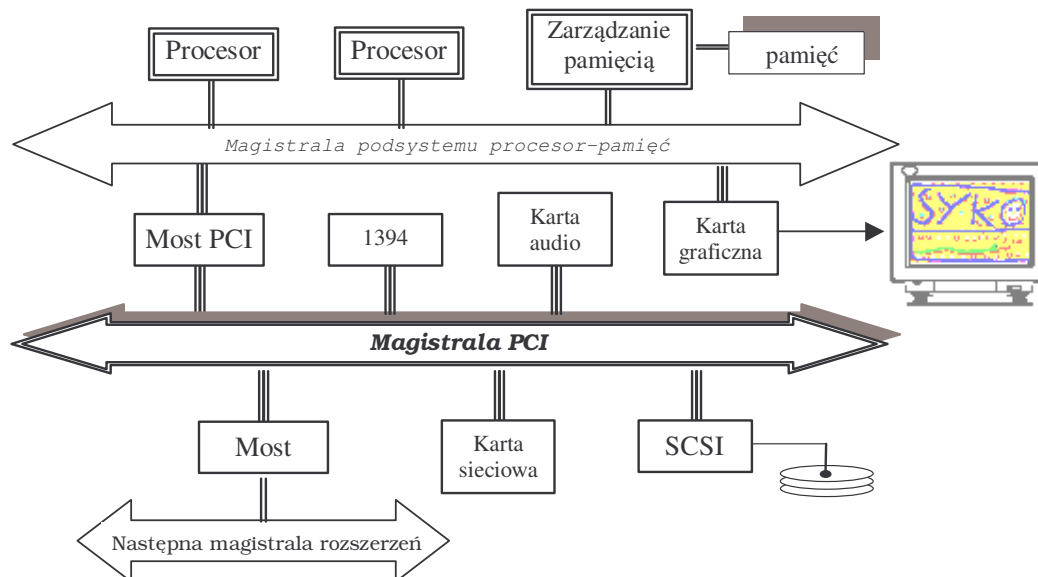


Magistrale systemowe: magistrala PCI

Magistrala grupuje wspólne dla kilku urządzeń połączenia wykorzystywane do przesyłania sygnałów, nadawanych z jednego z kilku możliwych źródeł do jednego lub kilku miejsc przeznaczenia. Zwykle o dostęp do magistrali może ubiegać się kilka urządzeń. Jeśli wysyłane sygnały mają być poprawnie odebrane to, w danej chwili, liniami magistrali powinno sterować tylko jedno urządzenie. Natomiast dane transmitowane magistralą mogą być odbierane przez wszystkie urządzenia do niej dołączone.

W systemach komputerowych występują różne magistrale łączące układy wykorzystywane na różnych poziomach systemowej hierarchii. Poniżej podano niektóre istotne parametry charakteryzujące magistrale transmisji danych:

- | | |
|---|--|
| Sposób wykorzystania linii magistrali | - linie dedykowane lub przełączane. |
| Szerokość ścieżki danych | - liczba równoległych linii umożliwiających jednoczesną transmisję bitów danych. |
| Sposób potwierdzania przesłania danych | transmisja synchroniczna lub asynchroniczna. |
| Taktowanie | - częstotliwość zegara taktującego (o ile występuje). |
| Rodzaje operacji transmisji danych | - zapis, odczyt, odczyt-modyfikacja-zapis, odczyt kontrolny, blokowe przesyłanie danych. |
| Arbitraż dostępu | - centralny lub rozproszony. |



Rysunek 1. Przykład systemu komputerowego z magistralą PCI.

Od blisko 10 lat do rozszerzania funkcji systemu komputerowego najczęściej wykorzystywana jest magistrala PCI (*Peripheral Component Interconnect*). Jest to magistrala, o dość dużej przepustowości (w wersji podstawowej do 132 MB/s), z centralnym arbitrażem i synchronicznym taktowaniem transmisji danych. Na magistrali PCI do przesyłania adresu i danych wykorzystywane są te same linie, przełączane odpowiednio w zależności od fazy operacji i szerokości ścieżki danych.

1. OPIS STANDARDU



W normie *PCI Local Bus* (ostatnia wersja: *Revision 2.2* pochodzi z 18. grudnia 1998) wydanej przez *PCI Special Interest Group*, zdefiniowano magistralę 32-bitową, ze wspólnymi przełączanymi liniami adresu i danych, synchronizowaną przebiegiem zegarowym o częstotliwości do 33 MHz. Przewidziano możliwość rozszerzenia ścieżki danych do 64 bitów i wprowadzenia dodatkowej częstotliwości zegara taktującego równej 66 MHz. Przyjęty protokół transmisji danych dostosowany jest do przesyłania sekwencyjnego. Przy częstotliwości zegara 33 MHz, magistralą PCI można transmitować 32-bitowe dane z szybkością do 132 Megabajtów/sekundę. 64-bitowe rozszerzenie i zastosowanie zegara 66 MHz pozwala tą szybkość zwiększyć czterokrotnie.

W normie założono, że bezpośredni styk z magistralą obsługiwany jest przez wyspecjalizowany układ scalony wielkiej skali integracji. Określone normą parametry elektryczne odnoszą się do tego układu. Nie przewidziano dodatkowego buforowania. Magistrala nie jest zakończona dopasowanymi terminatorami. Układy sterujące dostarczają jedynie połowę napięcia w stanie wysokim lub niskim; sygnał osiąga właściwy poziom po zsumowaniu się fali pierwotnej z odbitą od końca magistrali.

W hierarchii systemowej magistrala PCI usytuowana jest zwykle bezpośrednio poniżej magistrali podsystemu *jednostka centralna-pamięć podręczna-pamięć operacyjna* a z bezpośrednim otoczeniem procesora łączy ją most-sterownik magistrali PCI. Most PCI często dodatkowo pełni funkcję bufora danych oraz układu arbitrażu.

Dołączane za pośrednictwem magistrali urządzenia, które rozszerzają funkcje systemu komputerowego, mogą realizować swoje zadania niezależnie od jednostki centralnej. Moduły mogą przejmować kontrolę nad magistralą i inicjować bezpośrednią wymianę informacji z urządzeniami we-wy, tak samo jak jednostka centralna. Urządzenia skupione wokół magistrali PCI mogą współpracować ze sobą bez pośrednictwa podsystemu *jednostka centralna-pamięć podręczna-pamięć operacyjna*.

