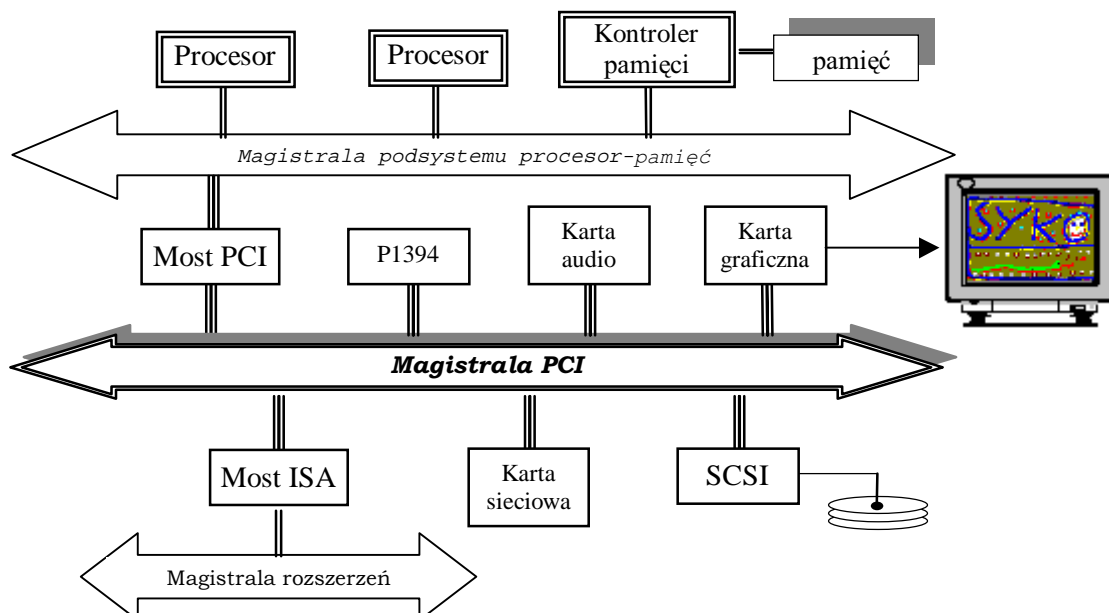


Magistrale systemowe: Magistrala PCI

według T. Jamrógowicza

Magistrala jest - wspólnym dla kilku urządzeń - zestawem połączeń przeznaczonych do przesyłania sygnałów, nadawanych przez jedno z kilku możliwych źródeł, do jednego lub kilku miejsc przeznaczenia. Dostęp do magistrali zwykle dzielony jest w czasie pomiędzy kilka urządzeń. Jeśli wysyłane sygnały mają być poprawnie odebrane to, w danej chwili, tylko jedno urządzenie może sterować liniami magistrali. Natomiast transmitowane dane są potencjalnie dostępne dla wszystkich pozostałych urządzeń dołączonych do magistrali. W systemie komputerowym magistrale łączą układy usytuowane na różnych szczeblach systemowej hierarchii. Kluczowe cechy, które je różnią i pozwalają klasyfikować to:

- | | |
|--|--|
| Sposób wykorzystania linii magistrali | - linie dedykowane lub przełączane. |
| Szerokość ścieżki danych | - liczba równoległych linii umożliwiających jednoczesną transmisję bitów danych. |
| Taktowanie | - synchroniczne lub asynchroniczne. |
| Rodzaje operacji przekazu danych | - zapis, odczyt, odczyt-modyfikacja-zapis, odczyt kontrolny, przekaz blokowy. |
| Arbitraż | - centralny lub rozproszony. |



Rysunek 1. Przykład systemu komputerowego z magistralą PCI.

Dla współczesnych systemów komputerowych ważną magistralą jest PCI (*Peripheral Component Interconnect*). Jest to lokalna magistrala, niezależna od typu procesora centralnego, o dużej przepustowości, z centralnym arbitrażem i synchronicznym taktowaniem transmisji danych. Do przesyłania adresu i danych wykorzystywane są te same linie magistrali, przełączane odpowiednio w zależności od fazy operacji.

1. OPIS STANDARDU

Norma PCI Local Bus (ostatnia wersja: Revision 2.2) wydana przez PCI Special Interest Group, definiuje 32-bitową (z możliwością rozszerzenia do 64 bitów) magistralę, ze wspólnymi przełączanymi liniami adresu i danych, synchronizowaną przebiegiem zegarowym o częstotliwości do 33 MHz (z możliwością rozszerzenia o dodatkową częstotliwość zegara - 66 MHz). W protokole transmisji danych wykorzystano mechanizm przesyłania sekwencyjnego. Przy częstotliwości zegara 33 MHz, magistralą PCI można transmitować 32-bitowe dane z szybkością do 132 Megabajtów/sekundę. 64-bitowe rozszerzenie pozwala podwoić przepustowość magistrali. W normie zakłada się, że bezpośredni styk z magistralą obsługiwany jest przez wyspecjalizowany układ scalonych wielkiej skali integracji. Określone normą parametry elektryczne odnoszą się do tego układu. Nie przewidziano dodatkowego buforowania. Magistrala nie jest zakończona terminatorami. Układy sterujące dostarczają jedynie połowę napięcia w stanie wysokim lub niskim; druga połowa uzyskiwana jest z fali odbitej od końca magistrali.

PCI usytuowana jest zwykle pomiędzy magistralą podsystemu procesor - pamięć - pamięć podręczna a magistralą rozszerzeń (ISA, EISA, MicroChannel). Od bezpośredniego otoczenia procesora oddziela ją most-sterownik magistrali PCI.

Most PCI może dodatkowo pełnić funkcję bufora danych oraz arbitra. Jeśli procesor przesyła dane do jednego z urządzeń peryferyjnych - takich jak np. kontroler dyskowy - to są one natychmiast zapamiętywane w pamięci buforowej mostu-sterownika magistrali. Pozwala to jednostce centralnej przejść do wykonywania następnych operacji nie czekając na zakończenie zapisu w pamięci dyskowej. Za pośrednictwem magistrali system komputerowy może być rozszerzany o inteligentne urządzenia, które mogą realizować swoje zadania niezależnie od jednostki centralnej systemu. Moduły te mogą przejmować kontrolę nad magistralą i inicjować bezpośrednią wymianę informacji z tymi samymi urządzeniami we-wy, z których korzysta jednostka centralna. Urządzenia skupione wokół magistrali PCI mogą współpracować ze sobą niezależnie od podsystemu procesor centralny - pamięć - pamięć podręczna. Charakteryzując magistralę PCI jako niezależną od typu procesora rozumiemy przez to, że karta kontrolera urządzenia we-wy - wykonana zgodnie z normą - może być użyta na magistrali PCI zarówno w stacji roboczej sterowanej procesorem Alpha jak i Pentium lub PowerPC. Warunkiem jest zainstalowanie odpowiedniego oprogramowania.

Istotną zaletą systemu PCI jest zdolność do automatycznej konfiguracji dołączanych urządzeń. Po włożeniu nowej karty w gniazdo rozszerzeń i włączeniu systemu następuje automatyczna konfiguracja przerwań, kanałów DMA, adresów I/O itd. Nie trzeba ustawiać zworek, przełączników, zegarów. Pozwala to systemowi operacyjnemu eliminować ewentualne konflikty adresów, numerów przerwań a także błędy niedopasowania prędkości działania współpracujących ze sobą urządzeń. Od urządzeń standardu PCI wymaga się - by w specjalnie do tego celu wydzielonej przestrzeni adresowej - udostępniały swoje charakterystyki i rejestry umożliwiające programowe ustawianie ich konfiguracji i zainicjowanie pracy.

2. OMÓWIENIE WAŻNIEJSZYCH TERMINÓW

Agent jednostka podejmująca działanie na magistrali komputera (ang. agent).

Blokowanie dostępu mechanizm umożliwiający modułowi nadrzędnemu zamknięcie, przed innymi agentami, dostępu do wspólnej pamięci na czas wykonywania sekwencji rozkazów, której wykonanie raz rozpoczęte musi zostać doprowadzone do końca (ang. locking).

Moduł docelowy urządzenie, które współdziała w przeprowadzeniu operacji zainicjowanej przez moduł nadrzędny: sygnalizuje, że zostało zaadresowane i potwierdza gotowość zakończenia każdej z faz transmisji danych (ang. target).

Moduł nadrzędny urządzenie, które inicjuje operacje przesyłania danych: steruje fazą adresową, wyznacza czas trwania operacji, potwierdza gotowość zakończenia każdej z faz transmisji danych (ang. master).

Most układ logiczny umożliwiający połączenie dwóch magistral tak, żeby agent jednej magistrali mógł mieć dostęp do agenta drugiej magistrali (ang. bridge).

Pamięć podręczna bardzo szybka pamięć buforowa wykorzystywana do przechowywania duplikatów aktualnie najczęściej adresowanych fragmentów pamięci głównej; jeśli procesor żąda danych, których kopie są dostępne w pamięci podręcznej, to żądanie takie zostaje przez nią przechwycone i szybko zrealizowane (ang. cache).

