# Rychlá vyrovnávací paměť (RVP) - paměť cache

- Procesory synchronizované vysokým kmitočtem potřebují spolupracovat s dostatečně rychlými pamětmi s krátkou vybavovací dobou.
- DRAM v době I80386 měly vybavovací dobu 60 - 100 ns, doby cyklu byly delší.
- Řešení tohoto problému představují prvky SRAM s dobou cyklu 10 - 20 ns (platilo v době I80386).

## Princip úvah, na základě nichž bylo rozhodnuto o vybavení I80386 rychlou vyrovnávací pamětí

- V době, kdy se objevil procesor I80386, existovala jistá technologie podpůrných komponent, paměťové prvky a jejich parametry byly důležité.
- Paměti: široká škála různě rychlých (z hlediska vybavovací doby) paměťových prvků DRAM.
- Začátek existence procesoru I80386 + odhad doby, po niž bude vyráběn + odhad rychlosti, na niž se procesor dostane (MHz) => bylo nutno se zabývat úvahou, zda paměti budou rychlostně stačit.

# Problém vybavovací doby

- Doba cyklu - doba, za niž je možné generovat nový požadavek na paměťovou operaci.
- Sestává:

  z vybavovací doby,

  z doby potřebné na ustálení přechodových dějů na sběrnicích (data, adresa).
- Typické vybavovací doby [ns] čipů instalovaných v PC v době, kdy se na trhu objevil procesor I80386: 250, 150, 120, 85, 70, 65, 53 ns.
- PC 286/386 - vybavovací doba čipů pod 100 ns.
- Údaj o vybavovací době byl součástí typového označení čipu:
- 4164-25 64 kbitů, vybavovací doba 250 ns
- 41256-10 256 kbitů, vybavovací doba 100 ns

- 4164-15 64 kbitů, vybavovací doba 150 ns
- V době, kdy končila 286 se pracovalo s paměťovými čipy s vybavovací dobou 60 - 80 ns.
- Stavy čekání (Wait State):

  Byl požadavek, aby paměť byla schopna na požadavek od CPU reagovat během dvou taktů: 8 MHz - takt 125 ns

  20 MHz - 50 ns

  33 MHz - 30 ns
- Příklad:

  Uvažujeme o možnosti navrhnout PC s hodinami 10 MHz a paměťovými čipy 256 kB, 120 ns.

  Synchronizace 10 MHz => musíme udržet cyklus paměti pod 2 x 100 = 200ns.

  Doba pro ustálení přechodových dějů pro tyto čipy - 90 ns

  => doba cyklu = 120 + 90 = 210 ns -

paměti jsou o 10 ns pomalejší než potřebujeme.

- Řešení:

  zpomalit procesor (nesmysl),

  přidat stavy čekání (nesmysl),

  zrychlit paměť DRAM (může být drahé a nerealizovatelné – dáno úrovní technologie),

  doplnit architekturu o další paměťový prvek, který bude rychlejší než paměť DRAM.

- Stav čekání - jeden takt/více taktů synchronizačních pulsů přidaný navíc do komunikace s pamětí nebo řadičem periferního zařízení.

- Princip generování čekacích stavů: v systémové sběrnici je signál (většinou označený jako READY, jímž zařízení – v tomto případě paměť dá najevo, že je schopno dokončit požadovanou operaci v požadovaném čase (tzn. když nenastaví signál READY, znamená to, že není zařízení schopno dokončit operaci včas – v tomto případě do 2 synchronizačních pulsů, procesor čeká, dokud zařízení (paměť) nenastaví READY).
- Tuto možnost musí mít i řadiče periferních zařízení.

| Vybavovací doba [ns] | Doba cyklu [ns] |
|---|---|
| 200 | 370 |
| 150 | 270 |
| 120 | 210 |
| 100 | 175 |
| 80 | 145 |
| 60 | 105 |
| 53 | 95 |

| f [MHz] | takt [ns] | DC1 [ns], 0 čekacích stavů (2 takty) | DC2 [ns], 1 čekací stav (3 takty) | Vybav. doba [ns] | Vybav. doba [ns] |
|---|---|---|---|---|---|
| 8 | 125 | 250 | 375 | 120 | 200 |
| 10 | 100 | 200 | 300 | 100 | 150 |
| 20 | 50 | 100 | 150 | 53 | 80 |
| 25 | 40 | 80 | 120 | není | 60 |
| 33 | 30 | 60 | 90 | není | 53 |

- Princip úvahy:

  Kmitočet procesoru 10 MHz, požadujeme, aby komunikace s pamětí probíhala bez čekacích stavů, tzn. doba cyklu musí být kratší než 200 ns (vybavovací doba + doba na ustálení přechodových dějů)

  => musíme zvolit paměťové čipy s vybavovací dobou 100 ns .