Zastosowanie magistrali PCI w systemie komputerowym umożliwia jego rozbudowę niezależnie od typu jednostki centralnej. I tak np. wykonana zgodnie z normą karta sterownika we-wy może być użyta na magistrali PCI zarówno w stacji roboczej sterowanej procesorem PowerPC, UltraSparc jak i Pentium. Różnice dotyczą oprogramowania systemowego.

W systemie PCI dołączane urządzenia konfigurowane są automatycznie. Po włożeniu nowej karty w gniazdo rozszerzeń i zainicjowaniu pracy systemu - numery przerwań, kanały DMA, adresy we-wy itd. przydzielane są automatycznie. 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. W urządzeniach standardu PCI, w specjalnie do tego

celu wydzielonej przestrzeni adresowej, dostępne są ich charakterystyki i rejestry umożliwiające programowe ustawianie konfiguracji oraz zainicjowanie pracy.

2. OMÓWIENIE WAŻNIEJSZYCH TERMINÓW

- Agent** jednostka podejmująca działanie na magistrali komputera.
(ang. agent)
- Blokowanie dostępu** mechanizm umożliwiający zamknięcie, przed innymi agentami, dostępu do wspólnej pamięci na czas wykonywania tzw. krytycznej 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 na magistrali PCI: 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)
- Śledzenie dostępu (do pamięci operacyjnej)** prowadzona przez podsystem pamięci podręcznej kontrola adresów wystawianych (przy dostępie do obszaru pamięci głównej w module docelowym PCI) przez inne niż procesor centralny urządzenie: jeśli zaadresowana komórka pamięci ma swój duplikat w jednym z wierszy pamięci podręcznej a wiersz ten był modyfikowany to moduł docelowy zostaje poinstruowany by zawiesił dostęp do tych danych do czasu usunięcia niespójności.
(ang. bus snooping)

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

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

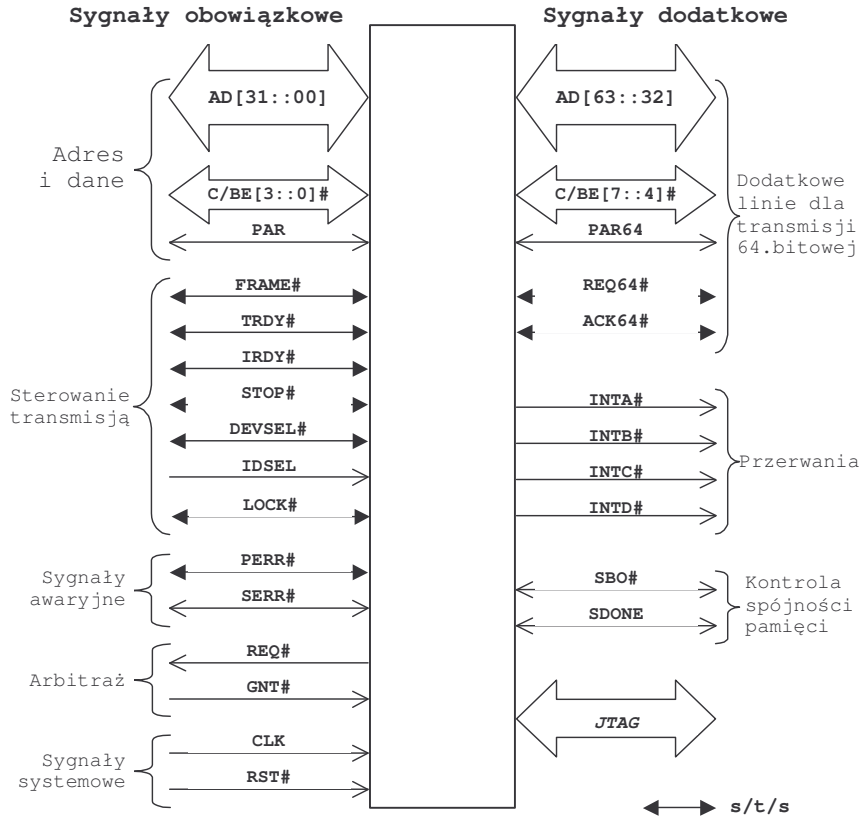
3. SYGNAŁY MAGISTRALI PCI

Magistrala PCI może być wykorzystywana jako magistrala 32. lub 64.bitowa. Sygnały obowiązkowe są wspólne dla obu konfiguracji. Wśród sygnałów dodatkowych są zarówno wspólne jak 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	Sygnał trójstanowy 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 być sterowana tylko z jednego źródła. 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 podtrzymywanie 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.



Rysunek 2. Zestawienie sygnałów magistrali PCI

3.2 Sygnały obowiązkowe

W zależności od funkcji można je zaliczyć do jednej z następujących 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 w urządzeniach zawierających fragment pamięci operacyjnej. Sygnały arbitrażu doprowadzane są jedynie do modułów nadrzędnych.

RST#	in	Sygnal ogólnego zerowania. Zeruje rejestry, liczniki oraz wskaźniki nadzorujące kolejność wykonywanych operacji. Działa asynchronicznie niezależnie od sygnału zegara systemowego.
IDSEL	in	Sygnal wykorzystywany do adresowania urządzeń PCI podczas operacji odczytu i zapisu rejestrów konfiguracyjnych.
CLK	in	Sygnal zegarowy synchronizujący wszystkie operacje na magistrali PCI. Sygnal 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	Indywidualny sygnał żądania przydziału magistrali wysyłany przez moduł nadrzędny do układu arbitrażu.
GNT#	t/s	Wysyłany przez arbitra sygnał potwierdzający przyznanie modułowi nadrzêdnemu prawa do magistrali. Każdy moduł nadrzêdny 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 przełączane. 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	Sygnal generowany przez moduł nadrzêdny, który uzyskał prawo do sterowania magistralą. Określa moment rozpoczęcia i czas trwania operacji: sygnał FRAME# aktywny w stanie niskim pojawia się na początku fazy adresowej a przed ostatnim przesłaniem danych 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 nadrzêdnego do odbioru informacji. Faza jest zakończona, kiedy dla tego samego narastającego zbocza impulsu zegarowego sygnały IRDY# i TRDY# są w stanie niskim.

