Architektura komputerów

sem. zimowy 2024/2025

cz. 4

Tomasz Dziubich

POLITECHNIKA GDAŃSKA

Zasady komunikacji z urządzeniami zewnętrznymi (1)

- Rozmaite rodzaje urządzeń zewnętrznych komputera wymagają doprowadzenia określonych sygnałów, specyficznych dla danego urządzenia, np. monitor ekranowy wymaga przekazywania, obok informacji o treści wyświetlanego obrazu, także impulsów synchronizujących.
- Niezbędne jest zainstalowanie układów pośredniczących, które dopasowują standardy sygnałowe procesora i płyty głównej do specyficznych wymagań poszczególnych urządzeń.

POLITECHNIKA GDAŃSKA

Zasady komunikacji z urządzeniami zewnętrznymi (2)

- Układy pośredniczące nazywane są często *układami wejścia/wyjścia*.
- Układy wejścia/wyjścia umieszczane są na kartach rozszerzeniowych lub na płycie głównej komputera.
- W takim ujęciu procesor nie steruje urządzeniami bezpośrednio, ale wykonuje to za pośrednictwem układów wejścia/wyjścia, które z jednej strony dostosowane są do standardów procesora i płyty głównej, a z drugiej strony są dopasowane do danego urządzenia.

POLITECHNIKA GDAŃSKA

Zasady komunikacji z urządzeniami zewnętrznymi (3)

- Układy wejścia/wyjścia umożliwiają testowanie stanu (gotowości) urządzenia, wysyłanie poleceń do urządzenia oraz wysyłanie i przyjmowanie danych.
- Od strony procesora ww. komunikacja odbywa się zazwyczaj poprzez zapis i odczyt rejestrów zainstalowanych w układach wejścia/wyjścia.
- Istnieje też przesyłania danych z pamięci operacyjnej (głównej) do urządzenia (lub odwrotnie) z pominięciem procesora — technika ta oznaczana jest skrótem DMA (ang. Direct Memory Access).

POLITECHNIKA GDAŃSKA

Zasady komunikacji z urządzeniami zewnętrznymi (4)

- W układach wejścia/wyjścia istotną rolę odgrywają zazwyczaj cztery rejestry:
  - **rejestr stanu** zawiera informacje o stanie urządzenia, w szczególności pozwala stwierdzić czy urządzenie jest zajęte, czy dane są gotowe do odczytania lub czy wystąpił błąd,
  - **rejestr sterujący** przyjmuje polecenia, które urządzenie ma wykonać,
  - **rejestr danych wysyłanych** do urządzenia,
  - **rejestr danych odebranych** z urządzenia.

POLITECHNIKA GDAŃSKA

Odczytywanie i zapisywanie rejestrów układów wejścia/wyjścia (1)

- Stosowane są dwie metody dostępu do zawartości rejestrów układów wejścia/wyjścia:
  - rejestry udostępniane są jako zwykłe komórki pamięci w przestrzeni adresowej pamięci — mówimy wówczas o *współadresowalnych układach wejścia/wyjścia*;
  - rejestry urządzenia dostępne są w odrębnej przestrzeni adresowej zwanej *przestrzenią adresową wejścia-wyjścia* lub *przestrzenią adresową portów*; takie rozwiązanie określane jest czasami jako *izolowane wejście-wyjście.*

POLITECHNIKA GDAŃSKA

Odczytywanie i zapisywanie rejestrów układów wejścia/wyjścia (2)

- Sterowanie pracą urządzeń wymaga dokładnej znajomości zasad ich funkcjonowania, a niewłaściwe sterowanie może doprowadzić do przedwczesnego zużycia lub zniszczenia urządzenia.
- Stosowana powszechnie wielozadaniowość w systemach komputerowych (możliwość jednoczesnego wykonywania kilku programów) stwarza możliwość wystąpienia kolizji w zakresie dostępu do urządzenia (np. gdy dwa programy kierują wyniki do tej samej drukarki).

POLITECHNIKA GDAŃSKA

Odczytywanie i zapisywanie rejestrów układów wejścia/wyjścia (3)

- Z podanych powodów, w komputerach powszechnego użytku sterowanie pracą urządzeń zewnętrznych komputera może być wykonywanie jedynie przez system operacyjny — wszelkie odwołania do rejestrów urządzeń wejścia/wyjścia w zwykłym programie traktowane są jako operacje nielegalne i powodują przerwanie wykonywania programu.
- Zwykły program może wydawać zlecenia dla urządzeń jedynie za pośrednictwem systemu operacyjnego — w tym celu program może wywołać odpowiednią funkcję z interfejsu API.