  Tabulka napovídá, že pro PC386/25 MHz resp. 33 MHz byly některé situace neřešitelné klasickými postupy (využitím DRAM).

  => pro vyšší kmitočty neexistovala dostatečně rychlá DRAM taková, aby vyhovovala z hlediska počtu Wait stavů, které si můžeme dovolit.

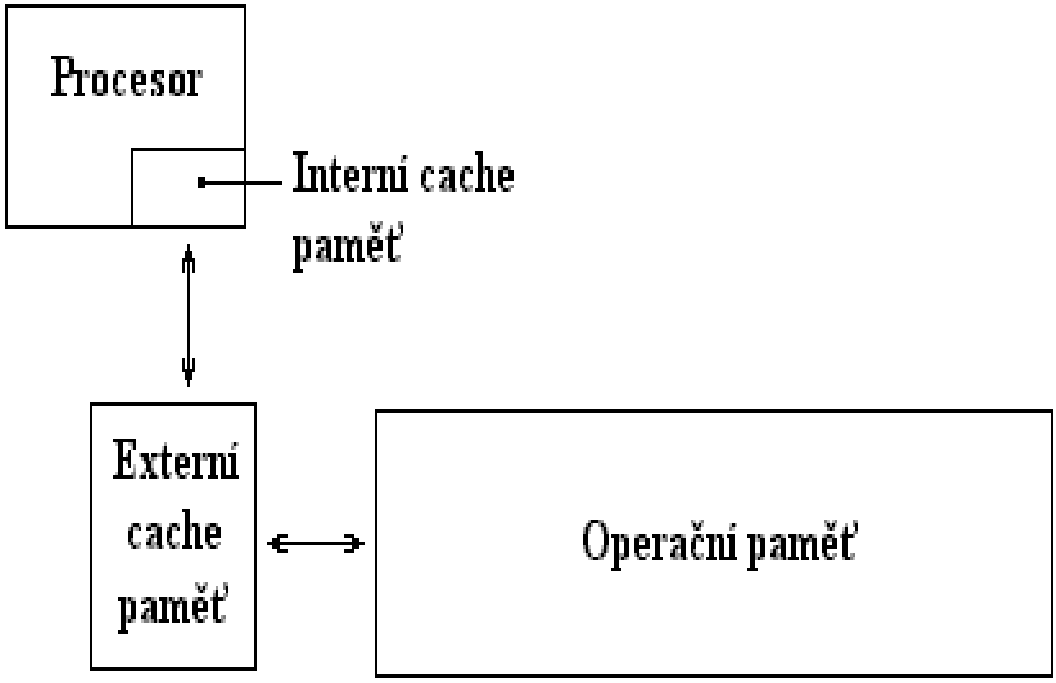
- Řešení - **paměť cache (rychlá vyrovnávací paměť) komunikující s procesorem na vyšší frekvenci**.

- Charakteristika:

  Rychlá vyrovnávací paměť I80386 měla výrazně menší kapacitu než paměť DRAM (64 kB, 128 kB, 256 kB).

  Byla realizována jako paměť SRAM (statická paměť RAM), byla rychlejší, což procesoru umožňovalo adresovat ji bez čekacích stavů. Její cena/bit byla vyšší (obecně: čím menší kapacita, tím vyšší cena/bit).
- PC 386 - vyrovnávací paměť byla řízena řadičem 80385.
- PC 486 - vyrovnávací paměť je integrována do procesoru spolu s řadičem vyr. paměti (on-chip cache).