TRDY#	s/t/s	Sygnal 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. Faza kończy się kiedy sygnały IRDY# i TRDY# są w stanie niskim podczas narastającego zbocza impulsu zegarowego.
DEVSEL#	s/t/s	Sygnal 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	Sygnal – wysyłany 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 - jest wysyłany przez urządzenie PCI, kiedy pojawił się błąd parzystości w fazie przesyłania adresu lub danych podczas cyklu specjalnego lub też pojawił się inny błąd niż błąd parzystości, który może spowodować katastrofalne skutki. Sygnal utrzymywany jest w stanie niskim podczas jednego impulsu zegarowego.
LOCK#	s/t/s	Sygnal wykorzystywany przez moduł nadrzędny, podczas wykonywania operacji nierozdzielnych (np. czytaj-modyfikuj-zapisz), 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 asynchronicznie przez kontroler przerwań) oznacza żądanie przerwania. Urządzenia, w których można wyodrębnić różne moduły funkcjonalne mogą wykorzystywać do czterech linii przerwań. Urządzenia z jednym modułem 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 podawany jest za pośrednictwem sygnału SBO# : stan wysoki oznacza, że dostęp do pamięci nie naruszy spójności pamięci. W przeciwnej sytuacji moduł docelowy powinien zasignalizować modułowi nadrzędnemu konieczność powtórzenia operacji.
SBO#	in/out	Sygnał wyjściowy mostu (pochodzący od układu kontrolującego zgodność zawartości pamięci podręcznej z zawartością buforowanych obszarów pamięci operacyjnej), wejściowy dla układów pamięci usytuowanych na magistrali PCI buforowanych przez pamięć podręczną. Ustawiony przez most w stan niski równocześnie z sygnałem SDONE (w stanie wysokim!) 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 wcześniej zmodyfikowany.

3.4 Sygnały związane z magistralą 64 bitową

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 Ogólne zasady transmisji danych

Stan wszystkich linii sygnałowych (za wyjątkiem tych, którymi przekazywane są działające asynchronicznie sygnały ogólnego zerowania i zgłoszeń przerw) badany jest podczas narastającego zbocza impulsu zegarowego **CLK**. W tym momencie stany nieustalone sygnałów są niedopuszczalne. Poza tym zboczem stany nieustalone są ignorowane.

Jeśli sterowanie linią magistrali przekazywane jest od jednego agenta do drugiego, to żeby uniknąć współzawodnictwa sygnałów, fazy wykorzystania danej linii magistralowej przez kolejnych agentów są rozdzielane dodatkowym cyklem magistrali. Na rysunkach cykle rozdzielające są przedstawiane w postaci dwóch strzałek wskazujących na swoje końce. Pojawiają się one w różnych momentach czasowych w zależności od sygnału.

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. Przesyłanie sekwencyjne dopuszczalne jest również w przestrzeni adresowej urządzeń we-wy.

Inicjatorem wymiany informacji jest moduł nadrzędny, który w wyniku przeprowadzonej procedury arbitrażu uzyskał dostęp do magistrali. Na czas wykonania operacji wymiany informacji pomiędzy modułami nadrzędnym i docelowym, magistrala przechodzi ze stanu jałowego w stan aktywny. Świadczy o tym przejście w stan aktywny sygnału **FRAME#**.

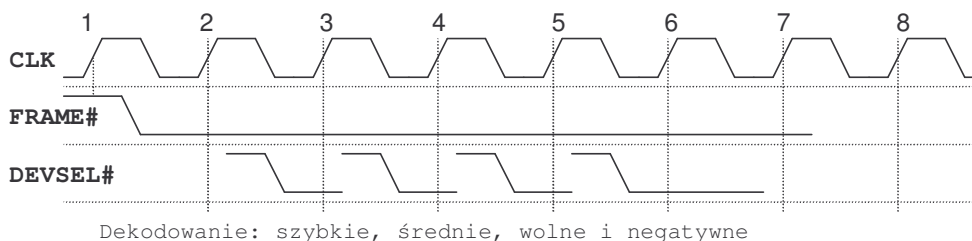
Sygnał **FRAME#** pozostaje aktywny aż do ostatniego przesłania danych. Podczas ostatniego przesłania danych sygnał **FRAME#** jest w stanie wysokim a o tym, że magistrala jest nadal aktywna decyduje obecność aktywnego sygnału gotowości modułu inicjującego **IRDY#**.

4.2 Faza przesyłania adresu

Dopóki sygnały **FRAME#** i **IRDY#** są nieaktywne (w stanie wysokim) magistrala pozostaje w stanie jałowym. Pierwsze narastające zbocze impulsu zegarowego, przy którym sygnał **FRAME#** jest już aktywny (stan niski) rozpoczyna fazę przesyłania adresu (liniami **AD[31::0]**).

Każde docelowe urządzenie PCI ma obowiązek dekodować przesyłane magistralą adresy. Jeśli rozpozna jeden ze swoich adresów potwierdza to sygnałem **DEVSEL#**. Raz ustawiony sygnał **DEVSEL#** powinien być podtrzymywany aż do końca operacji transmisji danych.

Dopuszczalne są dwa sposoby dekodowania adresu: dekodowanie pozytywne i negatywne. W pierwszym przypadku urządzenie dołączone do magistrali PCI samodzielnie dekoduje adres i określa, czy jest celem operacji. Urządzenia odpowiadające pozytywnie dzieli się, w zależności od szybkości reakcji (tak jak to pokazano na rysunku 3). Wyróżnia się urządzenia: szybkie, średnie i wolne.



Rysunek 3. Zależności czasowe przy wystawianiu sygnału **DEVSEL#**.

Moduł docelowy stosujący negatywny sposób dekodowania adresu powinien „poczuć się” zaadresowanym jeśli żadne inne urządzenie na magistrali nie zdekodowało adresu pozytywnie. Na danej magistrali ten sposób dekodowania może być wykorzystywany 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**.

W przypadku operacji odnoszących się do przestrzeni adresowej pamięci lub konfiguracji, w fazie przesyłania adresu sprawdzane są sygnały na liniach **AD[1::0]**. Jeśli stan linii **AD[1::0]** nie jest poprawny generacja sygnału **DEVSEL#** zostaje wstrzymana.

4.3 Faza przesyłania danych

Z chwilą pojawienia się sygnału **DEVSEL#** kończy się faza przesyłania adresu i rozpoczyna jedna lub sekwencja kilku faz przesyłania danych.

