Organizacja i architektura komputerów [1]

Wykład 12

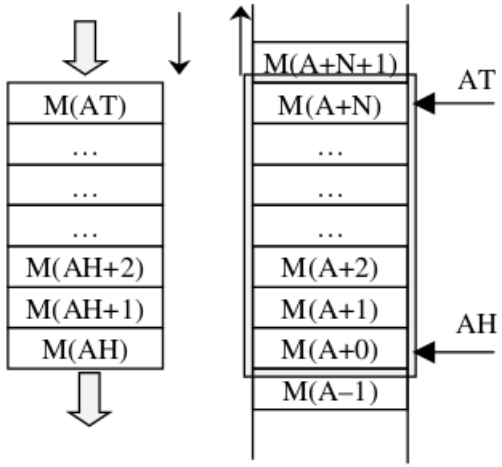
Piotr Patronik

29 maja 2015

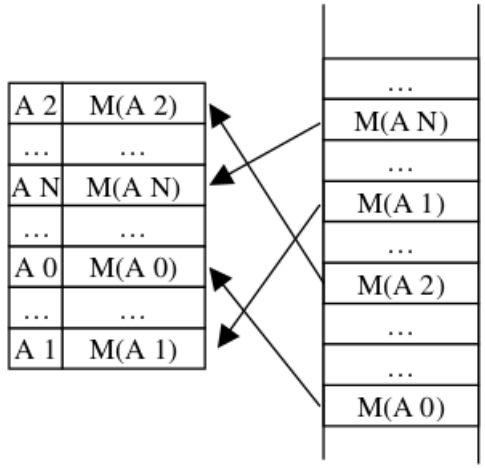
[1] (Prawie) dokładna kopia slajdów dr hab inż. J. Biernata

Bufory pamięci

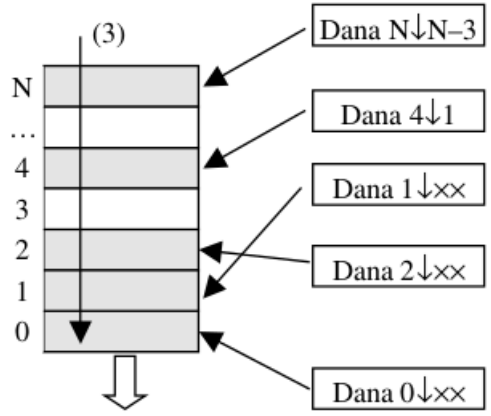
bufor zawiera **kopie** aktualnie przetwarzanych danych



bufor sekwencyjny – kolejka



bufor blokowo-skojarzeniowy



bufor skojarzeniowo-sekwencyjny

organizacja sekwencyjna – kolejka

- bufor rozkazów (*instruction queue*), bufor zapisów (*write buffer*)

organizacja blokowo-skojarzeniowa

- pamięć podręczna (*cache memory*) – zawiera **kopie** używanych danych

organizacja skojarzeniowo-sekwencyjna

- bufor przywracania kolejności (*reorder buffer*) (PISO *parallel-in serial-out*)

Zasada lokalności

W *komputerze z programem zintegrowanym* programy i dane mają tendencję do skupiania w wymiarze przestrzennym i czasowym

- **lokalność przestrzenna** (*spatial locality*)
  Jest prawdopodobne użycie informacji z sąsiednich lokacji pamięci
  - kody rozkazów – zależność *lokacyjna sekwencyjna* (licznik rozkazów)
  - struktury danych – skupione (zmienne robocze) lub regularne (tablice)
- **lokalność czasowa** (temporal locality)
  Kod użyty będzie zapewne ponownie użyty w nieodległym czasie
  - kody rozkazów – pętle programowe
  - struktury danych
    – zmienne robocze: ciągłe używanie
    – struktury regularne: wielokrotne użycie elementów

WNIOSEK: Utworzyć w pobliżu *procesora* **bufor** zawierający **kopie danych** z pamięci (głównej) aktualnie używanych i korzystać z kopii zamiast z oryginału

Zasady użycia buforów *cache*

- użycie bufora musi być przeźroczyste dla programu
  - niewidoczne na poziomie ISA (listy rozkazów),
  - realizowane na poziomie HSA (organizacji procesora/komputera)

!! sposób wykonania rozkazu nie może być zależny od obecności bufora, ale czas wykonania rozkazu w obecności bufora powinien być krótszy

!! użycie bufora pamięci podręcznej nie otwiera osobnej przestrzeni adresowej
$\longrightarrow$ nie są potrzebne specjalne rozkazy dostępu do bufora

- bufor pamięci podręcznej powinien
  - być wykonany w szybszej technologii (statycznej)
  - być umieszczony w bezpośrednim sąsiedztwie procesora

  $\longrightarrow$ bufor umożliwia wykorzystanie zalet szybkich pamięci statycznych i pojemnych lecz wolnych pamięci dynamicznych.

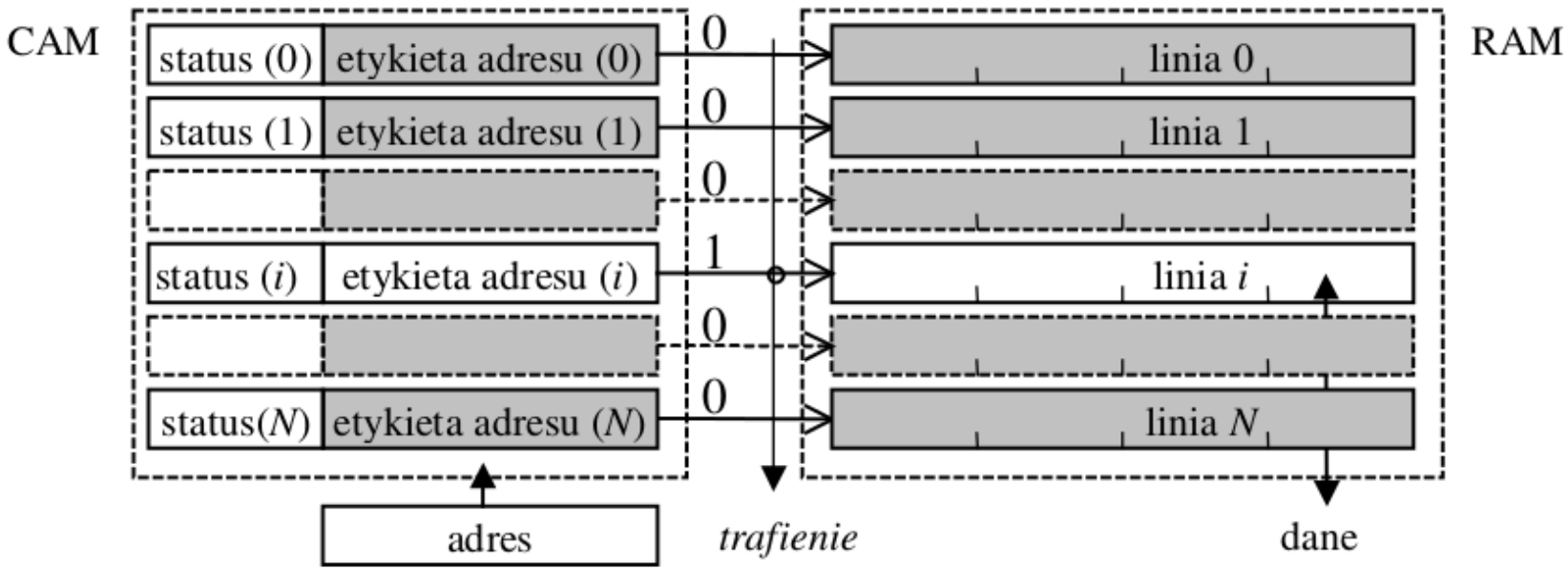
Skuteczność buforów cache

- Zasada lokalności *nie* gwarantuje obecności kopii danych w buforze cache
- Skuteczność użycia pamięci podręcznej zależy od *organizacji* (struktury) bufora
- Ilościowa ocena skuteczności:
  - współczynnik trafień (hit rate) h
  - współczynnik chybień (miss rate) $m = 1 - h$
  - t_{mp} – średnia strata czasu w razie chybienia (*miss penalty*)
- Średni czasu dostępu do pamięci
  - w pamięci jednopoziomowej
    $t_a = (1 - m)t_{ca} + m(t_{ram} + t_{mp})$
  - w pamięci dwupoziomowej (inkluzywnej – L1⊂L2)
    $t_a =$
    $(1 - m_1)t_{ca1} + (m_1 - m_2)(t_{ca2} + t_{mp1}) + m_2(t_{ram} + t_{mp1} + t_{mp2})$
    ($m_1 > m_2$, bo każde trafienie w L1 jest trafieniem w L2, a każde chybienie w L2 jest chybieniem w L1)

Organizacja pamięci podręcznej

Konieczna jest identyfikacja kopii w buforze cache

- jedynym identyfikatorem danej (także kodu rozkazu) jest adres
- kopiowanie pojedynczych bajtów/słów jest sprzeczne z zasadą lokalności
  – jedna lokacja (linia) w buforze zawiera wiele bajtów/słów
- jednorodna struktura ułatwia identyfikację, upraszcza i przyśpiesza dostęp
  – ustalony rozmiar pojedynczej lokacji (linii) w buforze



Schemat organizacji bufora pamięci podręcznej

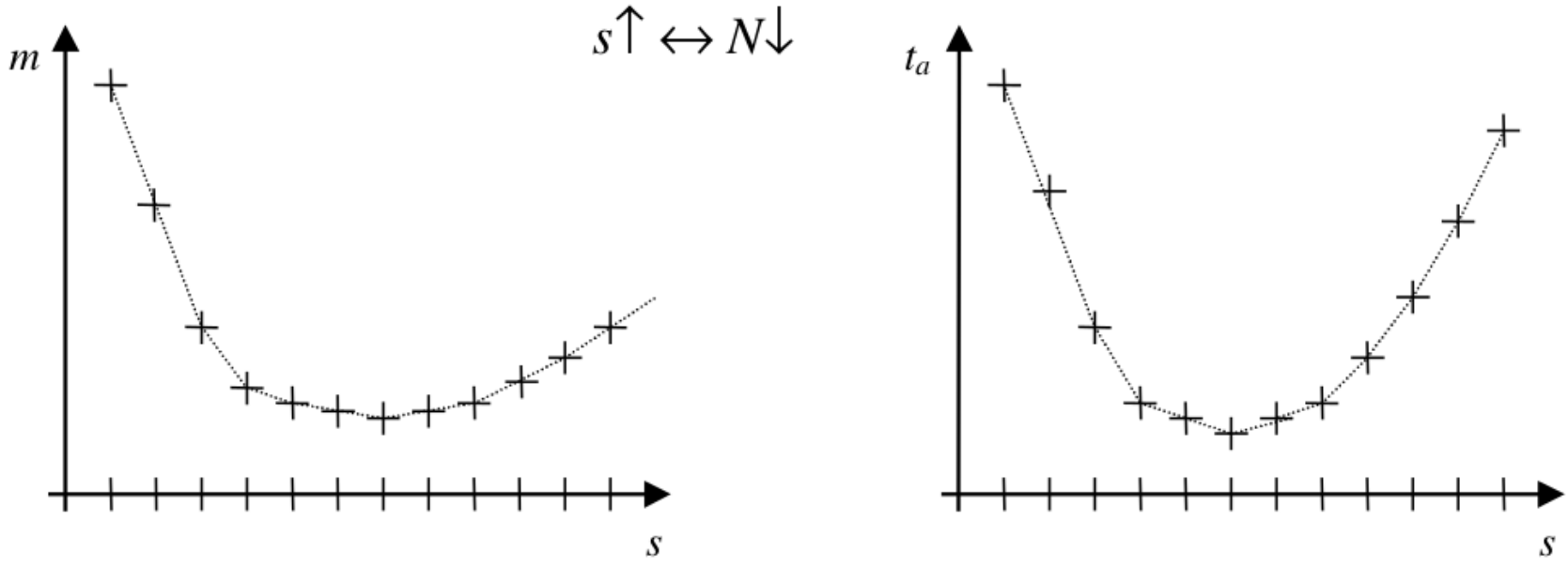
Organizacja odwzorowania danych w pamięci podręcznej