Przesyłanie sekwencyjne podstawowy dla PCI tryb transmisji danych: po jednej fazie przesłania adresu początkowego w następnych fazach przekazu danych otwierany jest dostęp do zaadresowanej i kolejnych komórek pamięci - już bez podawania ich adresów (ang. burst transfer).

Spójność pamięci zgodność zawartości pamięci głównej i podręcznej; jej **podręcznej** utrata może wystąpić przy dostępie do pamięci głównej **(z pamięcią** innych - niż procesor centralny - urządzeń (ang. **operacyjną)** cache coherency).

Śledzenie prowadzona przez podsystem pamięci podręcznej **dostępów** kontrola adresów wystawianych (przy dostępie do **(do pamięci** fragmentu pamięci operacyjnej w module docelowym PCI) **operacyjnej)** przez inne niż procesor centralny urządzenie: jeśli może dojść do konfliktu danych to moduł docelowy zostaje poinformowany by zawiesił dostęp do tych danych do czasu usunięcia niespójności (ang. bus snooping).

Wiersz liczba bajtów - stała dla danego kontrolera pamięci **(pamięci** podręcznej - przesyłana jednorazowo podczas wymiany **podręcznej)** zawartości z pamięcią główną (ang. line (cache)).

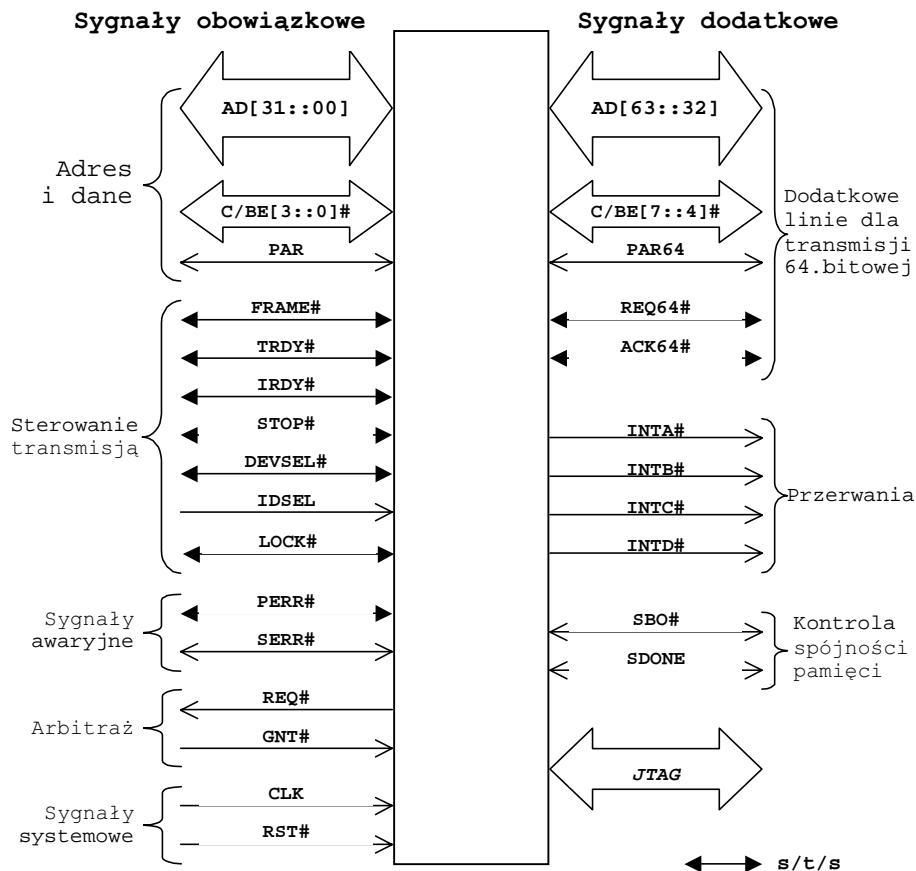
3. SYGNAŁY MAGISTRALI PCI

Magistrala PCI może być wykorzystywana jako magistrala 32 lub 64.bitowa. Sygnały obowiązkowe są wspólne dla obu konfiguracji magistrali. Wśród sygnałów dodatkowych są wspólne dla obu konfiguracji i specyficzne dla rozszerzenia 64.bitowego.

3.1 Oznaczenia wykorzystane w opisie sygnałów

In	Sygnał tylko wejściowy.
Out	Sygnał wyjściowy typu "Totem Pole".
t/s	Sygnał wejściowo-wyjściowy trzystanowy.
s/t/s	Trójstanowy sygnał podtrzymywany. Sygnał ten, aktywny w stanie niskim, może przejść ze stanu niskiego do stanu wysokiej impedancji jedynie poprzez aktywny stan wysoki, trwający co najmniej jeden impuls zegarowy. Linia takiego sygnału w danej chwili może sterować tylko jeden agent. Nowy agent może ustawić taki sygnał w stan niski nie wcześniej, niż w następnym impulsie zegarowym, po tym jak poprzedni agent pozostawił go w stanie wysokiej impedancji. W stanie biernym konieczne jest podtrzymanie stanu linii przez opornik dołączony do napięcia zasilającego.
o/d	Sygnał typu otwarty dren umożliwiający wielu urządzeniom jednoczesne sterowanie linią zgodnie z funkcją sumy galwanicznej.
#	Symbol # umieszczony na końcu nazwy sygnału oznacza, że jest to sygnał aktywny w stanie niskim

Na rysunku 2 przedstawiono wyprowadzenia sygnałów wykorzystywanych na magistrali PCI. Moduł nadrzędny wykorzystuje co najmniej 49 styków złącza, a moduł docelowy co najmniej 47 styków (pamięć 48).



Rysunek 2. Zestawienie sygnałów magistrali PCI

3.2 Sygnały obowiązkowe

Według pełnionych funkcji można je zaliczyć do jednej z grup:

- Sygnały systemowe – zegar systemowy i sygnał ogólnego zerowania.
- Adres i dane (wspólne linie – przełączane w zależności od fazy operacji magistralowej) wraz z sygnałami kodu operacji lub selekcji (wspólne linie) i sygnałem kontroli parzystości.
- Sygnały sterowania transmisją danych.
- Sygnały arbitrażu.
- Sygnalizacja sytuacji awaryjnych.

Spośród sygnałów sterowania transmisją danych obsługa sygnału **LOCK#** jest wymagana jedynie od urządzeń zawierających fragment pamięci operacyjnej. Sygnały arbitrażu są obowiązkowymi jedynie dla modułów nadrzędnych.

RST#	in	Sygnał ogólnego zerowania. Zeruje rejestry, liczniki oraz wskaźniki nadzorujące kolejność wykonywanych operacji. Działa asynchronicznie niezależnie od sygnału zegara systemowego.
IDSEL	in	Sygnał wykorzystywany do adresowania urządzeń PCI podczas operacji odczytu i zapisu rejestrów konfiguracyjnych.
CLK	in	Sygnał zegarowy synchronizujący wszystkie operacje na magistrali PCI. Sygnał jest doprowadzony do wszystkich stanowisk. Wszystkie sygnały z wyjątkiem RST# , INTA# , INTB# , INTC# oraz INTD# są próbkowane podczas narastającego zbocza sygnału CLK . W każdej chwili częstotliwość zegara może być (ale tylko w stanie niskim) zmieniona w zakresie od 0 do 33MHz (zegar może być zatrzymany przy przejściu na pracę krokową). Moduły PCI powinny być dostosowane do zmian częstotliwości zegara.
REQ#	t/s	Sygnał żądania przydziału magistrali wysyłany przez moduł nadrzędny do układu arbitra.
GNT#	t/s	Wysyłany przez arbitra sygnał potwierdzający przyznanie modułowi nadrzędnemu prawa do magistrali. Każdy moduł nadrzędny ma swój własny sygnał GNT# .
AD[31:00]	t/s	Przełączane linie sygnałów adresowych i danych. Po każdej fazie przesyłania adresu następuje jedna lub więcej faz przesyłania danych.
C/BE[3:0]#	t/s	Linie multipleksowane. W fazie przesyłania adresu przekazują kod rozkazu. W fazie przesyłania danych wykorzystywane są jako sygnały wyznaczające ścieżkę, którą przesyłane są ważne bajty [3-0] danych.
PAR	t/s	Bit parzystości (parzystej dla stanów wysokich), dotyczy wszystkich bitów przesyłanych liniami AD[31:00] i C/BE[3:0]# , ważny w zboczu impulsu zegarowego następującego po każdym cyklu fazy przesyłania adresu lub kończącym fazę przesyłania danych; wystawiany przez moduł wysyłający adres lub dane.
FRAME#	s/t/s	Sygnał generowany przez moduł nadrzędny, który ma prawo sterować magistralą. Określa moment rozpoczęcia i czas trwania operacji; aktywny w stanie niskim przed ostatnią fazą przesyłania danych sygnał FRAME# przechodzi w stan wysoki.