Transmisja danych magistralą PCI kontrolowana jest przez dwa sygnały, których koincydencja kończy każdą pojedynczą fazę przesyłania danych:

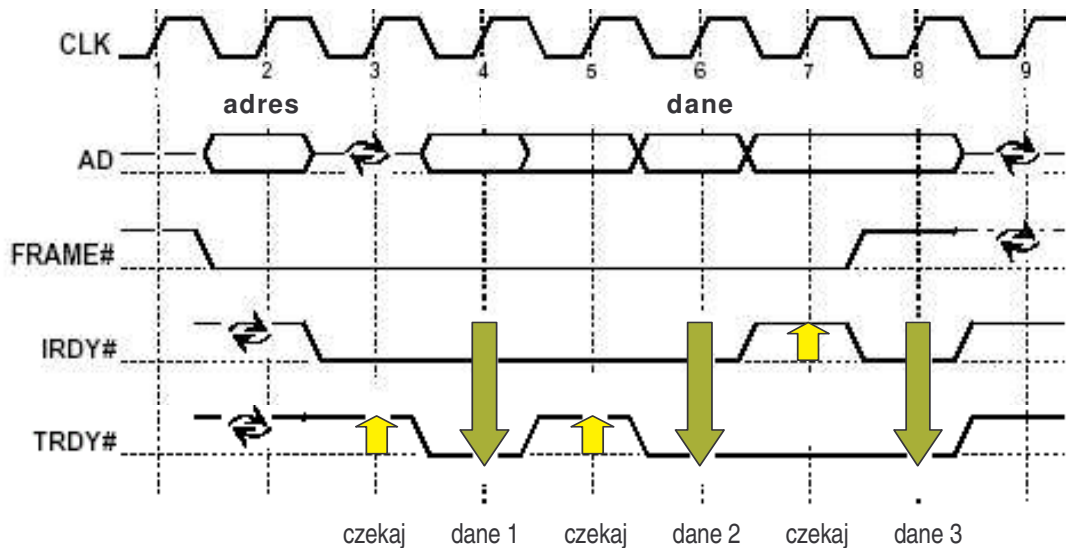
- IRDY#** sygnał gotowości wysyłany przez moduł nadrzędny,
- TRDY#** sygnał gotowości wysyłany przez moduł docelowy.

Sygnał **TRDY#**, potwierdzający gotowość modułu docelowego podczas transmisji danych, może zostać ustawiony w stan niski nie wcześniej niż razem z sygnałem **DEVSEL#**.

Rola sygnałów gotowości zmienia się wraz z kierunkiem przesyłania danych:

- Agent wysyłający dane ustawia swój sygnał gotowości **xRDY#** (sygnał **IRDY#** w cyklu zapisu lub **TRDY#** w cyklu odczytu) informując, że dane są już ważne.

- Agent odbierający dane jeśli jest gotowy przyjąć dane, to nie czekając na sygnał gotowości nadawcy ustawia swój sygnał **xRDY#** w stan niski.
- Za każdym razem kiedy oba sygnały **TRDY#** i **IRDY#** znajdują się w stanie aktywnym kończona jest jedna faza przesyłania danych.
- Jeśli jeden z sygnałów **IRDY#** lub **TRDY#** jest w stanie wysokim oznacza to, że moduł (odpowiednio nadrzędny lub docelowy) wprowadza stan oczekiwania.



Rysunek 4. Przebieg transmisji danych na magistrali PCI

Jeśli w narastającym zboczcu zegara sygnał **FRAME#** jest w stanie wysokim przy aktywnym sygnale **IRDY#** oznacza to, że bieżące przesłanie danych jest ostatnie.

Za pośrednictwem sygnałów zezwalających **C/BE[3::0]#** określa się, które bajty przesyłane liniami **AD[31::0]** podczas fazy przesyłania danych zawierają ważną informację.

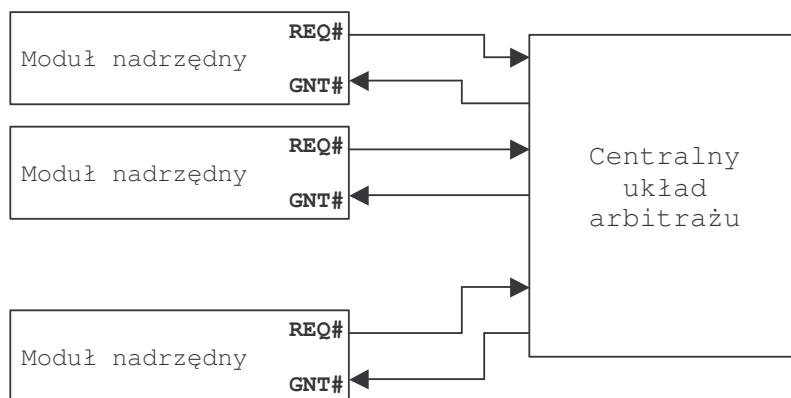
4.4 Arbitraż

W systemie wieloprocessorowym każdy z procesorów – chcąc uzyskać dostęp do jakiejś części wspólnych zasobów, powinien upewnić się, że w swym działaniu nie zderzy się z innym procesorem. Potrzebny jest mechanizm przyznawania dostępu do wspólnych zasobów dostępnych za pośrednictwem jednej magistrali.

W standardzie PCI o wyniku współzawodnictwa w dostępie do magistrali rozstrzyga centralny układ arbitrażu, z którym każdy moduł nadrzędny połączony jest indywidualnymi liniami sygnałów **REQ#** i **GNT#**. Moduł nadrzędny pragnący zainicjować jakąś operację na magistrali PCI sygnalizuje to wystawiając sygnał **REQ#** a następnie czeka na sygnał zezwolenia **GNT#**.

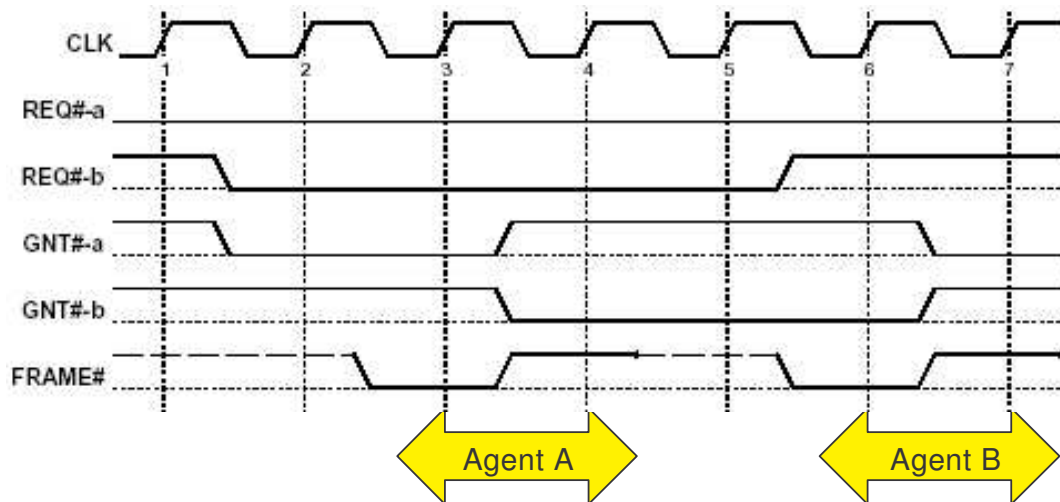
W normie PCI nie zdefiniowano algorytmu, według którego układ arbitrażu ma wskazać agenta uzyskującego przydział magistrali. Może to być zasada „kto pierwszy ten lepszy”, priorytet cykliczny lub inny sposób rozstrzygnięcia o dostępie do magistrali.