- linia – jednostka wymiany danych między buforem a pamięcią główną
- rozmiarem lokacji powinno być $s = 2^k$ bajtów/słów (linia wymiany)
  - przestrzeń adresowa procesora = suma rozłącznych linii
  - identyfikatorem lokacji w buforze może być skrócony adres
- czas wyszukiwania w buforze (zbiór nieuporządkowany) – $\lceil \log 2N \rceil$ (N – liczba linii) $\rightarrow$ optymalny rozmiar bufora $N = 2^n$ linii

Odwzorowanie ($i(B)$ – linia w buforze, $r(P)$ – linia w pamięci głównej)
BUFOR(linia $i(B)$) $\neq$ BUFOR(linia $i(B) \neq j(B)$)
PAMIĘĆ $\rightarrow$ BUFOR: identyfikator$\|$[linia $i(B)$] = adres(linia $r(P)$)$\|$[linia $r(P)$]
Jeśli $s = 2^k$ bajtów, to adres linii = bity adresu bez k najniższych

Charakterystyki skuteczności

- większa liczba linii (N) – lepsza lokalność czasowa
- większy rozmiar linii (s) – lepsza lokalność przestrzenna
- dla ustalonej pojemności, ze zwiększaniem rozmiaru linii, dominujący staje się wpływ zanikania lokalności czasowej i następuje wzrost strat czasu w razie chybienia (czas wymiany jest proporcjonalny do liczby transferów)



Zależność współczynnika chybień m i średniego czasu dostępu do pamięci t_a od rozmiaru linii s dla ustalonej pojemności $C = s \cdot N$ bufora cache

Pamięć podręczna wielopoziomowa

zasada lokalności $\rightarrow$ ma sens buforowanie bufora lub jego rozdzielenie

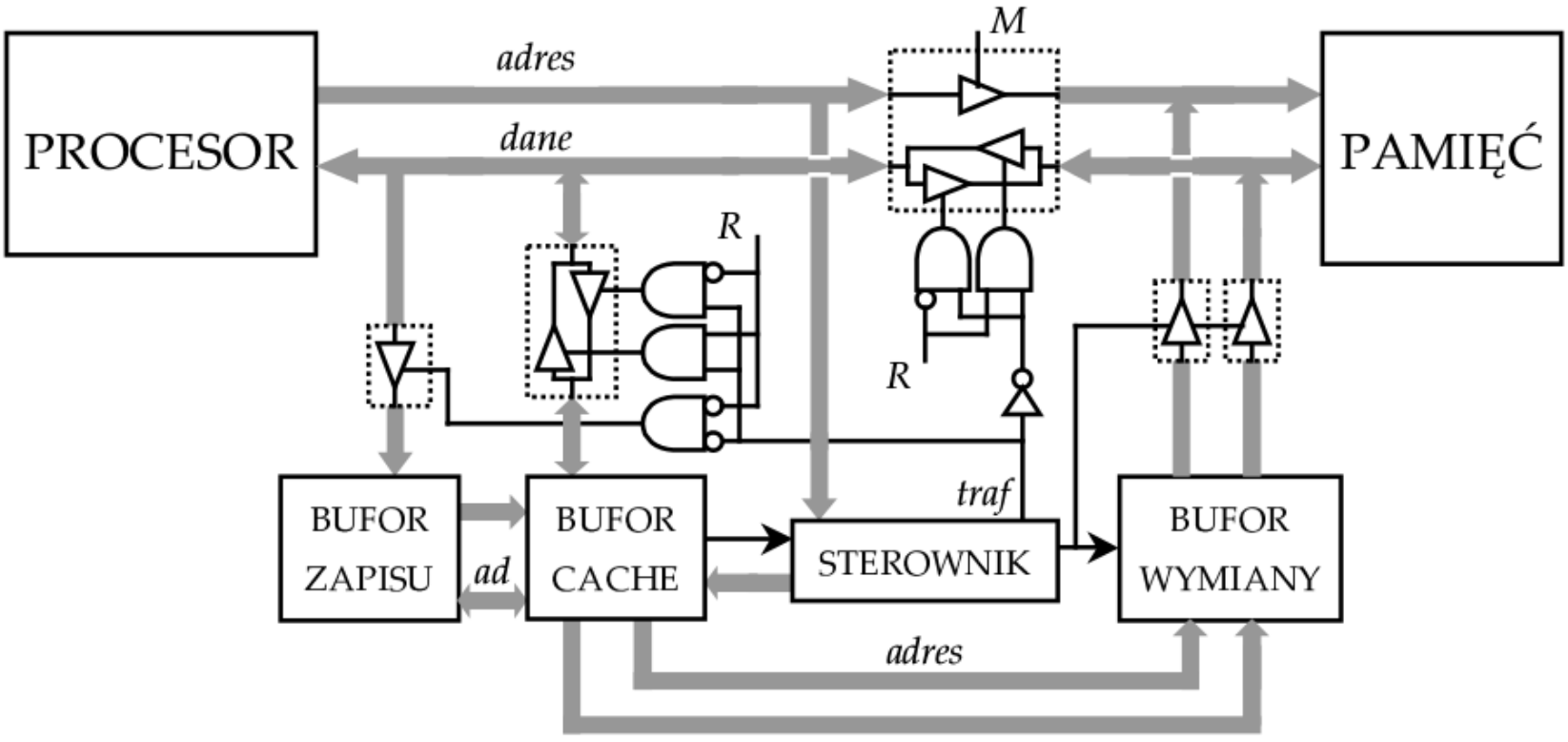
$\rightarrow$ pamięć podręczna wielopoziomowa

- organizacja hierarchiczna – pamięć inkluzywna
  - lokalizacja danej umieszczonej w pamięci $L(i)$ dostępna w $L(i+1)$
  - bufor $L(i)$ poziomu i zawiera kopie niektórych danych z bufora $L(i+1)$ poziomu $i+1$, niektóre linie w $L(i)$ mogą być zaktualizowane
  - dana aktualna dostępna *zawsze* na najniższym poziomie
- *organizacja separowana* – pamięć *nieinkluzywna* (ATHLON – victim cache)
  - żadna lokalizacja danej w buforze $L(i)$ nie jest dostępna w $L(i+1)$
  - dostęp do bufora $L(i)$ znacznie krótszy niż do $L(i+1)$
  - w buforze $L(i+1)$ są dane usunięte z $L(i)$

problem spójności

- zgodność wszystkich kopii informacji, albo
- znajomość (świadomość) lokalizacji informacji aktualnej

Sterowanie współpracą bufora cache z procesorem i pamięcią

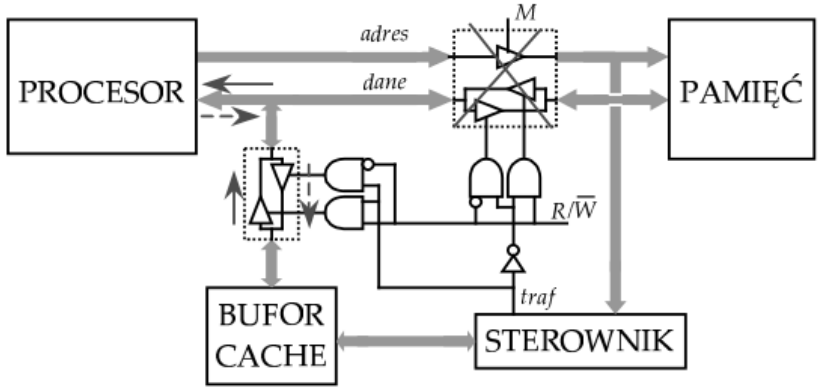


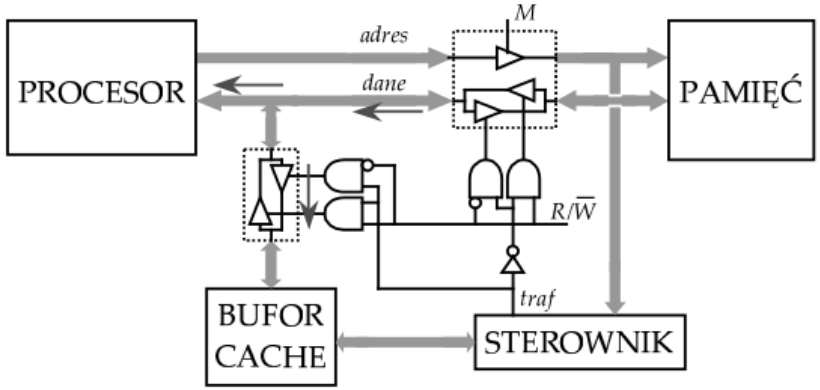
Sterowanie współpracą pamięci podręcznej i pamięci głównej (*traf* – sygnał trafienia w buforze cache, R – odczyt/zapis (1/0), M – transfer/blokada adresu (1/0))

Współdziałanie pamięci podręcznej z procesorem

Schemat uproszczony

trafienie/chybienie

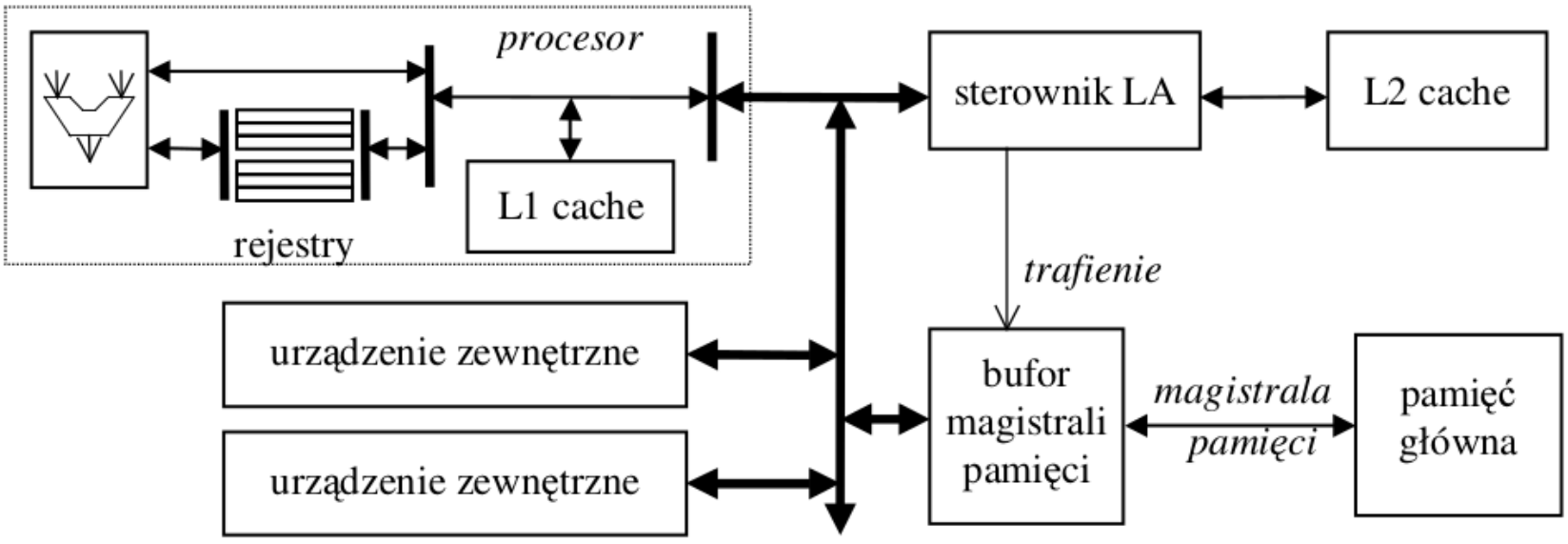




etap (1) – odczyt z pamięci (. . . blokowy) (. . . → usunięcie linii → wypełnienie)

etap (2) – kopiowanie odczytu

Architektura systemu pamięci podręcznej (1)

