

# Paměťový podsystém počítače

---

- typy paměťových systémů počítače
- virtuální paměť
- stránkování
- segmentace
- rychlá vyrovnávací paměť

# Organizace paměťového systému počítače

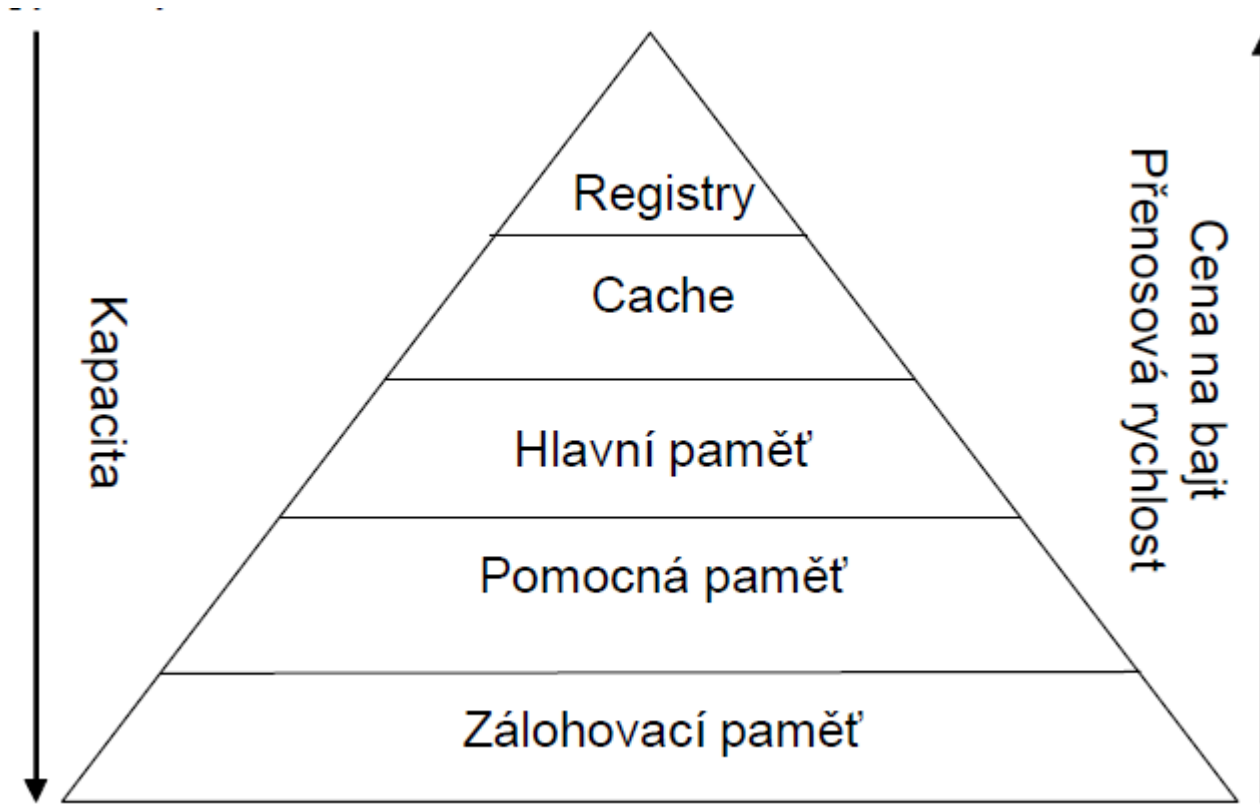
- Paměťová hierarchie ... několikaúrovňové usporádání pamětí různých typů (kapacit a rychlostí) s cílem dosáhnout optimálního poměru výkon x cena
- Cena paměti je cca přímo úměrná kapacitě nepřímě době přístupu (rychlosti)

# Paměťová hierarchie

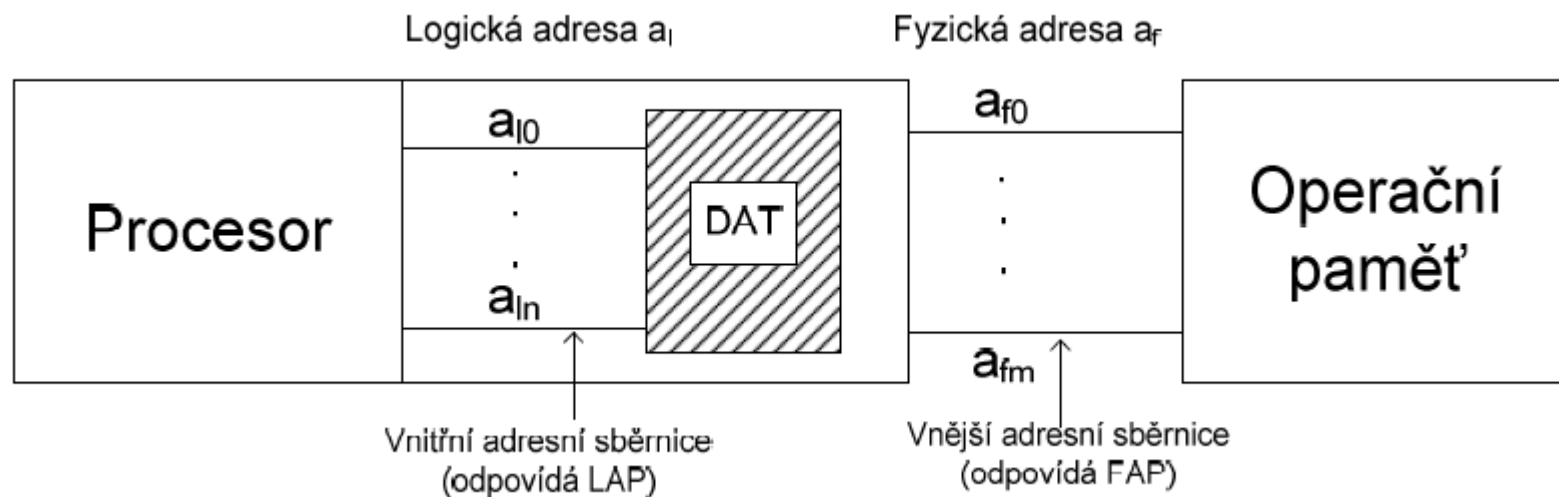
typ paměti	Typická realizace	Doba přístupu	Kapacita
registry	klopné obvody	jednotky ns	desítky-stovky B
vyrovnávací paměť	statická RAM	10-20ns	stovky kB-jednotky MB
hlavní paměť	dynamická RAM	50-70ns	desítky-stovky MB
vnější paměť	pevný disk	5-15ms	Jednotky-desítky GB
záložní paměť	optický disk mag. páska ZIP drive	stovky ms- Stovky s	Stovky GB- jednotky TB

CENA !!!