Agent, który przejął kontrolę nad magistralą i zamierza przeprowadzić więcej niż jedną operację może utrzymywać sygnał **REQ#** w stanie niskim do czasu uzyskania zezwolenia na ostatnią operację. W efekcie ponowny przydział magistrali będzie uzyskiwał za każdym razem kiedy inni agenci nie starają się o przydział magistrali lub mają niższy priorytet.



Rysunek 5. Centralny arbitraż dostępu do magistrali PCI.

Jeśli żaden inny agent nie korzysta z magistrali lub też inni agenci nie żądają jej przydziału magistrala może zostać przydzielona jednemu z agentów 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 trwa ta sytuacja (sygnał **GNT#** na złączu agenta w stanie niskim), rozpoczęcie przez niego pojedynczej operacji nie musi być poprzedzone przejściem jego sygnału **REQ#** w stan niski.



Rysunek 6. Zależności czasowe podczas arbitrażu

Na rysunku 6 przedstawiono przykładowy przebieg operacji arbitrażu. Agenci A i B ubiegają się o przydział magistrali:

- Agent A już wcześniej zażądał magistrali (ma sygnał **REQ#-a** ustawiony w stan niski).
- Podczas narastającego zbocza 1. impulsu zegarowego sygnał został odczytany przez układ arbitrażu. Przed przyjściem drugiego impulsu zegarowego układ arbitrażu, 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 aktywny. Operacja rozpoczyna się podczas impulsu zegarowego 3 i kończy wraz z 4. impulsem zegarowym (pojedyncze przesłanie danych). Agent A zamierza jeszcze przeprowadzić następne operacje - pozostawia więc sygnał **REQ#-a** w stanie aktywnym.
- Podczas narastającego zbocza 3. impulsu zegarowego przez układ arbitrażu odbierane są dwa zgłoszenia **GNT#-a** i **GNT#-b**. Zezwolenie dostaje agent B (podczas 4. impulsu zegarowego sygnał **GNT#-b** jest aktywny). Agent B czeka na przejście magistrali w stan jałowy.
- Agent A kończy swoją 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. Operacja prowadzona przez agenta B zostaje zakończona podczas impulsu zegarowego 7.
- Przed zakończeniem swojej operacji agent B zniósł sygnał zgłoszenia. Oznacza to, że potrzebował magistrali do przeprowadzenia tylko jednej operacji. W następnym kroku, jeśli nie będzie innych zgłoszeń, układ arbitrażu ponownie przydzieli magistralę agentowi A (jego sygnał **REQ#-a** pozostaje cały czas w stanie niskim).

4.4 Rozkazy magistrali PCI

W systemie PCI występują trzy przestrzenie adresowe:

1. danych,
2. we-wy,
3. konfiguracyjna.

W przesyłanym w pierwszej fazie transmisji danych (fazie adresowej), po liniach **C/BE[3::0]**, kodzie operacji zawarta jest informacja o tym, do której z przestrzeni adresowych odnosi się wystawiony równolegle na liniach **AD[31::00]** adres. Są też dwa rozkazy służące do inicjowania operacji bezadresowych (informacja kierowana jest do wszystkich urządzeń).

Wszystkie urządzenia muszą odpowiadać na rozkazy odnoszące się do przestrzeni adresowej konfiguracji. Pozostałe rozkazy nie są obowiązkowe.

Zestawienie kodów i odpowiadających im typów operacji przedstawiono w tabeli na następnej stronie. Na magistrali jedyńki reprezentowane są przez stan wysoki a zera przez stan niski.

	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

Potwierdzenia 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 pracy systemu”) 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 do 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 sekwencyjnym 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.

4.5 Operacje w przestrzeni adresowej pamięci

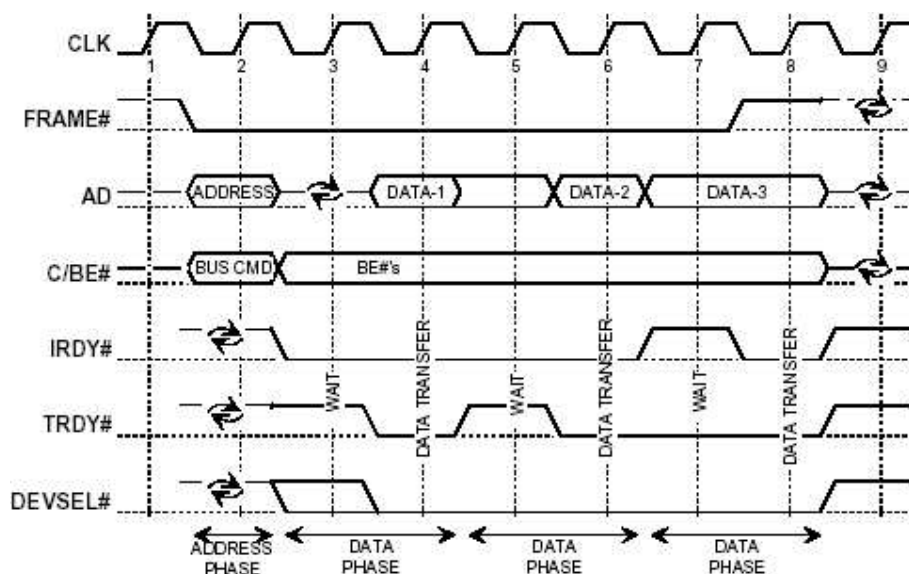
Podobnie jak w przypadku dostępu do rejestrów konfiguracji operacje wykonywane w przestrzeni adresowej pamięci wymagają sprawdzenia bitów odebranych z linii **AD[1::0]**. W zależności od ich wartości wykonywana jest operacja transmisji blokowej lub cykl jest kończony po pierwszej fazie przesyłania danych.

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]**.

Na rysunku 7 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 wysterowane impulsami zezwalającymi trwającymi od pierwszego impulsu zegarowego podczas fazy przesyłania danych, aż do zakończenia operacji.



