojí C [taktů]

$$E = \frac{\textit{doba práce}}{\textit{doba}(\textit{práce} + \textit{režie})} = \frac{\textit{doba práce}}{\textit{doba}(\textit{práce} + \textit{přepínání} + \textit{čekání})}$$

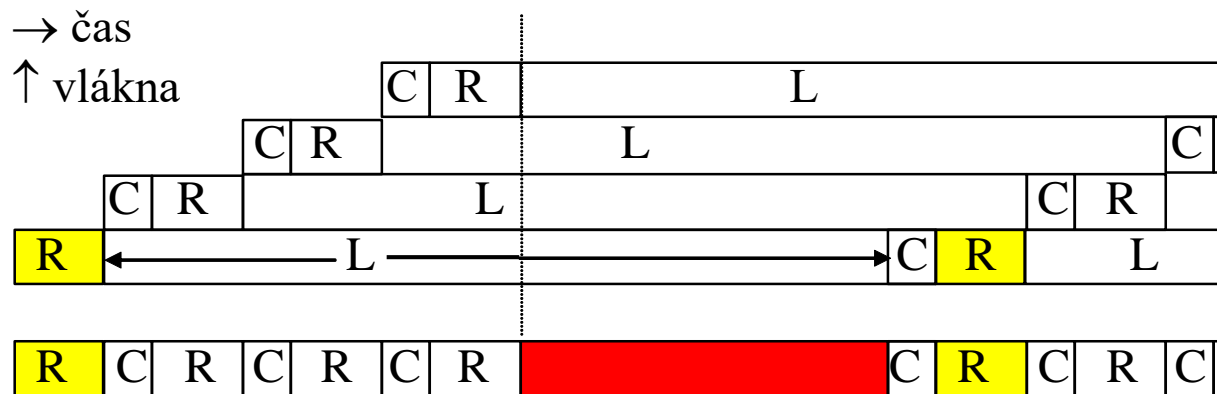
$$E_{sat} = \frac{NR}{N(R+C)} = \frac{R}{R+C}$$

Předpoklad:
výpadky
neblokují
přístupy
dalších
vláken do
cache

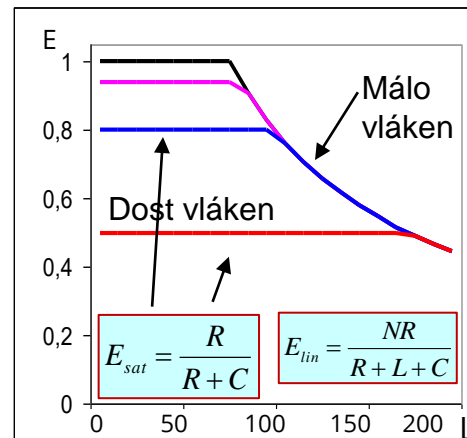
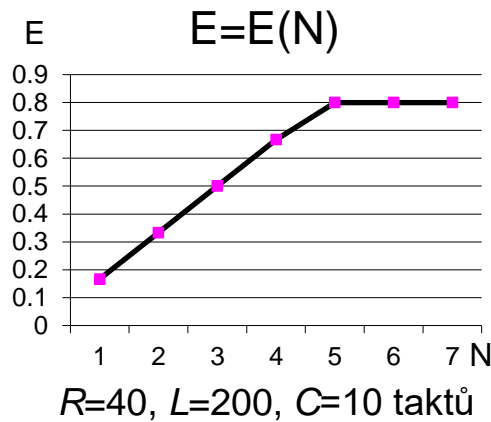


$$(N-1)(R+C) \geq L, \quad N \geq \frac{L}{R+C} + 1 = N_{sat}$$

$$E_{lin} = \frac{NR}{R + L + C}$$



$$(N - 1)(R + C) < L, \quad N < N_{sat} = \frac{L}{R + C} + 1$$



Multivláknová CPU pracuje s N kontexty

- přepnutí kontextu trvá 10 taktů
- doba práce v jednom kontextu činí průměrně 80 taktů.
- Přepnutí kontextu se provádí při každém výpadku v paměti cache, přičemž vyřízení výpadku trvá 270 taktů.

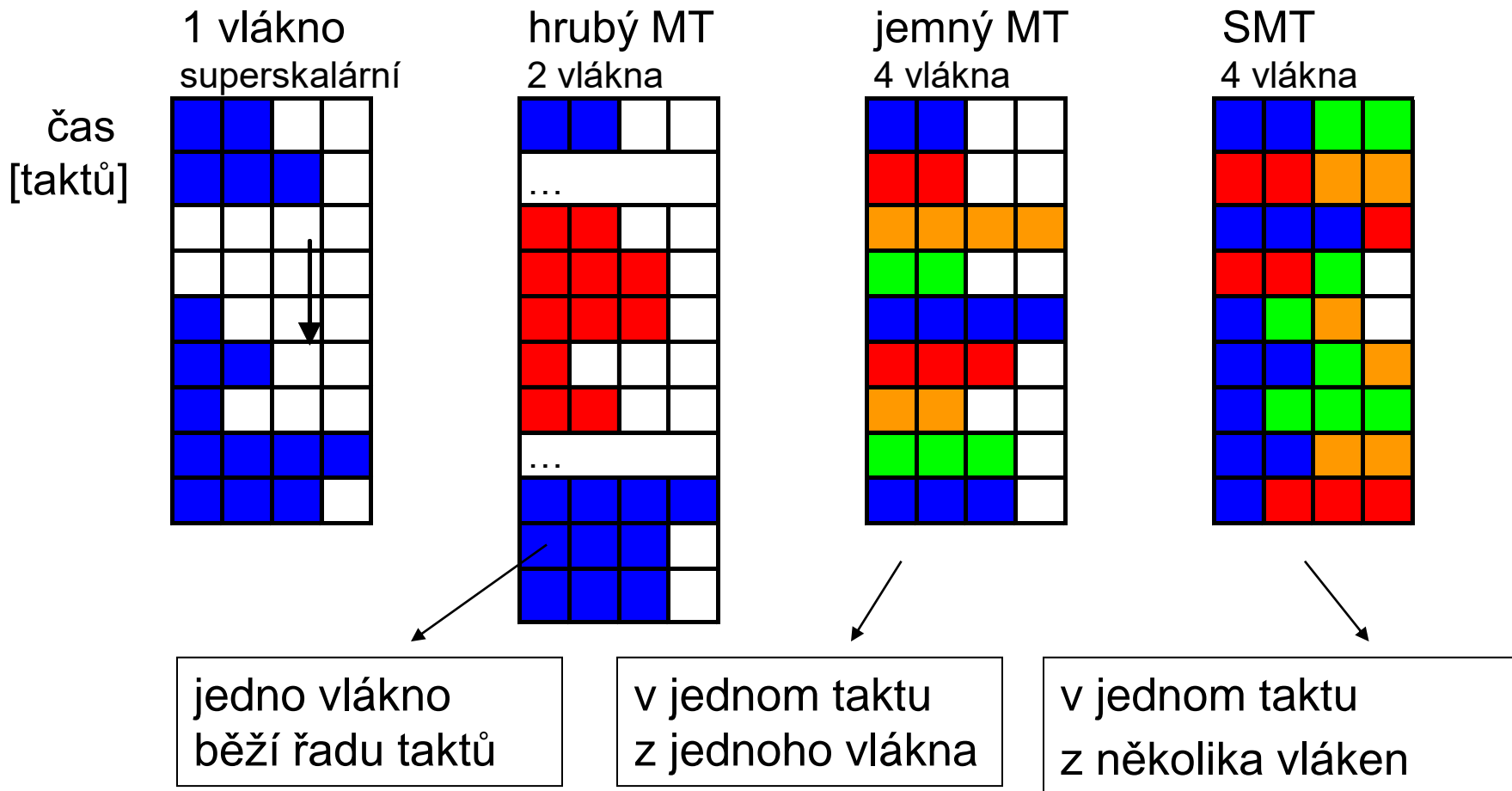
Najděte

1. kritický počet kontextů na hranici mezi lineární a saturovanou oblastí
2. využití CPU (tj. účinnost E) pro $N = 8$.

$$N \geq N_{sat} = \frac{L}{R + C} + 1 = \frac{270}{80 + 10} + 1 = 4$$

$$E_{sat} = \frac{R}{R + C} = \frac{80}{80 + 10} = 88,89\%$$

- **Jemný MT.** V každém taktu se přepíná na jiné vlákno (prokládání vláken).
 - výběr vláken např. cyklicky, vynechají se zablokovaná vlákna
 - umí krátká i dlouhá blokování
 - ale zpomalí provedení individuálních vláken
 - nulová doba přepnutí kontextu.
- **SMT**, simultánní MT (*Simultaneous MultiThreading*).
 - v jednom taktu se zpracovávají instrukce z několika vláken.
 - kontext se přepíná každý takt, podskupina připravených vláken dodá instrukce současně, opět s nulovou režií.



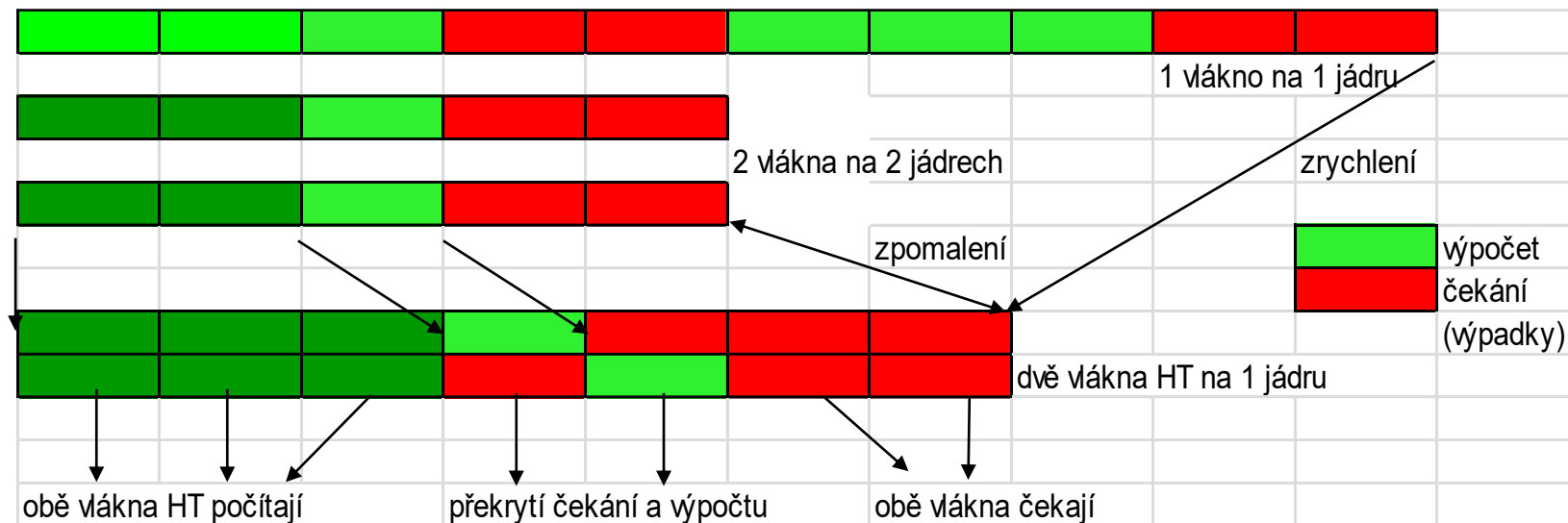
- SMT se kromě překrývání latence některých instrukcí snaží i o **lepší využití funkčních jednotek (vyšší ILP)**.
- Tomu odpovídá i strategie výběru vláken:
 - cyklicky v každém taktu pevný počet instrukcí z pevného počtu vláken
 - upřednostní vlákna s nejmenším počtem rozpracovaných výpadků nebo
 - s nejmenším počtem instrukcí ve stupních dekodování, přejmenování a v RS
 - upřednostní vlákna (s nejvyšší pravděpodobností) na správné větvi.

IMPLEMENTACE MULTITHREADINGU

- Hyperthreading je firemní název pro **dvouvláknový SMT**; zaveden u Pentia III a Pentia 4 (mikroarchitektura **NetBurst**).
- HT byl kritizován pro velkou spotřebu a vynechán u 1. gen. mikroarchitektury **Core**, nicméně u pozdějších vícejádrových **Nehalem, Sandy Bridge, Haswell** i **Atom** je znovu zaveden.
- HT lze povolit nebo zakázat v BIOSu
- První procesory **Itanium** používaly hrubý 2 vláknový MT nazývaný SoEMT–Switch on Event Multithreading.
- Procesory **Tukwila** a **Poulson** používají 2 + 2 vláknový SMT (2 vlákna v přední a 2 v zadní části linky).
- Intel **Xeon Phi** má 4 cestný jemný MT (vlákna se střídají). Na rozdíl od HT jej nelze vypnout.

