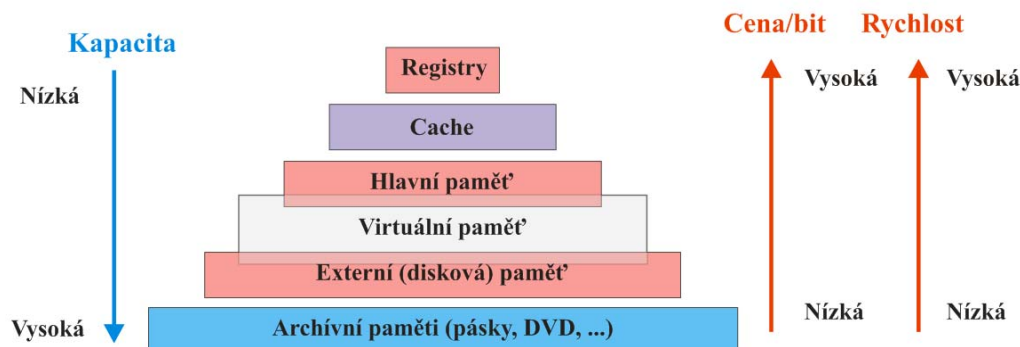


Některé pojmy k pokročilým architekturám počítačů

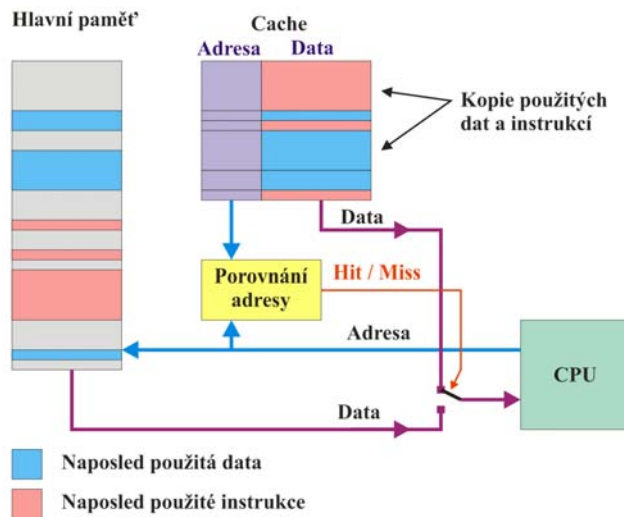
Hierarchie paměťového systému

- Obecně roste cena paměti s její rychlostí.
 - Z cenových důvodů nemůže být rychlá paměť libovolně velká.



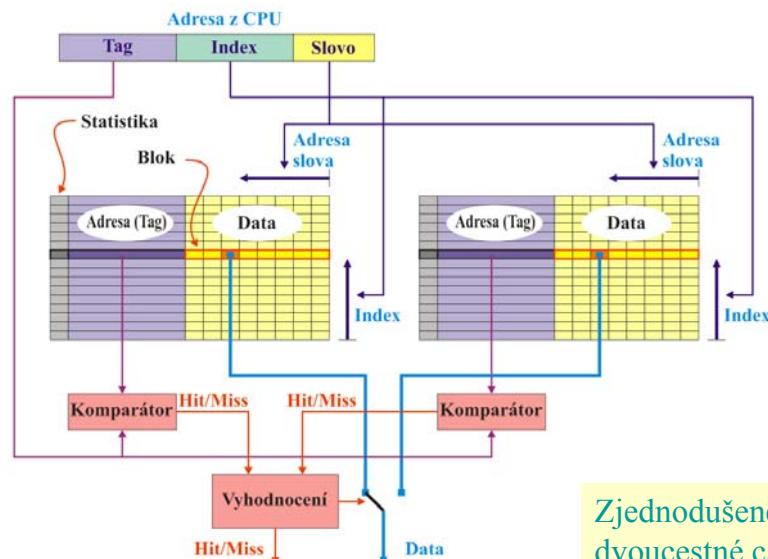
Cache

- Koncepte vychází z časové a prostorové lokality výpočtu.
 - Referovaná adresa v paměti bude **pravděpodobně** použita opakovaně.
 - **Pravděpodobně** bude použito i blízké okolí referované adresy.
- Cache obsahuje kopie naposled použitých slov paměti a okolních adres.



