

Paměti

Petr Jedlička, Cyril Klimeš, Tomáš Pitner

Předmluva

HTML · Slides · PDF | Přehled předmětu

◎ Cíle přednášky

- Seznámit se s použitím, typy a technickou realizací **pamětí**.
- Nejprve připomenutí jednotek pro udávání velikosti pamětí

🔑 Bez **pamětí** by si počítač nepamatoval ani jak počítat (program), ani s čím počítat (data). Vnitřní paměti jsou pro přímé užití během výpočtu, vnější pro delší ukládání i po vypnutí stroje.

Přehled jednotek velikosti dat

- Desítkové jednotky (SI) jako ve fyzice
- Dvojkové jednotky (IEC) jako v informatice

Desítkové jednotky (SI)

Jednotka	Název	Velikost v bajtech (B)
1 kB	kilobajt	1 000
1 MB	megabajt	1 000 000
1 GB	gigabajt	1 000 000 000
1 TB	terabajt	1 000 000 000 000
1 PB	petabajt	1 000 000 000 000 000

Dvojkové jednotky (IEC)

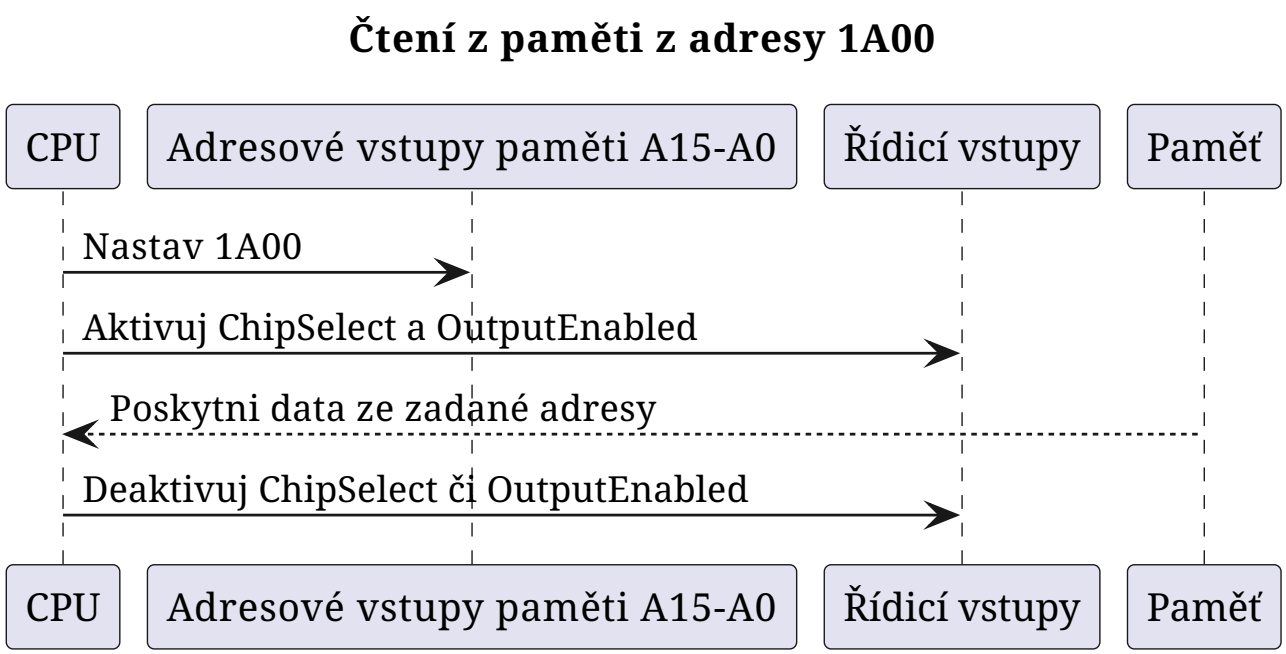
Jednotka	Název	Velikost v bajtech (B)
1 KiB	kibibajt	1 024
1 MiB	mebibajt	1 048 576
1 GiB	gibibajt	1 073 741 824
1 TiB	tebibajt	1 099 511 627 776
1 PiB	pebibajt	1 125 899 906 842 624

Účel pamětí

- **pamatovat** si údaje (zapíšu — přečtu i opakovaně)
- pokud možno **rychle** je poskytnout
- buďto **po dobu běhu** programu/počítače
- nebo **trvale**, evt. do přemazání

Přístup k paměti: čtení

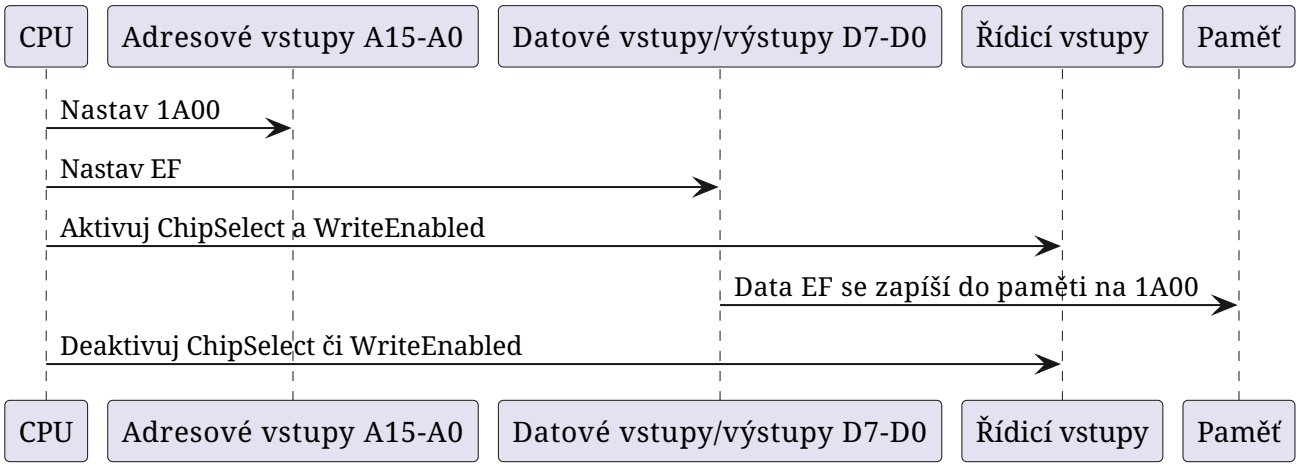
1. zadáme **adresu**
2. aktivujeme **požadavek** čtení
3. paměť nám **vrátí data** uložená na zadané adrese



Přístup k paměti: zápis

1. zadáme nová **data** k uložení
2. zadáme **adresu**
3. aktivujeme požadavek **zápisu**
4. paměť **uloží** zadaná data na zadanou adresu

Zápis do paměti bajt EF na adresu 1A00



Základní parametry pamětí

kapacita	*množství dat*, které lze do paměti uložit (obvyklé jednotky: bajt, kilobajt…)
vybavovací doba	čas od vydání příkazu ke čtení z paměti do okamžiku, kdy je informace k dispozici
rychlost toku dat	množství informací, které lze přečíst/zapsat *za jednotku času*

Další parametry: chybovost

chybovost	průměrná frekvence chyb, např. 10^{-11} bitů je v průměru chybně přečteno/zapsáno (*Mean Time Between Failure* je 300000 hod)
energetická závislost	zda musí být *stále připojeny* ke zdroji el. energie
destruktivita čtení	zda *čtení zničí* uložená data

Další parametry: přístupy, počty

způsob přístupu	• přímý • sekvenční ("postupně") • kombinovaný (index-sekvenční)
počet možných zápisů/čtení	u některých typů (např. *Flash NAND*) je relevantních údajem počet možných přepisů jedné buňky, neboť je statisticky omezen

Paměti dle technologické realizace

magnetické	založené na magnetických vlastnostech materiálu, informaci uchovává *směr magnetizace*
optické	využívá optických vlastností materiálu, např. *odraz světla*
polovodičové	využívá vlastností tranzistorů; buď se realizují *klopnými obvody* (technologie TTL), nebo *obnovováním elektrického náboje* (CMOS)
magnetooptické	pomocí světla (laser) se mění *magnetické vlastnosti* materiálu

Vnitřní a vnější paměti

- potřebuje každý počítač; jsou v ní *instrukce programu a data*, se kterými pracuje;
- část vnitřní paměti počítače nebo zařízení je obvykle **zapisovatelná** (RW, RAM, DRAM, DDR): pro data i zavedené programy (software)
- část je **jen pro čtení** (ROM, PROM, EPROM, EEPROM): pro kontroly, zabezpečení, zavaděč operačního systému…
- pevné disky (HDD, SDD)
- výměnná média (Flash, optická, magnetická)

Vnitřní paměti dle přístupu

Nyní se omezíme na vnitřní paměti, tedy realizované jako polovodičové:

Zapisovatelné paměti	• také RW (*read-write*) neboRAM (*Random Access Memory*) • pro data, programy
Jen pro čtení	• ROM (*read-only*), PROM (*Programmable ROM*), EPROM (*Erasable PROM*), EEPROM (*Electrically Erasable PROM*)

Volatilní paměti

Volatile (dočasné):

- energeticky závislé
- slouží pro průběžné ukládání vstupů, mezivýsledků, výstupů…
- pro další výpočty, zobrazování, před uložením ven…

Pracovní paměti

Vnitřní pracovní paměti počítače tvoří hierarchii od nejmenší a nejrychlejší po největší (a nejpomalejší):

1. **registry** CPU
2. **cache** (L1, L2, L3, případně i více úrovní)
3. **hlavní paměť** ("RAM", operační paměť)

Registry CPU

Pro nejrychlejší práci, vyhnout se zcela přístupu vně CPU

- V instrukcích se někdy pojmenovávají mnemotechnicky:
  - `AL` (8 bitů), `AX` (16), `EAX` (32), `RAX` (64), `BL`...,
- Někdy číselně:
  - `R0` .. `R31` (všechny třeba 32bitové).
- Bývají jich max desítky, např. 32 (obecně u RISC více).