1. **Replikace**, zdvojení stejných prostředků pro každé vlákno
 - Registry, RSB, ITLB (velké stránky)
2. **Rozdělení prostředků** na vlákna (staticky, s označením čísla vlákna)
 - Load buffer, store buffer, ROB, ITLB (malé stránky)
3. **Soutěživé sdílení**, závisí na dynamickém chování vláken.
 - Rezervační stanice, cache L1 až L3, datový TLB, TLB druhé úrovně
4. **Sdílené prostředky**, které si nejsou vědomy existence vláken
 - Funkční jednotky

- **Logické CPU se jeví** jako standardní oddělené CPU, ale mají jeden virtuální adresový prostor, rozdělený D-TLB (položky označeny ID bitem vlákna) a společnou virtuálně adresovanou L1 cache (virtuální index, fyzický tag).
- Položky v L1 cache nejsou označeny identitou vlákna, takže je žádoucí, aby jedno vlákno HT nepřepisovalo položky druhého.
 - Např. privátní zásobníky vláken musí začínat na jiných VA, aby se mapovaly na jiná místa v L1 cache (zařídí OS). Jinak by docházelo k vzájemnému vyhazování bloků z L1 do L2 (write back).
- **2 operační režimy:** sdílený a adaptivní (nastaví se v BIOSu).
 - **Sdílený:** soutěživé sdílení, každé vlákno má svou exkluzivní část L1 cache podle potřeby.
 - **Adaptivní** (default): každé vlákno má přístup do celé L1 cache.

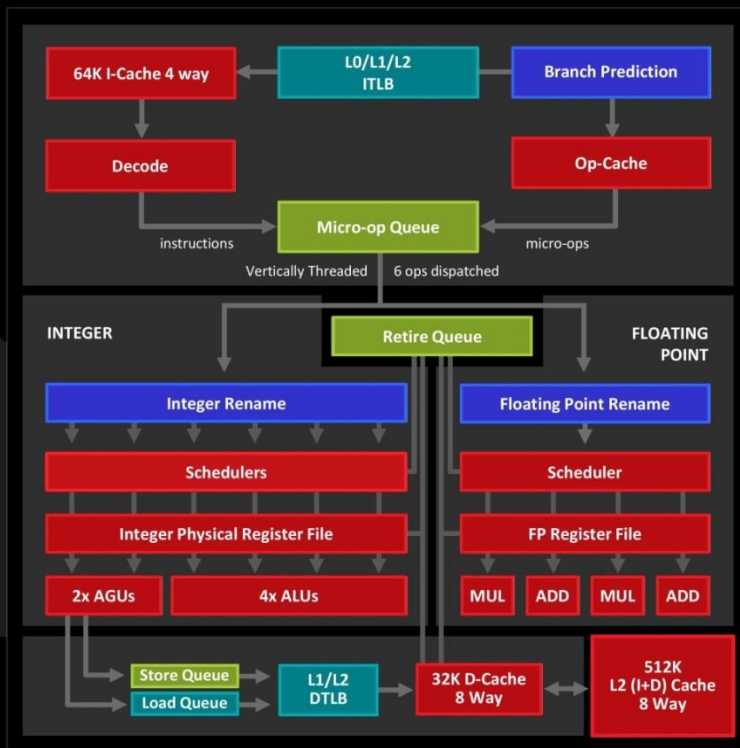


2 vlákna v HT běží pomaleji než 2 vlákna na 2 jádrech.

Důvody:

1. Dvě vlákna HT počítají déle protože sdílí FJ a obě třeba požadují stejnou FJ, takže jedno vlákno musí čekat na volnou FJ.
2. Dvě vlákna v HT čekají déle – úspěšnost 2 vláken v jedné cache je horší než úspěšnost každého vlákna ve vlastní cache.

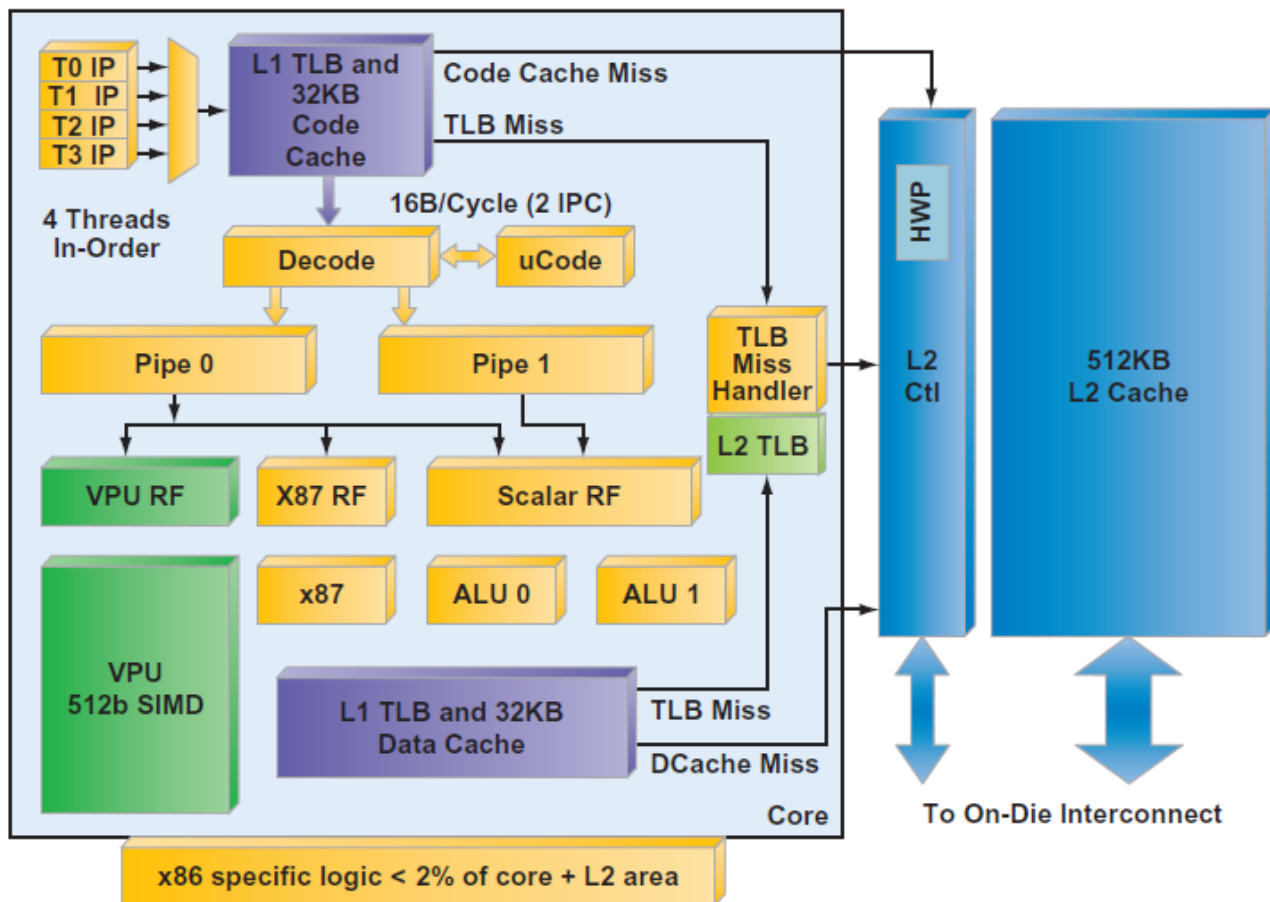
- **IBM POWER7** (2010) má 8 jader, každé 4 vlákna.
MT lze přepínat mezi SMT2 nebo SMT4 a tak optimalizovat jádro na nejkratší odezvu nebo maximální propustnost (tzv. **dynamický MT**).
- **IBM POWER8** (2013) má až 8 vláken na 1 jádro. Může běžet v 1-, 2-, 4- nebo 8-SMT módu a přepínat mezi nimi dynamicky.
- **Oracle/Sun Sparc M7** server procesor (2015, 20 nm, 32 jader, 10 miliard tranzistorů, 3,6 GHz); 1–8 vláken na jádro.
- Od r. 2016 **AMD** ve svých procesorech **Zen** opouští CMT (Clustered MultiThreading) a nasazuje SMT.



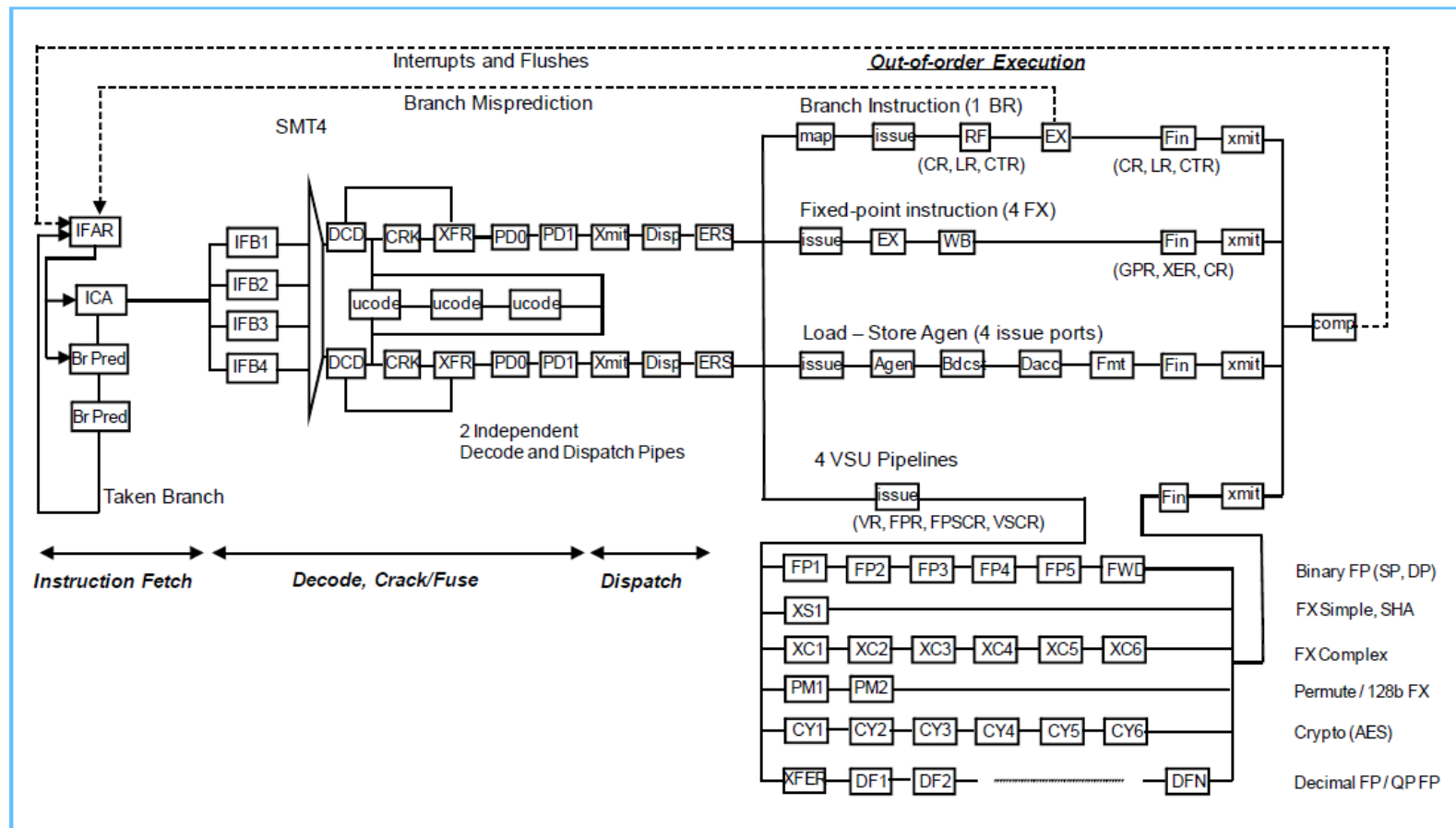
SMT OVERVIEW

- ▲ All structures fully available in 1T mode
- ▲ Front End Queues are round robin with priority overrides
- ▲ Increased throughput from SMT

- Competitively shared structures
- Competitively shared and SMT Tagged
- Competitively shared with Algorithmic Priority
- Statically Partitioned




- Pentium ISA včetně x87
- In-order zpracování
- 64b adresování
- 4 HW vlákna/jádro
- 2 cykly na dekódování instrukcí
- 2 instrukce za takt (vektorová + skalární)
- Latence vektorových operací je 4 takty
- Dvě pipeline