Cache

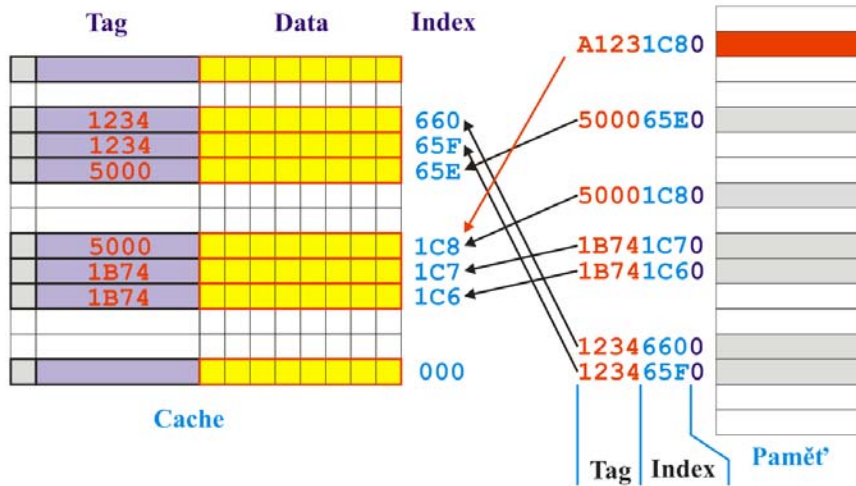
- Obvykle je cache rozdělena na několik částí (**cest**).
 - Část adresy z CPU (**index**) vybere určitou řádku ve všech cestách současně.
 - Zbytek adresy (**tag**) se porovná komparátory s adresou zapsanou u dané řádky.
 - Není-li shoda v žádném komparátoru, nahradí se celý **blok** ve zvolené cestě novými daty.



Zjednodušené schéma
dvoucestné cache

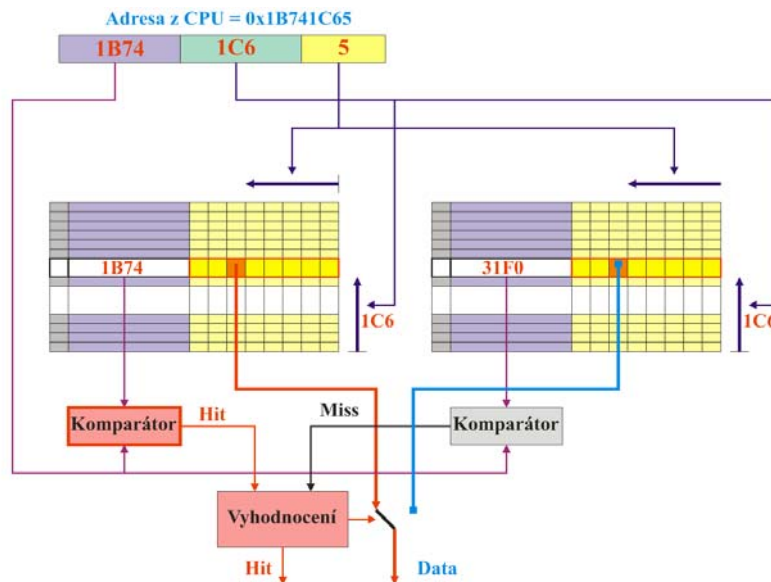
Cache

- Příklad uložení dat v jednocestné cache:
 - Čtení z adresy **A1231C80** přečte původní blok z adresy **50001C80** na indexu **1C8**.
 - Dvoucestná cache by mohla mít **A1231C80** v jedné cestě a **50001C80** ve druhé cestě.