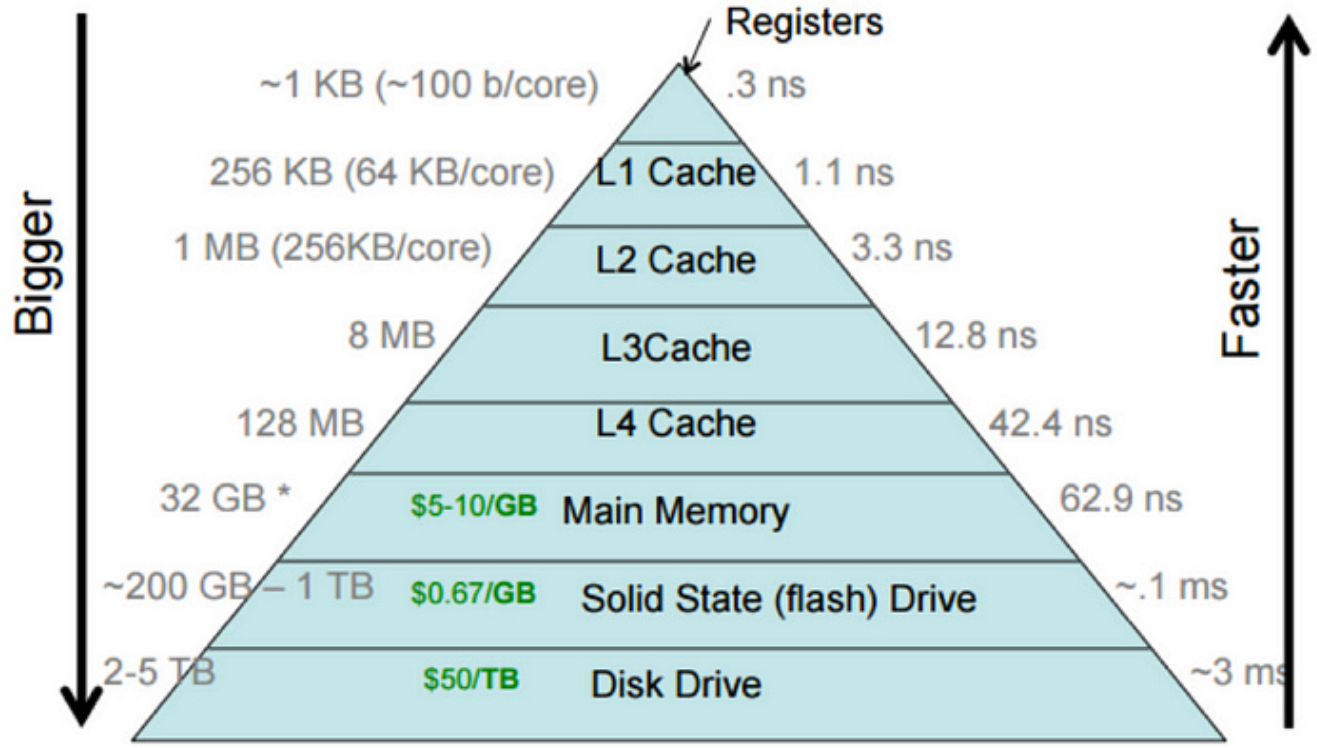
Procesorové cache

- hned nejbližší větší paměti než registry
- jejich použití je *transparentní*, programátor o ní nemusí *ani vědět*, procesor je plní sám při přístupech do hlavní paměti a při opakovaném čtení z hlavní paměti je využívá pro zrychlení
- u "velkých" CPU většinou více úrovní cache (*L1*, *L2*, *L3*)
- nižší úrovně jsou *zvlášť* u každého jádra, *L3* evt. *L4* pak *společné*
- cache **L1** je obvykle **100× rychlejší** než paměť RAM, kapacity u nových řad Intel 13. a 14. gen. 80 kB/jádro
- cache **L2** je přibližně **25× rychlejší**, kapacity 2-4 MB/jádro
- **L3** cache je největší paměťovou jednotkou v CPU, kapacita 24-36 MB/CPU

Hlavní (operační) paměť

Operační paměť

- největší, velikosti dnes běžně **GB až desítky GB**, *min. 8, spíše 16+ GB*
- protože jde vždy o paměť typu RAM (i pro zápis)
- používá se označení RAM jako synonymum k hlavní/operační paměti



Obr. 1.Hierarchie a rychlosti

Adresování pamětí

registry triviální, jak jsme ukazovali: adresa registru = jméno registru je přímo součástí instrukce, např. `add rax, 42`.

cache jako programátoři neadresujeme přímo, ale ptáme se cache na hodnotu *Y*, která se v hlavní paměti nachází na adrese *X*.

hlavní paměť přímo zadáním adresy (prvního bajtu) načítaného/zapisovaného slova.

Manipulace s daty

- Při získání údaje čtením se obsah paměti přímo nemaže,
- ALE: u *dynamických pamětí zeslábne náboj* v paměťové buňce a musí se *obnovit* (*refresh paměti*), o to se naštěstí ani programy ani OS nemusí starat.
- Lze do ní zapsat novou informaci tak, že se přepíše existující informace.
- Běžně se pro hlavní paměti na PC používají *dynamické paměti* (DRAM).
- Na vestavěných zařízeních (kontroléry) může být paměť *statická* (SRAM), aby se zjednodušilo obvodové řešení.

Adresování

- cílem je určit místo, kde chceme číst nebo zapisovat

Fyzické adresování:

- adresy, které jsou přímo posílány na adresové vstupy paměťových obvodů (=fyzické adresy)

Logické adresování:

- logické adresy jsou adresy používané v programu;
- jsou následně převedeny na fyzické adresy

Fyzické adresování

- **Přímé adresování** (adresou)
- **Asociativní** (klíčem)

Řízení přímého adresování na úrovni obvodu

řídicí vstupy aktivace čipu (`Chip Enable`), zápis (`Write Enable`)

adresové vstupy $A_0..A_{N-1}$

- adresa = číslo na adresových vstupech

datové vodiče vstup i výstup ($D_0..D_{M-1}$), M je počet bitů jedné adresovatelné jednotky

Přímé

- místo (adresa) pro zápis nebo čtení dat je přesně určeno hodnotou na *adresových vstupech* (adresové sběrnici)
- má tolik bitů, kolik adresovatelných pozic je třeba adresovat
- *N* adresových bitů umožní adresovat 2^N pozic

Asociativní

- Asociativní paměť je typem *paměti adresovatelné obsahem* (*Content-Addressable Memory*, CAM)
- na základě určitých dat (*klíče*) předložených paměti se získá místo (*adresa*) nebo rovnou *hodnota* na tomto místě
- princip asociativní paměti lze přirovnat k *asociativním polím* (hash, mapa) ve většině programovacích jazyků

Využití asociativních pamětí

- vyrovnávací (cache) paměti
- velmi rychlé vybavování

Logické adresování

- 👉 postup, jak paměť adresuje počítačový program
- ta se **adresovacími obvody** a look-up stránkovými **tabulkami** převede na fyzickou adresu
- využívá se technik *segmentování* (historicky) nebo *stránkování* (nyní)

Realita dle kategorie procesorů

Malé procesory a kontroléry:

- tzv. plochý adresový model; prostě adresa je jedno *dále nestrukturované číslo*
- fyzická = logická adresa
- často *oddělené* části paměťového prostoru dat a instrukcí (=*Harvardská architektura*)

Větší procesory:

- *společný* adresový prostor dat a instrukcí (=*von Neumannova architektura*)
- mapování logického na fyzický prostor většinou pomocí *stránkování*

Vývoj logického adresování v architekturách x86

Osmibitové procesory

- řada Intel kdysi začala 4bitovým procesorem **4004** pro kalkulačky, historicky prvním mikroprocesorem; adresování až *4 kB paměti* (2^{12} bajtů)

- následovaly osmibitové jako Intel **8080** s "kontrolérovými" varianta i jako 8048 nebo 8051: ta žije dodnes; adresování *64 kB paměti* (2^{16} bajtů)

16- a 32bitové procesory

- pak 16bitové **8086**: adresování *až 1 MB paměti* (2^{20} bajtů)
- 16bitové **80268**: první, kde fungovaly *izolování paměti* procesů a OS navzájem; adresování *až 16 MB paměti* (2^{24} bajtů)
- 32bitové **80386**: první, kde bylo *stránkování*; adresování *až 4 GB paměti* (2^{32} bajtů)

8086 (16bitový)

- Povinná segmentace: segmentové registry `CS`, `DS`, `SS`, `ES`
- Každý segment max 64 KB
- Adresace: `segment x 16 + offset`

80286 (16bitový)

- Zaveden *protected mode* s deskriptory
- Tabulka, kde je pro každý segment jeho začáteční adresa ve fyzickém prostoru a
- Stránkování ještě neexistovalo

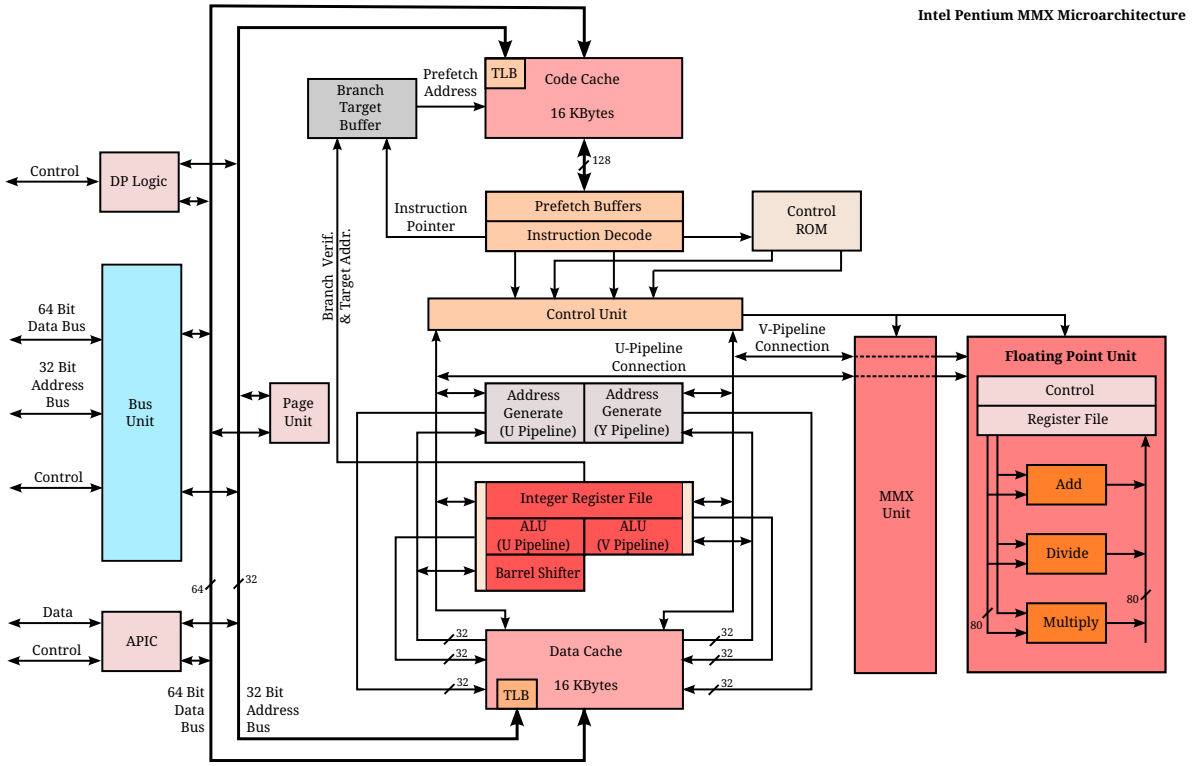
80386 (32bitový)

- Segmentace + nově **stránkování** (4KB stránky) používané dnes dominantně, viz dále
- Segmenty až 4 GB + nové registry FS a GS
- Možnost flat memory modelu (segmenty se základnou 0)
- 👉 dnešní způsob bez segmentace

80486 a Pentium

80486 (32bitový)

- 80486: fyzická adresa 32 bitů, logická 46 bitů
- Segmentace zůstává, ale OS používají flat model
- Paměťová ochrana řešena stránkováním
- Segmenty zůstávají kvůli zpětné kompatibilitě



Obr. 2. Pentium (64bitový)

x86-64 a Intel Core

- **Long mode** (64bit): segmentace do značné míry ignorována
- CS, DS, ES, SS mají nulovou základnu, fixní limit
- Pouze FS a GS zůstávají aktivní
- Paměť řízena výhradně *stránkováním*.

