

Funkcje systemu operacyjnego

wspomaganie użytkownika (ang. user functions)

- sterowanie i utrzymanie kontroli nad programem (ang. *program control*)
- obsługa wejścia/wyjścia (ang. *I/O handling*)
- obsługa plików (ang. *file system manipulation*)

funkcje systemu (ang. system functions)

- zarządzanie pamięcią (ang. *memory management*)
- ochrona zasobów (ang. *resource protection*)
- przydział zasobów (ang. *resource allocation*)
- obsługa wyjątków (ang. *exception handling*)
- harmonogramowanie (ang. *scheduling*)
- raportowanie (ang. *accounting*)

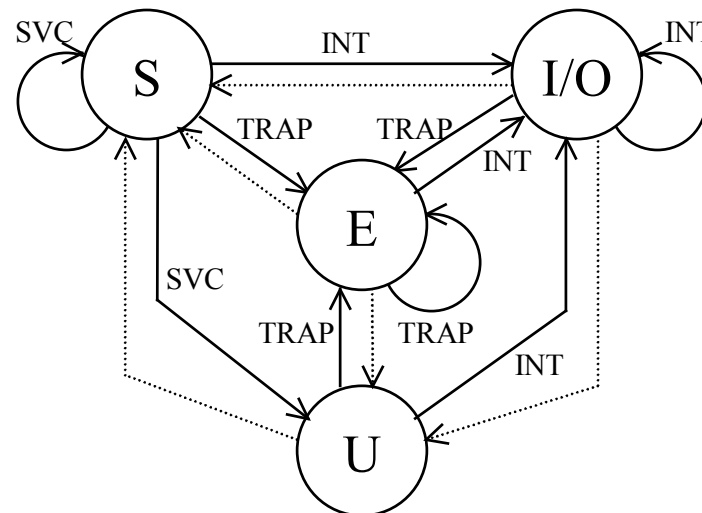
System z podziałem czasu (ang. *time sharing*)

– każde działanie jest *procesem* lub jego częścią

Istotność procesu i schemat przełączania

Poziomy uprzywilejowania (istotności) procesów

- obsługa wyjątków (ang. *exception handling*) – utrzymanie integralności systemu
- obsługa we/wy (ang. *I/O handling*) – funkcje krytyczne względem czasu
- zadania nadzoru (ang. *supervisor functions*) – zarządzanie procesami i pamięcią,
- zadania użytkowników (ang. *user jobs*)



- zdarzenia asynchroniczne – przerwanie zewnętrzne (INT)
- zdarzenia synchroniczne – wywołanie systemowe (SVC), pułapka (TRAP)

Przerwania i wyjątki

Przerwanie – (ang. *interrupt*) sygnalizowane zdarzenie **w środowisku procesora** wymagające obsługi programowej, najczęściej obsługa zewnętrznych urządzeń wejścia/wyjścia (ang. *input/output*)

Wyjątek – (ang. *exception*) sygnalizowane **zdarzenie podczas wykonania instrukcji** wymagające obsługi programowej (najczęściej błąd)

„Przerwanie” programowe – instrukcja procesora: wywołanie procedury systemowej, którą rozpoczyna automatyczne przechowanie słowa stanu procesora

Przerwania precyzyjne

- wewnętrzne (wyjątki, niesprawności, błędy wykonania) → różne efekty
- zewnętrzne (zdarzenia w środowisku) → zwłoka obsługi (ang. *interrupt latency*)

Przerwania nieprecyzyjne (zwykle niemaskowalne) – bez możliwości odtworzenia stanu procesora po zakończeniu obsługi, spowodowane błędem krytycznym

Obsługa przerwań

Priorytet obsługi – zależy przede wszystkim od jego pilności (ang. *urgency*):

- (H) spowodowane naruszeniem bezpieczeństwa lub błędem sprzętowym
- (M) krytyczne ze względu na czas obsługi (na przykład transmisja danych)
- (L) spowodowane obniżeniem przepustowości systemu lub związane z obsługą zdarzeń, których ważność jest określona przez użytkownika.

Przerwania nieprecyzyjne (zwykle niemaskowalne) – najwyższy priorytet (H)

Przerwania precyzyjne

– *wewnętrzne (błędy wykonania)* – priorytet H lub M

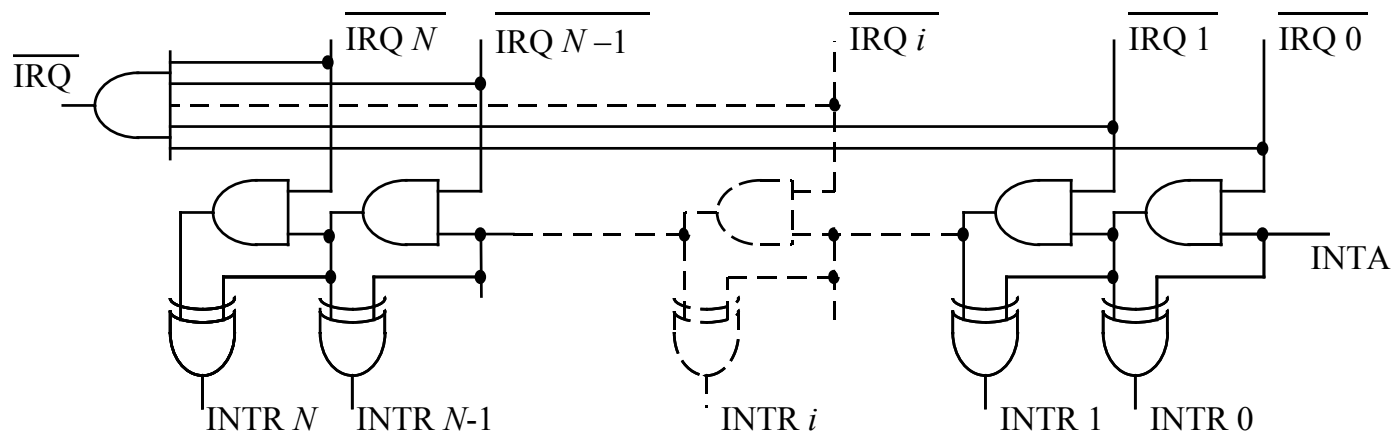
- instrukcja została *zakończona* (ang. *completed*), lecz wytworzony wynik jest błędny (na przykład wykryto nadmiar)
- instrukcja została *pominięta* (ang. *supressed*), bo naruszono reguły ochrony
- instrukcja została *zignorowana* (ang. *nulified*) – wykonanie było niemożliwe, lecz po usunięciu przyczyny możliwe jest powtórzenie działania
- instrukcja została *wstrzymana* (ang. *terminated*).