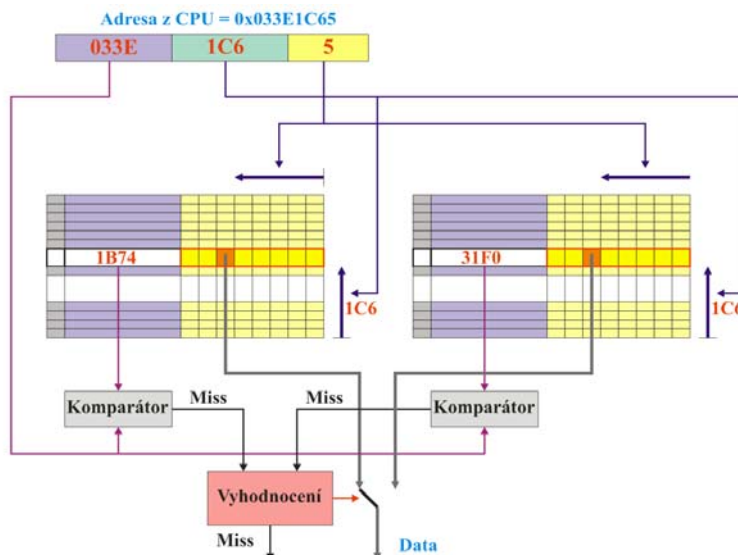
Cache hit

- Příklad: Situace „cache hit“:
 - Levá cesta má na indexu 1C6 stejný tag jako je v adrese vystavené procesorem.



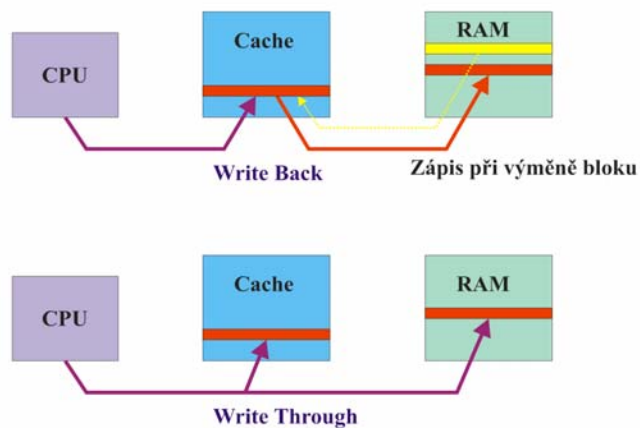
Cache miss

- Příklad: Situace „cache miss“:
 - Elektronika cache vybere některou z cest a nahradí data na indexu 1C6 daty z adresy 0x033E1C65.



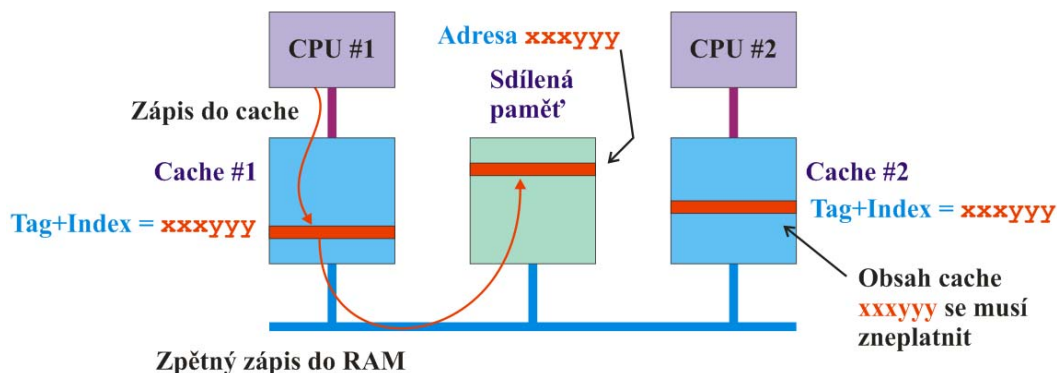
Zápis do cache

- Write Back
 - Data se zapisují jen do cache.
 - Do RAM se zapíší až při výměně bloku za jiný.
- Write Through
 - Data se zapisují do cache i do RAM.



Problému koherence

- Oba procesory mohou mít v cache kopii stejné adresy.
- Při změně obsahu jedním z procesorů se musí kopie ve druhé cache zneplatnit.
 - Alternativní řešení: sdílená oblast paměti se nesmí ukládat do cache.