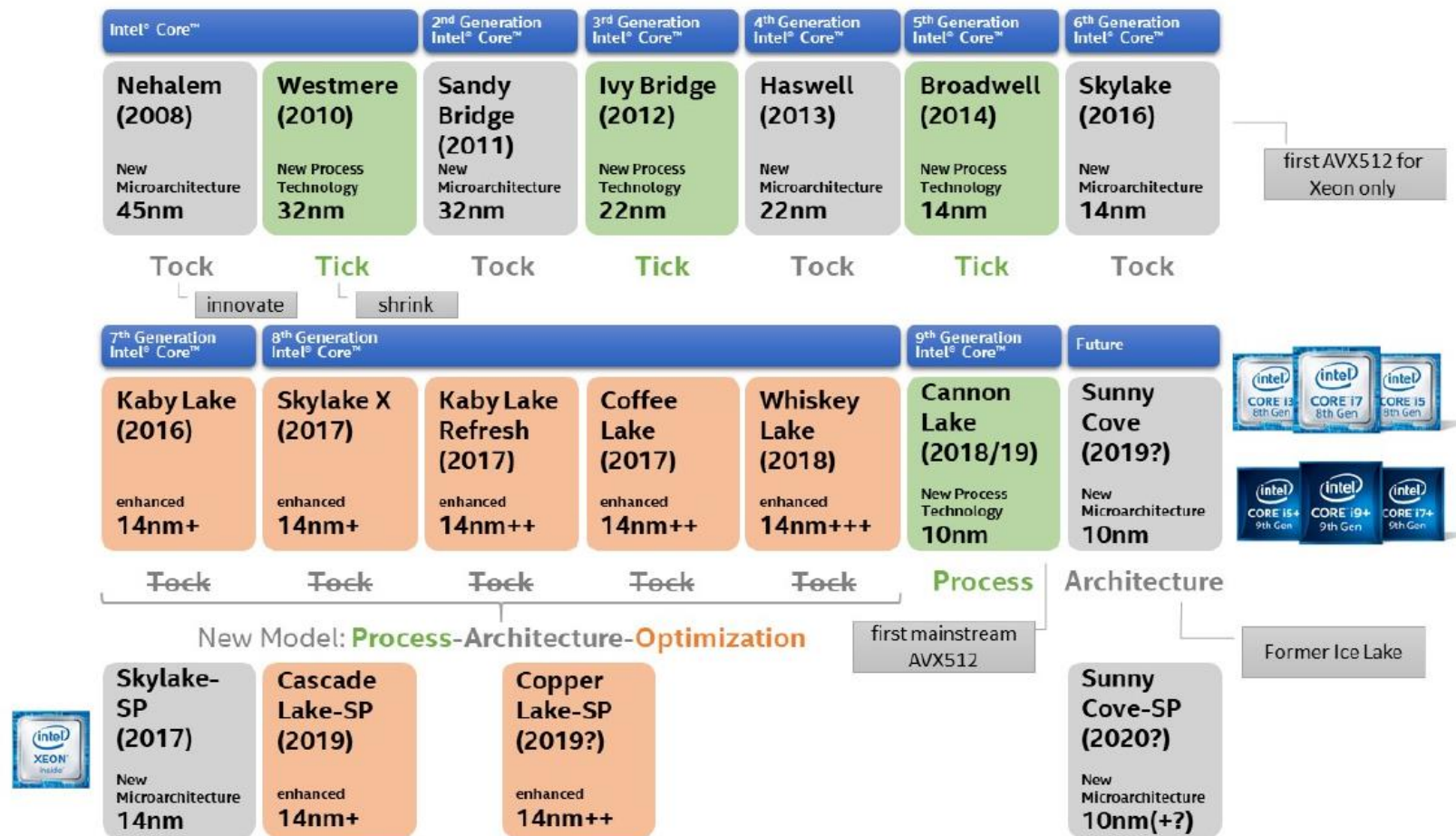


HISTORIE A PŘÍKLADY SUPERSKALARNÍCH ARCHITEKTUR

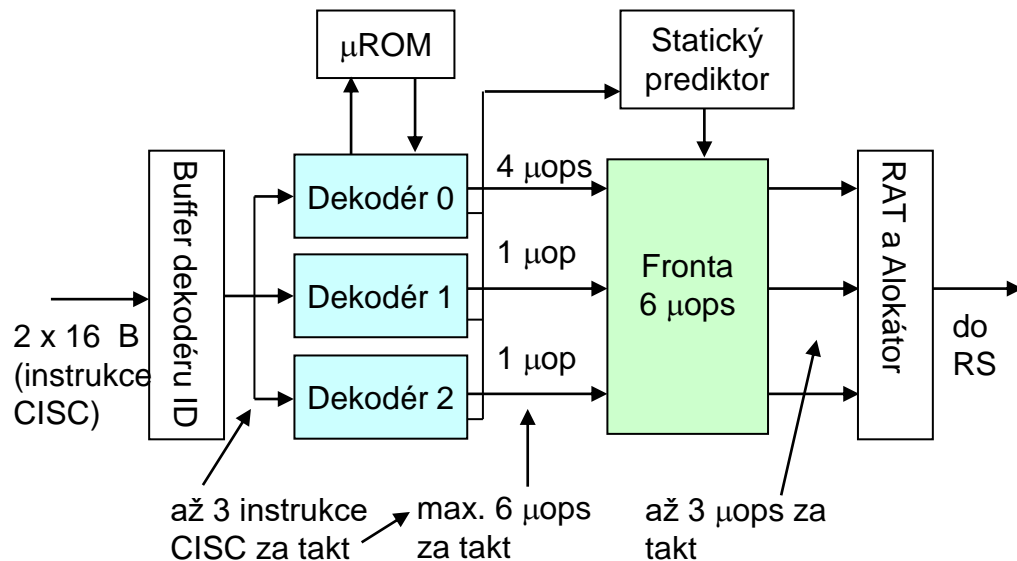
- P5 (1993):
 - první superskalární IA-32 mikroarchitektura – 1993:
 - In Order, dvojitá integer pipeline (**U** a **V**) 5 stupňů
 - Dokončují až 2 instrukce/takt. Kompilátor plánoval dvojice staticky.
- P6 (1995):
 - **OOO**, zavedeno **super-řetězení** (14 stupňů).
 - **Procesory**: Pentium Pro, Pentium II, III; MMX a SSE.
 - **Modernizovaná P6**: Pentium M, Core Solo, Core Duo.
- NetBurst (2000):
 - **Trace cache**, 31 stupňů, SSE2, SSE3, **hyper-threading HT**, EM64T.
 - **Procesory**: Pentium 4, Pentium D, Xeon
- Core (2006):
 - příkon ↓, 14 stupňů pipeline, 65 nm, **multi-core**, SSE3, **Intel 64**
 - Procesory: Pentium dual core, Celeron, Xeon, Core 2
- Nehalem (2008): řady: i3, i5, i7
 - 45 nm, HT, L3C, **Quick Path**, integrované **MemCtrl**, **bufer μops**.
 - 32 nm Nehalem = Westmere: **IGP** (Integrated GPU).

- Sandy Bridge 2010:
 - 32 nm, AVX 256 bitů, μ op-cache, HT.
 - 22 nm Sandy Bridge = Ivy Bridge: 3D-tranzistor.
- Haswell 2013:
 - 22 nm, 4 ALU, 3 AGU, 2 jednotky predikce skoků, AVX2, FIVR (Fully Integrated Voltage Regulator)
 - 35–40 MB LLC. Server procesory až 20 jader, možnost rozdělit jádra do 2 uzlů NUMA (COD, cluster on die)
 - 4 verze integrované GPU (až 40 EU), TDP 35–140 W.
 - 14 nm Sandy Bridge = Broadwell
- Skylake 2015:
 - 14 nm, 4 typy Y, U, H a S (TDP 4–95W) integrovaná L4 eDRAM cache (64/128 MB), podpora DDR3/4,
 - Kabylake 14 nm, optimalizované Skylake, podpora kódování a dekódování 4K videa odlehčí CPU.

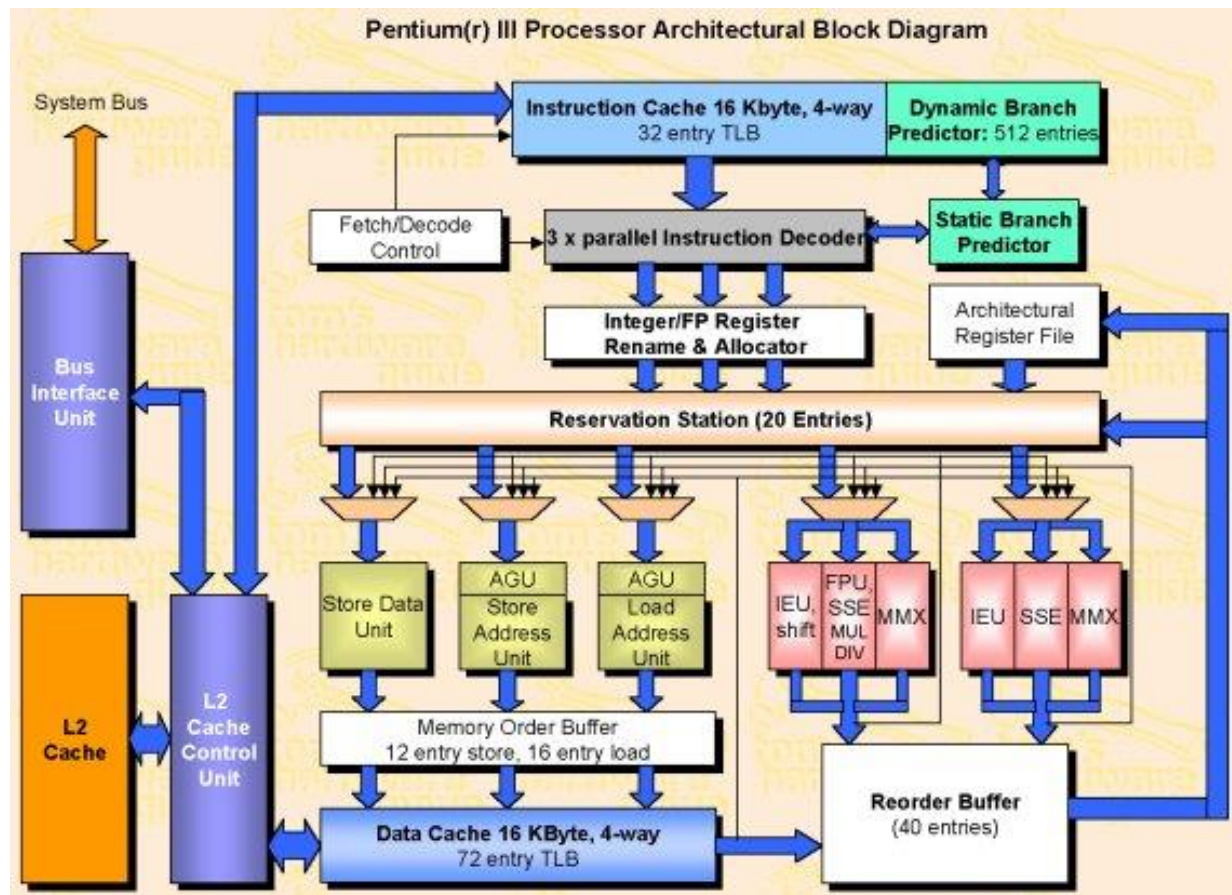
P5 (Pentium)	superskalární, „in-order“	5
P6 (Pentium Pro)	 superskalární „out-of-order“	14
P6 (Pentium III)		10
NetBurst Pentium 4 (180 a 130 nm)		20
NetBurst Pentium 4 (90 a 45 nm)		31
Core		14
Nehalem		16
Sandy Bridge		14–19
Ivy Bridge		14–19
Haswell		14–19
Bonnell (Atom)		16
Quark	skalární	5



- Řetězené zpracování CISC-ových instrukcí x86 se řeší transformací (dekódováním) na RISC-ové mikrooperace délky 72 bitů.
- Délka instrukcí x86: 1–15 B, **dekodér délky instrukcí** posílá až 3 instrukce x86 na 3 dekodéry:
 - D0 zpracovává 1. instrukci, která generuje až 4 μop /takt.
 - D1 a D2 zpracovávají jednodušší 2. a 3. instrukci, které nejsou delší než 8 B a generují jen 1 μop .
 - 2. a 3. instrukce musí čekat na D0, pokud to nesplňují.
 - Pro dekodování instrukcí, které generují víc než 4 μop je použita paměť mikrokódu a generování trvá 2 nebo více taktů.

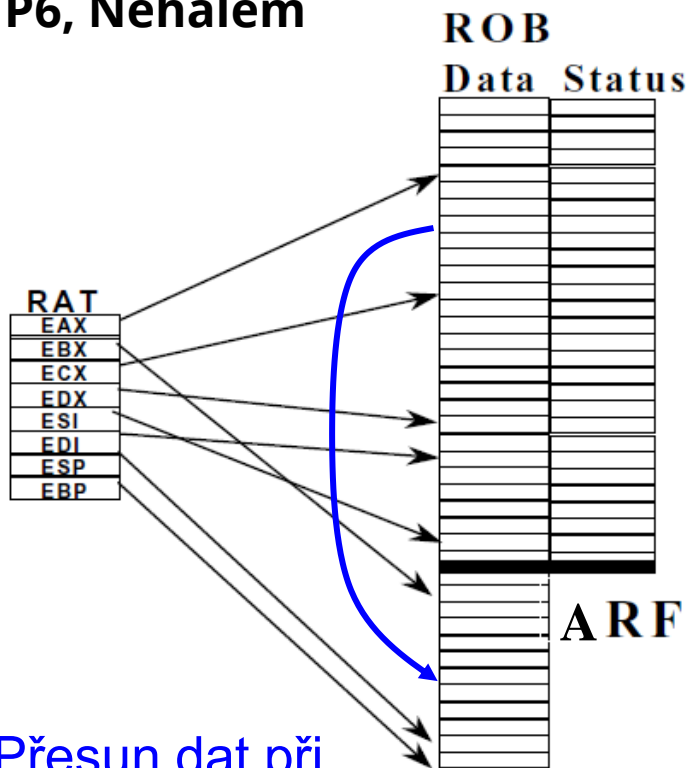


- Prediktor podmínky je 2 úrovněový adaptivní (autoři Yeh a Patt) s $k = 4$ –bitovým lokálním BHSR
 - perfektně predikuje libovolné periodické sekvence až $k + 1 = 5$ bitů
 - na 1 skok je třeba 36 bitů (= 16 dvoubitových prediktorů + 4 bity BHSR).
- BTB je organizován jako skupinově asociativní cache (128 skupin, 4 cesty, tj. 512 položek)
- Položka obsahuje adresu skokové instrukce b , cílovou adresu skoku t a 4 bitový lokální BHSR. Jeden index do PHT je část adresy skoku b , jako druhý index se použije obsah BHSR.
- **Pokuta za špatnou predikci je 10–20 taktů.**
- Není-li skok v BTB, použije se statická predikce (skok v kódu dopředu -, skok dozadu +)