– *zewnętrzne (zdarzenia w otoczeniu)* → priorytet L lub M, zwykle obsługa we/wy

Identyfikacja przerwania – odpytywanie

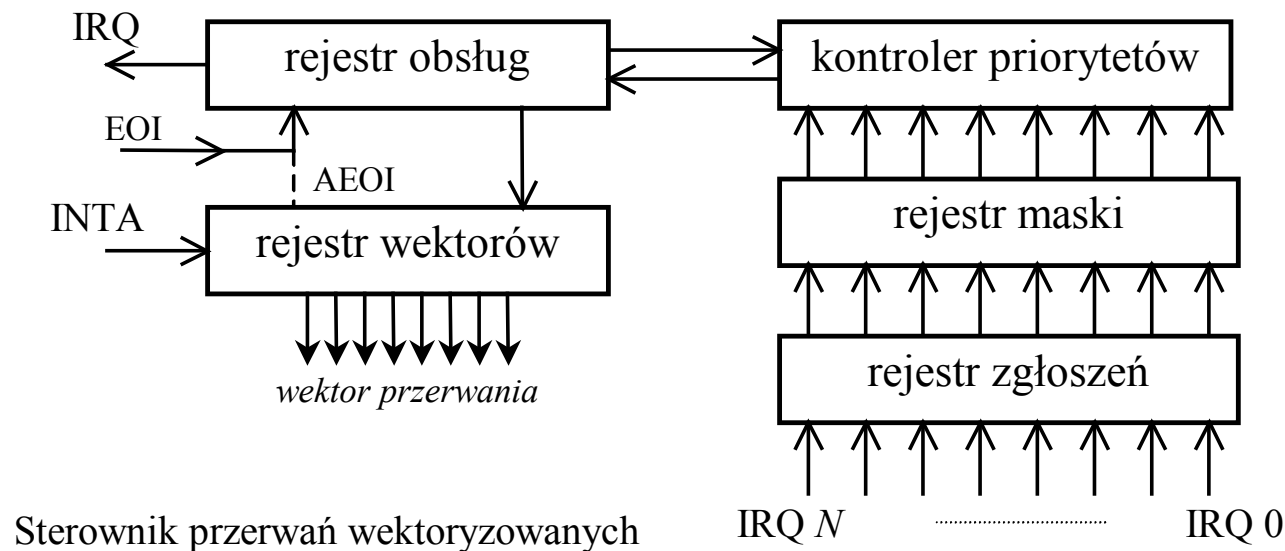
Identyfikacja źródła przerwania

- odpytywanie (ang. *polling*) – ciągle zaangażowanie procesora
- samoidentyfikacja
 - ustanowienie sztywnego łańcucha priorytetów zgłoszeń (ang. *daisy-chain*)
 - wektoryzacja przerw



Łańcuch zgłoszeń z ustalonymi priorytetami typu *daisy-chain*.

Identyfikacja przerwania – wektoryzacja



- zgłoszenie: zbocze impulsu (opadające) lub poziom sygnału IRQ#
- maska: selektywne blokowanie zgłoszeń
- priorytety: ustalanie priorytetu: zwykły, z rotacją, z rotacją specjalną
- obsługa: wykaz przerwń w trakcie obsługi (ang. *pending interrupt*)
- wektor: kod identyfikujący procedurę obsługi w *tablicy przerwń* (ang. *interrupt table*) w pamięci operacyjnej komputera

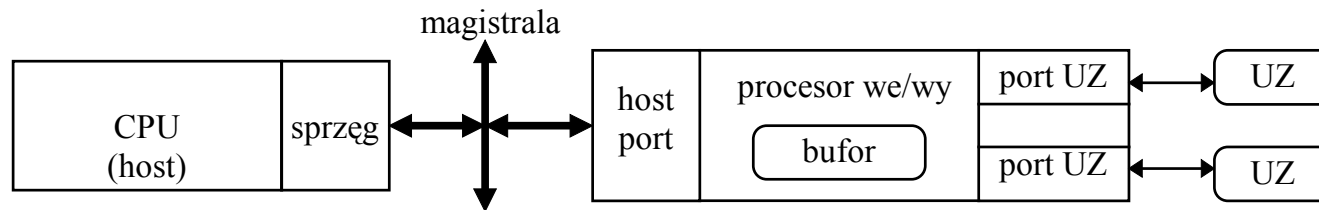
Identyfikacja przerwania – dystrybucja

Potrzeba równomiernego obciążenia zadaniami obsługi procesorów w systemie wieloprocessorowym

Protokół przerwania (schemat działania):