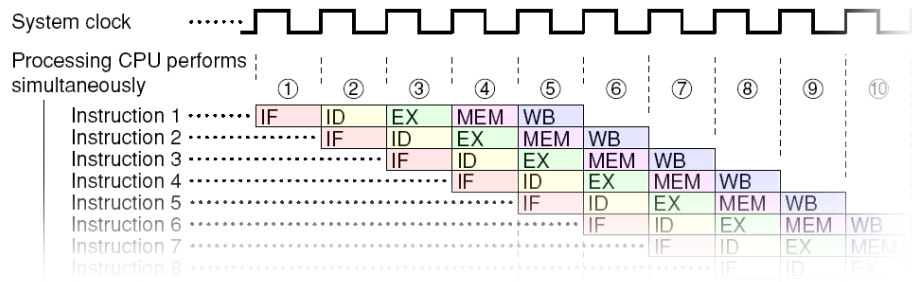


Paralelní výpočty

- „Klasické“ rozdělení výpočtů podle způsobu provádění:
 - **SISD** (Single Instruction, Single Data) – výpočet provádí jeden procesor jednoduchou operační jednotkou.
 - **SIMD** (Single Instruction, Multiple Data) – výpočet provádí jeden procesor, několik stejných operačních jednotek provádí tutěž instrukci (operaci) s různými daty.
 - **MIMD** (Multiple Instruction, Multiple Data) – výpočet provádí několik procesorů, každý podle svého programu se svými daty.
 - **Data Flow**: instrukce nejsou v programu zapsány sekvenčně. Je dána množina požadovaných operací. Operace se provede, jsou-li k dispozici potřebná data.

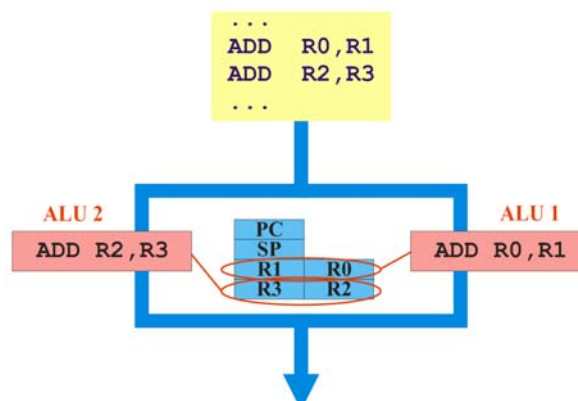
Proudové zpracování (pipeline)

- Každá instrukce se postupně zpracovává v několika stupních.
- Kompletní zpracování instrukce vyžaduje několik taktů CLK.
- V procesoru je současně v různých fázích zpracování několik instrukcí.
- V každém taktu dokončí procesor jednu instrukci.
- Problémy:
 - přístup k datům v paměti vyžaduje větší počet taktů,
 - instrukce skoků a podmíněných skoků narušují pipeline.



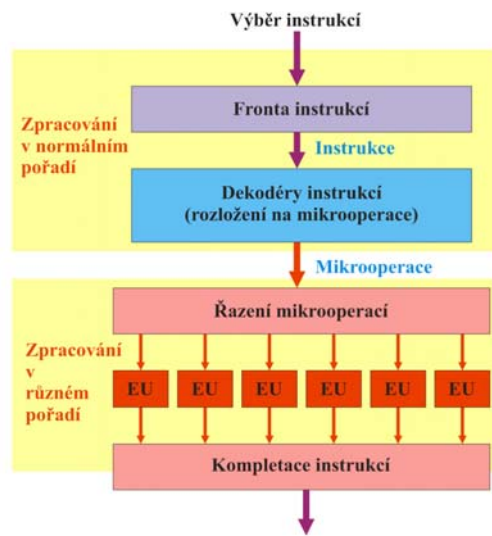
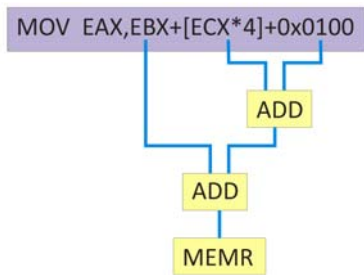
Superskalární procesor

- Procesor provádí několik instrukcí současně.
 - Procesor má ve frontě několik instrukcí.
 - Možnosti paralelního provedení rozpozná procesor.
- Problémy s různě dlouhými instrukcemi.
- Instrukce musí pracovat s nezávislými operandy.
- Vhodným pořadím instrukcí v programu ([optimalizujícím překladačem](#)) lze podpořit jejich paralelní provádění.



