B35APO: Architektury počítačů

Lekce 04. Hierarchie paměti

Pavel Píša
pisa@fel.cvut.cz

Petr Štěpán
stepan@fel.cvut.cz

19. dubna, 2023

Obsah

Motivace

Algoritmus A

```
int matrix[N][N];
int main() {
  long int i, j, sum1 = 0;
  for(i=0; i<N; i++)
    for(j=0; j<N; j++)
      sum1 += matrix[i][j];
}
```

Algoritmus B

```
int matrix[N][N];
int main() {
  long int i, j, sum1 = 0;
  for(i=0; i<N; i++)
    for(j=0; j<N; j++)
      sum1 += matrix[j][i];
}
```

Oba dva programy vypadají velmi podobně.
Program A prochází pole po řádcích, program B prochází pole po sloupcích.

Bude se nějak lišit doba výpočtu?

Motivace

Algoritmus A

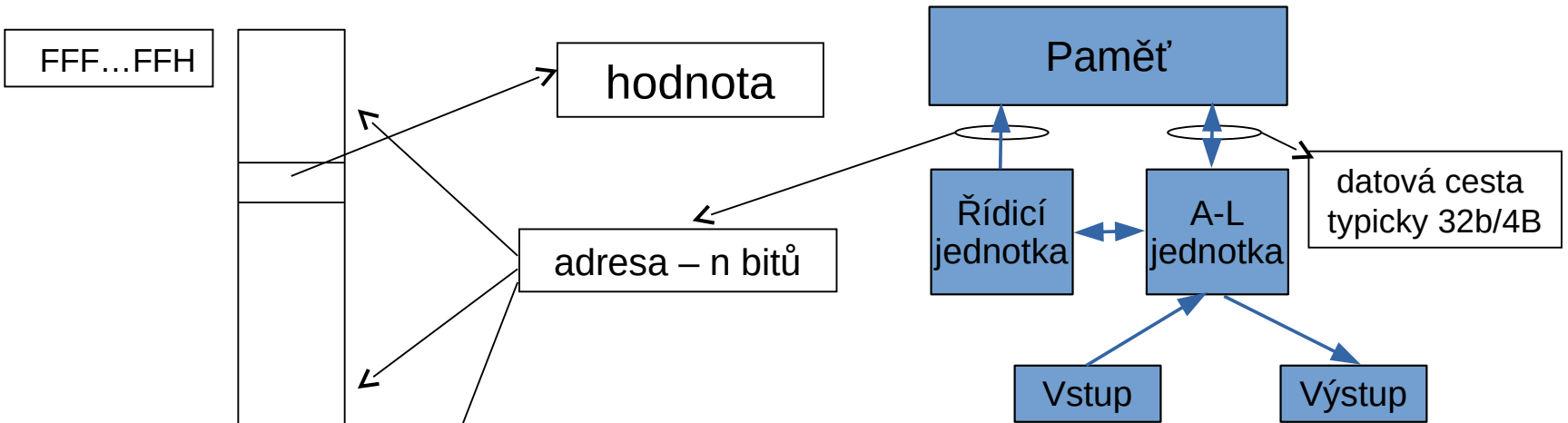
```
int matrix[N][N];
int main() {
  long int i, j, sum1 = 0;
  for(i=0; i<N; i++)
    for(j=0; j<N; j++)
      sum1 += matrix[i][j];
}
```

Algoritmus B

```
int matrix[N][N];
int main() {
  long int i, j, sum1 = 0;
  for(i=0; i<N; i++)
    for(j=0; j<N; j++)
      sum1 += matrix[j][i];
}
```

N	A	B
100000	12.791328s	138.047563s
10000	0.126945s	0.486535s
1000	0.001329s	0.001756s
100	0.000083s	0.000094s

Co je paměť



velikost paměťové buňky typicky 1B

n	2^n
8	256 různých buněk
16	64K (K=1024)
…	……
32	4G (4096M, $M=K^2$)

Co je paměť – popis

Paměť :

- je pole adresovatelných buněk
  - buňky mají shodnou šíři - většinou 8 bitů (byte) nebo jejich násobek
- velikost fyzického adresového prostoru je omezena počtem bitů adresy, kterou je CPU schopno nastavit na paměťové sběrnici, tedy kolik bitů lze použít k adresování paměti. Pro organizaci po bytech pak
  - 16 bitů adresy umožňuje maximálně 64KiB velkou paměť
  - 32 bitů adresy umožňuje maximálně 4GiB velkou paměť
  - 37 bitů adresy (maximum procesoru Intel Core i9-13900K, pak záleží ještě na základní desce) maximálně 128 GiB velkou paměť
- typ přístupu:
  - k vnitřní paměti počítače lze přistupvat v náhodném pořadí (= random access – RAM)
  - některé externí paměti (např. magnetopásková paměť využívaná pro zálohování – levnější než HDD a SSD) umožňjí pouze sekvenční přístup, tedy čtení a zápis bajtů po sobě
  - pro HDD, SSD, Flash je možný přítup po celých blocích, u SSD/Flash je zápis bloku možný vždy jednou do smazání po velkých celcích

Polovodičových paměti – terminologie

Adresa vstup, který vybírá svojí hodnotou jednu konkrétní buňku paměti

Hodnota vlastní informace

stavová data volitelně další informace (třeba o platnosti hodnoty, redundanci pro opravení chyb, apod.)

Parametry pamětí:

Vybavovací doba paměti (latence) kritický parametr. Délka časového intervalu mezi objevením se požadavku a okamžikem, kdy jsou data k dispozici.

Doba přístupu vybavovací doba + obnovení obsahu po destruktivním čtení připadně doba, kdy lze zadat další požadavek

Propustnost výkonový parametr. Schopnost zpracovat uvedené množství za jednotku času.

Obsah

Druhy paměti

Paměti dělíme podle možností zápisu na:

- **ROM** (Read-Only Memory) paměti ze kterých lze pouze číst, řadíme sem i paměti typu EEPROM (Electrically Erasable Programmable Read-Only Memory), které lze omezeně smazat vysokým napětím a pak tam zapsat nové údaje, při normálním provozu nejde do paměti zapsat.
- **RAM** (Random Access Memory) také RWM (Read-Write Memory) klasické paměti určené pro čtení a zápis libovolné buňky v libovolném pořadí – Random Access.

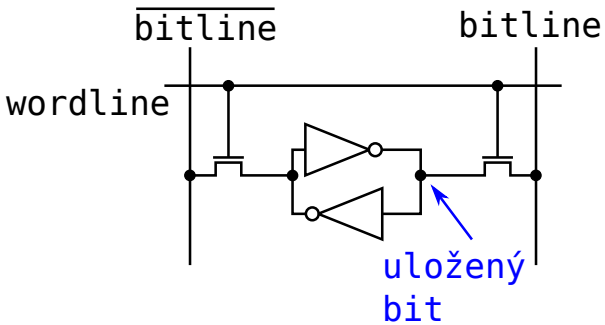
Paměti lze dělit i podle toho, zda data jsou uchována po vypnutí napájení:

- **Permanentní** (Non-volatile) paměť nepotřebuje k udržení informace napájení - např. Flash, EEPROM, EPROM, ROM, feromagnetické paměti, HDD, SSD, 3D-X Point – Intel Optane Memory
- **Volatilní** (Volatile) například paměť DDR SDRAM, SRAM, cache v klasickém počítači, k udržení informace je potřeba nepřetržité napájení a obnova dat.

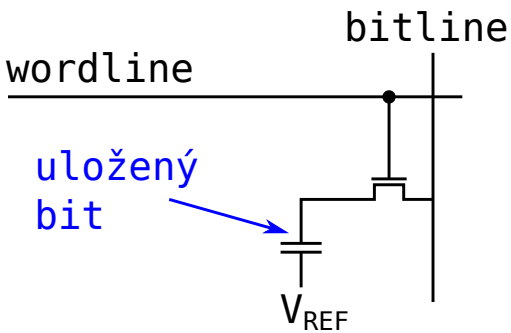
Konstrukce paměti

Podle kontrukce dělíme paměti RAM (RWM) na:

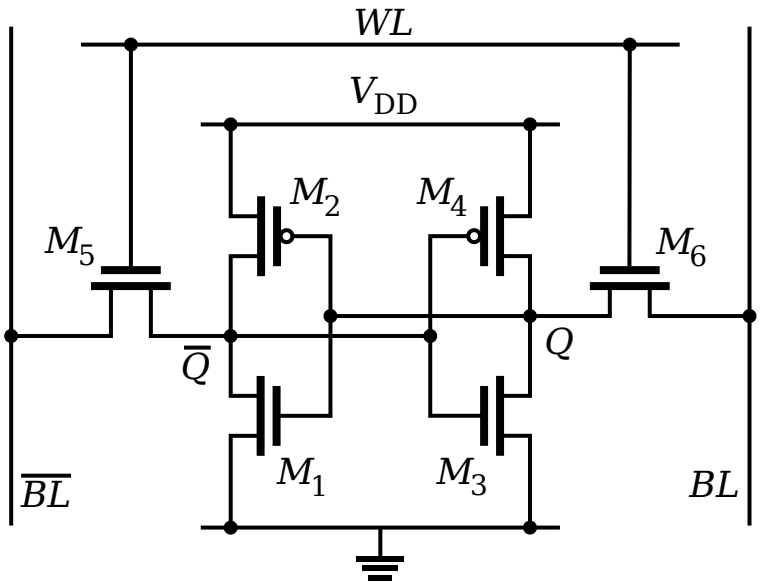
SRAM Statická RAM – veličina reprezentující hodnotu má v čase konstatní hodnotu. Typicky dva do smyčky zapojené invertory. Více součástek dražší paměť, nemusí se obnovovat uložená informace, potřebuje pouze napájení

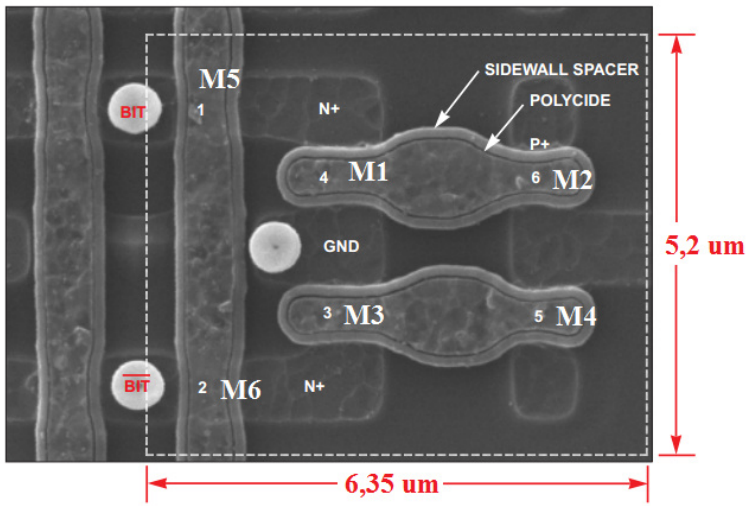


DRAM Dynamická RAM – veličina mění v čase svoji hodnotu. Typicky kondezátor, který se samovolně vybíjí a je proto nutné informaci obnovovat v pravidelných časech. Pouze kondenzátor a tranzistor – levné, zabere méně prostoru.