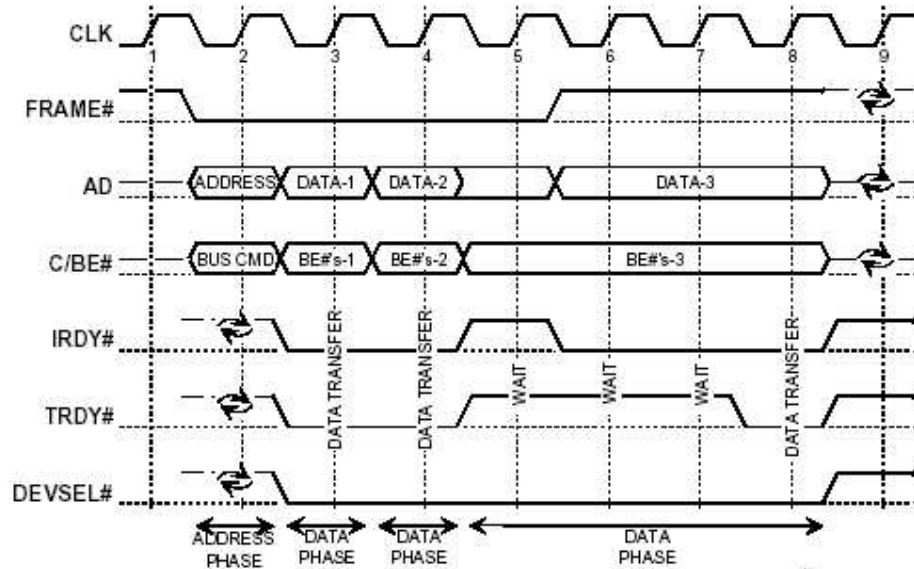
Rysunek 7. Operacja odczytu z pamięci

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. Bufory wyjściowe muszą być wysterowane impulsami zezwalającymi, które powinny pozostać stałymi, aż do końca operacji (żeby uniknąć fluktuacji sygnałów na liniach **AD#**).

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 8 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.



Rysunek 8. Operacja zapisu do pamięci

W przykładzie pokazanym na rysunku 8 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.

4.7 Dostęp wyłączny

W systemie wieloprocesorowym, w którym może dojść do kolizji podczas modyfikacji wspólnych zasobów danych, niezbędny jest mechanizm porządkujący dostęp procesorów do wspólnej pamięci. W systemach, w których zasoby mogą być dzielone między kilka procesorów, moduł nadrzędny może potrzebować dostępu wyłącznego do fragmentu pamięci modułu docelowego na czas przeprowadzenia tzw. krytycznej sekwencji operacji. Dla innych agentów w tym czasie dostęp do tego fragmentu pamięci powinien być zablokowany. W celu zablokowania dostępu do fragmentu pamięci modułu docelowego wykorzystuje się sygnał magistrali PCI **LOCK#**.

Agent żądający dostępu wyłącznego powinien zagwarantować sobie prawo sterowania sygnałem **LOCK#**. Przed wysłaniem sygnału **REQ#** sprawdza stan sygnału **LOCK#**. Sygnał **LOCK#** jest 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 poczekać aż się zwolni (opóźnić wysłanie sygnału **REQ#**). Jeśli **LOCK#** jest zwolniony a agent żądający dostępu

wyłącznego uzyskał dostęp do magistrali, to może przejść nad nim kontrolę. Pozostali agenci nie mogą teraz sterować sygnałem **LOCK#** nawet, jeśli uzyskali dostęp do magistrali.

Mimo, że jeden z agentów kontroluje sygnał **LOCK#**, pozostali mogą wykonywać operacje na magistrali PCI pod warunkiem, że nie będą w nich odwoływać się do zablokowanych zasobów. W danej chwili blokadą dostępu może być objęty tylko jeden fragment pamięci.

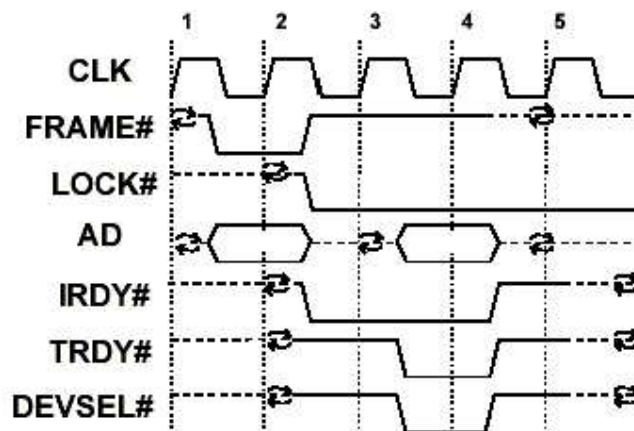
W ostatniej wersji systemu PCI (Revision 2.2) praktycznie zarzucono stosowanie dostępu wyłącznego. Może on być stosowany jedynie w przypadku kiedy jakaś „następna” magistrala rozszerzeń (patrz rysunek 1) wymaga zastosowania tego mechanizmu. Wtedy do honorowania tego sygnału, przekazywanego za pośrednictwem mostów łączących magistrale rozszerzeń, zobowiązany jest jedynie most-sterownik magistrali PCI.

Moduł inicjujący dostęp wyłączny powinien przestrzegać następujących zasad:

1. Pierwszą operacją w sekwencji objętej wyłącznym dostępem musi być odczyt z pamięci (rysunek 9).
2. Sygnał **LOCK#** ustawiany jest w fazie transmisji danych (poczynając od pierwszego zbocza zegara następującego po fazie adresowej).