# Základní principy konstrukce RVP v personálních počítačích




- Přístupy do paměti většinou probíhají tak, že jsou adresována paměťová místa, která následují bezprostředně za sebou $\Rightarrow$ má smysl uchovávat bloky instrukcí/dat v RVP.
- Paměť RVP - spojení výhod rychlé paměti SRAM s nízkou cenou paměti DRAM s cílem vytvořit výkonný paměťový systém.

- Paměť RVP - vlastní paměťové prvky a řadič paměti RVP.
- Všechny přenosy mezi pamětmi se realizují přes systémovou sběrnici, tzn. obrázku **nesmí být rozuměno** tak, že zobrazuje způsob propojení, spíše vyjadřují pořadí hledání adresovaných dat.
- Alternativy, kde může být paměť RVP spolu s řadičem RVP umístěna:
  1. Obě tyto komponenty jsou součástí procesoru (on-chip RVP).
  2. Obě tyto komponenty jsou samostatnými prvky.
  3. Kombinace těchto řešení - řadič paměti RVP byl součástí procesoru a vlastní RVP byla tvořena externí pamětí SRAM.

- **Závěr:**

  **Kombinací paměti RVP typu SRAM a operační paměti typu DRAM se dosáhne výrazného snížení vybavovací doby a tím zvýšení rychlosti.**

**Principy spolupráce mezi procesorem a DRAM/SRAM**

- Procesor žádá data z paměti - vloží na adresovou sběrnici adresu.
- Řadič paměti tuto adresu přijme a zahájí kroky k získání dat – zjistí se, zda jsou požadovaná data v RVP.
- Mohou nastat dvě situace: *cache hit* a *cache miss.*
- **Cache hit** - požadovaná data jsou v paměti RVP, řadič je přečte a předá je procesoru.

- To vše se odehraje bez čekacích stavů, tzn. maximální možnou rychlostí. Neproběhla žádná komunikace s operační pamětí.
- **Cache miss** - data nejsou v paměti RVP a musí být přečtena z operační paměti.
- Operační paměť DRAM je pomalejší než paměť RVP $\Rightarrow$ do komunikace je nutné vložit patřičný počet čekacích stavů.
- Pokud nastane *cache miss*, pak musí řadič paměti RVP provést tyto činnosti:

  Přečíst z operační paměti celý řádek - *cache line.* Tato operace je označována jako *cache line fill.*

  * Předtím je ale nutno nejprve uvolnit v paměti RVP dostatečný prostor – to je proces, jehož průběh musí být určen přesnými pravidly.
  * Pokud se musí z paměti RVP odstranit data, která byla

během předcházejících operací nějak modifikována, je nutno je nejprve přenést do operační paměti (předtím než se na toto místo v paměti RVP nahraje nový řádek).

* Data, která byla přečtena, se okamžitě přenášejí do procesoru,

  $\Rightarrow$ řadič paměti je prvkem, který je schopen autonomně provádět cykly sběrnice, během nichž se provádí zápis/čtení dat do/z operační paměti.

**Strategie, jimiž se řadič paměti RVP řídí**

- *write-through*
- *write-back*
- *write-allocate*

První dvě strategie ovlivňují *cache hit*, poslední ovlivňuje *cache miss*.

## Paměť RVP typu Write-through

- Tato strategie je implementována nejčastěji.
- Všechny operace zápisu do paměti RVP se provedou také do operační paměti.

  ⇒ operace zápisu do RVP vede vždy k zápisu do operační paměti, i když nastane *cache hit*.

  ⇒ taková strategie by mohla znamenat dlouhé vybavovací doby.
- Aby tomu tak nebylo, paměť RVP typu *write-through* využívá pro operaci zápis vyrovnávací paměť (buffer), operace zápisu do operační paměti se realizuje tak, aby nebyly zdržovány přenosy mezi procesorem a pamětí RVP.

## Paměť RVP typu Write-back

- Při zápisu do paměti RVP se aktualizuje pouze obsah paměti RVP, obsah operační paměti se nemění.

- Pokud je nutné z paměti RVP odstranit nějaké řádky, přenesou se do operační paměti pouze ty řádky, které byly předtím nějak změněny.
- Nevýhoda: výměna dat ve srovnání s metodou *Write-through* trvá déle, protože před zápisem nových řádků se musí napřed modifikované řádky uložit do operační paměti.

