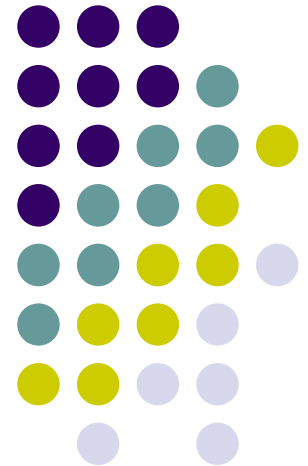
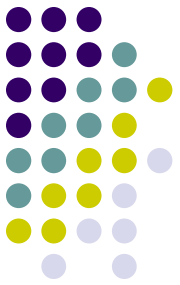


Cache paměti (3. část)

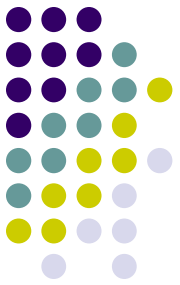
Hardware





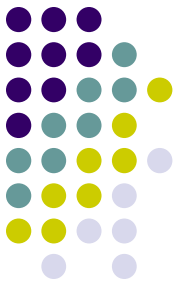
Přímo mapovaná cache

- V **plně asociativní cache** mohl být datový blok uložený na **jakémkoliv řádku**
- Z toho ale plyne zásadní nevýhoda – data je třeba hledat podle klíče na všech řádcích současně a k tomu je třeba **velké množství komparátorů**
- Základní myšlenka **přímo mapované cache** spočívá v tom, že informace **nemůže** být uložena v cache **kdekoliv**
- Místo, kde může v cache ležet blok dat z určité adresy je jednoznačně dáno
- Adresy z operační paměti jsou do cache **mapovány** na určené řádky
- Jednotlivé **řádky cache paměti se očísloví** a adresy v **operační paměti se rozdělí do tříd**
- Na nultém řádku v cache pak mohou ležet pouze data z adres, které patří do třídy 0
- Na prvním řádku v cache mohou ležet pouze data z adres, které patří do třídy 1
- Na druhém řádku v cache mohou ležet pouze data z adres, které patří do třídy 2
- Atd.....
- Požadovaná data pak v cache hledáme **na jednom konkrétním místě** (konkrétním řádku)
- Víme-li, do jaké třídy adresa patří, je jasné, na kolikátém řádku jí najdeme



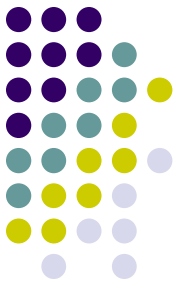
Přímo mapovaná cache

- Pokud má cache **8 řádků**, rozdělíme adresy do **8 tříd**
 - Pokud má cache **16 řádků**, rozdělíme adresy do **16 tříd**
 - Pokud má cache **1024 řádků**, rozdělíme adresy do **1024 tříd**
 - Atd....
-
- K **určení třídy**, do které daná adresa patří, se pouze **několik posledních bitů adresy**, které by jinak byly klíčem záznamu
-
- Pokud se adresy dělí do **8 tříd** (cache má 8 řádků) pak se k určení třídy použijí poslední **tři bity**
 - Pokud se adresy dělí do **512 tříd** (cache má 512 řádků) pak se k určení třídy použijí posledních **9 bitů**
 - Pokud se adresy dělí do **4096 tříd** (cache má 4096 řádků) pak se k určení třídy použijí posledních **12 bitů**
 - Atd....
-
- **třída** jsou tedy **poslední bity adresy** a říkají, na **kolikátém řádku** by mohla informace v cache ležet



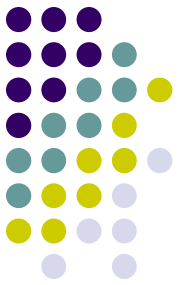
Přímo mapovaná cache

- **Příklad**
- Přímo mapovaná cache má kapacitu **1 kB**. Data jsou v cache ukládána po jednom bajtu (tedy nejsou ukládána po blocích) a cache má 1024 řádků
- Na kolikátém řádku v cache bude uložen bajt z adresy 12345678h?
- 12345678h = 00010010001101000101011001111000 b
- Cache má 1024 řádků – adresy se rozdělí do 1024 tříd
- K určení čísla třídy potřebujeme 10 bitů ($2^{10} = 1024$)
- Třidu určíme pomocí posledních 10 bitů adresy
- 0001001000110100010101 **1001111000**
- Třidu tedy určují bity **1001111000** – to dává číselně **632**
- Adresa 12345678h patří do třídy **632**
- Data z této adresy mohou být v cache uložena na **632. řádku**
- Jako **klíč** bude u tohoto záznamu uvedeno 22 bitů – 0001001000110100010101
- Není třeba uvádět jako klíč celou 32-bitovou adresu
- Zbývajících 10 bitů adresy vyplývá z toho, že data leží na 632. řádku
- Pokud chce mikroprocesor pracovat s bajtem z adresy 12345678h, stačí hledat ho na 632. řádku a není nutné hledat na všech řádcích v cache – stačit bude **jeden jediný komparátor**, který porovná horních 22 bitů adresy s klíčem



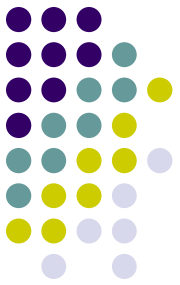
Přímo mapovaná cache

- **Příklad**
- Přímo mapovaná cache má kapacitu **256 B**. Data jsou v cache ukládána po jednom bajtu (tedy nejsou ukládána po blocích) a cache má 256 řádků
- Na kolikátém řádku v cache bude uložen bajt z adresy 12345678h?
- 12345678h = 00010010001101000101011001111000 b
- Cache má 256 řádků – adresy se rozdělí do 256 tříd
- K určení čísla třídy potřebujeme 8 bitů ($2^8 = 256$)
- Třidu určíme pomocí posledních 8 bitů adresy
- 000100100011010001010110 **01111000**
- Třidu tedy určují bity **01111000** – to dává číselně **120**
- Adresa 12345678h patří do třídy **120**
- Data z této adresy mohou být v cache uložena na **120. řádku**
- Jako **klíč** bude u tohoto záznamu uvedeno **24 bitů** – 000100100011010001010110



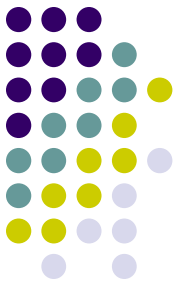
Přímo mapovaná cache

- **Příklad**
- Cache je stejná jako v minulém příkladu, ale změnila se adresa
- Přímo mapovaná cache má kapacitu **256 B**. Data jsou v cache ukládána po jednom bajtu (tedy nejsou ukládána po blocích) a cache má 256 řádků
- Na kolikátém řádku v cache bude uložen bajt z adresy **ABCDEF78h**?
- $ABCDEF78h = 10101011110011011110111101111000$ b
- Cache má 256 řádků – adresy se rozdělí do 256 tříd
- K určení čísla třídy potřebujeme 8 bitů ($2^8 = 256$)
- Třidu určíme pomocí posledních 8 bitů adresy
- 101010111100110111101111 **01111000**
- Třidu tedy určují bity **01111000** – to dává číselně **120**
- Adresa **ABCDEF78h** patří do také do třídy **120** – patří do stejné třídy jako adresa **12345678h**, se kterou jsme pracovali v minulém příkladu – obě adresy mají na konci stejných 8 bitů
- Data z této adresy mohou být v cache uložena na **120. řádku**
- **Adresy **ABCDEF78h** a **12345678h** patří do stejné třídy** a mohou uloženy pouze na 120. řádku
- V cache **nemohou být uložena data z obou těchto adres zároveň** – 120. řádek bude obsazen daty z jedné nebo druhé adresy
- Jako **klíč** bude u tohoto záznamu uvedeno **24 bitů** – 101010111100110111101111



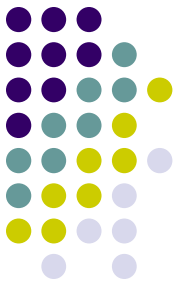
Přímo mapovaná cache

- **Příklad**
- Přímo mapovaná cache má kapacitu **8 kB**. Data jsou v cache ukládána po jednom bajtu (tedy nejsou ukládána po blocích) a cache má 8192 řádků
- Na kolikátém řádku v cache bude uložen bajt z adresy 12345678h?
- 12345678h = 00010010001101000101011001111000 b
- Cache má 8192 řádků – adresy se rozdělí do 8192 tříd
- K určení čísla třídy potřebujeme 13 bitů ($2^{13} = 8192$)
- Třidu určíme pomocí posledních 13 bitů adresy
- *0001001000110100010* **1011001111000**
- Třidu tedy určují bity **1011001111000** – to dává číselně **5752**
- Adresa 12345678h patří do třídy **5752**
- Data z této adresy mohou být v cache uložena na **5752. řádku**
- Jako **klíč** bude u tohoto záznamu uvedeno tentokrát jen **19 bitů** – *0001001000110100010*



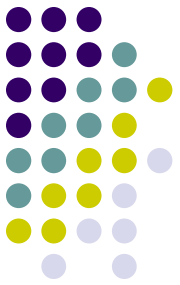
Přímo mapovaná cache

- **Příklad**
- Přímo mapovaná cache má kapacitu **8 kB = 8192 B**
- Data jsou v cache ukládána po **blocích o velikosti 8 B**
- Cache má tedy **1024 řádků**, na každém řádku leží **blok 8 B dat**
- Naolikátém řádku v cache bude uložen bajt z adresy 12345678h?
- 12345678h = 00010010001101000101011001111000 b
- Cache má 1024 řádků – adresy se rozdělí do 1024 tříd
- K určení čísla třídy potřebujeme 10 bitů ($2^{10} = 1024$)
- Třidu určíme pomocí posledních 10 bitů adresy, které by jinak sloužily jako klíč
- **Poslední tři bity adresy se nijak nevyužijí**, protože se do cache data ukládají po blocích velkých 8 bajtů
- Každý takový blok začíná adresou, která je **dělitelná číslem 8**, tedy víme, že adresa, kterou blok začíná **musí mít na konci tři nulové bity**
- 0001001000110100010 1011001111 000
- Třidu tedy určují bity **1011001111** – to dává číselně **719**
- Adresa 12345678h patří do třídy **719**
- Data z této adresy mohou být v cache uložena na **719. řádku**
- Jako **klíč** bude u tohoto záznamu uvedeno **19 bitů** – 0001001000110100010
- *Cache je velmi efektivní – na každém řádku je 8 B dat (64 bitů) a k nim jen 19b klíč. V celé cache je jeden jediný 19-bitový komparátor*



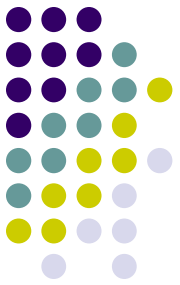
Přímo mapovaná cache

- **Příklad**
 - Přímo mapovaná cache má kapacitu **1 kB = 1024 B**
 - Data jsou v cache ukládána po **blocích o velikosti 16 B**
 - Cache má tedy **64 řádků**, na každém řádku leží **blok 16 B dat**
 - Na kolikátém řádku v cache bude uložen bajt z adresy 12345678h?
-
- 12345678h = 00010010001101000101011001111000 b
 - Cache má jen 64 řádků – adresy se rozdělí do 64 tříd
 - K určení čísla třídy potřebujeme 6 bitů ($2^6 = 64$)
 - Třidu určíme pomocí posledních 6 bitů adresy, které by jinak sloužily jako klíč
 - **Poslední 4 bity adresy se nijak nevyužijí**, protože se pracuje s **blokem 16 B**
 - Každý takový blok začíná adresou, která je **dělitelná číslem 16**, tedy víme, že adresa, kterou blok začíná **musí mít na konci 4 nulové bity**
 - 0001001000110100010101 **100111** 0000 – blok bude začínat adresou 12345670h
 - Třidu tedy určují bity **100111** – to dává číselně **39**
-
- Adresa 12345678h patří do třídy **39**
 - Data z této adresy mohou být v cache uložena na **39. řádku**
 - Jako **klíč** bude u tohoto záznamu uvedeno **22 bitů** – 0001001000110100010101
 - *Cache je velmi efektivní – na každém řádku je 16 B dat (128 bitů) a k nim jen 22b klíč. V celé cache je jeden jediný 22-bitový komparátor*



Přímo mapovaná cache

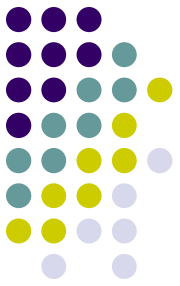
- **Příklad**
 - Přímo mapovaná cache má kapacitu **1 MB = 1048576 B**
 - Data jsou v cache ukládána po **blocích o velikosti 32 B**
 - Cache má tedy **32768 řádků**, na každém řádku leží **blok 32 B dat**
 - Na kolikátém řádku v cache bude uložen bajt z adresy 12345678h?
-
- 12345678h = 00010010001101000101011001111000 b
 - Cache má 32768 řádků – adresy se rozdělí do 32768 tříd
 - K určení čísla třídy potřebujeme 15 bitů ($2^{15} = 32768$)
 - Třidu určíme pomocí posledních 15 bitů adresy, které by jinak sloužily jako klíč
 - **Posledních 5 bitů adresy se nijak nevyužijí**, protože se pracuje s **blokem 32 B**
 - Každý takový blok začíná adresou, která je **dělitelná číslem 32**, tedy víme, že adresa, kterou blok začíná **musí mít na konci 5 nulových bitů**
 - 000100100011 **010001010110011** 00000 – blok bude začínat adresou 12345660h
 - Třidu tedy určují bity **010001010110011** – to dává číselně **8883**
-
- Adresa 12345678h patří do třídy **8883**
 - Data z této adresy mohou být v cache uložena na **8883. řádku**
 - Jako **klíč** bude u tohoto záznamu uvedeno **12 bitů** – 000100100011
 - *Cache je velmi efektivní – na každém řádku je 32 B dat (256 bitů) a k nim jen 12b klíč. V celé cache je jeden jediný 12-bitový komparátor*



Přímo mapovaná cache

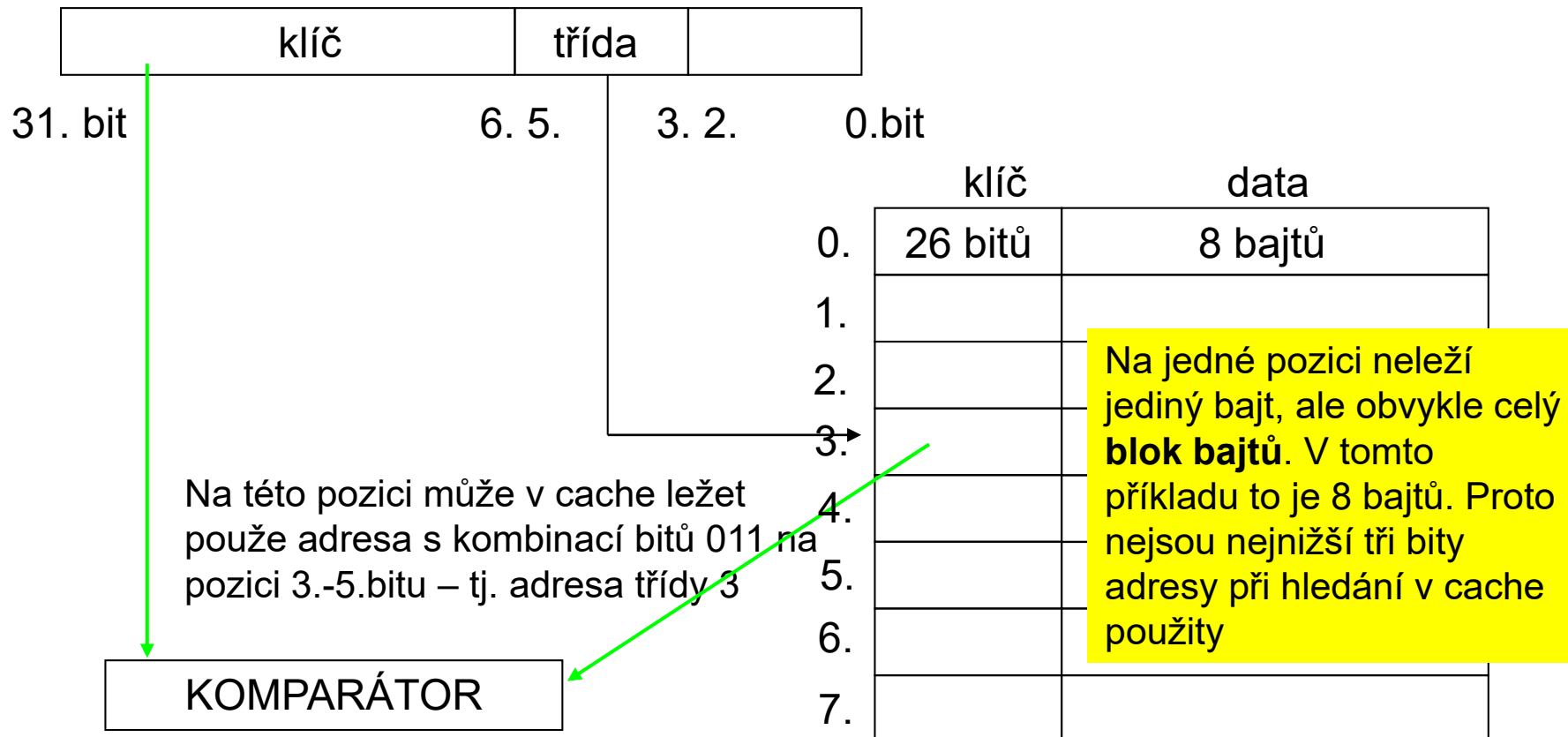
- Požadovaná data tedy v cache hledáme na konkrétním místě (konkrétním řádku)
- Protože obvykle ukládáme v cache data po blocích, několik posledních bitů adresy se nebude využívat
- Hledaná adresa bez posledních nevyužívaných bitů se rozdělí na **klíč** a **adresu třídy**
- Adresa **třídy** je přivedena na vstup dekodéru, který podle ní vybere jeden řádek v tabulce (na kterém řádku v cache je tato adresa mapována)
- Klíč na tomto řádku je následně porovnán s hledaným klíčem, čímž se rozhodne o přítomnosti resp. nepřítomnosti informace v cache paměti
- Adresa **třídy** jsou tedy bity adresy, které říkají, na jakém řádku by mohla informace v cache ležet
- **Klíčem** jsou zbývající horní bity adresy, které je potřeba otestovat na shodu

Příklad přímo mapované paměti cache s počtem tříd = 8

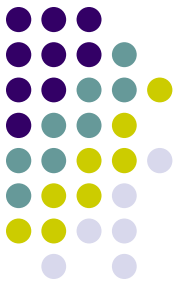


Fyzická adresa

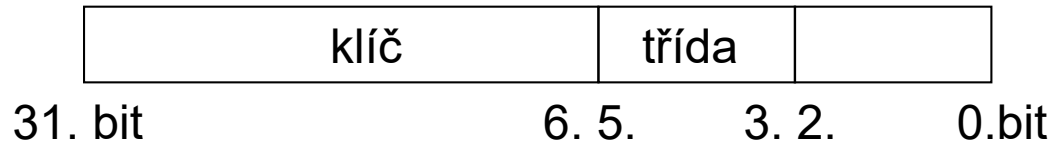
3.-5.bit adresy jí přiřazují do jedné z osmi tříd (na jeden z 8 řádků v cache)



Příklad přímo mapované paměti cache s počtem tříd = 8



Fyzická adresa



Jak poznat, zda je v paměti cache obraz adresy ABCD1234h ?

1. ABCD1234h =

101010111100110100010010001110100b

2. Jde o adresu třídy **6**

3. Je-li obsah této adresy cachován, může ležet pouze na **6. řádku** v cache

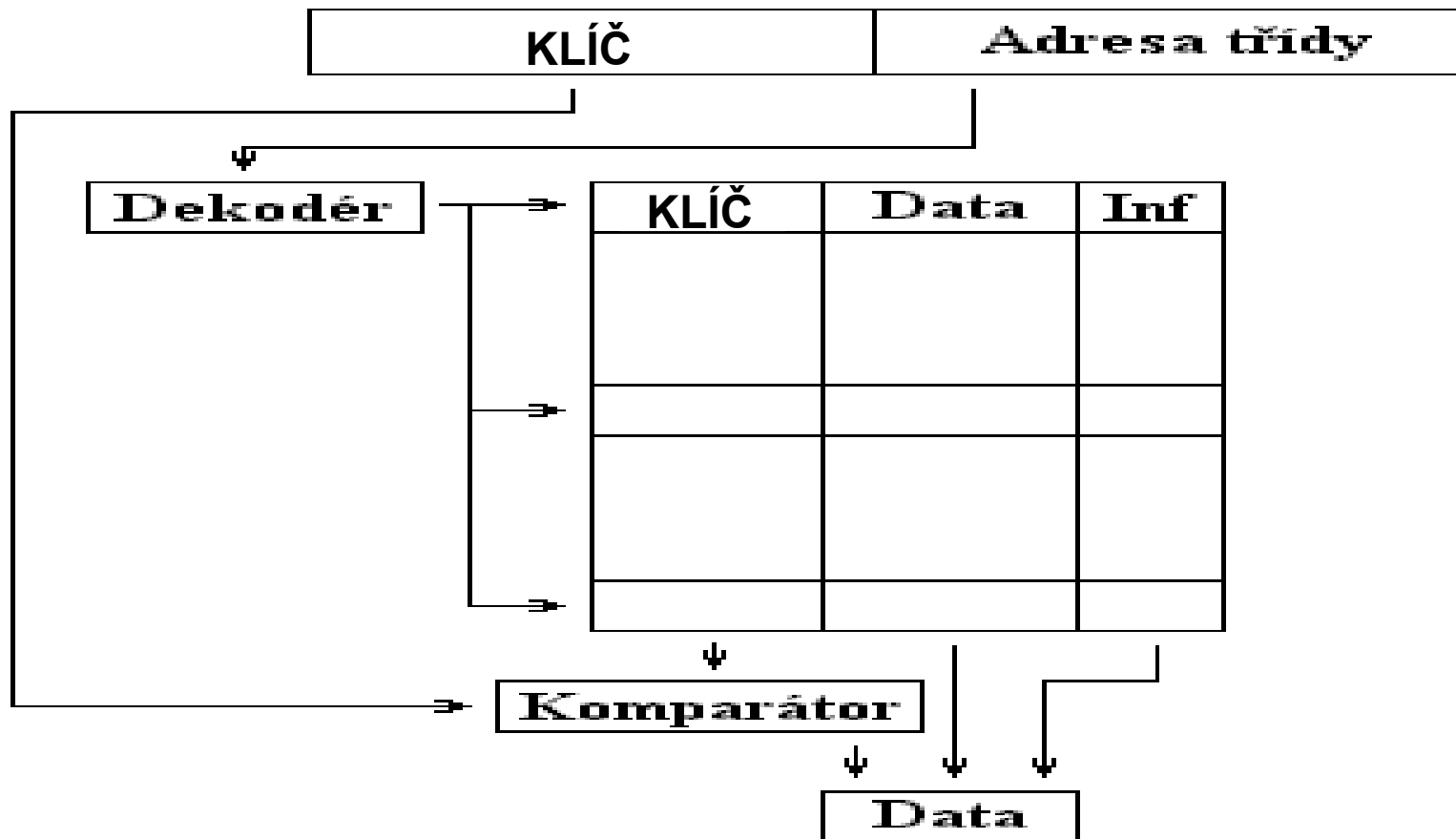
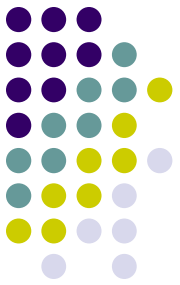
4. klíč na 6. pozici musí být roven

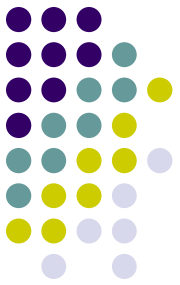
10101011110011010001001000 (tj. 26 nejvyšších bitů adresy)

	klíč	data
0.	26 bitů	8 bajtů
1.		
2.		
3.		
4.		
5.		
6.		
7.		

5. Došlo-li ke shodě klíčů, je v paměti cache uložen obsah adres ABCD1230h až ABCD1237h (cachován je blok 8 B)

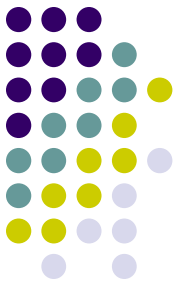
Přímo mapovaná cache





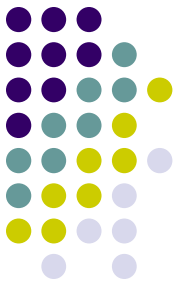
Přímo mapovaná cache

- V přímo mapované cache o velikosti 256 B je na 3. řádku uložen záznam
- **Klíč:** 101001010111011101101111 **Data:** B9
- Z jaké adresy byl tento bajt do cache zkopírován?
- Data nejsou v cache ukládána po blocích. Na každém řádku leží samostatně jeden bajt
- Cache má kapacitu 256 B a má tedy 256 řádků
- K určení třídy je třeba 8 bitů ($2^8=256$)
- Bajt leží na 3. řádku – bity, které určují třídu tedy musí být ve stavu **00000011**
- U záznamu je uložen 24-bitový klíč, který nám poskytuje informaci o tom, jak vypadá horních 24 bitů adresy
- Adresa tedy musí začínat bity 101001010111011101101111
- Kompletní 32-bitová adresa je 1010 0101 0111 0111 0110 1111 **0000 0011**
- Bajt B9 byl tedy do cache zkopírován z adresy **A5776F03h**



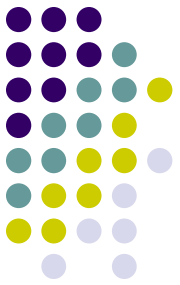
Přímo mapovaná cache

- V přímo mapované cache o velikosti 1 KB je na 19. řádku uložen záznam
- **Klíč:** 1010110000110101111101 **Data:** 7D
- Z jaké adresy byl tento bajt do cache zkopírován?
- Data nejsou v cache ukládána po blocích. Na každém řádku leží samostatně jeden bajt
- Cache má kapacitu 1024 B a má tedy 1024 řádků
- K určení třídy je třeba 10 bitů ($2^{10}=1024$)
- Bajt leží na 19. řádku – bity, které určují třídu tedy musí být ve stavu **0000010011**
- U záznamu je uložen 22-bitový klíč, který nám poskytuje informaci o tom, jak vypadá horních 22 bitů adresy
- Adresa tedy musí začínat bity 1010110000110101111101
- Kompletní 32-bitová adresa je **1010 1100 0011 0101 1111 0100 0001 0011**
- Bajt 7D byl tedy do cache zkopírován z adresy **AC35F413h**



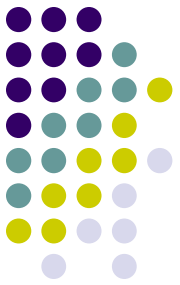
Přímo mapovaná cache

- V přímo mapované cache o velikosti 1 KB je na 19. řádku uložen záznam
- **Klíč:** 1010110000110101111101 **Data:** 7D F3
- Z jaké adresy byly tyto bajty do cache zkopírovány?
- Data jsou v cache ukládána po **blocích o velikosti 2B**
- Cache má kapacitu 1024 B a má tedy **512 řádků**
- K určení třídy je třeba 9 bitů ($2^9=512$)
- Bajt leží na 19. řádku – bity, které určují třídu tedy musí být ve stavu **000010011**
- U záznamu je uložen 22-bitový klíč, který nám poskytuje informaci o tom, jak vypadá horních 22 bitů adresy
- Adresa bloku tedy musí začínat bity 1010110000110101111101 a musí končit bitem 0 (aby byla dělitelná číslem 2)
- Kompletní 32-bitová adresa je *1010 1100 0011 0101 1111 0100 0010 0110*
- Bajt 7D byl do cache zkopírován z adresy **AC35F426h**
- Bajt F3 byl do cache zkopírován z adresy **AC35F427h**



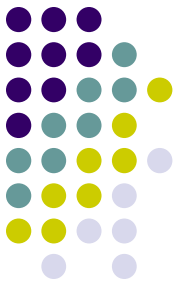
Přímo mapovaná cache

- V přímo mapované cache o velikosti 2 KB je na 6. řádku uložen záznam
- **Klíč:** 111100000101001101101 **Data:** 45 A2 C9 BD E8 00 F3 19
- Z jaké adresy byly tyto bajty do cache zkopírovány?
- Data jsou v cache ukládána po **blocích o velikosti 8B**
- Cache má kapacitu 2048 B a má tedy **256 řádků**
- K určení třídy je třeba 8 bitů ($2^8=256$)
- Bajt leží na 6. řádku – bity, které určují třídu tedy musí být ve stavu **00000110**
- U záznamu je uložen 21-bitový klíč, který nám poskytuje informaci o tom, jak vypadá horních 21 bitů adresy
- Adresa bloku tedy musí začínat bity 111100000101001101101 a musí končit bity 000 (aby byla dělitelná číslem 8)
- Kompletní 32-bitová adresa je 1111 0000 0101 0011 0110 1000 **0011 0000**
- Bajt 45 byl do cache zkopírován z adresy **F0536830h**
- Bajt A2 byl do cache zkopírován z adresy **F0536831h**
- Bajt C9 byl do cache zkopírován z adresy **F0536832h**
- Bajt BD byl do cache zkopírován z adresy **F0536833h**
- Bajt E8 byl do cache zkopírován z adresy **F0536834h**
- Bajt 00 byl do cache zkopírován z adresy **F0536835h**
- Bajt F3 byl do cache zkopírován z adresy **F0536836h**
- Bajt 19 byl do cache zkopírován z adresy **F0536837h**



Přímo mapovaná cache

- V přímo mapované cache o velikosti 1 MB je na 21. řádku uložen záznam
- **Klíč:** 110010101001 **Data:** 45 A2 C9 BD E8 00 F3 19 95 F2 C4 AA F7 20 D4 26
- Z jaké adresy byl do cache zkopírován označený bajt 00?
- Data jsou v cache ukládána po **blocích o velikosti 16B = 2⁴ B**
- Cache má kapacitu 1 MB = 2²⁰ B a má tedy 2²⁰ / 2⁴ = **2¹⁶ řádků**
- K určení třídy je třeba **16 bitů**
- Bajt leží na 21. řádku – bity, které určují třídu tedy musí být ve stavu **0000000000010101**
- U záznamu je uložen 12-bitový klíč, který nám poskytuje informaci o tom, jak vypadá horních 12 bitů adresy
- Adresa bloku tedy musí začínat bity 110010101001 a musí končit bity 0000 (aby byla dělitelná číslem 16)
- Kompletní 32-bitová adresa je 1100 1010 1001 **0000 0000 0001 0101 0000**
- Bajt 45 byl do cache zkopírován z adresy **CA900150h**
- Označený bajt 00 leží o 5 adres dále
- Bajt **00** byl do cache zkopírován z adresy **CA900155h**



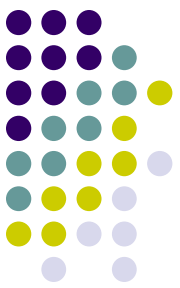
Přímo mapovaná cache

- Přímo mapovaná cache má kapacitu 4 kB, data jsou ukládána v blocích o velikosti 16 B
- Mohou být v této cache uložena zároveň data z adresy 12341234h a 9876523Bh ?
- 12341234h = 00010010001101000001 001000110100 b
- 9876523Bh = 10011000011101100101001000111001 b

- Uložené bloky dat musí začínat adresou, která je dělitelná číslem 16
- Bajt z adresy 12341234 bude uložen v rámci bloku, který začíná adresou 12341230h
- 12341230h = 00010010001101000001001000110000 b

- Bajt z adresy 9876523Bh bude uložen v rámci bloku, který začíná adresou 98765230h
- 9876523Bh = 10011000011101100101001000110000 b

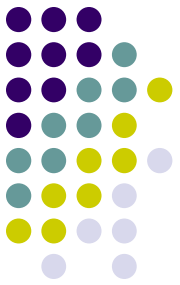
- Cache má kapacitu 4096 bajt a protože jsou data uložena v blocích 16 B, musí mít 256 řádků
- K určení třídy je třeba 8 bitů ($2^8=256$)
- Třidu určíme pomocí posledních používaných 8 bitů adresy
- Pro adresu 12341230h to bude 00010010001101000001**100100011**
- Pro adresu 9876523Bh to bude 10011000011101100101**100100011**
- **Obě dvě adresy patří do stejné třídy!** Uvedené bloky dat mohou v cache ležet pouze na 35. řádku
- **Není možné v této cache uložit zároveň bajt z adresy 12341234h a 9876523Bh**