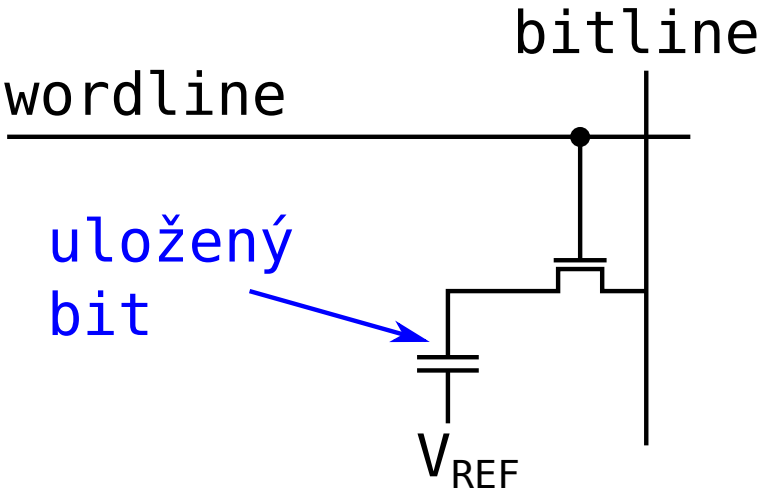
Detail paměťové statické paměti – SRAM

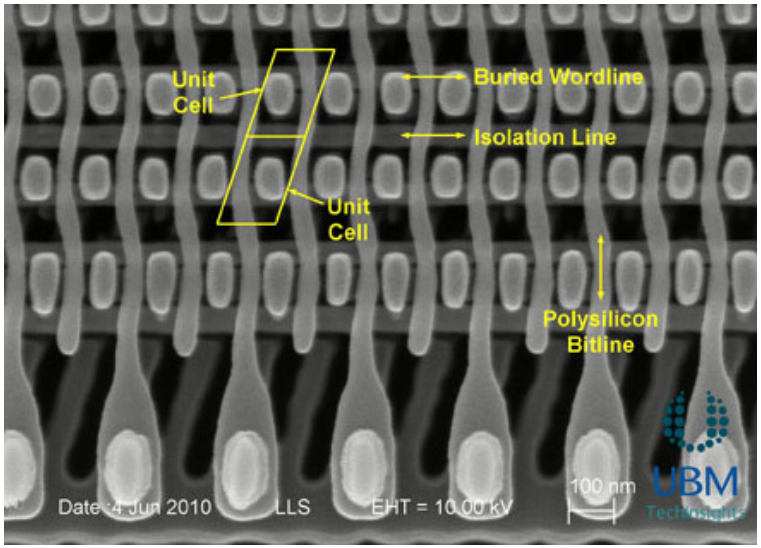




- Plně statické zapojení (kladná zpětná vazba)
- Pro udržení informace vyžaduje napájení (volatilní paměť)
- Nevýhoda, potřebuje 6 tranzistorů, velká plocha
- Čtení, po výběru slova (world line – WL) jsou data přenesena na příslušné nenapájené dráhy (bit line – BL)
- Při zápisu přes připojené bitové dráhy na log. 1 a log. 0 dojde k vnucení stavu vetším proudem než zajišťuje vnitřní smyčka

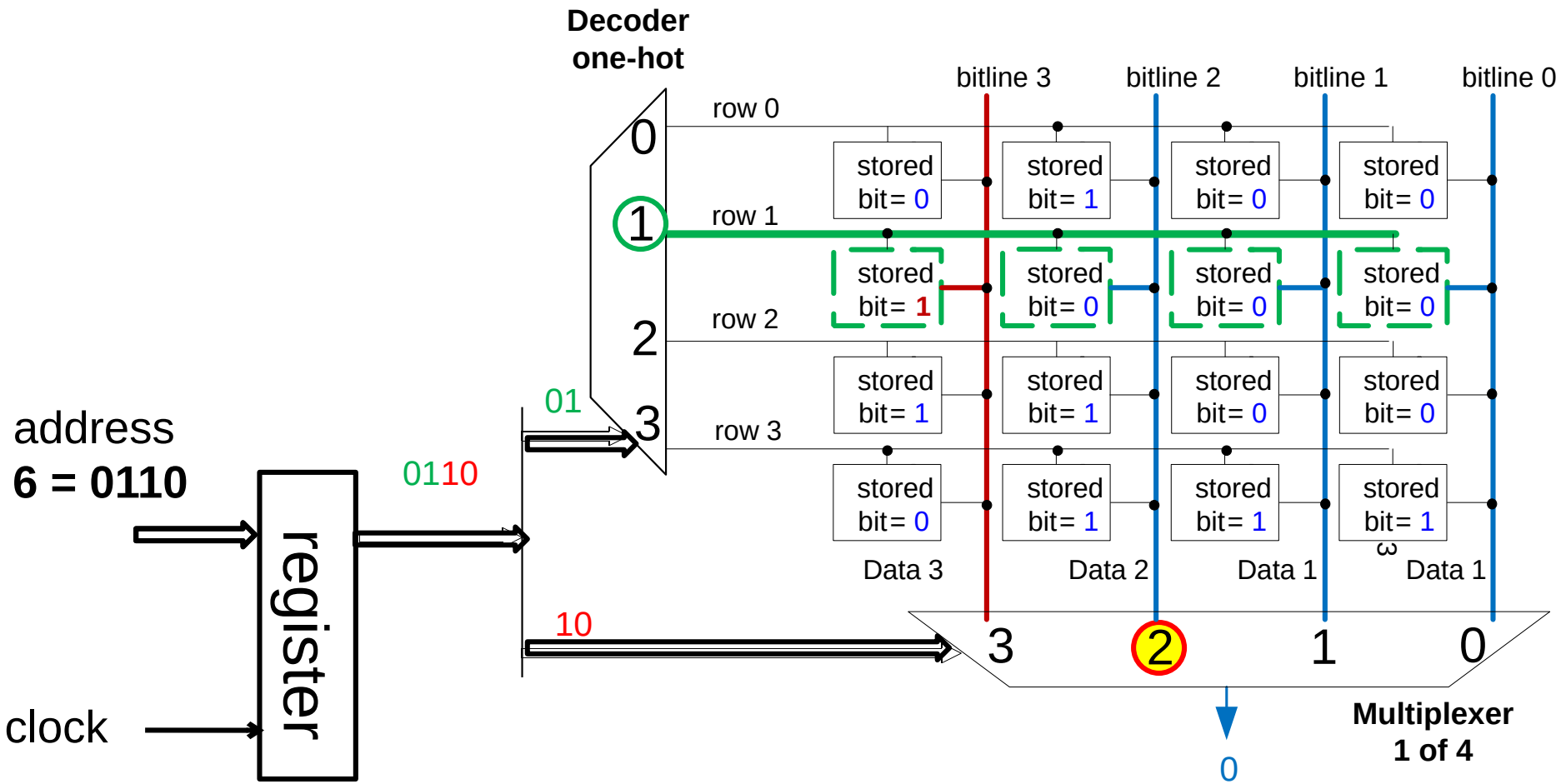
Detail paměťové buňky dynamické paměti – DRAM



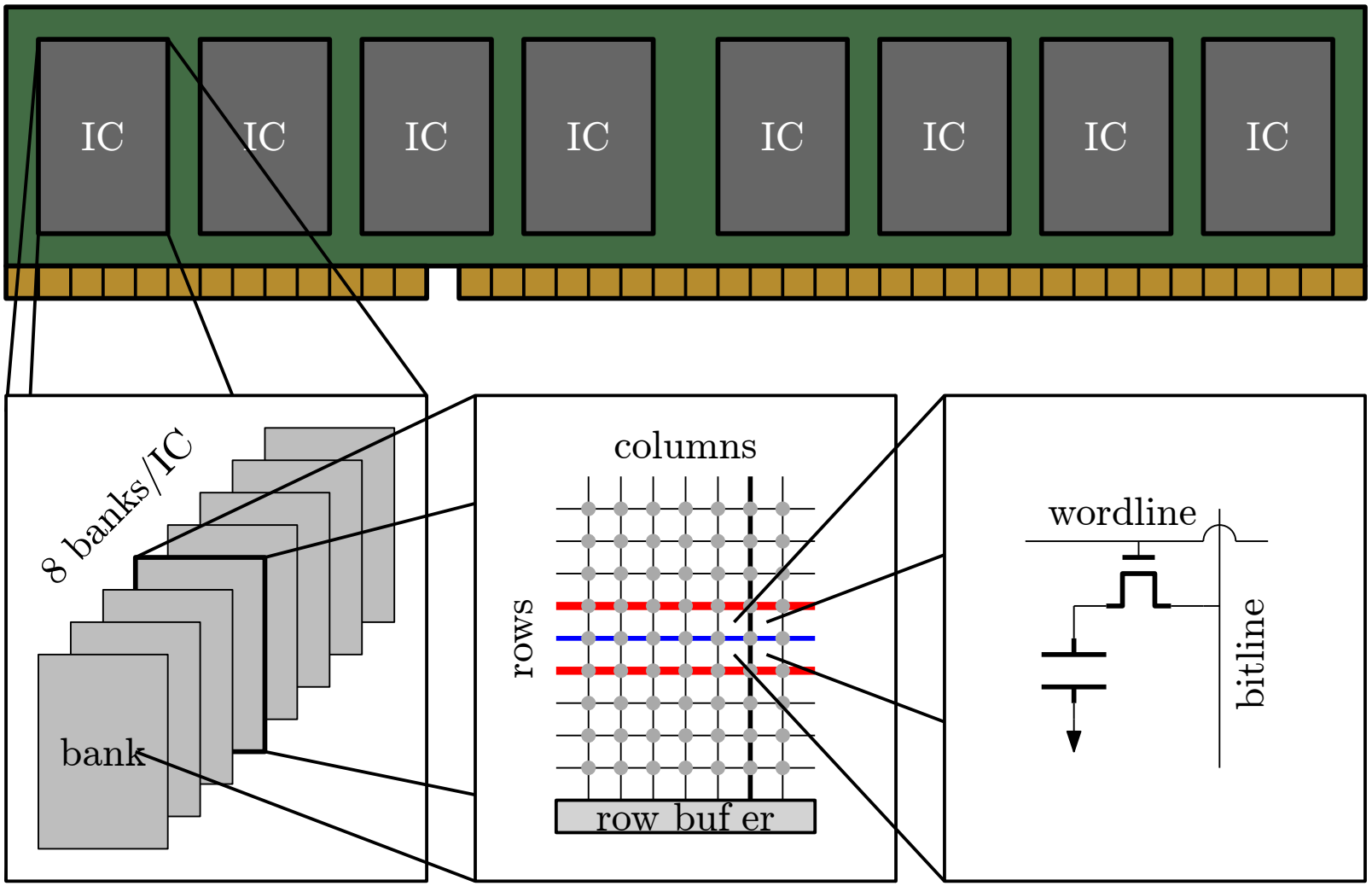


- nMOS tranzistor představuje přepínač, který připojí (nebo ne) kondenzátor na vodič „bitline“. Připojení je řízeno vodičem „wordline“.
- Proces čtení kondenzátor vybíjí. Proto musí být poté obnoven.
- Občerstvování pamětí (refresh) – náboj se z kondenzátoru samovolně ztrácí. Nezbytná pracovní fáze dynamické paměti. Negativně ovlivňuje (prodlužuje) průměrnou vybavovací dobu.