- r. 2000, IA-32 procesor (adresa 32 bitů, instrukce x86)
- SSE2 (Streaming SIMD Extension 2)
- **Trace Cache** (kapacita 12k μ ops, cca 64 bitů / μ op
 - TC může rozeslat do RS 3 μ ops/takt, vydat do FJ se může až 6 μ ops/takt a propustit opět 3 μ ops/takt.
- Přejmenování mapuje 8 standardních registrů x86 na 128 vnitřních fyzických registrů PRF, 2 tabulky RAT (front-end a propouštěcí) → není nutno kopírovat registry při propouštění.
- ROB: až 126 μ ops v pořadí bez hodnot dst operandů
- **Co bylo špatné**: chyběla L3 cache na čipu, malá L1 D-cache (8 KiB), výkonnost \approx Pentium III, velký příkon

P6, Nehalem



Přesun dat při propuštění instrukce

Sandy Bridge

NetBurst

přepis při špatné predikci skoku

Frontend RAT

EAX
EBX
ECX
EDX
ESI
EDI
ESP
EBP

Retirement RAT

EAX
EBX
ECX
EDX
ESI
EDI
ESP
EBP

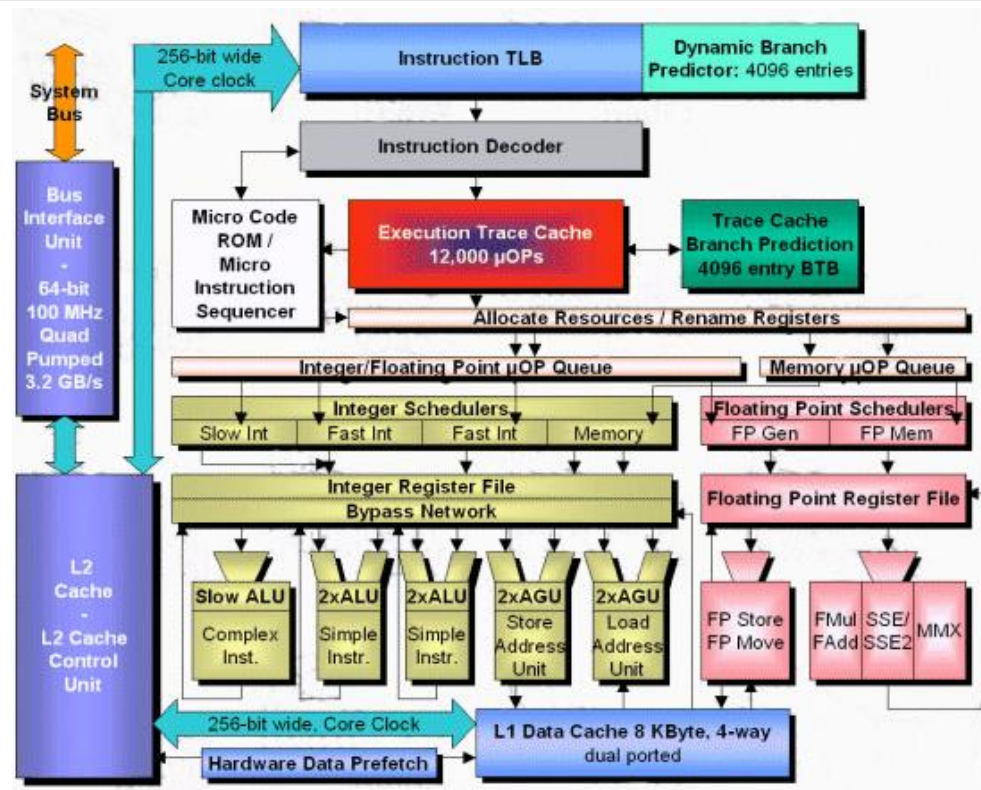
Při propuštění instrukce se sem vloží mapování jejího dst registru

PRF

Data

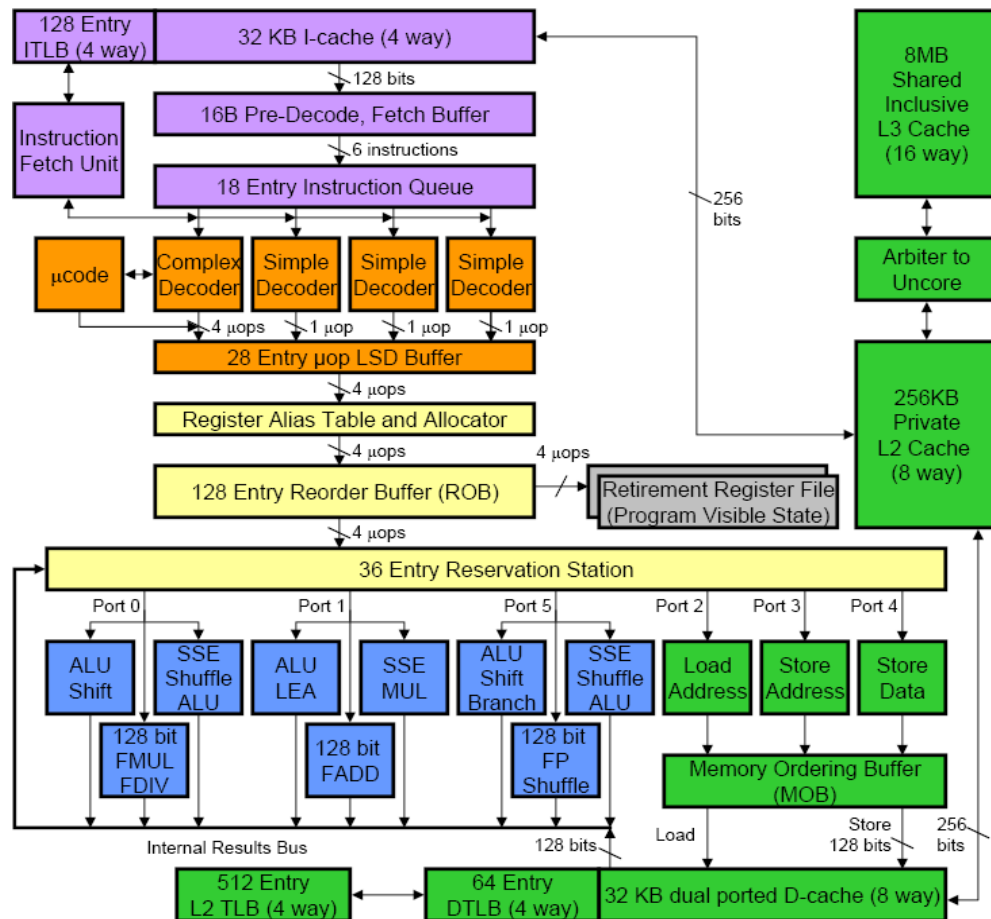
ROB

Status

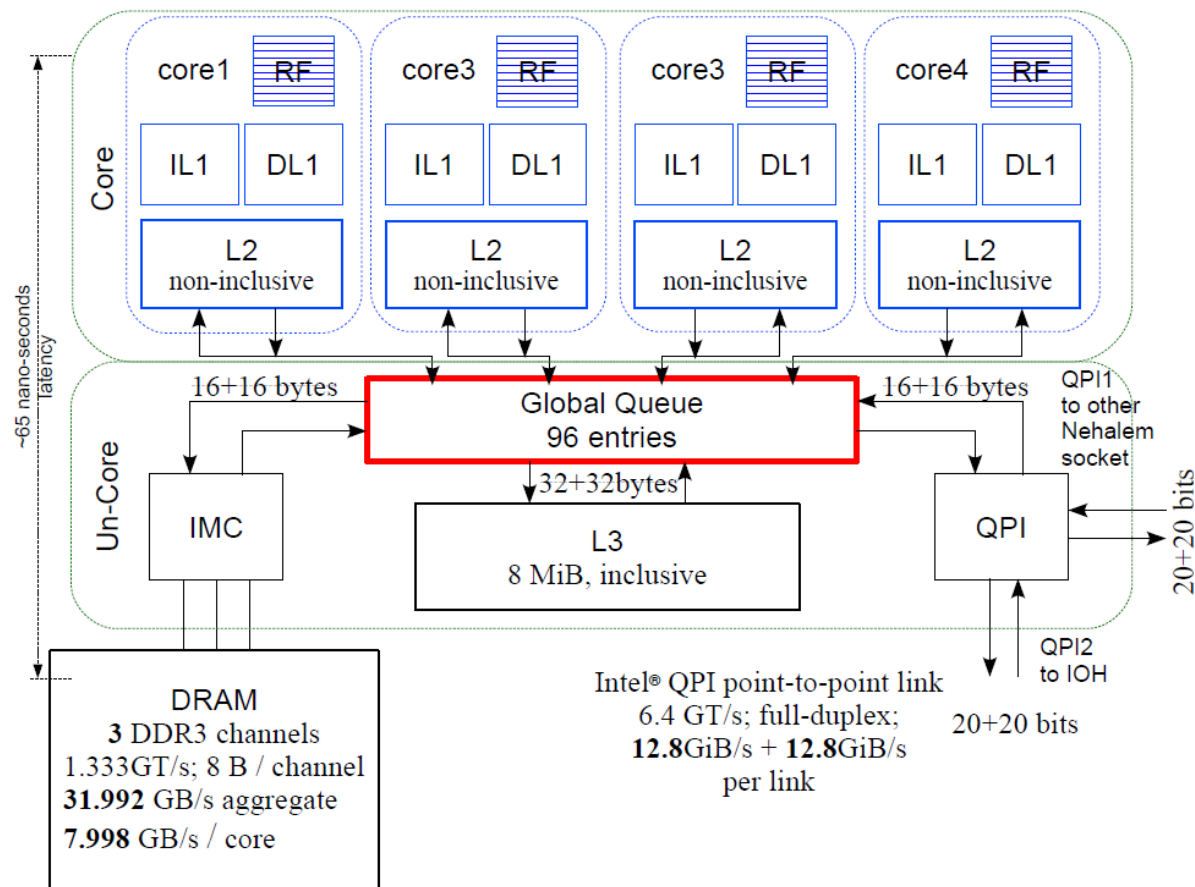


- Použita u vícejádrových procesorů se sdílenou pamětí cache L2. Snížení příkonu a zvýšení výkonnosti činí kolem 40 %. Vychází z P6.
- **Široká instrukční linka.**
 - Dekóduje a propouští až 4 instrukce za takt, rozesílat a provádět může až 5 μ ops.
 - Umí sdružovat instrukce x86 (*macrofusion*) a také sdružovat μ ops vzniklé z jedné x86 (*microfusion*), čímž lze dosáhnout až 6 μ ops za takt.
 - Ukazovatel zásobníku je modifikován speciálním HW. To dovoluje načítání dat ze zásobníku již na začátku linky (25 % všech načítání je ze zásobníku).
 - Tyto inovace zachovány i v novějších mikroarchitekturách
- **Pokročilá práce s multimédií.** Instrukce MMX, SSE, SSE2, SSE3 se 128 bity provedené za 1 takt znamenají výkonnost až 24 GFLOP/s (1 jádro na 3 GHz, SP).
- **Inteligentní napájení.** Dynamické odpojování subsystémů dle potřeb nebo přepojování do úsporného režimu neovlivňuje responzivitu.
- **Pokročilá chytrá cache.** Sdílená sjednocená cache úrovně 2 může být celá k dispozici jen 1 jádru, když druhé není aktivní. Špičková přenosová rychlost je 96 GB/sec @ 3 GHz.
- **Chytrý přístup do paměti.** Je zavedena podpora pro RPW i pro případ dosud neznámé adresy zápisu (dynamické rozlišování adres, *memory disambiguation*)
- **Přednačítání dat** do L1/L2 D-cache pomocí tabulky historie čtení

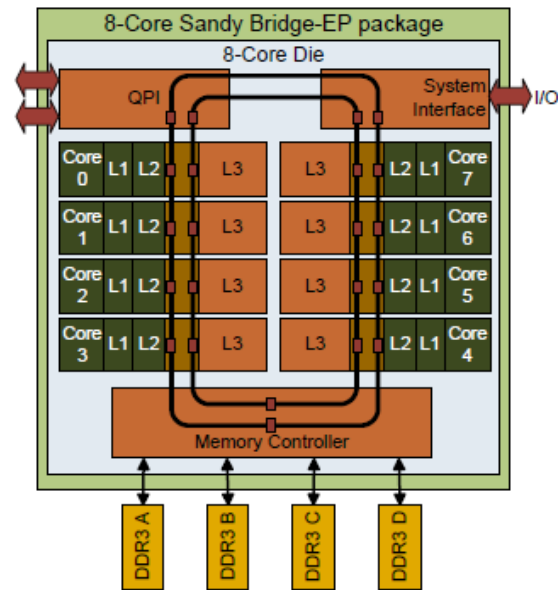
- Druhá generace architektury Core, orientace na výkonnost: 2, 4, 6 nebo 8 jader (45 nm, 4 jádra: 731 M tranzistorů)
- 16 stupňová linka, 6 FJ (3 paměťové, 3 výpočetní) a **HyperThreading (HT)**
- Větší cache a vyšší propustnost pamětí 32 KiB L1 I-cache, 32 KiB L1 D-cache, L2C: 256 KB.
- **Register Alias Table** (RAT) může přejmenovat až 4 μop za takt a každé přidělit dst. registr v ROB více rozpracovaných mikrooperací.
- **Dvouúrovňový prediktor skoků i dvouúrovňový TLB.**
- **Loop Stream Detector LSD**: ve frontě 18 před-dekódovaných instrukcí detekuje každé tělo smyčky, uloží je dekódované do LSD buferu (až 28 μop) a pak opakovaně používá (bez načítání a dekódování) až do špatné předpovědi skoku (malá náhrada Trace cache).
- **Turbo režim**: kmitočet hodin se zvyšuje, pokud není překročena teplotní mez.