Shrnutí

- Moderní OS (Windows, Linux) používají plochý/flat memory model
- Segmentace slouží pouze ke kompatibilitě
- Stránkování/paging zajišťuje ochranu, izolaci i virtualizaci

Stránkování (paging)

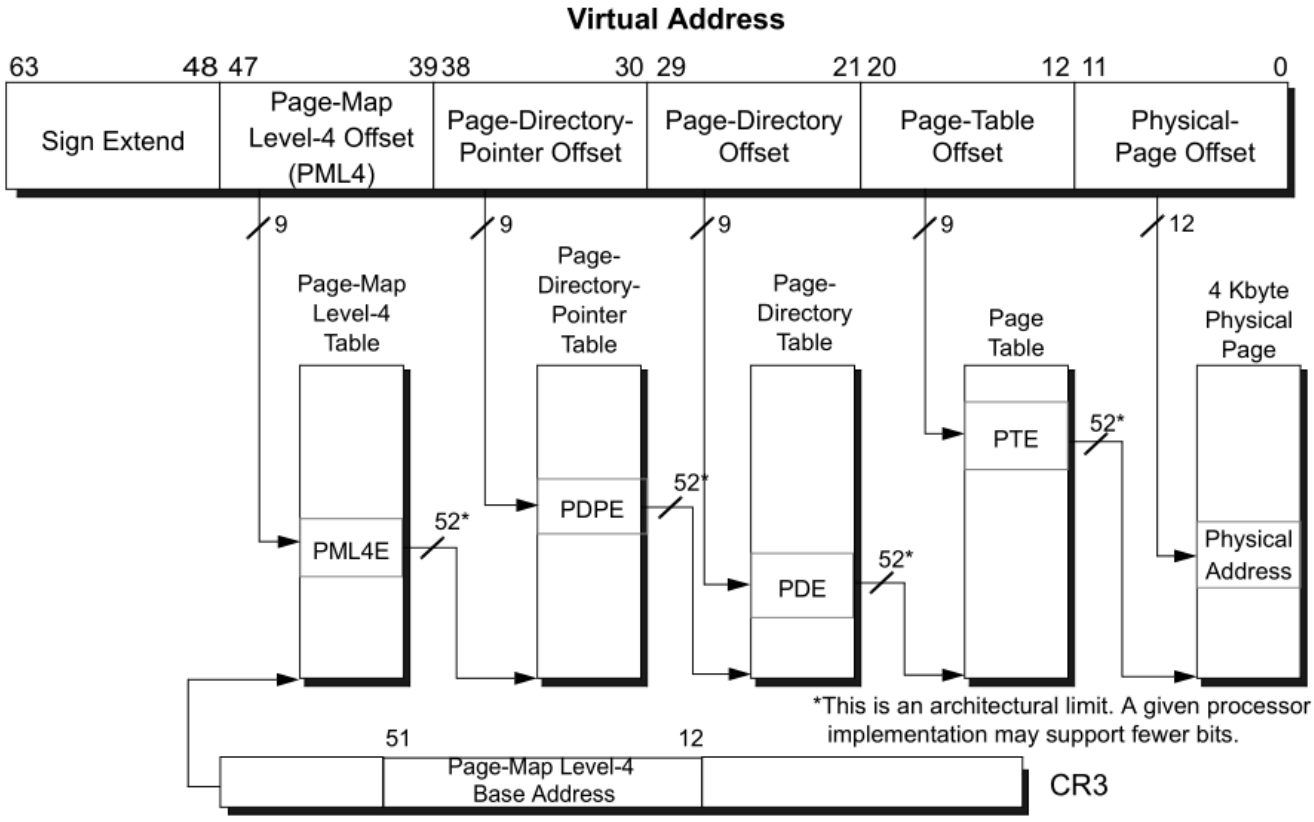


Figure 5-17. 4-Kbyte Page Translation—Long Mode

Obr. 3. Plné stránkování na Intel x64

- Yosifovich, Pavel, et al. Windows Internals: System architecture, processes, threads, memory management, and more, Part 1. Microsoft Press, 2017.

Princip stránkování x64

V podstatě postupný převod přes look-up tabulky:

1. Virtuální (logická) adresa se rozdělí na bitové úseky odspodu (od LSB) takto:
   - 12 bitů (0-11) bude adresa bajtu v rámci základní jednotky stránkování = **stránky**
   - 9 bitů (12-20) index do tzv. **tabulky stránek** (Page Table, PT) s 512 položkami
   - 9 bitů (21-29) index do tzv. **tabulky adresáře stránek** (Page-Directory Table, PDT) s 512 položkami
   - 9 bitů (30-38) index do tzv. **tabulky ukazatelů na adresáře stránek** (Page-Directory Table, PDT) s 512 položkami
   - 9 bitů (39-47) index do tzv. **tabulky ukazatelů na adresáře stránek** (Page-Directory Table, PDT) s 512 položkami

Princip stránkování x64

2. Nejvyšších 16 bitů (48-63) se nepoužije
3. Skupiny bitů se použijí jako 9bitové indexy do výše uvedených tabulek 👉

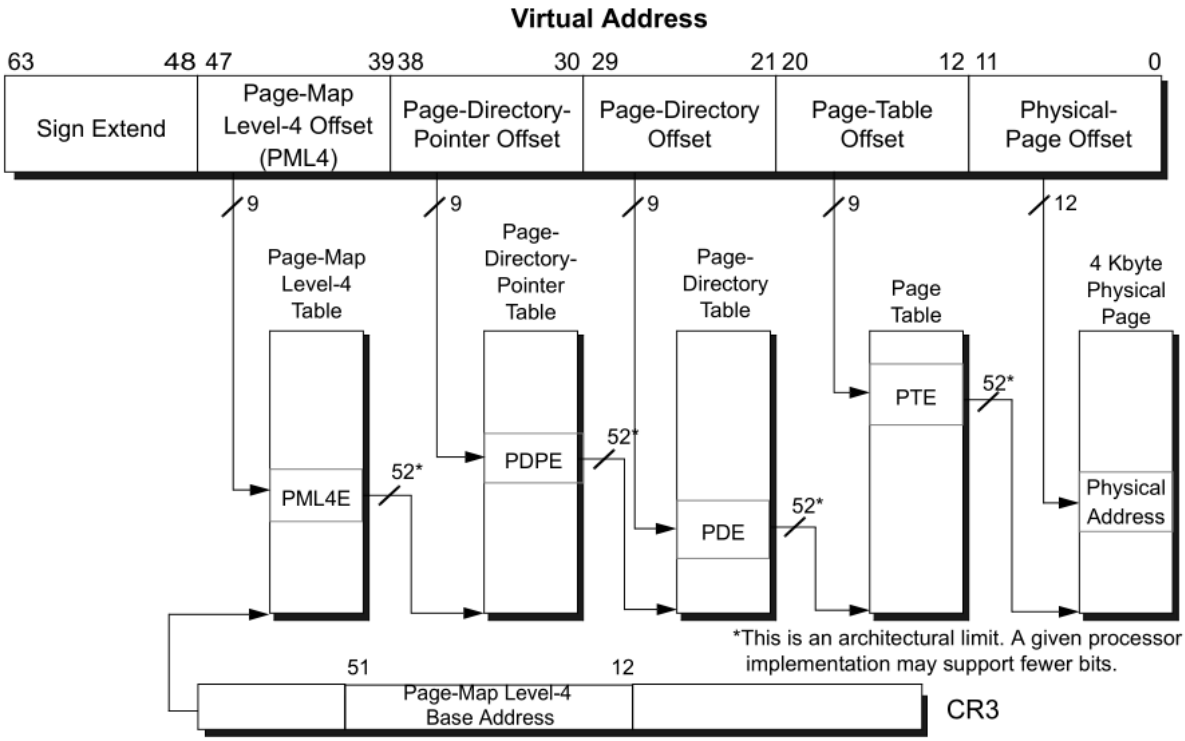


Figure 5-17. 4-Kbyte Page Translation—Long Mode

Shrnutí

Úroveň stránkování	Název tabulky	Rozsah bitů virtuální adresy	Funkce (index)	Počet položek
1.	Page-Map Level-4 Table (PML4)	47–39 (9 bitů)	Index do PML4	512
2.	Page-Directory-Pointer Table (PDPT)	38–30 (9 bitů)	Index do PDPT	512
3.	Page-Directory Table (PDT)	29–21 (9 bitů)	Index do PDT	512
4.	Page Table (PT)	20–12 (9 bitů)	Index do PT	512
—	Offset ve stránce	11–0 (12 bitů)	Posun ve stránce	4096 bajtů (4 KiB stránka)

- Paging on Intel x64

Adresování hlavní paměti

- Nejmenší adresovatelná jednotka bývá **bajt** (8 bitů).
- Tedy každý bajt má *jednoznačně danou adresu*.
- Datová sběrnice na 64bitových procesorech přenáší data po **64bitových slovech** (osmibajtových), přičemž se dále přenášejí celé **cache linie** (typicky 64 bajtů).



Čtení jednoho bajtu tedy fakticky znamená načtení celé *cache linie* do procesoru; procesor pak z ní využije jen požadovaný bajt. K libovolnému bajtu *uvnitř* zarovnané linie je přístup stejně rychlý — pokud ale bajt překročí hranici cache linie, může být přístup nepatrně pomalejší.

Zarovnávání adres

Proto se uložení datových struktur obvykle děje od adres, které jsou násobky aspoň 8 nebo 16 (bajtů) = provede se *zarovnání* (*alignment*):

Zarovnané po 8 bajtech

- `0x00000000`
- `0x00000008`
- `0x00000010`
- `0x00000018`

Zarovnané po 16 bajtech

- `0x00000000`
- `0x00000010`
- `0x00000020`
- `0x00000030`

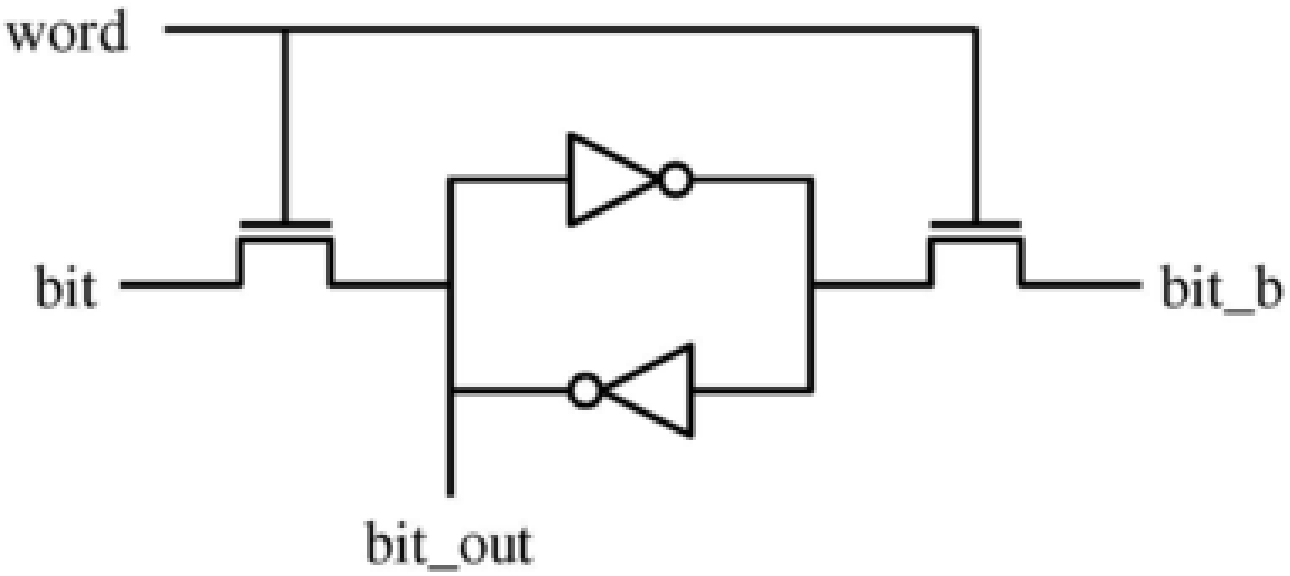
Statické paměti

- realizace jedné buňky (bitu) **jedním klopným obvodem** (flip-flop)
- **SRAM** je *volatilní*, závislá na napájení
- při odpojení se informace vytratí
- ale nepotřebuje na rozdíl od **DRAM** aktivní *oživování* (refresh)
- ani po čtení hodnoty

Princip SRAM

- matice *řádky * sloupce* a na průsečících jsou jednotlivé *buňky*
- buňka je obvykle na bázi **CMOS**
- bistabilní klopný obvod (dva CMOS invertory proti sobě)
- dva přístupové tranzistory (set/reset), každý zapojen na sloupcový a invertovaný sloupcový vodič

Buňka SRAM (1 bit)



Použití SRAM v současnosti

- RAM nebo vyrovnávací paměť v kontrolérech, (obvykle kolem 32 bajtů až 128 KB)
- *primární cache* (od 8 KB do mnoha MB) vCPU,
- stavové informace a *registry* CPU,
- v zákaznických obvodech (obvykle v řádu kilobajtů)
- v programovatelných hradlových polích (FPGA)

Pseudostatická paměť RAM (PSRAM nebo PSDRAM)

- PSRAM je dynamická paměť RAM s *vestavěným obvodem pro refresh* a řízení adres, díky čemuž se chová podobně jako statická RAM,
- Spojuje vysokou hustotu DRAM s jednoduchostí použití typickou pro skutečnou SRAM.
- PSRAM se používá například v Apple iPhone a dalších vestavěných systémech.