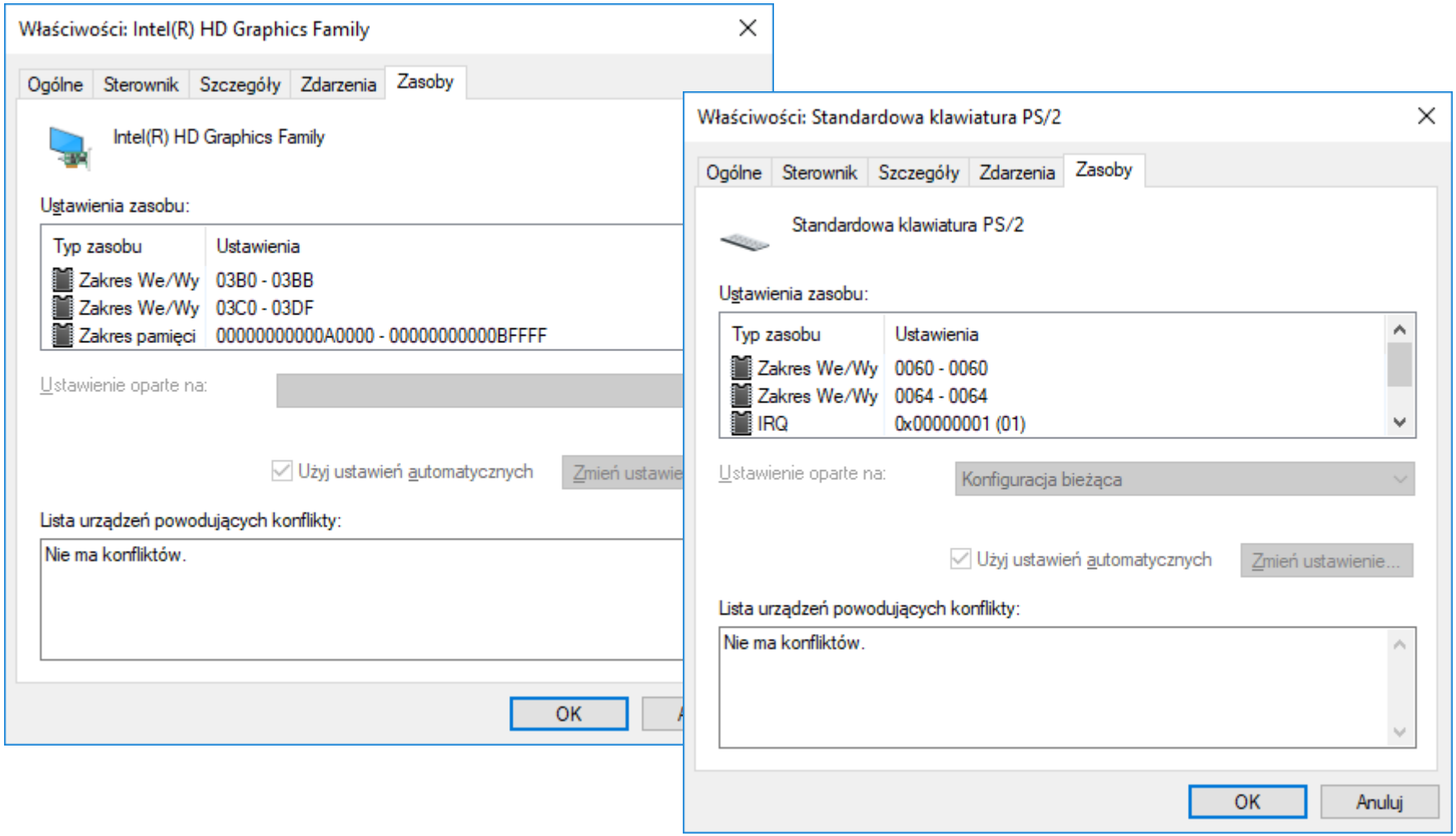
POLITECHNIKA GDAŃSKA

Odczytywanie i zapisywanie rejestrów układów wejścia/wyjścia (4)

- W początkowym okresie rozwoju komputerów PC podane ograniczenia nie były stosowane — nierzadko zwykłe programy bezpośrednio sterowały urządzeniami, co pozwalało na uzyskanie większej wydajności.
- Możliwe było także przejmowanie niektórych funkcji systemu operacyjnego przez zwykłe programy, co pozwalało także na wykonywanie ciekawych eksperymentów (przy stosowanym obecnie sprzęcie i oprogramowaniu nie mogą być wykonywane).

POLITECHNIKA GDAŃSKA

Odczytywanie i zapisywanie rejestrów układów wejścia/wyjścia (5)



