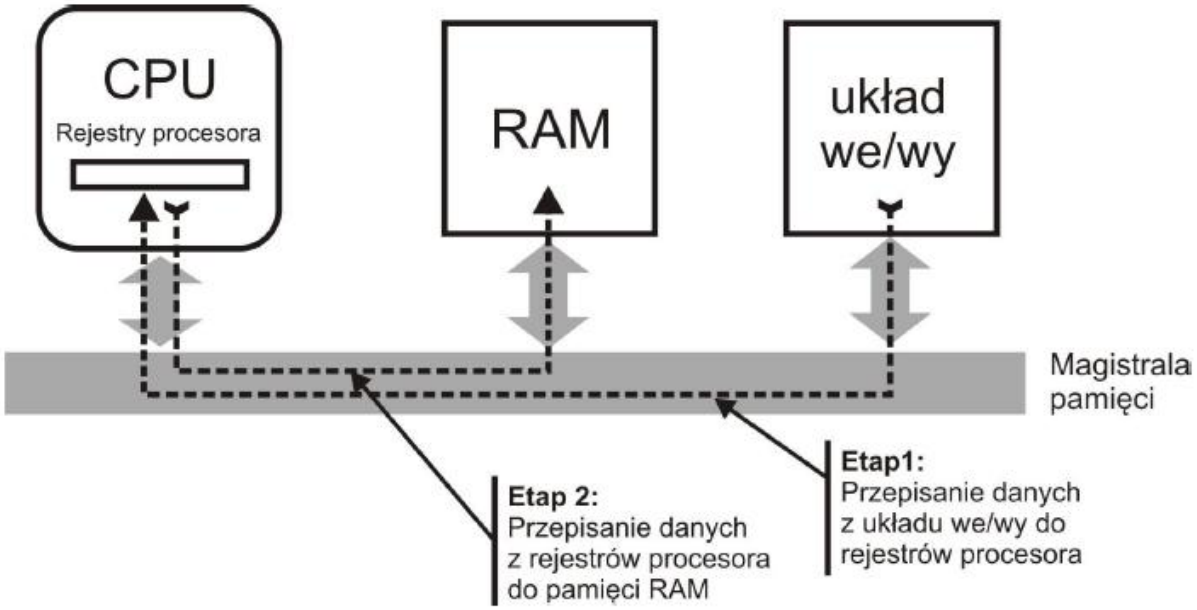
Transmisja w trybie DMA

Operacje wejścia-wyjścia z bezpośrednim sterowaniem przez procesor (PIO)

Definicja:

Operacje wejścia-wyjścia bezpośrednio sterowane przez procesor to takie, w trakcie których wszystkie dane przechodzą przez rejestry procesora.

Wymagają one dużego zaangażowania procesora w procesie transferu danych, dlatego jest używana coraz rzadziej, zwłaszcza, gdy wymagane są duże prędkości transmisji.



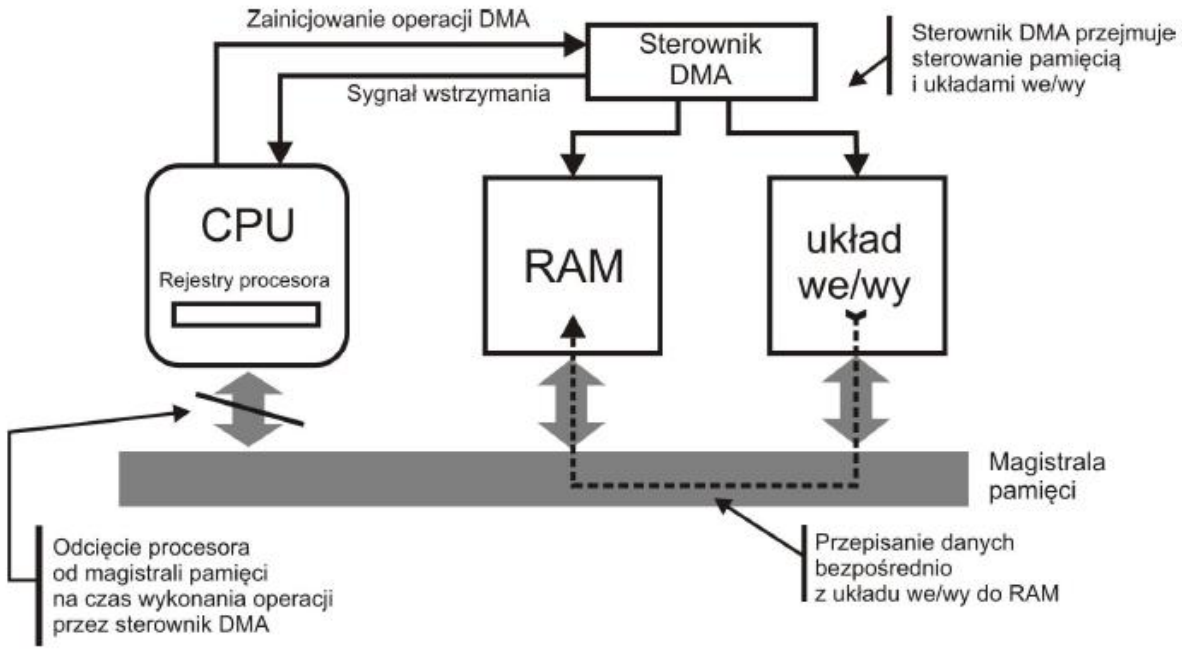
Operacje wejścia-wyjścia z pośrednim sterowaniem przez procesor (DMA)

Przy bezpośrednim dostępie do pamięci, zwanym operacją DMA, transmisja informacji przebiega pomiędzy układem wejścia-wyjścia a wydzielonym obszarem buforowym w pamięci.