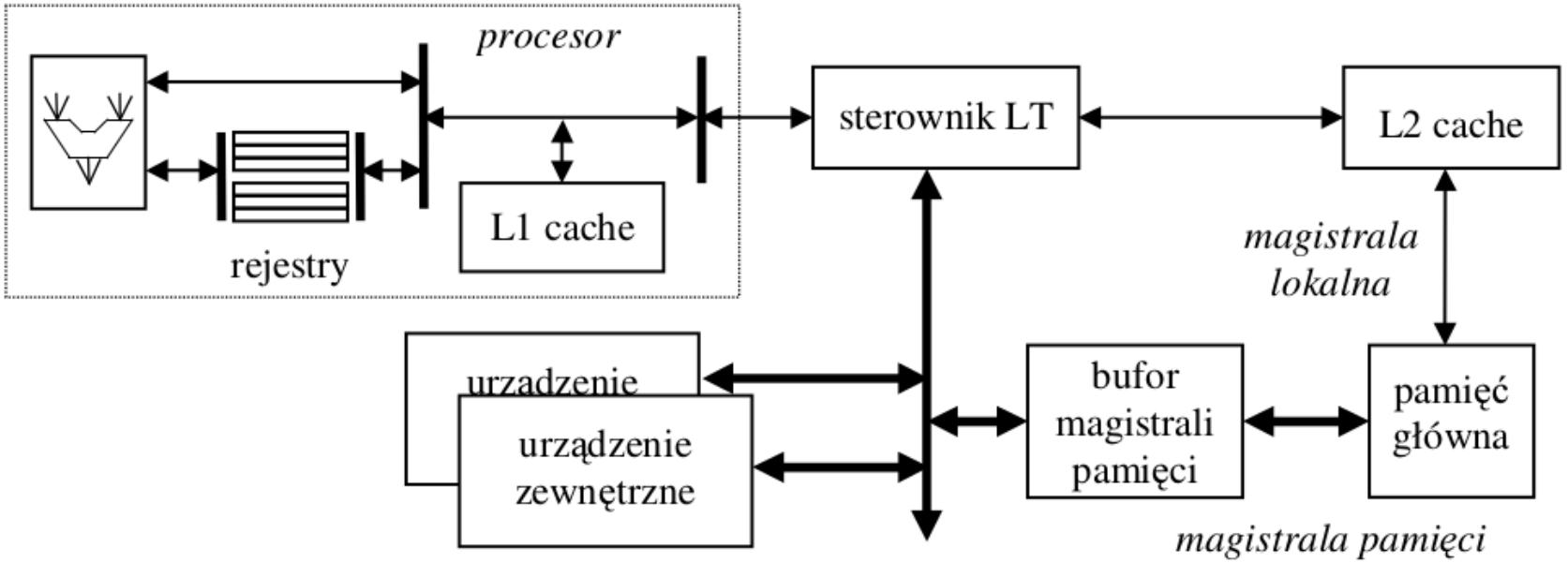


przesyłanie danych między procesorem a pamięcią podręczną

- przez magistralę systemową – sterownik typu look-aside (LA)
- przez separowaną magistralę lokalną – sterownik typu look-through (LT)

Architektura pamięci podręcznej typu look-aside

Architektura systemu pamięci podręcznej (2)



Architektura pamięci podręcznej typu look-through

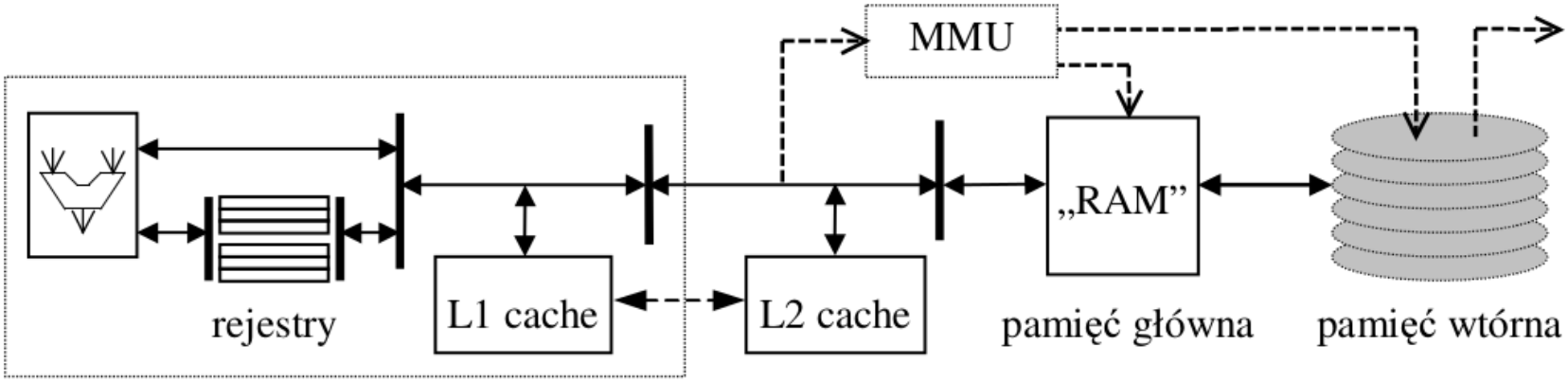
sterownik *podglądający* (look-aside)

- wada – blokowanie magistrali systemowej podczas transferów
- zaleta – prostota konstrukcji

sterownik *pośredniczący* (*look-through*)

- wada – skomplikowana struktura
- zaleta – odciążenie magistrali systemowej (transfery magistralą lokalną)

Pamięć wielopoziomowa



t_a	0,25–1ns	0,5–2ns	1–5ns	10–50ns	1–10ms
rozmiar	128B–1kB	32–128kB	0,25-1MB	256MB – 4GB	1*TB*
	register	*cache*	*cache*	*memory*	*storage*

Hierarchia pamięci wielopoziomowej

- niektóre przestrzenie adresowe lub ich części nie mogą być buforowane
- procesor adresuje pamięć główną – sterownik bufora przechwytuje transfery
- specjalne rozkazy – sposób działania wewnętrznego bufora cache

Parametry bufora cache

bufor poziomu L1

- rozmiar linii – dostosowany do rozmiaru magistrali danych procesora (systemu) i możliwości transferów blokowych
- rozmiar pamięci – wystarczający do pomieszczenia kilku (2^N) stron lub innych jednostek przydziału (bloków) pamięci

bufor poziomu L2

- rozmiar linii – identyczny jak w L1
- rozmiar bufora – wystarczający do pomieszczenia zbioru roboczego procesu

Typowe parametry

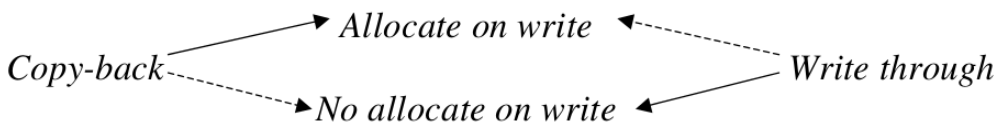
- szerokość magistrali danych – 2^k bajtów,
- transfer blokowy – nałożenie 2^m transferów całą szerokością magistrali
  - linia powinna zawierać $2^{k+m} = 2^B$ bajtów
- rozmiar strony – 2^P bajtów
  - rozmiar bufora L1 – 2^N stron $= 2^{P+N}$ bajtów
  - łączna liczba linii w buforze L1 $= 2^{P+N-B}$
  - rozmiar bufora L2 – 2^W 2^N stron

Aktualizacja zawartości bufora cache

Warunkiem utrzymania spójności jest aktualizacja linii, która obejmuje

- unieważnienie linii zawierającej dane przypadkowe lub nieaktualne
- wypełnienie linii (bloku) nową zawartością
- wymiana linii w razie braku wolnego miejsca w buforze
- zapis w obszarze skopiowanej linii
  - zapis skrośny (write through) – zgodność wszystkich kopii informacji
  - zapis lokalny (copy-back) – znajomość lokalizacji informacji aktualnej

Zapis może być poprzedzony kopiowaniem nieobecnej linii (allocate on write)



Obciążenie magistrali pamięci transferami powinno być minimalne

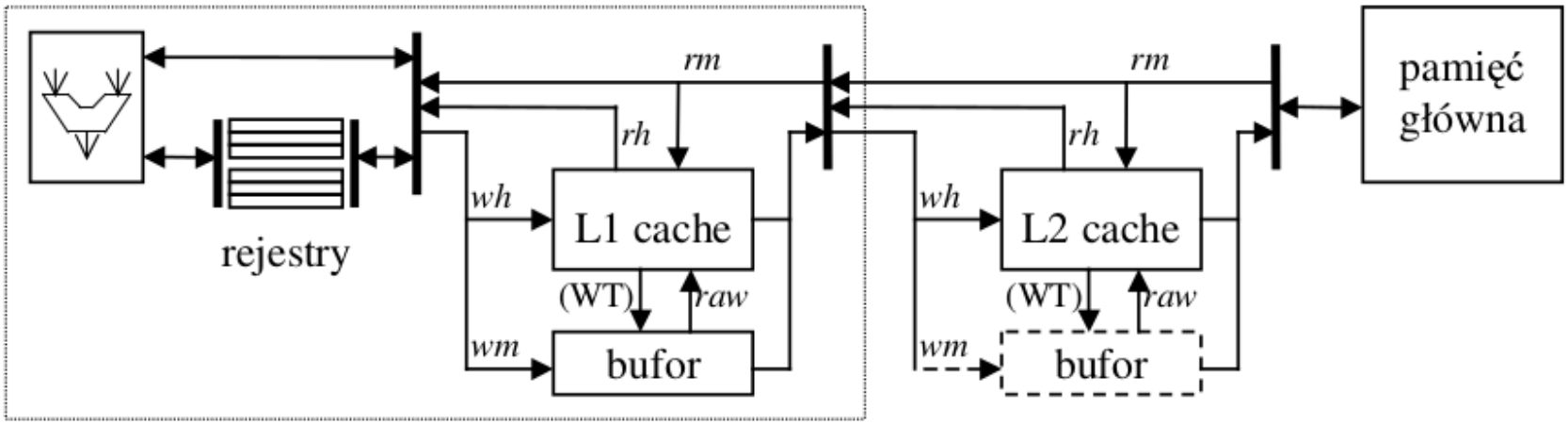
$$\text{memory traffic ratio} = \frac{\text{transfery(memory + cache)}}{\text{transfery(cache)}}$$

Bufory zapisu (*write buffer*)

Chybienie podczas próby zapisu

- w trybie WT (*no allocate-on-write*) wymaga transferu do poziomu wyższego
- w trybie CB uprzedzające kopiowanie (*allocate-on-write*) wymaga blokady zapisu oraz powoduje niezgodność kopii (zapis lokalny) po odblokowaniu
- w trybie CB kopiowanie odłożone (no allocate-on-write) do chybienia w razie odczytu *lub kolejnego zapisu* nie powoduje strat jeśli zapis jest buforowany

Bufor zapisu: pamięć FIFO (kolejka – *musi być zachowana kolejność zapisów*)
$\longrightarrow$ nasycenie bufora (*write buffer saturation*) $\longrightarrow$ blokada kolejnego zapisu