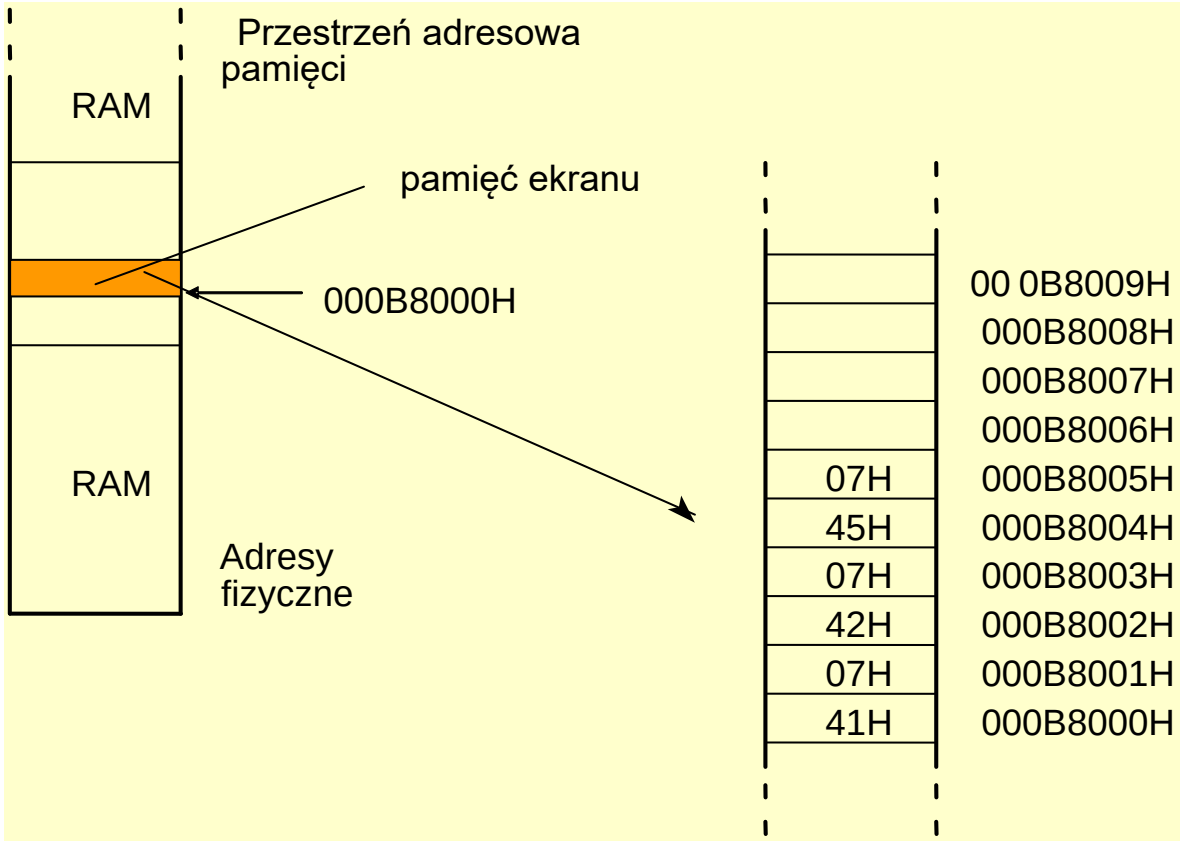
POLITECHNIKA GDAŃSKA

Pamięć ekranu w trybie tekstowym (1)

- Typowym przykładem wykorzystania techniki układów *współadresowalnych* jest pamięć ekranu w komputerach PC.
- W trybie tekstowym sterownika graficznego (karty graficznej) znaki wyświetlane na ekranie stanowią odwzorowanie zawartości obszaru pamięci od adresu fizycznego B8000H — pamięć ta należy do przestrzeni adresowej procesora, ale zainstalowana jest na karcie sterownika.
- W praktyce, tryb tekstowy używany jest przez system BIOS bezpośrednio po włączeniu (lub zresetowaniu) komputera, przed załadowaniem głównego systemu operacyjnego.

POLITECHNIKA GDAŃSKA

Pamięć ekranu w trybie tekstowym (2)



Bajt atrybutu

M	R	G	B	I	R	G	B

kolor tła kolor znaku

POLITECHNIKA GDAŃSKA

Pamięć ekranu w trybie tekstowym (3)

- Każdy znak wyświetlany na ekranie jest opisywany przez dwa bajty w pamięci ekranu: bajt o adresie parzystym zawiera kod ASCII znaku, natomiast następny bajt zawiera opis sposobu wyświetlania, nazywany *atrybutem znaku*.
- Kolejne bajty omawianego obszaru odwzorowywane są w znaki na ekranie począwszy od pierwszego wiersza od lewej do prawej, potem drugiego wiersza, itd. tak jak przy czytaniu zwykłego tekstu.

POLITECHNIKA GDAŃSKA

Przykład: wyświetlanie tekstu na ekranie (1)

- W podanym przykładzie zapis do pamięci ekranu wykonywany jest za pomocą rozkazów MOV postaci:

```
mov  byte PTR es:[4], 'E'
```

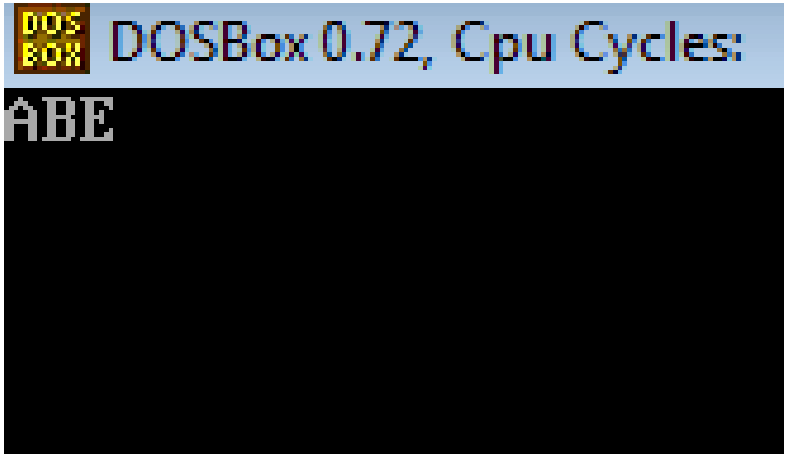
- Wyrażenia adresowe w tych rozkazach zawierają nazwę rejestru segmentowego (tu: **es**) — adres fizyczny komórki pamięci, do której zostanie zapisana podana wartość jest obliczany wg formuły

```
es * 16 + zaw.pola adresowego (tu: 4)
```

- Wyrażenia adresowe tego typu stosowane były przed wprowadzeniem pamięci wirtualnej.

