

Architektury systemów komputerowych

Lista 13

$$x_{13} = 3 \text{ (minimum na bdb)}$$

1. Rozważmy system pamięci zbudowany z jednopoziomowej pamięci cache oraz pamięci głównej. Pamięć cache jest sekcyjno-skojarzeniowa, składa się z czterech dwuwierszowych sekcji, algorytm wymiany to LRU. Rozmiar bloku (wiersza) to cztery słowa. Pamięć główna mieści 1024 słowa. Adresowalną jednostką jest słowo. Zakładamy, że czas odczytu z pamięci cache to 5 ns, natomiast czas przeniesienia całego bloku z pamięci głównej do cache wynosi 95 ns. Przyjmij, że w przypadku chybienia w pamięci cache najpierw wczytywany jest cały blok, a potem jeszcze odpowiednie słowo przenoszone jest do procesora (trwa to zatem w sumie 100 ns).
 - (a) Przedstaw format adresu w tym systemie (podziel adres na logiczne fragmenty, podaj ich długości w bitach)
 - (b) Rozważ wykonanie programu, który odczytuje kolejno słowa 2, 3, 4, 5, ..., 39, a następnie wykonuje pięciokrotnie pętlę odczytującą słowa 40, 41, ..., 47. Ile chybień w pamięci cache wystąpi podczas wykonywania tego programu? Jaki jest średni czas odczytu słowa w tym programie (wystarczy przedstawić wzór, nie trzeba koniecznie wyliczać dokładnej wartości)? Zakładamy, że przed wykonaniem programu pamięć cache jest pusta.
2. System pamięci wirtualnej ma rozmiar strony 1024 słowa, 8 wirtualnych stron i 4 fizyczne ramki (miejsca w pamięci głównej, w które można wstawić stronę). Tablica stron jest następująca:

Wirtualny numer strony	Numer ramki
0	3
1	1
2	-
3	-
4	2
5	-
6	0
7	-

- (a) sporządź listę wszystkich wirtualnych adresów, które mogą spowodować błędy stron
 - (b) jakie są adresy pamięci głównej dla następujących adresów wirtualnych: 0, 3782, 1023, 1024, 1025, 7800, 4096?
3. Dysponujesz systemem pamięci wirtualnej zawierającej 2-rekordowy bufor TLB, pamięcią podręczną używającą dwudroźnego (dwublokowego) mapowania sekcyjno-skojarzeniowego i tabelą stron dla procesu P. Przyjmij, że bloki w pamięci podręcznej liczą 8 słów, natomiast rozmiar strony wynosi 16 słów. W poniższym systemie pamięć główna została podzielona na bloki, z których każdy jest identyfikowany przy użyciu litery. Dwa bloki odpowiadają jednej ramce.

Strona	Ramka
0	3
4	1

Bufor TLB

Zbiór 0	znacznik	C	znacznik	I
Zbiór 1	znacznik	D	znacznik	H

Pamięć podręczna

Indeks	Ramka	Bit aktualności
0	3	1
1	0	1
2	-	0
3	2	1
4	1	1
5	-	1
6	-	1
7	-	1

Tabela stron

Ramka		Blok
0	C	0
	D	1
1	I	2
	J	3
2	G	4
	H	5
3	A	6
	B	7

Pamięć główna

Strona		Blok
0	A	0
	B	1
1	C	2
	D	3
2	E	4
	F	5
3	G	6
	H	7
4	I	8
	J	9
5	K	10
	L	11
6	M	12
	N	13
7	O	14
	P	15

Pamięć wirtualna przydzielona procesowi P

Dysponując powyższym opisem stanu systemu, odpowiedz na następujące pytania:

- Ile bitów jest zawartych w adresie wirtualnym powiązanim z procesem P.
- Ile bitów jest zawartych w adresie fizycznym powiązanim z procesem P.
- Utwórz format wirtualnego adresu 18 (wszystkie adresy podawane są w tym zadaniu dziesiętkowo) – podaj nazwy i rozmiary pól, którym system posłuży się przy translacji tego adresu na adres fizyczny, a następnie zamień ten adres na odpowiadający mu adres fizyczny.
- Określony adres wirtualny 6 jest zamieniony na adres fizyczny 54. Utwórz format adresu fizycznego (z uwzględnieniem nazw i rozmiarów pól), który jest używany do określania dla adresu lokacji w pamięci podręcznej. Wyjaśnij w jaki sposób użyć tego formatu, aby określić lokalizację adresu fizycznego 54 w pamięci podręcznej.