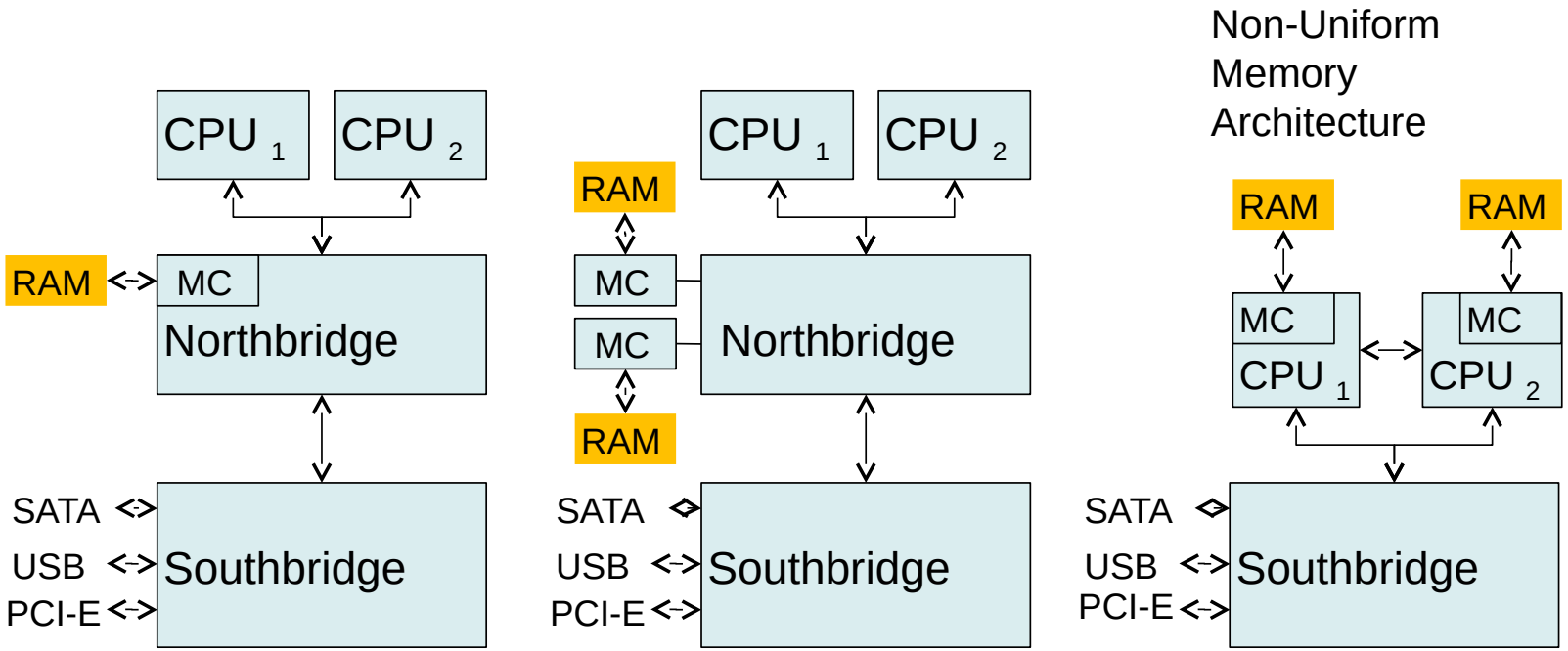
Paměťová matice – základ



Moduly s dynamickou pamětí



Paměť v dnešních osobních počítačích



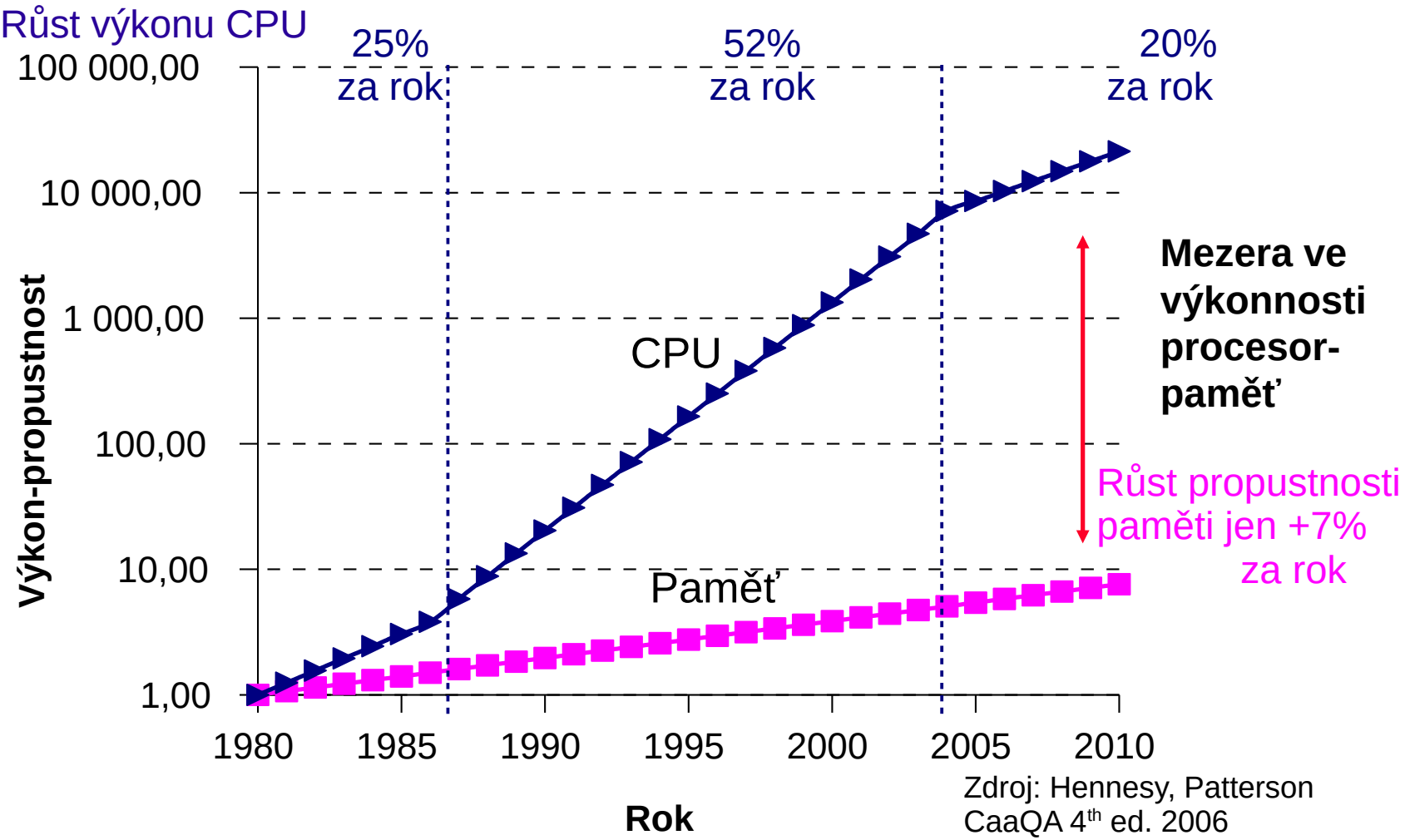
MC - Memory controller – paměťový kontrolér zajišťuje řízení čtení a zápisů do SDRAM. Dále obstarává obnovu informace (refreshing) každé paměťové buňky jednou za 64 ms.

Nejčastěji používané typy dynamických pamětí

- **SDRAM** – hodinová frekvence až 100 MHz, 2.5V, synchronní přenos dat na hodinové hraně
- **DDR SDRAM** – přenos dat na obou hodinových hranách, 2.5V, V/V hodiny až 100-200 MHz, 0.2-0.4 GT/s (miliard přenosů za skundu)
- **DDR2 SDRAM** – menší spotřeba, 1.8V, frekvence až 400 MHz, 0.8 GT/s
- **DDR3 SDRAM** – ještě nižší spotřeba při 1.5V, frekvence až 800 MHz, 1.6 GT/s
- **DDR4 SDRAM** – 1.05 – 1.2V, V/V svěrnicové hodiny 1.2 GHz, 2.4 GT/s
- **DDR5 SDRAM** – 1.1V, až 6.4 GT/s

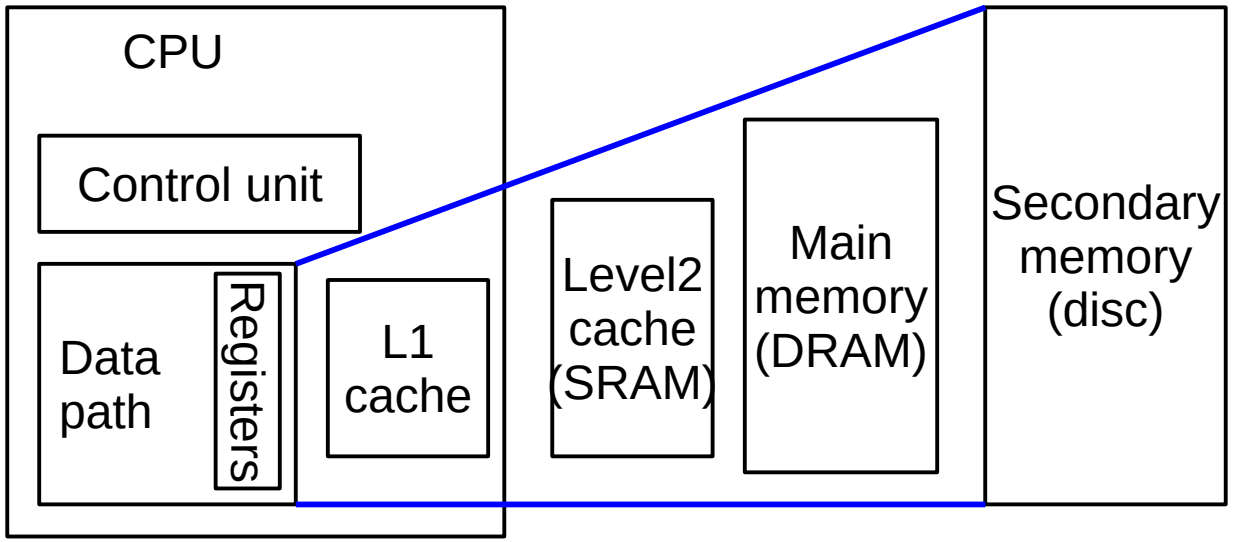
Všechny tyto druhy jsou převážně optimalizované na propustnost, nikoliv náhodný přístup, latence 20 až 35 ns.

Rozdíl růstu výkonu procesorů a pěmětí



Obsah

Paměťová hierarchie od registrů k SSD



Typ	L1 SRAM	Sync SRAM	DDR3	HDD
Velikost	32kB	1 MB	16 GB	3TB
Cena	10 kč/kB	300 kč/MB	123 kč/Gb	1 kč/GB
Rychlost	0.2...2ns	0.5...8 ns/word	15 GB/sec	100 MB/sec

Některá data mohou existovat ve více kopiích (úrovně, SMP). Pro modifikaci dat je potřeba mechanizmů pro udržení koherence slov a konzistence datových struktur.

Paměťová hierarchie – základní principy

- Programy/procesy přistupují v daném okamžiku jen k malé části svého adresového prostoru
- **Časová lokalita**
  - Položky, ke kterým se přistupovalo nedávno, budou zapotřebí brzy znovu.
  - Příklad: programová smyčka, proměnné instrukcí.
- **Prostorová lokalita**
  - Položky poblíž právě používaným budou brzy zapotřebí také.
  - Příklad: sekvenční přístup ke kódu (paměť programu), datová pole (paměť dat).

Princip se využívá jak v algoritmech (lokální proměnné), kompilátorech (přesun do registrů), na úrovni pamětí (automaticky), operačního systému (přesun z disku do paměti) případě opět programově, čtení a zápisy do souborů.

Skryté paměti – cache

- je označení pro vyrovnávací paměť používanou ve výpočetní technice
- Zařazujeme ji mezi dva subsystémy s různou rychlostí. Vyrovnává se jí rychlost přístupu k informacím.
- Účelem skryté paměti je urychlit přístup k často používaným datům na „pomalých“ médiích jejich překopírováním na média rychlá.
- Většinou z velké části pracuje automaticky a přispůsobuje se aktuální potřebě programu.
- Přesto je nutné o její existenci vědět a často potřebuje servisní obsluhu na úrovni operačního systému a v některých případech i programů.
- Při nepromyšlené volbě datových struktur a algoritmů se její efekt ztrácí.

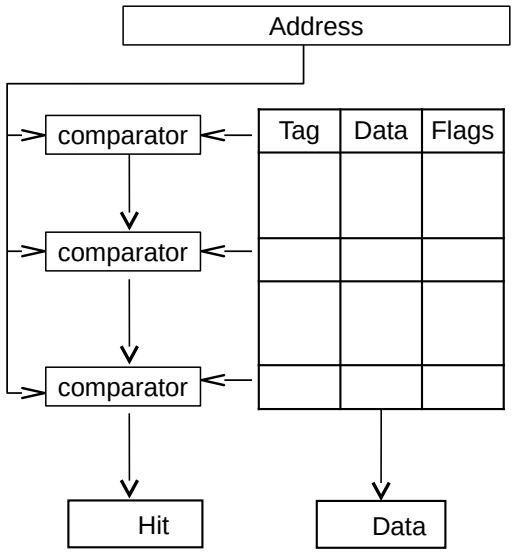
Skryté paměti – terminologie

- **Cache hit** (zásah) pojmenování situace, kdy požadovaná hodnota ve skryté paměti (cache) je.
- **Cache miss**, opak, **výpadek**, data v cache ještě nejsou.
- **Cache line** (řádka) nebo **Cache block** – základní kopírovatelná jednotka mezi hierarchickými úrovněmi.
- V praxi se velikost **Cache** Line pohybuje od 8B do 1KB, typicky 64B.
- **Hit rate** – počet paměťových přístupů obsloužených danou úrovní cache dělený všemi přístupy
- **Mis rate** – poměr přístupů, které je potřeba obsloužit z pomalejší paměti = 1 - Hit rate
- **Average Memory Access Time** (AMAT)

$$AMAT = HitTime + MissRate \times MissPenalty$$

- AMAT pro víceúrovňovou cache lze spočítat rekurzivní aplikací výšeuvedeného vztahu

Skryté paměti – realizace



Cache line of fully associative cache

Tag	Data	Flags

- **Tag** je index odpovídajícího bloku v operační paměti (v podstatě se jedná o hodnotu ukazatele/adresy dělenou délkou bloku).
- **Data** pole obsahující vlastní hodnoty na příslušné/ných adrese/ách.
- **Validity bit** – bit platnosti. Indikuje, zda je obsah pole Data vůbec platný.
- **Dirty bit** – rozšiřující pole v obsahu paměti. Indikuje, že v cache (cache) je jiná hodnota, než v paměti hlavní.