POLITECHNIKA GDAŃSKA

Przykład: wyświetlanie tekstu na ekranie (2)




```
mov  ax, 0B800H
mov  es, ax
mov  byte PTR es:[0], 'A'
mov  byte PTR es:[1], 00000111B
mov  byte PTR es:[2], 'B'
mov  byte PTR es:[3], 00000111B
mov  byte PTR es:[4], 'E'
mov  byte PTR es:[5], 00000111B
```

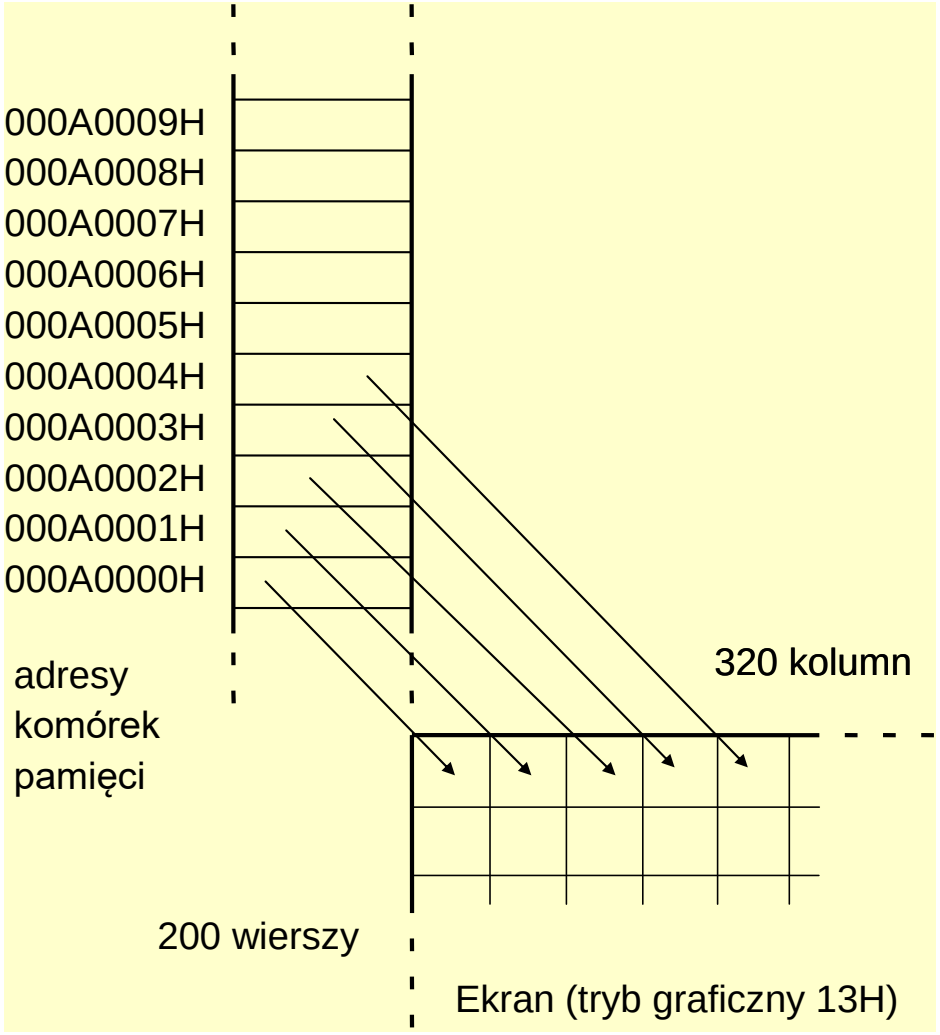
POLITECHNIKA GDAŃSKA

Pamięć ekranu w trybie graficznym (1)

- Współczesne sterowniki (karty) graficzne oferują zazwyczaj wiele trybów wyświetlania, różniących się rozdzielczością, liczbą kolorów i innymi parametrami — m.in. dostępny dość prosty tryb graficzny oznaczony numerem 13H.
- W trybie 13H obraz ma wymiary 320 * 200 punktów (pikseli), przy czym każdy piksel może być wyświetlany w jednym z 256 kolorów.
- Kolor piksela określa liczba zawarta w jednym bajcie, np. liczba 10 oznacza kolor jasnozielony.

POLITECHNIKA GDAŃSKA

Pamięć ekranu w trybie graficznym (2)



POLITECHNIKA GDAŃSKA

Pamięć ekranu w trybie graficznym (3)

- W takim ujęciu cały wyświetlany obraz stanowi odwzorowanie zawartości pewnego obszaru pamięci.
- Ponieważ obraz ma 200 linii, a w każdej linii jest 320 pikseli, więc do przechowania tego obrazu w pamięci potrzeba 320 * 200 = 64000 bajtów.
- W trybie 13H pamięć ekranu umieszczona jest od adresu fizycznego A0000H.