- **Inovace:** nové propojení soketů: front-side bus FSB nahrazen **linkami (QPI, Quick Path Interconnect)**.
- **Inovace:** **integrovaný řadič paměti**, podporuje 3 paměťové kanály DDR3 SDRAM nebo 4 FB-DIMM, severní most eliminován.
 - Firma AMD zavedla linky HyperTransport (HT) a integrované řadiče paměti již v roce 2003, o 5 let dříve.
- **Sdílená L3C:** 4–8 MB je inkluzivní, obsahuje data z L1 i L2 a info, kde jsou lokálně (menší komunikace).
- **Čip procesoru grafiky** v témže pouzdru jako CPU.
- **Řízení příkonu:** vestavěný mikrořadič a senzory teploty, proudu a příkonu, odepínání jader, možnost redukce příkonu pamětí a QPI.
- **Global Queue (GQ)** uchovává, spravuje a plánuje tok dat v „uncore“. Má 3 fronty požadavků:
 - WQ, žádosti zápisu z lokálních jader, 16 položek
 - LQ, žádosti čtení z lokálních jader, 32 položek
 - QQ, fronta QPI, žádosti jdoucí mimo čip, 12 položek
- Obsahuje **křížový přepínač** pro výměnu dat mezi propojenými částmi (L2, L3, IMC, QPI).
- **Funkce:**
 - lokální žádost jádra o čtení: GQ sonduje další jádra. Z více vlastníků kopií jedno jádro dodá data.
 - Když nikdo nemá kopii a L3 ano, dodá data inkluzivní L3
 - Výpadek v L3: data dodá lokální IMC (Integrated Memory Controller) za 65 ns, popřípadě vzdálený IMC přes QPI za 105 ns



- 4, 6 a 8 jader na 3,0–3,8 GHz s podporou HyperThreadingu (HT) a s technologií Turbo Boost
- Jádra, grafika, L3 cache a systémový agent jsou propojeny **kružnicovým propojením** s propustností 256 bitů/takt.
- Podpora **Advanced Vector Extension (AVX): 256 bitů**, 32 GFLOPS/jádro (8 FP/takt),
- Každé jádro: 32 KiB L1 D-cache + 32 KiB L1 I-cache (3 takty), 256 KiB L2 cache (8 taktů).
- 8 MiB **sdílená L3 cache** (25 taktů). Je též sdílena s integrovaným grafickým jádrem. Blok cache 64 byte.
- **Integrované jádro grafiky na 1–1,4 GHz, 16 ex. jednotek.**
- **Integrovaný řadič paměti** s max. propustností 25,6 GB/s, podporuje DDR3-1600 dual channel RAM.
- CPU ↔ L3 cache: průměrně jen 1,5 skoků (do lokálního bloku cache není třeba jít po kružnici). Latence sdílené L3 cache je 26 až 31 taktů.

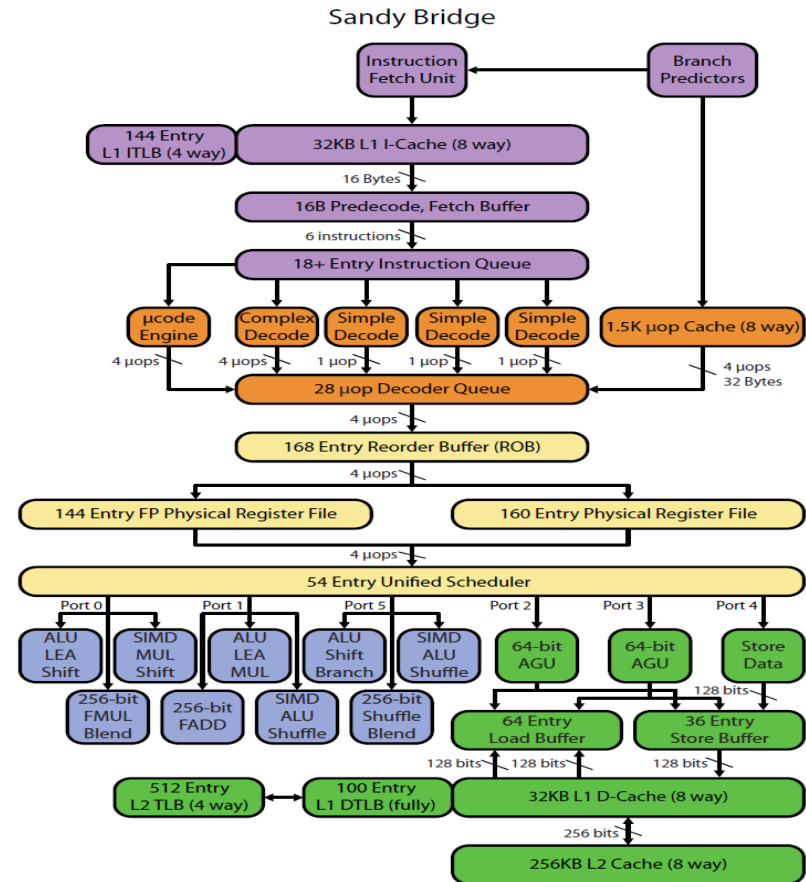


(a) 8-core Sandy Bridge-EP

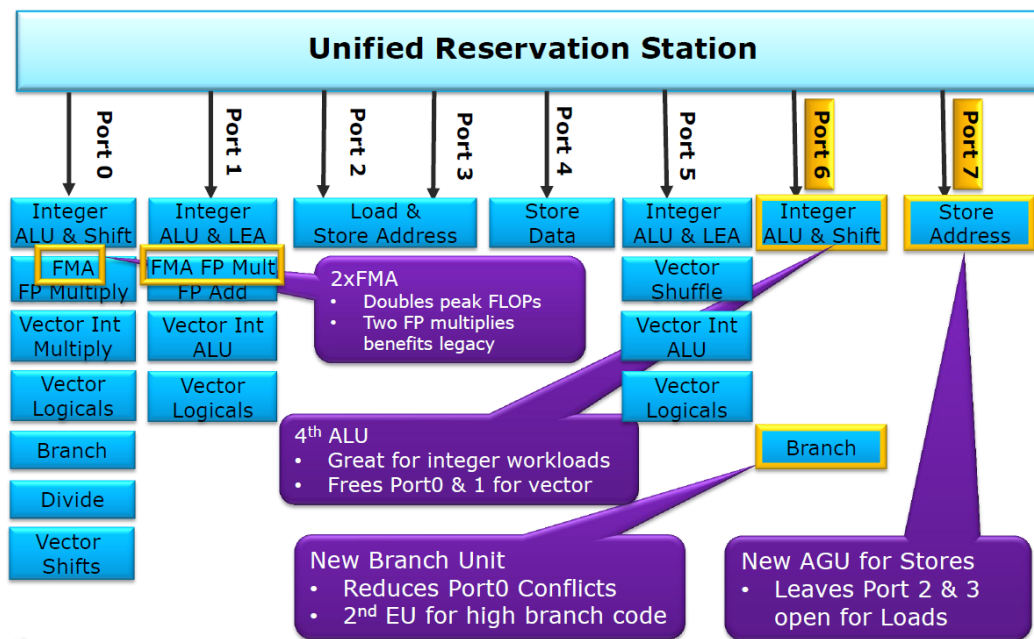
Dvě propojovací kružnice, jedna pro směr nahoru, druhá dolů. Pro každý přenos je vybrán směr kratší cesty do cíle, nejvíce 4 skoky.

μop-cache (dekódovaná I-cache) v Sandy Bridge

- Je částí L1 I-cache, zachovává výhody Trace cache, eliminuje složité dekódování při mnohem nižším příkonu.
- μop-cache má kapacitu 1536 μops, 10 % velikosti Trace cache Pentia 4.
- Mapování instrukcí do μop-cache probíhá po blocích 32 B instrukcí, 1 blok může zabrat až 18 μops. Každý blok μop-cache uchovává „metadata“ včetně počtu platných μops v bloku a délku odpovídajících x86 instrukcí.
- Jestliže okénko 32 B instrukcí má více než 18 μops, musí jít přes tradiční front-end.
- **Mikrokódované** instrukce nejsou v μop-cache – jsou reprezentovány ptr do ROM mikrokódu a případně několika prvními μops.



- Haswell je zaměřen na **nižší příkon pro mobilní zařízení** (hybridní laptop-tablety) ale i pro superpočítače. Dřívejší TDP (Thermal Design Power) 35 až 45 W pro mobilní procesory je redukován na ULT: 13,5 W a 15 W TDP, ULTX: 10 W TDP.
- Superpočítač v Ostravě obsahuje 24 192 jader Haswell-EP!
- Podpora pro AVX2 a MAD operace.
- Haswell má výkonnější grafiku GT3e, ze 16 jednotek na 1150 MHz (GT1 u Sandy Bridge) narostla na 40 ex. jednotek a 1300 MHz.
- eDRAM (embedded DRAM) 128 MiB je na vlastním čipu, ale ve stejném pouzdru jako procesor. Pracuje jako sdílená L4 cache jak pro grafiku, tak pro jádra procesoru. Vylepšuje paměťovou propustnost.

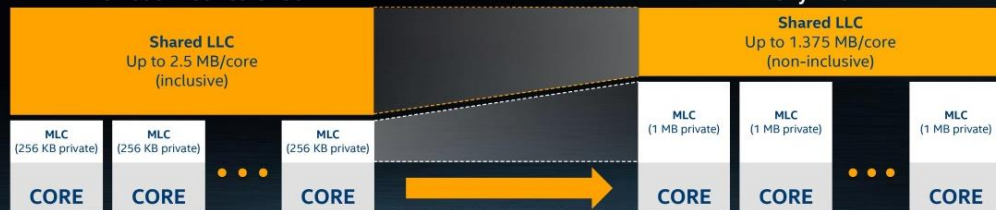


- Broadwell = Haswell předělaný na 14 nm (2014).

- **Podpora pro AVX-512**
- **Nová organizace cache**
 - Zvětšení L2 cache na úkor L3 cache
- **Změna propojovací sítě**
 - Z hierarchických kruhů na 2D mřížku

REBALANCING THE CACHE HIERARCHY¹

Previous X-series CPUs



- Shift cache balance from shared-distributed to private-local by enlarging MLC
- Shared LLC retained to benefit shared data and to enable capacity balancing

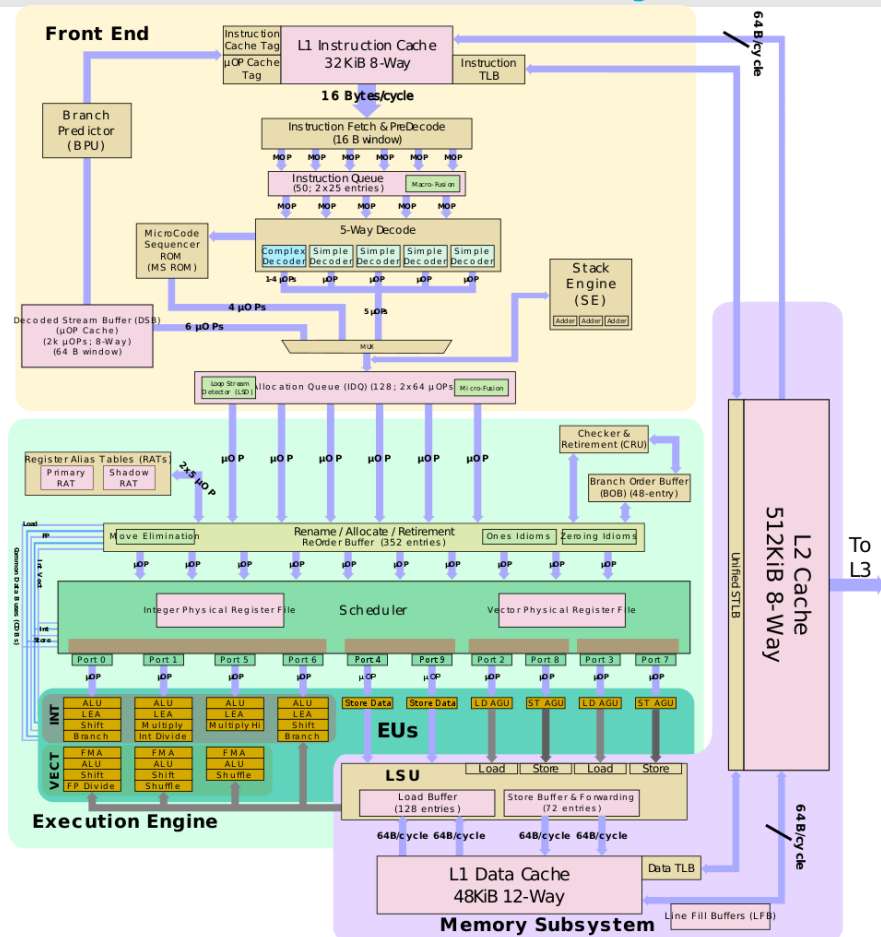
High hit rate on low-latency MLC increases performance