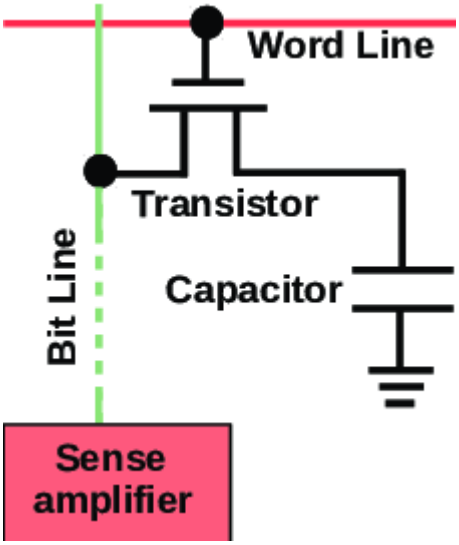
Dynamické paměti

- dynamická paměť ve smyslu "musí se obnovovat"
- realizace jedné buňky (bitu) pomocí dvojice CMOS + kapacitor držící náboj
- i čtení náboj snižuje
- nutné stálé periodické obnovování náboje

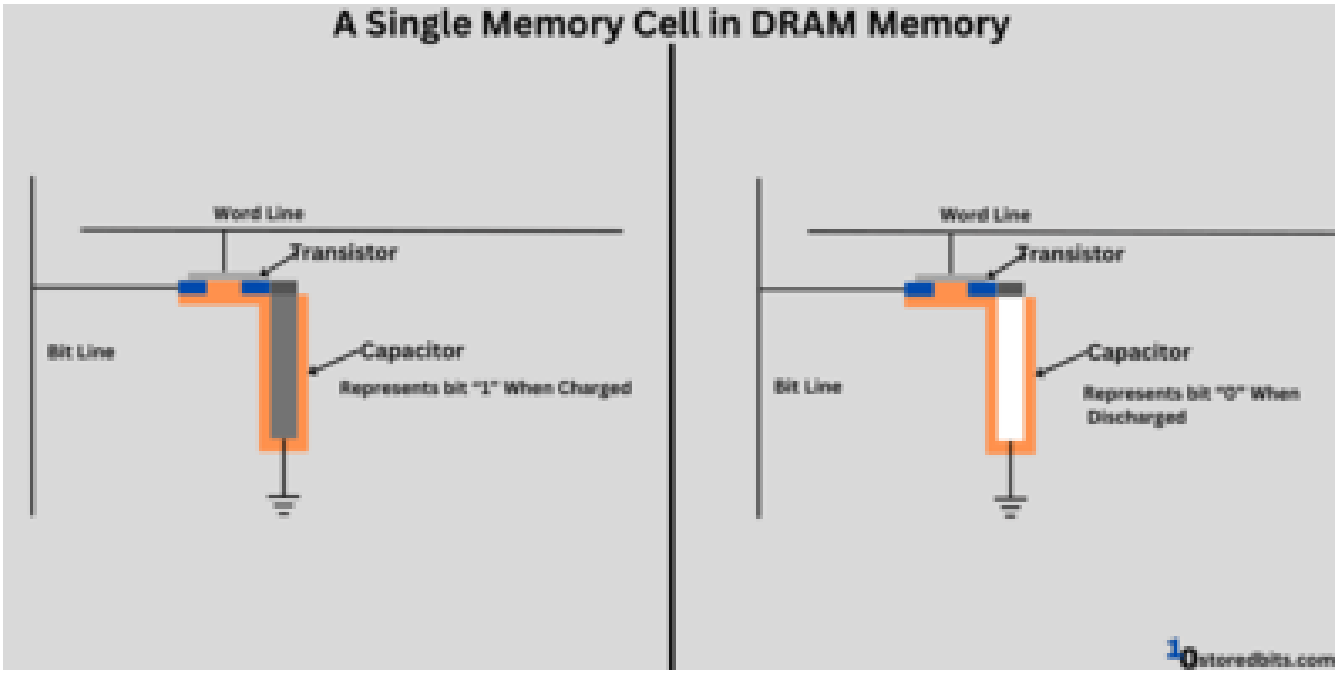
Principy

- Ever wondered how your computer’s memory works so fast? Dive deep into the world of DRAM!
- Understanding the DRAM: How does Computer Memory Work?