Zpracování výpadku skryté paměti, cache miss

- Data musí být nahrána z hlavní paměti, obvykle jsou již ale všechny položky vyrovnávací paměťi (cache) zaplněné daty z přetrvávajícími předchozího běhu programu.
- Některou z položek, kterou je možné využít pro uložení dat z dané adresy je potřeba uvolnit.
- Na výběru položky k vyřazení velmi záleží, pokud bude vybraná ta, která bude opětovně potřeba, dojde k snížení výkonu.
- **Cache replacement policy** – pravidla pro výběr položky k vyřazení
  - **Random** – vybraná je náhodná položka
  - **LRU** (Least Recently Used) – vybrána je nejdelší dobu neoužitá položka, do obvodů cache musí být ke každé skupině bloků přidané další informace, které umožňují sledovat pořadí posledních přístupů k jednotlivým položkám.
  - **LFU** (Least Frequently Used) – sleduje se, jak často/kolikrát se k položkám přistupuje, vyžaduje přidat zapomínání.
  - **ARC** (Adaptive Replacement Cache) – kombinace LRU a LFU

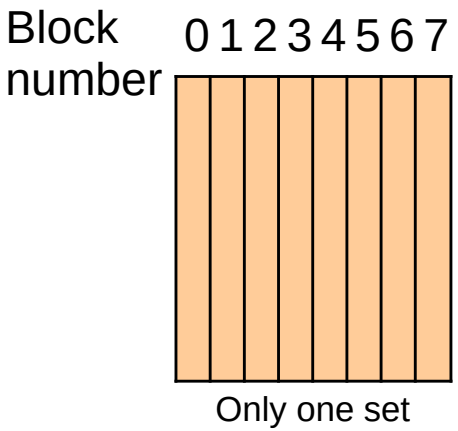
Zápisy procesoru do hlavní paměti

- V cestě je cache, která může blok, do kterého je zapisováno, obsahovat.
- Minimálně z pohledu daného procesoru je nutné zajistit koherenci dat pro daný procesor (často i pro více procesorů – vlákna) pro přístupy ke každé jednotlivé adrese i pokud existuje více cest přístupu
- **Write through** (zápis, propsání zkrz) cache – pokud se již data v cache nahází jsou modifikovaná, ve variantě s automatickou alokací jsou i při výpadku nahraná a pak modifikovaná. Data jsou zároveň odeslaná do hlavní paměti, buď přímo nebo přes zápisovou frrontu (write buffer)
- **Write back** – data jsou zapsaná do příslušného bloku cache, pokud není v cache, je blok nejdříve načtený. Blok je označený **dirty bit**em. Zápis do další úrovně až pokdu je potřeba danou položku cache umolnit pro jiná data (replacement) nebo je vyžádaná sychronizace procesorem, systémem (cache flush).

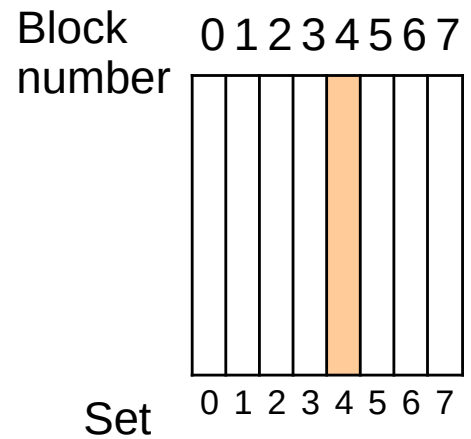
Základní typy organizace paměti cache

Uvažujeme cache o velikosti 8 bloků a kam může být mapovaný přístup na adresu/blok 12 pro tři varianty organizace skryté paměti

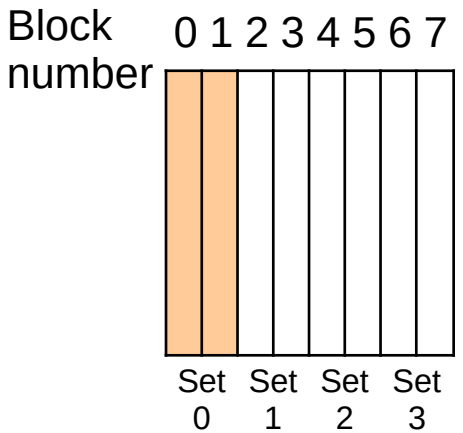
Plně asociativní
Fully associative:
Adresa 12 může být umístěna libovolně



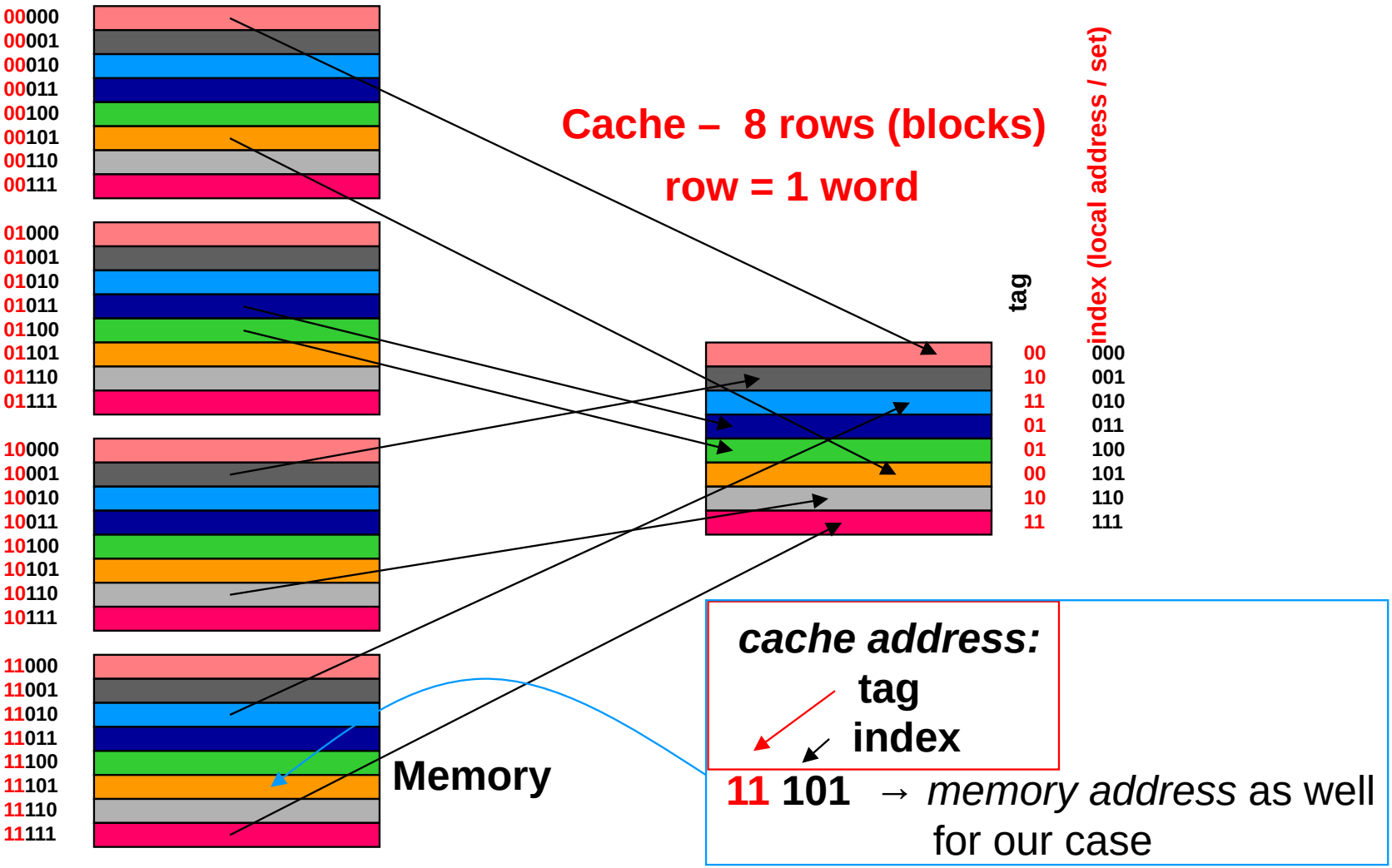
Přímo mapovaná
Direct mapped
Adresa 12 může být umístěna jen do bloku 4 (12 mod 8)



Dvoucestná
2-way associative
Adresa 12 může být umístěna do sady 0 (12 mod 4)

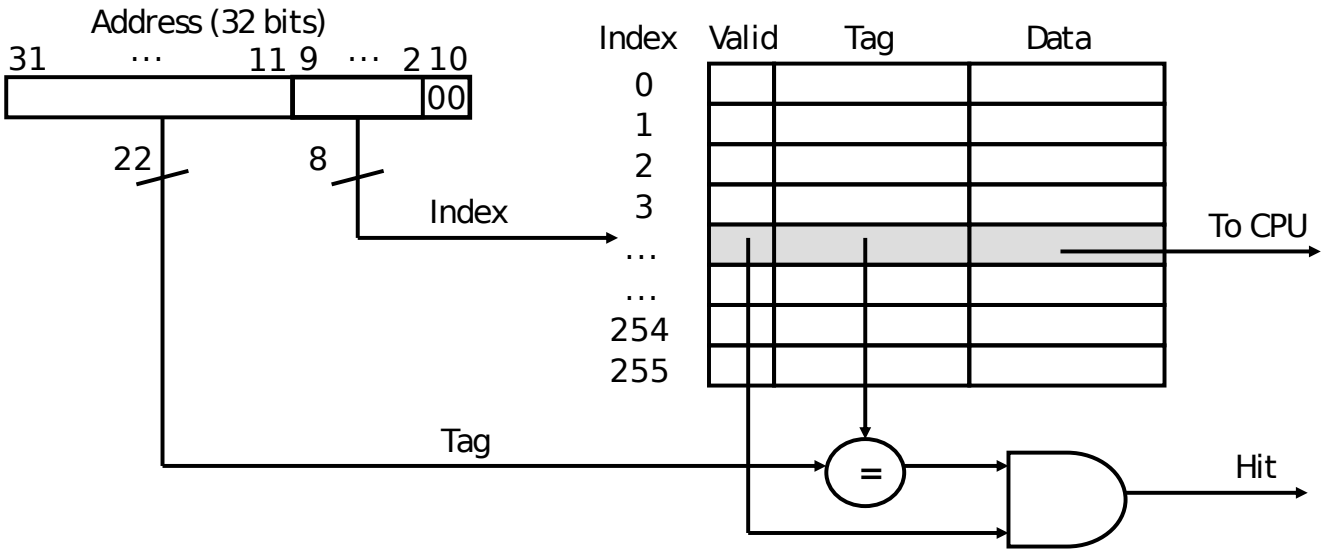


Přímo mapovaná paměť cache – mapování



Přímo mapovaná paměť cache s blokem rovným slovu

- **Capacity** – C ... kapacita (zde 1024 v bytech a 256 v slovech)
- **Number of sets** – SN .. počet setů, (zde 256)
- **Word size** – WS .. velikost slova (zde 4 byte)
- **Block size** – BS .. velikost bloku (zde 1 slovo, 4 byte)
- **Number of blocks** – BN .. počet bloků (zde 256)
- **Degree of associativity** – N . stupeň asciativity, počet cest (zde 1)