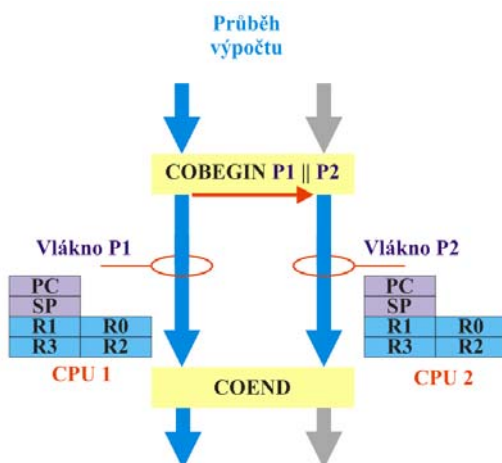
Paralelismus na bázi mikrooperací

- Provádění mikrooperací „v přeházeném pořadí“ („out of order“).
 - Několik instrukcí je paralelně rozloženo na mikrooperace.
 - Několik operačních jednotek (EU) provádí nezávislé mikrooperace v libovolném pořadí, pokud jsou k dispozici příslušné operandy.



Podpora více vláken

- Každé vlákno provádí procesor s vlastní sadou registrů.
- Paralelně prováděné části programu (**vlákna** nebo **procesy**) určí programátor resp. operační systém.
 - vlákna (**thread**) mají společný adresní prostor,
 - procesy (**process**) mají oddělené adresní prostory.

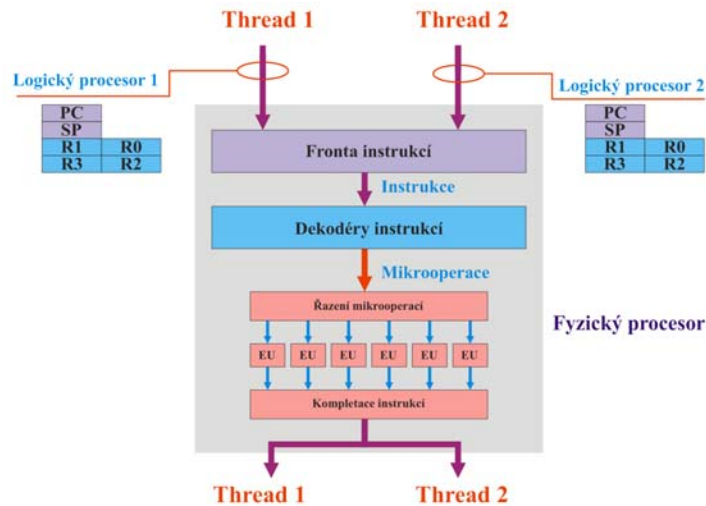


COBEGIN v programu vyvolá službu OS pro spuštění dalšího procesoru.



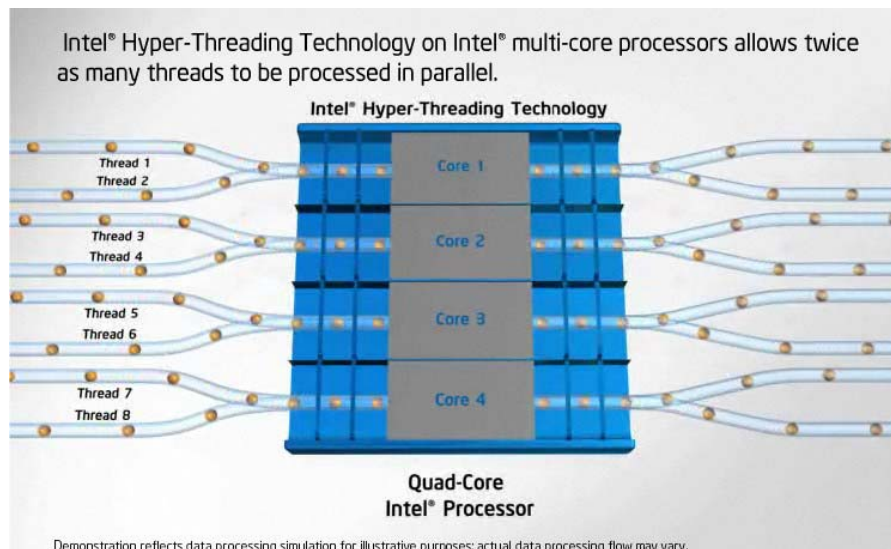
SMT (Simultaneous MultiThreading)

- Dvě (nebo více) vláken sdílí společné Execution Units.
 - Výhody: větší pravděpodobnost nezávislých mikrooperací \Rightarrow lepší využití EU.
- Každé vlákno pracuje se svým nezávislým logickým procesorem.
- Skutečné pracovní registry jsou součástí fyzického procesoru.