Bufory zapisu i ścieżki przepływu danych w systemie pamięci

Obsługa pamięci podręcznej

- unieważnianie linii (line invalidation)
  - przed pierwszym wypełnieniem
  - wskutek zewnętrznej zmiany oryginału danych w pamięci głównej
  - przełączanie procesów – unieważnienie wszystkich linii (*line flush*)
- wypełnianie linii (line fill) oraz wymiana linii (line exchange)
  - chybienie podczas odczytu (miss on read) lub (w trybie AOW) zapisu
- odczyt danej (read)
  - trafienie podczas odczytu (hit on read)
- zapis danej (write) $\longrightarrow$ rozbieżność kopii z oryginałem
  - trafienie podczas zapisu (hit on write)
    - zapis skrośny (jednoczesny) (write through, WT) – modyfikuje kopię w buforze wyższego poziomu
    - zapis lokalny (zwrotny) (write/copy back, WB/CB) – w kopii lokalnej, opóźniony zapis do bufora wyższego poziomu podczas usuwania linii
    - unieważnienie linii trafionej i zapis bezpośredni (omijający) do pamięci głównej (write aside) lub bufora poziomu wyższego
  - chybienie podczas zapisu (miss on write) – zapis do pamięci głównej lub bufora poziomu wyższego (NAOW) lub wypełnienie i zapis (AOW)

Zarządzanie pamięcią podręczną

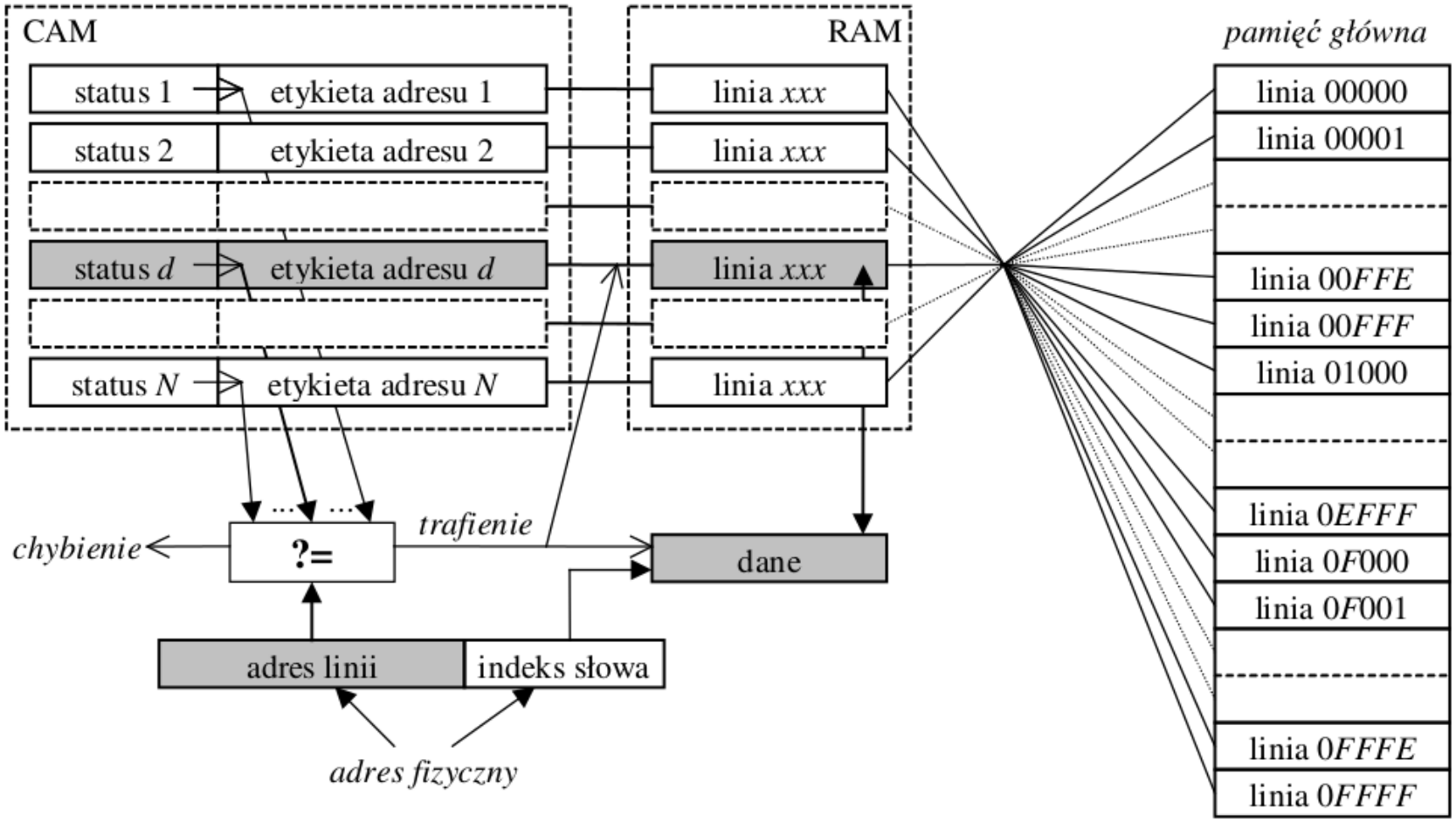
Sposoby umieszczania i aktualizacji danych w buforze:

- strategie wypełniania pamięci podręcznej – odwzorowanie pamięci głównej w liniach pamięci podręcznej
  - losowe – zbiór linii w zbiór lokacji linii (dowolnie)
  - bezpośrednie – podzbiór linii w linię lokacji
  - blokowo-skojarzeniowe – podzbiór linii w podzbiór lokacji linii
- strategie wymiany kopii w pamięci podręcznej w celu aktualizacji bufora (bezzasadne w odwzorowaniu bezpośrednim!)
  - losowa (random) – dowolnie (skuteczna przy odwzorowaniu losowym)
  - kolejkowa (FIFO) – kolejność wymiany linii jest zgodna z kolejnością ich wypełniania (usuwana jest linia najdawniej alokowana)
  - LRU (last recently used) – wymieniana jest linia najdawniej użyta (skuteczna przy odwzorowaniu blokowo-skojarzeniowym).
- strategie pobierania linii z pamięci głównej
  - pobranie wymuszone (demand fetching) – uaktywniane chybieniem
  - pobranie uprzedzające (prefetching) – na podstawie prognozy dostępu.

Odwzorowanie linii w buforze pamięci podręcznej

- całkowicie skojarzeniowe (fully associative) – każda linia pamięci głównej może być skopiowana w dowolnej lokacji linii pamięci podręcznej
  - wymiana linii konieczna wtedy, gdy wszystkie linie są użyte
  - największy współczynnik trafień, brak migotania (thrashing)
- bezpośrednie (direct mapped) – rozłącznym podzbiorom linii pamięci głównej przypisano unikatowe lokacje linii pamięci w podręcznej
  - najkrótszy czas kojarzenia – rekord indeksujący adresu (cache index)
  - najmniejszy współczynnik trafień
  - chybienia wskutek konfliktu odwzorowania (conflict miss) – migotanie
- wielodrożne (set-associative) – rozłącznym podzbiorom linii pamięci głównej przypisano rozłączne podzbiory lokacji linii w pamięci podręcznej (bezpośrednie odwzorowanie bloków, pełne skojarzenie w podzbiorze)
  - czas dostępu dłuższy niż dla pamięci z odwzorowaniem bezpośrednim
  - niewielkie migotanie, duży współczynnik trafień
  - konflikt odwzorowania maleje ze wzrostem liczby lokacji w bloku

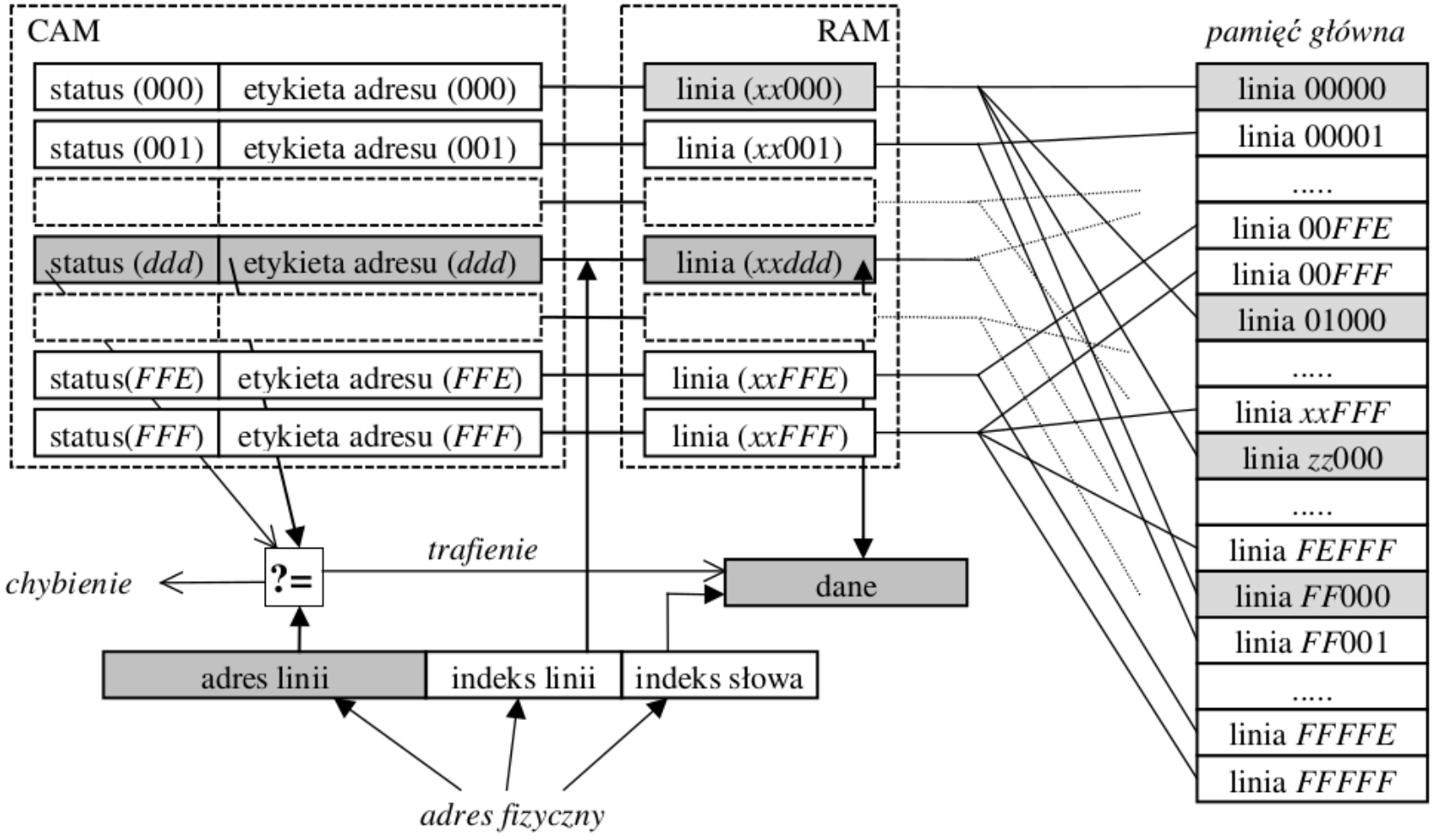
Odwzorowanie całkowicie skojarzeniowe



Bufor całkowicie asocjacyjny (fully associative)