- sekwencja programowania sterownika
- sekwencja zgłoszenia
 - identyfikator źródła przerwania
 - adresowanie procesora obsługującego
 - o geograficzne – jawnie wskazany procesor obsługujący
 - o logiczne i rozgłaszanie (ang. *broadcasting*) – obsługę podejmuje jeden z mniej obciążonych procesorów
 - kody kontrolne
- sekwencja zakończenia
 - identyfikator źródła przerwania
 - kod zakończenia (ang. *End Of Interrupt*, EOI)
 - kody kontrolne

Obsługa wejścia i wyjścia



Podsystem wejścia/wyjścia

Urządzenia we/wy:

- magazynujące (ang. *storage*) – przechowywanie danych
 - pamięci wtórne i tercjalne, archiwizery
- gromadząco-rozsyłające (ang. *source/sink*) – konwersja i rozsyłanie danych
 - komunikacja człowieka z komputerem: mysz, klawiatura, monitor ekranowy, drukarka
 - sprzęg procesów przemysłowych z komputerem: czujnik, regulator, konwerter A/C i C/A, sterownik
 - komunikacja między inteligentnymi terminalami: modem, łącze sieci komputerowej, łącze transmisji szeregowej lub równoległej.

Urządzenia wejścia i wyjścia

Cechy eksploatacyjne urządzeń zewnętrznych:

- czas dostępu (ang. *access time*), zwłoka dostępu (ang. *latency*) – znacznie ($>10^5$ razy) większe niż dla pamięci (dostęp sekwencyjny)
- przepustowość (ang. *bandwith*) – maksymalna liczba danych przesyłanych w kwancie czasu (transmisja równoległa z szybkością GB/s).
- ryzyko błędu (ang. *error rate*) – średnie ryzyko błędu ($<10^{-6}$)

Sterowanie urządzeniami (mikrokontrolery)

- programowanie sterownika (wysyłanie poleceń)
- testowanie sterownika
- obsługa przerw sygnałizowanych przez urządzenia
- obsługa błędów urządzeń.

Sterowniki

- łącza bezpośredniego (COM, LPT) – sprzęg (ang. *interface*) we/wy
 - szeregowego (protokół RS232, RS485, USB, FireWire)
 - równoległego (protokół Centronics)
- magistral dedykowanych lub współdzielonych.

Oprogramowanie we/wy

Struktura warstwowa (ang. *software layers*)

- „uchwyty przerwań” (ang. *interrupt handlers*) – procedury obsługi zgłoszeń
- sterowniki urządzeń (ang. *device drivers*) – działania specyficzne dla urządzeń (translacja poleceń logicznych („odczytaj blok danych”) na zestaw poleceń fizycznych („odczytaj sektor N na ścieżce P w dysku D”)):
 - buforowanie (kolejkowanie) poleceń logicznych
 - zmiana porządku poleceń w celu poprawy przepustowości transmisji
 - zmiana logicznych adresów urządzeń i bloków danych na adresy fizyczne
 - obsługa błędów transmisji.
- funkcje użytkownika – oprogramowanie niezależne od urządzeń:
 - zmiana symbolicznej nazwy urządzenia (standardowe wejście i wyjście, drukarka) na nazwę logiczną (adres bloku sterującego procesu obsługi)
 - przeformatowanie danych na postać wymaganą przez urządzenie (buforowanie, upakowanie, rozpakowanie, translacja kodu danych)
 - przydział i zwolnienie przydziału pamięci.
- uaktywnienie (proces użytkownika) – określenie parametrów transmisji, przygotowanie danych i zainicjowanie komunikacji we/wy.

Procesy wejścia i wyjścia

Sterowniki urządzeń (ang. *device drivers*) – zasoby chronione

Funkcje *obsługi wejścia/wyjścia* – osobne procesy na poziomie nadzoru.

Proces obsługi wejścia lub wyjścia:

- program wykonywany przez CPU
- program wykonywany przez sterownik

Klasyfikacja

- bezpośrednie we/wy (ang. *direct I/O*) – funkcje sterownika wykonuje procesor: testowanie statusu urządzenia (ang. *busy waiting*) i nadzór wykonania poleceń
- nakładane (ang. *overlapped*) we/wy – obsługa w trybie przerwań precyzyjnych, wymaga intensywnej synchronizacji
- autonomiczne (ang. *autonomous*) we/wy – bezpośredni dostęp do pamięci (ang. *Direct Memory Access, DMA*), wymaga minimum synchronizacji.

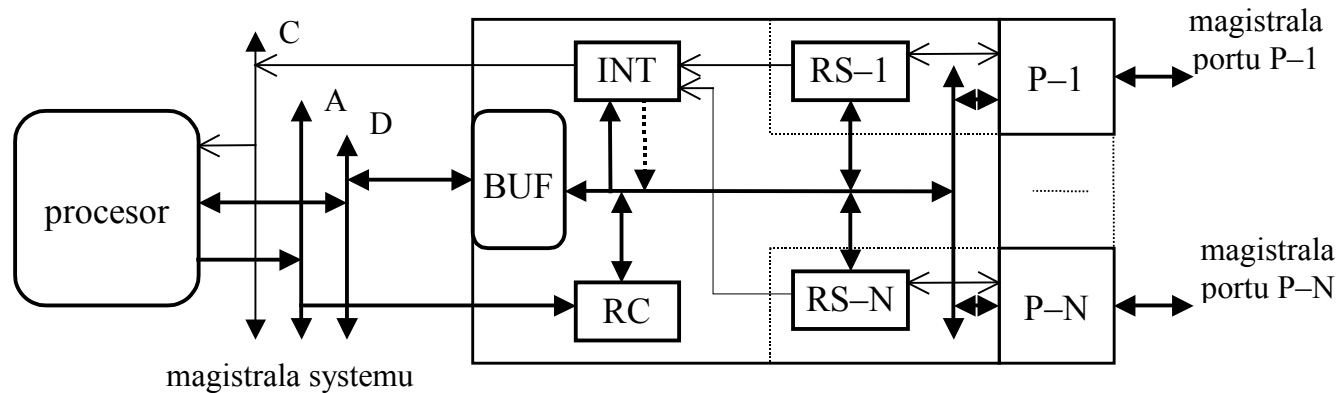
Nakładane we/wy (overlapped I/O)

Synchronizacja obsługi we/wy w trybie przerwania sygnalizujących:

- gotowość peryferala (urządzenia we/wy) do transmisji
- zakończenie operacji
- wystąpienie wyjątku sygnalizowanego (na przykład błąd transmisji).