LLC = Last-level cache; MLC = Midlevel cache

1. Not available with SKUs 7640X and 7740X

Comparison: Skylake-S and Skylake-SP Caches

Skylake-S	Features	Skylake-SP
32 KB 8-way 4-cycle 4KB 64-entry 4-way TLB	L1-D	32 KB 8-way 4-cycle 4KB 64-entry 4-way TLB
32 KB 8-way 4KB 128-entry 8-way TLB	L1-I	32 KB 8-way 4KB 128-entry 8-way TLB
256 KB 4-way 11-cycle 4KB 1536-entry 12-way TLB Inclusive	L2	1 MB 16-way 11-13 cycle 4KB 1536-entry 12-way TLB Inclusive
< 2 MB/core Up to 16-way 44-cycle Inclusive	L3	1.375 MB/core 11-way 77-cycle Non-inclusive



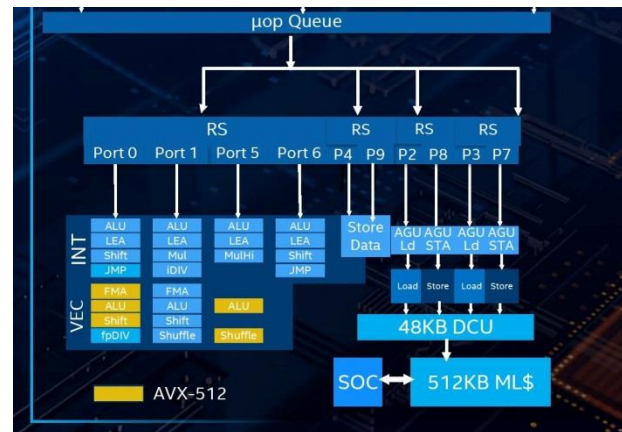
Změna velikostí cache

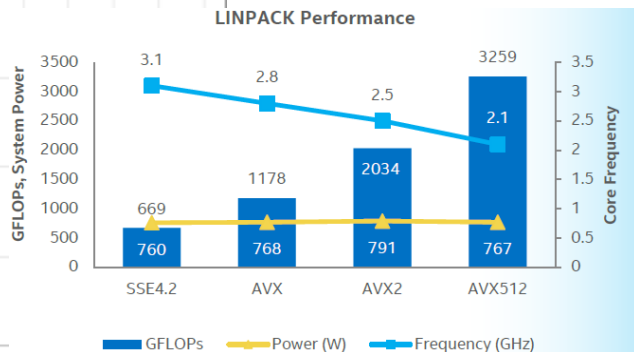
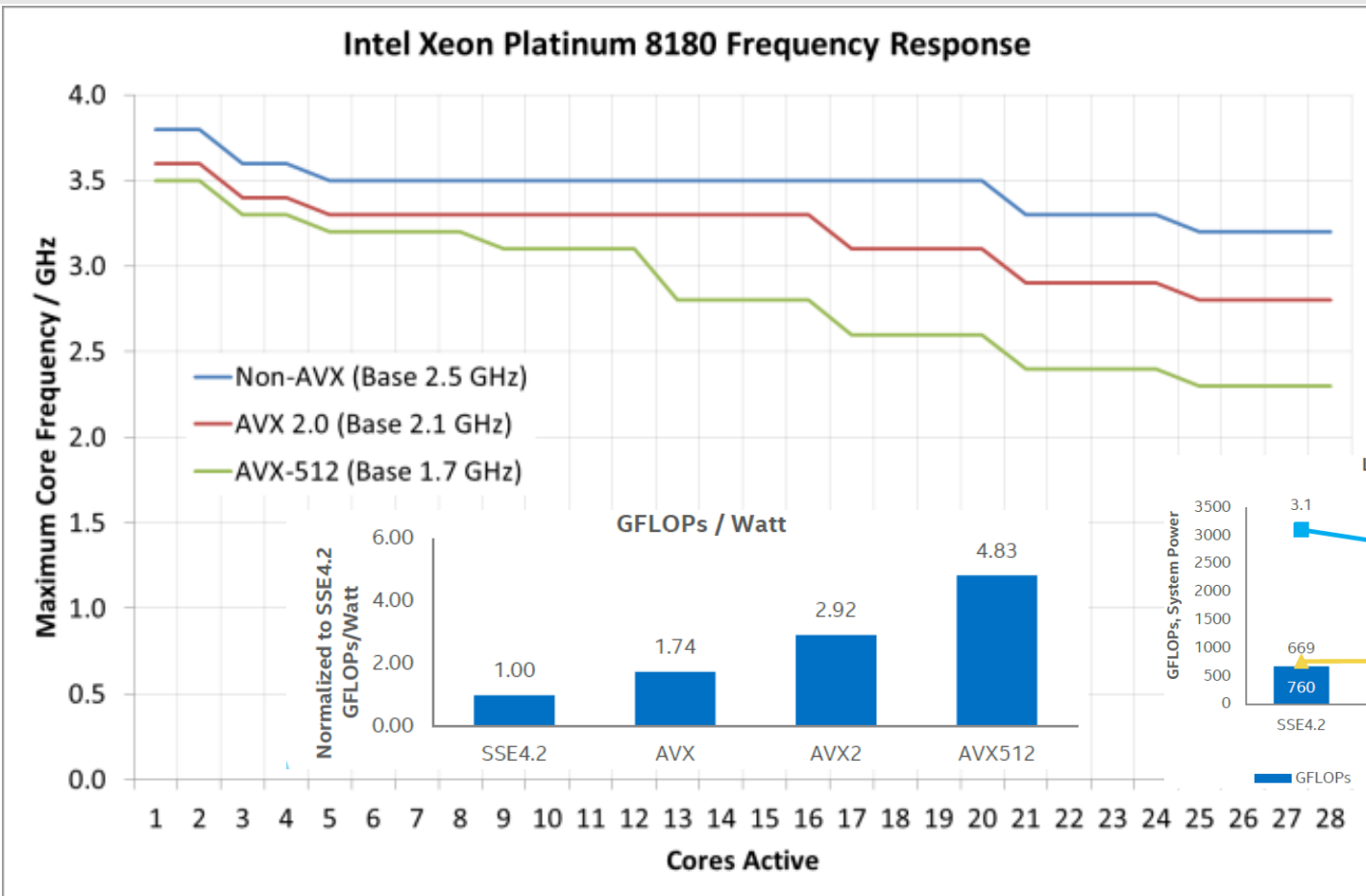
- L1 z 32 KB -> 48 KB
- L2 z 256 KB -> 512 KB
- Zvětšena TraceCache (2,25k)



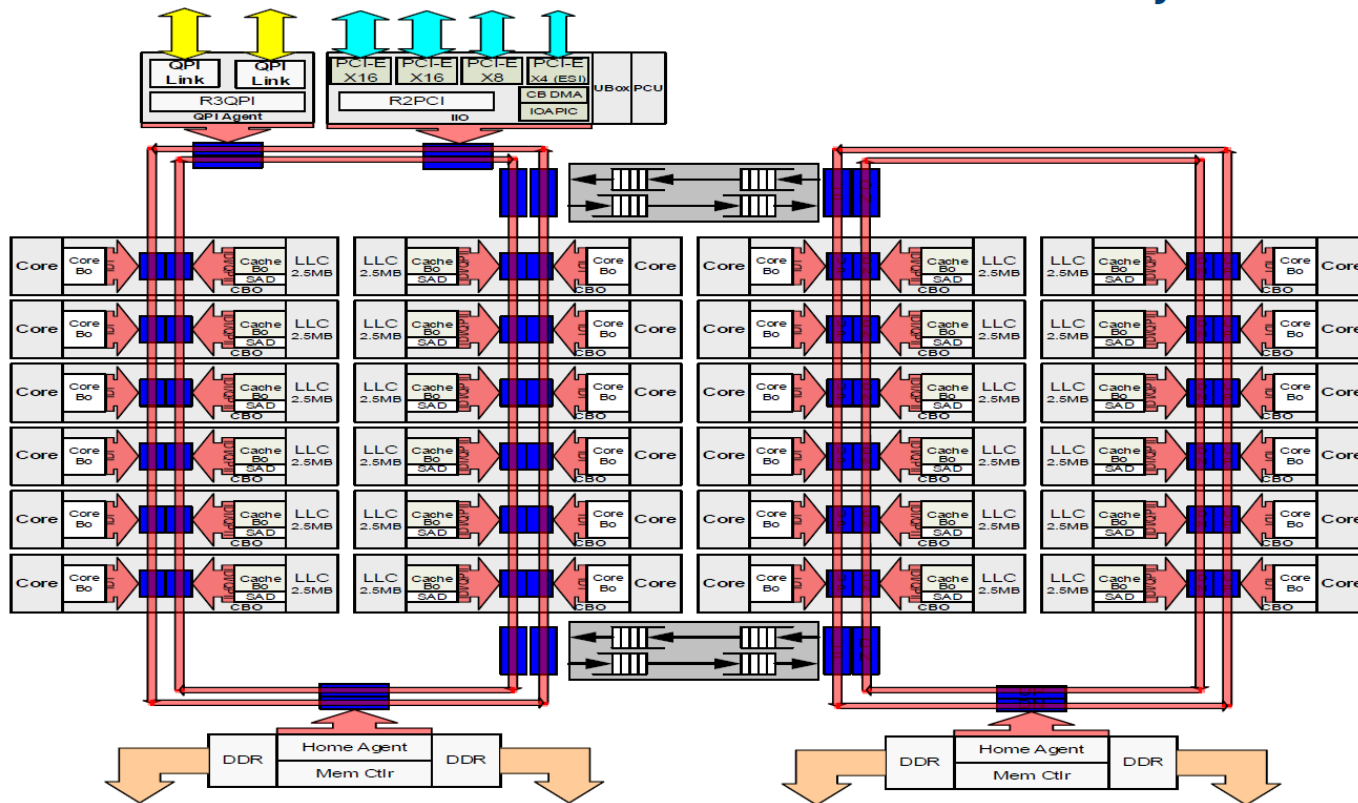
Back-end

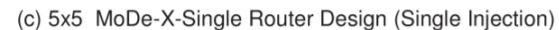
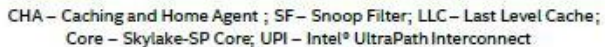
- Zvýšeno IPC o 18 %
- 8 -> 10 výpočetních linek
- ROB zvětšen z 224 na 352 záznamů
- Nová AGU jednotka (4)
- Výrazně zvětšeny load/store buffery