SMT: HyperThreading (Intel®)

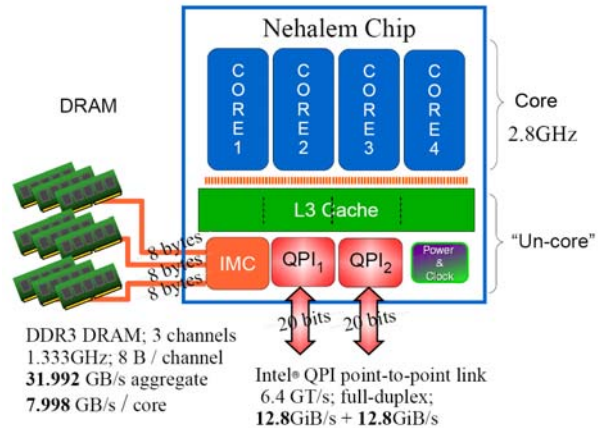
- Každé jádro je schopné provádět 2 thready.
- Oba procesory mají společnou EU (Execution Unit).
 - **!!** Paralelní běh (Thread 1) || (Thread 2) je jiný než např. (Thread 1) || (Thread 3).



Některé charakteristiky procesorů Core i7 (1)



- Procesory obsahují několik jader.
- Každé jádro může paralelně provádět 2 vlákna.
- Jádra mají společnou cache L3.
- Na čipu je trojnásobný kanál pro připojení DRAM.
 - Účinnost závisí na rozložení procesů do jednotlivých bloků paměti.
- Pro komunikaci s jinými procesory nebo se vzdálenými DRAM slouží kanály (sběrnice) QPI (**Q**uick **P**ath **I**nterconnect).



Obrázky: Intel,
Michael E. Thomadakis, Ph.D., Texas A&M University.

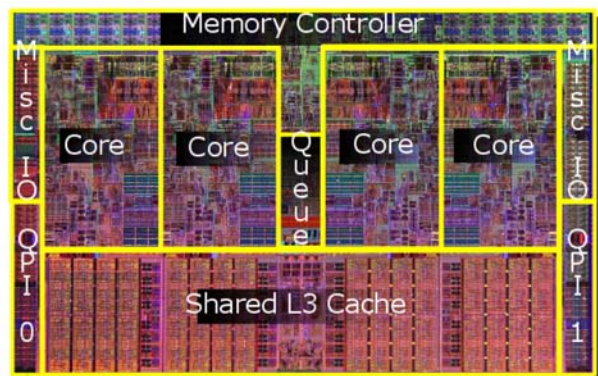
17

K.D. - přednášky POT

Některé charakteristiky procesorů Core i7 (2)



- Procesory obsahují několik jader.
- Každé jádro může paralelně provádět 2 vlákna.
- Jádra mají společnou cache L3.
- Na čipu je trojnásobný kanál pro připojení DRAM.
 - Účinnost závisí na rozložení procesů do jednotlivých bloků paměti.
- Pro komunikaci s jinými procesory nebo se vzdálenými DRAM slouží kanály (sběrnice) QPI (**Q**uick **P**ath **I**nterconnect).