IRDY#	s/t/s	Sygnalizuje, że agent inicjujący operację jest gotowy zakończyć bieżącą fazę przesyłania danych. Podczas operacji zapisu sygnał IRDY# wskazuje, że dane wystawione na liniach AD[31:00] już są ważne. Podczas operacji odczytu sygnalizuje gotowość modułu nadrzędnego do odbioru informacji. Faza jest kończona, kiedy dla tego samego narastającego zbocza impulsu zegarowego sygnały IRDY# i TRDY# są aktywne.
TRDY#	s/t/s	Sygnał wysyłany przez moduł docelowy uczestniczący w operacji transmisji danych. Za jego pośrednictwem moduł informuje, że jest gotowy zakończyć bieżącą fazę przesyłania danych. Podczas operacji odczytu aktywny sygnał TRDY# oznacza, że dane na liniach AD[31:00] są ważne. Jeśli sygnał TRDY# jest aktywny podczas operacji zapisu to wiadomo, że moduł docelowy jest gotowy przyjąć dane.
DEVSEL#	s/t/s	Sygnał wysyłany przez ten moduł docelowy, który po zdekodowaniu adresu poczuwa się do roli adresata zainicjowanej operacji.
PAR	t/s	Bit kontroli parzystości (parzystej dla stanów wysokich) obejmującej bity przesyłane wszystkimi liniami AD[31:00] i C/BE[3:0]# , ważny w zboczu impulsu zegarowego następującego po każdym cyklu fazy przesyłania adresu lub kończącym fazę przesyłania danych. Moduł nadrzędny wystawia sygnał PAR po fazie przesyłania adresu i przy zapisie danych. Moduł docelowy wystawia sygnał z bitem PAR przy odczycie danych.
STOP#	s/t/s	Sygnał - z modułu docelowego - żądania zaprzestania wykonywania przez bok nadrzędny bieżącej operacji.
PERR#	s/t/s	Sygnalizuje, że podczas jednej z operacji za wyjątkiem cyklu specjalnego, pojawił się błąd parzystości.
SERR#	o/d	Sygnalizuje wystąpienie błędu systemowego, kiedy pojawił się błąd parzystości w fazie przesyłania adresu lub danych podczas cyklu specjalnego lub też pojawił się inny błąd, który może spowodować katastrofalne skutki. Utrzymywany jest w stanie niskim na czas jednego impulsu zegarowego.

LOCK#	s/t/s	Sygnał wykorzystywany przez moduł nadrzędny, podczas wykonywania operacji nierozdzielnych, do blokowania dostępu (innym agentom) do fragmentu pamięci w bloku docelowym. Moduł docelowy pozostaje zablokowany do momentu, kiedy oba sygnały FRAME# i LOCK# nie znajdują się w stanie wysokim. Mimo, że moduł nadrzędny utrzymuje sygnał LOCK# w stanie aktywnym, inne moduły nadrzędne mogą uzyskać kontrolę nad magistralą i wykonywać operacje (z wyjątkiem wyłącznych), które nie dotyczą zablokowanego fragmentu pamięci. Tak więc przydział magistrali nie jest równoważny z przyznaniem kontroli nad sygnałem LOCK# .
--------------	--------------	---

3.3 Dodatkowe sygnały wspólne dla magistrali 32 i 64 bitowej

INT[A::D]#	o/d	Niski poziom sygnału (odbieranego przez kontroler przerw) oznacza żądanie przerwania. Urządzenia, w których można wyodrębnić różne moduły funkcjonalne mogą wykorzystywać do czterech linii przerw. Urządzenia z jednym modulem funkcjonalnym mogą korzystać jedynie z linii INTA# .
SDONE	in/out	Sygnał pochodzący od kontrolera pamięci podręcznej śledzącego dostępy do pamięci operacyjnej systemu, wejściowy dla modułu docelowego, zawierającego pamięć objętą buforowaniem. Aktywny SDONE (w stanie wysokim) sygnalizuje zakończenie weryfikacji. Wynik określa sygnał SBO# : w stanie wysokim informuje, że dostęp do pamięci nie naruszy spójności pamięci. W przeciwnej sytuacji moduł docelowy powinien zasignalizować modułowi nadrzêdnemu konieczność powtórzenia operacji.
SBO#	in/out	Sygnał pochodzący od układu kontrolującego zgodność zawartości pamięci podręcznej z zawartością buforowanych obszarów pamięci operacyjnej, wejściowy dla układów pamięci na magistrali PCI buforowanych przez pamięć podręczną. Ustawiony przez most w stan niski równocześnie z sygnałem SDONE (stan wysoki) oznacza, że skontrolowany dostęp do pamięci na magistrali PCI jest związany z odczytem lub modyfikowaniem danych, które są nieaktualne: wiersz pamięci podręcznej, w którym znajduje się ich duplikat, został zmodyfikowany.

3.4 Sygnały związane z magistralą 64 bitowa

AD[63::32]	t/s	Multipleksowane sygnały adresów i danych. W fazie przesyłania adresu nie przesyłają ważnej informacji. W fazie przesyłania danych, kiedy sygnały REQ64# i ACK64# są w stanie niskim, po liniach tych transmitowane są dodatkowe 32 bity danych.
C/BE[7::4]#	t/s	Sygnały multipleksowane. W fazie przesyłania adresu nie niosą ważnej informacji. Jeśli w fazie przesyłania danych sygnały REQ64# i ACK64# są w stanie niskim, linie zezwolenia określają które z bajtów [7-4] przenoszą ważną informację
PAR64	t/s	Sygnał kontroli parzystości dla linii AD[63::32] i C/BE[7::4]# .
REQ64#	s/t/s	Sygnał żądania 64.bitowej transmisji danych, wysyłany przez moduł nadrzędny równocześnie z sygnałem FRAME# .
ACK64#	s/t/s	Sygnał potwierdzenia 64.bitowej transmisji danych, wysyłany przez moduł docelowy równocześnie z sygnałem DEVSEL# .

Dodatkowe sygnały JTAG (TCK, TDI, TDO, TMS, TRST#) są wykorzystywane do testowania podzespołów urządzeń PCI zgodnie z zaleceniami standardu IEEE 1149.1.

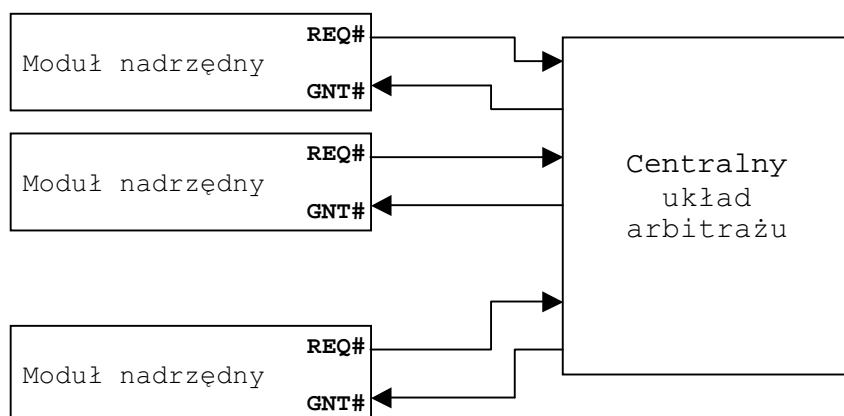
4. OPERACJE NA MAGISTRALI PCI

4.1 Arbitraż

W systemie wieloprocesorowym każdy z procesorów chcąc uzyskać dostęp do wspólnej magistrali powinien upewnić się, że w swym działaniu nie zderzy się z innym procesorem. Ponieważ może się zdarzyć, że kilka procesorów zechce wykorzystać magistralę w tym samym czasie, potrzebny jest mechanizm rozstrzygania o wyniku współzawodnictwa między procesorami zabiegającymi o dostęp do jednej wspólnej magistrali czyli tzw. arbitraż.

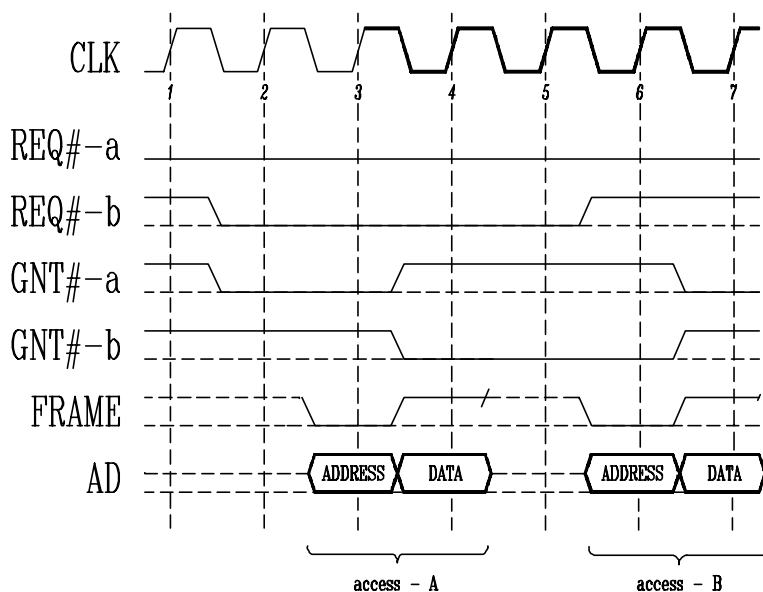
W standardzie PCI rozwiązano to przy pomocy jednego centralnego układu arbitrażu, z którym każdy moduł nadrzędny łączy się indywidualnymi liniami sygnałów **REQ#** i **GNT#**. Agent pragnący zainicjować na magistrali PCI operację transmisji danych (moduł

nadrzędny) wystawia sygnał **REQ#**. Do działania może przystąpić dopiero po odebraniu (z układu centralnego arbitrażu) sygnału **GNT#**.