**Paměť RVP typu Write-allocate**

- Tato strategie se používá pouze velmi zřídka.

**Situace, které mohou nastat, když do operační paměti může zapisovat další prvek (např. řadič DMA)**

- *Cache invalidation* - situace, když se data v operační paměti změní (při zápisu dat do operační paměti z periferního zařízení přes řadič DMA), takže data uložená v paměti RVP se stanou neplatná.

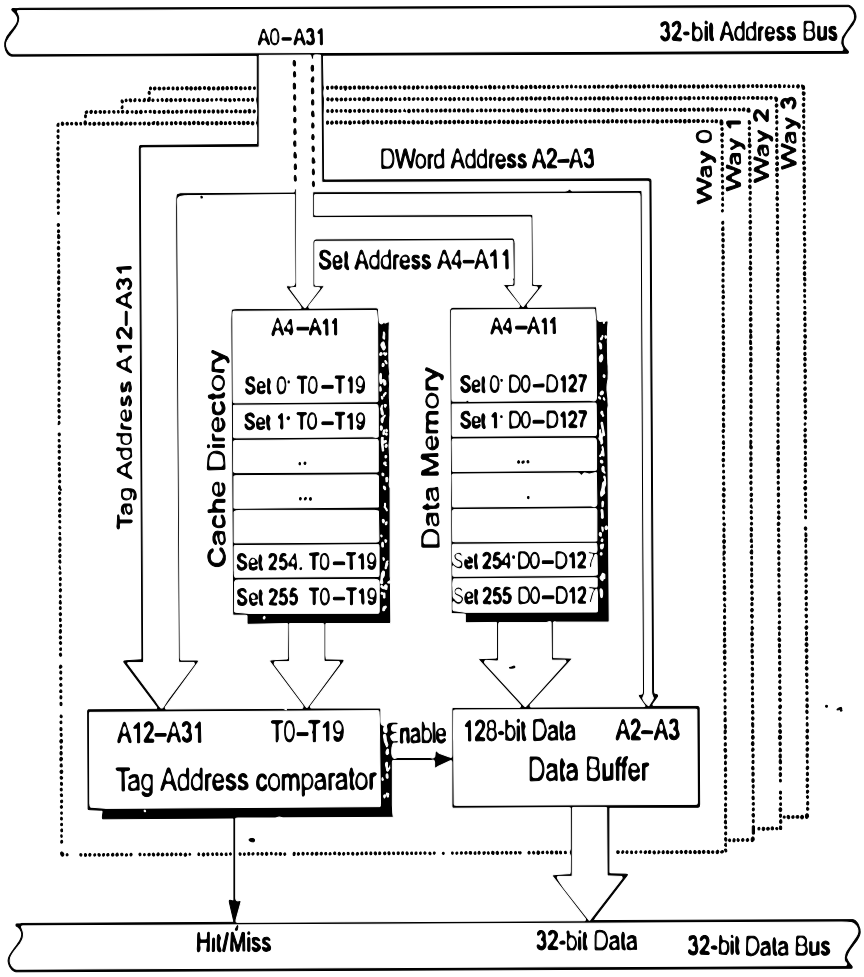
- *Cache flush* - situace, kdy má řadič DMA přenášet data z operační paměti do periferního zařízení, platná data jsou ale uložená v RVP.

## Organizace paměti RVP

- Data a programy vytvářejí bloky $\Rightarrow$ je pravděpodobné, že v dalším přístupu do paměti budou vyžadována data, která jsou součástí stejného řádku - zvyšuje se tak podíl *hit cache*.
- Řádky paměti RVP (cache lines) mají obvykle velikost 16 nebo 32 byte.
- Mnohé řadiče RVP jsou schopny pracovat v tzv. *nárazovém režimu (burst mode)* - čtou se celé bloky dat.

# Organizace RVP v I386

- Řadič RVP rozděluje 32 bitovou adresu na 20 bitovou adresu tagu A12 - A31, 8 bitovou adresu sady (set) A4 - A11 a 4 bitovou adresu bytu A0 - A3.