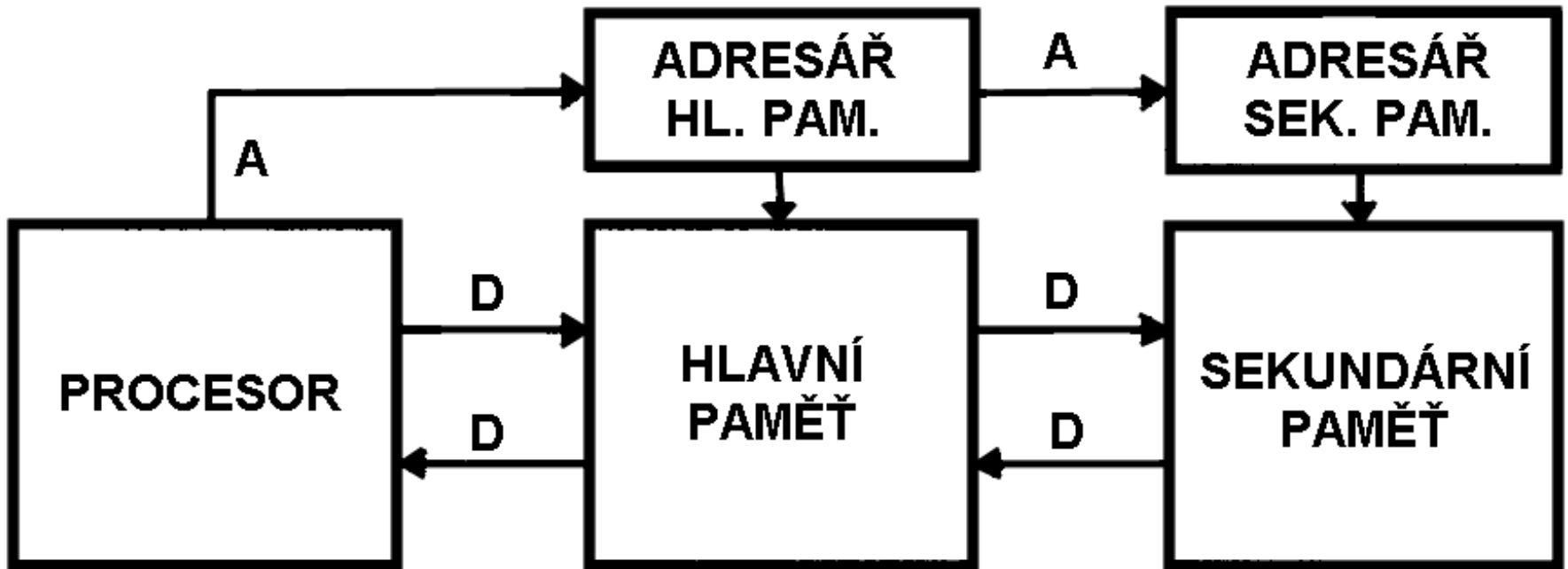
# Rozdělení paměťového subsystému



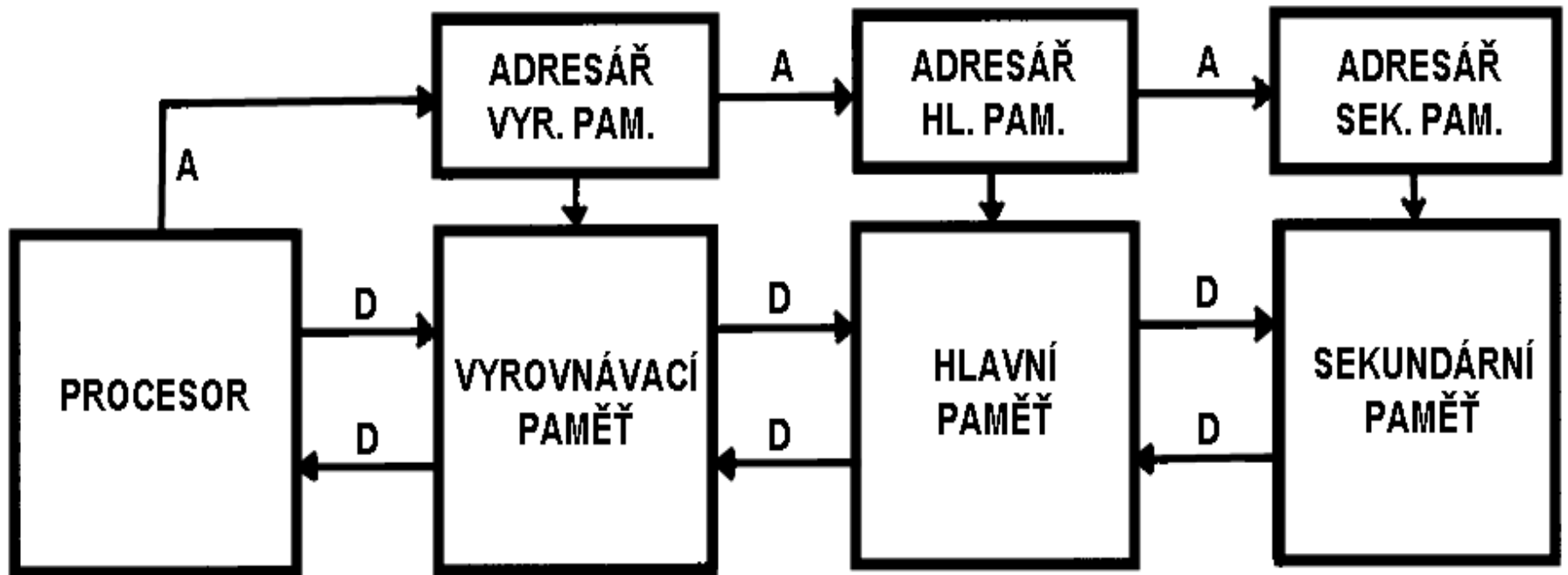
# Dynamická transformace adres



# Schéma dvouúrovňové paměti



# Trojúrovňový paměťový systém

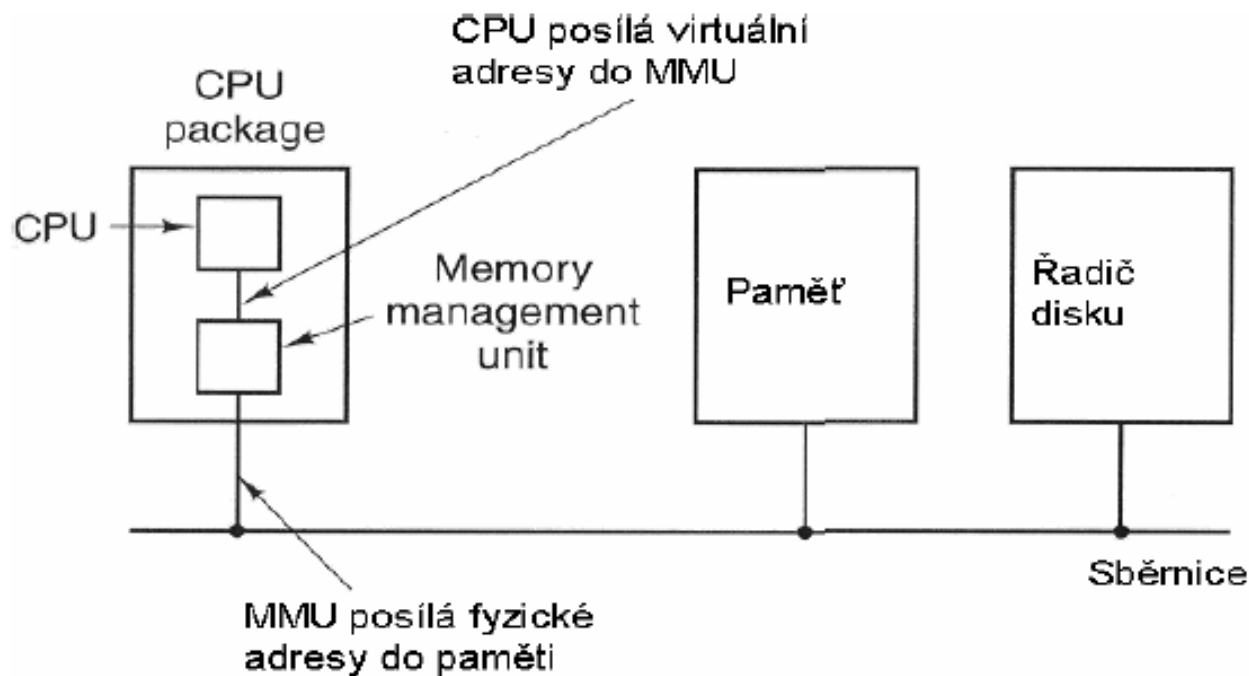


# Virtuální paměť

- systém několika pamětí s různými parametry (kapacita, rychlost), řízený tak, aby vytvářel paměťové prostory potřebné velikosti pro program a data
- umožňuje realizaci jednoho nebo několika logických (virtuálních) adresových prostorů, kde každý může být větší než skutečná kapacita hlavní paměti
- hlavní paměť ... fyzický paměťový prostor
- logické adresové prostory jsou ve skutečnosti ve vnější paměti
- části programů a data jsou přesouvány do hlavní paměti, požaduje-li k nim procesor přístup