POLITECHNIKA GDAŃSKA

Przykład: wyświetlanie linii pionowej na ekranie (1)

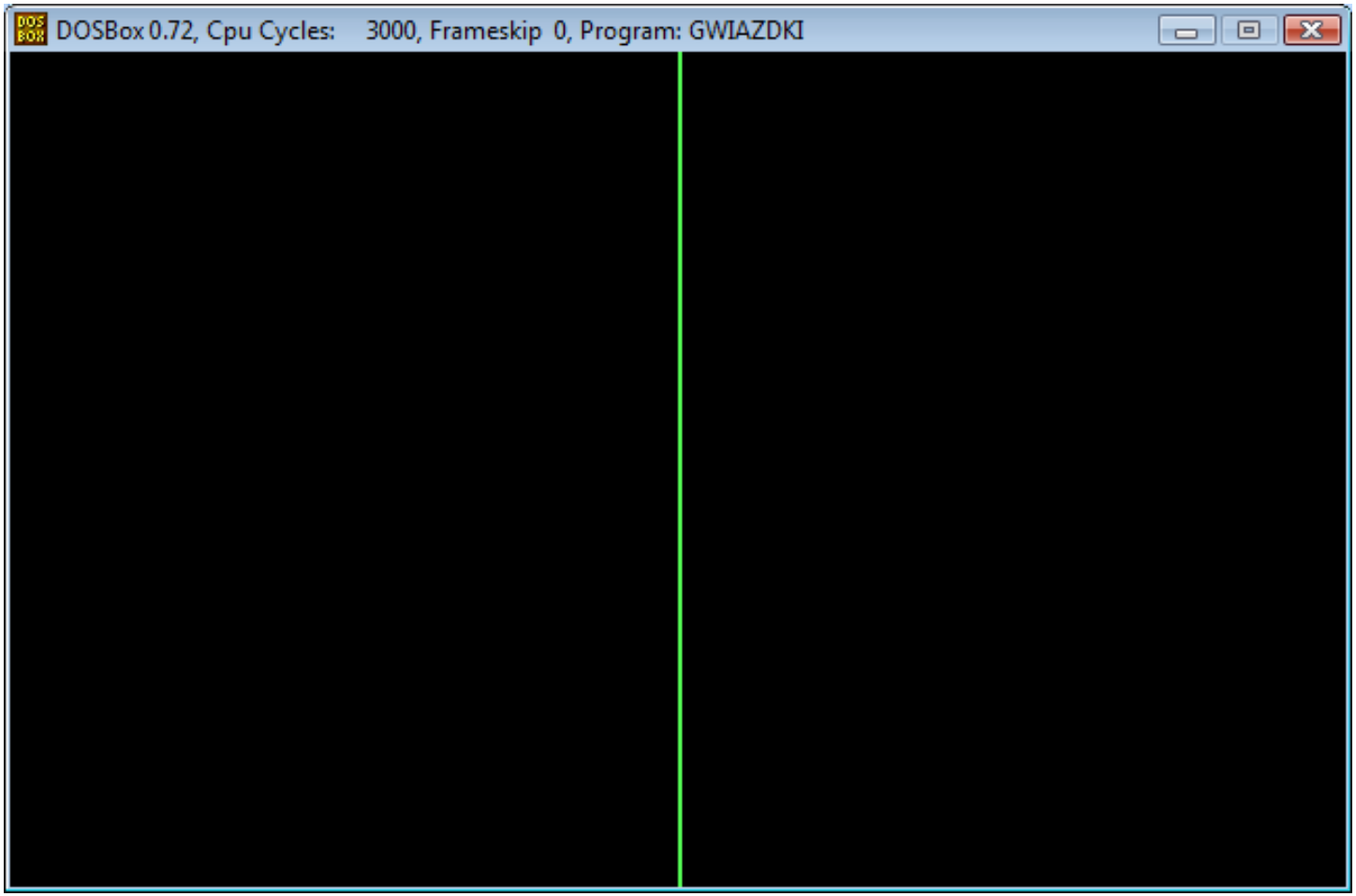
Podany dalej fragment programu powoduje wyświetlenie jasnozielonej linii pionowej w środku ekranu

```
mov     ah, 0     ; funkcja nr 0 ustawia tryb
                  ; sterownika
mov     al, 13H   ; nr trybu
int     10H       ; wywołanie funkcji systemu BIOS
```

(ciąg dalszy kodu za widokiem ekranu)

POLITECHNIKA GDAŃSKA

Przykład: wyświetlanie linii pionowej na ekranie (2)



