

# Architektury systemów komputerowych

## Ćwiczenia 5: "Pamięć podręczna i wirtualna"

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Hennesy & Patterson (5 edycja): 5.3, 5.4, 5.7, 5.8

Należy być przygotowanym do wytłumaczenia **wytłuszczonych** haseł.

### Zadanie 1

Rozważmy pamięć podręczną z **mapowaniem bezpośrednim adresowaną bajtowo**. Używamy adresów 32-bitowych o następującym formacie:  $(tag, index, offset) = (addr_{31..10}, addr_{9..5}, addr_{4..0})$ .

- Jaki jest rozmiar **bloku** w 32-bitowych słowach?
- Ile wierszy ma nasza pamięć podręczna?
- Jaki jest stosunek ilości bitów składających dane do ilości bitów składających **metadane**?

### Zadanie 2

Rozważmy pamięć podręczną z poprzedniego zadania. Mamy następującą sekwencję odwołań do słów pamięci:

0 4 16 132 232 160 1024 30 140 3100 180 2180

Załóż, że na początku cache jest pusty. Jak wiele bloków zostało **zastąpionych**? Jaka jest efektywność pamięci podręcznej (ilość **trafień** procentowo)? Podaj zawartość pamięci podręcznej po wykonaniu powyższych odwołań – każdy **ważny wpis** wypisz jako krotkę  $(tag, index, ...)$ . Dla każdego chybiecia wskaż, czy jest ono przymusowe (ang. **compulsory miss**), wynika z ograniczonej pojemności pamięci (ang. **capacity miss**) czy też kolizji na danym adresie (ang. **conflict miss**).

### Zadanie 3

Powtórz poprzednie zadanie dla następujących organizacji pamięci podręcznej:

- **sekcyjno-skojarzeniowa** 3-droźna, bloki długości dwóch słów, ilość bloków 24, **polityka wymiany** LRU;
- **w pełni asocjacyjna**, bloki długości słowa, ilość bloków 8, polityka wymiany LRU.

### Zadanie 4

Dysponujemy pamięcią cache typu: L1 – **write-through, write no-allocate**; L2 – **write-back, write allocate**.

- Opisz dokładnie procedurę obsługi chybiecia w cache L1. Rozważ wszystkie elementy systemu, które mogą być wykorzystane i możliwość zastępowania **zmodyfikowanego bloku**.
- Powtórz powyższe polecenie dla wielopoziomowej rozłącznej (ang. **exclusive**) pamięci podręcznej (tj. blok może występować tylko na co najwyżej jednym poziomie).

### Zadanie 5

Wiemy, że im większa pamięć podręczna tym dłuższy czas dostępu do niej. Załóżmy, że dostęp do pamięci głównej trwa 70ns, a dostępy do pamięci stanowią 36% wszystkich instrukcji. Rozważmy system z pamięcią podręczną o następującej strukturze: L1 – 2 KiB, współczynnik chybień 8.0%, czas dostępu 0.66ns; L2 – 1 MiB, współczynnik chybień 0.5%, czas dostępu 5.62ns. Odpowiedz na pytania:

- Jaki jest średni czas dostępu do pamięci dla procesora tylko z cache L1, a jaki dla procesora z L1 i L2?
- Zakładając, że procesor charakteryzuje się CPI (ang. **clocks per instruction**) na poziomie 1.0 bez robienia dostępow do pamięci, oblicz CPI dla procesora tylko z cache L1 i dla procesora z L1 i L2.

### Zadanie 6

Niech system posługuje się 32-bitowymi adresami wirtualnymi, rozmiar strony ma 4KiB, a rozmiar wpisu tablicy stron ma 4 bajty. Dla procesu, który łącznie używa 1GiB swojej przestrzeni adresowej podaj rozmiar tablicy stron: (a) jedno-poziomowej, (b) dwupoziomowej, gdzie **katalog tablicy stron** ma 1024 wpisy. Dla drugiego przypadku – jaki jest maksymalny i minimalny rozmiar tablicy stron?

## Zadanie 7

W tym zadaniu będziemy analizowali w jaki sposób system operacyjny musi aktualizować **tablicę stron** wraz z kolejnymi dostęпами do pamięci głównej. Załóż, że strony są wielkości 4KiB, **TLB** jest w pełni asocjacyjne z zastępowaniem LRU. Jeśli potrzebujesz **wtoczyć stronę** z dysku użyj następnego numeru większego od największego istniejącego w tablicy stron.

Początkowy stan TLB:

Ważna	Tag	Numer strony
1	11	12
1	7	4
1	3	6
0	4	9

Początkowy stan tablicy stron:

Ważna	Numer strony lub na dysku
1	5
0	Dysk
0	Dysk
1	6
1	9
1	11
0	Dysk
1	4
0	Dysk
0	Dysk
1	3
1	12

Adresy dostępow do pamięci:

4669
2227
13916
34587
48870
12608
49225

- Dla powyższych danych podaj ostateczny stan systemu po wykonaniu wszystkich dostępow do pamięci. Dla każdej operacji dostępu do pamięci wskaż czy było to **trafienie w TLB**, trafienie w tablicę stron, czy też **błąd strony**.
- Powtórz poprzednie polecenie przy założeniu, że strona ma 16KiB. Jakie wady i zalety ma stosowanie większych stron?

## Zadanie 8

Projektant pamięci podręcznej ma zwiększyć pojemność 4KiB pamięci podręcznej **indeksowanej adresami wirtualnymi o znacznikach będących adresami fizycznymi** (VIPT). Wyłumacz w jaki sposób taka pamięć podręczna współpracuje z mechanizmem translacji adresów. Jeśli nasz system posiada 8KiB strony to czy możemy zaprojektować 16KiB cache mapowany bezpośrednio przy założeniu, że blok składa się z dwóch 32-bitowych słów? W jaki sposób projektant mógłby zwiększyć pojemność pamięci podręcznej?

## Zadanie 9 [\$5.10, bonus]

Wyjaśnij na czym polega problem współdzielenia pamięci w systemach wieloprocesorowych – tzw. problem spójności pamięci podręcznej (ang. *cache coherence*). Następnie wyłumacz na czym polega różnica między protokołami kontroli spójności – z **podglądaniem** i **katalogowym**. Narysuj diagram protokołu MESI i dla poszczególnych krawędzi podaj zdarzenia, które inicjują zmianę stanu.

## Zadanie 10 [\$5.13, bonus]

Wyszukaj w Internecie dokładnych informacji na temat organizacji systemu pamięci (adresy wirtualne, pamięć podręczna, TLB) w współczesnych procesorach (np. Intel, ARM, POWER). Posługując się uproszczonym diagramem organizacji systemu pamięci odpowiedz na pytania dotyczące:

- szerokości adresów wirtualnych i fizycznych, rozmiaru strony i organizacji tablicy stron,
- ilości poziomów, pojemności i organizacji pamięci podręcznej i TLB, czasu obsługi trafienia,
- polityki wymiany, alokacji przy zapisie, przydziału rozłącznego (ang. *exclusive cache*),
- indeksowania i znaczników pamięci podręcznej (VIVT / VIPT / PIPT).