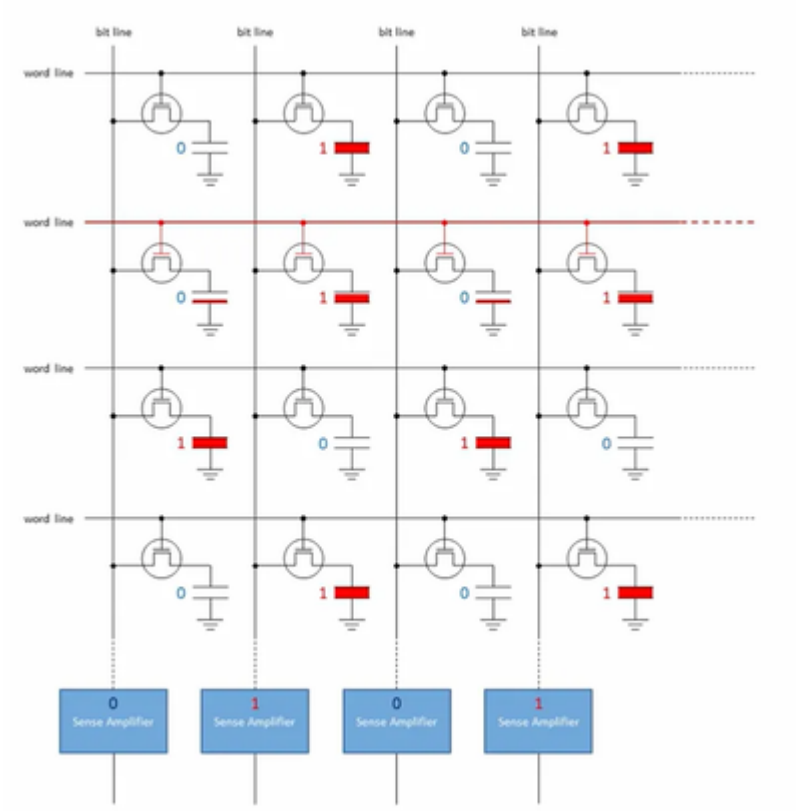
Buňka DRAM paměti



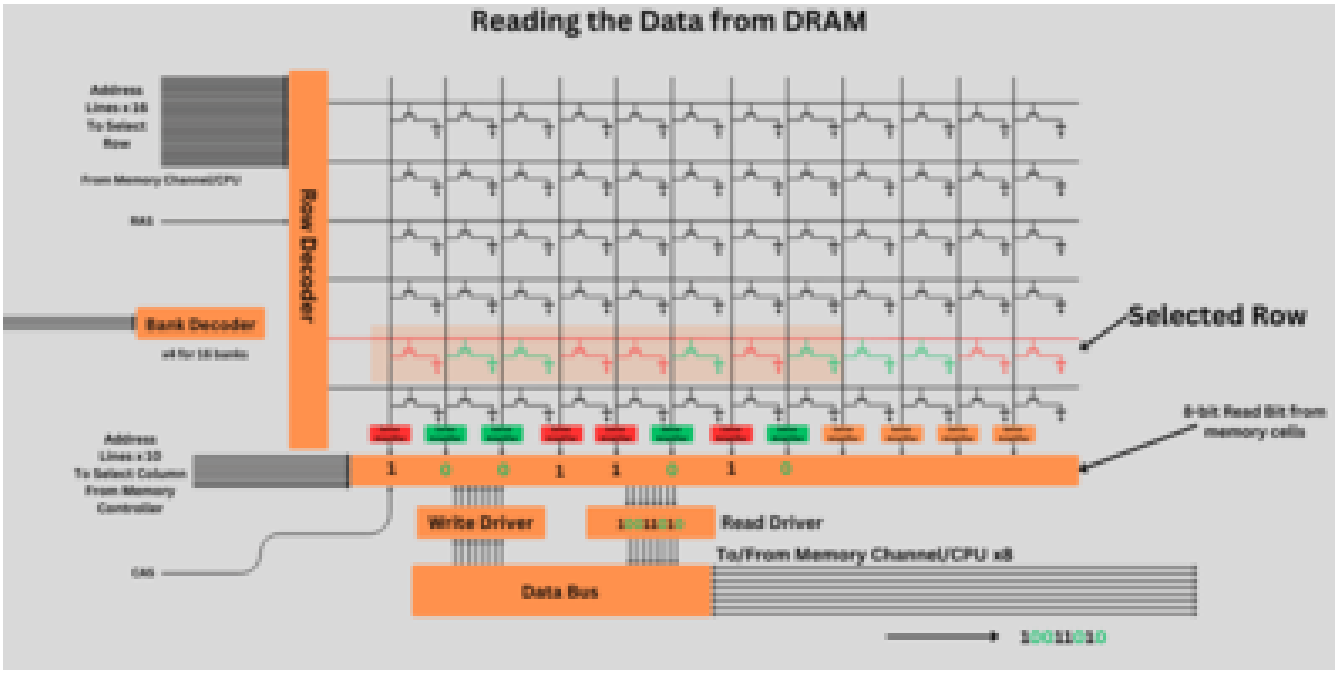
Uložený bit v buňce DRAM



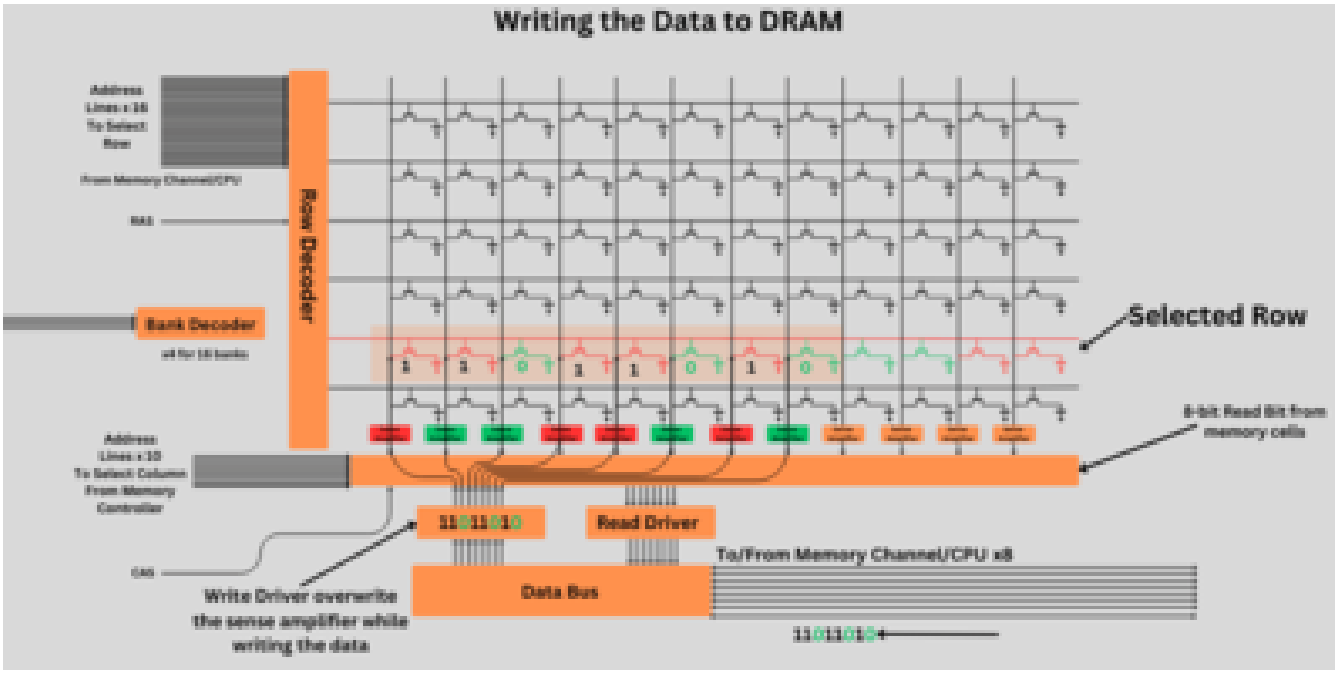
Čtení z buňky DRAM



Čtení z DRAM celkově



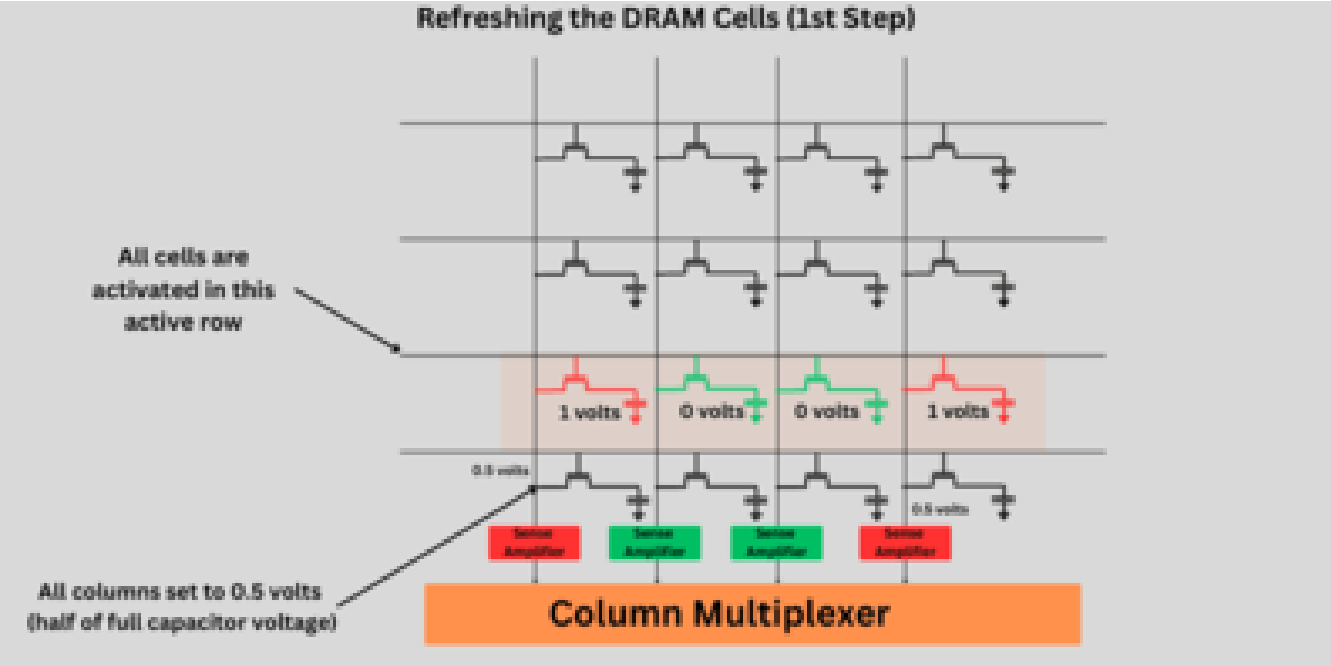
Zápis do DRAM



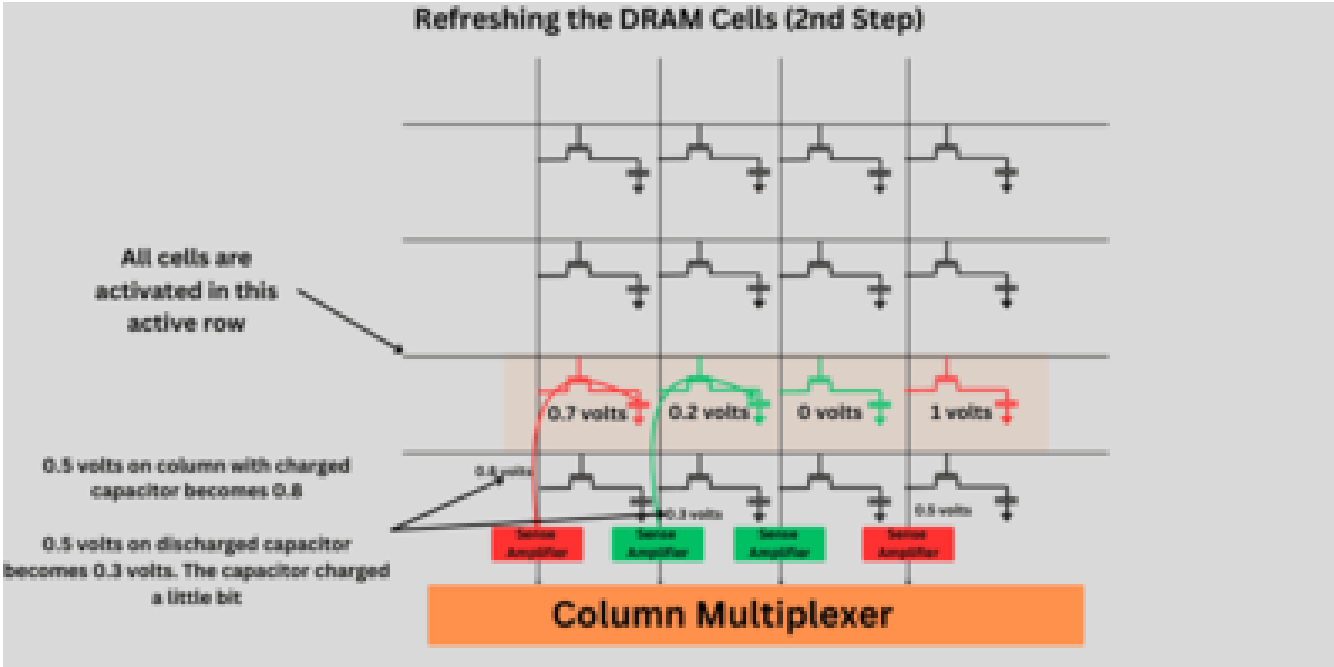
Refresh (oživování) DRAM

1. Detekce
2. Zesílení
3. Zpětný zápis

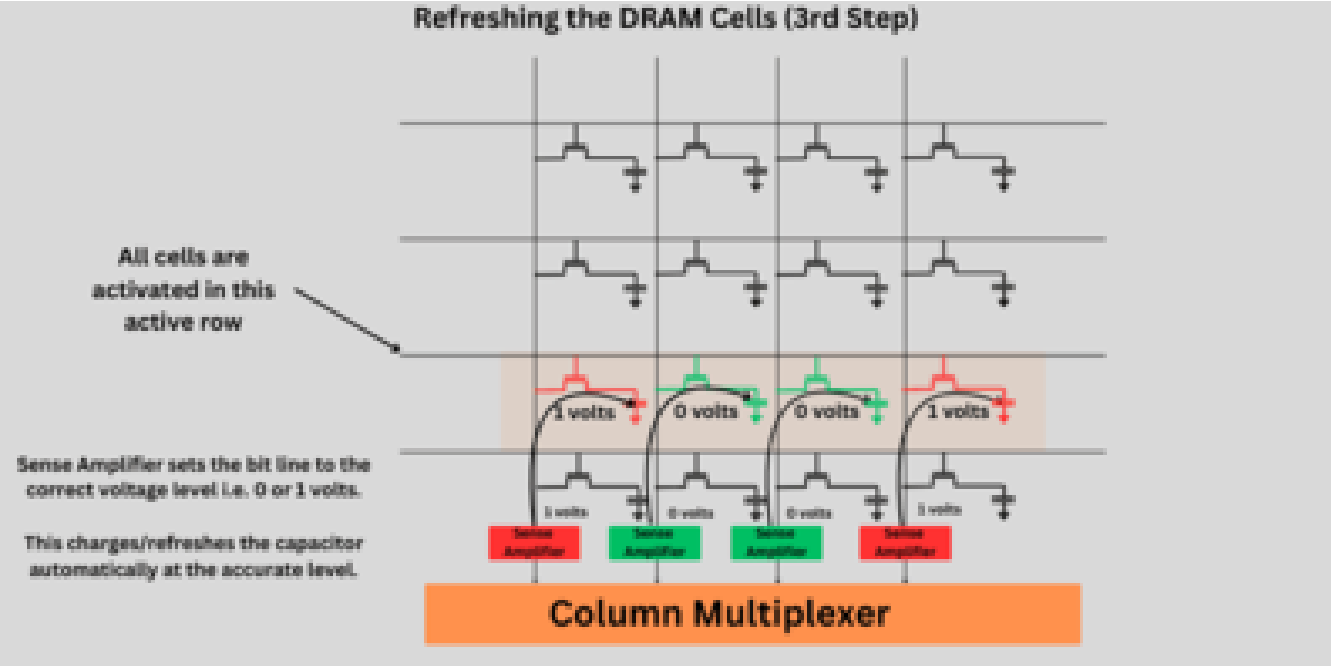
Refresh fáze 1



Refresh fáze 2

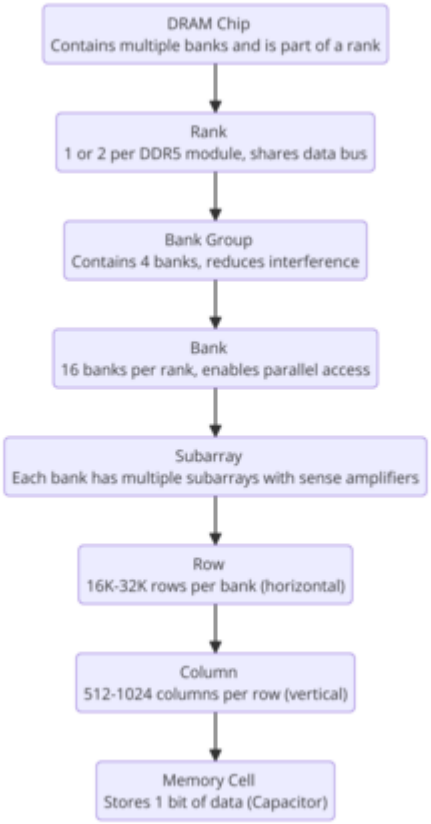


Refresh fáze 3



Hierarchie v DRAM pamětech

- pro DDR5 paměti



Vyrovnávací paměti (cache)

- **Vyrovnávací paměti (cache)** tvoří most mezi procesorem, který potřebuje rychle dostat data, a hlavní pamětí, která je sice velká, ale mnohem pomalejší než procesor. Pro programátora je obvykle zcela transparentní, přímo nemusí její existenci ani vnímat.