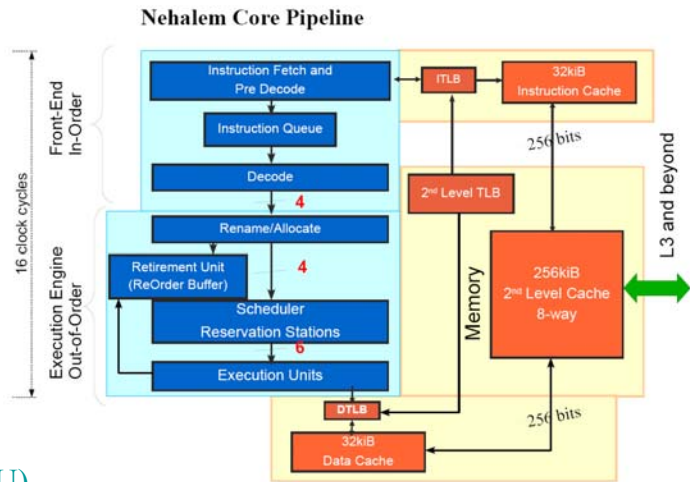
18

K.D. - přednášky POT

Některé charakteristiky procesorů Core i7 (3)

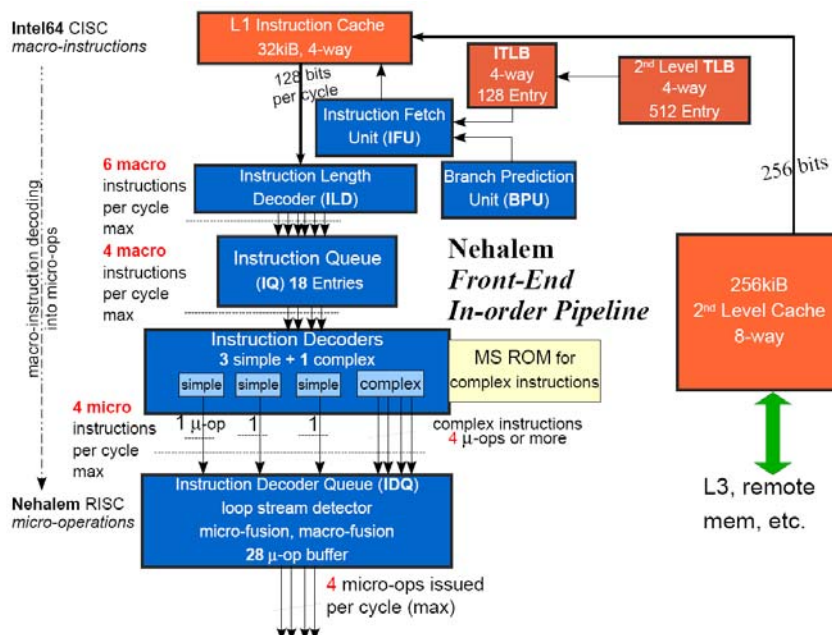
V každém jádru prochází instrukce postupně několika stupni:

- V pořadí daném programem se provádí:
 - Čtení instrukcí a řazení do fronty.
 - Dekódování instrukcí.
 - Do fronty instrukcí se současně vkládají instrukce ze 2 threadů.
- Provádění instrukcí v různém pořadí („out of order“).
 - Několik instrukcí je paralelně rozloženo na mikrooperace.
 - Několik operačních jednotek (EU) provádí nezávislé mikrooperace v libovolném pořadí, pokud jsou k dispozici příslušné operandy.



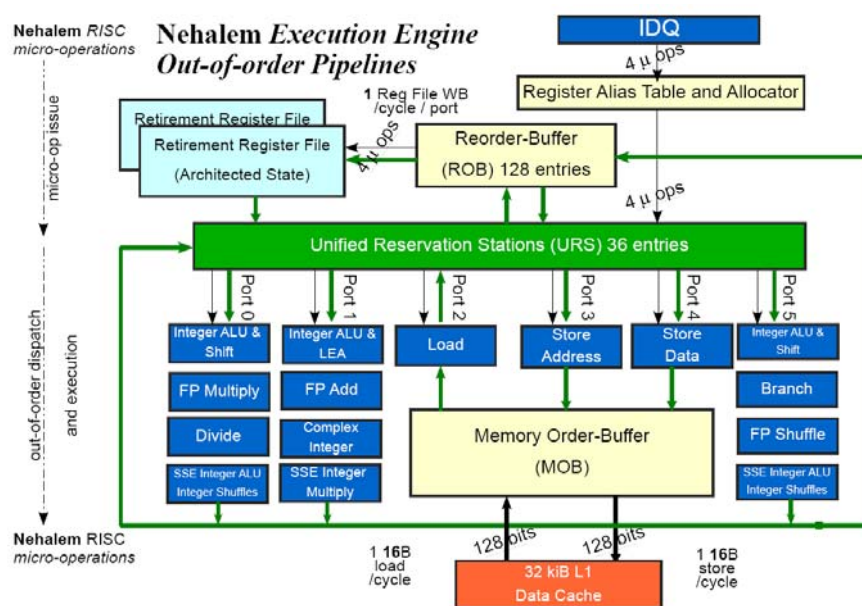
Některé charakteristiky procesorů Core i7 (4)

- Podrobnosti stupňů pro výběr a dekódování instrukcí.