Przebieg operacji nadzoruje sterownik DMA, poprzez generację wszystkich sygnałów sterujących i adresów potrzebnych do realizacji wymiany. W tym celu sterownik DMA przejmuje na czas wymiany informacji kontrolę nad magistralami, stając się zarządcą magistral (*busmaster*)

Definicja:

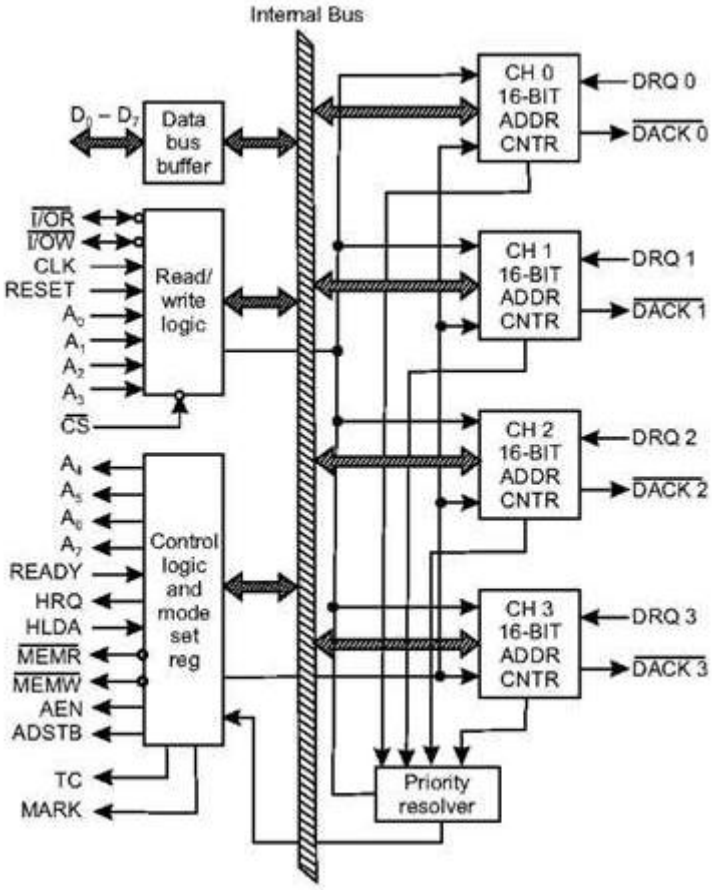
Bezpośrednim dostępem do pamięci nazywamy operację wejścia-wyjścia jedynie inicjowaną przez mikroprocesor, który przekazuje sterowanie jej realizacją specjalizowanemu układowi zwanemu sterownikiem DMA.



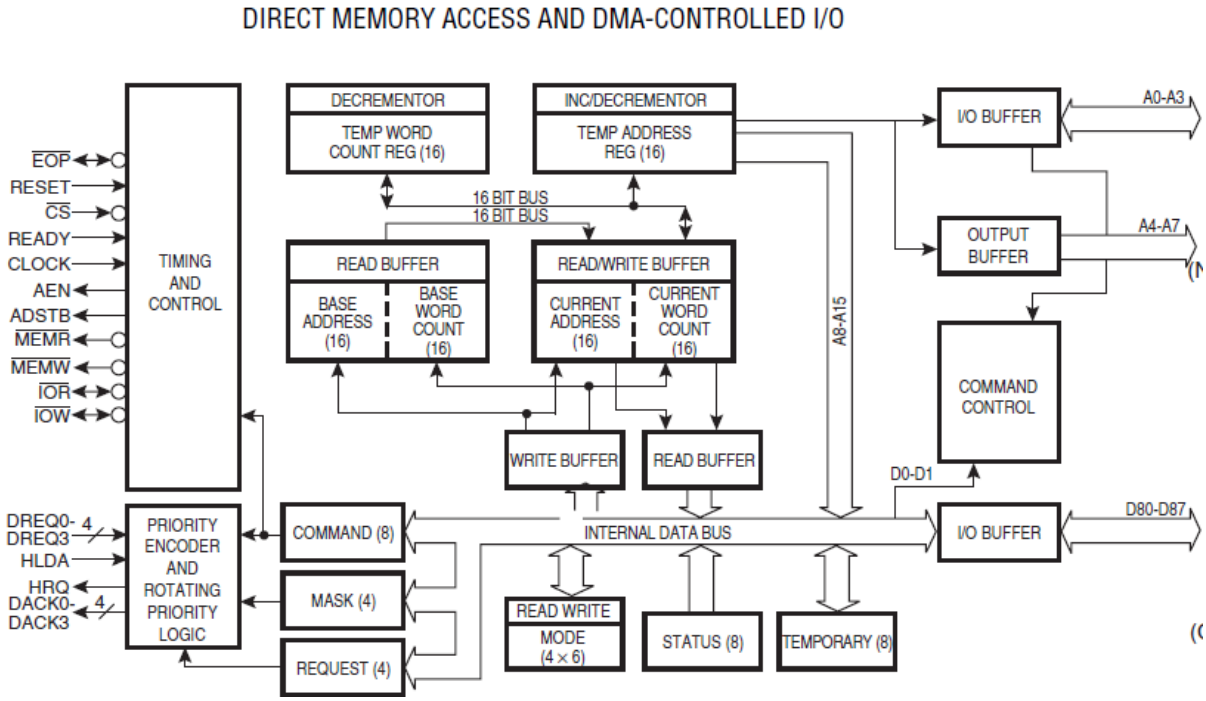
W realizacji operacji DMA możemy wyróżnić trzy podstawowe etapy:

1. inicjacja operacji DMA
2. realizacja operacji DMA
3. zakończenie operacji.

1. Żądanie przejęcia kontroli nad magistralami jest zgłaszane do procesora za pomocą sygnału sterującego HOLD.
2. W odpowiedzi na ten sygnał procesor przechodzi w tak zwany stan zawieszenia, polegający na elektrycznym odseparowaniu się od magistral (stan wysokiej impedancji).
3. Przejście w stan zawieszenia jest sygnalizowane przez mikroprocesor stanem aktywnym na wyjściu HLDA (*hold acknowledge*). Przejście to nie wymaga żadnych zmian stanu rejestrów procesora.
4. Po zakończeniu transmisji (pojedynczego słowa lub bloku, w zależności od trybu realizacji operacji) sterownik DMA zwraca mikroprocesorowi kontrolę nad magistralami.



