Rozproszona pamięć dzielona - 1

Wieloprocesor - wiele CPU ma dostęp do wspólnej pamięci głównej
Wielokomputer - każdy CPU ma własną pamięć główną; nie ma współdzielenia pamięci

Aspekt sprzętowy:

- Skonstruowanie wieloprocesora jest trudne;
  - w wieloprocesorze z pojedynczą szyną (ang. *bus-based multiprocessor*) szyna jest często wąskim gardłem
  - w wieloprocesorze z wieloma połączeniami (ang. *switched multiprocessor*) można dodawać procesory, ale system jest kosztowny, wolny i złożony
- Duże wielokomputery buduje się łatwo

Aspekt programowy:

- Programowanie wieloprocesorów jest proste: dostęp do wspólnej pamięci, programowe mechanizmy synchronizacji procesów
- Programowanie wielokomputerów jest trudne: komunikacja poprzez przesyłanie komunikatów; problem gubionych komunikatów, buforowania, zakładania blokad, etc.

Komunikacja w wielokomputerach

- Przesyłanie komunikatów między różnymi przestrzeniami adresowymi
- **DSM** (**D**istributed **S**hared **M**emory) (1986, Li & Hudak)

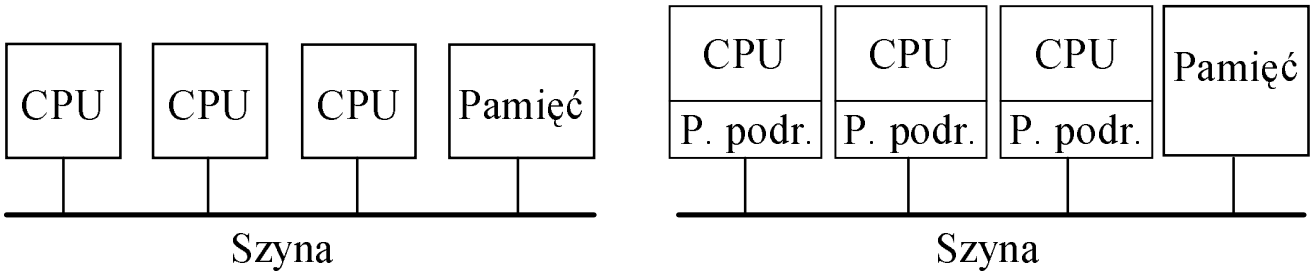
Zbiór stacji roboczych połączonych w sieć lokalną współdzieli jedną stronicowaną wirtualną przestrzeń adresową. Odwołania do stron lokalnych odbywają się sprzętowo (w tradycyjny sposób), odwołania do stron zdalnych powodują błąd braku strony, SO wysyła komunikat do zdalnej maszyny, która znajduje i przesyła żądaną stronę

Zalety: łatwe do zbudowania i programowania

Wady: słaba wydajność („zdalne migotanie”)
Problem: jak zmniejszyć ruch stron w sieci?
Możliwe rozwiązania:
- dzielić nie całą przestrzeń adresową, lecz tylko niektóre zmienne
- powielać dzielone zmienne na różnych maszynach
- podejście obiektowe: dzielone obiekty

Sposoby realizacji pamięci dzielonej

Wieloprocesor z pojedynczą szyną



- Dostęp do pamięci poprzez szynę
- Rozstrzyganie konfliktów w dostępie do szyny
- Szyna potencjalnym wąskim gardłem (kłopoty ze skalowalnością) - do 64 CPU
- *Podsłuchująca pamięć podręczna* (ang. *snooping cache*)
- Protokoły zapewniające zgodność danych przechowywanych w pamięci podręcznej (ang. *cache consistency protocol*)

Natychmiastowe pisanie (ang. *write through*)
Akcja podjęta przez pamięć podręczną w reakcji na działanie własnego CPU:
- Czytanie
  - nie ma: pobierz dane z pamięci i umieść w pamięci podręcznej
  - jest: pobierz dane z lokalnej pamięci podręcznej
- Pisanie
  - nie ma: uaktualnij dane w pamięci i umieść w pamięci podręcznej
  - jest: uaktualnij pamięć i pamięć podręczną

Akcja podjęta przez pamięć podręczną w reakcji na działanie zdalnego CPU:

- Pisanie
  - jest: unieważnij dane w lokalnej pamięci podręcznej
- Wpp: nic nie rób

Jednokrotne pisanie (ang. *write once*)

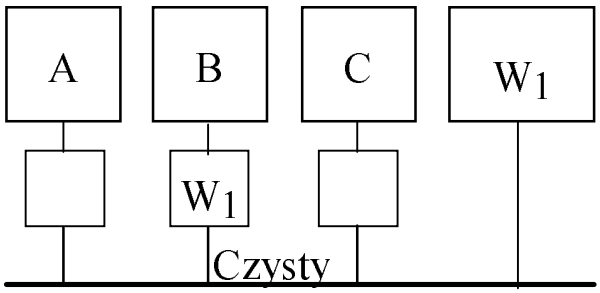
Możliwy stan bloku pamięci podręcznej:

Unieważniony - blok pamięci podręcznej nie zawiera aktualnych danych

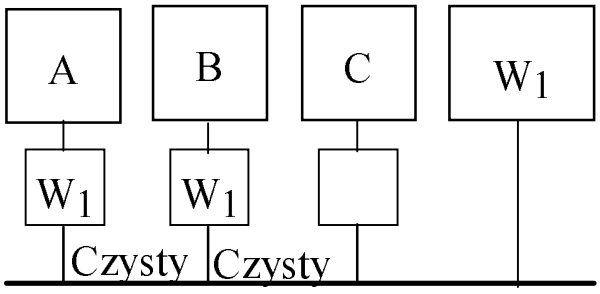
Czysty - pamięć zawiera aktualne dane, blok może przebywać w innych pamięciach podręcznych

Brudny - pamięć nie zawiera aktualnych danych, bloku nie ma w innych pamięciach podręcznych

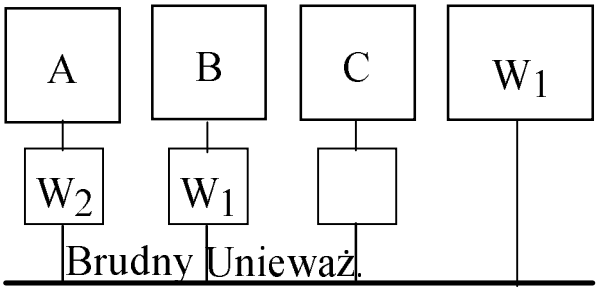
Przykład:



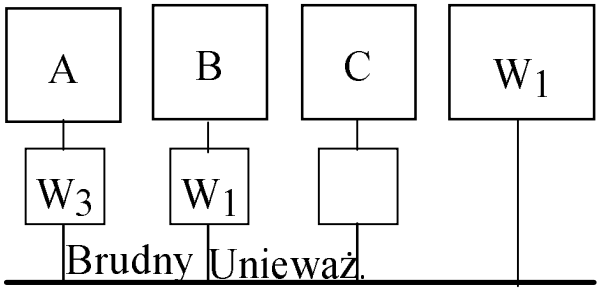
Stan inicjalny
Słowo W o wartości W_1 jest w pamięci i w pamięci podręcznej maszyny B



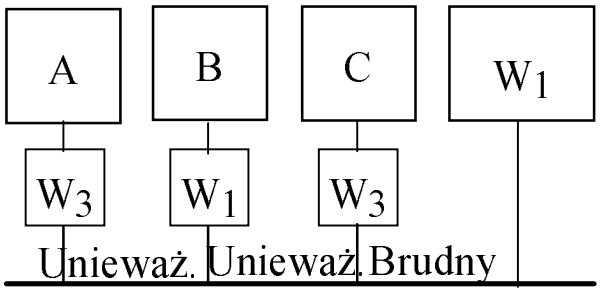
A czyta słowo W i dostaje W_1. B nie reaguje na polecenie czytania, robi to pamięć główna



A zapisuje W_2. B podsłuchuje szynę, widzi polecenie pisania i unieważnia swoją kopię. Kopia A jest oznaczona jako brudna



A znowu zapisuje W. Ta i kolejne operacje pisania są wykonywane lokalnie, bez obciążania szyny