Procesy we/wy są niezależne:

- jednoczesna obsługa wielu urządzeń, problem identyfikacji źródła
- konieczność przełączania procesów

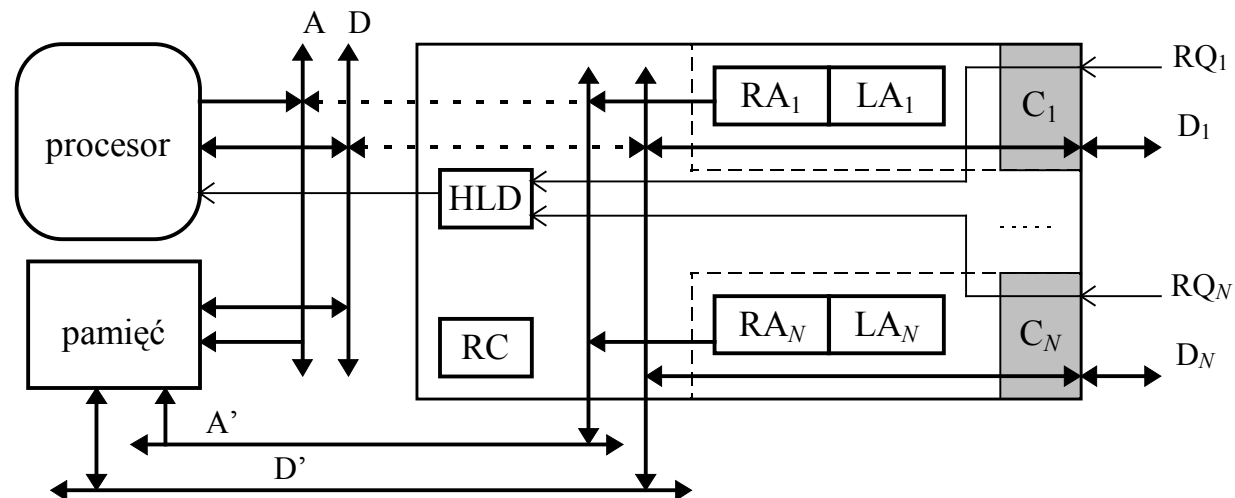


Realizacja nakładanego we/wy

Autonomiczne we/wy

Transmisja bloku danych zamiast transmisji danych pojedynczych:

- zmniejszenie częstości synchronizowania procesów
- redukcja narzutów czasowych synchronizowania
- konieczne bufory danych – najprościej w pamięci głównej
→ transmisja z pominięciem CPU (DMA)



Realizacja autonomicznego we/wy

Bezpośredni dostęp do pamięci (DMA)

Obsługa w trybie bezpośredniego dostępu do pamięci

- programowanie procesora DMA – podanie parametrów transmisji: adresu źródłowego i docelowego danych, rozmiaru bloku i protokołu transmisji
- zainicjowanie transmisji – skutek zgłoszenia żądania transmisji przez kanał DMA (procesor DMA ma zwykle kilka niezależnych kanałów).

Procedura:

- zgłoszenie w *kanale* DMA żądania transmisji (ang. *bus request, hold*)
- potwierdzenie udostępnienia magistral (ang. *bus grant, hold acknowledge*)
- wykonanie transferu DMA i zwolnienie magistral.

Transfer pojedynczy (ang. *single-cycle DMA*) – wykradanie cykli (ang. *cycle-stealing*)

Transfer blokowy (ang. *burst-mode DMA*).

- blokowanie dostępu do magistral –pamięci dwuportowe
- różny rozmiar bloków danych w urządzeniach uczestniczących
 - przerwanie po każdym transferze bloku
 - powiązanie danych (ang. *data chaining*) – autoprogramowanie kanału DMA
→rejestr powiązań (ang. *data chain register, DCR*) – adres kolejnego bloku

Systemy magistral (1)

Magistrala (ang. *bus*) – zestaw linii połączeń między elementami systemu cyfrowego

- magistrala równoległa – sygnały przesyłane jednocześnie i niezależnie
 - połączenia elementów komputera
- magistrala szeregową – sygnały przesyłane sekwencyjnie przez jedną linię
 - transmisja między komputerami

Klasyfikacja magistral

- lokalna (ang. *local*) – dostosowana do procesora
 - połączenie procesor – pamięć podręczna / lokalne sterowniki we/wy
 - połączenia głównych bloków w obrębie płyty (ang. *board level*)
- systemowa (ang. *system*) – standardowa, np. VME, Multibus
 - połączenia wszystkich modułów systemu
 - połączenia modułów wymiennych (ang. *backplane level*),.
- sprzęgająca (ang. *interface*) – standardowa, np. SCSI, PCI
 - przyłączenie pamięci masowych i urządzeń we/wy za pomocą sterowników (SCSI – *Small Computer System Interconnect*)
 - połączenie urządzeń zewnętrznych (PCI, *Peripheral Component Interconnect*)

Systemy magistral (2)

Magistrale specjalizowane (ang. *dedicated*)

- pojedyncze przesłanie między procesorem i pamięcią lub we/wy
- transfer informacji jednego typu (adresu, polecenie, dane)
- duża przepustowość, wysoki koszt, duża podatność na błędy.