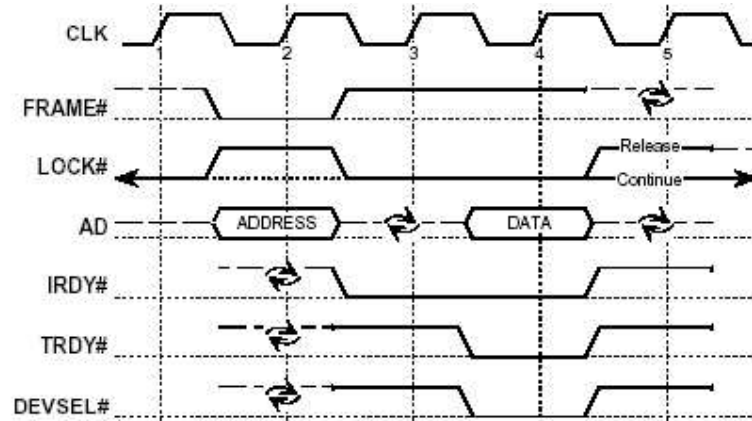
Moduł docelowy obowiązują następujące reguły:

1. Most, do którego kierowana jest operacja dostępu wyłącznego blokuje swoją pamięć jeśli sygnał **LOCK#** zdjęty w fazie adresowej jest aktywny w fazie transmisji danych.
2. Dany fragment pamięci, pozostaje zablokowany dopóki nie zostanie wykryty, w narastającym zboczu zegara, brak aktywnych sygnałów - zarówno **LOCK#** jak i **FRAME#**.
3. Most, w czasie kiedy jego pamięć jest zablokowana, nie powinien akceptować żadnego żądania dostępu za wyjątkiem tych, które kierowane są przez moduł aktualnie sterujący sygnałem **LOCK#**.



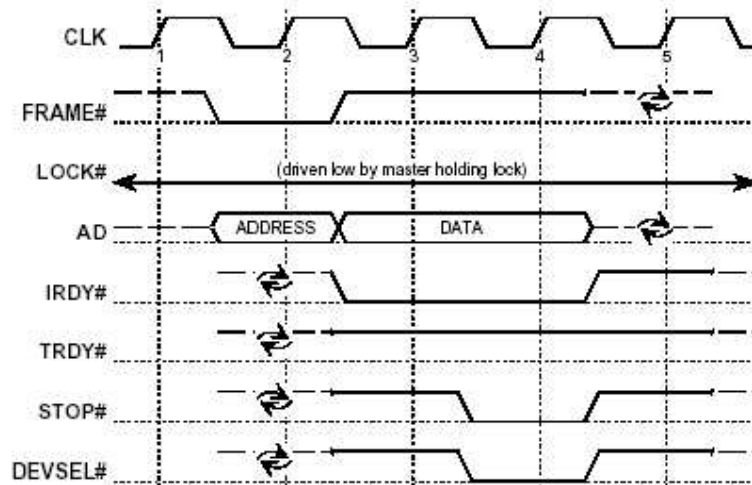
Rysunek 9. Operacja rozpoczynająca dostęp wyłączny

Na rysunku 9 przedstawiono przykład operacji (odczyt z pamięci) rozpoczynającej dostęp wyłączny. Zablokowanie pamięci nastąpiło w momencie zakończenia pierwszej fazy transmisji danych.



Rysunek 10. Kontynuacja dostępu wyłącznego

Od tego momentu do zablokowanych zasobów dostęp może uzyskać tylko agent, który steruje sygnałem **LOCK#** tak, że jest on znoszony na czas trwania fazy adresowej (zobacz rysunek 10). Próba dostępu do zablokowanej pamięci podjęta przez innego agenta kończy się tak jak na rysunku 11. Ponieważ w fazie adresowej sygnał **LOCK#** znajdował się w stanie niskim zaadresowany moduł kończy operację wystawiając sygnał **STOP#** (bez sygnału gotowości **TRDY#**).



Rysunek 11. Próba dostępu do zablokowanych zasobów

4.8 Wykorzystanie pamięci podręcznej

Norma PCI umożliwia współpracę bloku pamięci DRAM na magistrali PCI a 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 śledzenie dostępu do magistrali zostało zakończone. Sygnał **SBO#** przechodzi w stan niski, kiedy bieżący dostęp do pamięci dotyczy adresów, w których przechowywane są dane nieaktualne. Kiedy sygnał **SBO#** oraz sygnał **SDONE** są w stanie wysokim, oznacza to że śledzenie zostało zakończone, a 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 Rozkazy konfiguracyjne

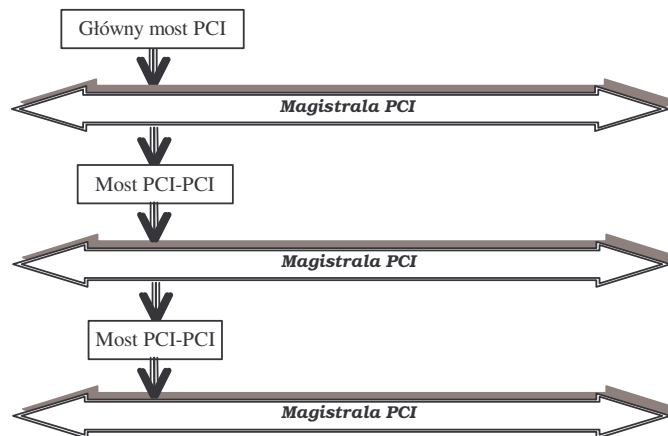
Zakłada się, że urządzenia PCI będą konfigurowane a ich praca inicjowana programowo. Przewidziano do tego oddzielną konfiguracyjną przestrzeń adresową. Każde urządzenie PCI powinno posiadać 256 bajtów w przestrzeni rejestrów konfiguracji. W przypadku dostępu do przestrzeni adresowej przeznaczonej dla potrzeb konfiguracji dekodowanie adresu odbywa się poza magistralą a moduł wybierany jest przy pomocy indywidualnego sygnału **IDSEL**. 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 12. Struktura adresu (typu 0) przy dostępie do rejestrów konfiguracji

W systemach z wieloma magistralami PCI połączonymi np. tak jak na rysunku 13, kierowana do odpowiedniego mostu PCI-PCI, informacja adresowa (dla przestrzeni rejestrów konfiguracji) ma bardziej złożoną postać.



Rysunek 13. Wielopoziomowa konfiguracja magistral PCI

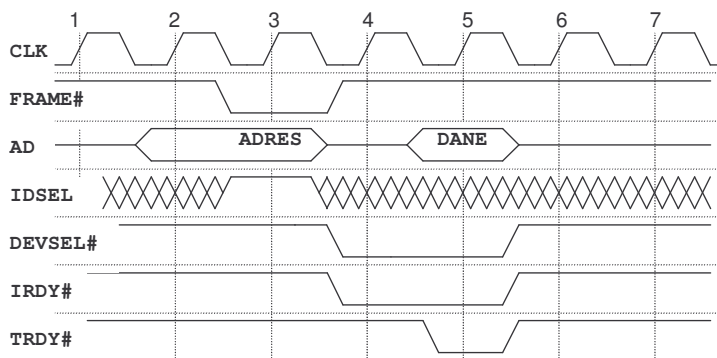
W tym przypadku przekazywany jest numer magistrali i urządzenia, które następnie ma być wybrane.