POLITECHNIKA GDAŃSKA

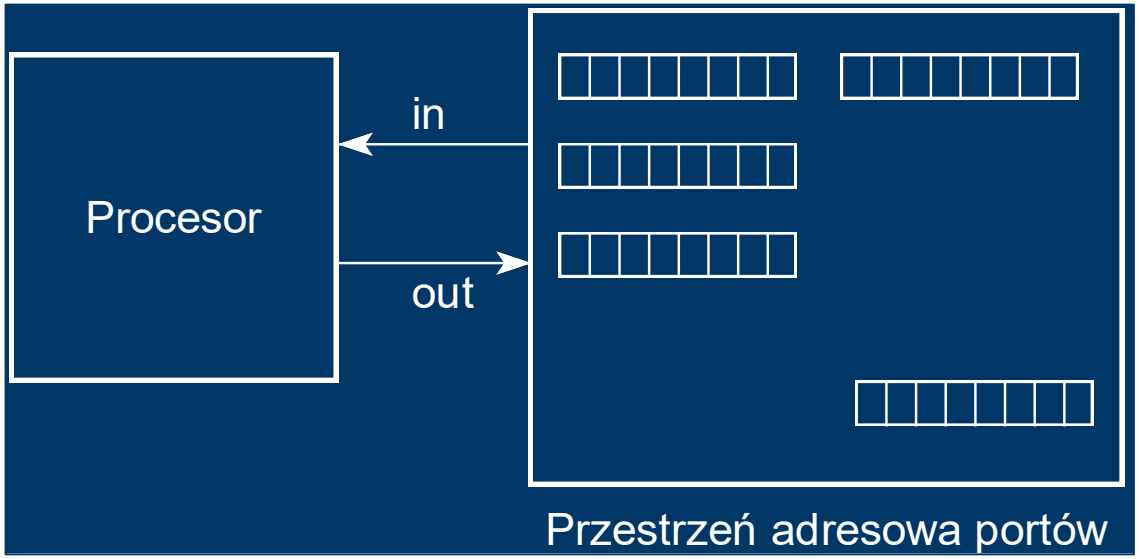
Przykład: wyświetlanie linii pionowej na ekranie (3)

```
  mov     ax, 0A000H  ; adres pamięci ekranu
  mov     es, ax
  mov     cx, 200 ; liczba linii na ekranie
  mov     bx, 160 ; adres początkowy
ptl_lin:
  mov     byte PTR es:[bx], 10 ; kolor
                               ; jasnozielony
  add     bx, 320
  loop    ptl_lin
```

Za pomocą wywołania INT 10H / AH=0 / AL=3 można powrócić do trybu tekstowego.

Przestrzeń adresowa portów (1)

- Rejestry układów wejścia/wyjścia mogą być też udostępniane w odrębnej przestrzeni adresowej, specjalnie zaprojektowanej do komunikacji z urządzeniami.
- W architekturze x86 do zapisu i odczytu danych w przestrzeni adresowej portów stosuje się rozkazy IN i OUT oraz ich rozszerzenia.



Przestrzeń adresowa portów (2)

- Przykłady:

in al, 60H — przesłanie zawartości portu o numerze 60H do rejestru AL (w typowych komputerach: odczyt numeru naciśniętego klawisza)

out 64H, al — przesłanie zawartości rejestru AL do portu 64H

Przestrzeń adresowa portów (3)

- Poniższa tablica zawiera wybrane numery portów układów płyty głównej i układów wejścia/wyjścia

Adres	Nazwa układu
000H - 01FH	Sterownik DMA nr 1
020H - 03FH	Sterownik przerwań 8259A (master)
040H - 05FH	Generatory programowalne
060H - 06FH	Sterownik klawiatury
070H - 07FH	Zegar czasu rzeczywistego

POLITECHNIKA GDAŃSKA

Przykład zmiany palety w trybie graficznym (1)

- Poniższy przykład ilustruje sposób wykorzystania rozkazów **OUT**
- W omawianym wcześniej trybie graficznym 13H używana jest standardowa paleta, w której kod 10 oznacza kolor jasnozielony — podany niżej fragment programu dokonuje zmiany palety, w taki sposób, że kod 10 oznaczać będzie kolor żółty.

POLITECHNIKA GDAŃSKA

Przykład zmiany palety w trybie graficznym (2)

- Zmiana palety dokonywana jest poprzez wpisanie kodu koloru do portu 3C8H, a następnie przesłanie składowych: R (czerwony), G (zielony), B (niebieski) do portu 3C9H.
- Poszczególne składowe mogą przyjmować wartości z przedziału <0, 63>.