Účel cache pamětí

Jako vyrovnávací paměti

- Běžné velké operační paměti v posledních desetiletích ani zdaleka nedosahují vybavovacích časů, které by nezdržovaly CPU.
- *Cache tak tvoří most* mezi potřebou rychle dostupných dat a potřebou velké kapacity paměti.
- Velká a současně rychlá paměť by byla drahá, případně zcela nerealizovatelná.
- Proto se mezi procesor a hlavní paměť vkládá jako mezistupeň cache nebo dokonce více úrovní cache (L1, L2, …) za sebou.



Technická nemožnost udělat současně dost velkou a rychlou paměť plyne i z fyzikálních omezení: velkou kapacitu *nedostaneme dostatečně blízko k procesoru*, který data píše a čte.

Princip čtení přes cache

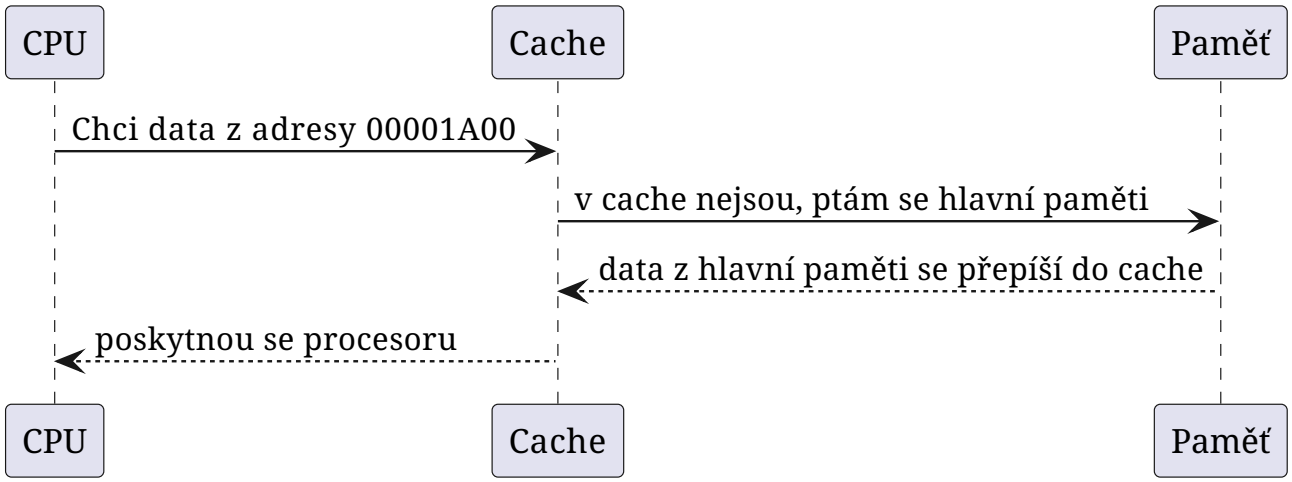
1. Procesor se zeptá cache, zda obsahuje požadovaná data.
2. Jsou-li dostupná v cache, jsou velmi rychle poskytnuta procesoru.
3. Nejsou-li dostupná, musejí se načíst z hlavní paměti, poskytnou procesoru a uložit do cache.
4. To po čase vyvolá potřebu vyřadit (přepsat) v cache méně používané údaje a uvolnit místo pro poptávanější data.



Algoritmy určující, která data lze z cache uvolnit, používají různé strategie, například se vyřadí *dlouho nepoužívaný údaj* (*Least Recently Used*, LRU).

Algoritmus čtení přes cache (případ *cache-miss*)

- Scénář, kdy kýžený obsah není v cache



Požadavky na zápis dat s cache

Klíčový požadavek:

- Udržet dlouhodobě *konzistenci* (shodu) mezi hlavní pamětí a cache, aby se nestalo, že program vezme z cache údaj neshodující se s hlavní pamětí.
- Při určitém systému práce s cache mohou být přechodně některé záznamy v cache čerstvější než v hlavní paměti; tehdy musejí být označeny "dirty" příznakem, aby se "časem" přepsaly do hlavní paměti a tím byl paměťový systém opět uveden do konzistentního stavu.

Postup při zápisu přes cache

1. Program chce zapsat údaj do paměti.
2. Údaj se nejprve vždy zapíše do cache.
3. Poté se postup liší dle používané strategie pro danou cache:

write-through

zápis do cache *vždy souběžně* se zápisem do hlavní paměti [1]

write-back

zápis do hlavní paměti se odkládá buďto *do uplynutí určeného času* nebo na chvíli, kdy se musí *řádek v cache uvolnit*, vyměnit — tehdy se změněný údaj už *musí* zapsat do hlavní paměti.

Principy vyhledávání a ukládání v cache

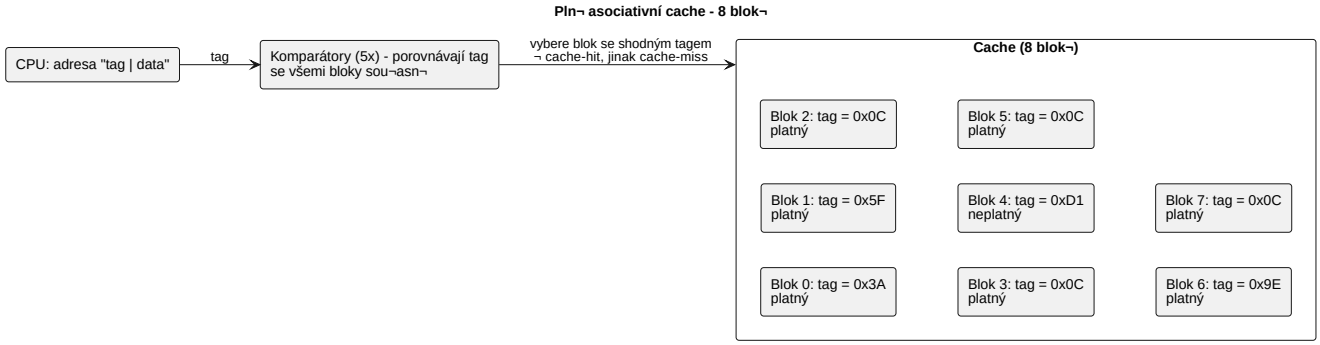
Plně asociativní	současně se porovnávají značky u všech záznamů v cache na shodu; kde se shoduje, odtud se údaj vrátí
N-cestně asociativní	kombinace asociativního a přímého adresování: z prostředních bitů adresy se určí jedna ze *skupin*, kde údaj *může být* (a nikde jinde); pak se ve vybrané skupině s N místy asociativním adresování vybere řádek, kde údaj *může být*, a zjistí, zda tam opravdu je.

Přímo adresovaná	nic se neasociuje, rovnou se sáhne na spočtenou adresu do cache a zkontroluje, zda je tam kýžený údaj

Plně asociativní

- je nutné *velké množství komparátorů*
- vzhledem k tomu, že se musí v každém řádku tabulky uchovávat celý tag, musí mít cache paměť velkou kapacitu, do které se tyto tagy ukládají a kterou není možné využít k uchovávání dat
- z těchto důvodů se plně asociativní paměti prakticky nepoužívají

Příklad: plně asociativní cache s osmi bloky



N-cestně asociativní

Zaadresování

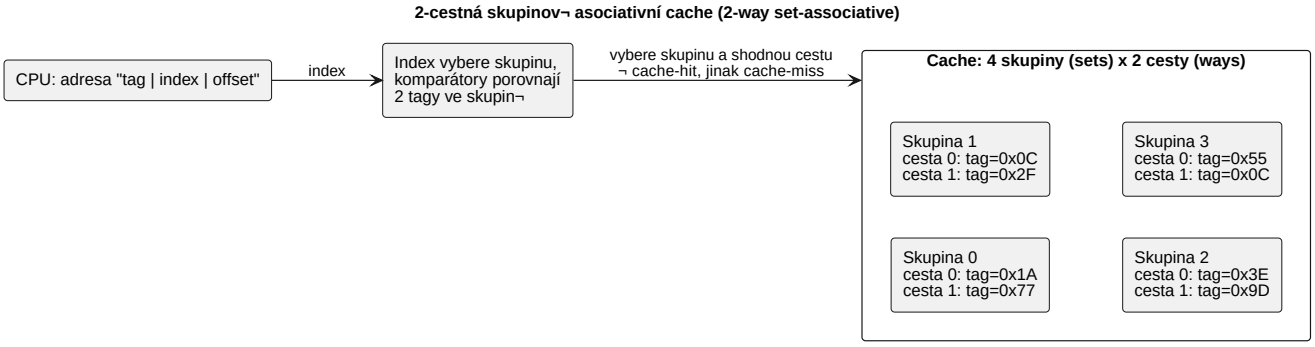
1. *N-cestně asociativní paměti* (*N-way associative*) pracují tak, že zadaná adresa se rozdělí na dvě části: *tag* a *adresa skupiny* (někdy též *adresa třídy*, v angličtině *set*).
2. Adresa skupiny je přivedena na *N* dekodérů (zařízení, které na základě vstupní hodnoty vybere jeden ze svých výstupů, na který umístí hodnotu log. 1, a na ostatní výstupy umístí hodnotu log. 0), které v každé tabulce vyberou jeden řádek.
3. Z těchto řádků se potom vezmou příslušné tagy a komparátorem se porovnají se zadaným tagem.

N-cestně asociativní (2)

Shoda?

1. Podobně jako u plně asociativních cache pamětí **pokud jeden z komparátorů signalizuje shodu**, je **informace v cache paměti** přítomna. Nastává cache-hit.
2. V opačném případě je nezbytné informaci hledat jinde cache-miss.
3. N-cestně asociativní paměti *částečně eliminují nevýhody* plně asociativních cache pamětí a v současnosti jsou nejpoužívanějším typem cache pamětí.