- (e) Określony adres wirtualny 25 jest zlokalizowany na wirtualnej stronie 1, przesunięciu 9. Opisz dokładnie, w jaki sposób adres ten zostanie zamieniony na odpowiadający mu adres fizyczny i jak będzie realizowany dostęp do danych. Uwzględnij rolę, jaką w operacji odgrywa bufor TLB, tabela stron oraz pamięć podręczna i główna.
4. Dysponując systemem pamięci wirtualnej, zawierającym bufor TLB, pamięć podręczną i tablicę stron, przyjmij:
- trafienie w buforze TLB zajmuje 0.5 ns
 - trafienie w pamięci podręcznej zajmuje 1 ns
 - odwołanie do pamięci zajmuje 40 ns
 - odwołanie do dysku zajmuje 100 ms (operacja obejmuje uaktualnienie tabeli stron, pamięci podręcznej i bufora TLB)
 - współczynnik trafień w buforze TLB wynosi 99%
 - współczynnik trafień w pamięci podręcznej wynosi 96 %
 - współczynnik błędów strony wynosi 0,0001 %
 - w przypadku wystąpienia chybienia w buforze TLB lub pamięci podręcznej wymagany czas dostępu uwzględnia czas potrzebny na uaktualnienie bufora TLB i (lub) pamięci podręcznej, ale operacja dostępu **nie** jest wykonywana od początku
 - w przypadku wystąpienia błędu strony jest ona pobierana z dysku, są wykonywane wszystkie operacje uaktualnienia, ale operacja dostępu **jest** wykonywana od początku

Dla każdego wymienionego poniżej przypadku stwierdź, czy jest możliwe jego wystąpienie. Jeśli tak, podaj czas wymagany do uzyskania dostępu do wymaganych danych.

- (a) trafienie w buforze TLB i pamięci podręcznej
(b) chybienie w buforze TLB, trafienie w tabeli stron i pamięci podręcznej
(c) chybienie w buforze TLB i pamięci podręcznej, trafienie w tabeli stron
(d) chybienie w buforze TLB i tabeli stron, trafienie w pamięci podręcznej
(e) chybienie w buforze TLB i tabeli stron

Poniżej przedstawiam jeszcze kilka dodatkowych zadań dotyczących kodów detekcyjno-korekcyjnych. Może ktoś będzie miał na nie ochotę. Materiał ten jest nadprogramowy i nie obowiązuje na egzaminie.

Najprostszym sposobem wykrywania pojedynczych błędów podczas transmisji jest dodawanie do przesyłanych informacji tzw. bitu parzystości. Oczywiście ten sposób nie daje możliwości automatycznego poprawiania błędów. Istnieją jednak kody, które, po wykryciu przekłamania, potrafią je od razu poprawić.

5* (2 pkt.) Opisz działanie kodu Hamminga poprawiającego pojedyncze błędy. Kod Hamminga można zastosować np. do znaków ASCII - do każdego kodu znaku dopisywane są cztery dodatkowe bity; jeżeli podczas transmisji wystąpi pojedynczy błąd, błąd ten można automatycznie poprawić, bez potrzeby ponownego przesyłania (odczytu).

6* W kodzie korekcyjnym/detekcyjnym tylko niektóre ciągi bitów reprezentują bezpośrednio znaki oryginalnego alfabetu. W kodzie parzystości są to te ciągi, które mają parzystą liczbę jedynek. Odległością Hamminga dla pary ciągów bitów nazywamy liczbę pozycji, na których się one różnią. Np. odległość Hamminga pomiędzy 010101010 a 010101100 wynosi 2. Odległością Hamminga dla całego kodu (rozumianego jako zbiór ciągów bitów tej samej długości) nazywamy minimalną odległość Hamminga dla dwóch ciągów reprezentujących znaki oryginalnego alfabetu. Np. dla kodu parzystości odległość Hamminga wynosi 2.

Jaką odległość Hamminga powinien mieć kod, który umie

(a) Wykrywać p przekłamań

(b) Poprawiać p przekłamań

7* Załóżmy, że oryginalny alfabet jest kodowany na m bitach, a kod korygujący pojedyncze błędy dodaje r bitów nadmiarowych. Wykaż, że zachodzi następująca zależność:

$$m + r + 1 \leq 2^r.$$

Korzystając z tej zależności pokaż, że nie da się zaprojektować kodu korygującego pojedyncze błędy dla alfabetu 128-znakowego używającego mniej niż 4 bitów dodatkowych.

Emanuel Kieroński