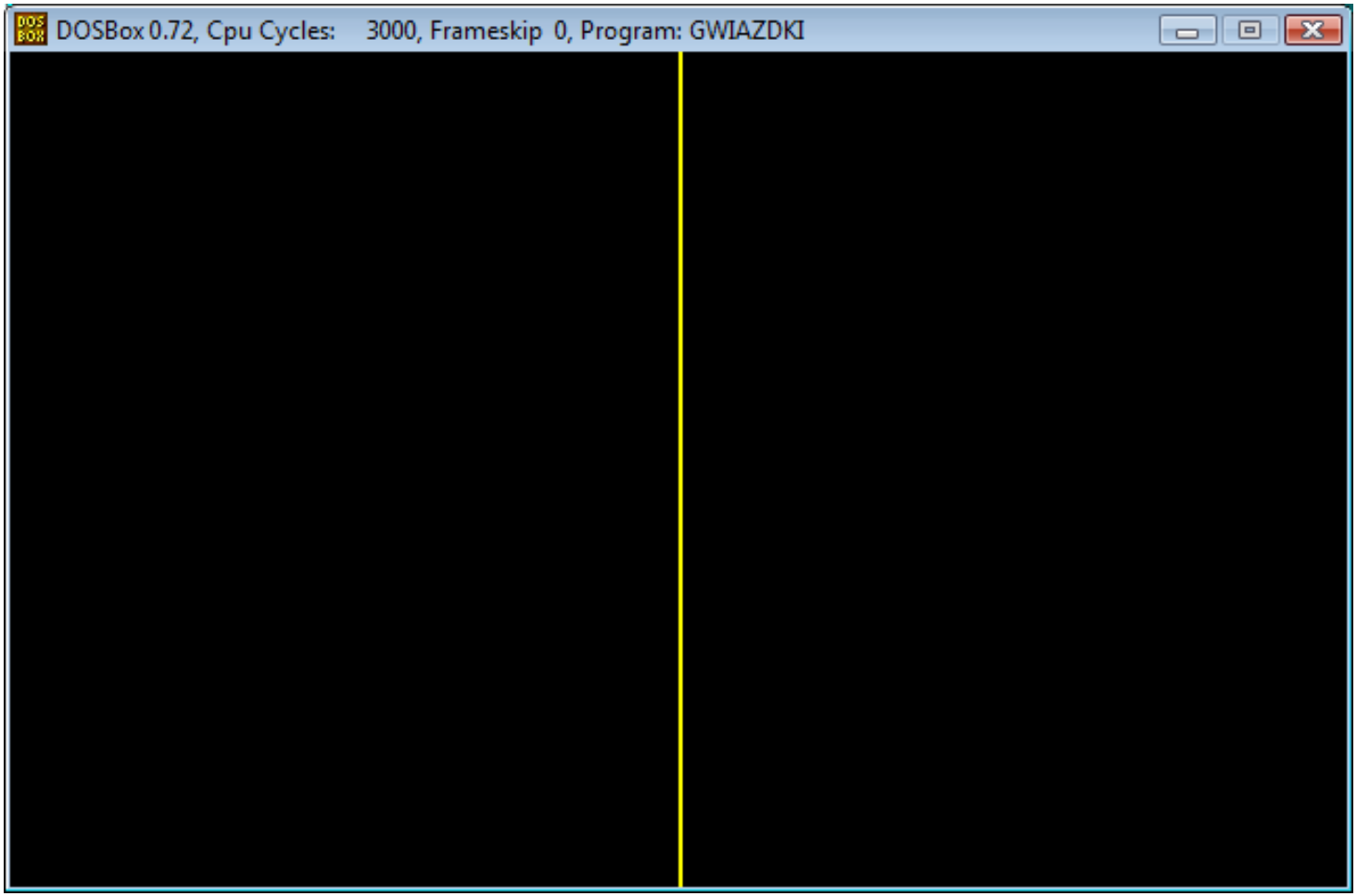
POLITECHNIKA GDAŃSKA

Przykład zmiany palety w trybie graficznym (3)

```
mov     dx, 3C8H
mov     al, 10          ; kod koloru
out     dx, al
mov     dx, 3C9H
mov     al, 63    ; składowa czerwona (R)
out     dx, al
mov     al, 63   ; składowa zielona (G)
out     dx, al
mov     al, 0    ; składowa niebieska (B)
out     dx, al
```

POLITECHNIKA GDAŃSKA

Przykład zmiany palety w trybie graficznym (4)