Některé charakteristiky procesorů Core i7 (5)

- Podrobnosti EU pro provádění instrukcí „out of order“:



K.D. - přednášky POT

21

Predikce skoků

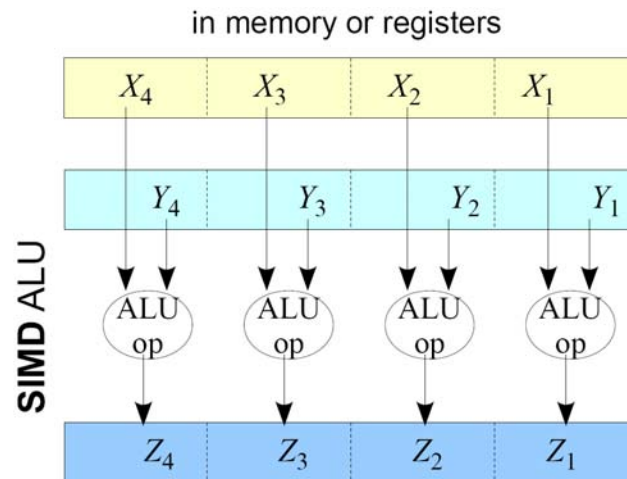
- Predikce skoků
 - Maximální výkon architektury i7 je dosažen při plynulém čtení instrukcí z paměti.
 - Podmíněné skoky mohou podle výsledku testu vést k nutnosti zrušit část obsahu fronty instrukcí atd.
 - Predikce skoků se snaží podle adresy a cíle skokové instrukce a podle historie predikovat, zda se skok provede nebo ne.
- Detekce krátkých cyklů
 - Krátké cykly jsou detekovány na úrovni mikrooperací. Instrukce v cyklu nejsou znovu čteny a dekodovány.

K.D. - přednášky POT

22

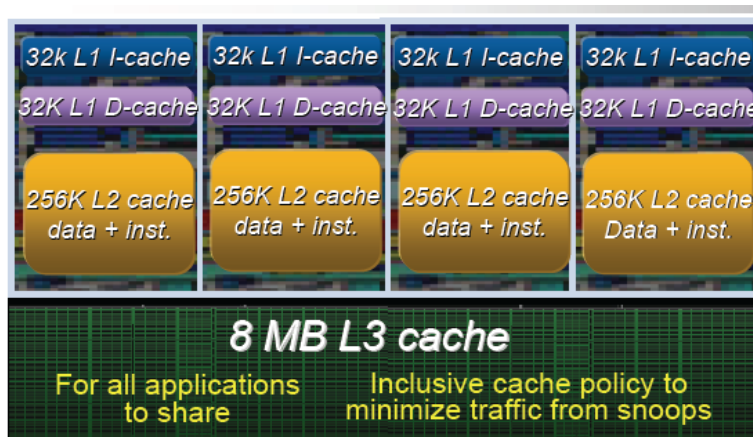
Některé charakteristiky procesorů Core i7 (6)

- Instrukce SIMD (Single Instruction Multiple Data):
 - Stejná instrukce se provádí s několika daty.
 - Vhodné pro vektorové a maticové operace, grafické aplikace, zpracování signálů, ...



Některé charakteristiky procesorů Core i7 (7)

- Architektura paměti cache:
 - **L1** cache: oddělená datová a kódová cache, samostatná pro každé jádro.
 - **L2** cache: společná pro kód a data, samostatná pro každé jádro.
 - **L3** cache: společná pro kód a data, společná pro všechna jádra.



Některé charakteristiky procesorů Core i7 (8)

- Porovnání rychlostí různých datových cest a cache:

