

# Architektury systemów komputerowych

Lista 12

$$x_{12} = 6 \text{ (minimum na bdb)}$$

1. W tym i w następnym zadaniu rozważamy 4-gigabajtową przestrzeń adresową, adres określa pojedynczy bajt pamięci, a więc ma 32 bity. Przyjrzymy się kilku wariantom bezpośredniego mapowania przestrzeni adresowej w pamięci cache.

Założmy, że mamy do dyspozycji pamięć cache, która może przechowywać 16 kB danych (nie liczymy miejsca przeznaczonego na znaczniki). Każdy bajt pamięci cache jest adresowany osobno, a więc każdy musi posiadać własny znacznik. Ile bitów ma adres bajtu w pamięci cache, a ile bitów musi mieć znacznik? Dlaczego w naszej sytuacji nie musimy wyróżniać fragmentu adresu określającego położenie bajtu w bloku? Zakładając, że znacznik to bardziej znacząca część adresu, powiedz pod jakim adresem w pamięci cache może być przechowywana zawartość rzeczywistego adresu 1A2B3C4D (podanego szesnastkowo)? Ilu przerzutników potrzeba aby zrealizować naszą pamięć cache (łącznie ze znacznikami)? Przypominam, że przerzutnik pamięta jeden bit.

2. Założmy teraz, że adresowalną jednostką pamięci cache jest wiersz (blok) złożony z czterech 4-bajtowych słów. Ile wierszy ma nasza 16 kB pamięć cache? Gdzie w pamięci cache będzie przechowana zawartość adresu 1A2B3C4D (w którym wierszu, w którym słowie tego wiersza, w którym bajcie tego słowa)? Ilu przerzutników potrzeba aby zrealizować naszą pamięć cache (łącznie ze znacznikami)?
3. Rozważmy maszynę, która ma 1Mb pamięci głównej, słowo o długości jednego bajta, pamięć cache o pojemności 64kB i rozmiarze wiersza (bloku) 16 bajtów, mapowanie jest bezpośrednie. Jaki znacznik, adres bloku oraz pozycję słowa w bloku mają następujące adresy pamięci głównej: F0010 oraz 01234 (podane szesnastkowo)? Podaj dowolne dwa adresy o różnych znacznikach, które są mapowane do tego samego wiersza pamięci cache.
4. Rozważamy cache z mapowaniem sekcyjno-skojarzeniowym. Mamy 64 wiersze cache podzielone na sekcje o wielkości czterech wierszy. W pamięci głównej mamy 4K bloków, każdy składający się ze 128 słów. Jaki jest format adresu w pamięci głównej?
5. Przypomnij jaka jest różnica pomiędzy schematem zapisu jednoczesnego (*write through*), a zapisu opóźnionego (*write back*). Czas dostępu do pamięci cache wynosi 5 ns, a do pamięci głównej 40 ns (zakładamy dla uproszczenia, że w tym czasie jesteśmy w stanie przenieść cały blok). Przewidujemy, że 85 % odwołań programu do pamięci to odczyty, 15 % – zapisy. Zakładamy, że współczynnik trafień w pamięci cache wynosi 0.95 (zarówno w przypadku odczytów, jak i zapisów). Jaki jest średni czas dostępu do pamięci w systemie przy obydwu schematach zapisu (doprecyzuj szczegóły dotyczące realizacji obydwu schematów, jeśli to konieczne).
6. Rozważamy pamięć podręczną z mapowaniem sekcyjno-skojarzeniowym, wykorzystującą algorytm wymiany LRU (*least recently used* – najdawniej używany). Rozmiar sekcji – 4 bloki. Do każdego bloku dodane są dwa bity, używane do obliczenia, który blok był ostatnio używany. Licznik działa następująco:
  - kiedy pojawi się „chybienie”, to blok z licznikiem ustawiony na 0 jest podmieniany, a nowo wstawiony blok dostaje licznik 3; pozostałe bloki mają liczniki zmniejszane o 1;
  - kiedy pojawi się „trafienie”, to licznik bloku, do którego się odwołujemy ustawiamy na 3 i wprowadzmy konieczne modyfikacje w pozostałych licznikach.

Jeśli w pamięci cache znajdują się cztery bloki A, B, C, D z licznikami ustawionymi odpowiednio na 0, 1, 2, 3, to które bloki pozostaną w zbiorze i jakie będą wartości ich liczników po sekwencji odwołań: E, B, E, D, A, E?

7. Algorytm wymiany stosowany w systemach z procesorem Intel 80406 jest określany jako „pseudo-LRU”. Z każdą ze 128 sekcji zawierających po 4 wiersze (oznaczone jako L1, L2, L3, L4) są skojarzone trzy bity (zwracam uwagę, że bity są skojarzone z całą sekcją a nie z pojedynczym wierszem): B0, B1, B2. Bit B0 określa która para wierszy (0-1, czy 2-3) była używana dawniej, bity B1 i B2 wskazują, które wiersze w poszczególnych parach były wykorzystywane dawniej. Algorytm, w zależności od B0, wyrzuca wiersz wskazywany przez B1 lub B2.
- (a) Pokaż, że algorytm stanowi tylko przybliżenie prawdziwego algorytmu LRU (wskaz sekwencję odwołań, która rozróżnia te dwa algorytmy).
  - (b) Wykaż, że prawdziwy algorytm LRU wymaga co najmniej 6 bitów na jeden zbiór.
8. System komputerowy zawiera pamięć główną o pojemności 32K słów 16-bitowych. Ma także pamięć podręczną 4K słów podzieloną na 4-wierszowe sekcje z 64 słowami w każdym wierszu. Załóż, że pamięć podręczna jest na początku pusta. Procesor pobiera słowa z lokacji 0,1,2,...,4351 w tym właśnie porządku. Następnie powtarza tę sekwencję pobierania jeszcze 9 razy. Pamięć podręczna jest 10 razy szybsza niż pamięć główna. Oszacuj korzyść wynikającą z zastosowania pamięci podręcznej. Przyjmij, że wymiana bloków jest zgodna z algorytmem LRU.
9. Rozważmy następujący pseudokod:

```
int array[10000,100000];
for each element array[i][j] {
    array[i][j]=array[i][j]*2
}
```

Napisz dwa programy w języku C implementujące powyższy algorytm: Pierwszy program powinien przechodzić przez tablicę wierszami, a drugi kolumnami. Porównaj czasy działania obydwu implementacji. Skomentuj wyniki.

*Emanuel Kieroński*