Magistrale współdzielone (ang. *shared*)

- mała przepustowość, niski koszt, możliwa korekcja błędów
- możliwość konfliktu dostępu, konieczny arbitraż

Magistrale dzielone strukturalnie – zestaw magistral specjalizowanych

- procesor – domniemany zarządca magistrali (ang. *bus master*)
- dominujące przesłania procesor–pamięć

Magistrale dzielone funkcjonalnie

- zorientowane na przesłanie wiadomości (ang. *message*) – bloków danych wraz z informacją adresową, statusową oraz kontrolną (korekcja błędów)
- standardy transmisji są niezależne od sprzętu
- charakterystyka – przepustowość (ang. *throughput*).

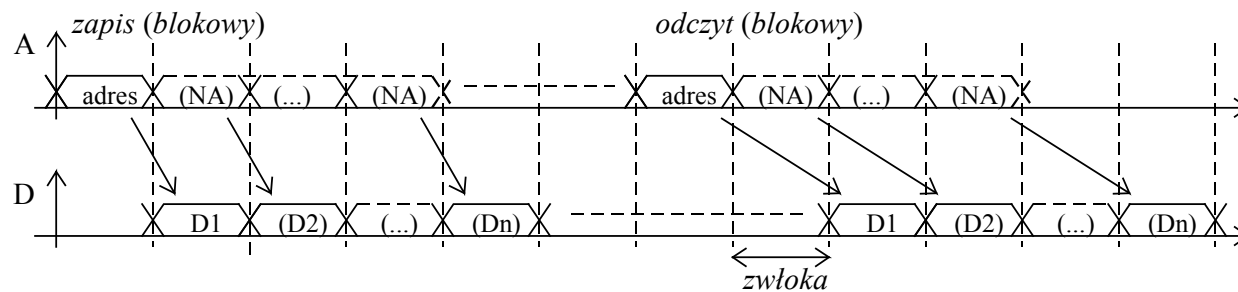
Transakcje (1)

Transakcja (ang. *bus transaction*) – przesłanie po magistrali

- żądanie dostępu (ang. *request*)
- arbitraż (ang. *arbitration*)
- zaadresowanie (ang. *addressing*)
- przesłanie danych (ang. *data transfer*)
- detekcja i sygnalizacja błędów (ang. *error detection and signalling*).

Magistrale niemultipleksowane – transakcje (pojedyncze lub blokowe):

- *zapis* danych – zarządca magistrali wysyła w tym samym cyklu adres i dane
- *odczyt* danych – zarządca magistrali wysyła tylko adres

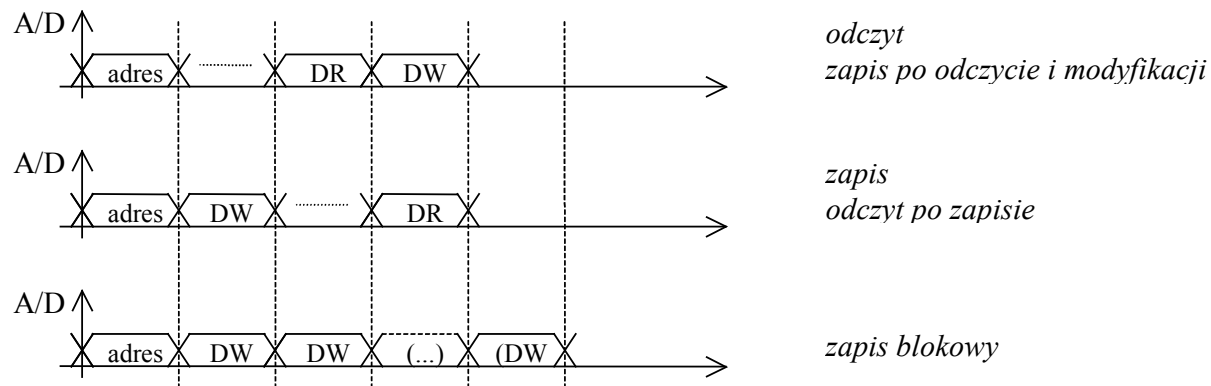


Schemat transakcji na magistrali niemultipleksowanej
(A – linie adresowe, D – linie danych, NA – kolejny adres)

Transakcje (2)

Magistrala multipleksowana – rodzaje transakcji:

- zapis (*write*), pojedynczy lub blokowy (*burst mode*)
- odczyt (*read*), pojedynczy lub blokowy (*burst mode*)
 - transakcja rozdzielona – przełączanie komunikatów (ang. *message switching*)
adres ... oczekiwanie ... zwrotny zapis
 - transakcja nierozdzielona – przełączanie układów (ang. *circuit switching*).
- zapis po odczycie i modyfikacji (ang. *read-modify-write*),
- odczyt po zapisie (ang. *read-after-write*),



Schemat transakcji na magistrali multipleksowanej

Adresowanie obiektów przyłączonych do magistrali

Adresowanie logiczne – arbitralne przypisanie urządzeniu identyfikatora

- wymaga dodatkowych układów adresowych.

Adresowanie geograficzne – rozpoznanie na podstawie fizycznej lokalizacji

- umożliwia programową identyfikację urządzeń przez system operacyjny
- adres urządzenia:
 - identyfikator łącza (ang. *slot number*)
 - identyfikator urządzenia (adres lokalny) na płycie.