# Virtuální paměť

- v hlavní paměti jsou jen ty programy a data, s kterými procesor právě pracuje
- lze přemístit části programu bez nutnosti je znovu překládat
- zajištění ochrany dat před neoprávněným přístupem a modifikací
- práce s logickými adresami ne absolutními
- hlavní paměť se adresuje fyzickými adresami
- překlad logických adres na fyzické zajišťuje mechanismus virtuální paměti (Memory Management Unit MMU)



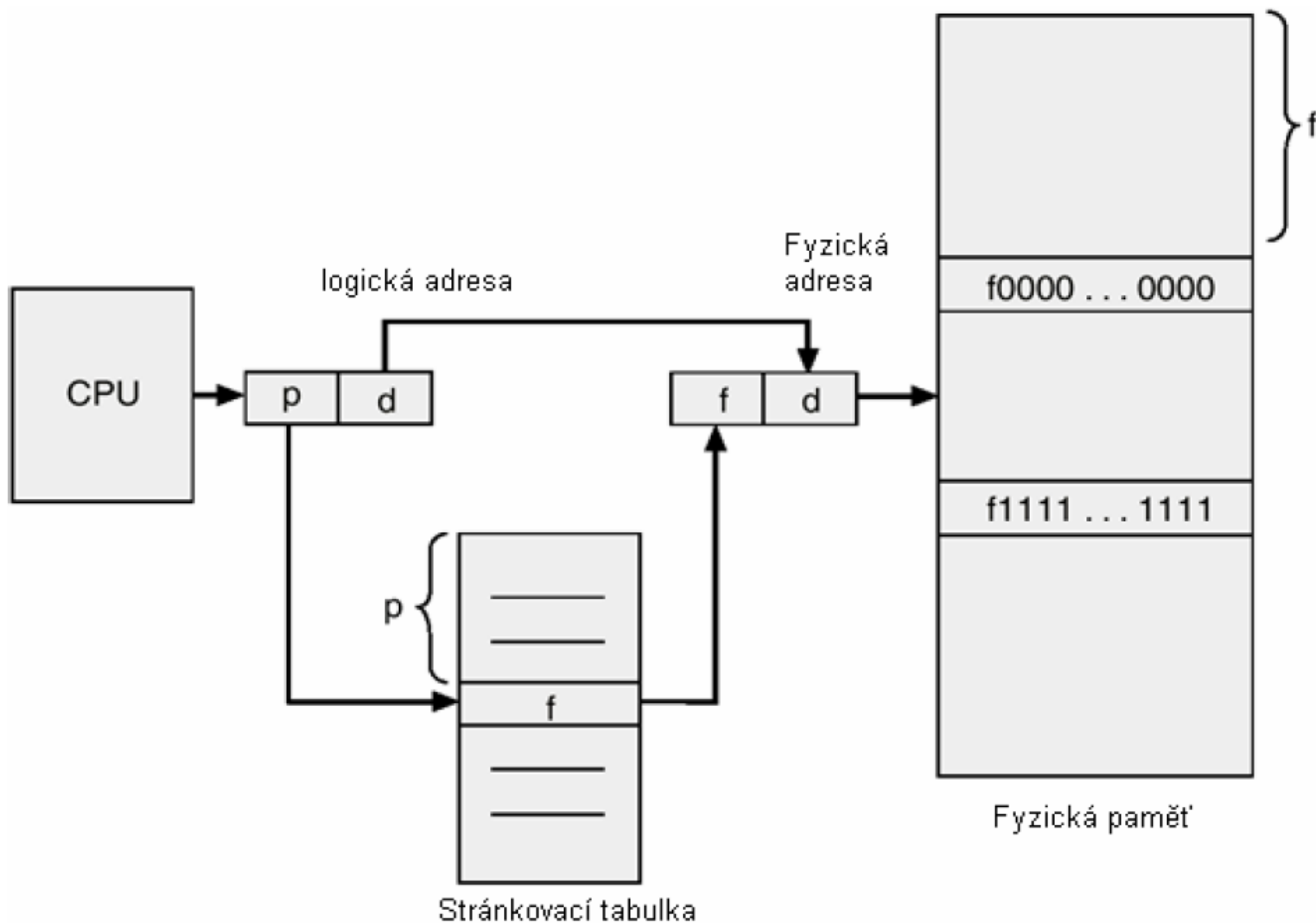
# Stránkování

- **Logický** adresový prostor je rozdělen na úseky pevné délky - **stránky** (logické stránky)
- **Fyzický** adresový prostor je rozdělen na stejně velké úseky - **stránkové rámce** (fyzické stránky)
  - ↗ Logický adresový prostor je realizován ve vnější paměti. Data (úseky programu) se přesouvají do hlavní paměti po jednotlivých stránkách, jsou-li v průběhu výpočtu požadována a pokud se příslušná stránka již v paměti nenachází.
- Překlad (určení kam se stránka do hlavní paměti přesune) používá datovou strukturu ... **tabulku stránek**

# Tabulka stránek

- je uložena v hlavní paměti
- obsahuje pro každou logickou stránku jednu položku
- položka obsahuje informaci, zda se daná stránka nachází v hlavní paměti a pokud ano, tak kde (v kterém stránkovém rámci)

# Stránkovací mechanismus



# Realita, problémy

- tabulka stránek musí obsahovat 1 položku pro každou stránku v logickém adresovém prostoru, i když není použita
- při mnohem větším logickém prostoru proti velikosti hlavní paměti, může tabulka stránek zabírat velkou část hlavní paměti

Příklad:

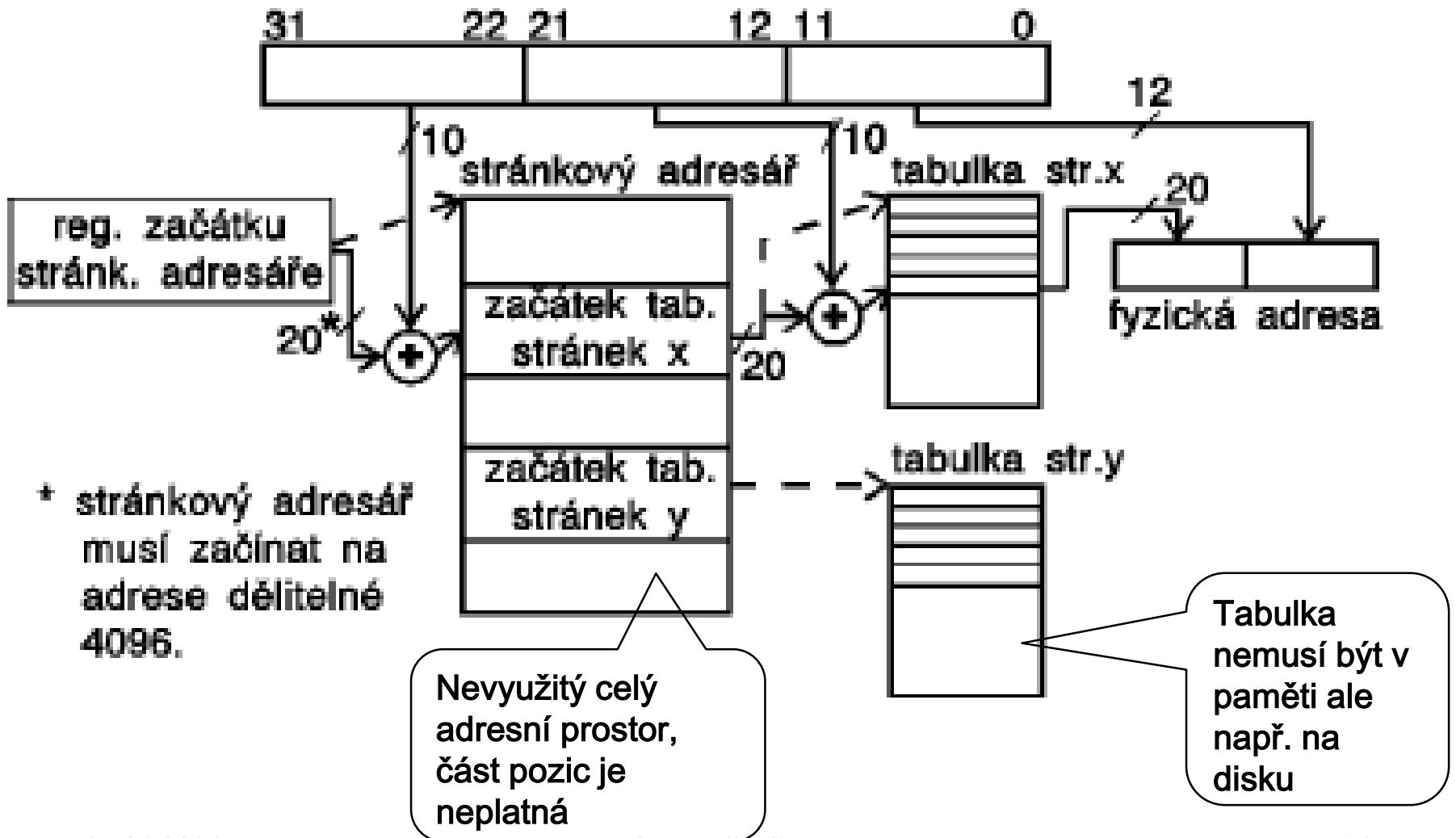
virt. paměť	4GB	.....	32b
hlavní paměť	4MB	.....	22b
velikost stránky	4kB	.....	12b
LA	┌ 20 ────┤		12 ────┘
FA	┌ 10 ────┤		12 ────┘

Tabulka stránek má  $2^{20}$  položek

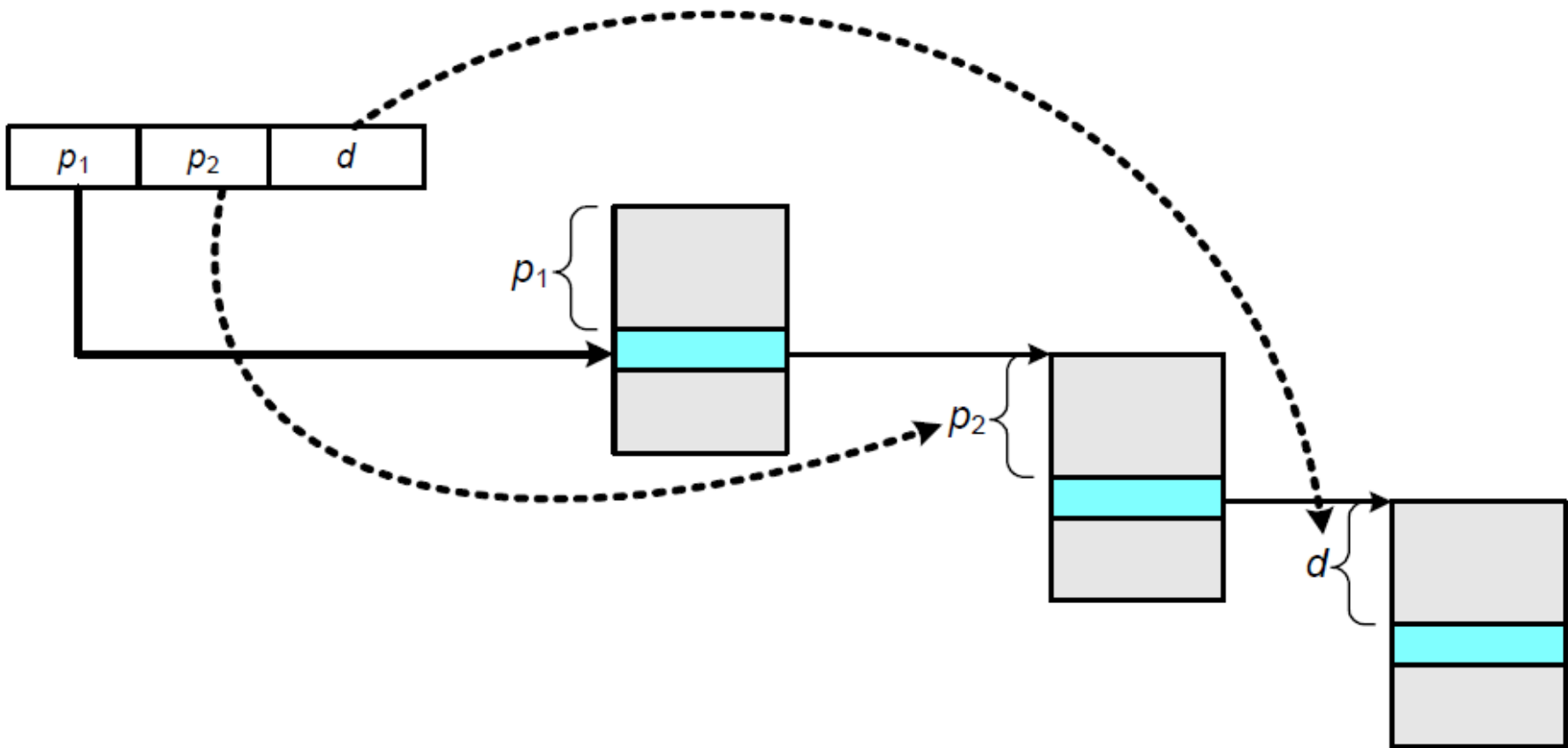
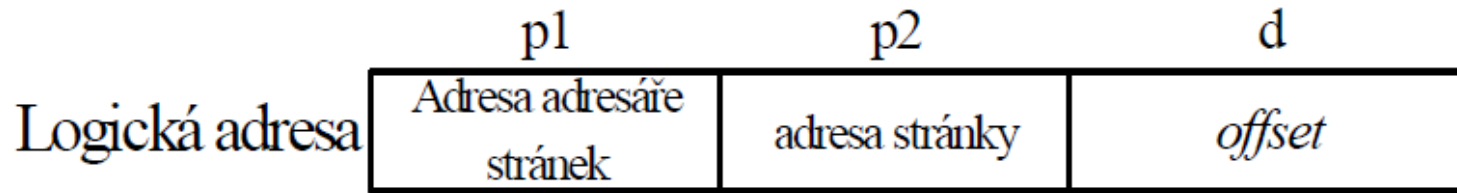
1 položka ... 10 bitů horní část fyzické adresy } 2B pro  
1 bit platnost } 1 položku

⇒ tabulka stránek zabírá 2MB - nepřipustné.