Schemat blokowy sterownika 8257

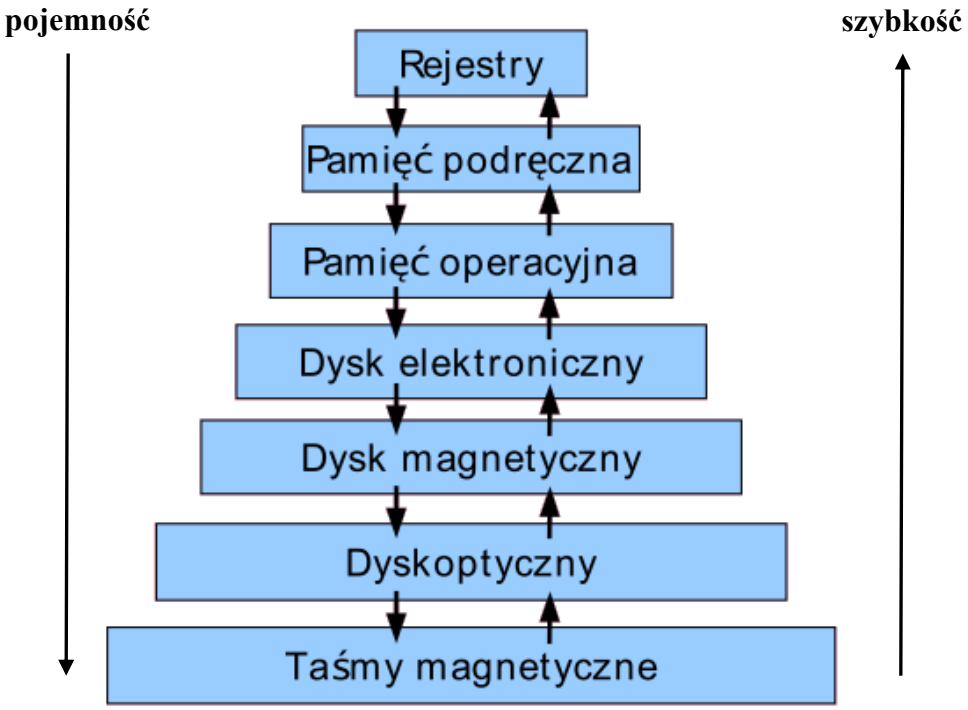


Struktura wewnętrzna układu sterownika DMA

Układ 8237 pozwala realizować transmisje pamięć-pamięć

Hierarchia pamięci w systemie komputerowym

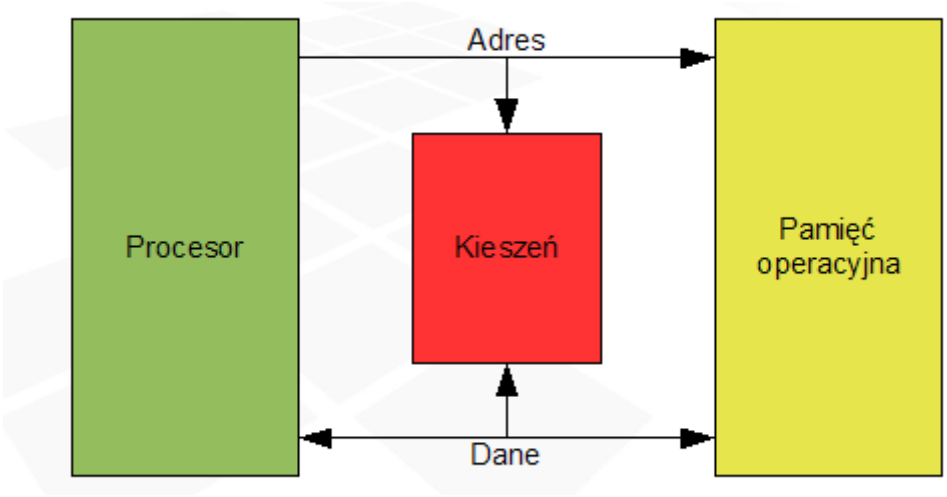
Aby procesor mógł do końca wykorzystać swą wysoką częstotliwość taktowania musi mieć możliwość odpowiednio szybkiego pobierania danych do przetworzenia. Pamięć RAM nawet ta najszybsza i tak nie jest w stanie zapewnić wystarczającej przepustowości danych. Dlatego też od dawna już wykorzystuje się do tego celu szybką pamięć podręczną cache L1 i L2, która na początku (np. procesory 80386) była wlutowana na płycie głównej i współpracowała jedynie z procesorem by z biegiem czasu stać się częścią procesora (od momentu wprowadzenia procesorów 80486).



Pomysł na stosowanie pamięci podręcznej cache wynika z budowy i możliwości pamięci SRAM i DRAM. Pamięci dynamiczne są tańsze od pamięci statycznych i charakteryzują się mniejszym poborem mocy niż pamięci statyczne, są natomiast od nich wolniejsze, zbyt wolne w stosunku do obecnie produkowanych procesorów (chodzi tu zarówno o częstotliwości jak i architekturę procesora). Pamięci statyczne natomiast są o wiele szybsze od dynamicznych jednak ich cena, stopień ich scalenia oraz ilość pobieranej energii dyskwalifikują je jako podstawowy budulec pamięci operacyjnej. Dlatego też zrodził się pomysł żeby pamięć komputera składała się z kilku do kilkuset MB (obecnie kilka GB) pamięci operacyjnej dynamicznej i z kilku - kilkuset KB pamięci podręcznej statycznej. Dodatkowo pamięci te wsparte są przez sterownik cache odpowiadający za współpracę pamięci i reszty podzespołów. Ponad to sterownik ten cały czas monitoruje czy potrzebna informacja jest przechowywana w pamięci podręcznej czy też nie. Jeżeli tak to mamy do czynienia z cache hit (z ang. trafienie) tzn. dane nie muszą być wyszukiwane i pobierane od początku ponieważ znajdują się już w pamięci cache mogą być więc bezpośrednio wysłane do CPU, co znacznie zwiększa wydajność. W drugim przypadku mamy do czynienia