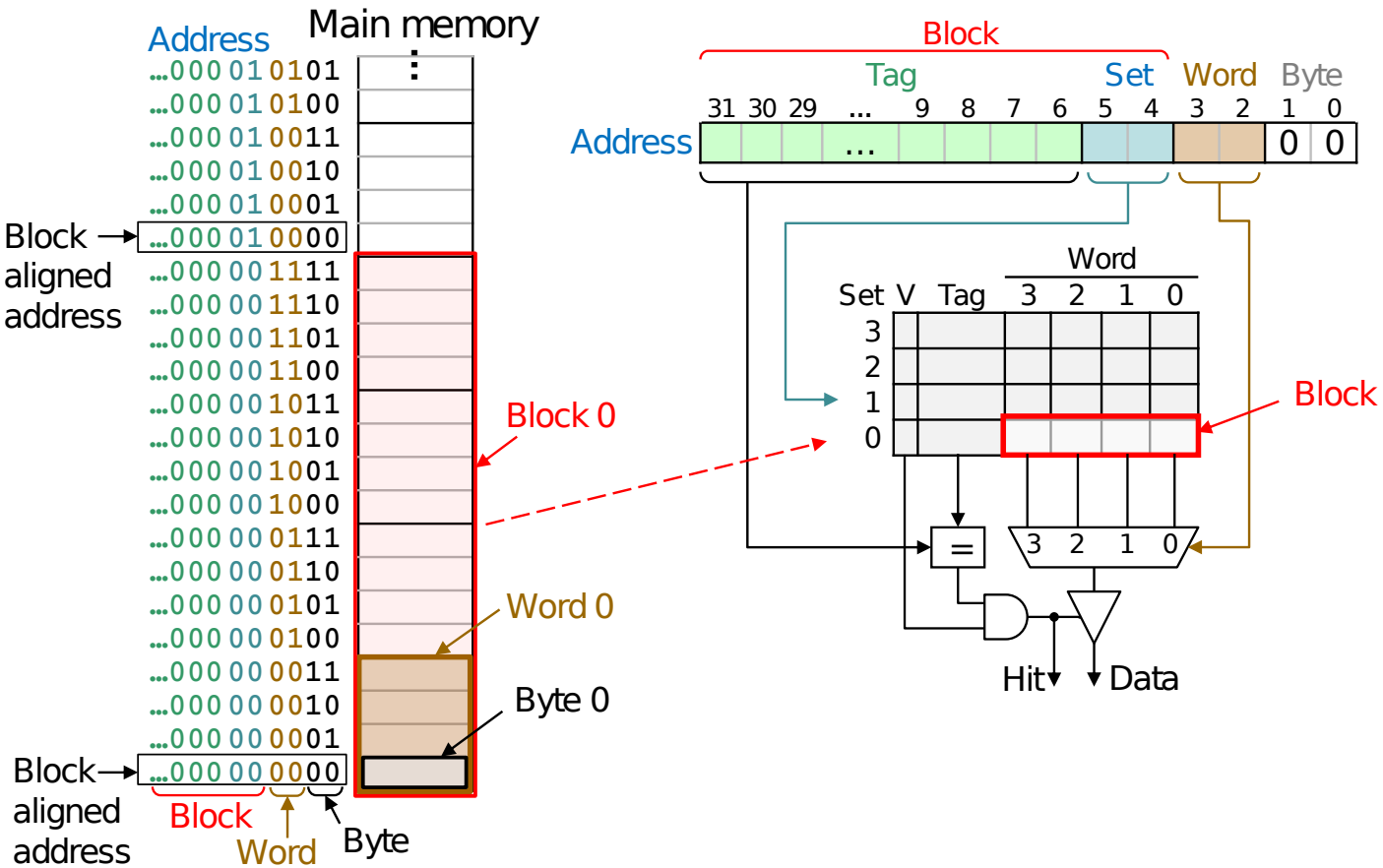


C = 1024 bytů, SN = BN = 256, BS = 1, N = 1

Přímomapovaná paměť cache s většími bloky

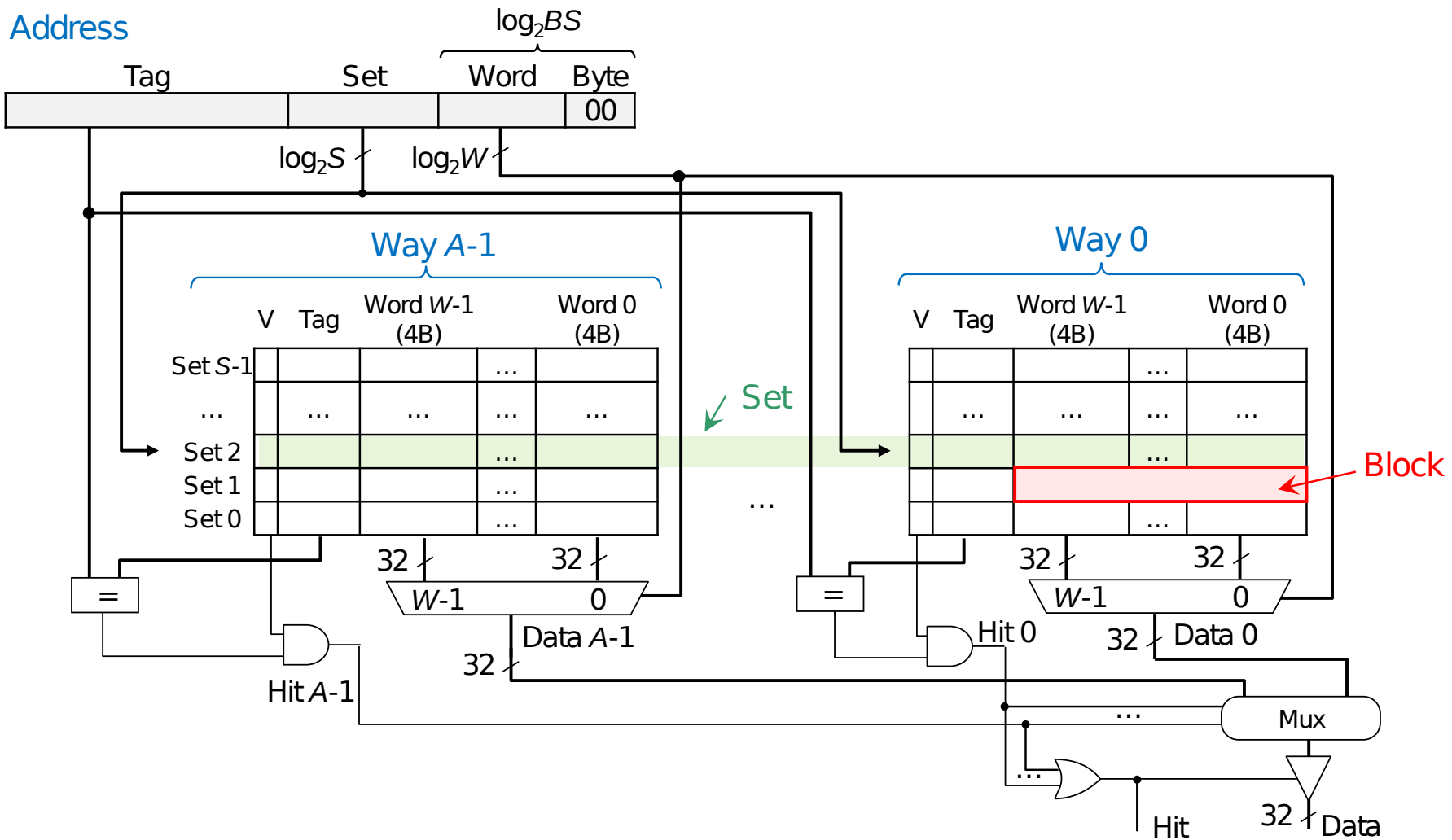
BS = 4 (16 byte, slovo WS = 4 byte), setů SN = 4

set = (adresa div BS) mod SN



Zdroj: Michal Štepanovský

Obecná organizace paměti cache s N cestami



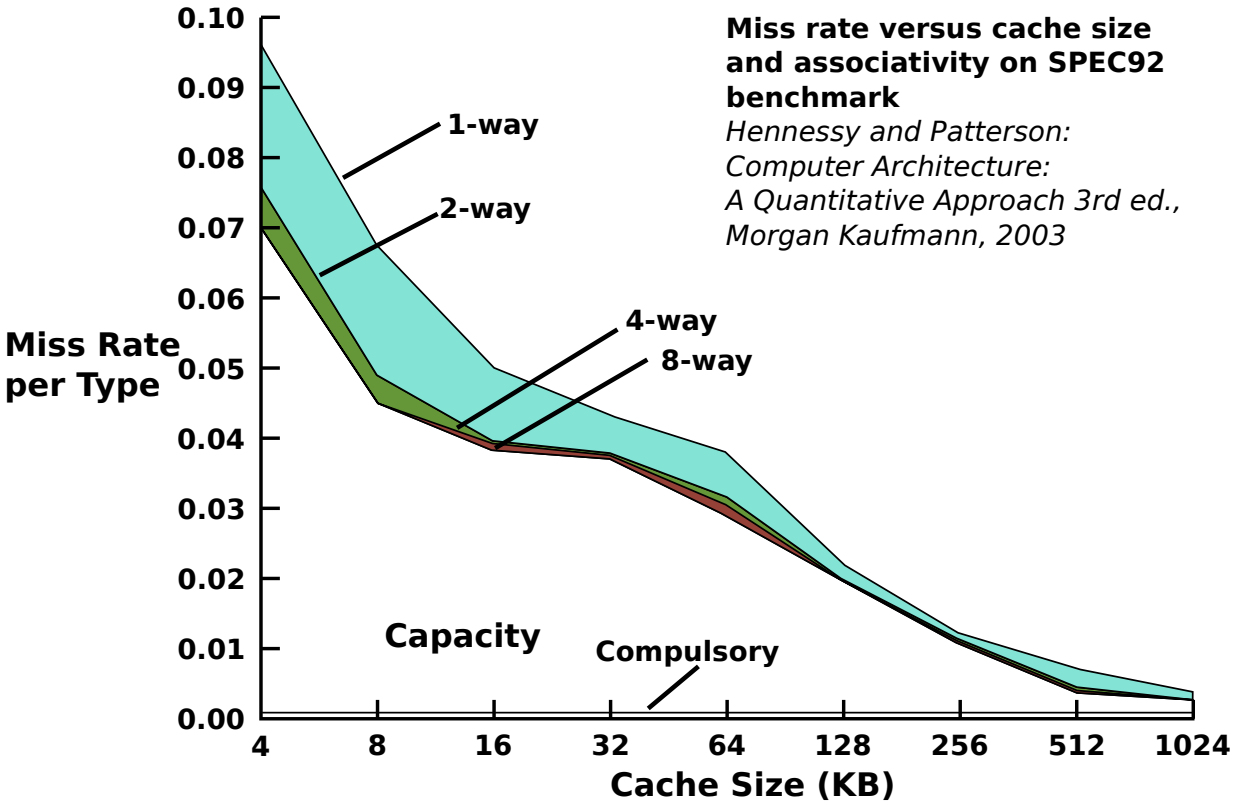
Zdroj: Michal Štepanovský

Ukázka: QtRvSim vect-inc, vect-add2, vect-add

Začneme s přímomapovanou skrytou pamětí (direct mapped cache) a `vect-inc`. Při velikosti bloku větší než jedno slovo a velikosti vektoru větší než kapacita jedné cesty bude u `vect-add2` docházet k značné degradaci výkonu. Po navýšení stupně asociativity na dvě cesty i při zachování celkové kapacity již ke kolizím při každé iteraci docházet nebude. Ovšem po rozšíření algoritmu na práci s třemi vektory (`vect-add`) opět problém při zpětném/odloženém zápisu (write-back) nastane a nejhorší situace nastane pro politiku záměny nejdéle nepoužívaného prvku (LRU). Zvýšení stupně na tři, nebo spíše obvyklejší čtyři, cesty se opět pro na začátek prázdnou cache počet výpadků omezí na jeden za každý načatý blok.