Rysunek 3. Arbitraż dostępu do magistrali PCI.

Norma PCI nie definiuje algorytmu, według którego arbiter ma wskazać agenta uzyskującego przydział magistrali. Może to być zasada „kto pierwszy ten lepszy”, priorytet cykliczny lub inny sposób rozstrzygnięcia o dostępie do magistrali.



Rysunek 4. Zależności czasowe podczas arbitrażu

Na rysunku 4 przedstawiono przebiegi czasowe podczas arbitrażu. Agenci A i B ubiegają się o przydział magistrali. Agent A wcześniej zażądał magistrali i ma sygnał **REQ#-a** ustawiony w stan niski. Podczas narastającego zbocza 1. impulsu zegarowego sygnał został odczytany przez arbitra. Przed przyjściem drugiego impulsu

zegarowego arbiter, przez ustawienie sygnału **GNT#**a (w stan niski), przyznał agentowi A dostęp do magistrali. Agent A może zainicjować operację transmisji danych, ponieważ magistrala jest w stanie jałowym (sygnały **FRAME#** i **IRDY#** są w stanie wysokim) a sygnał **GNT#** jest w stanie niskim. Operacja rozpocznie się podczas impulsu zegarowego 3, kiedy **FRAME#** znajdzie się w stanie niskim.

Ponieważ agent A zamierza przeprowadzić następne operacje - pozostawia sygnał **REQ#**a w stanie aktywnym. Podczas 3. impulsu zegarowego kiedy sygnał **FRAME#** jest w stanie niskim, arbiter przydziela magistralę agentowi B, w związku z tym podczas impulsu zegarowego 4 sygnał **GNT#**b przechodzi w stan niski, a sygnał **GNT#**a w stan wysoki. Agent A kończy rozpoczętą operację podczas impulsu zegarowego 4 i zwalnia magistralę. Agent B wykrywa koniec operacji analizując sygnały **FRAME#** i **IRDY#**. Operacja jest zakończona, kiedy oba te sygnały są w stanie wysokim. Agent B przejmuje kontrolę nad magistralą podczas impulsu zegarowego 5, kiedy **FRAME#** i **IRDY#** są w stanie wysokim. Operacja prowadzona przez agenta B zostanie zakończona podczas impulsu zegarowego 7. Podczas impulsu zegarowego 6 sygnał **REQ#**b jest w stanie wysokim a sygnał **FRAME#** w stanie niskim. Oznacza to, że agent B potrzebował magistrali dla przeprowadzenia jednej operacji. W następnym kroku, jeśli nie będzie innych zgłoszeń, arbiter ponownie przydzieli magistralę agentowi A ponieważ jego sygnał **REQ#**a pozostaje cały czas w stanie niskim. Agent, który przejął kontrolę nad magistralą i zamierza przeprowadzić więcej niż jedną operację, utrzymuje sygnał **REQ#** w stanie niskim do czasu zakończenia ostatniej operacji. Ponowny przydział magistrali nastąpi w wypadku, kiedy inni agenci nie starają się o przydział magistrali lub dany agent ma wyższy priorytet niż pozostali agenci.

Sygnał **GNT#** w stanie niskim, wysyłany przez arbitra, przydziela magistralę tylko na czas wykonania jednej operacji.

Jeśli żaden inny agent nie korzysta z magistrali lub też inni agenci nie żądają jej przydziału arbiter może jednemu z agentów przydzielić magistralę na czas nieokreślony. Może to być moduł nadrzędny, który ostatnio korzystał z magistrali lub moduł wytypowany domyślnie. Mówi się wówczas, że magistrala została zaparkowana (ang. parked) przy danym agencie. Tak długo jak trawa ta sytuacja (sygnał **GNT#** na złączu agenta w stanie niskim), rozpoczęcie przez niego pojedynczej operacji nie musi być poprzedzane przejściem jego sygnału **REQ#** w stan niski.

4.2 Protokół transmisji danych

Magistrala przechodzi ze stanu jałowego w stan aktywny na czas wykonania operacji wymiany informacji pomiędzy modułami nadrzędnym i docelowym. Inicjatorem wymiany informacji jest moduł nadrzędny, który w wyniku przeprowadzonej procedury arbitrażu uzyskał dostęp do magistrali.

Podstawową operacją na magistrali PCI jest transmisja sekwencyjna w przestrzeni adresowej pamięci. Składa się ona z jednej fazy przesyłania adresu (za wyjątkiem podwójnego cyklu adresowego) oraz kilku faz przesyłania danych. Norma PCI dopuszcza przesyłanie sekwencyjne również w przestrzeni adresowej urządzeń we-wy.

Wszystkie sygnały (za wyjątkiem działających asynchronicznie ogólnego zerowania i zgłoszeń przerwań) są próbkowane podczas narastającego zbocza impulsu zegarowego. W tym momencie stany nieustalone sygnałów są niedopuszczalne. Poza tym zboczem stany nieustalone są ignorowane.

Podstawowe operacje na magistrali PCI są kontrolowane przez trzy sygnały:

FRAME# wysyłany przez moduł nadrzędny sygnał, który wyznacza czas trwania operacji.

IRDY# sygnał wysyłany przez moduł nadrzędny w celu zakończenia pojedynczej fazy przesyłania danych.

TRDY# sygnał wysyłany przez moduł docelowy w celu zakończenia pojedynczej fazy transmisji danych.

Magistrala pozostaje w stanie jałowym jak długo sygnały **FRAME#** i **IRDY#** są nieaktywne. Pierwsze narastające zbocze impulsu zegarowego z aktywnym sygnałem **FRAME#** (stan niski) rozpoczyna fazę przesyłania adresu (po liniach **AD[31:0]**). W tym czasie liniami **C/BE[3:0]** przesyłany jest kod rozkazu.

Następne narastające zbocze **CLK** rozpoczyna jedną lub sekwencję kilku faz przesyłania danych pomiędzy modułami nadrzędnym i docelowym. Za każdym razem kiedy oba sygnały **TRDY#** i **IRDY#** znajdują się w stanie aktywnym jest kończona jedna faza przesyłania danych. Jak długo jeden z sygnałów **IRDY#** lub **TRDY#** jest w stanie wysokim, moduł nadrzędny lub docelowy wprowadza cykle oczekiwania.

Agent wysyłający dane powinien bezwarunkowo ustawić sygnał **xRDY#** w stan niski, kiedy dane są ważne (sygnał **IRDY#** dla cyklu zapisu oraz **TRDY#** dla cyklu odczytu). Agent odbierający dane ustawia sygnał **xRDY#** w stan niski lub wysoki, zależnie od tego, czy jest gotowy dane przyjąć czy też nie.

Moduł nadrzędny po ustawieniu sygnału **IRDY#** w stan niski, nie może zmieniać sygnałów **IRDY#** oraz **FRAME#**, aż zostanie zakończona bieżąca faza przesyłania danych, niezależnie od stanu sygnału **TRDY#**.

Moduł nadrzędny odbierający dane, potwierdza przyjęcie kolejnej porcji danych po każdej fazie przesyłania danych, tak długo jak sygnał **FRAME#** pozostaje w stanie niskim, przez ustawienie sygnału **IRDY#** ze stanu wysokiego w stan niski. Przejście sygnału **FRAME#** do stanu wysokiego oznacza, że bieżąca transmisja danych jest ostatnią.

4.3 Rozkazy magistrali PCI

Przesyłany w pierwszej fazie operacji transmisji danych (fazie adresowej), po liniach **C/BE[3::0]**, kod operacji wskazuje przestrzeń adresową, do której odnosi się wystawiony równolegle na liniach **AD[31::00]** adres. Rozkaz może też inicjować jedną z dwóch operacji bezadresowych. Rozróżnia się trzy przestrzenie adresowe:

- i. danych,
- ii. we-wy,
- iii. konfiguracyjną.