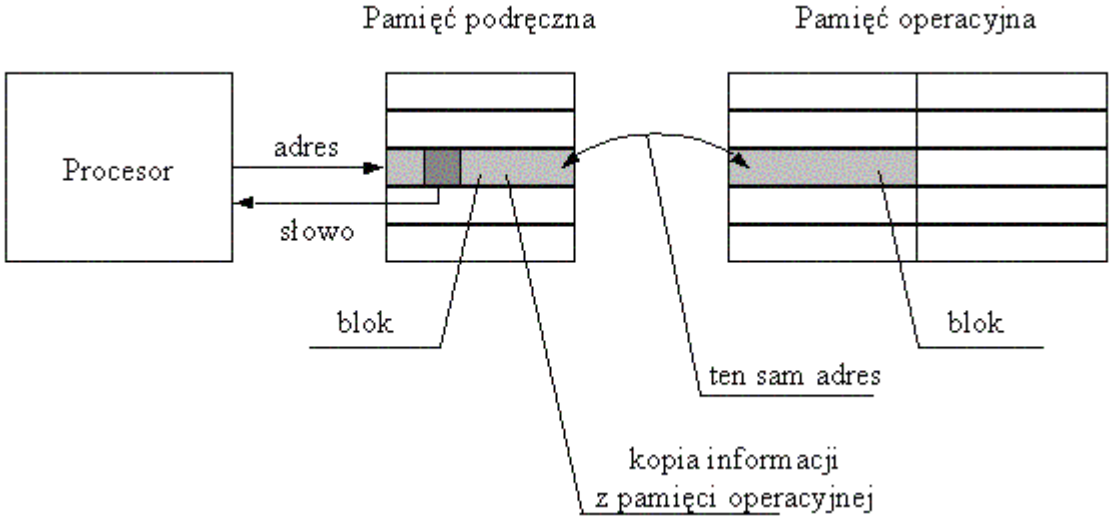
z cache miss (z ang. chybienie) czyli należy pobrać nie obecne w cach'u adresy i dane potrzebne procesorowi w tym celu konieczny jest dostęp do pamięci (wolniejszej) gdzie owe dane są przechowywane i pobranie ich co w rezultacie spowalnia pracę CPU i owocuje spadkiem wydajności komputera.

Pamięć buforowa (cache), pierwszy raz wprowadzona w komputerach IBM S/370 około 1968 roku, stanowi bufor dla pamięci operacyjnej umieszczony między rejestrami a pamięcią operacyjną. Jest ona niewidoczna w użytkowym modelu programowym, a jest niezbędna we współczesnych komputerach z powodu znaczącej dysproporcji pomiędzy wydajnością procesora i pamięci operacyjnej.



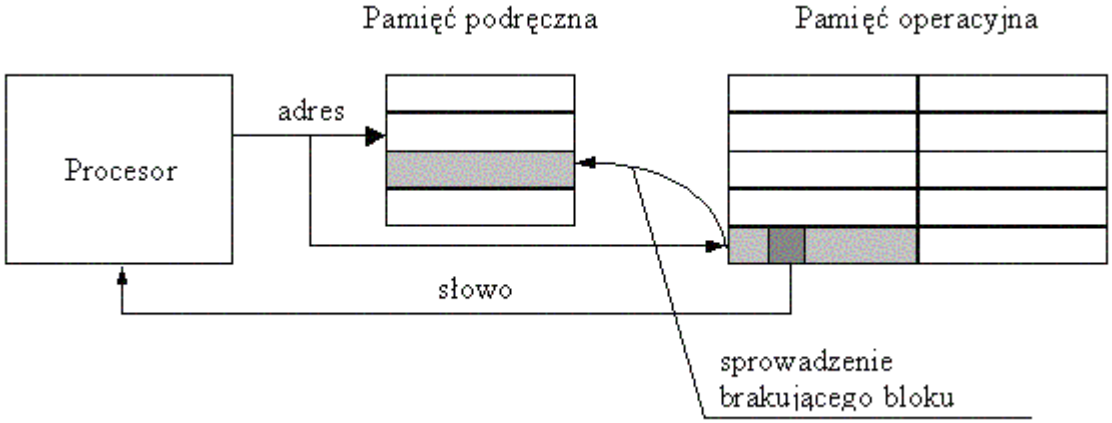
Przy każdym odwołaniu procesora do pamięci następuje sprawdzenie, czy dana spod określonego adresu znajduje się w pamięci buforowej.

- Odnalezienie danej w buforze – trafienie (cache hit)
  - dana zostaje odczytana z bufora, odwołanie do pamięci operacyjnej jest zbędne. Czas dostępu do danej tym razem jest znaczne krótszy niż dostępu do pamięci operacyjnej.



Realizacja odczytu w pamięci podręcznej przy trafieniu

- Brak danej w buforze – chybienie (cache miss)
  - dana zostaje odczytana z pamięci i przesłana do procesora
  - „po drodze" dana wraz z jej adresem jest zapisywana do bufora; jeśli bufor jest pełny – trzeba z niego coś usunąć
  przy następnym odwołaniu dana będzie już w buforze



Realizacja odczytu w pamięci podręcznej przy chybieniu

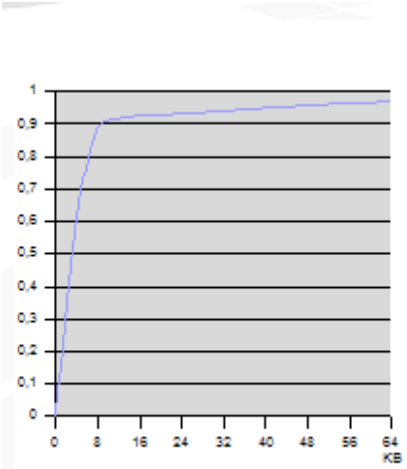
Współczynnik trafień (hit ratio) definiowany jest jako stosunek liczby trafień do całkowitej liczby odwołań w badanym przedziale czasu.

$$h = \frac{n_{cache}}{n_{total}}$$

Zależy on od:

- pojemności bufora
- organizacji pamięci buforowej
- wykonywanego programu

Wiarygodny pomiar i porównanie współczynnika trafień wymaga uzgodnienia budowy testu.



- Wykres przedstawia orientacyjny przebieg zależności współczynnika trafień od pojemności kieszeni
- W zakresie wartości od 0 do 0,9 h zależy głównie od pojemności kieszeni
  - wartość 0,9 jest osiągana przy pojemności kieszeni rzędu 8 KB
- W zakresie powyżej 0,9 istotny wpływ ma również organizacja i algorytm wymiany
  - wyższa asocjacyjność daje wyższy współczynnik trafień

Średni czas dostępu do hierarchii pamięci złożonej z bufora i pamięci operacyjnej wynosi:

$$t_{AVG} = h \cdot t_{cache} + (1 - h) \cdot t_{MEM}$$

h – współczynnik trafień
m=1-h – współczynnik chybień

Jeśli przyjmiemy h=0,9 to

$$t_{AVG} = 0{,}9 \cdot 2ns + (1 - 0{,}9) \cdot 10ns = 2{,}8ns$$

Zasada lokalności odwołań – w ograniczonym odcinku czasu odwołania procesora do pamięci są skupione na niewielkim fragmencie przestrzeni adresowej.