# Dvouúrovňová organizace tabulky stránek



# Dvouúrovňová tabulka stránek



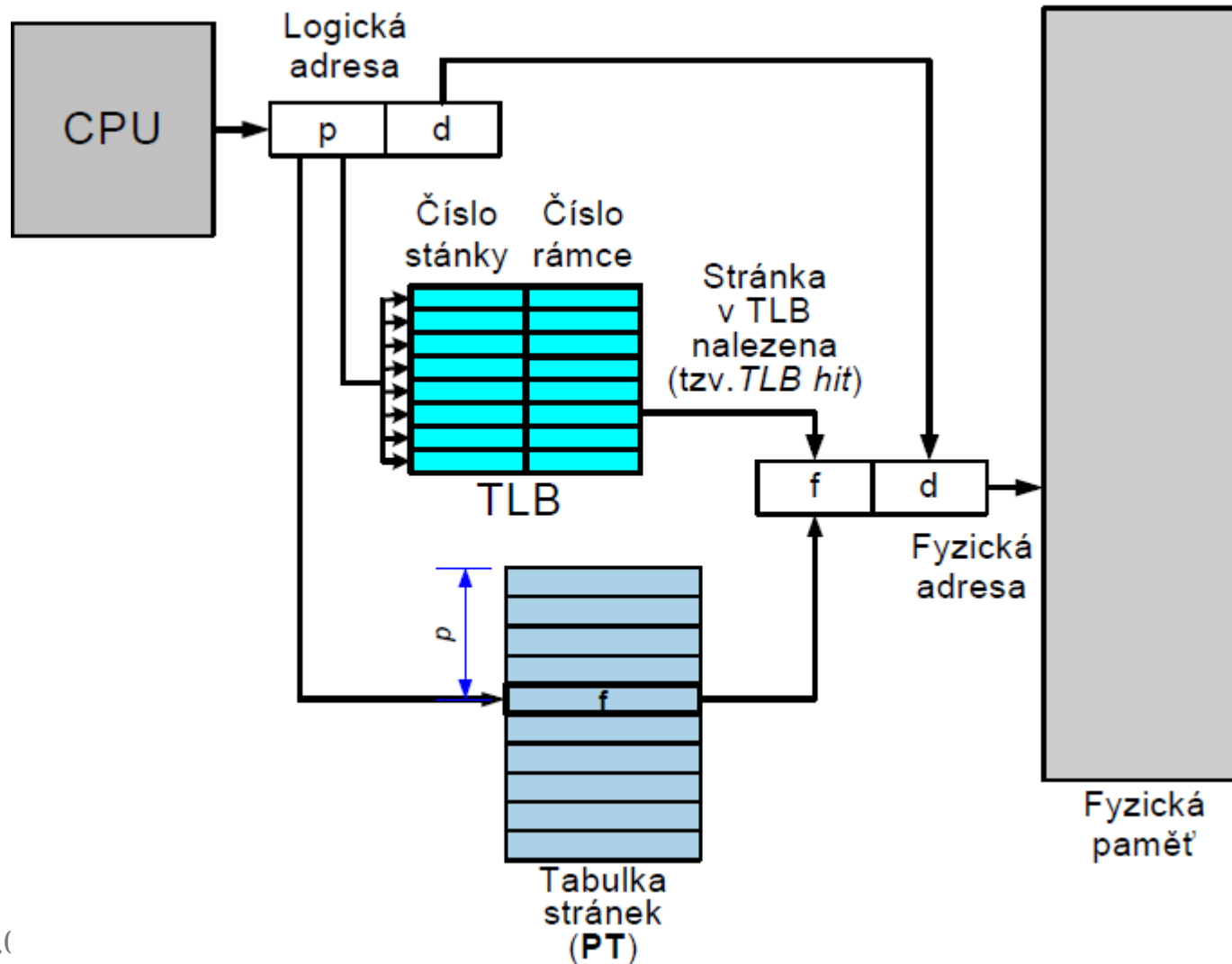
# Obsah tabulky stránek

- horní část fyzické adresy
- příznak přítomnosti stránky v hlavní paměti
- příznak změny dat ve stránce (zda byla po dobu přítomnosti v HP do stránky zapisováno) ... **Dirty bit**
- další bity, např. určující, zda je vhodné stránku přepsat (vyhodit z hlavní paměti, podle toho jak a kdy byla použita)
- ....

# ***TLB cache (Translation lookaside buffer).***

- Tabulka stránek obvykle umístěna částečně v hlavní paměti a částečně na disku – snížení výkonnosti hledání v tabulce
- TLB - uchování právě načtených řádků tabulky stránek
- plně asociativní paměť s řádově desítkami vstupů
- Pokud se rámce nenaleznou, hledá se v tabulce stránek

# TLB cache



# Stránkovací mechanismus

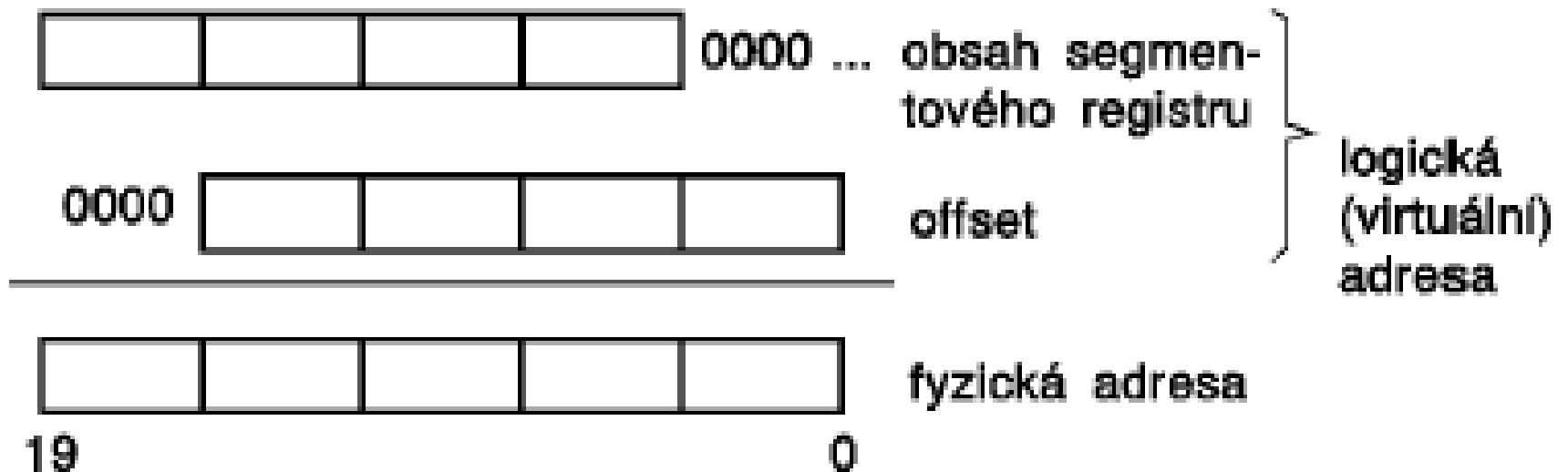
- je-li stránka přítomna v hlavní paměti, přeloží se logická adresa na fyzickou (příznak přítomnosti stránky v HP)
- není-li stránka přítomna, vyvolá se přerušení. Přerušovací mechanismus vyvolá načtení stránky z vnější paměti.
- pokud při načítání není volný žádný stránkový rámeček v HP, je třeba nějaký uvolnit ... přesun vhodné stránky do vnější paměti (např. nejdéle nepoužité, viz další bity, sl. 11 ...)
- nebylo-li do stránky zapisováno (Dirty bit), nemusí se přesouvat do vnější paměti.
- V současnosti se používá většinou tříúrovňové stránkování