Odwzorowanie bezpośrednie

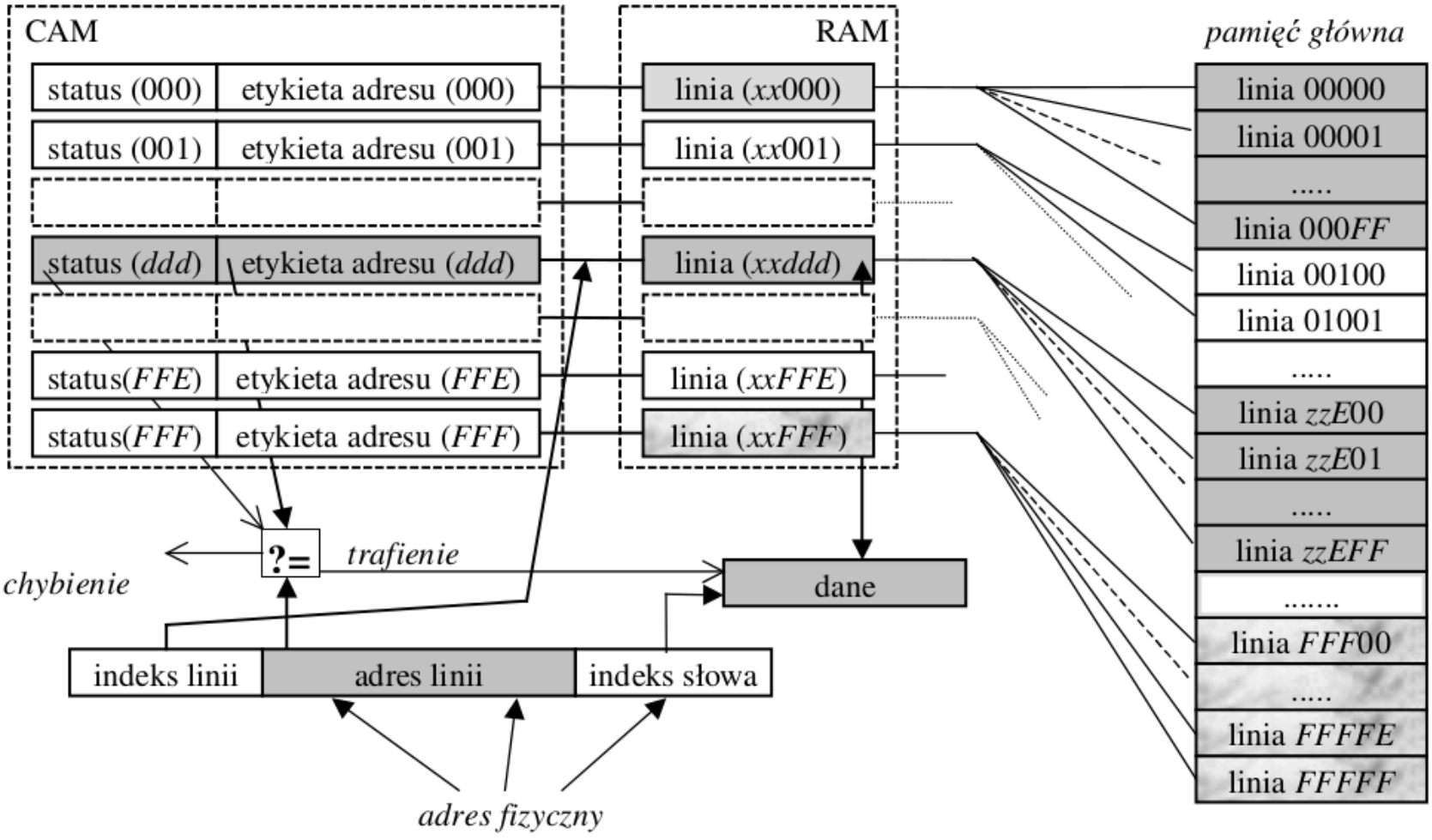
z przeplotem



Bufor z odwzorowaniem bezpośrednim (direct mapped)

Odwzorowanie bezpośrednie

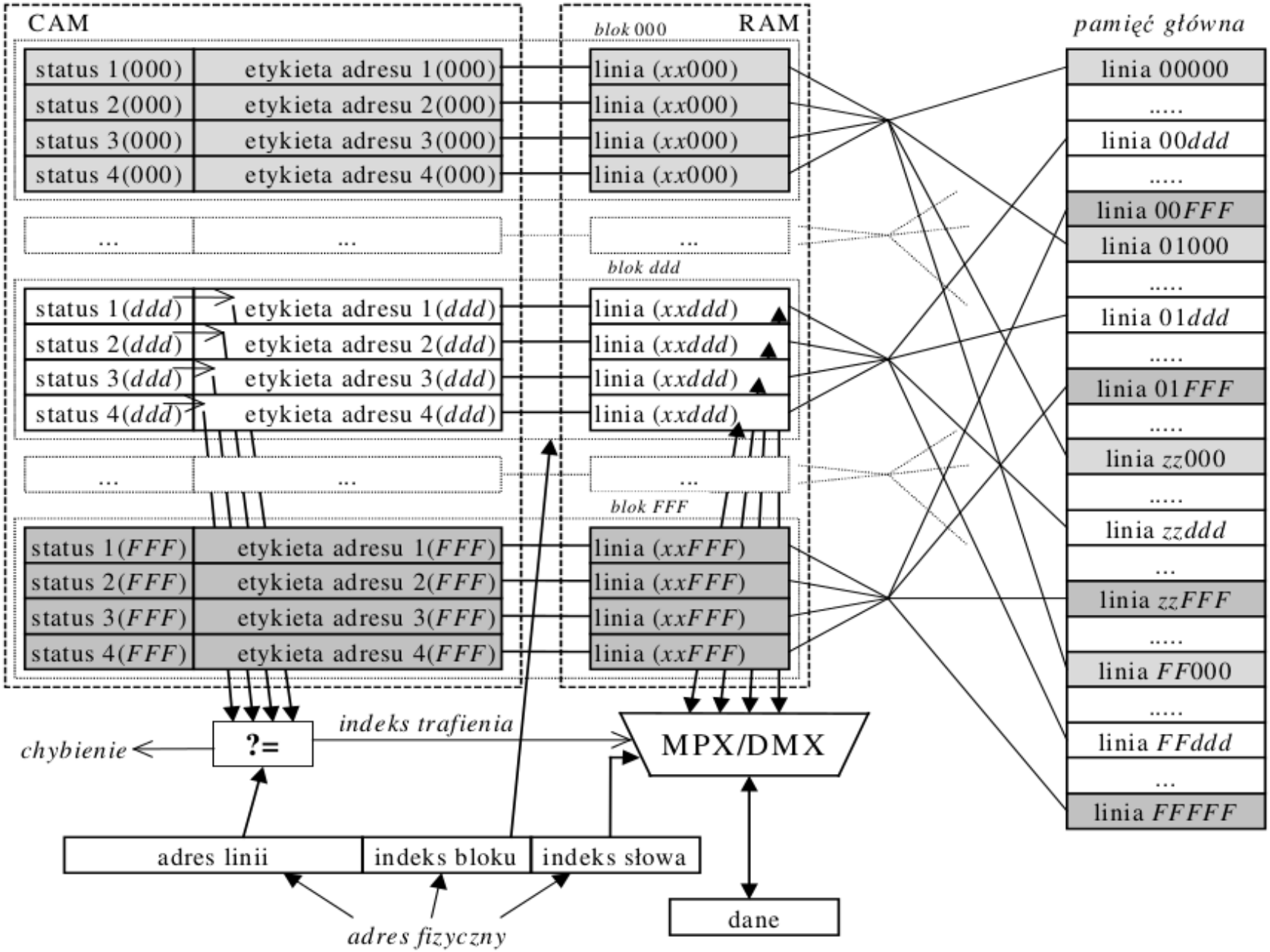
bez przeplotu



Odwzorowanie bezpośrednie bez przeplotu – b.częsty konflikt odwzorowania

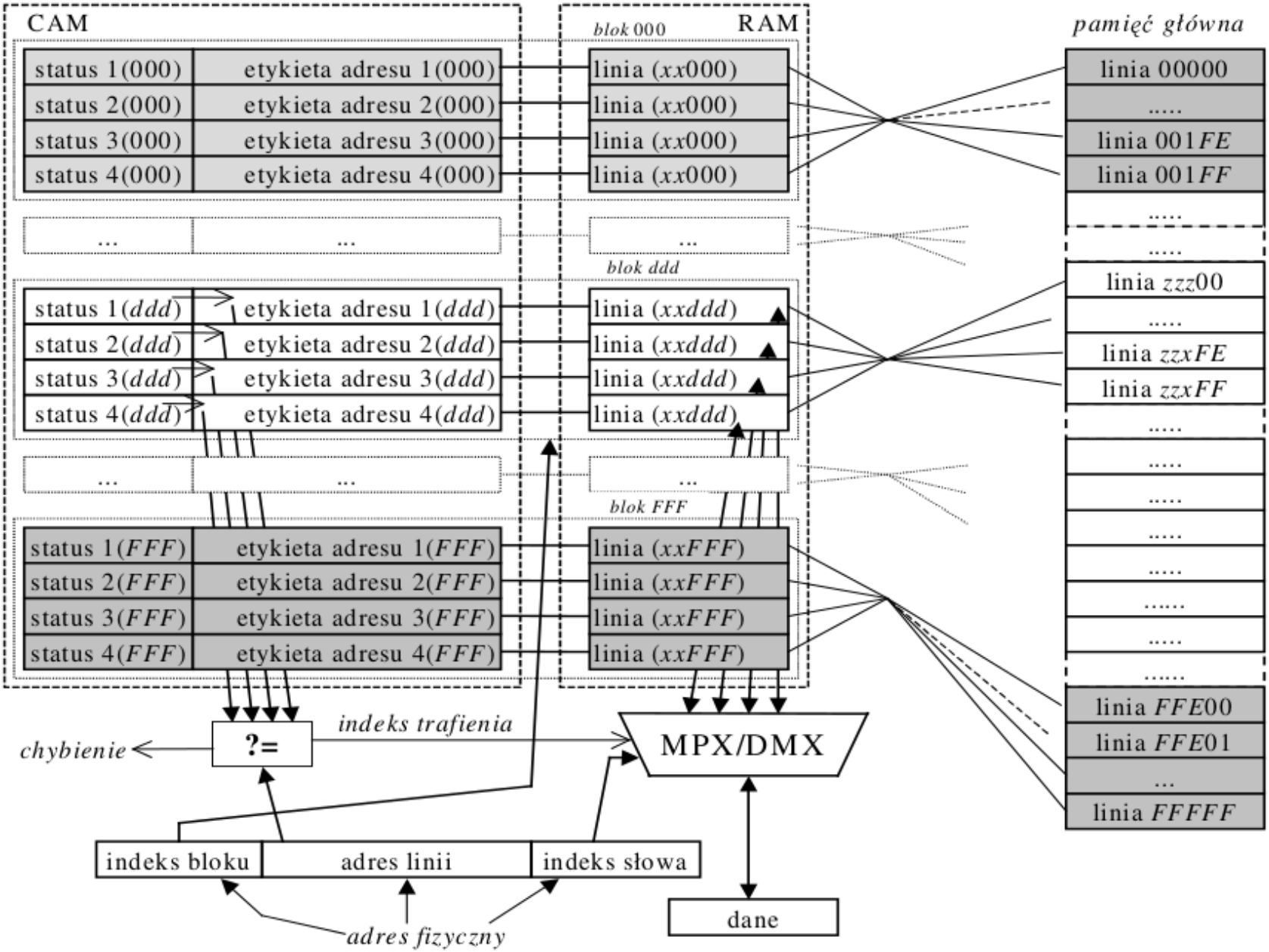
Odwzorowanie blokowo-skojarzeniowe (wielodrożne)

z przeplotem

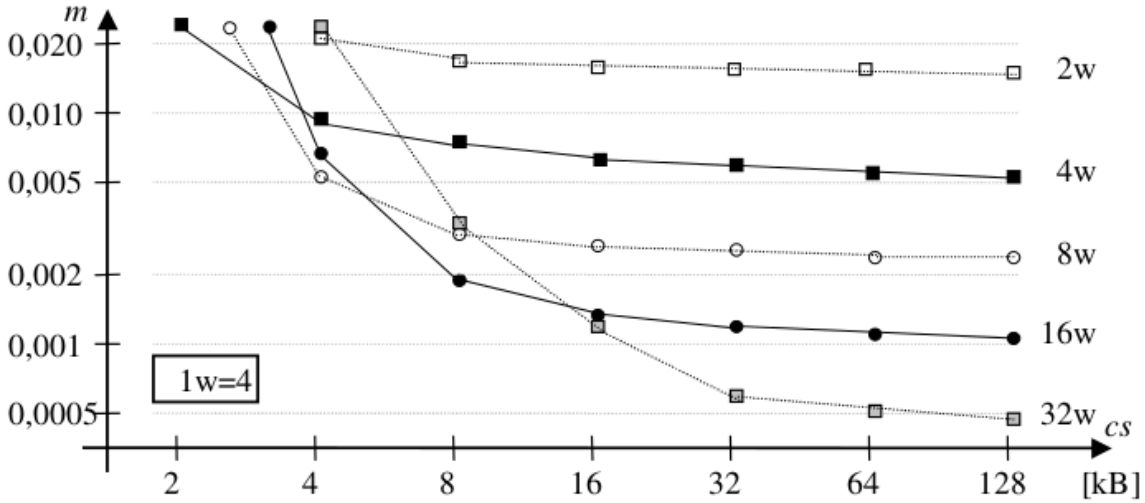


Odwzorowanie blokowo-skojarzeniowe (wielodrożne)