- Zakres odwołań jest ograniczony:
  - zakres adresów, do których odwołuje się procesor w ograniczonym odcinku czasu nazywa się **zbiorem roboczym**
  - stosunkowo niewielki bufor może przechowywać znaczącą część obiektów, do których w danym odcinku czasu odwołuje się procesor.
- Odwołania są na ogół powtarzane
  - należy zapamiętać dane, do których procesor wykonuje dostęp, bo zapewne wkrótce będzie znów ich potrzebował
- Bardzo prawdopodobne są kolejne odwołania do kolejnych adresów
  - przy napełnianiu bufora wskutek dostępu ze strony procesora warto pobrać z pamięci kilka kolejnych komórek „na zapas"

Dla uproszczenia wyjaśnienia sposobu organizacji pamięci podręcznej założyliśmy tylko jeden poziom pamięci podręcznej w komputerze. Jeżeli występują dwa lub więcej poziomów pamięci podręcznej, to przy chybieniu na pierwszym poziomie, adres jest przekazywany sprzętowo do pamięci podręcznej drugiego poziomu. Jeśli tam uzyskamy trafienie, to żądane słowo jest pobierane z pamięci drugiego poziomu a blok zawierający dane słowo jest sprowadzany do pamięci podręcznej pierwszego poziomu. O ile i w drugiej pamięci podręcznej mamy chybienie, to słowo jest pobierane z pamięci operacyjnej a bloki zawierające te słowa są sprowadzane do pamięci podręcznych obu poziomów. Rozmiar bloku pamięci pierwszego poziomu wynosi od 8 do kilkudziesięciu bajtów (liczba stanowi potęgę dwu). Rozmiar bloku pamięci drugiego poziomu jest wielokrotnie większy od rozmiaru bloku pamięci pierwszego poziomu.

Podzespół pamięci podręcznej może być różnie podłączony do procesora i pamięci operacyjnej:

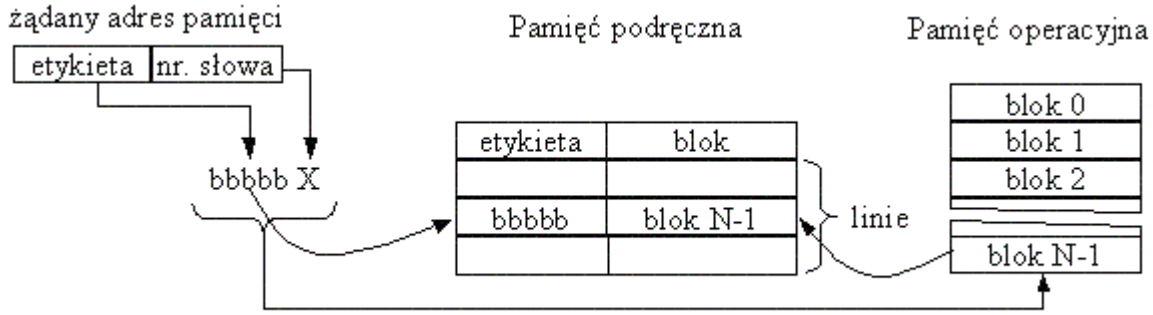
- jako dodatkowy układ podłączony do szyny systemowej łączącej procesor z pamięcią operacyjną
- jako układ pośredniczący między procesorem a pamięcią operacyjną
- jako układ osobno podłączony do procesora, równolegle do pamięci operacyjnej.

To trzecie rozwiązanie jest obecnie stosowane najczęściej.

Obecnie omówimy różne rodzaje organizacji informacji w pamięci podręcznej. Znane są trzy główne metody odwzorowania w pamięci podręcznej informacji przychodzącej z pamięci operacyjnej:

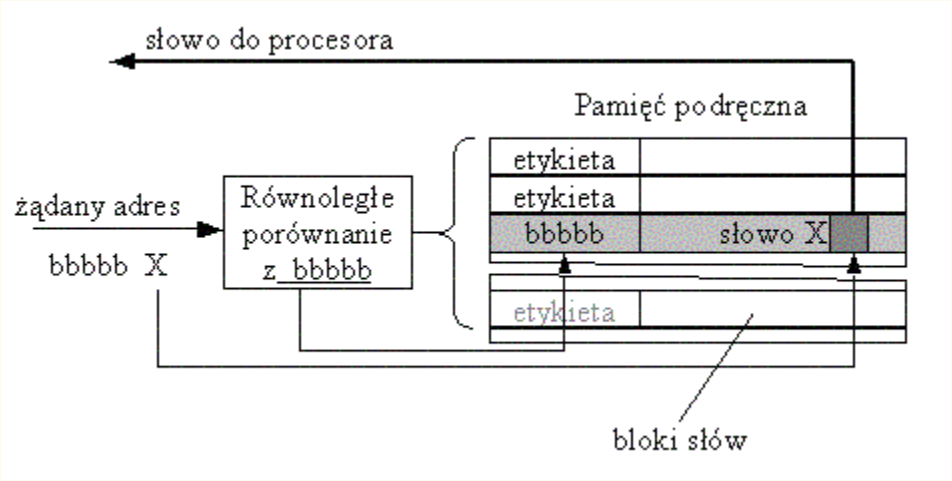
- **odwzorowanie asocjacyjne** (ang. **associative mapping**)
- **odwzorowanie bezpośrednie** (ang. **direct mapping**)
- **odwzorowanie zbiorowo asocjacyjne** (ang. **set associative mapping**).