Zestawienie kodów i odpowiadających im typów operacji przedstawiono w tabeli:

	C/BE[3::0] #	Typ operacji	Komentarz
1.	0000	Potwierdzenie przerwania	Bezadresowa, odczyt
2.	0001	Cykl specjalny	Bezadresowa, zapis
3.	0010	Operacja czytaj z we-wy	We-wy
4.	0011	Operacja zapisz we-wy	We-wy
5.	0100	Zarezerwowane	-
6.	0101	Zarezerwowane	-
7.	0110	Czytaj z pamięci	Pamięć
8.	0111	Zapisz w pamięci	Pamięć
9.	1000	Zarezerwowane	-
10.	1001	Zarezerwowane	-
11.	1010	Czytaj konfigurację	Konfiguracja
12.	1011	Zapisz konfigurację	Konfiguracja
13.	1100	Wielokrotny odczyt z pamięci	Pamięć
14.	1101	Podwójny cykl adresowy (64 bity)	Pamięć
15.	1110	Czytaj wiersz pamięci podręcznej	Pamięć
16.	1111	Zapisz w pamięci i unieważnij	Pamięć

Potwierdzenie przerwania jest poleceniem odczytu wektora przerwania - skierowanym do urządzenia, które na magistrali PCI pełni rolę kontrolera przerwania. Ponieważ tylko jeden agent może odpowiadać na rozkaz **potwierdzenia przerwania** (kontroler przerwania) rozkaz ten nie wymaga przesyłania adresu. Linie adresowe **AD[31::00]** nie są podczas tej operacji wykorzystywane i ich stan w fazie przesyłania adresu jest nieokreślony (ale musi być stabilny). Kontroler przerwania potwierdza sygnałem **DEVSEL#**, że odebrał polecenie. W fazie transmisji danych linie **C/BE[3::0]#** określają z ilu bajtów składa się odczytywany wektor przerwania.

Rozkazy typu **cykl specjalny** są wykorzystywane do przesyłania komunikatów (np. „rozpoczęto procedurę zamknięcia systemu” lub „nastąpiło zatrzymanie”) adresowanych do wszystkich modułów (moduły nie potwierdzają tego sygnałem **DEVSEL#**). W fazie adresowej stan linii adresowych jest nieokreślony i powinien zostać zignorowany.

Rozkazy **czytaj z we-wy** i **zapisz we-wy** są wykorzystywane przy dostępie do rejestrów urządzeń we-wy. W fazie adresowej po liniach **AD[31::00]** (wszystkie muszą być dekodowane) przesyłany jest adres wskazujący, z dokładnością co do bajtu, miejsce w przestrzeni adresowej odwzorowujące rejestr urządzenia we-wy.

Rozkazy **czytaj z pamięci** i **zapisz do pamięci** służą realizacji dostępu do pamięci przy przesyłaniu niewielkich porcji danych (najlepiej jeśli długość danych nie przekracza połowy liczby bajtów w jednym wierszu pamięci podręcznej).

Rozkazy typu **czytaj konfigurację** i **zapisz konfigurację** są wykorzystywane odczytywania i zapisu informacji w przestrzeni adresowej (wybranego agenta) zarezerwowanej dla potrzeb konfiguracji.

Wielokrotny odczyt pamięci działa podobnie jak rozkaz **czytaj z pamięci**; moduł nadrzędny informuje tu dodatkowo o przystąpieniu do odczytu sekwencyjnego większej liczby bajtów - na ogół ponad to co da się skopiować w trzy wiersze pamięci podręcznej.

Rozkazy typu **podwójny cykl adresowy** są stosowane w wypadku korzystania z adresacji 64.bitowej.

Rozkazy typu **czytaj wiersz pamięci**, podobne w działaniu do rozkazów typu **czytaj z pamięci** są stosowane przy seryjnym odczycie danych jeśli liczba bajtów mieści się w przedziale od połowy do trzech wierszy pamięci podręcznej.

Rozkazy typu **zapisz w pamięci i unieważnij** są podobne do rozkazów typu **zapisz w pamięci** z tym, że zagwarantowane jest przesłanie kompletu bajtów odwzorowywanych w jednym lub kilku wierszach pamięci podręcznej. Moduł nadrzędny zapisuje wszystkie bajty podczas jednej operacji. Wykonanie tego rozkazu wymaga sprawdzenia

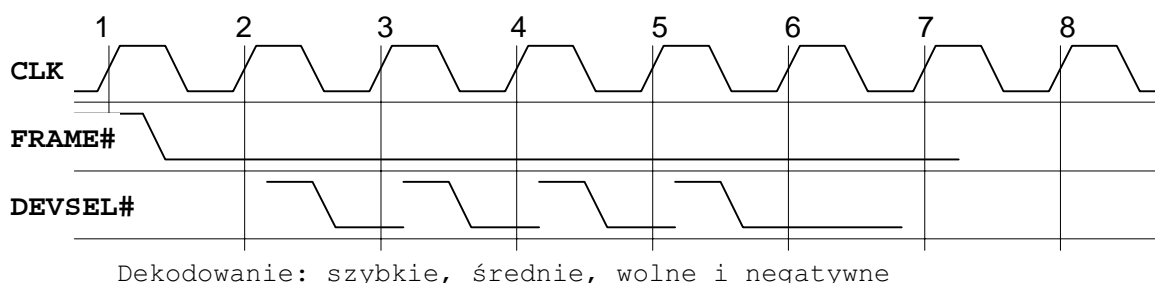
w rejestrze konfiguracyjnym bloku nadrzędnego rozmiaru wiersza pamięci podręcznej.

Wszystkie bloki muszą odpowiadać na rozkazy typu czytaj lub zapis konfigurację. Pozostałe rozkazy nie są obowiązkowe.

4.4 Adresowanie

Każde docelowe urządzenie PCI ma obowiązek dekodować przesyłane magistralą adresy. Jeśli rozpozna jeden ze swoich adresów to odpowiada sygnałem **DEVSEL#**. Sygnał **TRDY#**, potwierdzający gotowość modułu docelowego podczas transmisji danych, może przejść w stan niski nie wcześniej niż razem z sygnałem **DEVSEL#**. Raz ustawiony sygnał **DEVSEL#** musi być podtrzymywany aż do końca transmisji danych.

Przewiduje się dwa rodzaje dekodowania adresu: pozytywne i negatywne. W pierwszym przypadku urządzenie dołączone do magistrali PCI samodzielnie określa, czy jest celem operacji. Moduł docelowy stosujący drugi sposób dekodowania adresu czuje się zaadresowany jeśli żadne inne urządzenie nie zdekodowało pozytywnie adresu na magistrali. Ten sposób dekodowania może być wykorzystywany, na danej magistrali, tylko przez jedno urządzenie. Bywa stosowany przez mosty łączące magistralę PCI z magistralą rozszerzeń. Najkrótszy dopuszczalny czas reakcji przy dekodowaniu negatywnym musi być dłuższy od najwolniejszej reakcji pozytywnej ale nie więcej niż o jeden okres przebiegu taktującego **CLK**. Urządzenia odpowiadające pozytywnie dzieli się na 3 kategorie: szybkie, średnie i wolne tak jak to pokazano na rysunku 5.



Rysunek 5. Zależności czasowe przy wystawianiu sygnału DEVSEL#.

Operacje wykonywane w przestrzeni adresowej pamięci oraz konfiguracji, wymagają sprawdzenia sygnałów **AD[1::0]**, przesyłanych w fazie przesyłania adresu. Od poprawnego stanu linii **AD[1::0]** uzależniona jest generacja sygnału **DEVSEL#**.

Sygnały zezwalające **C/BE[3::0]#** określają, które bajty przesyłane liniami **AD[31::0]** podczas fazy przesyłania danych zawierają ważną informację. Mogą się one zmieniać się między fazami przesyłania danych, lecz muszą mieć określoną wartość podczas narastającego zbocza impulsu zegarowego, który rozpoczyna fazę przesyłania danych oraz podczas jej trwania. Moduł docelowy w trakcie operacji

czytania powinien sprawdzać parzystość wszystkich sygnałów **AD[31::0]** i **C/BE[3::0]#**, niezależnie od stanu sygnałów na liniach zezwalających.

Jeśli sterowanie linią magistrali przekazywane jest od jednego do drugiego agenta, to żeby uniknąć współzawodnictwa sygnałów, fazy wykorzystania linii magistralowej przez kolejnych agentów powinien rozdzielać specjalny cykl rozdzielający. Pojawiają się one w różnych momentach czasowych dla poszczególnych sygnałów.

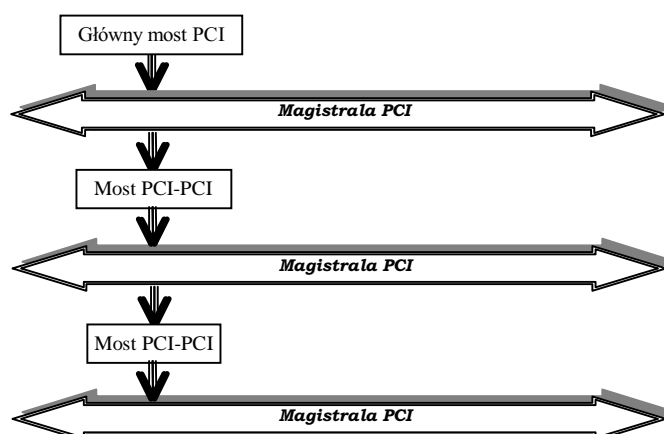
4.5 Rozkazy konfiguracyjne

W normie zakłada się, że urządzenia PCI będą konfigurowane a ich praca inicjowana programowo. Przewidziano do tego oddzielną konfiguracyjną przestrzeń adresową. Każde urządzenie PCI powinno posiadać 256 bajtów w przestrzeni rejestrów konfiguracji. Dekodowanie adresu w przypadku dostępu do przestrzeni adresowej przeznaczonej dla potrzeb konfiguracji powinno się odbywać poza magistralą i być sygnalizowane przy pomocy sygnału **IDSEL**, który w tym wypadku pełni rolę klasycznego sygnału selekcji. Rozkaz konfiguracyjny ma prawo być wykonany przez urządzenie, jeśli w fazie przesyłania adresu **IDSEL** jest w stanie wysokim i spełniony jest warunek **AD[1::0]="00"**. Wewnętrzne adresowanie rejestrów wybranego urządzenia odbywa się na podstawie adresu odczytanego z linii **AD[7::2]**.