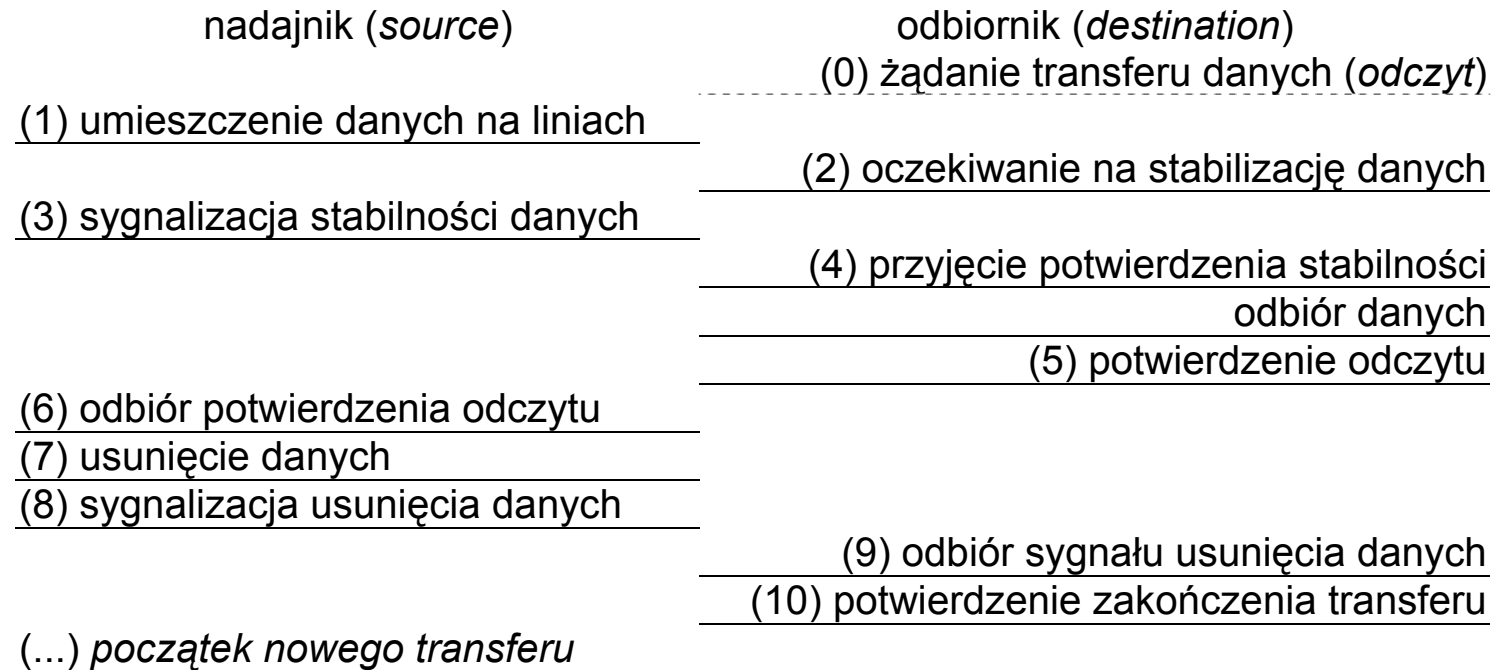
Adresowanie zbiorowe

- rozgłaszanie (ang. *broadcast*) – wysłanie informacji jednocześnie do wielu odbiorców (RESET, aktualizacja pamięci podręcznych kilku procesorów)
- wywołanie (ang. *broadcall*) – identyfikacja źródła lub źródeł sygnału
 - jednoczesny odbiór informacji z wielu źródeł
 - forsowanie stanu: suma lub iloczyn logiczny sygnałów
 - separowanie źródeł sygnałów (zgłoszenie przypisane do linii).

Protokół magistrali

Protokół – reguła transakcji (sposób przesyłania, relacje czasowe sygnałów)

- synchronicznie – ustalone chwile pojawienia się każdego sygnału
- asynchronicznie – ustalone tylko następstwo sygnałów
- półsynchronicznie – ustalone następstwo i chwile wystąpienia sygnałów.

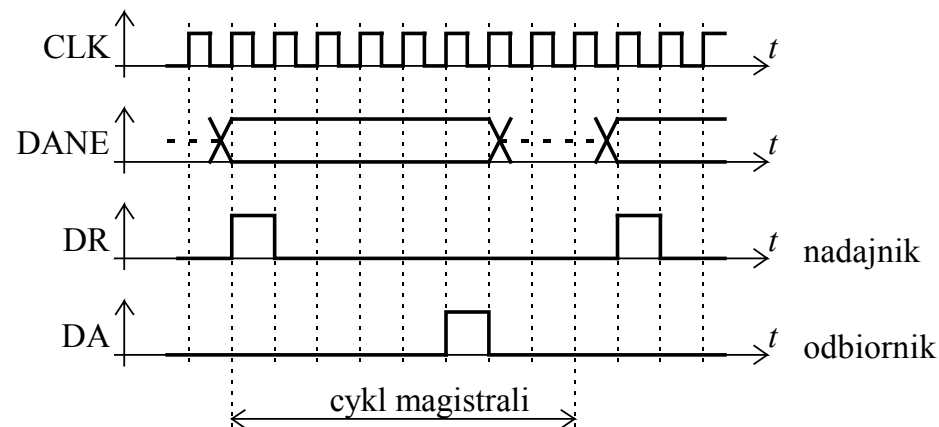


Schemat transakcji

Synchroniczna realizacja protokołu magistrali

Przestania synchroniczne

- *ustalone chwile pojawienia się sygnałów w relacji do impulsów taktujących (CLK), niezależnie od szybkości nadajnika i odbiornika*
 - gotowość danych (data ready – DR)
 - potwierdzenie (data accepted – DA)
- *ustalone parametry elektryczne sygnałów*
- *brak możliwości potwierdzenia odbioru lub sygnalizacji błędów*
- *niezawodność transferu – kodowanie informacji kodami korekcyjnymi*