Organizace RVP – I80386

- Adresa paměti RVP sestává z těchto částí:
  **adresy položky v adresáři** (cache directory entry) – obsah položky v adresáři reflektuje, zda jsou aktuálně adresovaná data k dispozici v RVP
  **adresy položky v paměti RVP** (cache memory entry)
- **Adresa položky v adresáři** (*Tag*):
  * Je to informace o tom, která data jsou v RVP uložená.
  * Tato informace může být uložená buď v řadiči paměti nebo v externí paměti.
  * V takovém případě je většinou pro paměť RVP potřeba více čipů než odpovídá její kapacitě.
  * Důvod je v tom, že část paměťových prvků musí uchovávat položky adresáře, v ostatních jsou uložena vlastní data.

* V našem případě, pokud v T0 – T19 jsou ve všech set uloženy stejná čísla, pak v D0 – D127 (16 slabik) jsou ve všech set (4 kB) uložena data, která tvoří celek.
* Na obsah T0 – T19 se můžeme dívat jako na adresu bloku (stránky) o velikosti 4 kB, vlastní data jsou uložena v Data Memory.

- **Adresa položky v paměti** - na této adrese jsou uložena vlastní data.

**Tag**

- Pomocí této položky řadič paměti zjišťuje, zda nastal *cache hit* nebo *cache miss*.
- V každé položce Tag je uloženo 20 bitů.

- Tag je platný pouze tehdy, když je nastaven *valid bit* (bit platnost). Pokud tomu tak není, pak jsou data uložená v patřičném řádku neplatná.
- Je-li nastaven bit *write protection,* pak řádek s takto nastavenou ochranou nemůže být přepsán.

**Set**

- Set je tvořen tagem uloženým v paměti tagů a odpovídajícím řádkem RVP (cache line) uloženým v paměti dat.
- Adresa setu sestává z 8 bitů A4 - A11 (256 možností).

**Way**

- Paměť RVP sestává ze 4 ways. Pojem way vlastně představuje asociativnost paměti (čte se z té adresy, jejíž obsah v paměti tagů souhlasí s adresou zadanou z procesoru).

- Princip :

  Tagy uložené v paměti jsou srovnávány s tagy generovanými procesorem, výsledkem je *cache hit* nebo *cache miss*.

  $\Rightarrow$ skupina dat odpovídající jednomu řádku může být RVP uložena na jedné ze 4 různých posic (4 ways).

- Každá way obsahuje 256 sets, každá set obsahuje tag uložený v paměti tagů a řádek velikosti 16 slabik v paměti dat.

- Výpočet kapacity RVP:

  kapacita = počet way x počet sets x velikost řádku

  Pro i386 tak dostaneme 16 kbyte kapacity RVP (way - 4, set - 256, řádek - 16 byte) .

## Činnost řadiče RVP, když nastane *cache hit*

- Procesor vloží na adresovou sběrnici 32 bitovou adresu paměťového místa, z něhož mají být data čtena na adresovou sběrnici.
- Řadič paměti tuto kombinaci rozdělí na tyto části: tag, adresu setu a bytu.
- 20 bitová adresa tagu dělí 4 Gbytový adresový prostor procesoru i386 na $2^{20}$ stránek paměti RVP, každá kapacity 4 kbyte:
  - Každá stránka obsahuje 256 set, každá set představuje jeden řádek (cache line) velikosti 16 byte.
  - Paměť RVP sestává ze 4 way, v každé z nich se zjišťuje, zda adresa tagu souhlasí.
- Řadič RVP využívá pro adresu setu bity A4 - A11. Tyto bity určují jednu z 256 set v každé ze 4 way 0 - 3.

- Na této adrese přečte tag, ten se pak pošle do komparátoru tagu. Na jeho druhý vstup se přivádí tag z adresové sběrnice.
- Pokud nastane *cache hit*, je řádek přečtený z RVP řádkem hledaným.
- Z vyrovnávací paměti (buffer) se vybere podle hodnoty A2 - A3 double word (32 bitů).

**Komparátor adresy tagu**

- Komparátor srovnává adresu tagu z procesoru s adresou tagu přečtenou z RVP.
- Jestliže jsou tyto dvě adresy shodné, je aktivována vyrovnávací paměť (buffer) a data jsou k dispozici.
- Jestliže jsou tyto dvě adresy rozdílné, je výstup z bufferu zablokován, došlo ke *cache miss* $\Rightarrow$ řadič RVP pak musí:

  adresovat operační paměť a z ní přenést data do procesoru

zahájit plnění řádku těmito daty v RVP.