W urządzeniach złożonych każdy moduł funkcjonalny MF może mieć własne 256 bajtów rejestrów konfiguracji. Numer modułu funkcjonalnego (wewnątrz wybranego urządzenia złożonego) przesyłany jest po liniach **AD[10::8]**.

Zarezerwowane	MF	Nr rejestru	0	0
---------------	----	-------------	---	---

Rysunek 6. Struktura adresu (typu 0) przy dostępie do rejestrów konfiguracji



Rysunek 7. Wielopoziomowa konfiguracja magistral PCI

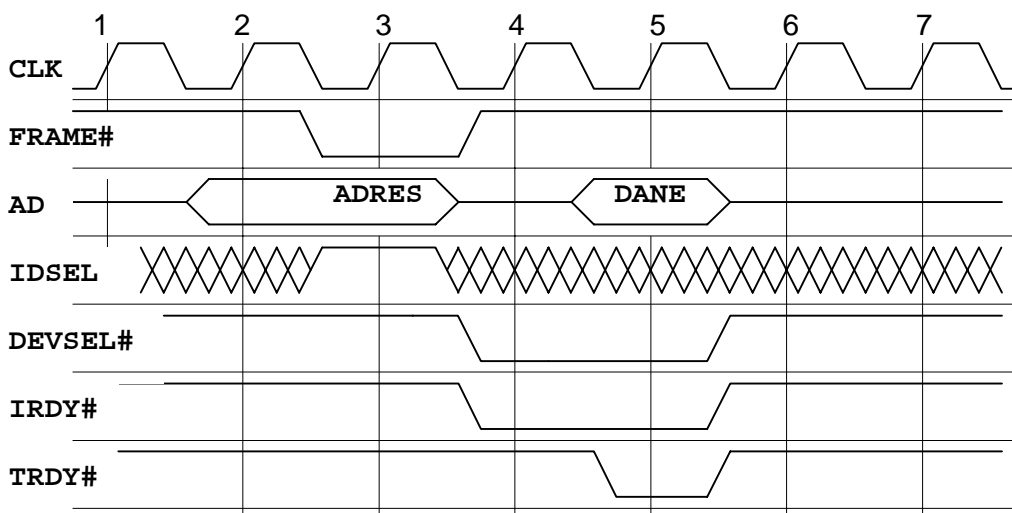
W systemach z wieloma magistralami PCI połączonymi np. tak jak na rysunku 7, kierowana do odpowiedniego mostu PCI-PCI, informacja adresowa (dla przestrzeni rejestrów konfiguracji) ma bardziej złożoną postać. Przekazywany jest numer magistrali i urządzenia, które następnie ma być wybrane.

Zarezerwowane	Nr magistrali	Urządzenie	MF	Nr rejestru	0	1
---------------	---------------	------------	----	-------------	---	---

Rysunek 8. Struktura adresu (typu 1) przy dostępie do rejestrów konfiguracji

Dopiero na poziomie odpowiedniego mostu zaadresowanej magistrali następuje zdekodowanie informacji odebranej (z wyższej magistrali) po liniach **AD[15:11]**, wysterowanie (na niższej magistrali) odpowiedniej linii **IDSEL** i przesłanie na magistralę informacji adresowej okrojonej do postaci typu 0 (rysunek 6).

Sposób sterowania linią **IDSEL** znajduje się w gestii odpowiedniego mostu. W niektórych rozwiązaniach występuje długo utrzymujący się nieustalony poziom logiczny sygnału wyjściowego (na rysunku 9 oznaczony „XXXXXXX”). Dlatego dopuszcza się wcześniejsze wystawianie adresu (tak jak na rysunku 9) aby dać czas na ustabilizowanie się zdekodowanego na jego podstawie sygnału **IDSEL**. Takie rozwiązanie jest możliwe ponieważ adres jest wykorzystywany tylko w fazie adresowej i ważny tylko w połączeniu z aktywnym sygnałem **FRAME#**.



Rysunek 9. Przykład operacji odczytu z rejestru konfiguracji

Rozkaz konfiguracyjny, może dotyczyć zarówno przesłania bajtu, słowa (16 bitów), podwójnego słowa (32 bity) jak i bloku danych. Podczas operacji przesłania bloku danych kolejne adresy są

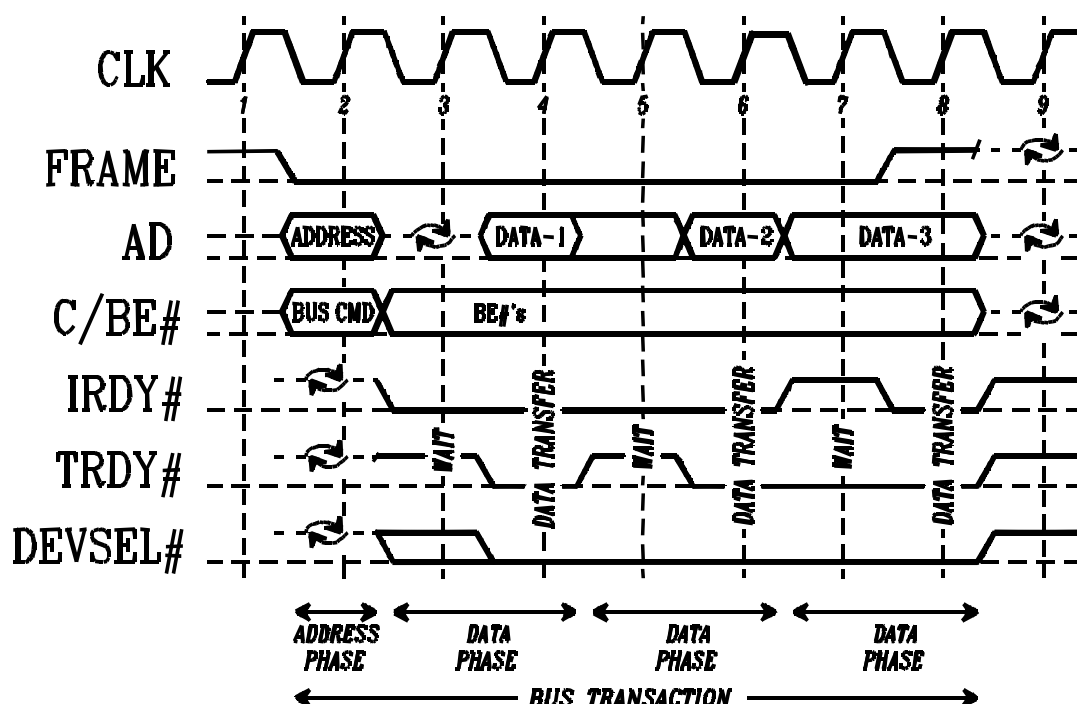
obliczane przez zwiększenie adresu początkowego (odczytanego wcześniej z linii adresowych **AD[7::2]**).

4.6 Operacje w przestrzeni adresowej pamięci

Podobnie jak w przypadku dostępu do rejestrów konfiguracji operacje wykonywane w przestrzeni adresowej pamięci wymagają sprawdzenia bitów odebranych z linii **AD[1::0]**. W zależności od ich wartości wykonywana jest operacja transmisji blokowej lub cykl jest kończony po pierwszej fazie przesyłania danych.

AD1	AD0	Rodzaj transmisji
0	0	transmisja blokowa z liniowym zwiększaniem adresów
0	1	przekazanie wiersza pamięci podręcznej
1	X	kończona po pierwszej fazie przesyłania danych

Podczas transmisji bloku danych kolejne adresy są obliczane przez zwiększenie adresu początkowego, który został przesłany po liniach adresowych **AD[31::02]**.