# Segmentace

- segmenty jsou funkčně samostatné části programu **proměnné délky**, které lze do hlavní paměti zavádět v případě potřeby
- adresy v segmentu jsou relativní vůči začátku (tzv. bázi) segmentu ... umožňuje přemístitelnost segmentů v HP
- logická adresa se skládá z báze segmentu a offsetu (posunutí)
- báze segmentu je
  - ↗ v segmentovém registru
  - ↗ tabulce popisovačů segmentů
- Setřásání segmentů

# Segmentové registry

Příklad z Intel 80x86 .... 16 bitové registry, 20 bitová adresová sběrnice



<b>Stránkování</b>	<b>Segmentace</b>
FAP je dělen na elementy (chunks) pevné délky – rámce	FAP není rozdělen
Program (LAP) je kompilátorem (MMU) dělen na stránky	Segmenty LAP určuje programátor
Interní fragmentace uvnitř rámce	Žádná interní fragmentace
Žádná externí fragmentace	Externí fragmentace
OS udržuje tabulku stránek pro každý proces s určením, které rámce s nimi souvisí	OS musí udržovat tabulku segmentů pro každý proces. Určuje zaváděcí adresu a délku každého segmentu
OS udržuje seznam volných rámců	Operační systém udržuje seznam volných děr
Procesor používá číslo stránky a offset k výpočtu fyzické adresy	Procesor používá číslo segmentu a offset k výpočtu fyzické adresy
Všechny stránky procesu nemusí být při běhu procesu ve FAP, zavádění stránek probíhá podle potřeby	Všechny segmenty se zavádí do FAP podle potřeby
Zavedení stránky do FAP může způsobit požadavek zápisu některé stránky na disk	Zavedení segmentu do FAP může způsobit zápis některého segmentu na disk

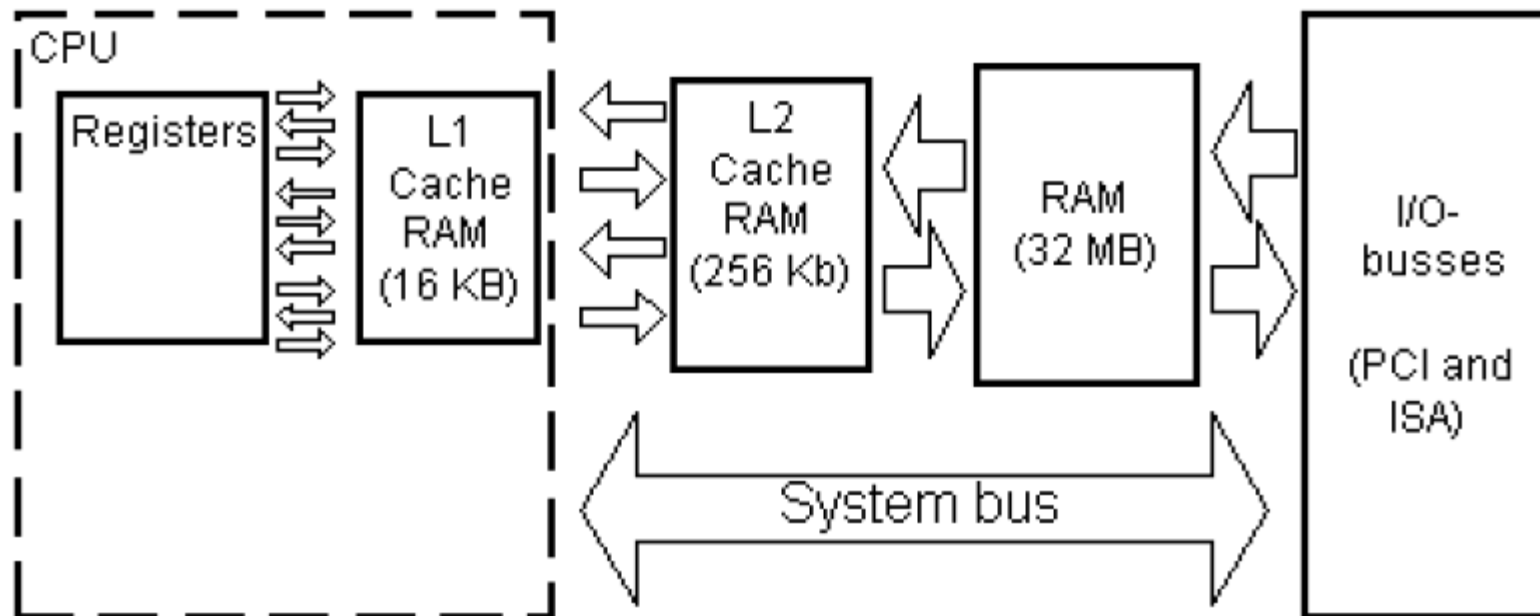
# Paměti CACHE

- Softwarová - vyrovnávací paměť pro pomalé (vzhledem k rychlosti operační paměti a procesoru) vnější paměti (pevný disk počítače),
- Hardwarová cache (mezipaměť) – pro komponenty s větší přenosovou rychlostí, speciální paměťové obvody přímo na komponentě. (cache v procesoru), může poskytovat data opakovaně

# Hw paměť' CACHE

- "malá" rychlá paměť' zařazená mezi procesor a hlavní paměť'
- využití asociativního přístupu k položkám
- obsahuje kopie nejčastěji používaných položek hlavní paměti
- realizována SRAM
- synonyma .... cache, buffer memory, skrytá paměť' (SP), vyrovnávací paměť', ...

# L1 (interní) a L2, L3 (externí) paměť cache



Od roku 2009 se začala používat L3 cache i v běžných procesorech (Intel Core i7, AMD Phenom a další)

## zápis ...

- pokud položka v cache není zapíše se zpravidla jen do hlavní paměti
- pokud je, postupuje se různými způsoby pro různé typy cache pamětí:
  - u **write-through** cache paměti v případě zápisu procesoru do cache paměti dochází okamžitě i k zápisu do operační paměti. Procesor tak obsluhuje jen zápis a o další osud dat se stará cache paměť.
  - u **write-back** cache paměti jsou data zapisována do operační paměti až ve chvíli, kdy je to třeba, a nikoliv okamžitě při jejich změně. K zápisu dat do operační paměti tedy dochází např. v okamžiku, kdy je cache zcela zaplněna a je třeba do ní umístit nová data. Tento způsob práce cache paměti vykazuje oproti předešlému způsobu vyšší výkon.

**čtení ...** zahájení cyklu čtení současně z cache i z hlavní paměti. Pokud se položka v cache nalezne, cyklus hlavní paměti se nedokončí.

V opačném případě se přečtou data z hlavní paměti (a obvykle zároveň uloží do cache)

# Hodnocení cache

- **Hit**: zásah, požadovaná data jsou nalezena na vyšší úrovni paměti
- Hit rate = četnost zásahu, podíl úspěšných přístupů do paměti vyšší úrovně k celkovému počtu přístupů
- **Hit time** = vybavovací doba vyšší úrovně, zahrnuje: samotný vybavovací čas pamětí vyšší úrovně
- **Miss**: požadovaná data nebyla nalezena v dané úrovni paměti a musela být hledána v nižších úrovních. Procesor musel čekat dokud nebyla data načtena z nižších úrovní.
- **Miss rate** =  $1 - \text{hit rate}$
- **Miss penalty** = čas na získání dat z nižší úrovně paměti + čas na přenesení dat do vyšší úrovně paměti + čas na doručení dat do procesoru
- **Hit time**  $\ll$  **Miss penalty**