- Všechny tyto operace, tzn.výběr a srovnání tagu se provádí stejným způsobem ve všech 4 way ⇒ jsou zapotřebí 4 komparátory, čtyři vyrovnávací paměti ⇒ jestliže RVP sestává z více way, počet těchto prvků se zvyšuje.
- Paměť RVP typu *direct mapped cache* (přímo mapovaná - paměť RVP s jednou way)) je nejjednodušší z hlediska konstrukce, ale má nejnižší počet úspěšných hledání (*hit rate*).
- Častější sestava, paměť RVP se 2 way je jistým kompromisem. Ve srovnání s pamětí RVP se 4 way má výrazně nižší úspěšnost hledání, ale logika je výrazně jednodušší.
- RVP cache s 8 way je nejsložitější, ale úspěšnost vyhledávání je vysoká.
- Počet set nemusí být omezen na 256.

Příklad:

Typická RVP L2 kapacity 512 kbyte:

organizovaná jako asociativní RVP se 2 way,

s velikostí řádku 64 byte,

bude mít celkem 8192 set (512k/64)

s rozdělením adresy takto:

adresa bytu A0-A5 ($2^6$ = 64), adresa setu A6-A18 (8192 sets), tag A19-A31.

- RVP a velkou kapacitou - pro uložení tagu a dat se používají externí SRAM.
- Tagy musí být k dispozici dříve než data, poněvadž přečtený tag musí být ještě srovnáván v komparátoru s tagem dodaným z procesoru

  $\Rightarrow$ na systémové desce je nožné nalézt jeden nebo dva čipy SRAM s velmi krátkou vybavovací dobou (asi 15 ns), ostatní čipy mají vybavovací dobu delší (20 ns).

Čipy SRAM s krátkou vybavovací dobou - jsou v nich uloženy tagy,

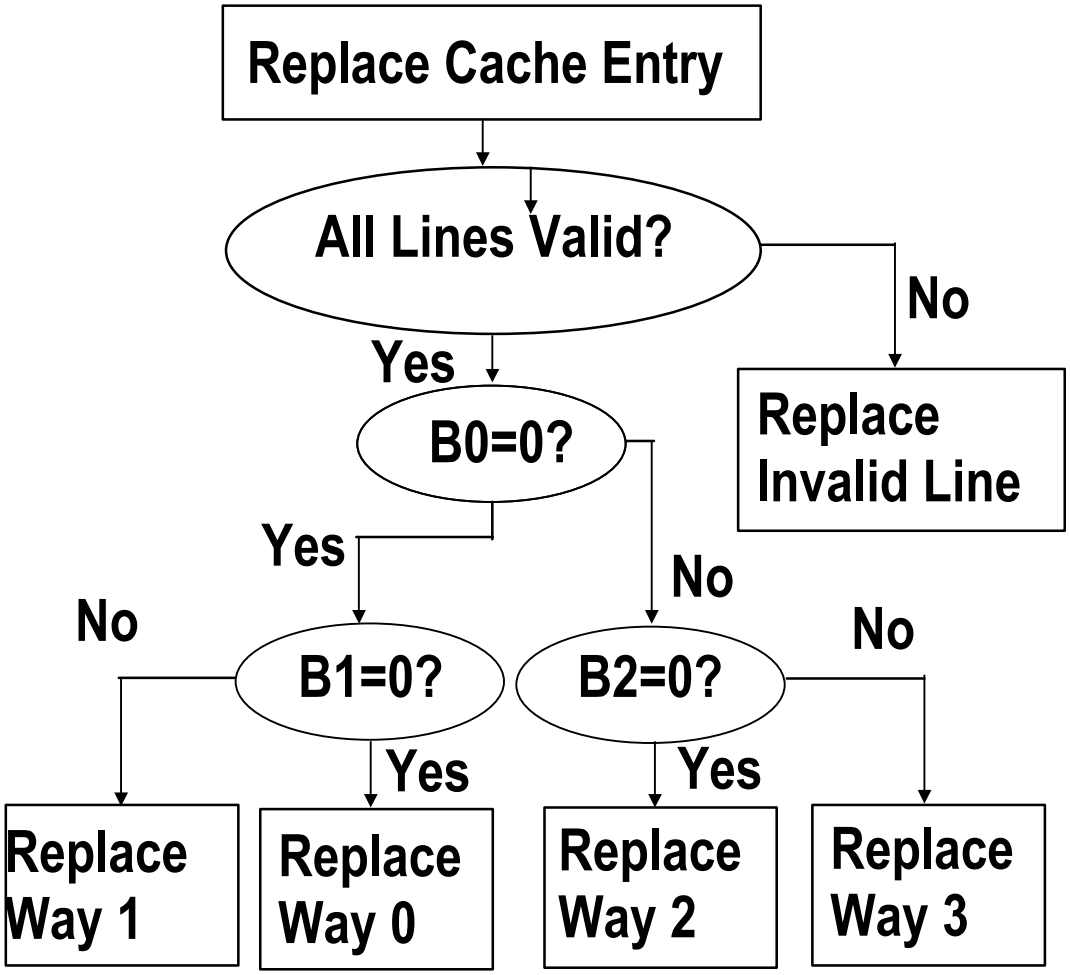
čipy s delší vybavovací dobou - data.