Pamięć asocjacyjna (skojarzeniowa)



Asocjacyjna organizacja informacji w pamięci podręcznej

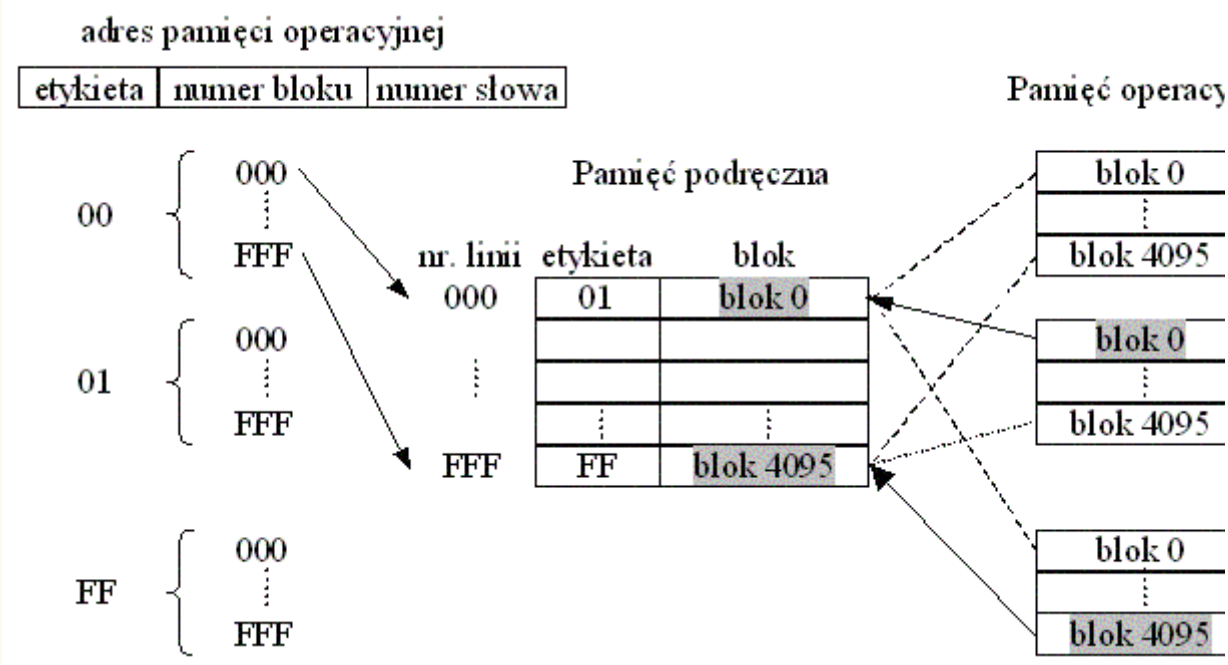
Określa się ją jako pamięć adresowana zawartością, nie posiada adresów, a dostęp do danej następuje poprzez porównanie części danej zwanej kluczem z wzorcem dostarczonym z zewnątrz. Pamięć odpowiada poprzez wystawienie danych zgodnych z wzorcem lub informacji, że takich danych nie ma.



Odczyt z pamięci podręcznej z odwzorowaniem asocjacyjnym przy trafieniu

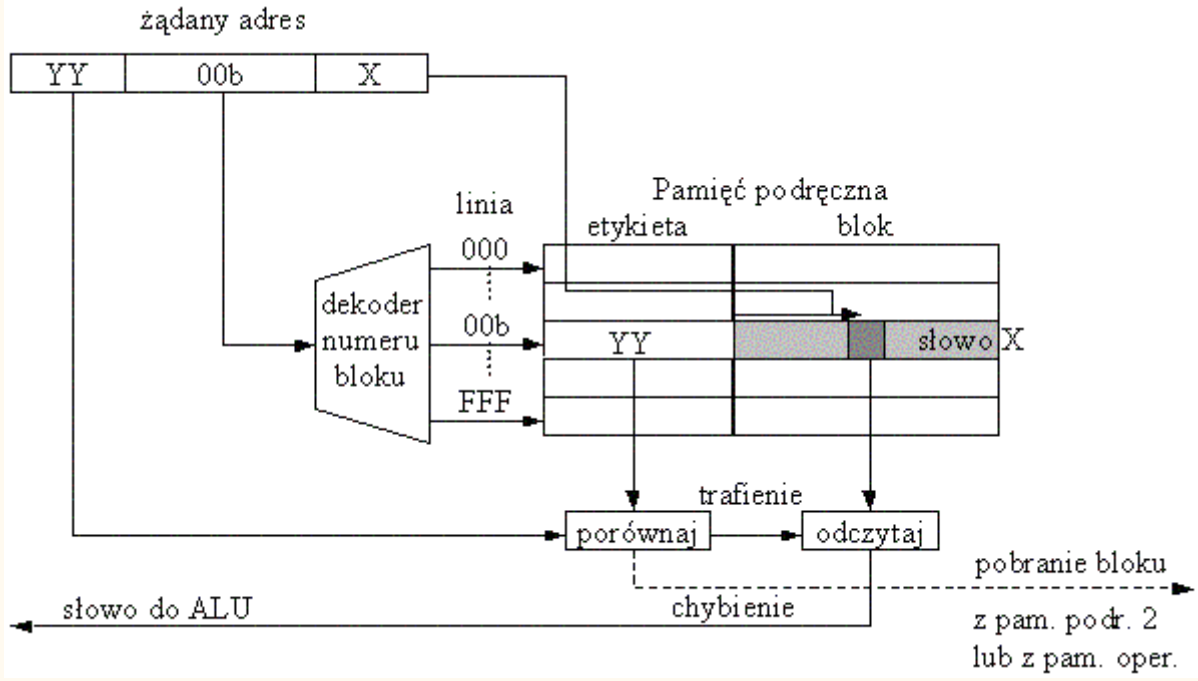
W każdej komórce pamięci buforowej może być przechowywana dana spod dowolnego adresu – duża elastyczność w porównaniu z następnymi architekturami. Linia do usunięcia wyznaczana według algorytmu LRU (kosztowna implementacja) lub losowo.
Każda komórka wyposażona jest w komparator znacznika – trudna implementacja, niewielka pojemność pamięci.
Jeśli rozmiar zbioru roboczego przekracza pojemność pamięci buforowej to wszystkie odwołania będą kończyły się chybieniami.
Dane są przechowywane w buforze nie w postaci pojedynczych słów czy bajtów, lecz bloków zazwyczaj o długości 4x większej od rozmiaru słowa pamięci. Element zawierający blok danych i związane z nim znaczniki jest nazywany linią. Najmniej znacząca część adresu służy do wyboru bajtu lub słowa z linii. Kolejne bity adresu są używane do stwierdzenia, czy poszukiwana dana znajduje się w buforze.