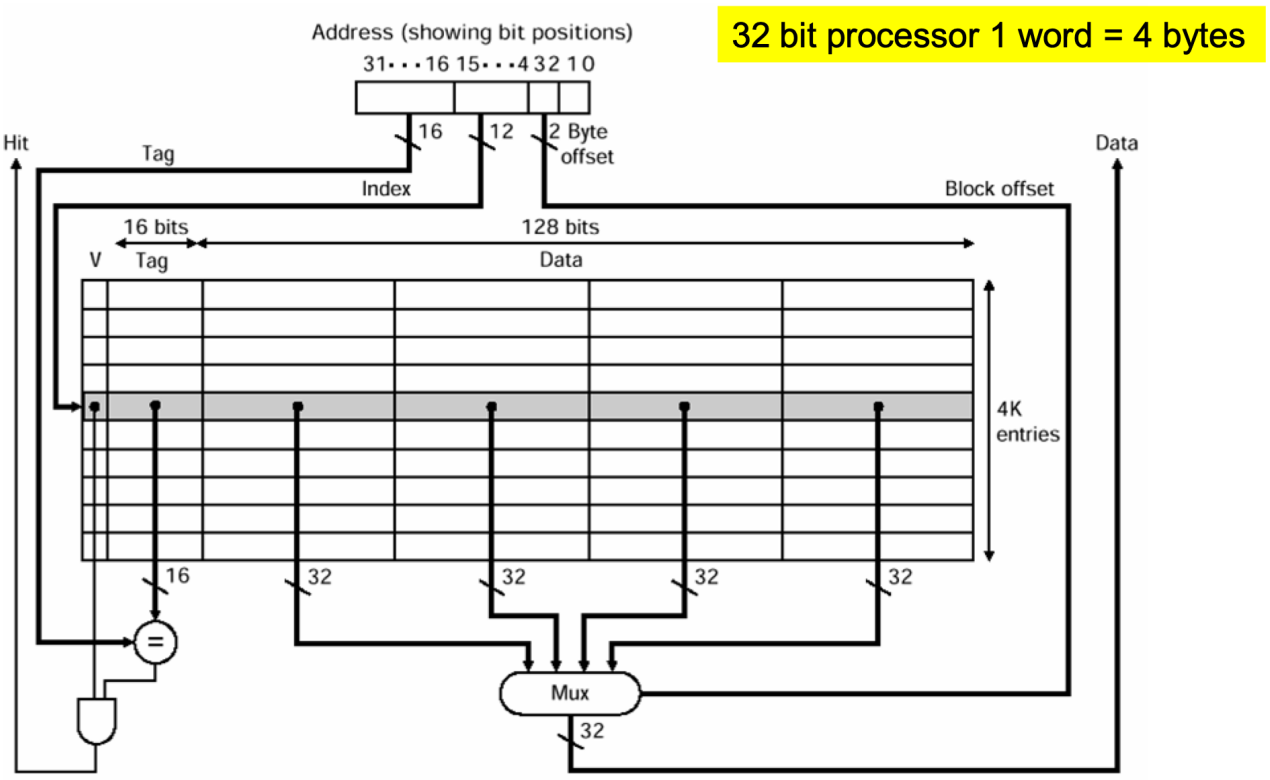
2-cestná cache



Přímo adresovaná

- Není vůbec asociativní, pro data "se jde rovnou".
- Vlastně jde o 1-cestnou cache.
- Princip jednoduše řečený na příkladu:
  1. Z adresy požadovaných dat se vezme střední část bitů, sáhne se na *jimi danou pozici, řádek* v cache.
  2. Je-li záznam na řádku *platný* a současně odpovídá-li jejich *tag* nejvyšší části adresy požadovaných dat, požadovaná data v cache jsou a je možné je vrátit.

Příklad: přímo adresovaná cache



Obr. 4. Přímo adresovaná cache, https://cw.fel.cvut.cz/b182/_media/courses/b35apo/en/lectures/05-e-cache.pdf

Strategie uvolňování cache

Cache je vždy až o několik řádů menší než hlavní paměť — uvolňování méně používaných záznamů je tedy nutností.

Možné strategie (politiky):

Random uvolní se náhodně zvolený řádek (záznam)

LRU (Least Recently Used) uvolní je nejdéle nepoužívaný řádek (musíme si někde pamatovat poslední přístupy)

LFU (Least Frequently Used) uvolní se nejméně používaný (opět si musíme něco pamatovat navíc)

ARC (Adaptive Replacement Cache) kombinace LRU a LFU

[1] Pro zápis do hlavní paměti se využije zápisový buffer kvůli relativní dlouhému trvání zápisu do hlavní paměti.

Trvalé paměti (úložiště)

Nyní se vrátíme k vnějším pamětem:

- Slouží jako úložiště pro *dlouhodobé ukládání* dat,
- *programového kódu* (operačního systému, knihoven i aplikací),
- *výměnu* dat (přenosná média),
- *archivaci* dat.

Technické provedení paměťových médií

vyměnitelnost média	• *výměnná* média (výměnné disky, diskety, CD, **pásky, flash-disky**...) • *nevýměnná* (pevná) média (**pevné disky, vestavěné SSD**)
fyzikální princip	• *magnetický* záznam (**magnetické pásky, pevné disky**, diskety) • *optický* záznam (**CD, DVD, Blue-ray**)
mechanické provedení	• *s točivými* díly (**HDD** - klasické pevné disky) • *bez pohyblivých* částí (**SSD disky**)

Pevné disky

- Pevné disky neboli *hard disks*, hard disky, slangově také *hárd*, *hadrdisk*, *hadr*, jsou velkokapacitní datová úložná zařízení.
- *Pevné* proto, že médium je *pevně zabudované* do jednotky disku, tedy nevýměnné.
- Nevýměnnost je relativní: často máme hlavně vedlejší pevné disky celé ve vysouvacím šuplíku.
- Vyjímatelnost umožňuje odpojení kvůli riziku hackingu a virů, dovoluje přenos nebo uschování na bezpečné místo.

Princip záznamu

- Je *magnetický*, podobně jako u pásek a disket.
- Pevný disk je mechanicky tvořen kotoučem nebo vícero souosými kotouči ("plotnami") s vrstvou z feromagnetického materiálu.
- Mezi kotouči se na ramínkách pohybují *záznamové hlavy* — podobně jako u disket.
- Kotouče rotují typicky rychlostí *5400-7200 rpm*.
- Oproti disketám je kapacita HD mnohonásobná.



Můj první HDD měl 20 MB; to není chyba, opravdu MB!

Solid-state Drives (SSD)



Principy a vlastnosti SSD

- Na polovodičovém principu **Flash-NAND** nebo **Flash-NOR**.
- Sofistikované algoritmy pro *obsazování místa* při zápisech: nepíše se čistě sekvenčně.
- Sofistikované *samoopravné kódy* kvůli chybovosti (u NAND).
- Postupné zpomalování při obsazenosti disku.

Negativa

- **Omezený počet zápisů** jedné buňky: nehodí se pro extrémní zátěže.
- **Omezená životnost záznamu**: bez dodávky elekřiny měsíce max. roky, pak se data pomalu vytrácejí.
- Násobně vyšší cena za TB než u HDD.

Použití

- Ideální do *přenosných počítačů*: SSD je mechanicky odolný, lehký, má nízkou spotřebu a vysokou

rychlost.

- V notebooku jsou obvykle stále pod napětím, odpadá tak postupná ztráta dat.
- Zejména vhodný pro *start systému* a spouštění častých aplikací.
- Vhodný také v kombinaci HDD (cena, kapacita) a SSD (rychlost).



Nevhodné pro zálohování a archivaci!

Základní princip

- Flash paměť využívá MOSFET *s plovoucím hradlem* (floating gate).
- Zápis a mazání probíhá elektricky pomocí tunelování náboje.
- Paměť je nevolatilní: pro udržení hodnot netřeba stále dodávat proud.

Flash paměti NAND a NOR

- Obě technologie uchovávají bity pomocí *náboje na plovoucím hradle tranzistoru*.
- Liší se ale způsobem propojení buněk a přístupovými metodami.

NOR Flash

- Paměťové buňky jsou propojeny **paralelně** (jako v NOR logice).
- Umožňuje přímý **náhodný přístup** (random access) **na úrovni bajtu**.
- Často používaná pro BIOS a firmware, kde je nutné spouštět kód přímo z paměti (XIP).
- **Pomalejší zápis a mazání**, vyšší cena za bit.

NAND Flash

- Buňky jsou spojeny **sériově do bitových řetězců** (NAND logika).
- Přístup se provádí **po stránkách** (page) – typicky 2 až 16 KB.
- Mazání je možné pouze po blocích (např. 128 stránek = 512 KB).
- Vysoká hustota, nízká cena za bit – ideální pro SSD, karty a USB.
- Vyžaduje řadič pro ECC, přemapování a wear leveling.

Srovnání NAND vs. NOR (1)

Vlastnost	NOR Flash	NAND Flash
Topologie buněk	Paralelní	Sériová
Přístupová jednotka	Bajt	Stránka (např. 4 KB)
Podpora XIP	Ano	Ne

Vlastnost	NOR Flash	NAND Flash
Hustota	Nižší	Vyšší
Rychlost čtení	Vysoká při malém rozsahu	Efektivní při sekvenci

Srovnání NAND vs. NOR (2)

Vlastnost	NOR Flash	NAND Flash
Rychlost zápisu	Pomalejší	Rychlejší
Mazání	Po blocích	Po větších blocích
Spolehlivost	Vyšší	Vyžaduje ECC
Cena za bit	Vysoká	Nízká
Počet zápisových cyklů	10^5	10^3 .. 10^4
Použití	Firmware, BIOS	SSD, paměťové karty

Adresování NOR a NAND

NOR Každý bajt je přímo adresovatelný.

NAND Přístup se provádí přes bloky → stránky → buňky.



NAND vyžaduje sofistikovaný řadič pro překlad adres, kontrolu chyb (ECC) a rozložení zápisů (wear leveling).

Shrnutí

Technologie NOR Bootloadery, mikrokontroléry; výhoda: přímý přístup (XIP)

Technologie NAND Ukládání dat (mass storage); výhoda: cena, kapacita

Porovnání HDD a SSD (1) – výkon

Vlastnost	HDD	SSD
Rychlost čtení/zápisu	Pomalejší (~100 MB/s)	Rychlejší (500 MB/s až 7000+ MB/s)
Přístupová doba	Vysoká (~10 ms)	Nízká (~0.1 ms)
Mechanické části	Ano (rotační části)	Ne (zcela elektronické)

Porovnání HDD a SSD (2) – provoz

Vlastnost	HDD	SSD
Spotřeba energie	Vyšší	Nižší
Hlučnost	Slyšitelný hluk	Tichý provoz
Odolnost vůči nárazům	Nízká (citlivý na otřesy)	Vysoká (žádné pohyblivé části)
Životnost (přepisů)	~1-2 miliony hodin MTBF	Omezený počet cyklů

Porovnání HDD a SSD (3) – nasazení

Vlastnost	HDD	SSD
Kapacita	Vyšší (až 20 TB)	Nižší (běžně 0.25-4 TB)
Cena za TB	Mezi 1-20 TB je nejvýhodnější: kolem $10	Nejméně $40, a to jen u menších 1-4 TB
Vhodné pro	Archivace, velká data	Rychlý běh systému, aplikací

Srovnání cen za TB



Info k cenám disků HDD, SDD, flash a optických disků a pásek: https://diskprices.com/

Magnetické pásky



Historie magnetického záznamu

- Technicky je nosičem tenká pevná **páska** ze syntetických materiálů s **feromagnetickou vrstvou**.
- Ta se v magnetofonu pohybuje definovanou rychlostí v kontaktu se *čtecí a záznamovou hlavou*.
- Hlava ji **střídavě zmagnetovává** působením magnetického pole.
- Při čtení zase naopak změny magnetického toku v pohybující se pásce **indukují proud v hlavě** a ten se dále zesiluje a zpracovává.

Nejprve zvukové pásky

- Pásky s magnetickým záznamem se nejprve používaly pro **analogový záznam zvuku** a to jim vydrželo až do devadesátých let, kdy byly nahrazeny záznamem digitálním na CD.
- Pásky byly ovšem také jedním z prvních médií na dlouhodobé uchovávání dat, na počítačích (od roku 1951).

Způsob záznamu a přístupu

- Páska je vždy předurčena k tomu, aby data byla zaznamenávána **sekvenčně** "za sebou".
- Zásadně se tedy nehodí, potřebujeme-li častý přímý přístup někam "doprostřed" nebo střídavě na různých místech.
- Rovněž obvykle **nedokážeme přemazat/přepsat** jen třeba jeden vybraný soubor. Musíme psát dále na volné místo a přemazávat vcelku.
- Přístup ke konkrétnímu údaji je o **několik řádů pomalejší** než například z diskových médií.