Intel® Xeon® Processor E5 v4 Product Family HCC

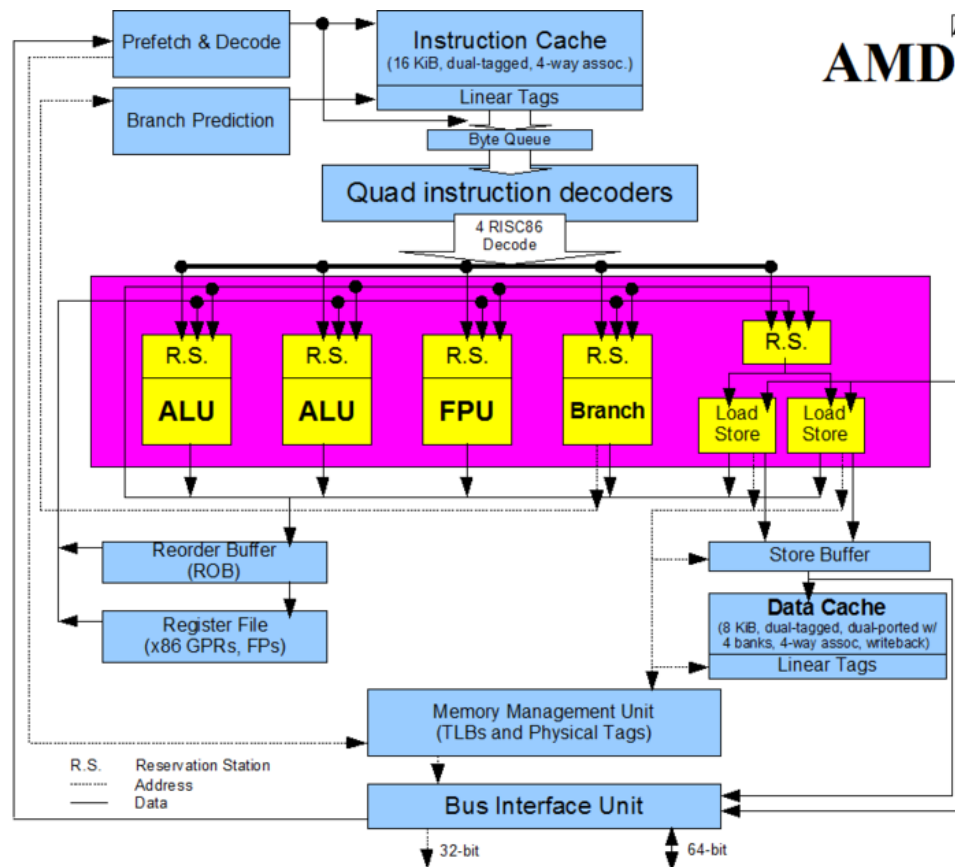




PROCESORY AMD

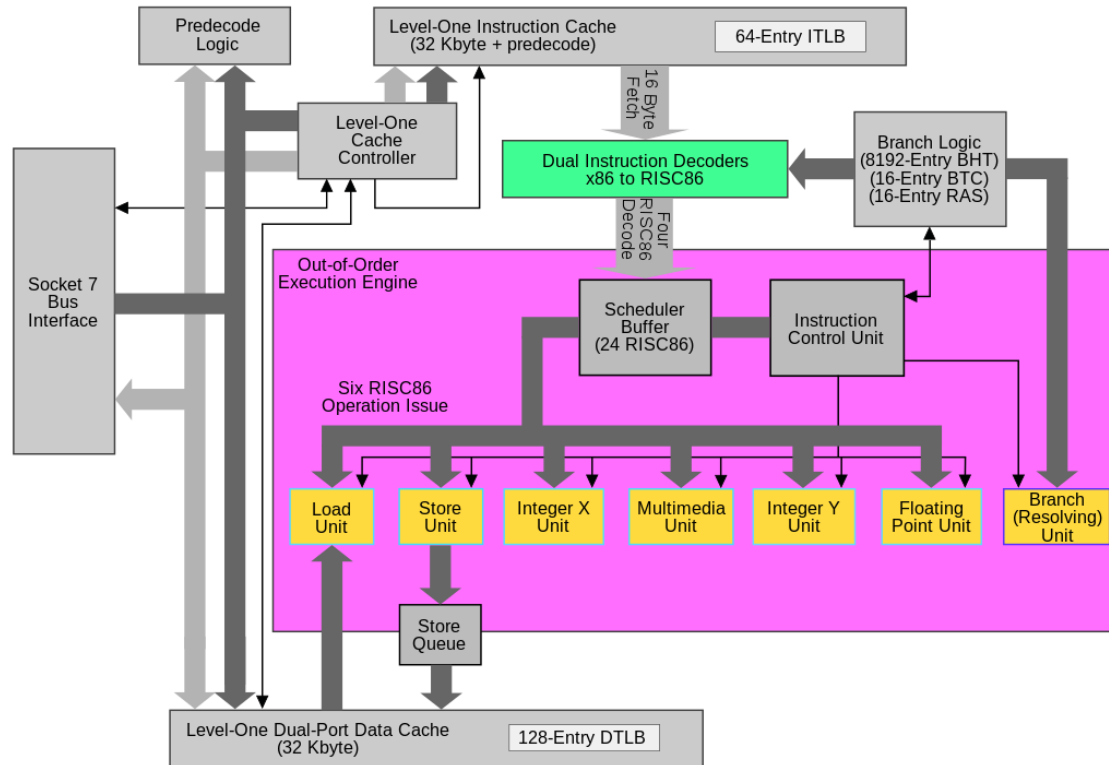
• AMD K5 (1996)

- První vlastní OOO procesor AMD
- 6 výpočetních jednotek, 4 vydání instrukce za takt, 5 stupňů linky.
- Spekulativní provádění podél 3 predikovaných větví
- Penalta 3 takty při špatné predik
- Přejmenování registrů
- 16 KB L1, přístup do L1 v 1 taktu!
- Podpora MESI cache coherent protokolu



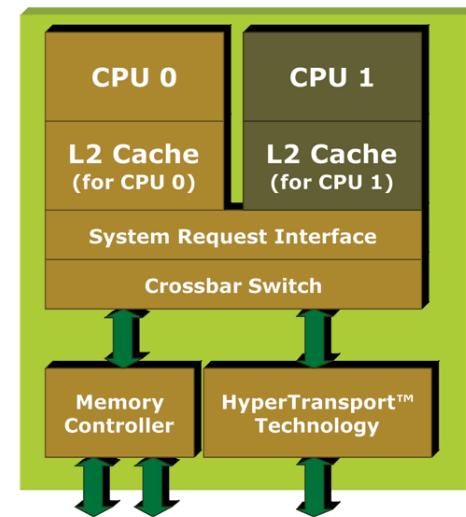
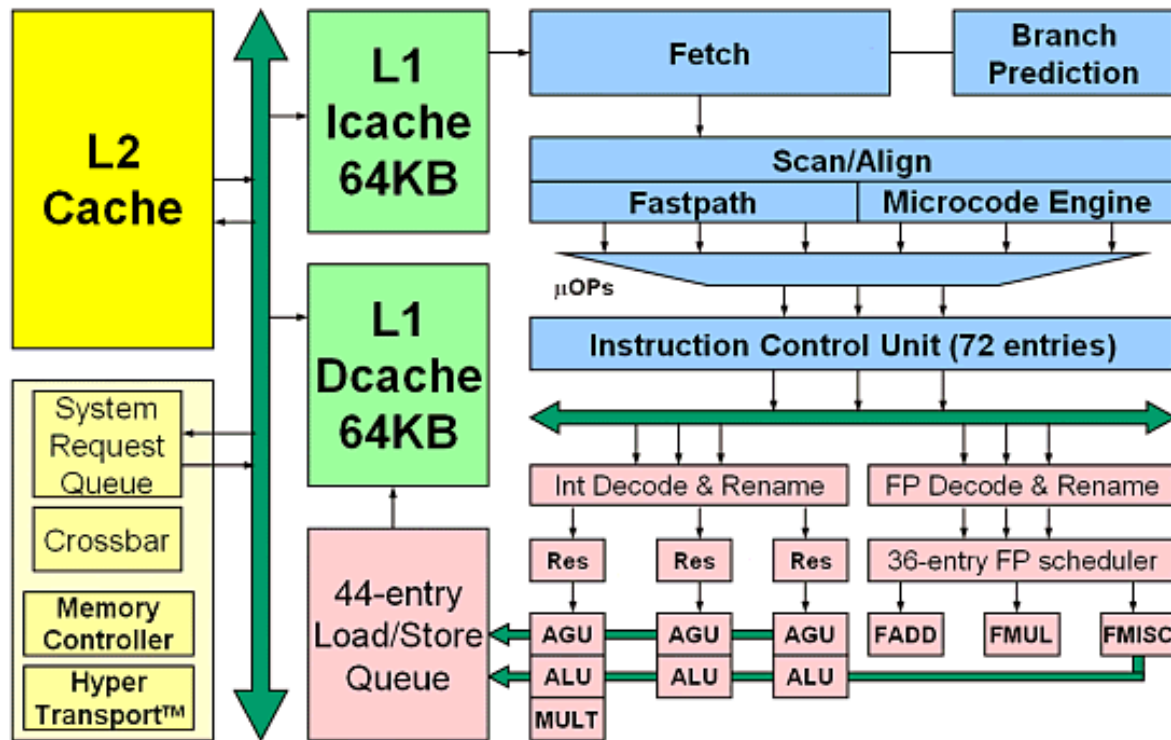
AMD
AMD-K5

- Uvedena na trh v roce 1997
- Přináší instrukce MMX, později 3DNow

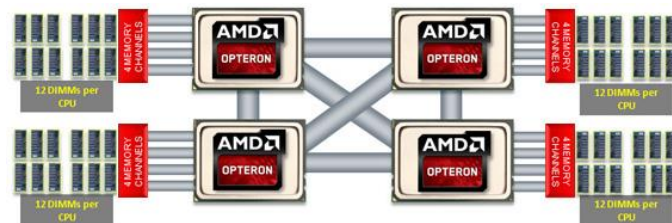


První CPU s instrukcemi x86 na 64 bitech, kompatibilní s Windows (2003), 32 bitové i 64 bitové aplikace, SW investice nezhodnoceny. Reakce Intelu: Extended Memory 64-bit Technology) EM64T a pak Intel® 64.

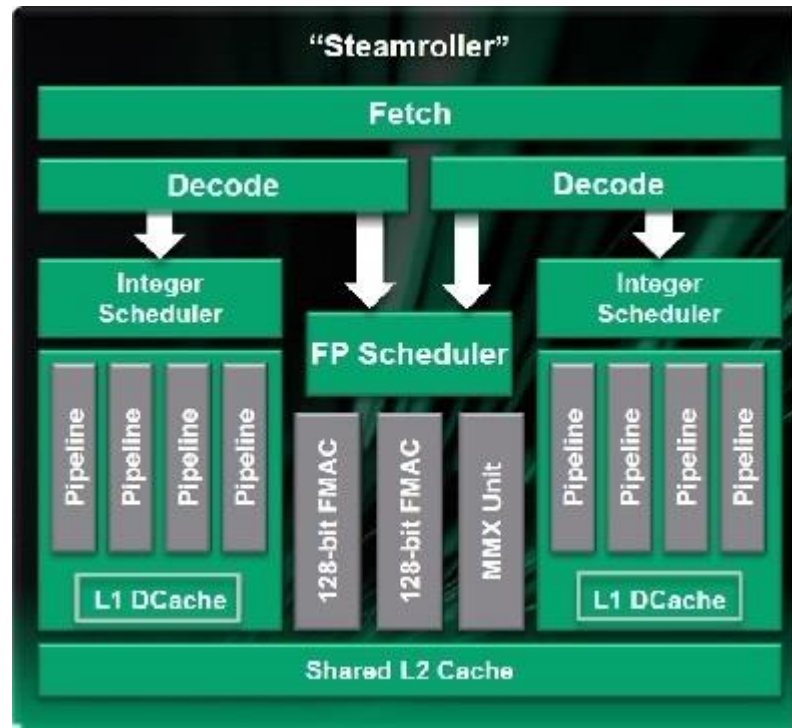
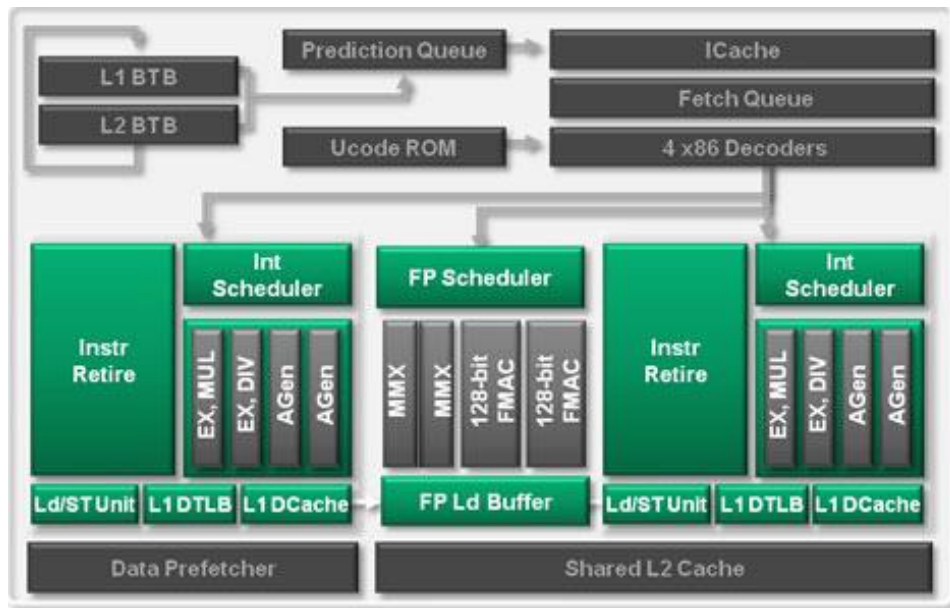
- **2, 4, 6, 8, 12, nebo 16 jader**, linka 12 stupňů, technologie SOI (Silicon on Insulator)
- **Linky Hyper Transport – 2003** (point-to-point) pro propojení s dalšími CPU (nahradily sběrnici) nebo I/O. Šířka 16 bitů, při 800 MHz to znamená 3,2 GB/s. Umožněna stavba multiprocesorů bez dalších součástek.
- **Řadič paměti DDR na čipu – 2003**, 128 bitové rozhraní na 333 MHz paměť.
- **Mikroarchitektura K10**: Phenom II, 2,3,4 nebo 6 jader, 2008–12.

