



# Architektura komputerów

## Wykład 11

Jan Kazimirski



# Urządzenia wejścia i wyjścia

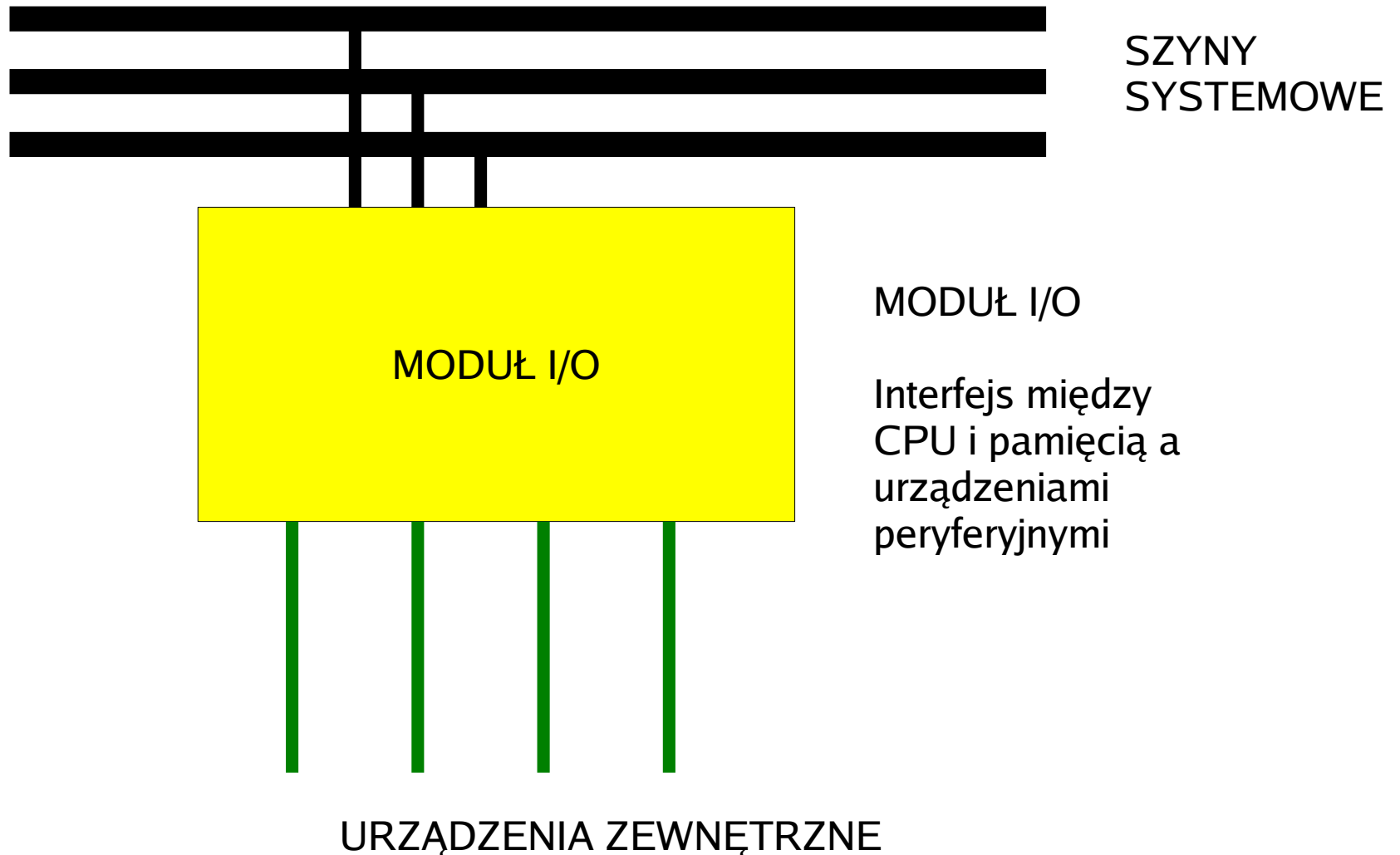


## Urządzenia we/wy

- Trzeci, obok CPU i pamięci, kluczowy element architektury komputera
- Charakterystyka:
  - Duża różnorodność urządzeń
  - Duże różnice w szybkościach
  - Stosowane różnorodne formaty danych
- Konieczność stosowania modułów I/O