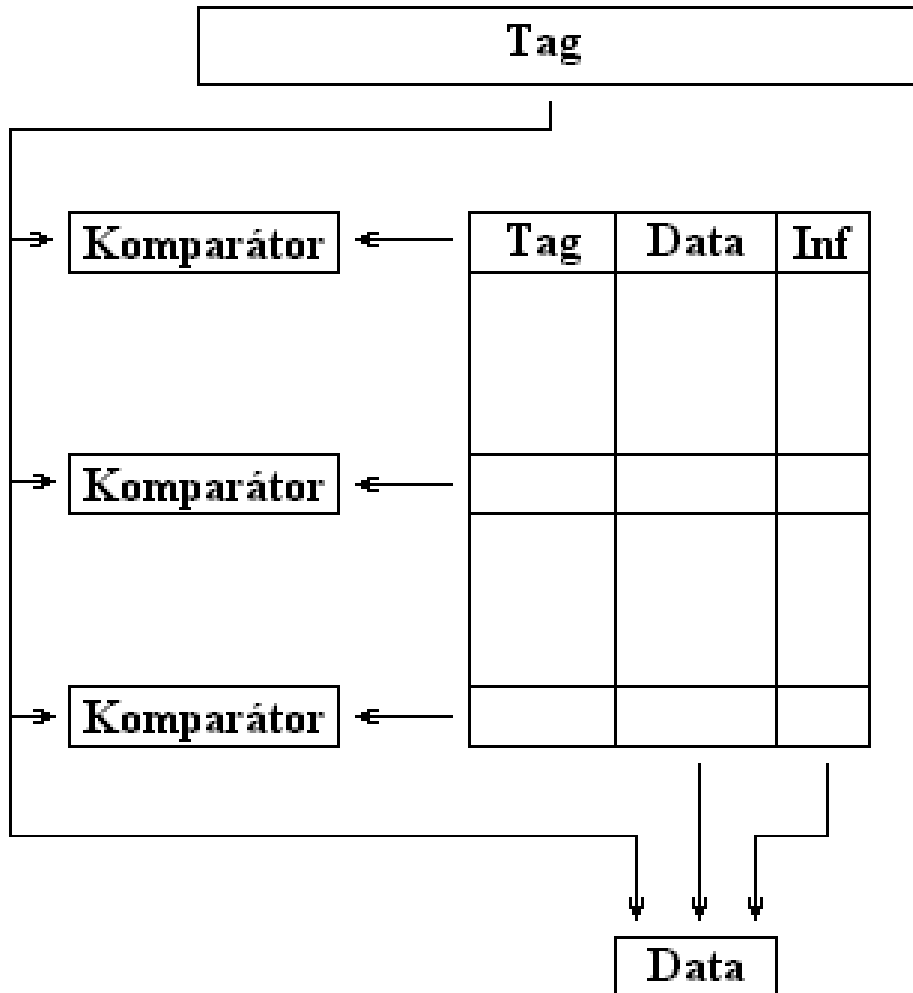
# Princip cache

- Organizovány jako tzv. asociativní paměti
- Asociativní paměti jsou tvořeny tabulkou (tabulkami), která obsahuje vždy sloupec, v němž jsou umístěny tzv. **tagy** (klíče), podle kterých se v asociativní paměti vyhledává.
- Dále jsou v tabulce umístěna **data**, která paměť uchovává, a popř. další informace nutné k zajištění správné funkce paměti.

# Typy paměti cache

- plně asociativní – celá adresa je brána jako tag
- přímo mapovaná – adresa třídy přivedena na dekodér a porovnána s tagem
- n-cestně asociativní - adresa třídy přivedena na n dekodérů a porovnána s tagem

# Schéma funkce plně asociativní cache paměti



- adresuje se částí datové položky - **klíčem (tag)**
- na rozdíl od adresovatelné paměti např. SRAM neobsahuje adresový dekodér. ale adresář

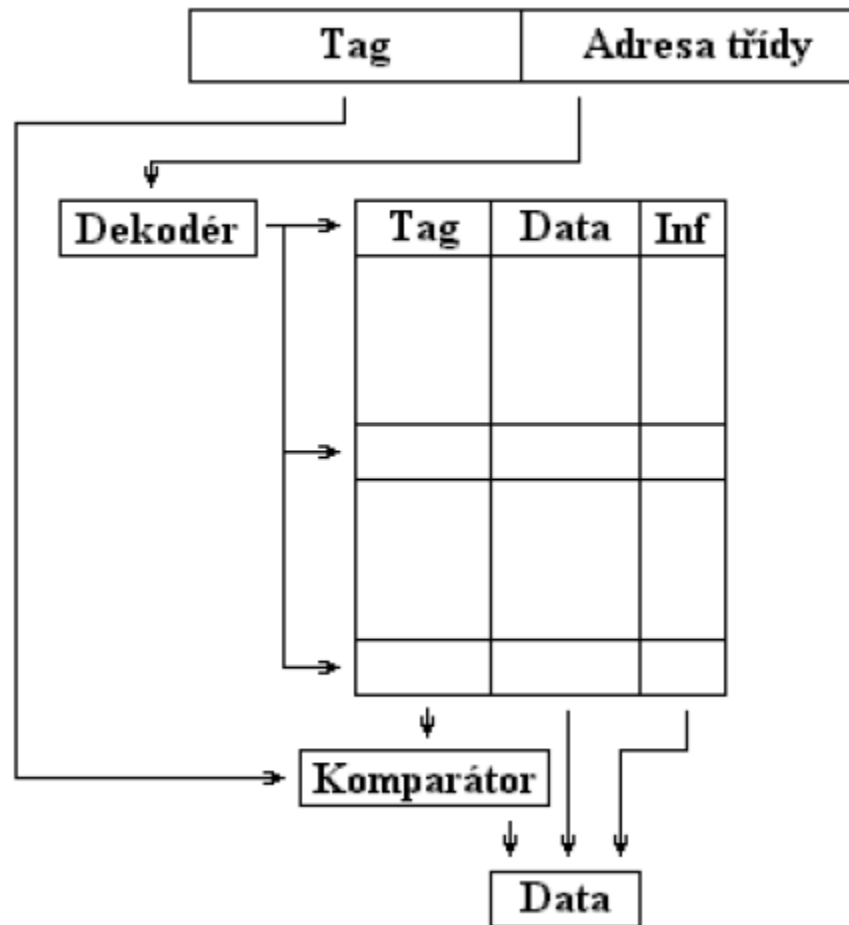
# Asociativní paměť s omezeným stupněm asociativity

- nevýhoda plně asociativní paměti:
  - adresář je tvořen speciálními obvody
  - při stejné kapacitě cca trojnásobná plocha čipu
- řešení ... omezený stupeň asociativity, tzn. každé položce je určeno místo (nebo několik míst podle stupně asociativity), kde se může nacházet. Toto místo je určené částí adresy položky.