Protokół synchroniczny: DR –gotowość danych, DA –zakończenie transakcji

Protokoły asynchroniczne

Przesłania asynchroniczne

- *ustalone relacje czasowe sygnałów*
 - gotowość danych (*data ready* – DR)
 - potwierdzenie (*data accepted* – DA)
- *ustalone wybrane parametry elektryczne sygnałów*
- *niezawodność transferu – sygnalizacja błędów*

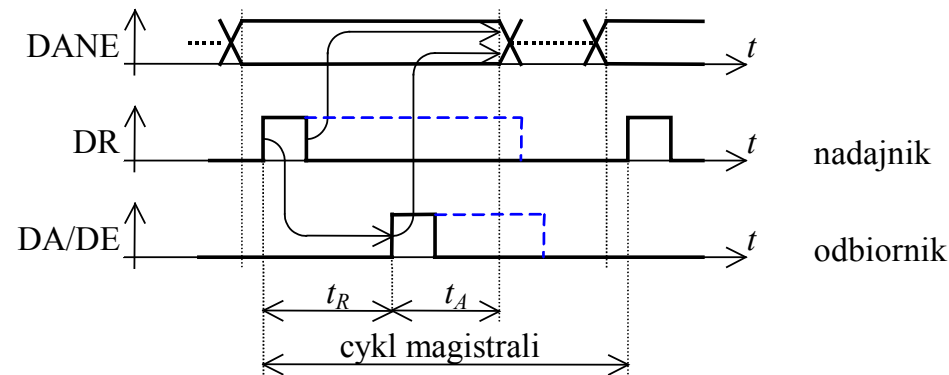
Przesłanie asynchroniczne

- *niepowiązane (ang. non–interlocked)*
 - ustalony czas trwania sygnału gotowości (DR)
 - ustalony czas trwania sygnału potwierdzenia (DA) lub błędu (DE)
- *częściowo powiązane (ang. half–interlocked)*
 - powiązanie sygnału gotowości z sygnałem potwierdzenia
- *w pełni powiązane (ang. fully interlocked) (z potwierdzeniem, ang. handshaking)*
 - wzajemne powiązanie sygnału gotowości i potwierdzenia

Przesłania niepowiązane

Przesłanie niepowiązane

- zależne od szybkości odbiornika opóźnienie potwierdzenia t_R
- zależne od szybkości nadajnika opóźnienie zakończenia t_A



Protokół asynchroniczny – przesłania niepowiązane: DR –gotowość danych, DA/DE – potwierdzenie transakcji lub sygnał błędu

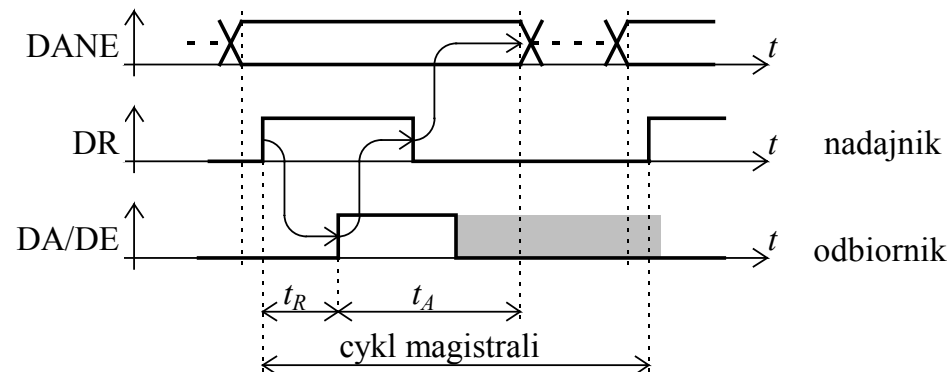
Problem – ustalona szerokość ale zmienny odstęp t_R sygnałów DR i DA

- zbyt szybki nadajnik – usunięcie danych w czasie aktywnego poziomu DR
- zmienny czas reakcji na potwierdzenie odbioru – możliwe zbyt szybkie rozpoczęcie kolejnego transferu i usunięcie danych przy aktywnym DA

Przesłania częściowo powiązane

Wyeliminowana możliwość usunięcia danych podczas aktywnego DR:

- wysłanie nowych danych wymaga wcześniej wygaszenia sygnału DR
- wygaszenie DR możliwe dopiero po uaktywnieniu potwierdzenia DA.
- możliwość zbyt wczesnego rozpoczęcia kolejnego transferu i zmiany danych podczas trwania aktywnego stanu sygnału DA – nadajnik nie otrzymuje potwierdzenia zakończenia odbioru (deaktywacja sygnału DA).