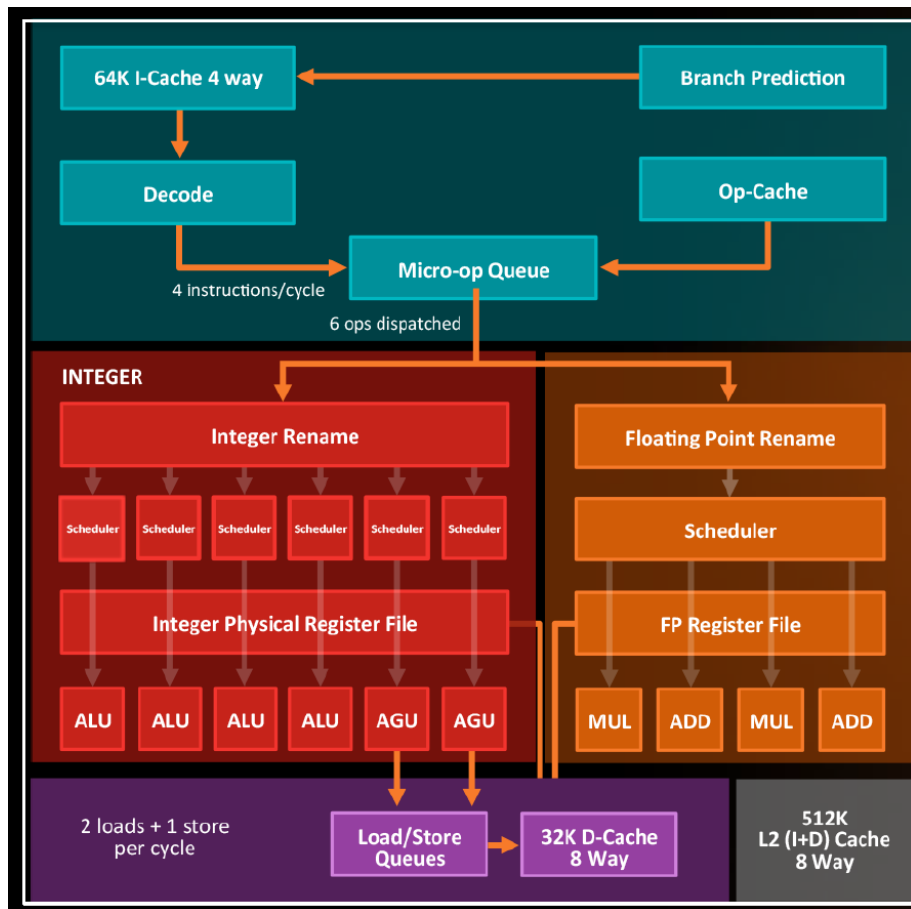


AMD Athlon™ 64 X2
Dual-Core Processor Design



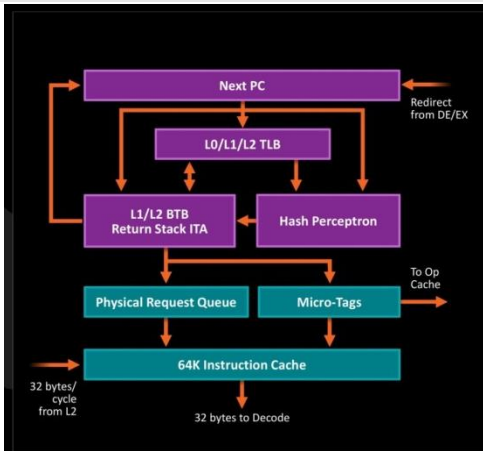
- **Mikroarchitektura Bulldozer:** 1 až 4 moduly se 2 jádry (tzv. CMT, Clustered MultiThreading, 1 až 2 vlákna/modul).
 - 10 až 100 W, 3,6–4 GHz, 2012.
 - Každý 2-jádrový modul sdílí L1-I cache, stupně načítání a dekodování, L2 cache, FPU.
 - Později přidány dedikované dekodéry (Steamroller)





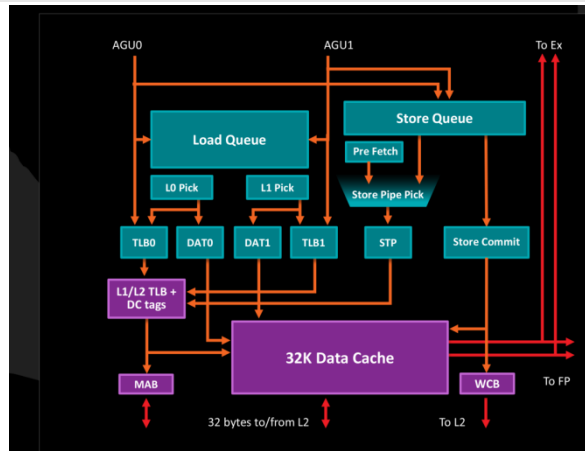
ZEN MICROARCHITECTURE

- ▲ Fetch Four x86 instructions
- ▲ Op Cache instructions
- ▲ 4 Integer units
 - Large rename space – 168 Registers
 - 192 instructions in flight/8 wide retire
- ▲ 2 Load/Store units
 - 72 Out-of-Order Loads supported
- ▲ 2 Floating Point units x 128 FMACs
 - built as 4 pipes, 2 Fadd, 2 Fmul
- ▲ I-Cache 64K, 4-way
- ▲ D-Cache 32K, 8-way
- ▲ L2 Cache 512K, 8-way
- ▲ Large shared L3 cache
- ▲ 2 threads per core



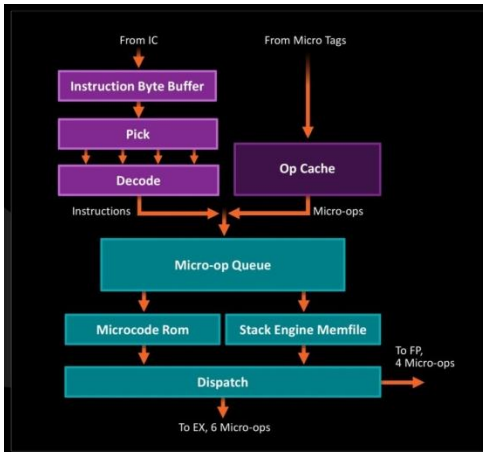
FETCH

- Decoupled Branch Prediction
- TLB in the BP pipe
 - 8 entry L0 TLB, all page sizes
 - 64 entry L1 TLB, all page sizes
 - 512 entry L2 TLB, no 1G pages
- 2 branches per BTB entry
- Large L1 / L2 BTB
- 32 entry return stack
- Indirect Target Array (ITA)
- 64K, 4-way Instruction cache
- Micro-tags for IC & Op cache
- 32 byte fetch



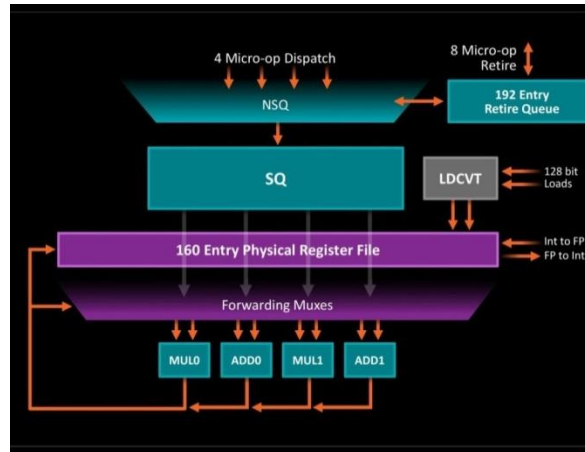
LOAD/STORE AND L2

- 72 Out of Order Loads
- 44 entry Store Queue
- Split TLB/Data Pipe, store pipe
- 64 entry L1 TLB, all page sizes
- 1.5K entry L2 TLB, no 1G pages
- 32K, 8 way Data Cache
 - Supports two 128-bit accesses
- Optimized L1 and L2 Prefetchers
- 512K, private (2 threads), inclusive L2



DECODE

- Inline Instruction-length Decoder
- Decode 4 x86 instructions
- Op cache
- Micro-op Queue
- Stack Engine
- Branch Fusion
- Memory File for Store to Load Forwarding



FLOATING POINT

- 2 Level Scheduling Queue
- 160 entry Physical Register File
- 8 Wide Retire
- 1 pipe for 1x128b store
- Accelerated Recovery on Flushes
- SSE, AVX1, AVX2, AES, SHA, and legacy mmx/x87 compliant
- 2 AES units

[illegible]

The diagram illustrates the high-level architecture of the RISC-V processor. At the top, a **32k Cache (8-Way)** and a **Branch Prediction** unit feed into the **Decode** and **Opcache** stages, respectively. These stages output to a **Micro-op Queue**, which is labeled with **6 Instructions/Cycle** and **6 Ops Dispatched**. The queue feeds into two main processing blocks: the **Integer Rename** unit and the **Floating Point Rename** unit. The **Integer Rename** unit contains eight **Sched** (scheduler) blocks and an **Integer Physical Register File** with eight **ALU** (Arithmetic Logic Unit) blocks. The **Floating Point Rename** unit contains a **Scheduler** block and an **FP Register File** with four **Mul** (Multiplier) and four **Add** (Adder) blocks. The **Integer Rename** unit is associated with **2 loads x 1 store per cycle**. The **Floating Point Rename** unit is associated with a **32k Cache (8-Way)** and a **512k L2 (1+0) Cache (8-Way)**.

The diagram illustrates the memory hierarchy for Core 0. It consists of three main components: a Core 0 block, a 512K L2 Cache, and a 16M L3 Cache.

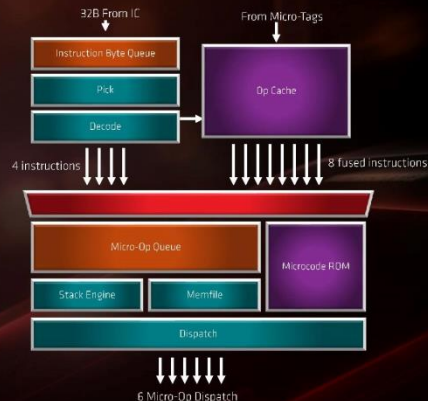
- Core 0:** Contains two 32K L1 caches (I-Cache 8-way and D-Cache 8-way) and a 512K L2 Cache.
 - The I-Cache is connected to the L2 Cache via a 32B/cycle bus.
 - The D-Cache is connected to the L2 Cache via a 32B/cycle bus.
 - The L2 Cache is connected to the L3 Cache via a 32B/cycle bus.
- 512K L2 Cache:** Contains an I-D Cache 8-way.
- 16M L3 Cache:** Contains an I-D Cache 16-way.

Data flow is indicated by arrows:

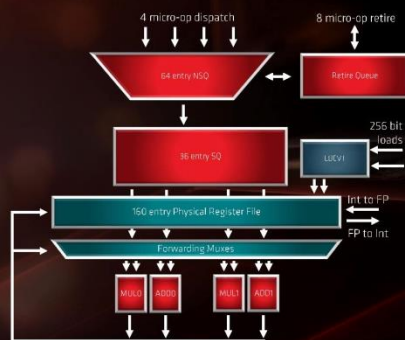
- A 32B fetch path from the L2 Cache to the I-Cache.
- A 32B/cycle path from the L2 Cache to the I-Cache.
- A 32B/cycle path from the L2 Cache to the D-Cache.
- A 32B/cycle path from the D-Cache to the L2 Cache.
- A 32B/cycle path from the L2 Cache to the L3 Cache.
- A 32B/cycle path from the L3 Cache to the L2 Cache.

A red dashed box highlights the L1 D-Cache and the L2 Cache, with labels indicating a 2*32B load and a 1*32B store path between them.

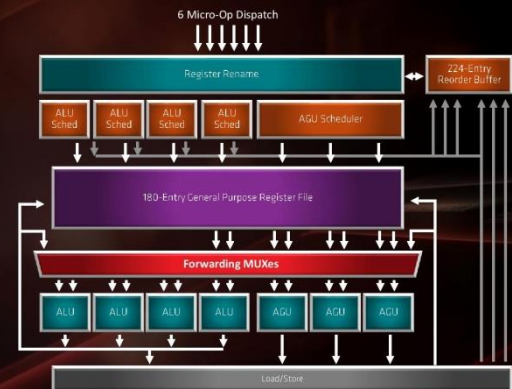
- Op cache improvements
- Doubled capacity to 4K fused instructions
- Better instruction fusion
- Increased effective throughput



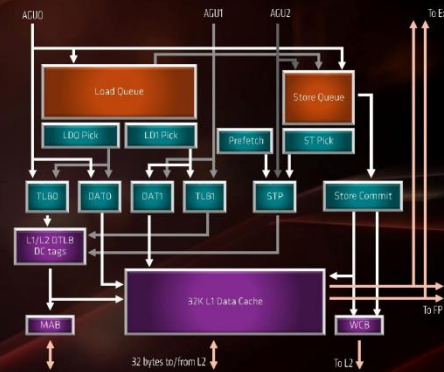
- Doubled Floating Point & Load Store bandwidth from 128b to 256b
- Improved performance for instructions using 256b ymm registers which are generated by AVX Intrinsics or /arch:[AVX|AVX2] compiler flags
 - Faster inline memory copy & memset
 - Faster physics simulation
 - Faster audio effects processing (Microsoft® XAudio2_9)
- Improved mul latency from 4 to 3 cycles



- 92 entry integer scheduler, up from 84
- 4, 16-entry ALU queues
- 1, 28-entry AGU queue
- 180 entry physical register file (up from 168)
- 7 issue per cycle, up from 6
- 4 ALUs, 3 AGUs
- 224 entry ROB, up from 192
- Improved SMT fairness for ALU and AGU schedulers
- Watermarked ALU tokens to manage spinlocks



- 48 entry store queue, was 44
- 2K entry L2 DTLB, 1G as 2M, was 1.5K no 1G
- Improved L2 DTLB latency
- 32KB, 8-way L1 data cache
 - Two 256-bit reads
 - One 256-bit write
- 64B load, 32B store alignment boundaries
- Increased Load/Store bandwidth to 32B/clk (up from 16B/clk)
- Faster string copy and float-point point performance
- Improved write-combining buffer performance
- While using multiple streams, the hardware avoids closing buffers before they are completely full
- Improved prefetch throttling



Pokračování příště