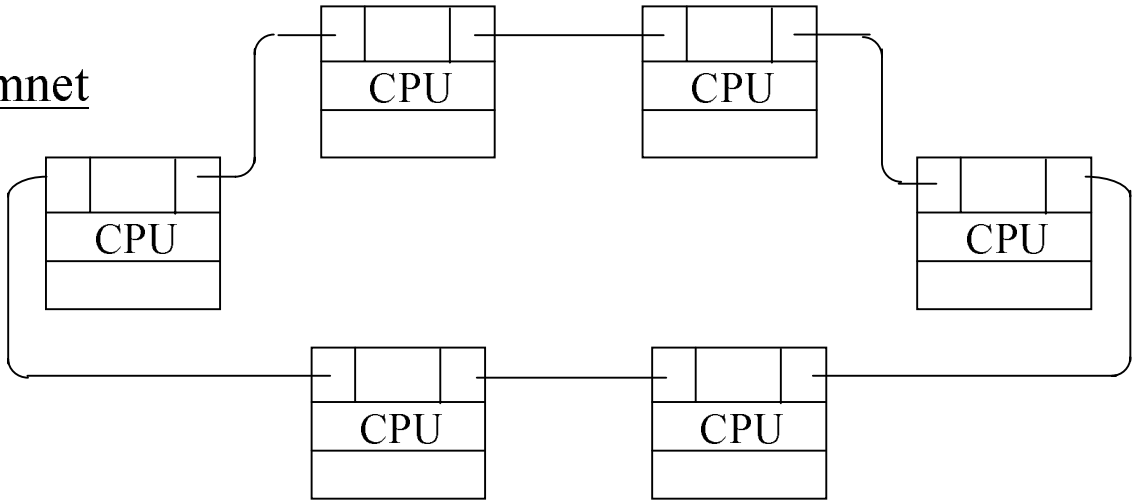
C czyta lub zapisuje W. A widzi żądanie (podsłuchuje szynę), dostarcza wartość i unieważnia własną kopię. C ma teraz jedyną aktualną kopię

Cechy tego protokołu:

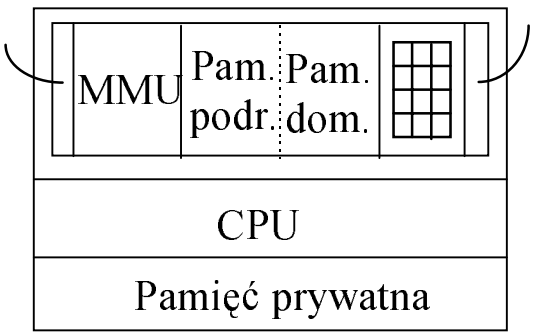
- Zgodność uzyskuje się dzięki temu, że wszystkie pamięci podręczne podsłuchują co transmituje szyna
- Protokół jest wbudowany w jednostkę zarządzającą pamięcią
- Cały algorytm wykonuje się w czasie krótszym od cyklu pamięci