Vícecestná cache

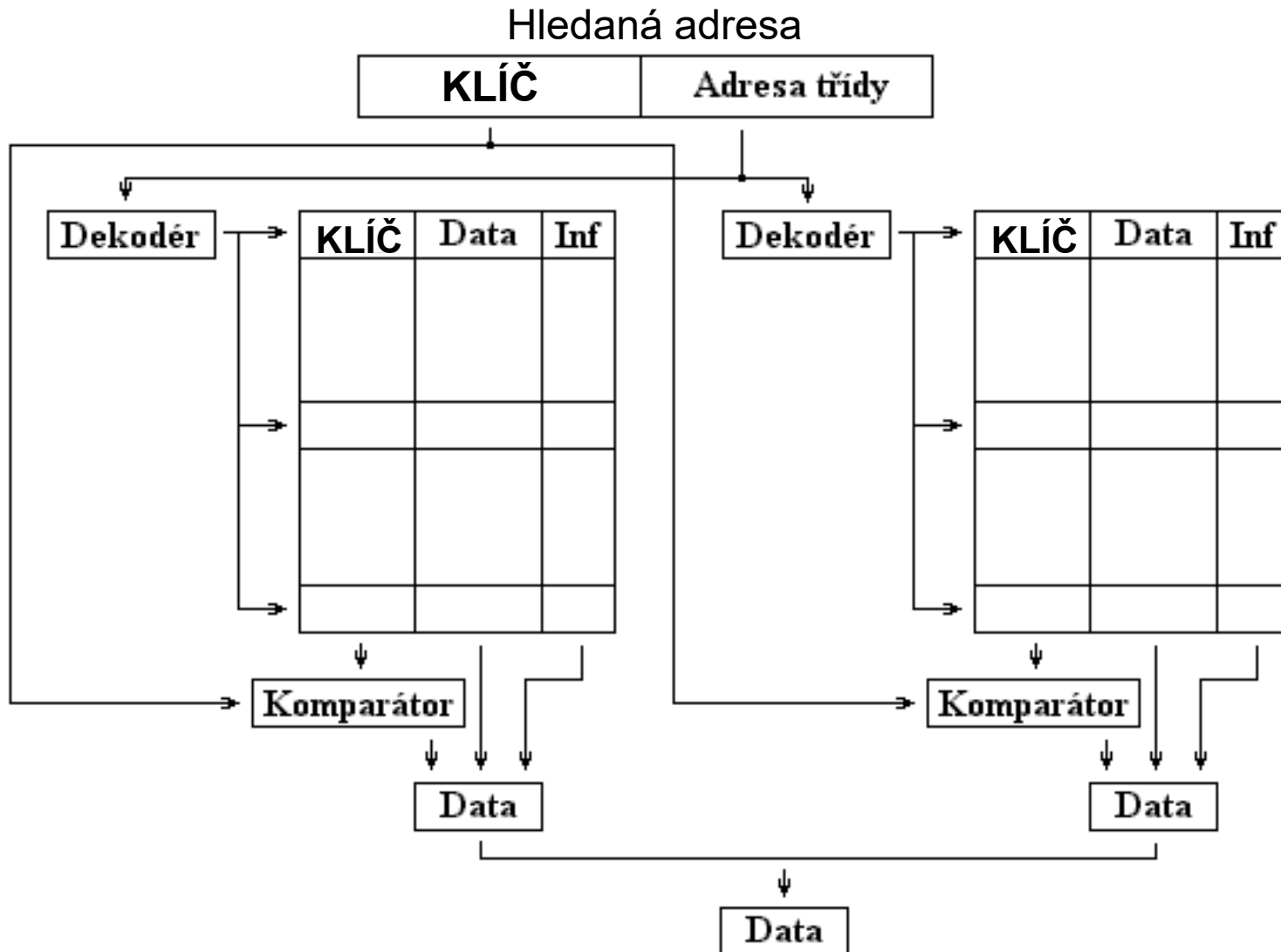
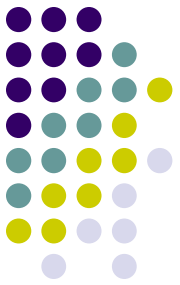
- Z předchozího příkladu vyplývá **zásadní nedostatek** přímo mapované cache paměti
- Data lze do cache uložit pouze na jeden konkrétní řádek dle třídy adresy
- Tento řádek ale již může být **obsazený** úplně jinými daty z úplně jiné adresy, která však patří do **stejné třídy**
- **Nelze uložit data z více různých adres patřících do stejné třídy**
- Ze všech adres patřících do stejné třídy může být v cache uložena pouze jedna
- Řešení, které odstraní uvedený problém – v cache bude **více tabulek**
- Cache, která má **více tabulek** = **vícecestná cache**
- Příklad:
- Bude-li cache tvořena dvěma tabulkami (dvoucestná cache), lze do ní uložit dva různé bloky dat patřící do stejné třídy
- Například data mají uložena na 17. řádek, ale ten je už obsazený – nevadí, je tu ještě druhá tabulka, ve které je 17. řádek volný
- Taková cache bude vyžadovat dva komparátory - víme, na jakém řádku data hledat, ale je třeba hledat ve dvou tabulkách současně

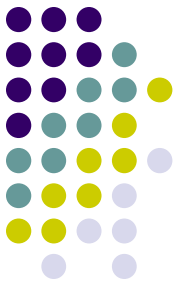
Vícecestná cache (s omezeným stupněm asociativity)



- Paměť s omezeným stupněm asociativity je organizována stejně jako přímo mapovaná cache, s tím rozdílem, že je tzv. **vícecestná** – jakoby zde bylo několik přímo mapovaných cache paralelně
- **stupeň asociativity** se obvykle značí písmenem ***n*** nebo ***a*** (např. ***n=2***)
- Adresa třídy je přivedena na *n* dekodérů které v každé tabulce vyberou stejný řádek
- Z těchto řádků se potom vezmou příslušné klíče a komparátory se porovnají s hledaným klíčem.
- Podobně jako u plně asociativních cache pamětí, pokud jeden z komparátorů signalizuje shodu, je informace v cache paměti přítomna
- Vícecestné paměti jsou **nejpoužívanějším** typem cache pamětí

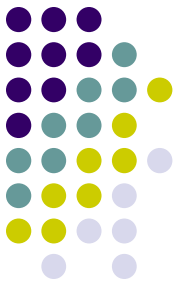
Cache se stupněm asociativity $n=2$ (dvoucestná)





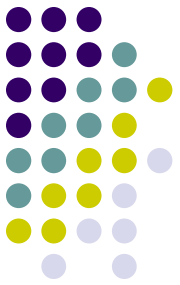
Vícecestná cache

- **Příklad**
- Přímo mapovaná cache má kapacitu **1 kB a je dvoucestná** Data jsou v cache ukládána po jednom bajtu (tedy nejsou ukládána po blocích)
- Na kolikátém řádku v cache bude uložen bajt z adresy 12345678h?
- 12345678h = 00010010001101000101011001111000 b
- Cache je dvoucestná – je tvořena dvěma tabulkami 2 x 512 B = 1024 B = 1 KB
- Každá tabulka má 512 řádků – adresy se rozdělí do 512 tříd
- K určení čísla třídy potřebujeme 9 bitů ($2^9 = 512$)
- Třidu určíme pomocí posledních 9 bitů adresy
- 00010010001101000101011 **001111000**
- Třidu tedy určují bity **001111000** – to dává číselně **120**
- Adresa 12345678h patří do třídy **120**
- Data z této adresy mohou být v cache uložena na **120. řádku**
- Jako **klíč** bude u tohoto záznamu uvedeno 23 bitů – 00010010001101000101011
- Data mohou ležet v jedné ze dvou tabulek
- Pokud chce mikroprocesor pracovat s bajtem s adresy 12345678h musí se podívat na **120. řádek v obou tabulkách** a pomocí **dvou komparátorů** zkontrolovat, zda se zde uložený 23-bitový klíč neshoduje s horními 23 bity hledané adresy



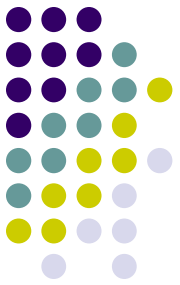
Vícecestná cache

- **Příklad**
- **Čtyřcestná** cache má kapacitu **8 kB = 8192 B**
- Data jsou v cache ukládána po **blocích o velikosti 8 B**
- Na kolikátém řádku v cache bude uložen bajt z adresy 12345678h?
- 12345678h = 00010010001101000101011001111000 b
- Cache má **4 tabulky** s kapacitou **2 KB** (4 x 2 KB = 8 KB)
- Každá tabulka má **256 řádků** (256 řádků x 8 B blok = 2048 B)
- K určení čísla třídy potřebujeme **8 bitů** ($2^8 = 256$ řádků)
- **Poslední tři bity adresy se nijak nevyužijí**, protože se do cache data ukládají po blocích velkých 8 bajtů
- Každý takový blok začíná adresou, která je **dělitelná číslem 8**, tedy víme, že adresa, kterou blok začíná **musí mít na konci tři nulové bity**
- 000100100011010001010 11001111 000
- Třidu tedy určují bity **11001111** – to dává číselně **207**
- Adresa 12345678h patří do třídy **207**
- Data z této adresy mohou být v cache uložena na **207. řádku** v jedné ze čtyř tabulek
- Jako **klíč** bude u tohoto záznamu uvedeno **21 bitů** – 000100100011010001010
- V cache musí být **4 komparátory**, kterými se porovnává 21-bitový klíč z vybraného řádku ve všech 4 tabulkách s horními 21 bity hledané adresy