- `https://gitlab.fel.cvut.cz/b35apo/stud-support/-/tree/master/seminaries/qtrvsim/vect-inc`
- `https://gitlab.fel.cvut.cz/b35apo/stud-support/-/tree/master/seminaries/qtrvsim/vect-add2`
- `https://gitlab.fel.cvut.cz/b35apo/stud-support/-/tree/master/seminaries/qtrvsim/vect-add`

Vliv velikosti a organizace cache na počty výpadků

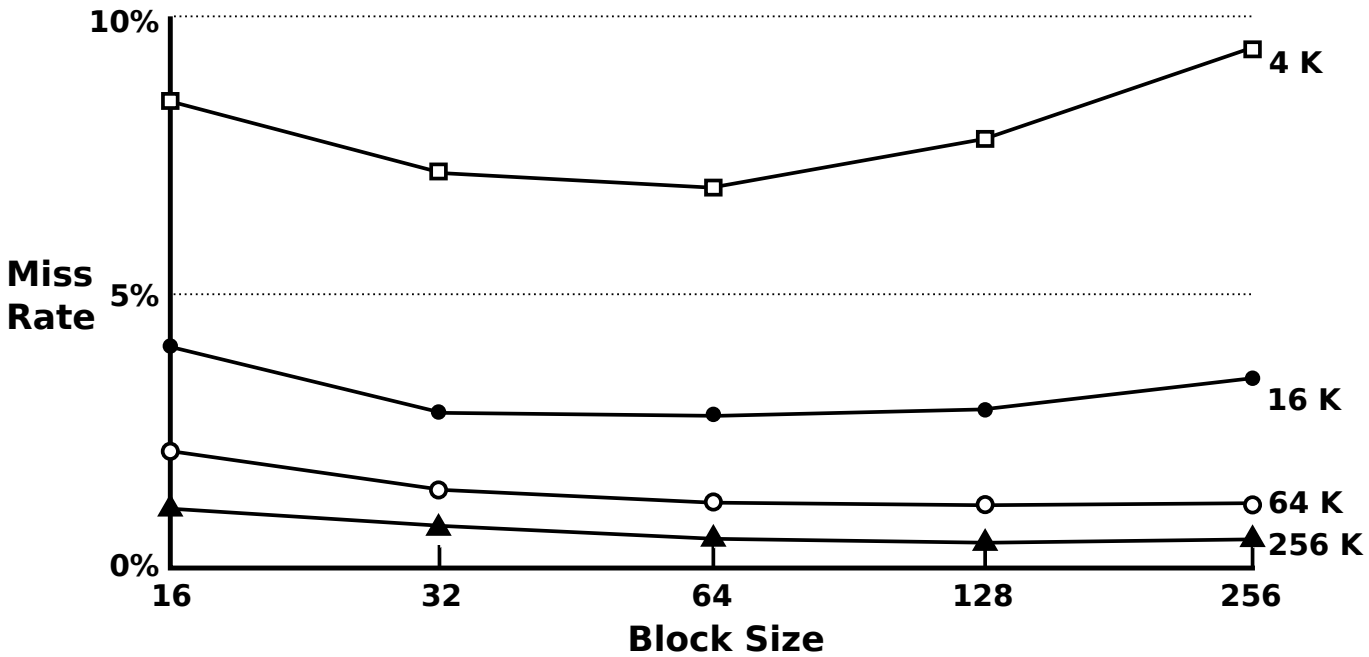


- miss rate není vlastnost, parameter cache
- miss rate není vlastnost programu

Jedná se o kombinaci jak algoritmů v programu, tak parametrů cache a často i zpracovávaných dat.

Vliv velikosti bloku a celé cache na počty výpadků

Miss rate versus block size and cache size on SPEC92 benchmark



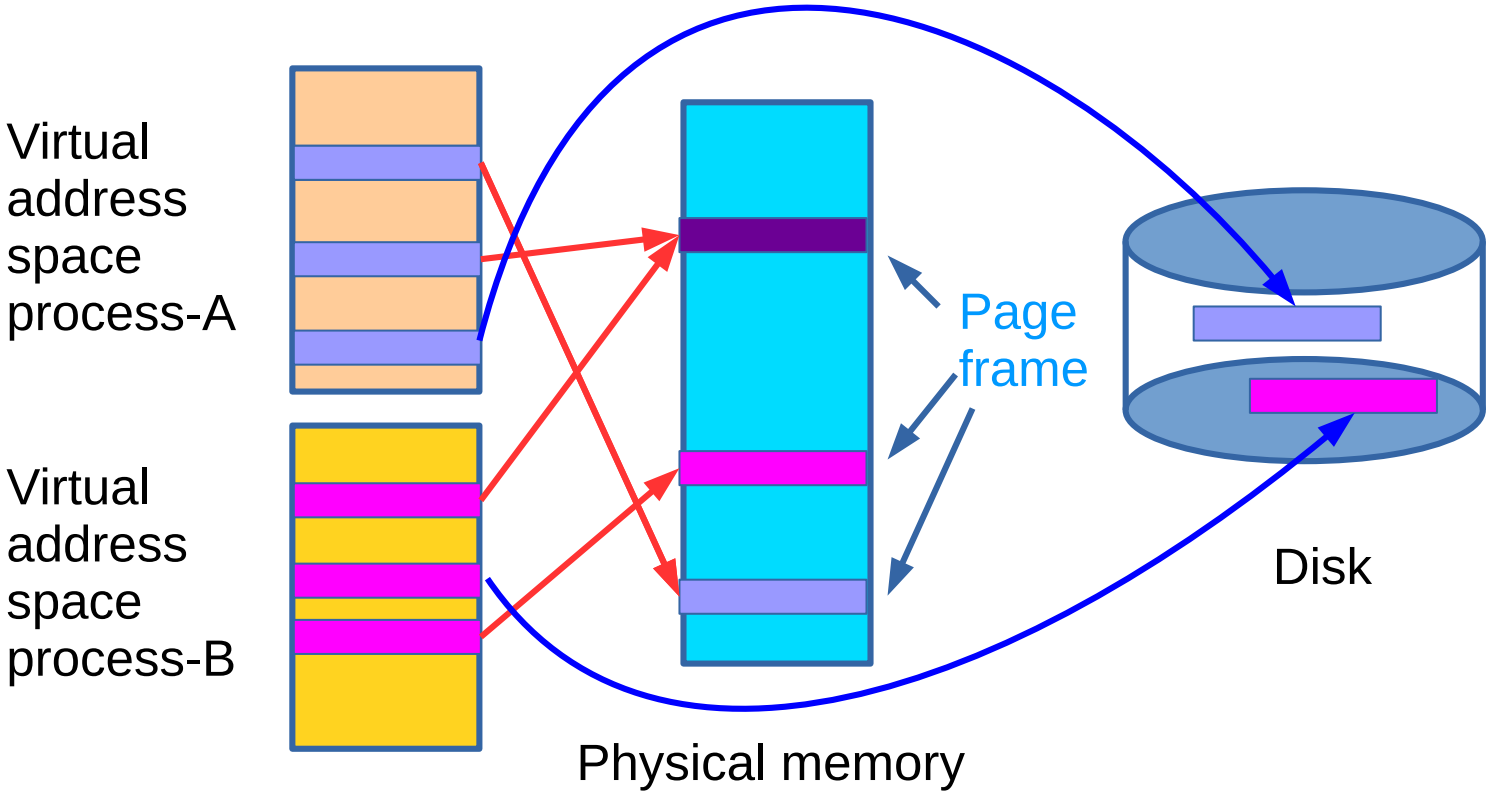
Zdroj: Hennessy and Patterson: Computer Architecture: A Quantitative Approach 3rd ed., Morgan Kaufmann, 2003

Zvětšení velikosti bloku napomáhá načtení sousedních dat, která mohou být často následně potřeba. Ale pokud tomu tak není, tak se zvyšuje cache miss penalty a zároveň dochází častěji ke kolizím a obsazení cache nepotřebnými daty.

Obsah

Paměťové prostory procesů a odkládání na disk

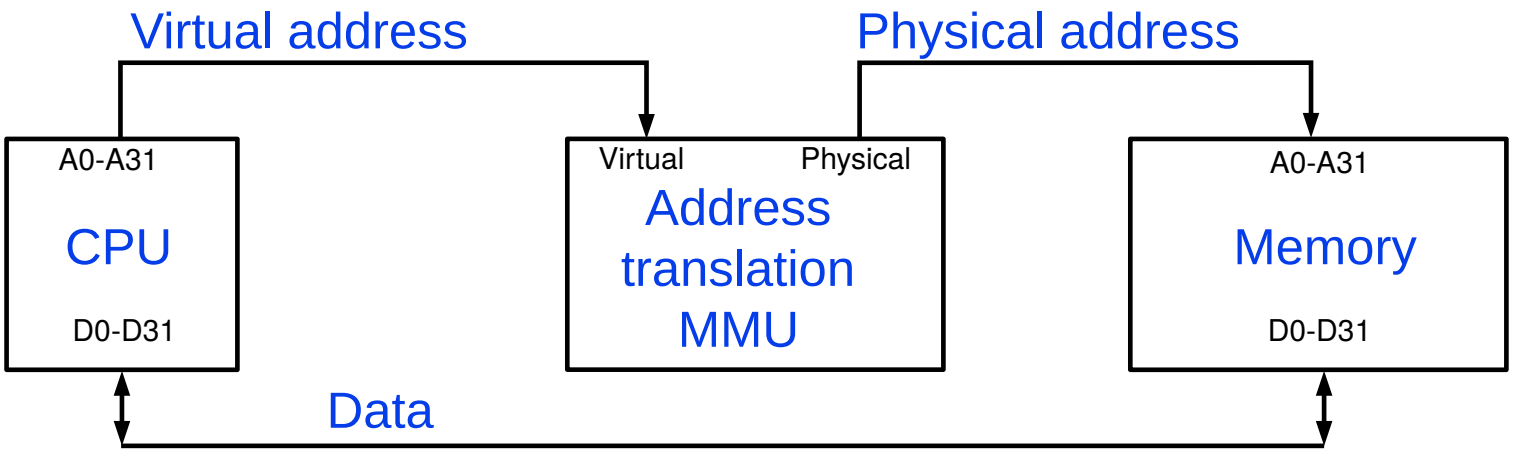
Více procesů, každý vlastní paměťový prostor. Vzájemná ochrana a možnost rozšíření kapacity hlavní paměti o prostor na sekundární – swap.



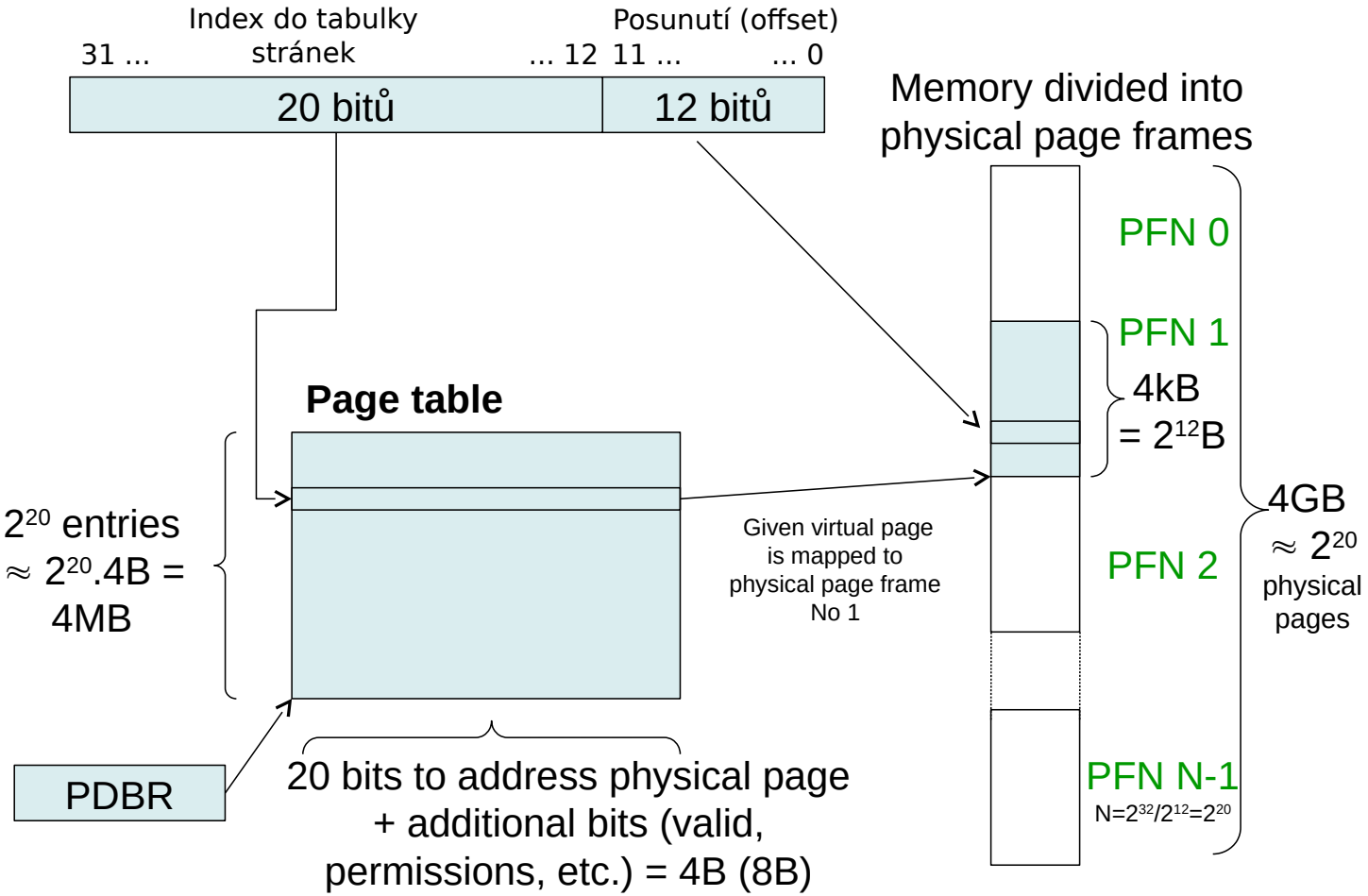
Překlad po jednotlivých bytech by byl drahý, prostor je rozdělěný na stránky (zarovnané), typicky 4 kB, někdy i větší, např. 64 kB.