Rysunek 10. Operacja odczytu pamięci

Na rysunku 10 przedstawiono przebiegi czasowe 32.bitowej operacji odczytu pamięci. Przechodzący w stan niski sygnał **FRAME#**

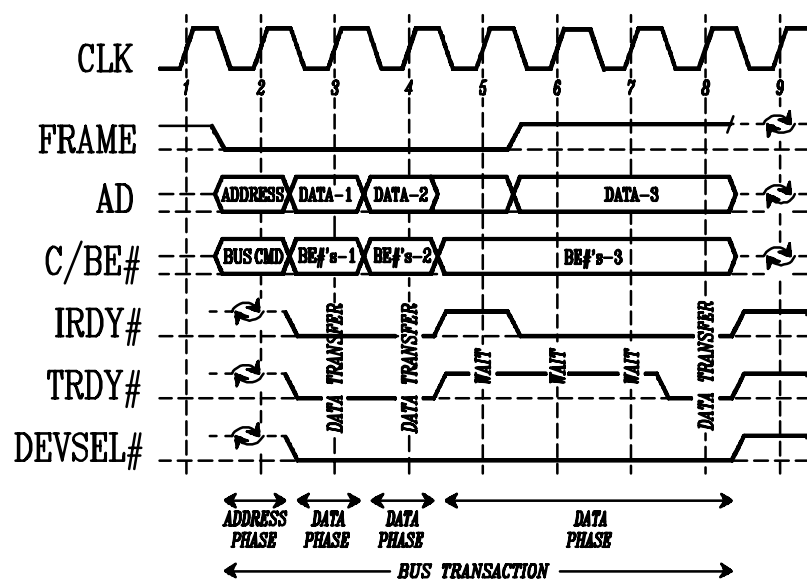
rozpoczyna fazę przesyłania adresu. Podczas narastającego zbocza 2. impulsu zegarowego stan linii **AD[31::0]** oraz linii **C/BE[3::0]#** wyznacza ważne kody adresu i rozkazu. Faza przesyłania danych rozpoczyna się od impulsu zegarowego 3. Teraz linie **C/BE#** określają, które bajty na liniach danych zawierają ważną informację. Faza przesyłania danych może zostać wydłużona w oczekiwaniu na sygnał **TRDY#**, który potwierdza ważność danych wysyłanych przez moduł docelowy. Mówi się wówczas, że wprowadzane zostały cykle oczekiwania. Bufory wyjściowe przesyłające sygnały **C/BE#** muszą być wysterylizowane impulsami zezwalającymi trwającymi od pierwszego impulsu zegarowego podczas fazy przesyłania danych, aż do zakończenia operacji.

Między fazą przesyłania adresu i pierwszą fazą przesyłania danych, kiedy sygnał **TRDY#** jest w stanie wysokim, konieczne jest wprowadzenie sygnału rozdzielającego, ponieważ podczas następnego impulsu zegarowego, kiedy sygnał **DEVSEL#** przejdzie w stan niski, moduł nadrzędny przestanie sterować multipleksowanymi liniami adresów i danych, a kontrolę nad nimi przejmie moduł docelowy.

Podczas fazy przesyłania danych, kiedy sygnał **TRDY#** lub **IRDY#** jest w stanie wysokim, wprowadzane są cykle oczekiwania. Moduł docelowy może najwcześniej uzyskać dane podczas impulsu zegarowego 4. Faza przesyłania danych kończy się, kiedy sygnały **IRDY#** i **TRDY#** są w stanie niskim podczas tego samego impulsu zegarowego. Dane są przesyłane podczas impulsu zegarowego 4, 6 i 8, natomiast cykle oczekiwania są wprowadzane podczas impulsów 3, 5 i 8. Pierwsza faza przesyłania danych realizuje się w minimalnym czasie. Druga faza jest wydłużona, ponieważ **TRDY#** jest w stanie wysokim, natomiast ostatnia faza jest wydłużona, ponieważ sygnał **IRDY#** jest w stanie wysokim podczas impulsu zegarowego 7. Moduł nadrzędny otrzymuje informację (sygnał **FRAME#** w stanie wysokim), że po impulsie zegarowym 7 nastąpi ostatnia faza przesyłania danych. Moduł nadrzędny nie jest jeszcze gotowy, żeby zakończyć ostatnią fazę przesyłania danych podczas impulsu zegarowego 7, wobec czego zakończenie operacji nastąpi dopiero podczas impulsu zegarowego 8.

Na rysunku 11 przedstawione zostały przebiegi czasowe podczas 32-bitowej operacji zapisu do pamięci. Rozpoczyna się ona od przejścia sygnału **FRAME#** w stan niski w czasie impulsu zegarowego 2. Operacja zapisu jest podobna do operacji odczytu z tym wyjątkiem, że nie jest potrzebne wprowadzenie cyklu rozdzielającego po fazie przesyłania adresu, ponieważ ten sam moduł (nadrzędny) wysyła zarówno adres jak i dane. Fazy przesyłania danych są takie same dla obu operacji.

W przykładzie pokazanym na rysunku 11 pierwsza i druga faza operacji kończona jest bez wprowadzania cykli oczekiwania. W fazie trzeciej są wprowadzone przez moduł docelowy trzy cykle oczekiwania. W fazie piątej zarówno moduł nadrzędny jak i docelowy wprowadzają cykle oczekiwania. Ostatnia faza przesyłania danych jest sygnalizowana przez moduł nadrzędny (sygnał **FRAME#** w stanie wysokim) w trakcie impulsu zegarowego 6. Jej zakończenie następuje podczas impulsu zegarowego 8.



Rysunek 11. Operacja zapisu do pamięci

4.7 Dostęp wyłączny

Dla prawidłowego działania systemu wieloprocessorowego, aby zapobiec równoczesnej modyfikacji danych przez kilka procesorów, stosowany jest mechanizm porządkujący dostęp do wspólnej pamięci. W systemach, w których zasoby mogą być dzielone, moduł nadrzędny może potrzebować dostępu wyłącznego do fragmentu pamięci modułu docelowego dla przeprowadzenia sekwencji operacji nierozdzielnych. W tym czasie ten fragment pamięci powinien być niedostępny dla innych agentów. Blokowanie jest mechanizmem gwarantującym wyłączność w dostępie do zasobów wykorzystywanych przez dane zadanie (w tym czasie nie mogą być użyte przez inne zadanie).

W celu zablokowania dostępu do fragmentu pamięci modułu docelowego wykorzystuje się sygnał magistrali PCI **LOCK#**. Agent żądający dostępu wyłącznego przed wysłaniem sygnału **REQ#** powinien sprawdzić stan sygnału **LOCK#**. Sygnałowi **LOCK#** przypisuje się stan zajęty od momentu, w którym znalazł się w stanie niskim. Sygnał uważa się za zwolniony jeśli oba sygnały **FRAME#** i **LOCK#** znajdują się w stanie wysokim. Jeśli sygnał **LOCK#** jest zajęty, agent żądający dostępu wyłącznego powinien opóźnić wysłanie sygnału **REQ#** - poczekać aż się zwolni. Jeśli **LOCK#** nie jest zajęty oraz agent uzyskał dostęp do magistrali, może przejąć nad nim kontrolę. Pozostali agenci nie mogą teraz sterować sygnałem **LOCK#** nawet, jeśli uzyskają dostęp do magistrali.

W czasie kiedy jeden z agentów kontroluje sygnał **LOCK#**, pozostali (nie wykorzystujący dostępu wyłącznego) mogą efektywnie wykonywać

operacje na magistrali PCI pod warunkiem, że nie będą one odwoływać się do zablokowanych zasobów.

4.8 Wykorzystanie pamięci podręcznej

Norma PCI umożliwia współpracę bloku pamięci na magistrali PCI z mostem sprzęgającym, za którym znajduje się pamięć podręczna (lub pamięcią podręczną innego agenta) i wprowadza mechanizm kontroli zgodności zawartości pamięci oraz śledzenia operacji dostępu. W tym celu wykorzystuje się sygnały magistrali **SDONE** i **SBO#**. Przy ich pomocy przesyła się informację o statusie pamięci podręcznej. Każdy agent korzystający z buforowania pamięcią podręczną musi monitorować te sygnały i w odpowiedni sposób na nie reagować. Sygnał **SDONE** jest ustawiany w stan wysoki, kiedy procedura kontrolna została zakończona. Stan niski sygnału **SBO#** oznacza, że bieżący dostęp do pamięci dotyczy adresów, w których przechowywane są nieaktualne dane. Kiedy sygnały **SBO#** i **SDONE** są w stanie wysokim oznacza to, że dane są aktualne i mogą być wykorzystane przez agenta.

4.9 Operacje w przestrzeni adresowej rejestrów we-wy

W przypadku operacji wykonywanych w przestrzeni adresowej zarezerwowanej dla operacji we-wy, wszystkie linie **AD[31::0]** określają adres pierwszego (najmłodszego) bajtu przesyłanych danych. Konieczność operowania adresami zmienianymi z dokładnością co do bajtu podyktowana jest tym, że może zdarzyć się połączenie, w ramach tego samego podwójnego słowa (32 bity), rejestrów należących do różnych urządzeń we-wy.

Na ogół w tej przestrzeni adresowej mamy do czynienia z pojedynczymi przesłaniami danych. Transmisja blokowa nie jest unormowana chociaż jest dopuszczalna. Adres, na magistralę PCI, powinien być przekazywany dokładnie w takiej postaci w jakiej został wygenerowany przez jednostkę centralną. Szerokość słowa danych, wyznaczana przez **C/BE#**, musi być zgodna z możliwościami zaadresowanego sterownika we-wy.