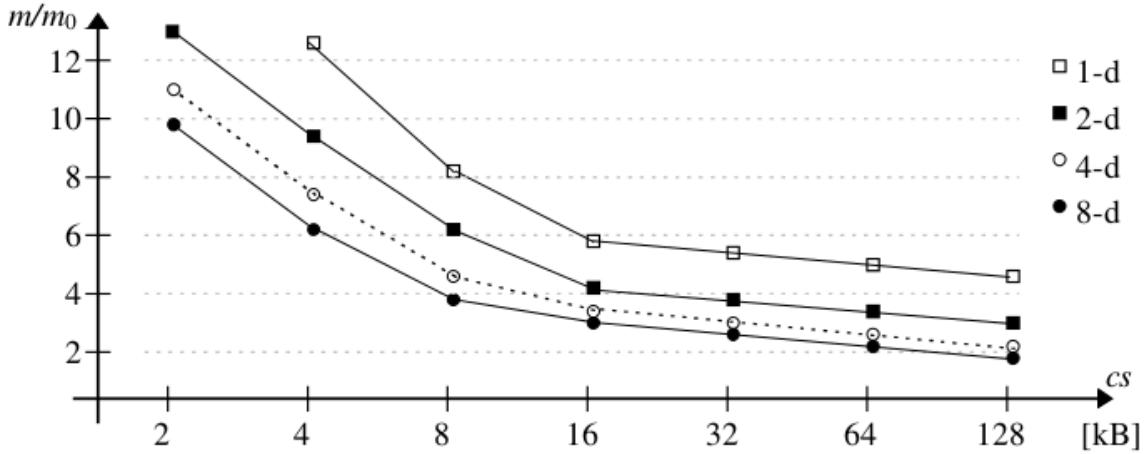
bez przeplotu



Organizacja pamięci podręcznej a charakterystyki skuteczności



Zależność współczynnika chybień od rozmiaru linii (pamięć dwudrożna)



Wpływ organizacji pamięci na częstość konfliktów odwzorowania

Strategie wymiany linii

Aktualizacja bufora podczas przełączania zadań

- bufor „ciepły" (warm cache) – część bufora nie jest wymieniana
- bufor „zimny" (cold cache) – unieważnianie całego bufora

Algorytmy wymiany

- losowy – tylko w buforze całkowicie asocjacyjnym
  - ryzyko usunięcia potrzebnej linii – $2 - K$ (K – liczba lokacji w buforze)
- kolejkowy (FIFO)
  - liczba bitów historii – $\log_2 S$ (S – liczba lokacji w podzbiorze)
  - ryzyko usunięcia potrzebnej linii – $p2^{-S}$
- wg używalności (LRU)
  - liczba bitów historii – $(S-1)\log_2 S$ (S – liczba lokacji w podzbiorze)
  - ryzyko usunięcia potrzebnej linii – $p2^{-S}$

Strategie pobierania linii

- pobranie wymuszone (demand fetching) – uaktywniane chybieniem
- pobranie uprzedzające (prefetching) – na podstawie prognozy dostępu, nie powinno powodować opóźniania pobrań wymuszonych (rozmiar linii wpływa na szybkość wypełniania)

Antycypacja jednostopniowa (one block lookahead, OBL)

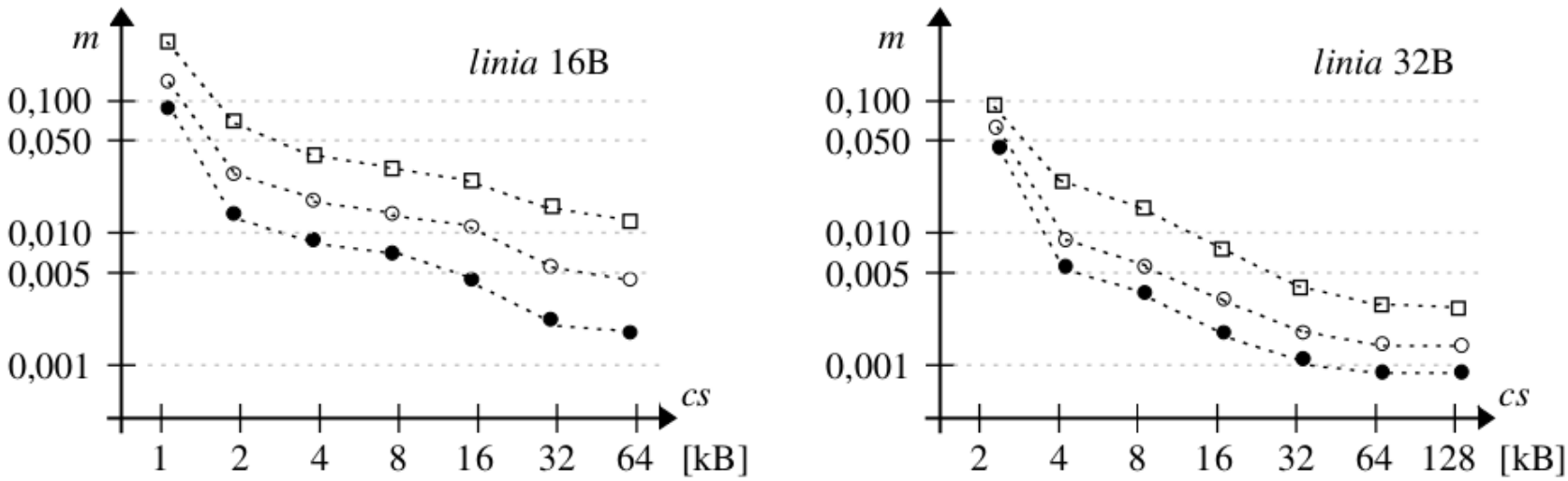
- w chwili pobrania linii i-tej z pamięci głównej należy też pobrać linię $i + 1$

Pobranie uprzedzające (antycypowane)

- automatyczne (prefetch always), inicjowane podczas każdej próby dostępu
  - przy okazji zwrotu do linii i jest zawsze pobierana linia $i + 1$
- implikowane w razie chybienia (prefetch on a miss)
  - wraz z wymuszonym wskutek chybienia pobraniem linii i zawsze jest pobierana linia $i + 1$
- markowane (tagged prefetch)
  - podczas pierwszego, markowanego (tagged) zwrotu do linii i-tej, wraz z linią i jest pobierana linia $i + 1$

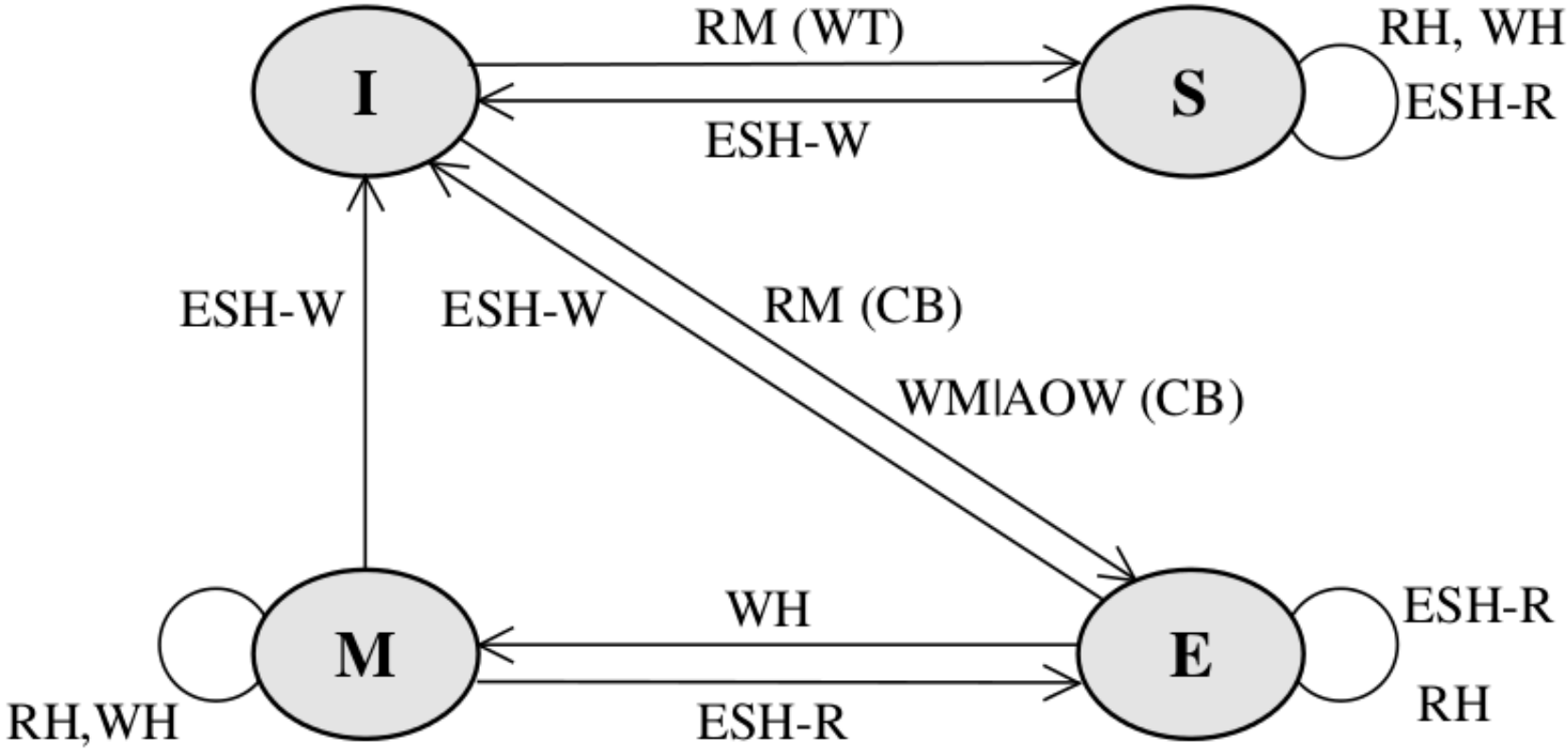
Intensywność pobrań uprzedzających

- automatyczne (prefetch always)
  - obciążenie magistral o 20–80% większe niż w pobraniach wymuszonych
- implikowane w razie chybienia (prefetch on a miss)
  - obciążenie magistral nieznacznie większe niż w pobraniach wymuszonych
- markowane (tagged prefetch)
  - obciążenie magistral podobne jak w pobraniach wymuszonych



Współczynnik chybień m w funkcji rozmiaru pamięci podręcznej *cs* dla strategii pobrań (□ – wymuszone, o – implikowane, · – markowane, automat.)