## Moduł I/O

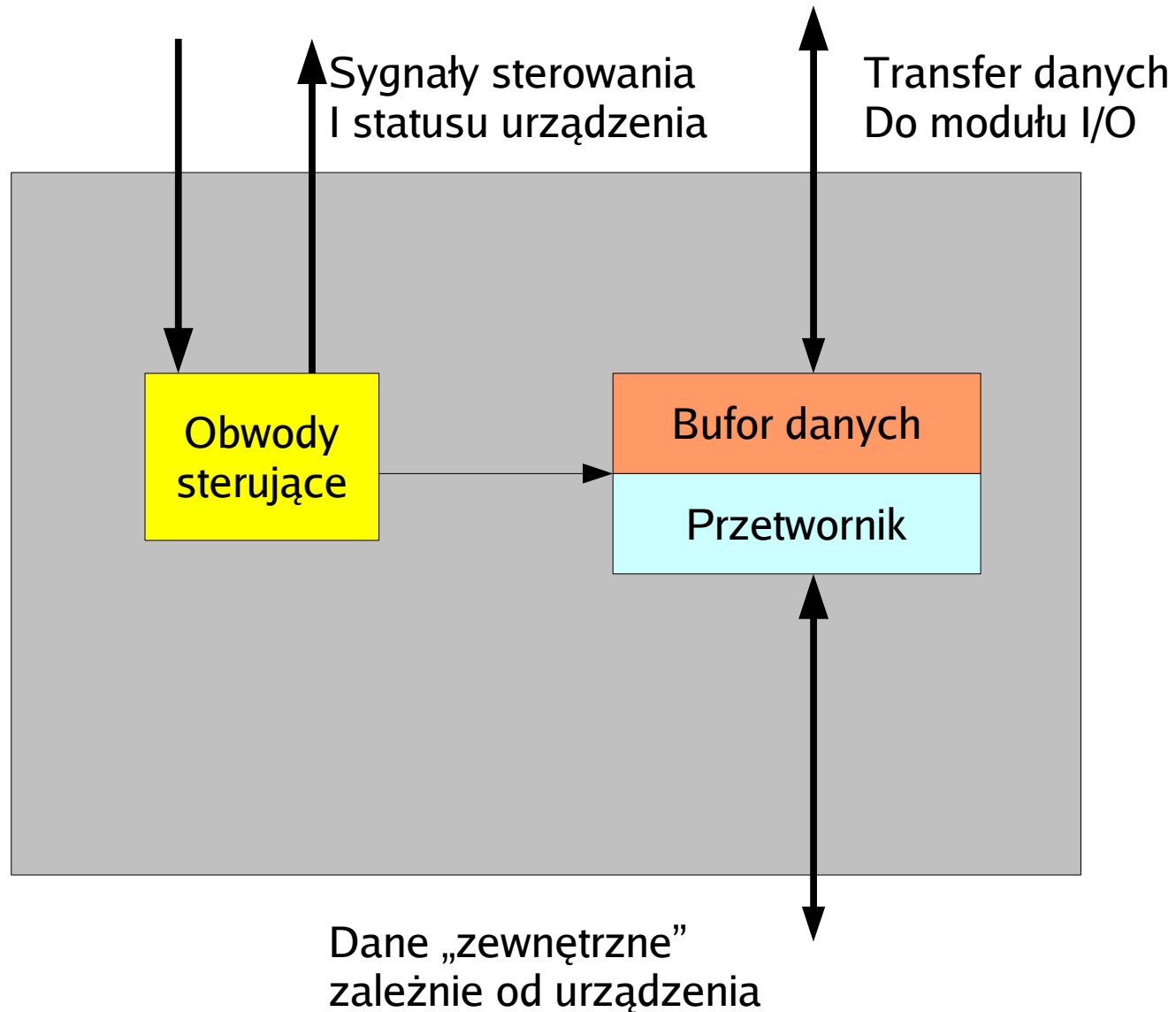




## Urządzenia zewnętrzne

- Komunikacja z człowiekiem – klawiatura, drukarka, monitor, mysz
- Komunikacja z urządzeniami – kontrolery dysków, sensory i czujniki, sterowniki urządzeń
- Wymiana danych między komputerami – modemy, karty sieciowe