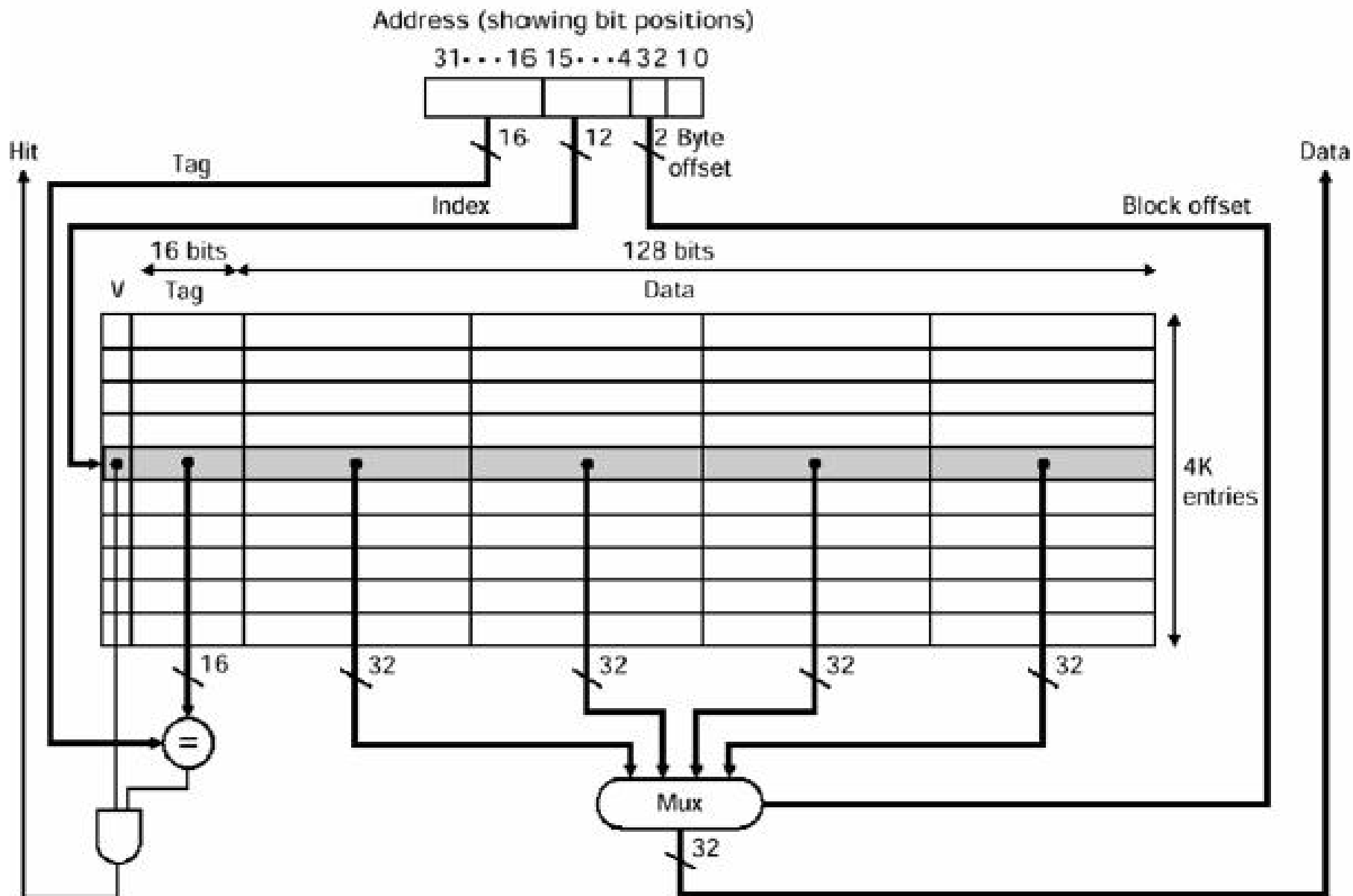
# ... asociativní paměť s omezeným stupněm asociativity

- adresář je možné realizovat běžnou pamětí RAM
- přítomnost položky se zjistí porovnání s klíčem (nebo několika klíči) uloženým v adresáři
- klíčem je část adresy
- stupeň asociativity je počet míst, na kterých se položka může nacházet
- pro zvýšení efektivity jsou data uložena po blocích (řádcích) např. 16 slabik

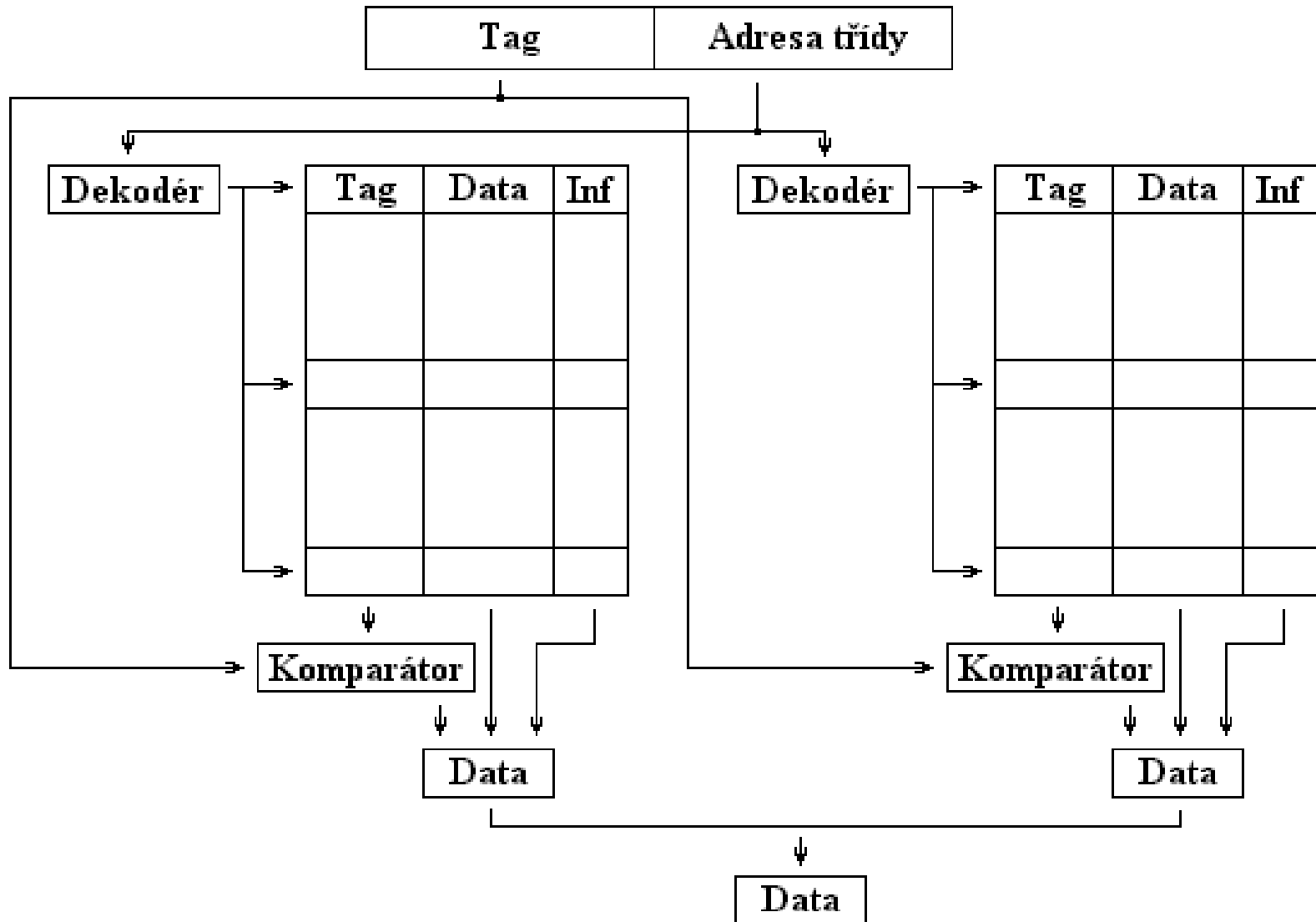
# přímo mapovaná cache



# schéma přímo mapované (1 cestné) paměti cache



# Schéma funkce n-cestně asociativní cache paměti (n=2)



# Příklad - cache v procesoru I80487

- 32 bitová fyzická adresa
- 8 KB cache, stupeň
- stupeň asociativity 4
- bloky dat 16 B (slabik)
  
- uvolňování položek - různé strategie, např. LRU, náhodný výběr z počtu položek v řádku (stupeň asociativity 4 ... tedy ze 4)

# Schéma funkce interní cache paměti procesoru 80486 (4-cestně asociativní)