Překlad virtuální a fyzické adresy

Překlad překlad zajišťuje **Memory Management Unit** (MMU).
Pro po vyplnění stránkovacích tabulek a nastavení **Page Directory Base Register** (PDBR) operačním systémem probíhá překlad na stránky přítomné v paměti většinou automaticky.
Naopak výpadky stránek ošetřují rutiny operačního systému. Pokud je přístup do namapované oblasti, načtou data z disku, sítě, odkládacího oddílu.
Stějně jako u cache je zde nutné a náročné hledat prostor pro nově potřebné stránky (princip podobný LRU).

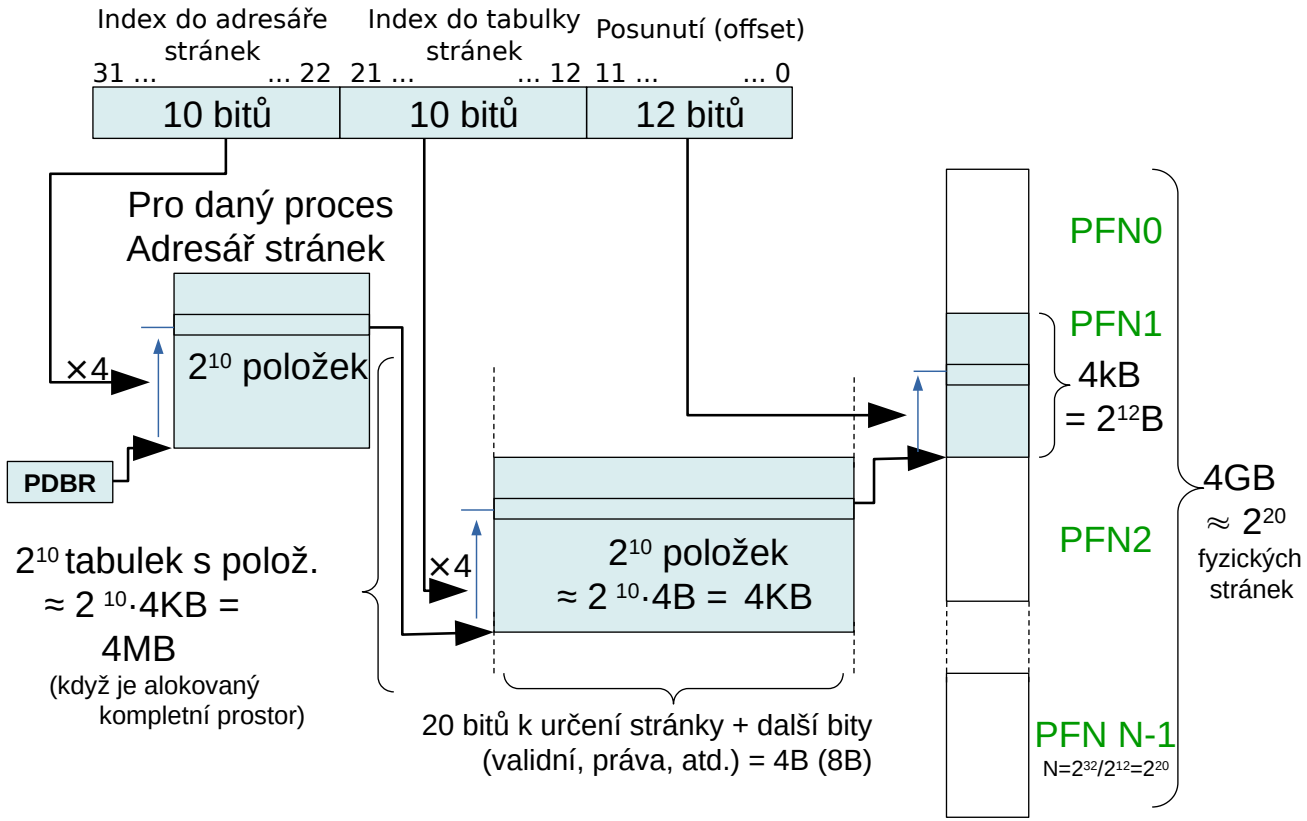


Jednoúrovňová stránkovací tabulka



Řešení je nevýhodné, pro každý, i malý, spuštěný proces je na 32-bitovém systému pro 4 kB stránky potřeba alokovat 4 MB (překlad 20-bitů adresy, 4-byte na 32-bit položku)

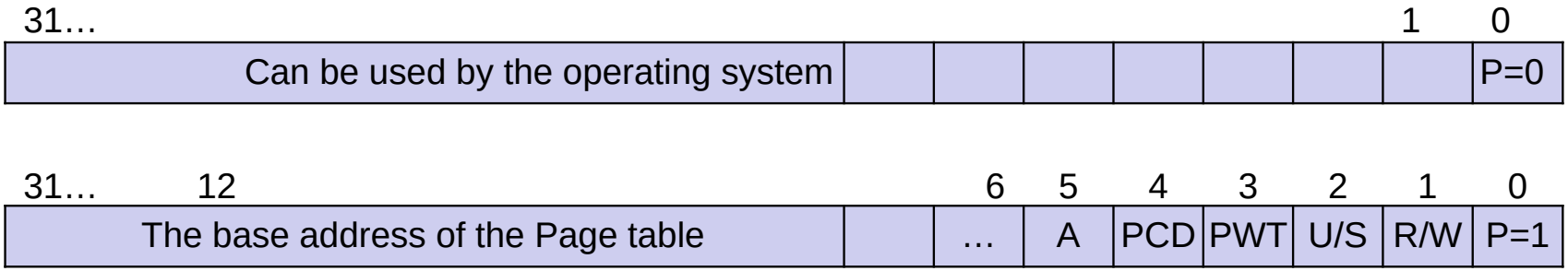
Dvouúrovňové stránkování



Adresář stránek pro každý proces, úroveň tabulek s vlastními položkami se alokuje jen, když je daná oblast virtuálního prostoru použitá.

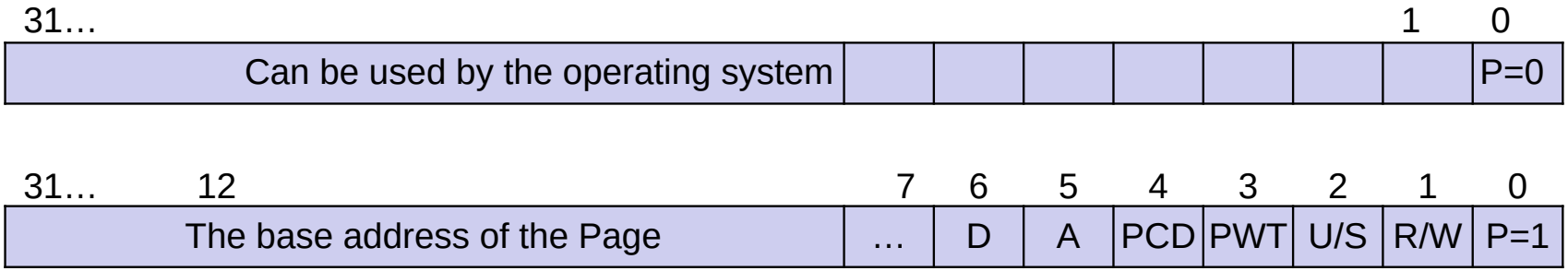
Za pozornost stojí, že při překladu po 10-ti bitech vychází pole položek adresáře i tabulky na velikost 4 kB. Nepotřebuje alokace vyššího řádu ohrožené fragmentací.

Položky adresářů stránek – **Page Directory Entry** (PDE)



- bit 0: Present bit – další úroveň nebo stránka je přítomná v paměti (1). Pokud není (0) tak jsou data uložena na disku nebo oblast není mapovaná. Někdy také bit označovaný V – valid bit.
- bit 1: Read/Write: 1 – R/W – zápis povolený; 0 – read only (RO)
- bit 2: User/Supervisor: 1 – přístupné uživateli; 0 – přístup jen operační systém
- bit 3: Write-through/Write-back – nastavení typu přístupu
- bit 4: Cache disabled/enabled – důležité pro paměťově mapované periferie, přímý přístup do registrů bez cache
- bit 5: Accessed – nastaven MMU při prvním přístupu z procesoru

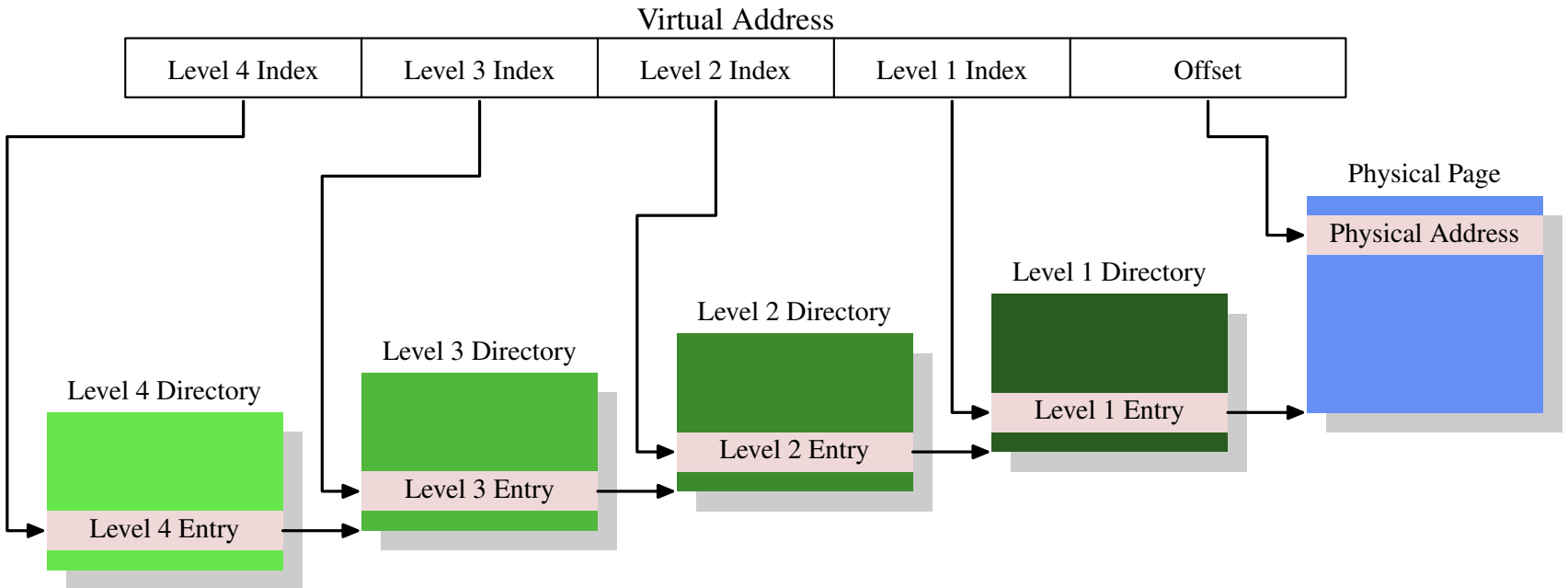
Položky koncových tabulek – **Page Table Entry** (PTE)



- bit 6: Dirty bit, případně někdy Modified – jen nastavený MMU, pokud došlo k zápisu do dané stránky od poslední kontroly a případného nulování příznaku operačním systémem.
- Ostatní bity mají stejný význam jako v položce adresáře

Bit **Accessed** je klíčový, když operační systém hledá stránky fyzické paměti k uvolnění. Obvykle počítá podle bitů další, obsáhlejší, statistiku a při průchodu seznamem bity nuluje. Pokud je nastavený **Dirty**, tak je při uvolnění fyzciké stránky (PFN) je nutné data sychronizovat zpět do souboru, odkládacího oddílu atd.