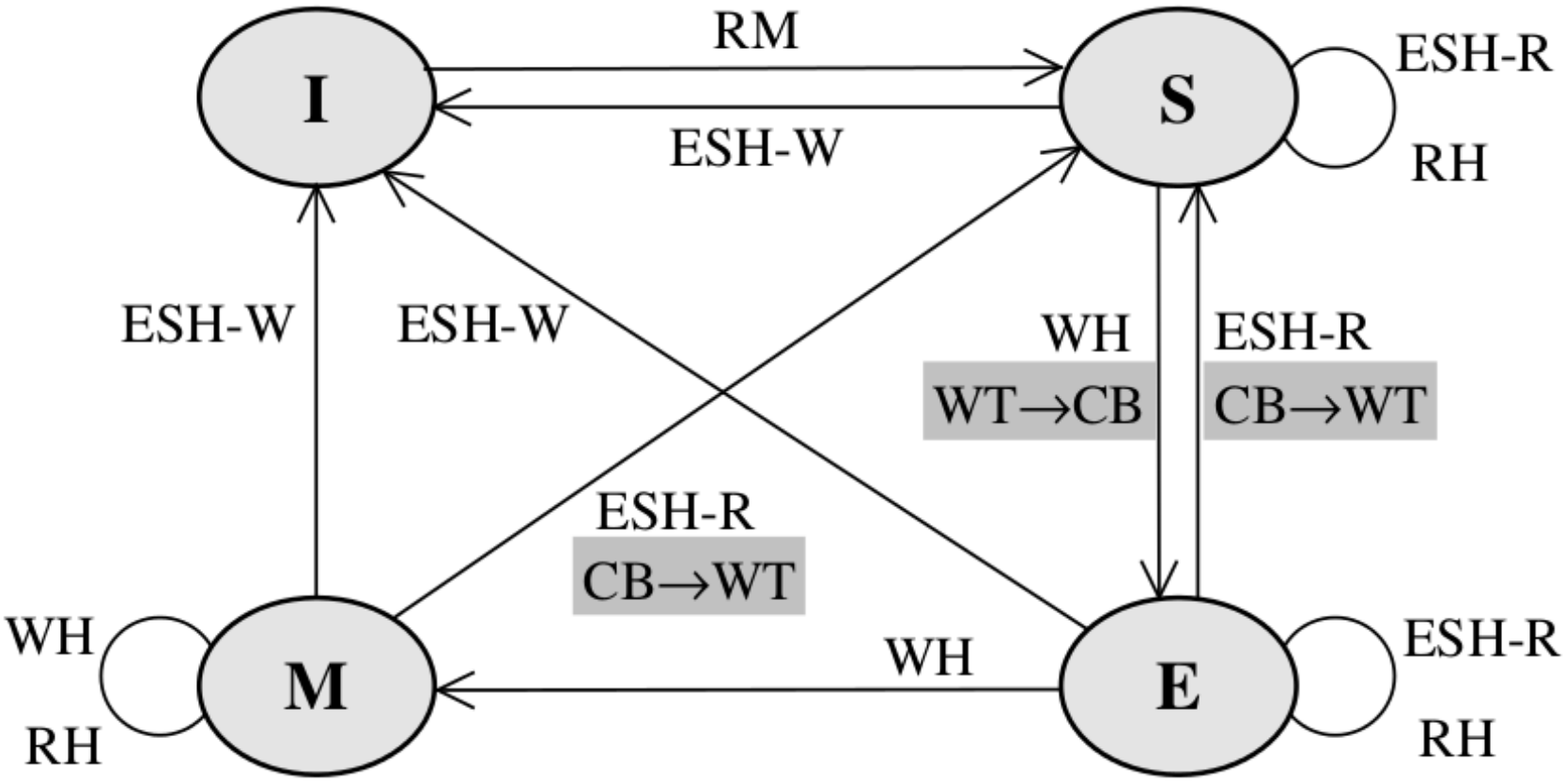
Zintegrowany model spójności pamięci podręcznej



Stany linii pamięci podręcznej obsługiwanej w trybie WT lub CB (I – nieważny (invalid), E – zgodny (exclusive), M – zmieniony (modified), S – współdzielony (shared))
M – chybienie (miss), H – trafienie (hit), R – podczas odczytu, W – podczas zapisu, ESH – podglądnięcie trafienia zewnętrznego (external snoop hit)

Strategia zapisu jednorazowego (write-once)

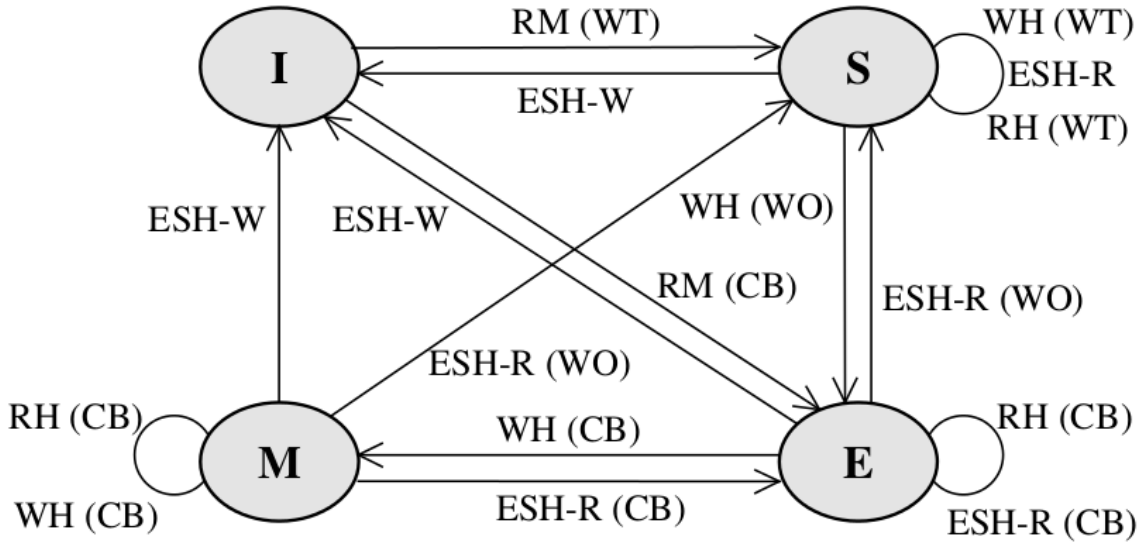
(Pentium) niewiele rejestrów → dużo zmiennych roboczych w pamięci → pierwszy zapis skrośny (WT), kolejne zapisy lokalne (CB)



Stany linii pamięci podręcznej obsługiwanej w trybie WO L1: WT→CB, L2: CB – krótki czas zapisu, częściowa zgodność danych

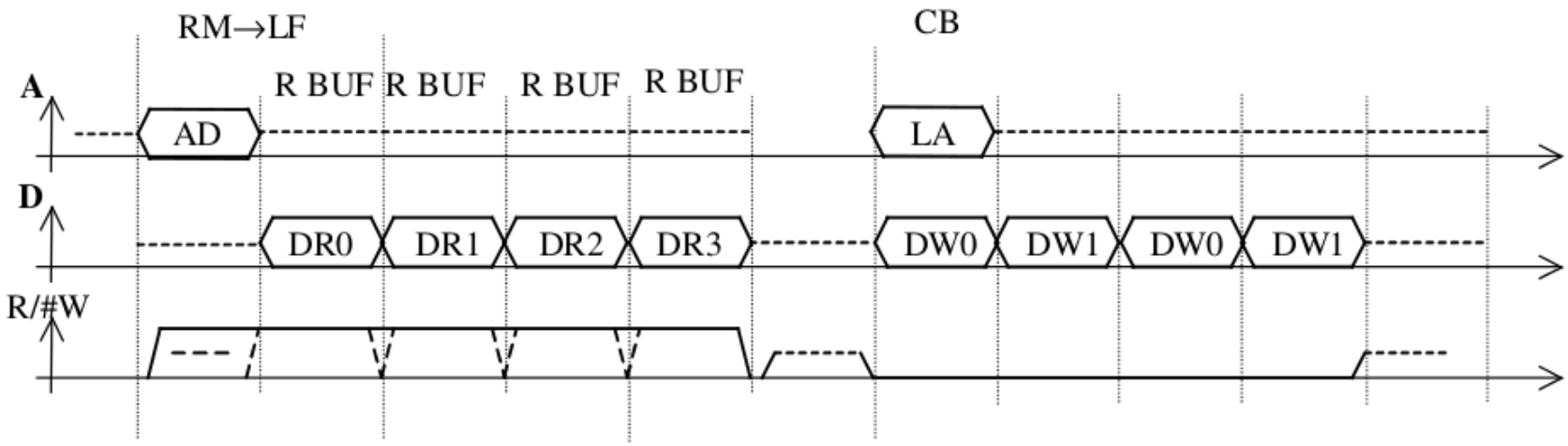
Pełny model spójności (MESI)

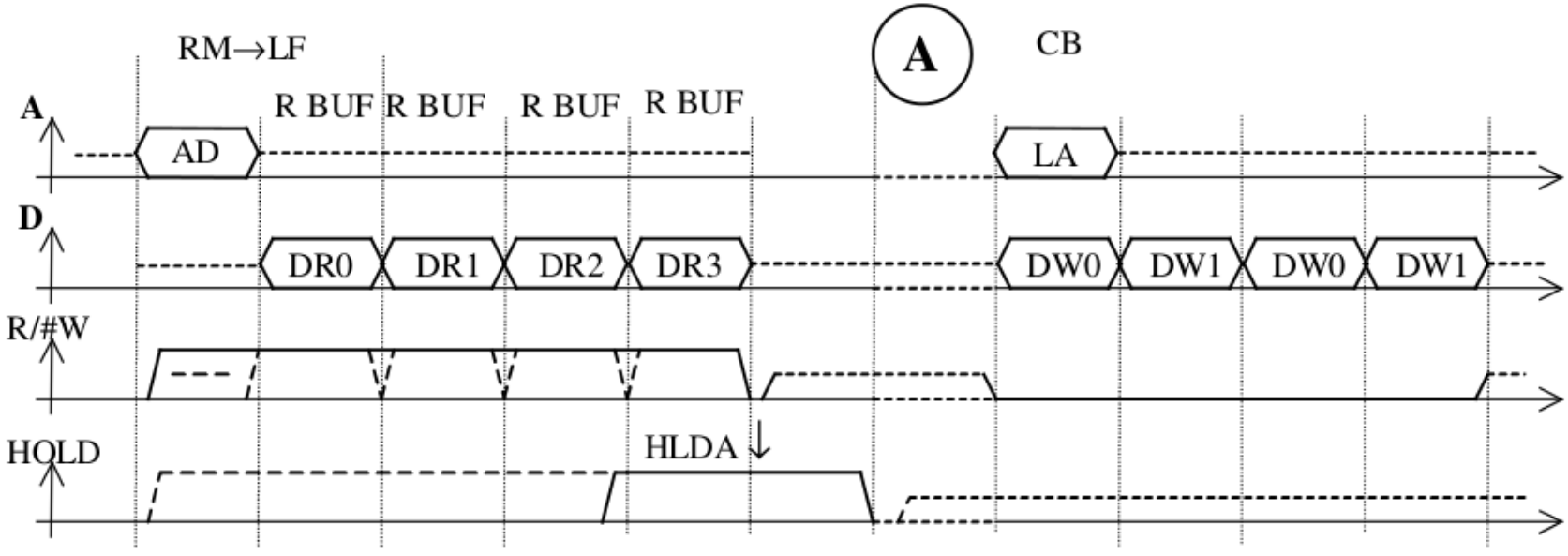
Model spójności wielopoziomowej pamięci podręcznej (protokół MESI)