# Urządzenia zewnętrzne - budowa





## Funkcje modułu I/O

- Sterowanie i synchronizacja urządzeń zewnętrznych
- Komunikacja z procesorem
- Komunikacja z urządzeniami peryferyjnymi
- Buforowanie danych
- Detekcja i korekcja błędów



## Sterowanie i synchronizacja

- Przykład działania modułu I/O
  - CPU zgłasza pytanie o stan urządzenia peryferyjnego
  - Moduł określa status urządzenia i generuje odpowiedź
  - CPU zgłasza żądanie transferu danych
  - Moduł pobiera porcję danych z urządzenia
  - Porcja danych przesyłana jest do CPU





## Komunikacja z procesorem

- Dekodowanie rozkazów CPU wysyłanych do modułu I/O
- Pobieranie i dekodowanie adresu urządzenia
- Przekazywanie informacji o statusie urządzeń peryferyjnych
- Pobieranie/wysyłanie danych do CPU



# Komunikacja z urządzeniem

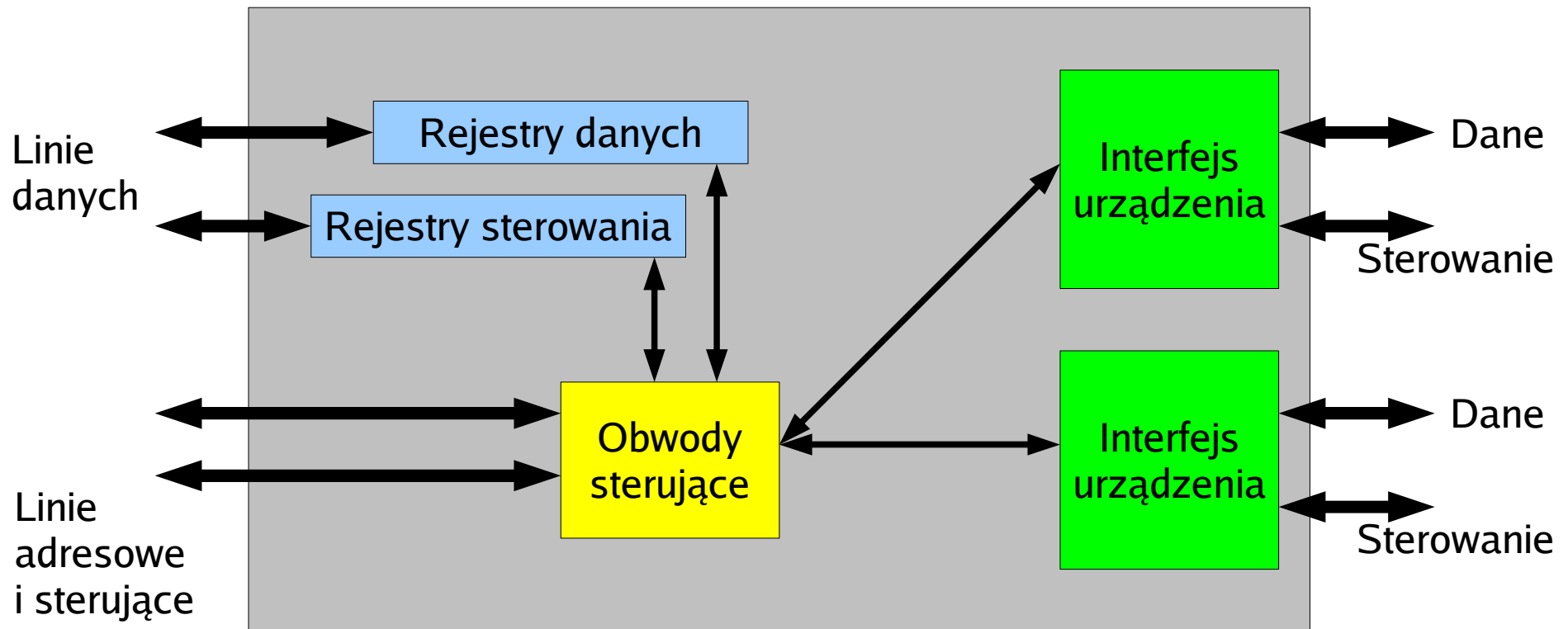
- Przekazywanie sygnałów sterujących do urządzenia
- Pobieranie informacji o stanie urządzenia, błędach itp.
- Wysyłanie i odbieranie porcji danych z urządzenia