Pamięć bezpośrednio adresowana (odwzorowanie bezpośrednie)



Organizacja pamięci podręcznej z odwzorowaniem bezpośrednim

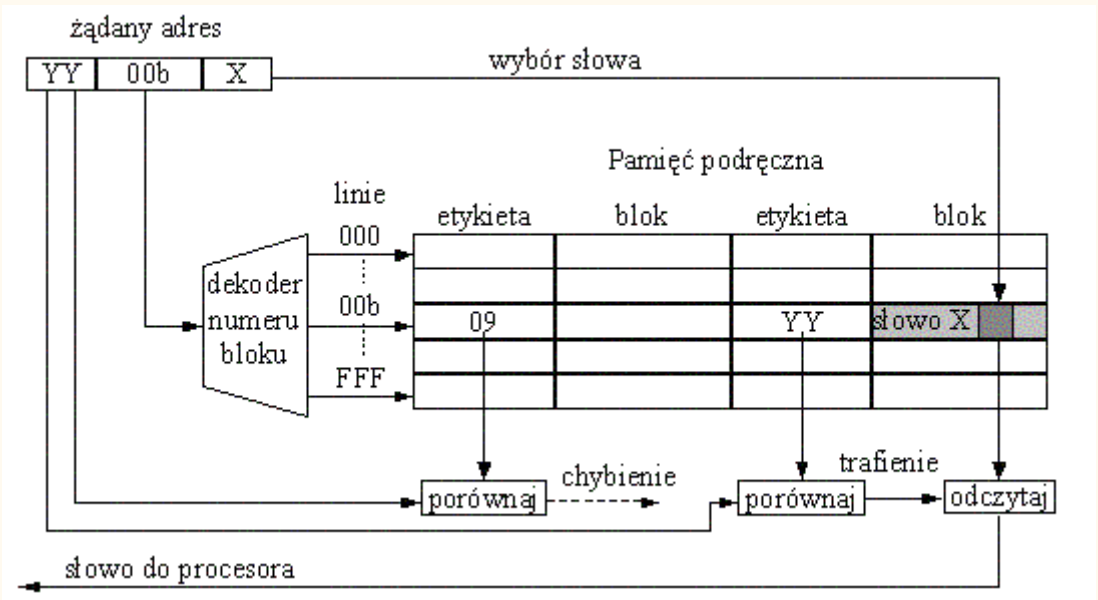
Zbudowana jest na bazie zwykłej, szybkiej pamięci RAM i jednego komparatora. Dzięki prostocie budowy może mieć stosunkowo dużą pojemność.



Odczyt z pamięci podręcznej z odwzorowaniem bezpośrednim

Najmniej znaczące bity adresu służą do wyboru bajtu z linii. Środkowa, mniej znacząca część adresu procesora służy jako adres pamięci RAM; na jej podstawie w każdym cyklu dostępu jest wybierana pojedyncza linia.
Każda linia zawiera znacznik adresu i dane. Pole znacznika adresu zawiera bardziej znaczącą część adresu danej zapamiętanej w polu danych – jest ono porównywane z najbardziej znaczącą częścią adresu wystawionego przez procesor.
Pamięć stwierdza trafienie, jeśli znacznik adresu wybranej linii jest równy najbardziej znaczącej części adresu wystawionego przez procesor. W przypadku trafienia dane są transmitowane z bufora do procesora. W przypadku chybienia wymianie podlega wybrana linia – w polu znacznika zostaje zapisana najbardziej znacząca część adresu, w polu danych zostają zapamiętane dane odczytane z pamięci.
Pamięć buforowa bezpośrednio adresowana charakteryzuje się niskim kosztem, dużą pojemnością i wysoką wydajnością.
Algorytm zastępowania linii wymuszony jest przez budowę pamięci buforowej. Dane spod określonego adresu mogą znaleźć się wyłącznie w jednej, z góry określonej linii pamięci – w pamięci buforowej nie można zapamiętać dwóch danych, których środkowe części adresu są identyczne.

Pamięć buforowa zbiorowo-asocjacyjna



Odczyt z pamięci podręcznej z odwzorowaniem zbiorowo asocjacyjnym przy trafieniu

Pamięć taka powstaje przez połączenie pewnej liczby pamięci buforowych bezpośrednio adresowanych zwanych blokami. Dana spod określonego adresu może być przechowywana w tylu miejscach ile jest bloków. W każdym cyklu dostępu następuje poszukiwanie danej w pojedynczej linii każdego z bloków. Zestaw linii wybieranych w każdym cyklu jest nazywany **zbiorem**. Zbiór zachowuje się jak mała pamięć asocjacyjna.

Liczba bloków jest zwana stopniem asocjacyjności, używa się również określeń „pamięć buforowa dwudrożna” lub „czterodrożna”. Pamięć buforowa zbiorowo-asocjacyjna może być również rozpatrywana jako złożenie pewnej liczby pamięci asocjacyjnych.
Budowa pamięci buforowej musi gwarantować, że dana spod określonego adresu może zostać zapisana tylko w jednym bloku. W przypadku chybienia należy wyznaczyć ze zbioru jedną linię do zastąpienia (według algorytmu LRU lub losowego).
Ten typ pamięci buforowej charakteryzuje się mniejszą wrażliwością na nakładanie się adresów danych.

Pamięci buforowe wielopoziomowe

Wymóg nadążania pamięci buforowej pierwszego poziomu (L1) ogranicza jej pojemność i asocjacyjność – im większa pamięć tym wolniejsza, im większa asocjacyjność tym dłuższy czas dostępu.
Pamięć drugiego poziomu (L2) może być wolniejsza (np. 5 razy) – dzięki czemu może być znacząco większa i mieć wyższą asocjacyjność. Jeżeli pamięć buforowa L2 nie zapewnia odpowiednio krótkiego średniego czasu dostępu, w strukturze komputera umieszcza się pamięć L3, większą i wolniejszą od L2.

Operacja zapisu do pamięci.

W dotychczasowych rozważaniach rozpatrywaliśmy wyłącznie odczyt danych lub instrukcji z pamięci. W przypadku zapisu do pamięci możliwe są następujące rozwiązania:

- zapis przeźroczysty – zapis wykonywany zawsze do pamięci, a w przypadku trafienia również do bufora,
- zapis zwrotny – zapis do pamięci wykonywany tylko wtedy, kiedy jest to niezbędne.