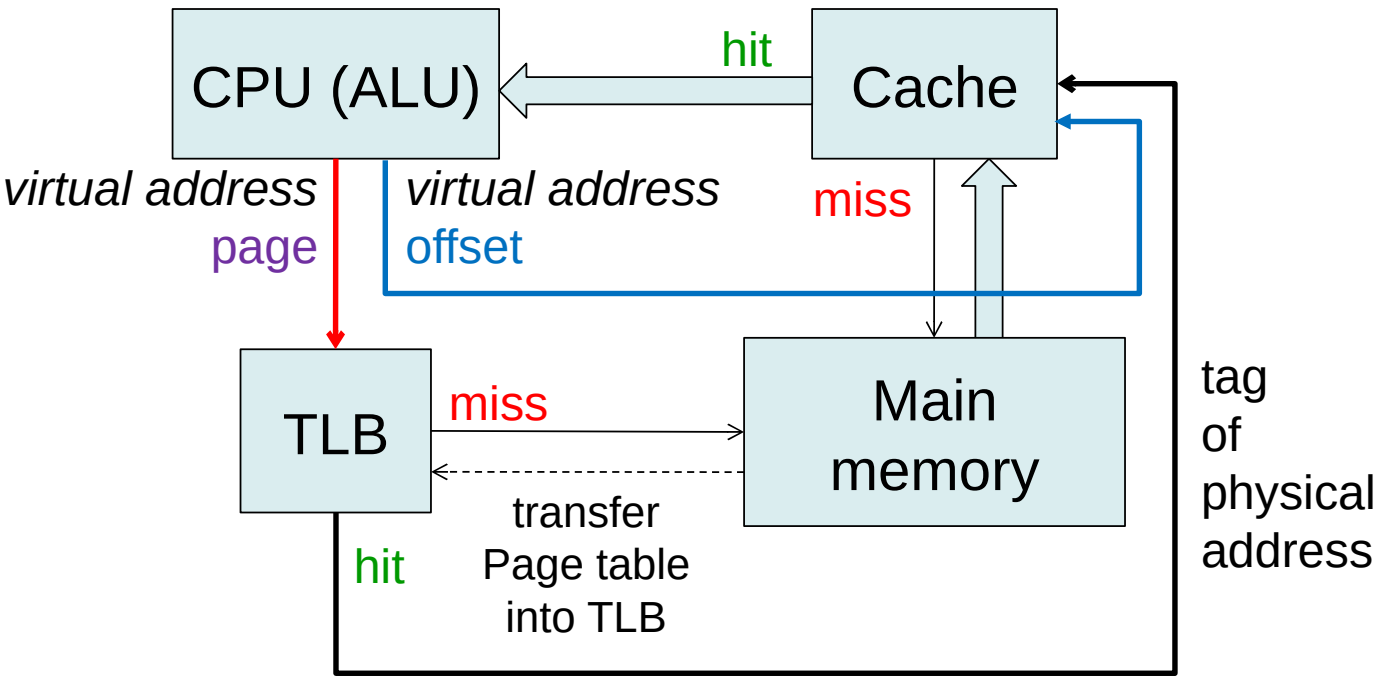
Překlad na 64-bitových architekturách



Vyžaduje více úrovní. Pro 64-bitů a 4 kB stránky je teoreticky potřeba přeložit 52-bitů. Pokud jednotlivé části tabulek mají zaplnit jednu stránku, tak vychází překlad na 9 bitů ($512 \times 8 = 4096$) na úroveň a tedy $ceil(52/9) = 6$. Často se ale jedna až dvě úrovně vynechávají, více později.

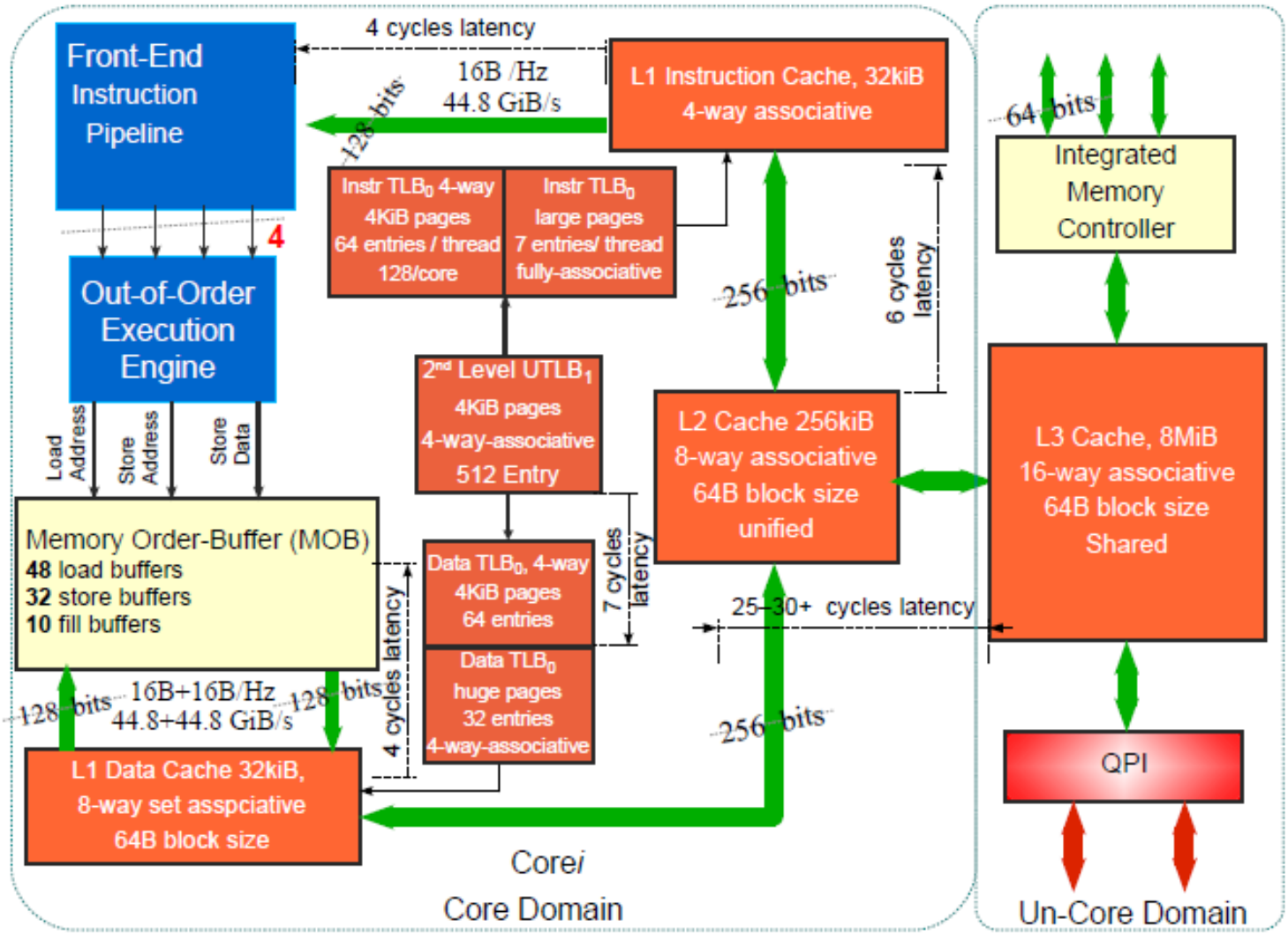
Zrychlení opakovaného překladu

Každý překlad představuje několik přístupů do paměti a i když je urychlený přístupy přes cache, tak zpomaluje. Zavádí se **Translation Look-Aside Buffer** (TLB). Pracuje na shodném principu jako paměťová cache, ale ukládá k virtuální adrese její překlad na fyzickou. Opět omezená kapacita, LRU a nutnost brát v potaz při programování.



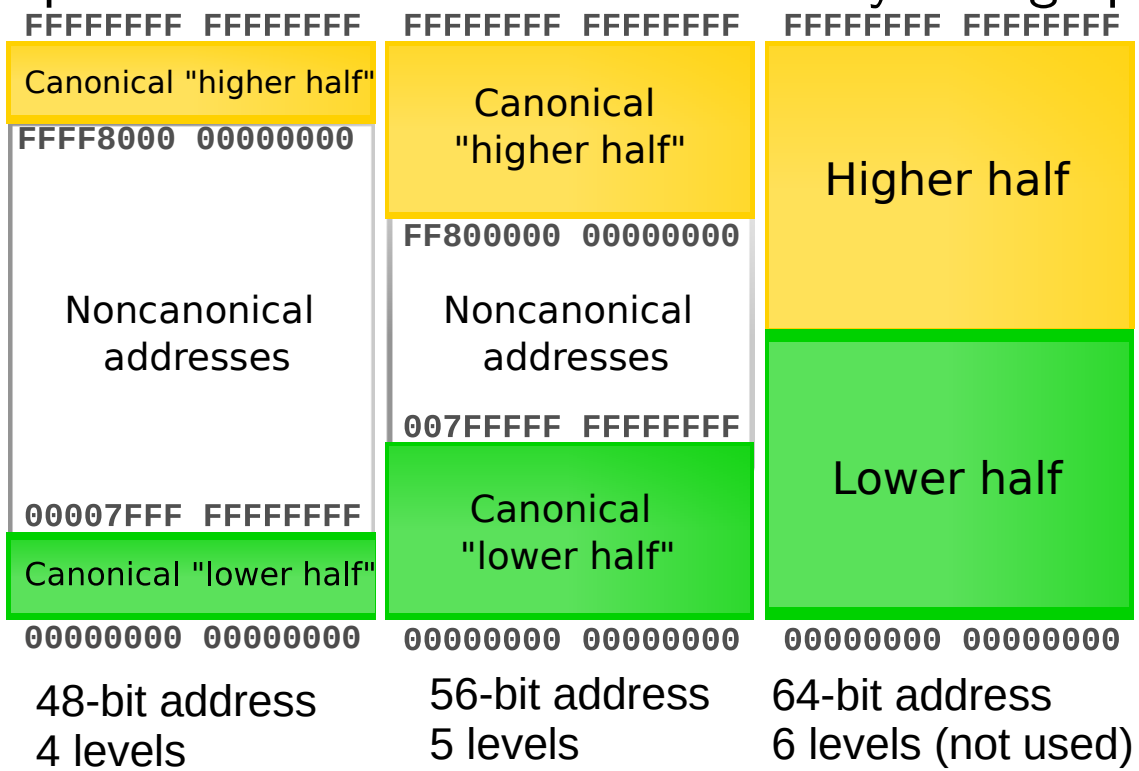
Obsah

Paměťový subystém - Intel Nehalem



Stránkování na 64-bitových architekturách

Plná 64-bitová délka fyzické adresy nemá (zatím) použití. Ani 64-bitová virtuální adresa. Úrovně zpomalují běh. Nejvyšší bity se nahrazují znaménkvým rozšířením. Horní oblast operační systém, spodní pro aplikace. Další optimalizace volitelné velké stránky – huge pages.



Odkazy a literatura

- Ulrich Drepper: What every programmer should know about memory
- Agner Fog: Software optimization resources. C++ and assembly
- `https://www.7-cpu.com/cpu/Haswell.html`
- `https://www.7-cpu.com/cpu/Skylake.html`
- WikiChips: Zen - Microarchitectures - AMD

Prostor na poznámky