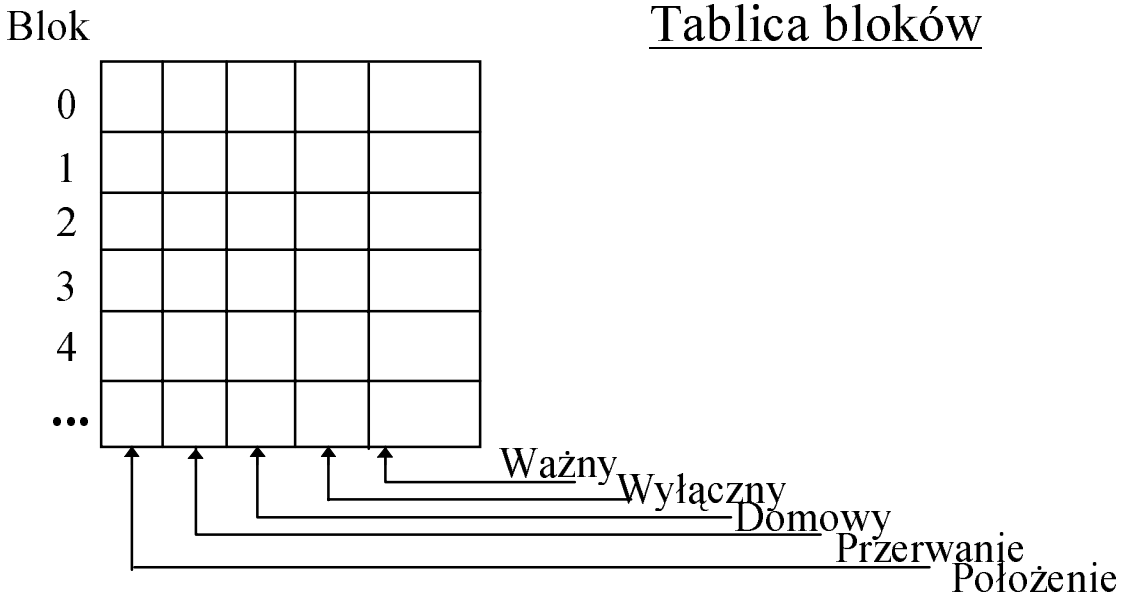
Wieloprocesor w pierścieniu

Memnet



Pojedyncza maszyna





- Pojedyncza przestrzeń adresowa składa się z części prywatnej i dzielonej
- Część dzielona jest wspólna dla wszystkich maszyn i rozproszona między nimi; składa się z bloków 32-bajtowych (jednostka transmisji). Każdy taki blok ma swoją maszynę domową (która trzyma dla niego pamięć fizyczną)
- Tablica bloków zawiera po jednej pozycji dla każdego bloku pamięci dzielonej. Znaczenie bitów (pól):
  Ważny - blok jest w pamięci podręcznej i jest aktualny
  Wyłączny - lokalna kopia jest jedyną
  Domowy - komputer jest domowym komputerem tego bloku
  Przerwanie - używany do wymuszania przerwań
  Położenie - położenie bloku w pamięci podręcznej (jeśli jest obecny i aktualny)
- Protokół:

Czytanie: jeśli blok jest dostępny, to zostaje odczytany; wpp interfejs czeka na żeton, wkłada do pierścienia pakiet z żądaniem odczytu (adres + 32 bajtowe pole). Interfejs maszyny, która ma ten blok, wstawi go do tego pakietu (ew. czyszcząc bit wyłączności) Jeśli żądająca maszyna nie ma w pamięci podręcznej miejsca na ten blok, to odsyła „do domu” losowy „cudzy” blok

Pisanie: Jeśli zapisywany blok jest obecny i jest to jedyna kopia, to zapis jest lokalny. Jeśli jest obecny, ale nie jest to jedyna kopia, to interfejs wysyła najpierw komunikat z poleceniem unieważnienia pozostałych kopii. Po powrocie tego komunikatu wykonuje lokalny zapis i ustawia bit wyłączności. Jeśli bloku nie ma, to zostaje wysłany komunikat z żądaniem odczytu i unieważnienia. Pierwsza maszyna posiadająca blok kopiuje go do pakietu i likwiduje własną kopię. Wszystkie pozostałe likwidują swoje kopie. Blok po dotarciu do adresata zostaje zapamiętany i zapisany

Wieloprocesor z wieloma połączeniami

Problem skalowalności: rozbudowanie systemu o dużą liczbę procesorów wymaga zwiększenia przepustowości łączy komunikacyjnych
Możliwy sposób: budowa systemu jako hierarchii
Pojedyncze **grono** (ang. *cluster*) składa się z kilku CPU i z pamięci połączonych szyną. System składa się z wielu takich gron połączonych szyną poprzez specjalny interfejs (tzw. **supergrono**). Supergrona również można łączyć ze sobą szyną (poprzez interfejs) tworząc jeszcze większe systemy

Przykład: maszyna **Dash** (**D**irectory **A**rchitecture for **Sh**ared Memory) zbudowana w Stanford University.
Składa się z 16 gron, każde grono zawiera 4 CPU, 16M pamięci globalnej (czyli łącznie jest 256M pamięci globalnej) i urządzenia we-wy. Każde CPU ma pamięć podręczną i może podsłuchiwać lokalną szynę (i tylko tę). W specjalnych **katalogach** (1M pozycji 18-to bitowych, po jednej dla każdego bloku) przechowuje się informacje o położeniu 16. bajtowych bloków, których właścicielem jest dane grono (mapa bitowa: nr bloku × numer grona oraz opis stanu: *Niebuforowany*, *Czysty*, *Brudny*). Dostęp do bloku może wymagać przesłania wielu komunikatów

Wieloprocesory typu NUMA

Sprzętowe buforowanie w dużych wieloprocesorach jest kosztowne

Inne rozwiązania: **NUMA** (**N**on**U**niform **M**emory **A**ccess).