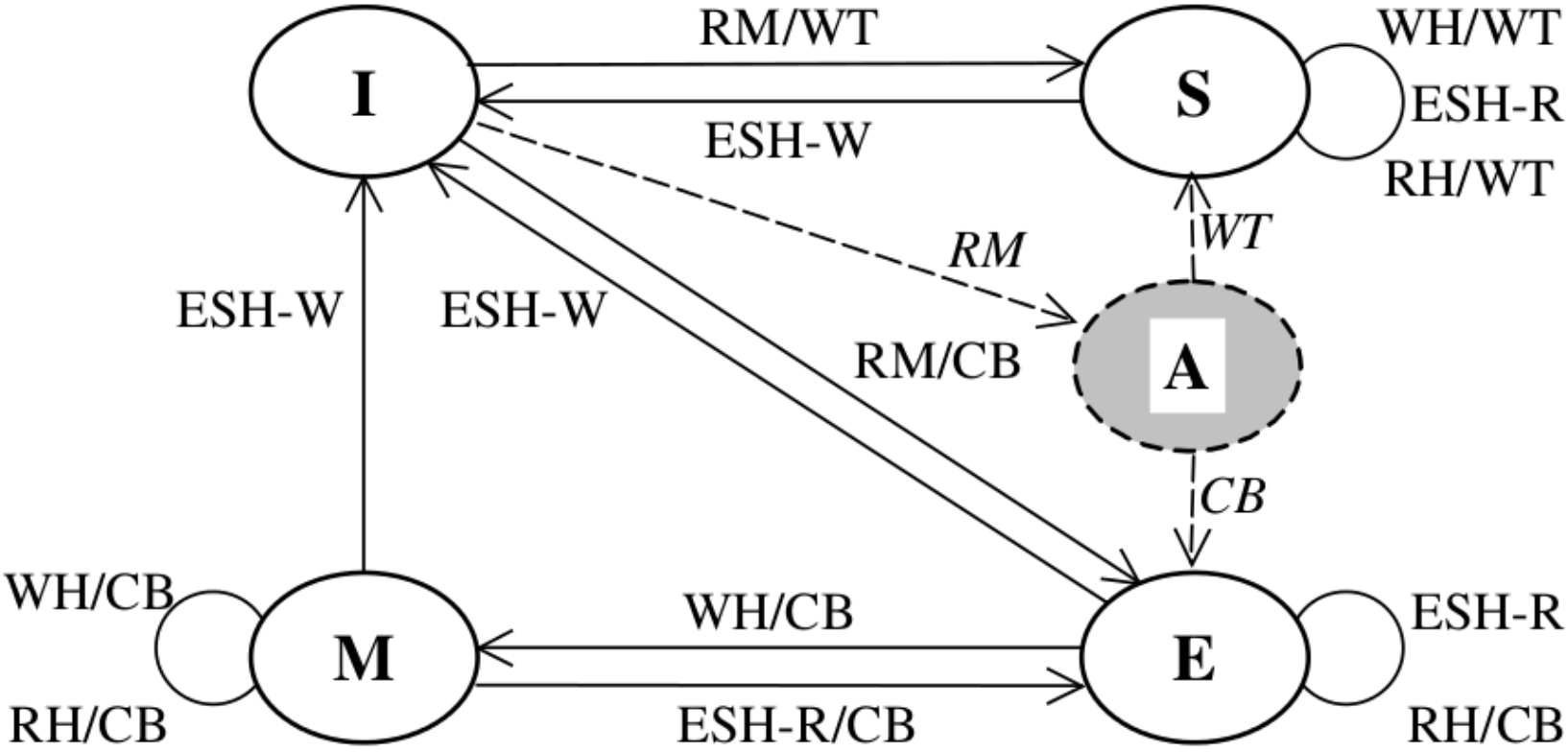
Obsługa pamięci dwupoziomowej:
L1: WT, L2: WT – długi czas zapisu, całkowita zgodność danych
L1: CB, L2: CB – najkrótszy czas zapisu, utrata zgodności danych po zapisie
L1: WT, L2: CB – krótszy czas zapisu, zgodność danych L1-L2
L1: CB, L2: WT – utrata zgodności danych po zapisie, długi czas wymiany

Wymiana linii – aspekty czasowe





Rozszerzony model spójności



(I – nieważny (invalid), E – zgodny (exclusive), M – zmieniony (modified), A – przydzielony (allocated), S – współdzielony (shared))
RH / WH – trafienie podczas odczytu (R) / zapisu (W), RM – chybienie podczas odczytu (read miss), ESH – podgl ą dni ę cie trafienia zewn ę trznego (external snoop hit),

Spójność pamięci w systemie wieloprocesorowym

Spójność

- zgodność wszystkich kopii informacji, albo
- znajomość (świadomość) lokalizacji informacji aktualnej

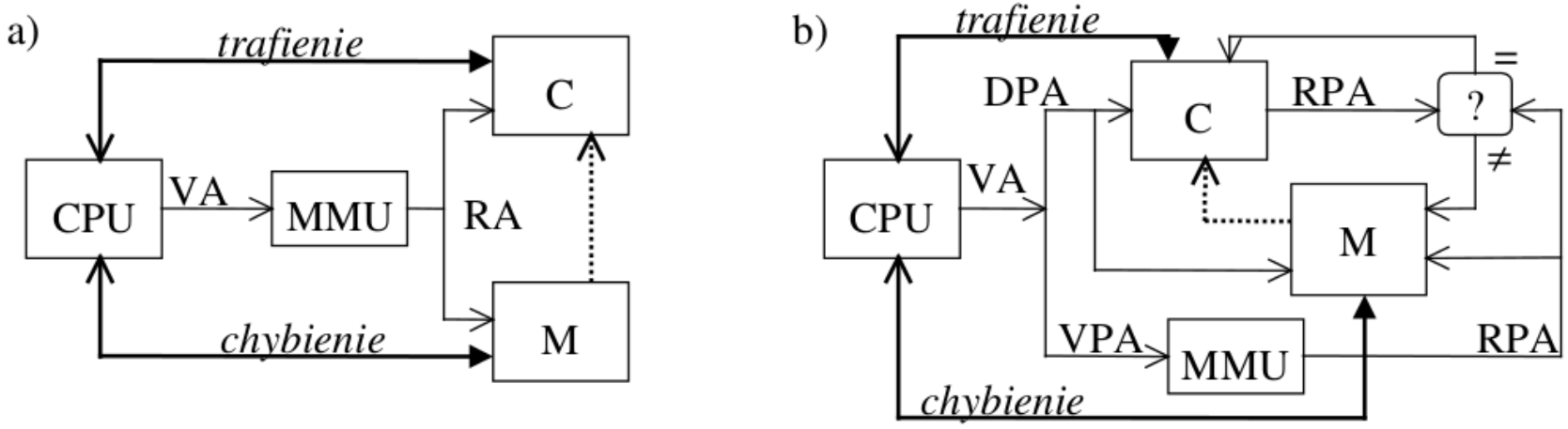
Protokół uzgadniania

- działania procesora (CPU action)
- działania na magistrali (bus action) – podglądanie magistrali (snooping)

Modele spójności systemu wieloprocesorowego ze wspólna pamięcią

- zapisz–unieważnij (write–invalidate)
  - Synapse– aktualizacja lokalna, inne kopie unieważniane
  - Illinois – aktualizacja lokalna, kopia markowana (wyłączna – dzielona), podobny protokół jak w modelu MESI
  - Berkeley – aktualizacja lokalna, inne kopie tylko do odczytu
- zapisz–aktualizuj (write–update)
  - Firefly – aktualizacja globalna, kopia markowana (wyłączna – dzielona)
  - Dragon – aktualizacja ograniczona (z wyłączeniem pamięci głównej), kopia markowana (wyłączna – dzielona)

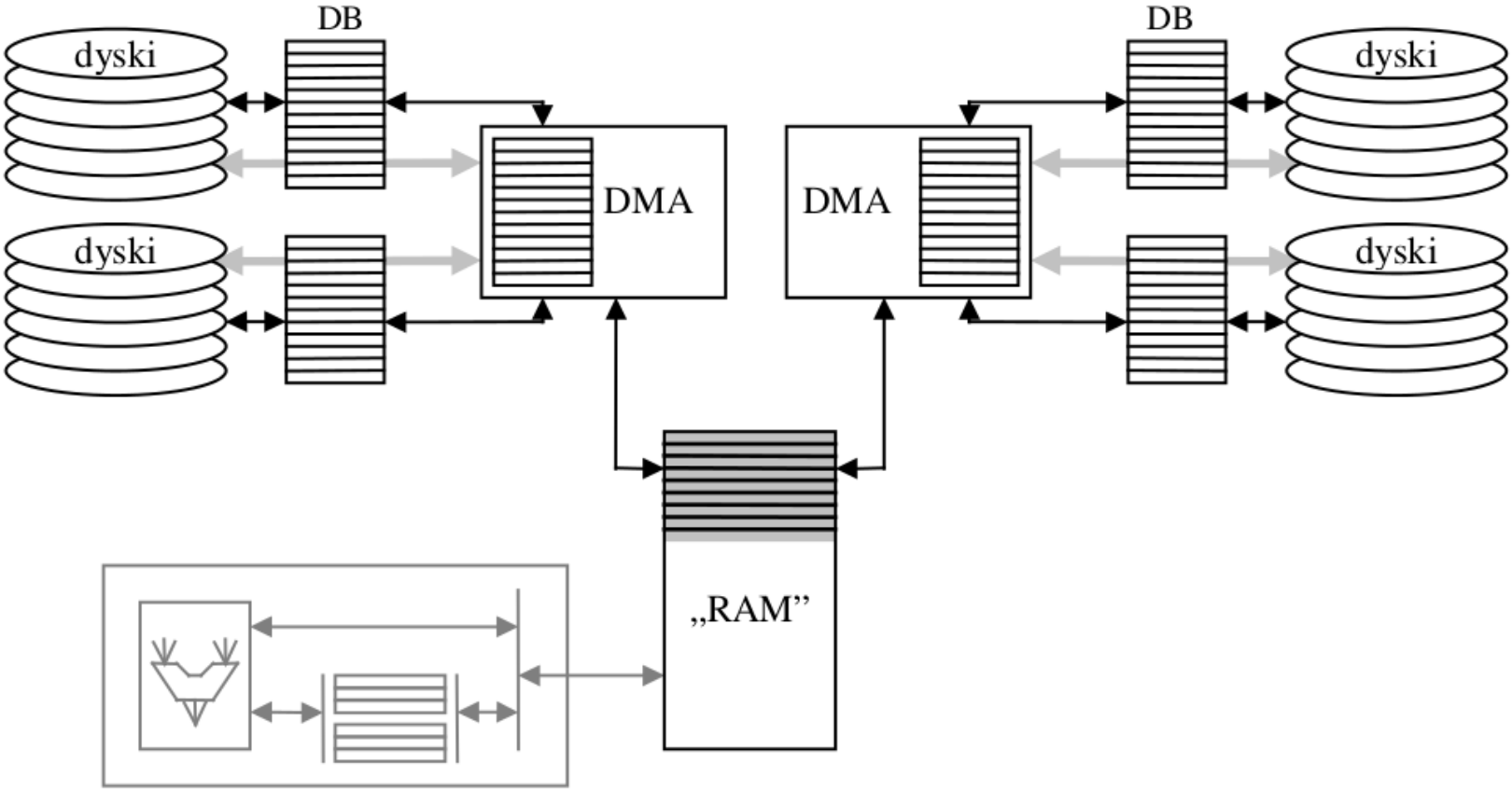
Przyśpieszanie translacji adresu podczas dostępu do pamięci podręcznej



Translacja adresu wirtualnego podczas dostępu do pamięci podręcznej: a) sekwencyjna, b) równoległa. CPU – procesor, MMU – jednostka zarządzania pamięcią, M – pamięć główna, C – pamięć podręczna, VA – adres wirtualny, RA – adres rzeczywisty, VPA – adres strony wirtualnej, RPA – adres strony rzeczywistej, DPA – bezpośredni adres na stronie

Pamięć podręczna dysku (disk cache)

obsługa : copy back (write through wymusza niepotrzebne zapisy)
rozmiar: sektor, ścieżka, cylinder itp.



Możliwe usytuowania bufora pamięci podręcznej dysku