## Buforowanie i korekcja błędów

- Buforowanie
  - „Dopasowywanie” szybkości transferów urządzeń i CPU
  - Wykorzystywanie transferów blokowych
- Detekcja i korekcja błędów
  - Wykrywanie błędów urządzenia (np. awarie sprzętowe)
  - Detekcja błędów i przekłamań danych (bity parzystości, sumy kontrolne)
  - Korekcja błędów

# Moduł I/O - budowa





## Moduły I/O – decyzje projektowe

- W jakim stopniu własności urządzeń będą dostępne dla CPU
- Ile urządzeń może współpracować z jednym modułem I/O
- Jakie funkcje obsługi urządzenia przejmuje moduł I/O, a jakie pozostają pod kontrolą CPU
- Decyzje mające związek z SO.



# Realizacja obsługi urządzeń

- Realizacja programowa
- Obsługa urządzeń sterowana przerwaniem
- Bezpośredni dostęp do pamięci



## Realizacja programowa

- CPU zgłasza żądanie operacji I/O
- Moduł I/O wykonuje operację
- Moduł I/O ustawia status wykonania operacji
- CPU okresowo sprawdza status operacji I/O
- Po stwierdzeniu zakończenia operacji CPU pobiera dane z bufora modułu I/O



## Realizacja programowa c.d.

- CPU ma pełną kontrolę nad urządzeniem
- CPU oczekuje na realizację polecenia (aktywne oczekiwanie)
- CPU jest zaangażowany w transfer danych między urządzeniem i pamięcią





## Rozkazy sterujące modułem I/O

- Selekcja urządzenia (jeżeli moduł obsługuje kilka urządzeń).
- Sterowanie (np. aktywacja urządzenia).
- Status (czy urządzenie gotowe, ew. błędy).
- Odczyt danych z urządzenia do bufora modułu I/O.
- Zapis danych z bufora modułu I/O do urządzenia.



## Adresowanie urządzeń I/O

- Mapowane na pamięć
  - wykorzystuje część przestrzeni adresowej pamięci
  - rozkazy transferów z/do pamięci dotyczą też I/O
- Izolowana przestrzeń I/O
  - osobna przestrzeń adresowa I/O
  - specjalny sygnał na szynie sterującej określający operację I/O
  - wymagane specjalne rozkazy CPU do obsługi I/O



# Obsługa urządzeń sterowana przerwaniem

- CPU zgłasza żądanie operacji
- Moduł I/O realizuje operacje i zgłasza przerwanie
- CPU odbiera informację o zgłoszeniu przerwania i wywołuje odpowiednią procedurę obsługi przerwania
- Nie występuje aktywne oczekiwanie. Transfer danych realizowany jest poprzez CPU



## Obsługa przerwania (CPU)

- Pod koniec każdego cyklu rozkazowego CPU sprawdza czy nie wystąpiło przerwanie.
- Jeżeli przerwanie wystąpiło to jest obsługiwane:
  - zachowywany jest kontekst aktualnie wykonywanego programu.
  - sterowanie przekazywane jest do procedury obsługi przerwania.
  - po zakończeniu procedury obsługi przerwania przywracany jest kontekst poprzedniego programu.



## Przerwania - problemy

- Jak rozpoznać urządzenie, które zgłosiło przerwanie?
- Jak obsługiwać wielokrotne lub jednoczesne przerwania?
- Jak pogodzić przerwania z problemami synchronizacji i współbieżności?