Maszyna typu NUMA ma pojedynczą wirtualną przestrzeń adresową widzianą przez wszystkie CPU. Dostęp do zdalnej pamięci jest dużo wolniejszy niż dostęp lokalny i nie próbuje się niwelować tej różnicy za pomocą sprzętowego buforowania

Pierwsza maszyna typu NUMA: **Cm*** (Jones, 1977)

Każde grono składa się z CPU, mikroprogramowalnego MMU, modułu pamięci, ew. urządzeń we-wy połączonych szyną. Nie ma pamięci podręcznej ani nasłuchiwania szyny. Maszyna składa się z wielu gron połączonych szyną

MMU realizuje żądanie do lokalnej pamięci w tradycyjny sposób. Żądanie do zdalnej pamięci zamienia na pakiet i wysyła szyną do zdalnego MMU

W maszynach typu **UMA** (**U**niform **M**emory **A**ccess) lokalizacja strony nie ma kluczowego znaczenia (bo dzięki buforowaniu strona i tak zostanie przeniesiona tam, gdzie jest potrzebna)

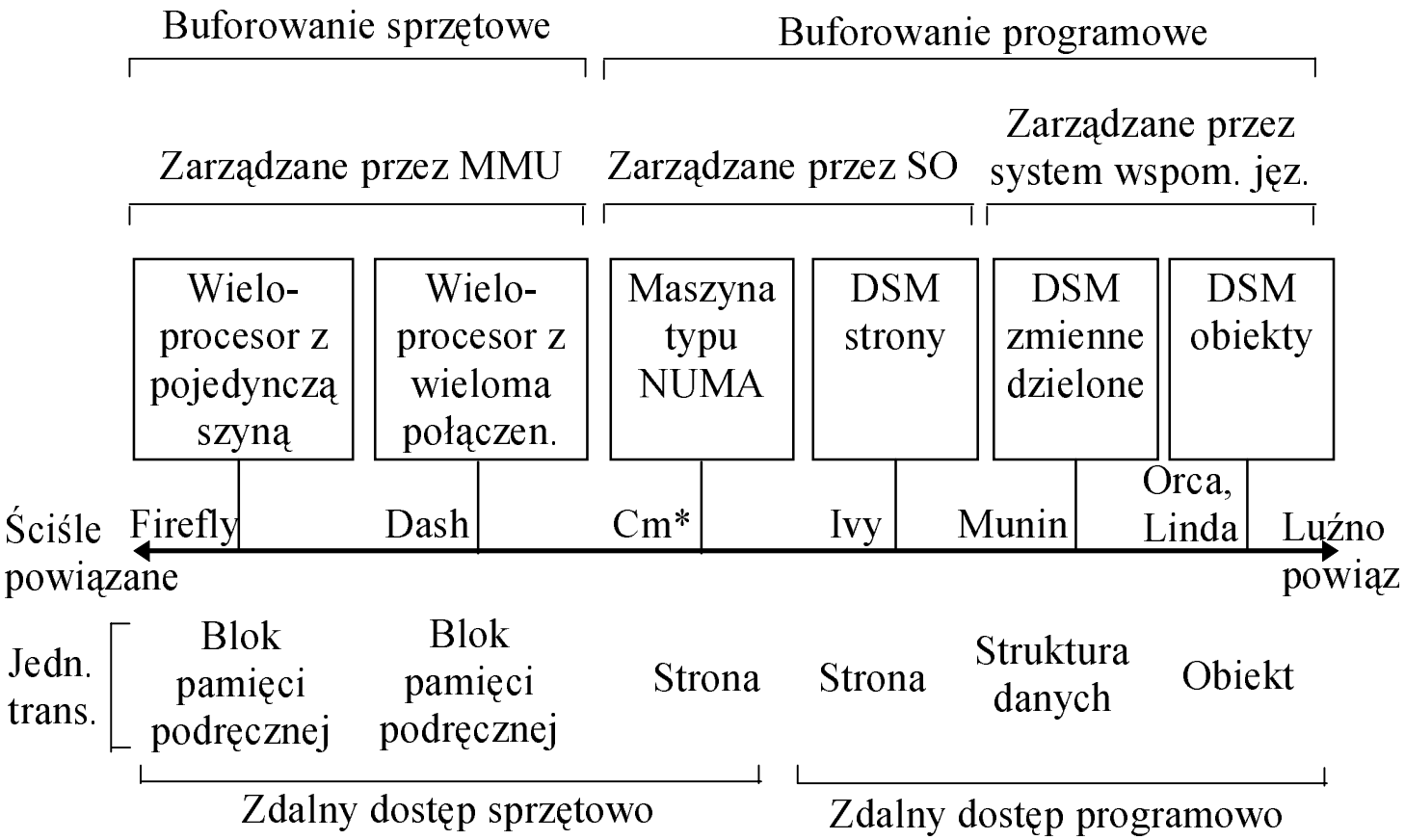
W maszynach typu NUMA lokalizacja ma decydujące znaczenie dla wydajności systemu:

- Strony mogą mieć z góry przypisane położenie
- Jeśli CPU odwołuje się do strony, która nie jest odwzorowana do jego przestrzeni adresowej, to jest generowany błąd braku strony; SO może wówczas podjąć decyzję o:
  - utworzeniu kopii strony lub odwzorowaniu jej do pamięci zdalnej (strona tylko do czytania)
  - o przesłaniu strony do procesora zgłaszającego żądanie lub odwzorowaniu jej do pamięci zdalnej (strona do czyt. i pisania)

  Niezależnie od przyjętego rozwiązania kolejne odwołania do tej strony odbywają się sprzętowo

- Zwykle specjalny proces demon (*page scanner*) okresowo zbiera statystyki o odwołaniach lokalnych i zdalnych i ew. modyfikuje wcześniejsze decyzje. Może także „zamrozić” lokalizację strony na jakiś czas

Porównanie systemów pamięci dzielonej



1. Wieloprocesor z pojedynczą szyną: obsługa pamięci dzielonej realizowana całkowicie sprzętowo
2. Wieloprocesor z wieloma połączeniami: sprzętowe buforowanie, ale programowe struktury danych przechowują informacje o położeniu buforowanych bloków. Zgodność zachowuje się dzięki stosowaniu złożonych algorytmów, zwykle realizowanych prez mikrokod MMU
3. Maszyny typu NUMA: rozwiązanie hybrydowe. CPU może czytać/pisać z/do wspólnej wirtualnej przestrzeni adresowej, ale buforowanie (kopiowanie i migracja stron) jest kontrolowane programowo

4. DSM - strony: CPU nie może bezpośrednio sięgać do pamięci zdalnej; obsługa błędów braku zdalnej strony odbywa się programowo (SO)
5. DSM - zmienne dzielone: nie ma pojedynczej pamięci wspólnej, informacje o dzielonych strukturach danych dostarcza użytkownik,
6. DSM - obiekty: zdalny dostęp tylko poprzez chronione metody (ułatwia zachowanie zgodności obiektów), wszystko realizowane programowo

Podsumowanie

1. Liniowa, dzielona wirtualna przestrzeń adresowa?
2. Możliwe operacje
3. Kapsułkowanie i metody?
4. Czy zdalny dostęp jest możliwy w sprzęcie?
5. Kto zamienia zdalne żądania dostępu do pamięci w komunikaty?
6. Środek transmisji
7. Kto realizuje migrację danych?
8. Jednostka transmisji

	Pojedyn-cza szyna	Wiele połączeń	NUMA	DSM strona	DSM zm. dziel.	DSM obiekt
1	Tak	Tak	Tak	Tak	Nie	Nie
2	Czyt/Pis	Czyt/Pis	Czyt/Pis	Czyt/Pis	Czyt/Pis	Ogólne
3	Nie	Nie	Nie	Nie	Nie	Tak
4	Tak	Tak	Tak	Nie	Nie	Nie
5	MMU	MMU	MMU	SO	System wspom. j.	System wspom. j.
6	Szyna	Szyna	Szyna	Sieć	Sieć	Sieć
7	Sprzęt	Sprzęt	Oprogr.	Oprogr.	Oprogr.	Oprogr.
8	Blok	Blok	Strona	Strona	Zmienna dzielona	Obiekt