Przy zapisie zwrotnym możliwe są warianty:

- bez alokacji przy chybieniu zapisu – w razie chybienia zapis zachodzi tylko do pamięci, w razie trafienia – tylko do bufora,
- z alokacją przy chybieniu zapisu – zapis jest zawsze wykonywany tylko do bufora.

Przy zapisie zwrotnym usunięcie linii z bufora może wymagać zapisu usuwanej linii do pamięci.

Strategie zapisu danych

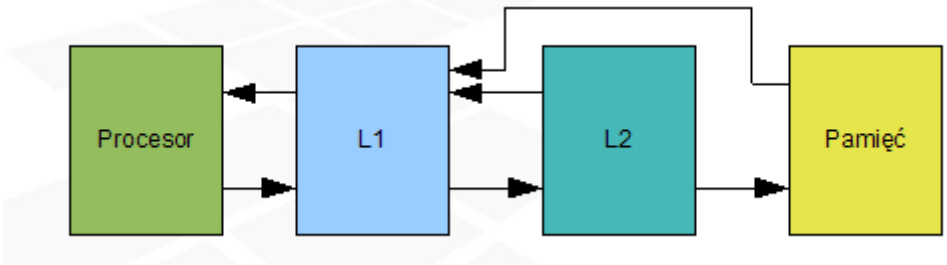
Typ kieszeni	trafienie odczytu	chybienie odczytu	trafienie zapisu	chybienie zapisu
zapis przezroczysty	P←C	C←M, P←C	P→C, P→M	P→M
zapis zwrotny bez alokacji przy chybieniu zapisu	P←C	C→M, C←M, P←C	P→C	P→M
zapis zwrotny z alokacją przy chybieniu zapisu	P←C	C→M, C←M, P←C	P→C	C→M, C←M, P→C

Oznaczenia: C – kieszeń, P – procesor, M – pamięć
Kolor czerwony – dane usuwane z kieszeni w wyniku zastąpienia linii

Ścieżki danych

We wcześniejszych rozwiązaniach dane pobierane z pamięci głównej przekazywane były do pamięci buforowej L2, a stamtąd do L1 i procesora. Każdy obiekt zawarty w warstwie wyższej obecny był również w warstwie niższej. Efektywna sumaryczna pojemność pamięci buforowej jest równa pojemności największej z warstw. Pojemność L2 powinna być znacząco większa od L1.

W obecnych rozwiązaniach bufor L2 jest napełniany wyłącznie obiektami usuwanymi z L1 (jest to tzw. bufor ofiar – victim cache). L2 zawiera głównie obiekty nieobecne w L1. Efektywna sumaryczna pojemność pamięci buforowej jest równa sumie pojemności poszczególnych warstw. Pojemność L2 może być większa lub równa L1. Asocjacyjność L2 powinna być większa od asocjacyjności L1 – w przeciwnym przypadku sprawność przechwytywania ofiar byłaby niewielka.
Przykładami takich rozwiązań są procesory K7 i K8 firmy AMD oraz Pentium 4 i Core firmy Intel.



Spójność hierarchii pamięci.

Hierarchię pamięci musi cechować spójność co oznacza, że każdy dostęp do danego adresu musi zwrócić tę samą wartość danej, niezależnie od warstwy, w której jest fizycznie zrealizowany.
Problem z utrzymaniem spójności powstaje, gdy istnieje więcej niż jedna ścieżka dostępu do hierarchii pamięci, np.

- procesor o architekturze Harvard-Princeton z oddzielnymi buforami L1 kodu i danych,
- dwa procesory z oddzielnymi buforami L1 i wspólną dalszą warstwą hierarchii pamięci,
- procesor z pamięcią buforową i sterownik wejścia-wyjścia, wykonujący bezpośredni dostęp pamięci z pominięciem pamięci buforowej.

Uwaga: Spójność nie wymaga utrzymywania identycznej zawartości we wszystkich warstwach, a jedynie zagwarantowania, że każdy dostęp będzie dotyczył aktualnej wartości danej.

Metody utrzymania spójności opierają się na selektywnej zmianie stanu linii po stwierdzeniu zgodności adresu, przesłanie między buforami. Wymagają realizacji złożonych protokołów utrzymania spójności.
Automat zrealizowany jest oddzielnie dla każdej linii, która może znajdować w następującym stanie:

- M – modified – linia ważna, jedyna aktualna kopia we własnym buforze, zawartość pamięci nieaktualna,
- E – exclusive – linia ważna, jedyna kopia we własnym buforze, identyczna z zawartością pamięci,
- I – invalid – linia nieważna,
- S- shared – linia ważna, jednakowa kopia u wszystkich, identyczna z zawartością pamięci,
- O – owned – linia ważna, jednakowa kopia u wszystkich, u pozostałych stan S, zawartość pamięci nieaktualna.

Nazwy protokołów utrzymania spójności pochodzą od zbioru obsługiwanych stanów – MEI, MESI, MOESI. Im więcej stanów, tym mniej zbędnych unieważnień.

Pamięć podręczna zawiera kopie danych zawartych w pamięci operacyjnej. Gdy następuje zmiana danych w pamięci podręcznej (np. modyfikacja w wyniku wykonania operacji przez procesor) zawartość komórek pamięci operacyjnej i pamięci podręcznej o tym samym adresie różnią się. Aby zlikwidować tę niespójność stosowane są dwie metody:

- **uaktualnianie natychmiastowe** (ang. **write through**), zapis nowej zawartości do pamięci operacyjnej następuje od razu po zakończeniu zapisu do pamięci podręcznej,
- **uaktualnianie opóźnione** (ang. **write back**), zapis nowej zawartości do pamięci operacyjnej nie następuje od razu po zakończeniu zapisu do pamięci podręcznej a dopiero wtedy, gdy dany blok w pamięci podręcznej jest zastępowany nowym blokiem sprowadzonym z pamięci operacyjnej. Po zapisie pamięci podręcznej następuje jedynie zmiana bitu stanu w bloku, wskazująca, że blok został zmodyfikowany.

Uaktualnianie opóźnione jest bardziej ekonomiczne czasowo, dlatego, że jeśli poszczególne komórki bloku były wielokrotnie modyfikowane w czasie gdy pozostawały w pamięci podręcznej, ich uaktualnienie wykonane jest jednokrotnie.