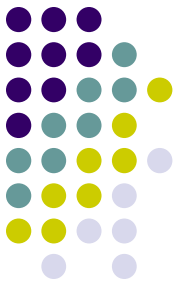
Vícecestná cache

- V **osmicestné** cache o velikosti 1 MB je na 21. řádku uložen záznam
- **Klíč:** 110010101001010 **Data:** 45 A2 C9 BD E8 00 F3 19 95 F2 C4 AA F7 20 D4 26
- Z jaké adresy byl do cache zkopírován označený bajt 00?
- Data jsou v cache ukládána po **blocích o velikosti 16B**
- Cache má kapacitu 1 MB a je tvořena **8 tabulkami**. V každé tabulce je uloženo **128 kB** dat (8 x 128 kB = 1 MB)
- Každá tabulka má **8192 řádků** (8192 řádků x 16 B blok = 128 kB)
- K určení třídy bude třeba **13 bitů** ($2^{13}=2048$)
- Bajt leží na 21. řádku – bity, které určují třídu tedy musí být ve stavu **0000000010101**
- U záznamu je uložen 15-bitový klíč, který nám poskytuje informaci o tom, jak vypadá horních 15 bitů adresy
- Adresa bloku tedy musí začínat bity 11001010100101010 a musí končit bity 0000 (aby byla dělitelná číslem 16)
- Kompletní 32-bitová adresa je 1100 1010 1001 010 **00000 0001 0101 0000**
- Bajt 45 byl do cache zkopírován z adresy **CA940150h**
- Označený bajt 00 leží o 5 adres dále
- Bajt **00** byl do cache zkopírován z adresy **CA940155h**



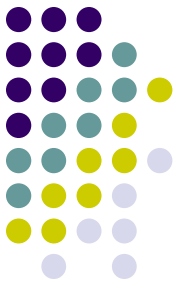
Vícecestná cache

- Jakou kapacitu má dvoucestná cache, do které se data ukládají v blocích o velikosti 8 B a používá 15-bitový klíč?
- Data se ukládají v blocích o velikosti 8 B – poslední tři bity adresy se nepoužívají
- 15 bitů adresy je uloženo jako klíč záznamu
- Fyzická adresa má 32 bitů
- $32 \text{ bitů} - 3 \text{ bity} - 15 \text{ bitů} = 14 \text{ bitů}$ pro určení třídy
- Jestliže je třída určena pomocí 14 bitů, pak tabulky musí mít 2^{14} řádků
- Cache má tedy dvě tabulky, každá tabulka má 2^{14} řádků a na každém řádku leží 8B dat
- **Kapacita** = $2 \times 2^{14} \times 8 \text{ B} = 2^1 \times 2^{14} \times 2^3 = 2^{18} \text{ B} = 256 \text{ kB}$



Vícecestná cache

- Jakou kapacitu má čtyřcestná cache, do které se data ukládají v blocích o velikosti 16 B a používá 12-bitový klíč?
- Data se ukládají v blocích o velikosti 16 B – poslední 4 bity adresy se nepoužívají
- 12 bitů adresy je uloženo jako klíč záznamu
- Fyzická adresa má 32 bitů
- $32 \text{ bitů} - 4 \text{ bity} - 12 \text{ bitů} = 16 \text{ bitů}$ pro určení třídy
- Jestliže je třída určena pomocí 16 bitů, pak tabulky musí mít 2^{16} řádků
- Cache má tedy 4 tabulky, každá tabulka má 2^{16} řádků a na každém řádku leží 16B dat
- **Kapacita** = $4 \times 2^{16} \times 16 \text{ B} = 2^2 \times 2^{16} \times 2^4 = 2^{22} \text{ B} = 4 \text{ MB}$

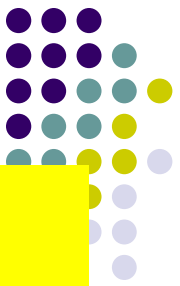


L1 Cache v 80486

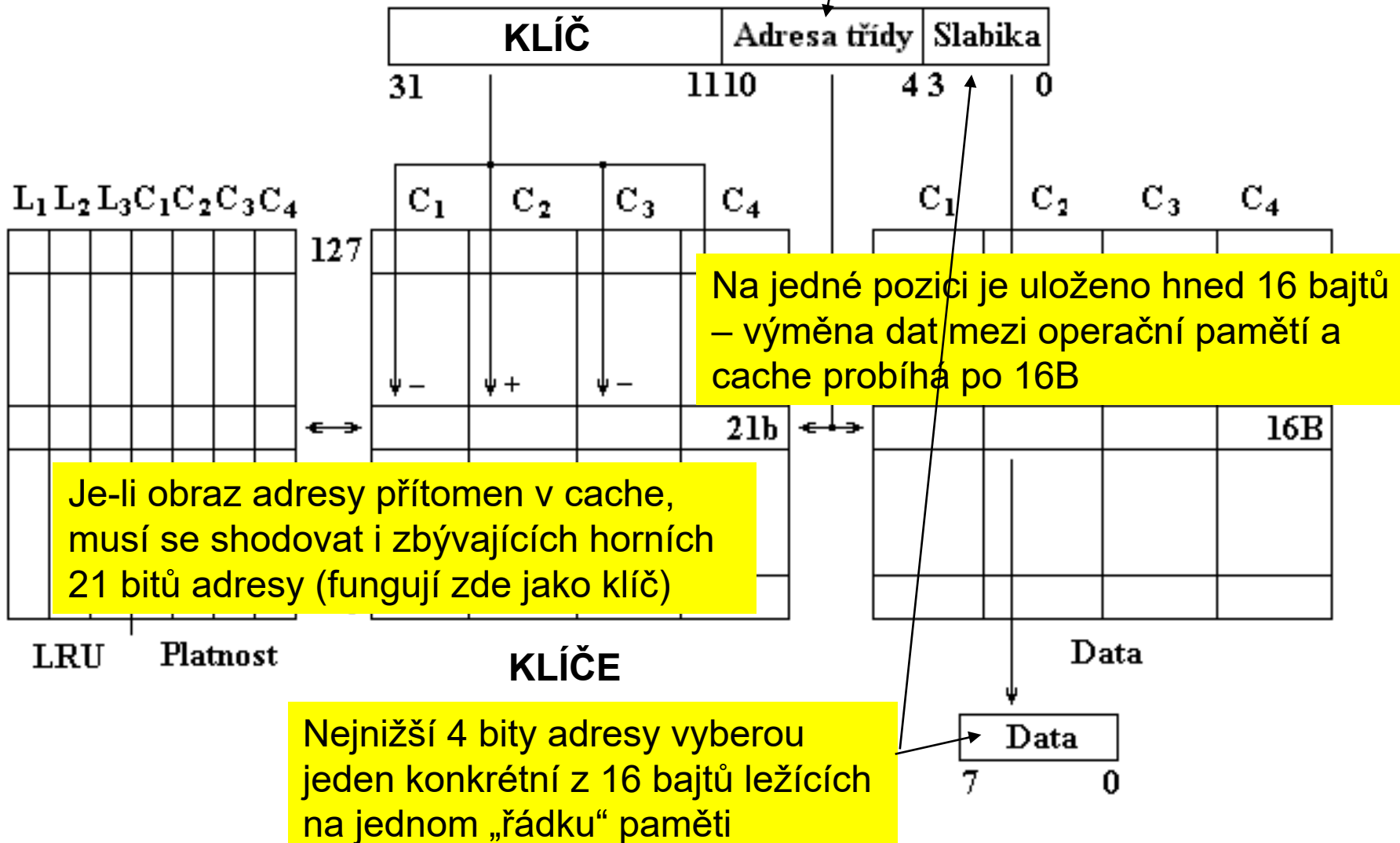
- Prvním mikroprocesorem, který má integrovanou interní cache je Intel 80486
- **Interní cache** procesoru 80486 má stupeň asociativity 4 (je čtyřcestná) a je rozdělena do 128 tříd
- Jeden záznam (blok) má šířku 16 bajtů
- Celková kapacita cache je 128 řádků x 4 cesty x 16 B = **8 kB**
- Fyzické adresy jsou rozděleny do **128 tříd** podle 4.-10. bitu (7 bitů)
- V paměti mohou být uloženy až 4 různé adresy spadající do stejné třídy, ale pozor, nejsou zde 4 tabulky, protože informace se přeskupily jiným způsobem