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

Rysunek 14. 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]** i wystawianie (na niższej magistrali) odpowiedniej linii **IDSEL** i przesłanie na magistralę informacji adresowej okrojonej do postaci typu 0 (rysunek 12).

Sposób sterowania linią **IDSEL** znajduje się w gestii odpowiedniego mostu. W niektórych rozwiązaniach występuje długo utrzymujący się nieustalony poziom logiczny sygnału wyjściowego (na rysunku 15 oznaczony „XXXXXXX”). Dlatego dopuszcza się wcześniejsze wystawianie adresu (tak jak na rysunku 15) aby dać czas na ustabilizowanie się dekodowanego 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 15. 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]**).

4.11 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 w toku

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.

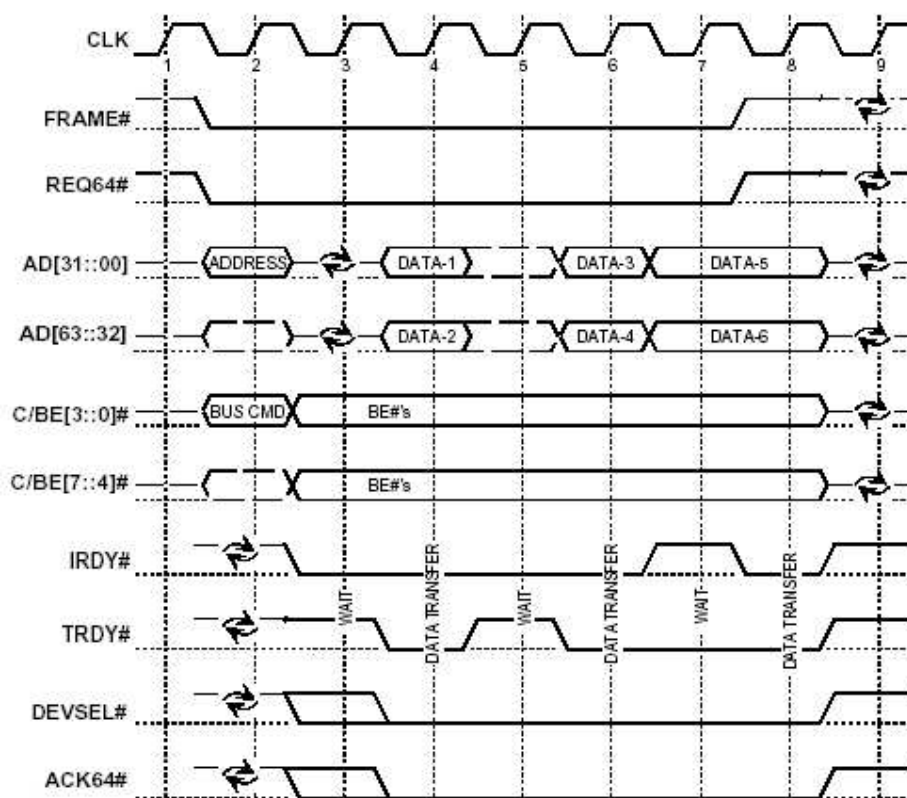
Do sygnalizacji błędów są wykorzystywane dwa sygnały: **PERR#** oraz **SERR#**. Pierwszy jest używany wyłącznie dla kontroli parzystości podczas wszystkich operacji, z wyjątkiem rozkazów specjalnych. Protokół zapewnia, że tylko jeden agent w danej chwili może ustawić ten sygnał w stan niski. Powinien 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. W

normie zaleca się, 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

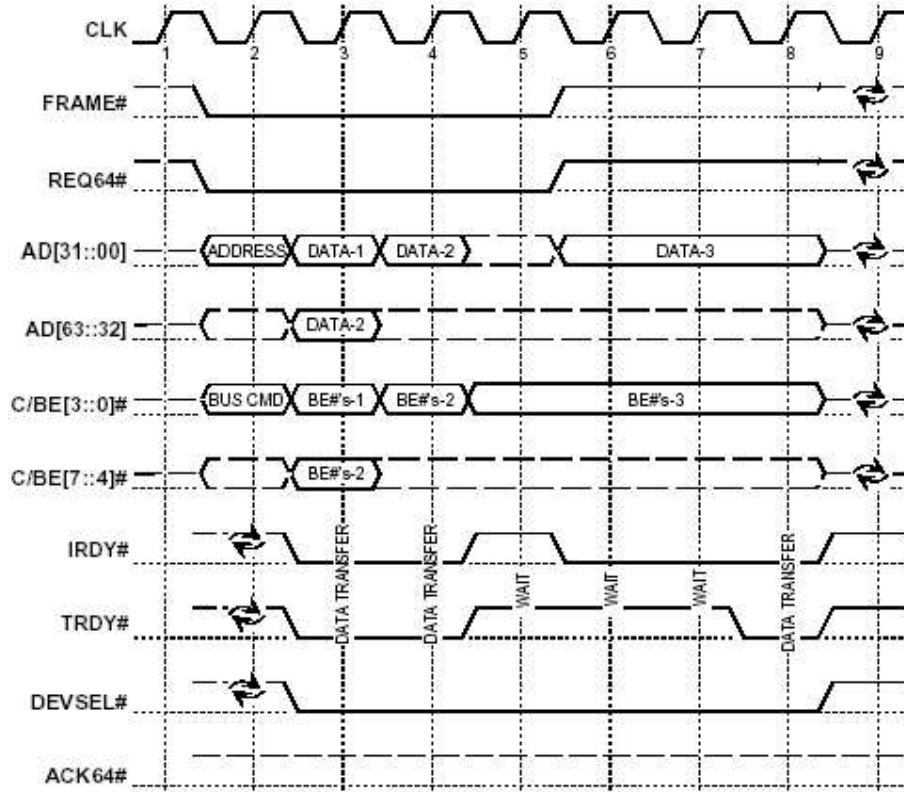
Rozszerzenie magistrali do 64 bitów wymaga 39 dodatkowych wyprowadzeń dla sygnałów: **REQ64#**, **ACK64#**, **AD[63::32]**, **C/BE[7::4]** oraz **PAR64**. Operacje 64-bitowe są za każdym razem negocjowane. Blok nadrzędny wystawia **REQ64#**. Moduł docelowy, jeśli jest przygotowany do 64-bitowej operacji,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



Rysunek 16. 64-bitowa operacja odczytu

Na rysunku 16 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.



Rysunek 17. Operacja zapisu zgłoszona jako 64.bitowa a zrealizowana jako 32.bitowa

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. (Zobacz rysunek 17)