Protokół asynchroniczny – przesłania częściowo powiązane:

DR – gotowość danych, DA/DE – potwierdzenie transakcji / sygnał błędu

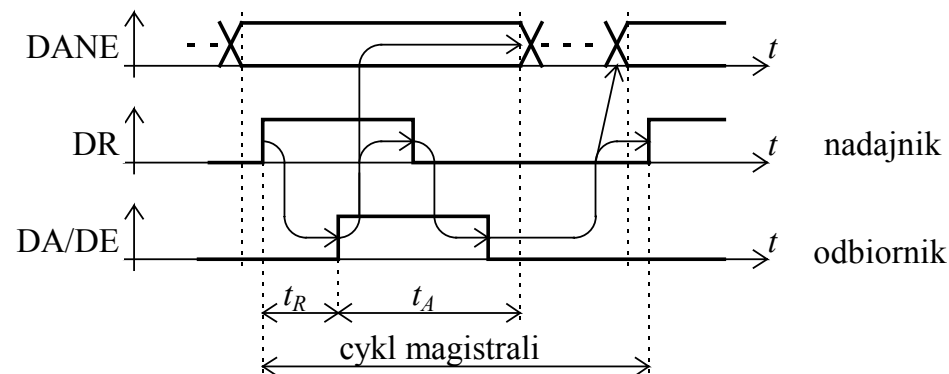
Przesłania z potwierdzeniem (1)

Przesłanie z potwierdzeniem (ang. *handshaking*)

- jeden odbiornik – protokół czteroboczowy
- wiele odbiorników (rozgłaszanie) – protokół sześcioboczowy

Czteroboczowy protokół przesłań w pełni powiązanych (ang. *fully-interlocked*).

- kasowanie sygnału DR przednim zboczem DA/DE
- kasowanie sygnału DA/DE tylnym zboczem sygnału DR
- nowy transfer – po wygaszeniu sygnału potwierdzenia lub błędu DA/DE



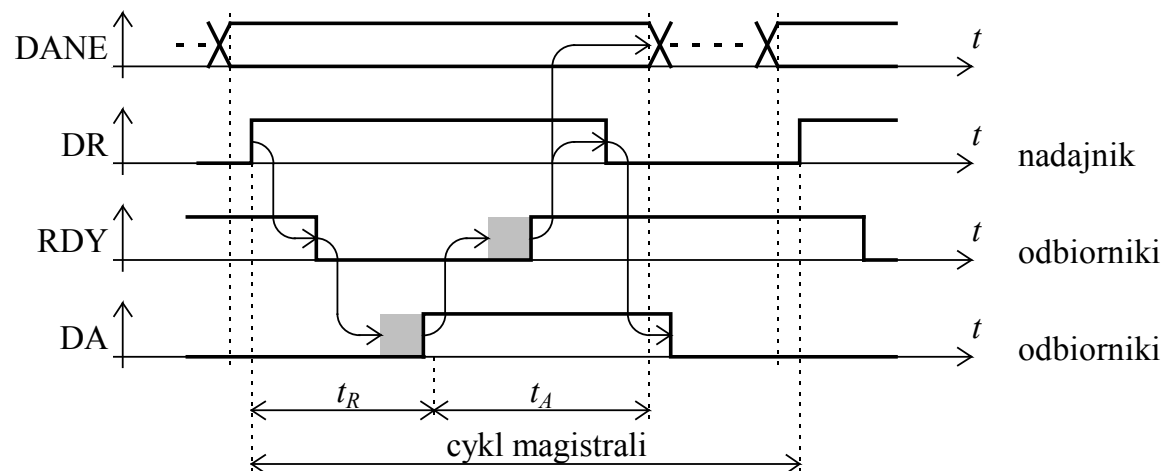
Protokół asynchroniczny – czteroboczowe przesyłanie z potwierdzeniem:

DR – gotowość danych, DA/DE – potwierdzenie transakcji / sygnał błędu

Przesłania z potwierdzeniem (2)

Protokół rozgłaszania (ang. *broadcasting*) z potwierdzeniem (ang. *six-edge handshaking*)

- sygnał ogólnej gotowości RDY (iloczyn logiczny gotowości odbiorników)
- ... $RDY\uparrow \rightarrow DR \rightarrow RDY\downarrow \rightarrow DA = \&DA_i \rightarrow RDY\uparrow$
- sygnał upływu czasu – eliminacja długich okresów oczekiwania na RDY



Protokół asynchroniczny – sześcioboczowe przesyłanie z potwierdzeniem.

DR – gotowość danych, DA – potwierdzenie transakcji

Arbitraż magistrali

Arbitraż statyczny – cykliczne przejmowanie magistrali przez zarządców,

- kwantyzacja dostępu, przesyłanie synchroniczne
- prosta realizacja, duże straty czasu (przydział bez zapotrzebowania)

Arbitraż dynamiczny – przydzielanie magistrali zgodnie z zapotrzebowaniami

- małe straty czasu
- dobór algorytmu przydziału
 - uwzględnianie ważności (priorytet) żądań
 - wykluczenie zablokowania obsługi (zagłodzenia)
 - zasada uczciwości (ang. *fairness*)
 - rotacja priorytetów
 - rotacja karuzelowa (ang. *round-robin*)
 - rotacja automatyczna – cykliczna zmiana po każdej transakcji
- zwalnianie magistrali i przejmowania dostępu
 - na żądanie
 - po wykonaniu operacji
 - odblokowujące (ang. *pre-emption*)

Problemy arbitrażu

Realizowane centralnie lub lokalnie, wymagają trzech sygnałów:

- żądanie dostępu (ang. *bus request/hold*)
- gwarancja (potwierdzenie) dostępu (ang. *bus grant/hold acknowledge*)
- zajętość magistrali (ang. *bus busy*).

Zgłoszenia żądań mogą być

- dzielone (ang. *shared request*)
- przekazywane w łańcuchu zgłoszeń typu *daisy-chain*
- niezależne na osobnych liniach żądań – arbitraż centralny

Niezawodność – zapewnienie satysfakcjonującej obsługi błędów

- wykrywanie niesprawności sprzętu
- sygnalizowanie błędów transmisji

Rodzaje błędów:

- błędy adresowania – naruszenie reguł dostępu lub adres „donikąd”
- błędy danych – kody korekcyjne
- błędy arbitrażu – limitowanie czasu potwierdzenia