# Przerwania – identyfikacja urządzeń

- Wiele linii przerwań
  - Wymaga wielu doprowadzeń linii przerwań do CPU
  - Niepraktyczne (mała liczba urządzeń)
- Odpytywanie urządzeń
  - Po wystąpieniu przerwania CPU „odpytuje” wszystkie urządzenia
  - Wolne!



# Przerwania – identyfikacja urządzenia c.d.

- Metoda łańcuchowa (daisy chain)
  - Urządzenia połączone są w łańcuch
  - CPU wysyła potwierdzenie odbioru przerwania do pierwszego urządzenia
  - Jeżeli urządzenie wygenerowało przerwanie to wystawia informację na szynę danych, jeżeli nie
    - to przesyła potwierdzenie dalej.



# Przerwania – identyfikacja urządzenia c.d.

- Zastosowanie sterownika przerwań
  - Obsługą przerwań zajmują się dedykowany układ
  - Sterownik przerwań zawiera wiele linii przerwań do urządzeń
  - Po zgłoszeniu przerwania przez urządzenie, sterownik informuje CPU poprzez sygnał INTR i wystawia na szynę danych informacje o przerwaniu.





## Przerwania - priorytety

- Priorytety przerwań pozwalają na:
  - określenie kolejności obsługi kilku jednocześnie zgłoszonych przerwań
  - wyeliminowanie ryzyka wielokrotnego zagnieżdżenia procedur obsługi przerwania
- Sterownik przerwań pozwala na zdefiniowanie priorytetów poszczególnych przerwań.



## Przerwania i współbieżność

- Specjalne rozkazy procesora umożliwiają zablokowanie obsługi przerw na czas pobytu w sekcji krytycznej
- Specjalne rozkazy atomowe CPU gwarantują, że w trakcie ich wykonywania nie nastąpi przerwanie.



## Kontroler przerwań 8259A

- Układy Intel 80x86
  - Jedna linia zgłoszenia przerwania (INTR)
  - Jedna linia potwierdzenia przerwania (INTA)
- Współpracujący z układem Intel 8086 sterownik przerwań 8259A
  - 8 linii przerwań
  - możliwość łączenia w kaskady (kilka elementów)
  - możliwości programowania trybu pracy



# Bezpośredni dostęp do pamięci (DMA)

- CPU zgłasza żądanie operacji
- Moduł I/O realizuje transfer pomiędzy pamięcią i urządzeniem
- Moduł I/O zgłasza przerwanie informując CPU o zakończeniu operacji
- Brak aktywnego oczekiwania. Bezpośredni transfer danych z/do pamięci – wydajne przestania blokowe



## DMA c.d.

- Przesłania DMA wykonywane są w cyklach „kradzionych” CPU (jest wtedy odłączony od magistrali)
- CPU nie obsługuje transferu, nie musi przełączać kontekstu
- DMA w niewielkim stopniu spowalnia CPU



## Architektura podsystemu we/wy

- Wspólna magistrala
  - Wszystkie urządzenia na jednej magistrali. Mało wydajne – potrzebne 2 transfery (I/O-DMA, DMA-RAM)
- Wspólna magistrala – zintegrowany moduł I/O
  - Kontroler DMA zintegrowany z I/O. Pojedynczy transfer (DMA-RAM)
- Osobna magistrala I/O
  - Osobna magistrala dla urządzeń dołączona do kontrolera DMA



## Interfejs USB

- Sprzętowy port komunikacyjny komputera
- Zastąpił wcześniejsze porty szeregowo i równoległe
- Możliwość podłączenia wielu różnych urządzeń
- Automatyczne wykrywanie i konfigurowanie urządzeń
- Możliwość podłączania i odłączania urządzeń w czasie pracy komputera



## Interfejs USB c.d.

- Standardy:
  - USB 1.0/1.1      12 Mbit/s
  - USB 2.0            480 Mbit/s
  - USB 3.0            3.2 Gbit/s





## Podsumowanie

- Charakterystyka urządzeń zewnętrznych
- Moduły I/O
- Obsługa operacji I/O
  - Programowa
  - Sterowana przerwaniem
  - DMA
- Interfejs USB