- Veškerá data jsou uložena v jedné datové tabulce
- Veškeré klíče jsou uloženy v jedné klíčové tabulce
- Veškeré další doplňkové informace jsou uloženy v jedné informační tabulce

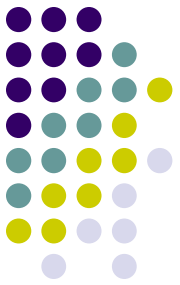
Cache s $n=4$ použitá jako interní L1 u



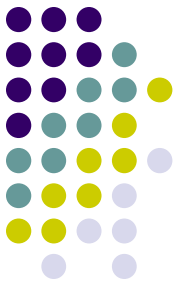
Adresa třídy (7 bitů) určuje, na které ze 128 možných pozic je v cache obraz této adresy



Kontrolní otázky



- Jaký typ (technologie) paměti se používá pro výrobu cache a proč ?
- Která úroveň cache bude nejrychlejší L1, L2 nebo L3 ?
- Která úroveň cache bude mít největší kapacitu L1, L2 nebo L3 ?
- Který mikroprocesor jako první obsahoval integrovanou interní cache ?
- Co je to LRU strategie ?
- Jak funguje write-back ? Proč je to lepší řešení než write-through ?
- Vysvětlete pojem "asociativní paměť"
- Proč se dnes prakticky nepoužívají plně asociativní cache paměti ?
- Plně asociativní paměť s kapacitou 1 kB používá blok dat 1B. Fyzická adresa má 32b. Jak široký bude klíč ? Kolik komparátorů bude paměť obsahovat ?
- Plně asociativní paměť s kapacitou 1 kB používá blok dat 2B. Fyzická adresa má 32b. Jak široký bude klíč ? Kolik komparátorů bude paměť obsahovat ? Jaký další bajt se do paměti přesune spolu s bajtem z adresy 12344321h ? Jaký klíč bude u tohoto dvoubajtového bloku uveden ?
- Plně asociativní paměť s kapacitou 1 kB používá blok dat 4B. Fyzická adresa má 32b. Jak široký bude klíč ? Kolik komparátorů bude paměť obsahovat ? Jaké další tři bajty (z jakých adres) se do paměti přesune spolu s bajtem z adresy 43211234h ? Jaký klíč bude u tohoto čtyřbajtového bloku uveden ?
- Plně asociativní paměť s kapacitou 2 kB používá blok dat 8B. Fyzická adresa má 32b. Jak široký bude klíč ? Kolik komparátorů bude paměť obsahovat ? Jakých dalších sedm bajtů (z jakých adres) se do paměti přesune spolu s bajtem z adresy ABCDABCDh ? Jaký klíč bude u tohoto osmibajtového bloku uveden ?



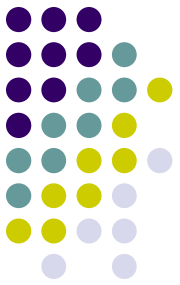
Kontrolní otázky

- V **přímo mapované** cache jsou uloženy bajty a klíče znázorněným způsobem. Paměť pracuje s blokem dat o velikosti 4B. Určete originální adresu dvou podtržených bajtů (na jaké fyzické adrese v operační paměti leží originál těchto bajtů). Zobrazeny jsou první dva řádky paměti Fyzická adresa je 32-bitová.

<u>Klíč</u>	<u>Bajty</u>
C2 6	A5 <u>C4</u> 00 2B
D4A	D2 A3 B5 <u>C1</u>

- Do osmicestné paměti Cache s kapacitou 512 kB byl přesunut bajt z adresy 12349876h. Paměť používá 4096 tříd.
 - Jaké další sousední bajty (z jakých adres) byly přesunuty do Cache spolu s bajtem z uvedené adresy ?
 - Jaký klíč je uveden u tohoto bloku ?
 - Na kolikátém řádku tabulky cache bude blok uložen ?
 - Kolik komparátorů musí obsahovat tato cache ?

Kontrolní otázky



- Vysvětlíte, proč třídu určují nižší bity a jako klíč jsou použity nejvyšší bity fyzické adresy a ne naopak
- Kolikabitový klíč bude použit v paměti cache se stupněm asociativity 3, počet tříd=1024, fyzická adresa je 32-bitová, ukládá se blok 16 bajtů.
- Jakou kapacitu má paměť cache, platí-li $n=2$, počet tříd=256, cachuje se blok 4 bajtů, fyzická adresa je 32-bitová
- Jaký stupeň asociativity používá 16 kB cache paměť, počet tříd=512, ukládá se blok 8 bajtů. Dále určete které bity adresy budou klíčem a které určují třídu
- Jakou kapacitu má 4-cestná cache paměť, jestliže se používá 12b klíč a data se ukládají v blocích 16 B ?
- Cache paměť má kapacitu 128 kB, je jednocestná a používá blok 16B. Mohou ležet v cache paměti současně bajty z adres 1ABC1234h a 12344567h ?
- Cache paměť má kapacitu 128 kB, je dvoucestná a používá blok 16B. Mohou ležet v cache paměti současně bajty z adres 1ABCDEF1h a 1234ABCDh ?
- Do paměti cache byl uložen bajt z adresy 2345AABBh. Paměť používá 8-bajtové bloky dat. Jaké další adresy byly spolu s tímto bajtem v bloku zkopírovány do cache ?