**Strategie výměny informace**

- Bit LRU - Last Recently Used
- Strategie LRU:

  LRU bit B0 = 1, poslední přístup byl do way 0 nebo way 1:
  B1 = 1, poslední přístupbyl do way 0
  B1 = 0, poslední přístup byl to way 1

  LRU bit B0 = 0, poslední přístup byl do way 2 nebo way 3:
  B2 = 1, poslední přístup byl do way 2
  B2 = 0, poslední přístup byl do way 3




## Paměti on-chip (first level) a paměti Second level

- i486 byl vybaven *on-chip RVP pamětí (first level cache, L1 cache)* kapacity 8 kB.
- Systémové desky nabízejí také možnost nainstalovat externí RVP *(second-level cache*, or *L2-cache).*

- Na první pohled se instalace RVP typu L1 kapacity 8 Kb může stát nevýznamná.
- Přesto tato malá RVP L1 malé kapacity dovolila významným způsobem zvýšit výkon např. u procesorů DX2, kdy je synchronizována stejným kmitočtem jako procesor, tudíž 2x vyšším než systémová deska a RVP typu L2.
- Velikost RVP paměti L2 bývá mezi 128 - 512 kB.
- Mechanismus vyhledávání dat, když je nainstalována RVP L1 i L2:
  1. Jestliže procesor zjistí, že hledaná data nejsou v interní RVP paměti L1,
  2. musí řadič paměti zjistit, zda tato data nejsou v externí RVP typu L2,
  3. v případě, že nastane *cache miss*, přepne se požadavek na čtení na operační paměť.
- Rozdíl mezi RVP typu L1, resp. L2:

  V případě RVP L1 čte CPU data z

RVP v cyklech sestávajících pouze z 1 synchronizačního pulsu, zatímco

při přístupu do paměti L2 je nutné provést sběrnicový cyklus, sestávajícího ze 2 synchronizačních pulsů,

$\Rightarrow$ přístup do RVP L1 je výrazně rychlejší (platí to pochopitelně za předpokladu, že data jsou sekvenčně uložena v paměti).

- K výraznému zhoršení dojde, pokud jsou data a kód uloženy natolik nahodile, že nemohou být současně uchována v RVP L2.
- K přepisu jednoho řádku do RVP se využívá nárazový (burst) režim.

## Procesor I486 a jeho on-chip RVP

- RVP L1 procesoru i486 byla organizována jako asociativní paměť sestávající ze 4 way.
- Každý way obsahovala 128 set (nebo řádků) velikosti 16 byte, celková

kapacita je 8 kB.

## Upozornění na pojmy

- Z hlediska architektury je nutné rozlišovat tyto typy vyrovnávacích pamětí:
  1. RVP pro komunikaci s hlavní pamětí (anglicky cache)
  2. vyrovnávací paměť pro komunikaci s floppy diskem a pevným diskem, nebo libovolným periferním zařízením (anglicky buffer).