W przestrzeni rejestrów we-wy linie **AD[1::0]** (inaczej niż w przypadku pamięci) wskazują położenie najmłodszego bajtu w transmitowanym słowie danych.

AD1	AD0	C/BE#3	C/BE2#	C/BE1#	C/BE0#
0	0	X	X	X	L
0	1	X	X	L	H
1	0	X	L	H	H
1	1	L	H	H	H

4.10 Przerwania

Wykorzystanie systemu przerw na magistrali PCI jest opcjonalne. Niski poziom napięcia na jednej z linii **INT#x** sygnalizuje żądanie przerwania i powoduje po pewnym czasie wykonanie rozkazu potwierdzenia przerwania (linie **C/BE[3::0]="0000"**) dla odczytania wektora przerwania. W fazie przesyłania adresu linie **AD[31::0]** nie przesyłają ważnej informacji, ponieważ agent, którego dotyczy operacja jest określony (tylko jeden może odpowiadać na rozkaz potwierdzenia przerwania i ustawiać sygnał **DEVSEL#**). Wektor przerwania jest przesyłany po liniach **AD[31::0]** w fazie przesyłania danych, kiedy **TRDY#** jest w stanie niskim. W razie potrzeby mogą być wprowadzane cykle oczekiwania lub operacja może być zakończona przed czasem.

4.11 Kończenie operacji na magistrali PCI

Zakończenie operacji na magistrali PCI może zostać zainicjowane przez moduł nadrzędny lub docelowy. Moduł nadrzędny zachowuje całkowitą kontrolę nad magistralą także wtedy, kiedy zakończenie operacji zostało zainicjowane przez moduł docelowy.

Moduł docelowy może zainicjować zakończenie operacji, kiedy nie jest w stanie jej wykonać lub też czas niezbędny do zakończenia bieżącej fazy przesyłania danych okaże się dłuższy niż trwa 8 impulsów zegarowych. W takiej sytuacji moduł docelowy chcąc poinformować moduł nadrzędny o konieczności zakończenia bieżącej fazy przesyłania danych wystawia sygnał **STOP#**. Sygnał **STOP#** raz ustawiony w stan niski powinien być trzymany dopóki sygnał **FRAME#** nie przejdzie w stan wysoki.

Moduł nadrzędny rozpoczyna procedurę zakończenia operacji, przez ustawienie sygnału **FRAME#** w stan wysoki, kiedy sygnał **IRDY#** jest w stanie niskim. Informuje w ten sposób moduł docelowy, że oczekiwane jest zakończenie przesyłania danych, które nastąpi, kiedy sygnały **IRDY#** i **TRDY#** znajdą się w stanie niskim. Kiedy obydwa sygnały znajdą się w stanie wysokim magistrala przejdzie w stan jałowy. Moduł nadrzędny może zainicjować zakończenie operacji, kiedy wszystkie dane zostały przesłane lub kiedy nastąpiło przekroczenie limitu czasu przewidzianego na wykonanie danej operacji.

4.12 Cykl specjalny

Rozpoznawanie tego typu rozkazów nie jest obowiązkowe. Umożliwiają one przesyłanie przez moduł nadrzędny komunikatów do jednego lub więcej modułów docelowych. Moduł docelowy musi sam określić czy komunikat jest skierowany do niego. Rozkazy te odróżnia od innych rozkazów to, że w odpowiedzi na nie bloki docelowe nie wysyłają sygnału **DEVSEL#**. W fazie przesyłania adresu linie **C/BE[3::0]** przesyłają informację, że będzie wykonywany rozkaz typu cykl specjalny, natomiast linie **AD[31::0]** nie przesyłają ważnej

informacji. W fazie przesyłania danych liniami **AD[15::0]** przesyłany jest zakodowany komunikat, który jest dekodowany przez moduł docelowy. Także linie **AD[31::16]** mogą być wykorzystane opcjonalnie do przesyłania komunikatów.

4.13 Wykrywanie i sygnalizacja błędów

Wykrywane i sygnalizowane są błędy parzystości pojawiające się podczas fazy przesyłania adresów i danych po liniach **AD[31::0]** i **C/BE[3::0]**, oraz błędy systemowe na magistrali PCI. Agent, który uzyskał przydział magistrali, musi sterować wszystkimi liniami **AD**, ponieważ kontrola parzystości dotyczy wszystkich linii, niezależnie od tego, czy przesyłana nimi informacja jest ważna.

Do sygnalizacji błędów są wykorzystywane dwa sygnały: **PERR#** oraz **SERR#**. Pierwszy jest używany wyłącznie dla kontroli parzystości podczas wszystkich operacji, z wyjątkiem rozkazów specjalnych. Protokół zapewnia, że tylko jeden agent w danej chwili może ustawić ten sygnał w stan niski. Powinien on pozostać w tym stanie jeszcze przez dwa impulsy zegarowe po zakończeniu operacji. Sygnał **SERR#** jest używany dla sygnalizacji błędu parzystości podczas wykonywania rozkazu specjalnego oraz błędów systemowych. Specyfikacja zaleca, żeby sygnał **SERR#**, który może być ustawiony w stan niski przez kilku agentów równocześnie, był generowany z opóźnieniem dwóch impulsów zegarowych w stosunku do zauważonego błędu.

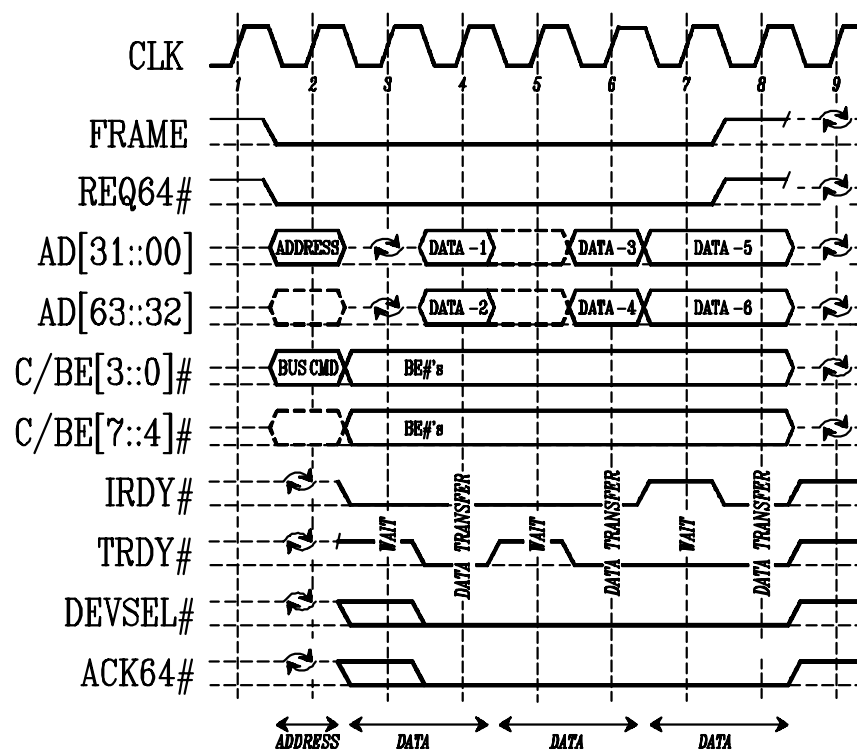
4.14 Rozszerzenie magistrali PCI do 64 bitów

Rozszerzenie magistrali do 64 bitów wymaga 39 dodatkowych wyprowadzeń dla sygnałów: **REQ64#**, **ACK64#**, **AD[63::32]**, **C/BE[7::4]** oraz **PAR64**. Operacje 64.bitowe są za każdym razem negocjowane. Blok nadrzędny wystawia **REQ64#**. Moduł docelowy, jeśli jest przygotowany do 64.bitowej operacji, odpowiada sygnałem **ACK64#**.

Tylko rozkazy odnoszące się do przestrzeni adresowej pamięci mogą być wykonywane jako 64.bitowe. Potwierdzenie przerwania oraz cykle specjalne są z definicji rozkazami odnoszącymi się do transmisji 32.bitowych. W razie wykrycia błędu parzystości na liniach **AD[63::32]** lub **C/BE[7::4]** sygnał **PAR64** przechodzi w stan niski.

Na rysunku 12 przedstawiono przebiegi czasowe 64.bitowej operacji odczytu. Moduł nadrzędny żąda 64.bitowej operacji ustawiając sygnał **REQ64#** w stan niski. Moduł docelowy potwierdza gotowość wykonania operacji przez ustawienie sygnału **ACK64#** w stan niski. Sygnały te są odpowiednikami sygnałów **FRAME#** i **DEVSEL#** przy operacjach 32.bitowych.

Agenci 32.bitowi mogą współdziałać z 64.bitowymi. Jeśli 64.bitowy agent nie uzyskał potwierdzenia możliwości przeprowadzenia operacji 64.bitowej, wówczas automatycznie powinien przełączyć się na wykonanie operacji 32.bitowej.



Rysunek 12. 64.bitowa operacja odczytu