Provedení klasických páskových médií

- Pásky měly dříve výlučně formu v dlouhých metrážích navinutou na kotoučích, později i na kazetách s hracími dobami obvykle do 90 minut.



Obdobná situace byla u videozáznamů: páskové (kazetové) videokamery se používaly až do doby technické a ekonomické dostupnosti velkých flash pamětí.

Datové pásky

- Pásky jsou zřejmě nejdéle sloužící paměťové **médium pro levné dlouhodobé ukládání**

(zálohování, archivaci) dat — a pravděpodobně jím ještě dlouho zůstane.

- Podstatnou výhodou je **vyjímatelnost páskového média** — lze je pro archivaci schovat do trezoru, rozmístit na odlehlá místa, atd.
- Umožňuje též rychlý **transport dat** i na velké vzdálenosti: sbalit arzenál pásek do kufru a přepravit ho letadlem je pro velké objemy způsob významně rychlejší než posílání po seberychlejší síti přes půl světa.
- Moderní pásky mohou mít běžně kapacity **v řádech terabajtů na kus**.

Moderní standard LTO

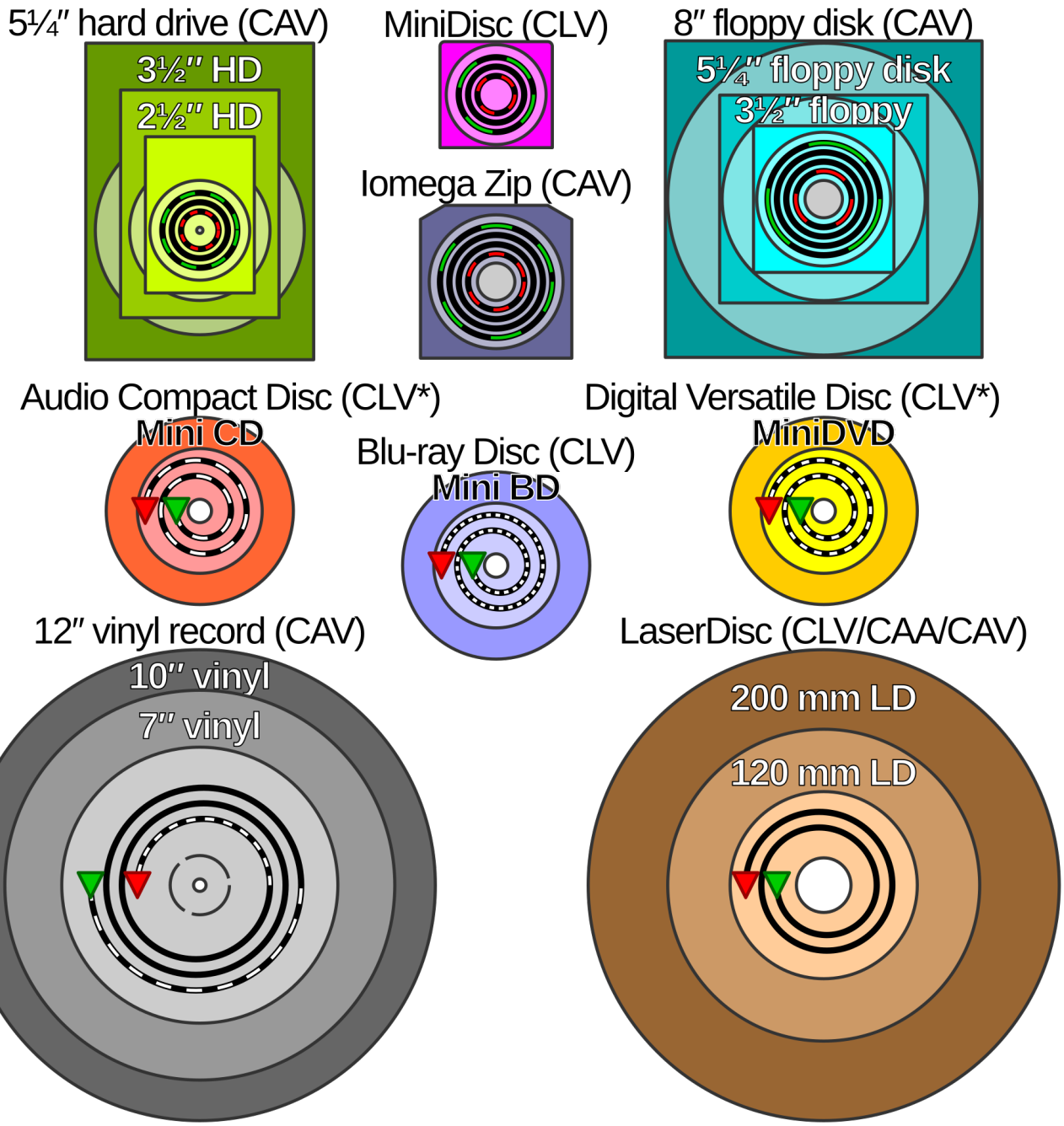
- *Linear Tape-Open* (LTO), a.k.a. *LTO Ultrium* formát je typ moderních páskových *kartridží* (kazet, *cartridge*) zavedený společnostmi Hewlett Packard Enterprise, IBM a Quantum.
- Nejnovější generace **LTO-10** z roku 2025 nabízí datovou kapacitu kartridže až 3O TB dat.
- Páskové jednotky mívají integrovanou aspoň nějakou kompresi dat, takže se obchodně nabízí kapacita až 75 TB ("with industry-standard 2.5x compression").
- Pásky LTO mají šířku 1/2" (12,65 mm) a celá kartridž 102,0 × 105,4 × 21,5 mm.

Příklad: Kartridž LTO-3



Obr. 5. LTX400G od společnosti Sony, zdroj: https://en.wikipedia.org/wiki/Linear_Tape-Open

Optické disky: CD, DVD, Blue-ray



Co jsou CD?

- *Compact Disc*, česky též *kompaktní disk*, lidově *cédéčko* nebo *cédo* je vysokokapacitní úložné médium o standardní kapacitě 700 MB dat.
- Jde o vynález společností **Philips** (vyvinul mechaniku) a **Sony** (digitální kódování), určený původně na záznam digitalizované hudby, jako náhrada vinylových gramodesek.
- Při běžném formátu záznamu a vzorkování zvukového záznamu (44.1 kHz * 16 bitů, stereo) odpovídá datová kapacita až *80 minutám nekomprimované hudby*.
- Komerčně na hudbu byl uveden v roce 1982, v ČSSR kolem poloviny 80. let.

Mini-CD

- Běžné CD má 12 cm průměr,
- mini-CD používané dříve typicky "na rozdávání" jako nosič reklamních informací, videí, sborníků v elektronické formě před rozšířením sítí a flash-disků mělo průměr 8 cm.



Organizace záznamu

- Systém samotného zápisu bitů je dán typem CD (-ROM, -R, -RW,).
- Bity jsou ukládány na dráze odvíjející se po spirále od středu k okraji disku.
- Nemá tedy organizaci na stopy a sektory jako pevné disky a diskety.
- Organizací ukládání jde tedy o médium sekvenční a vhodné pro kontinuální čtení velkých souborů, ačkoli z logického pohledu můžeme mít na CD filesystém jako na jiných discích.

Nezapisovatelné CD (CD-ROM)

- Data jsou *vylisována* při výrobě jako:
  - `pits` (důlky) a `lands` (rovné plochy)
- Laser při čtení detekuje změny v odrazu světla.
- Disk je pouze pro čtení — nelze zapsat ani přepsat.

Zapisovatelné CD (CD-R)

- Obsahuje *organickou barvicí vrstvu*.
- Při zápisu laser trvale *změní optické vlastnosti* vrstvy.
- Laserem "vypálené" oblasti napodobují `pits`.
- Pouze jeden zápis, ale čtení je jako u CD-ROM.

Přepisovatelné CD (CD-RW)

- Využívá *slitinovou vrstvu* (např. Ag-In-Sb-Te).
- Pomocí laseru lze měnit mezi:
  - *krystalickým stavem* (více odrazivý)
  - *amorfním stavem* (méně odrazivý)
- Lze přepsat typicky až 1000×

DVD

- *Digital Video Disc*, *Digital Versatile Disc* je následníkem CD s významně vyšší kapacitou až 4,7 GB .
- Používáno komerčně od roku 1995.
- Motivace: záznam videa → digitální náhrada do té doby populárních videokazet.
- Běžná stopáž videa při běžné (MPEG-2) kompresi jsou 2 hodiny.



DVD

- **DVD-R** a **DVD+R** se dají DVD rekordérem jednorázově zaznamenat.
- Přepisovatelná DVs (**DVD-RW**, **DVD+RW** a **DVD-RAM**) lze zapisovat a mazat mnohokrát.
- Některé typy DVD jsou *vícevrstvé* a/nebo *oboustranné*, kapacita pak naroste až na 17,1 GB (standard DVD-18).

Srovnání CD a DVD

Laser	• CD používá infračervený laser (780 nm), velikosti pitu minimálně 0,83 µm • DVD používá červený laser (650 nm) → menší bod → hustší zápis
Záznamová vrstva	• DVD má tenčí vrstvy • DVD má menší důlky (minimálně cca 0,40 µm)
Formáty	• CD-ROM, CD-R, CD-RW • DVD-ROM, DVD-R, DVD+R, DVD±RW, DVD-RAM

Blue-ray (BR)

- Je opět vynálezem firmy Sony, přesněji *Blu-ray Disc Association (BDA)*, vedené Sony.
- Motivace: filmy ve Full-HD (H.264) kvalitě.
- Pro zvýšení hustoty záznamu se musí používat modrý laser (405 nm), odtud i obchodní označení.
- Kapacita je výrazně vyšší, nejméně 25 GB, u vícevrstvých až 128 GB.
- Opět existují varianty: **BD-ROM**, **BR-R** (jednou zapsatelné), **BD-RE** (přepisovatelné), **BD-XL**.

Trvanlivost

- Při zavedení CD/DVD byly obavy, že médium nebude mít delší životnost než řádově roky.
- U lisovaných CD/DVD-ROM odhad až 100 let.
- CD-R 5-30 let (organické barvivo časem degraduje)
- CR-RW 5-10 let, až 1000 přepisů



Vlastní zkušenost: CD-R i po 25 let bez potíží čitelné — tedy máte-li kde jej přečíst :)

Diskety

- Diskety (*floppy disks*, také *pružné disky*) jsou dnes již takřka nepoužívaným magnetickým médiem pro záznam dat.
- Materiál diskety i princip záznamu je podobný jako u pásky s tím hlavním rozdílem, že k datům je přímý přístup a odpadá nutnost procházet sekvenčně jako u pásky.
- Typické kapacity disket běžných formátů disket 8", později 5 1/4" a 3,5" byly do 1,44 MB, tedy o více řádů menší než pevné disky, ale i optické disky CD, DVD.

Organizace záznamu na disketu

- Data jsou ukládána do *stop* ve tvaru soustředných kružnic a *sektorů* jakožto výsečí těchto kružnic.
- Čas potřebný k načtení (nebo zápisu) dat sestává z doby *vystavení hlavy* na příslušnou stopu a *vyčkáním*, než se disk pootočí na příslušný sektor.
- Na rozdíl od pevného disku se disketa roztočí až při potřebě čtení či zápisu.
- Disketa se pohybuje *běžně ve vzduchu* a nikoli jako kotouče pevného disku zapouzdřené v bezprašném prostředí diskové jednotky.
- To **omezuje spolehlivost**, trvanlivost i možnou hustotu záznamu na disketě.

Příklady disket



Na další kapitolu: Pokročilé architektury

HTML · Slides · PDF | Přehled předmětu