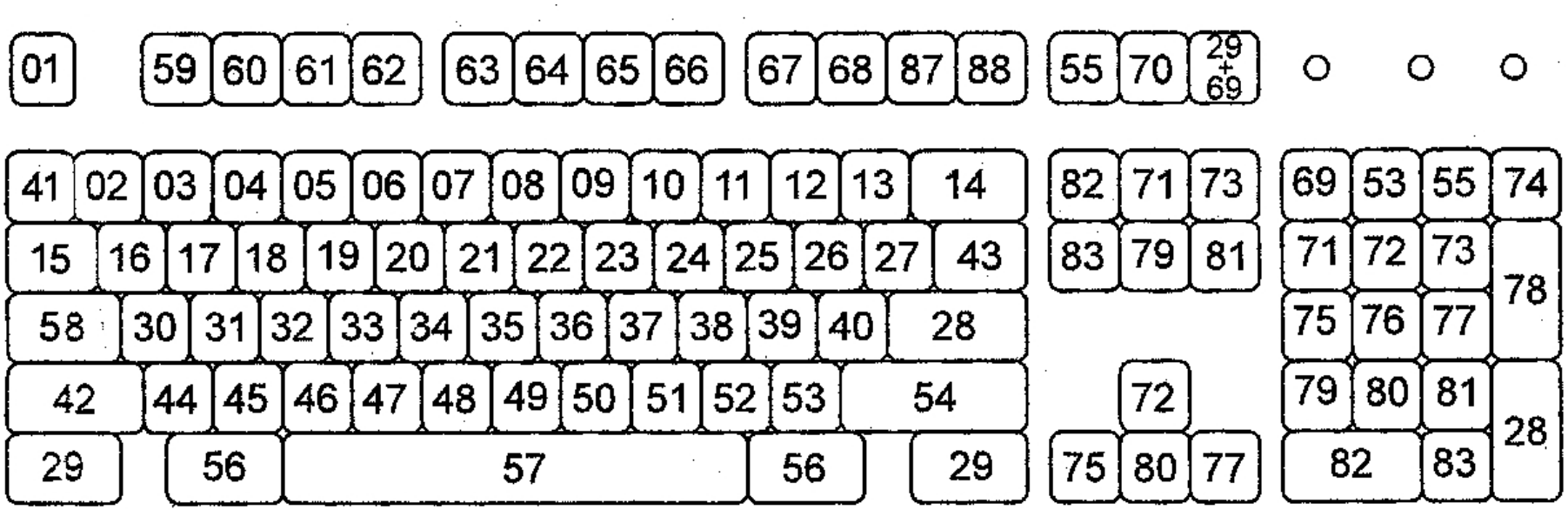
POLITECHNIKA GDAŃSKA

Wprowadzanie danych z klawiatury (1)

- Sterownik klawiatury stosowany w komputerach PC stanowi przykład układu wejścia-wyjścia sterowanego przez specjalizowany procesor.
- Przyjęte tu rozwiązania, zarówno od strony sprzętowej jak i programowej, można uważać za dość typowe, a zarazem nieskomplikowane.
- Każdemu przyciskowi klawiatury przyporządkowano ustalony kod 8-bitowy, nazywany *kodem pozycji* albo numerem klawisza (ang. scan code).

POLITECHNIKA GDAŃSKA

Wprowadzanie danych z klawiatury (2)



POLITECHNIKA GDAŃSKA

Wprowadzanie danych z klawiatury (3)

- Jednolity system numeracji obejmuje zarówno zwykłe klawisze znakowe jak też wszystkie klawisze sterujące i funkcyjne (np. Shift, Ctrl, ...); wyjątki od tej zasady omawiane są dalej.
- Po naciśnięciu lub zwolnieniu dowolnego klawisza mikrokontroler (mikroprocesor) klawiatury formuje *kod naciśnięcia* (ang. make code) lub kod zwolnienia (ang. break code) klawisza — kod ten zostaje przesłany szeregowo do układów płyty głównej komputera.
- Kod zwolnienia klawisza zawiera kod naciśnięcia poprzedzony bajtem F0H.

POLITECHNIKA GDAŃSKA

Wprowadzanie danych z klawiatury (4)

- Kody naciśnięcia i zwolnienia jednoznacznie określają klawisze, ale nie są identyczne z kodami pozycji.
- Kody naciśnięcia i zwolnienia nie występują na poziomie programowania — przykładowe kody dla kilku klawiszy podaje poniższa tabelka.

Klawisz	kod pozycji	kod naciśn.	kod zwoln.
A	30	1CH	F0H 1CH
B	48	32H	F0H 32H
C	46	21H	F0H 21H
Q	17	15H	F0H 15H

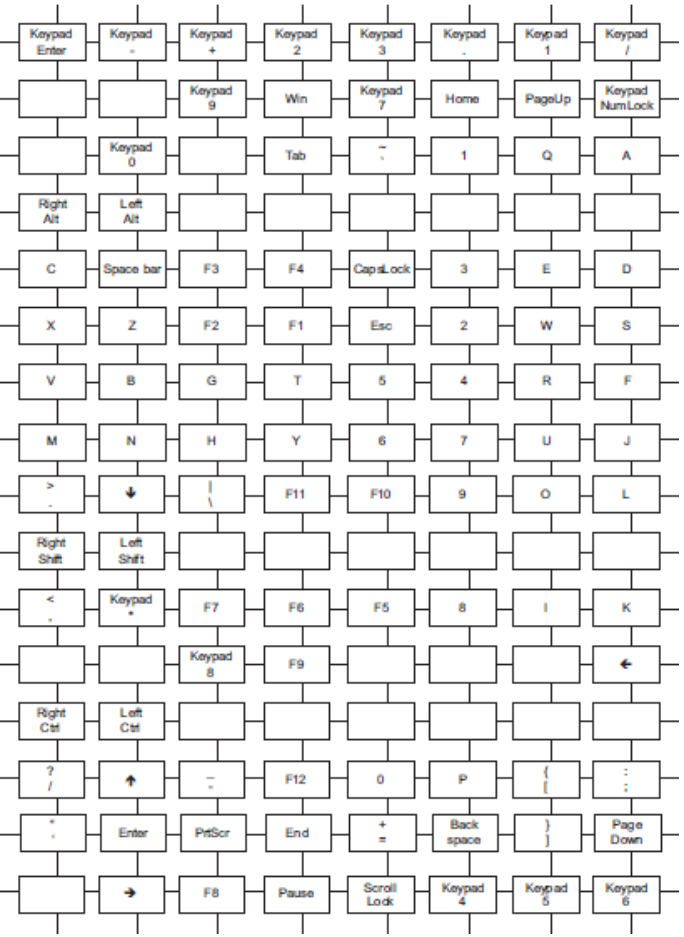
POLITECHNIKA GDAŃSKA

Identyfikacja naciśniętego klawisza (1)

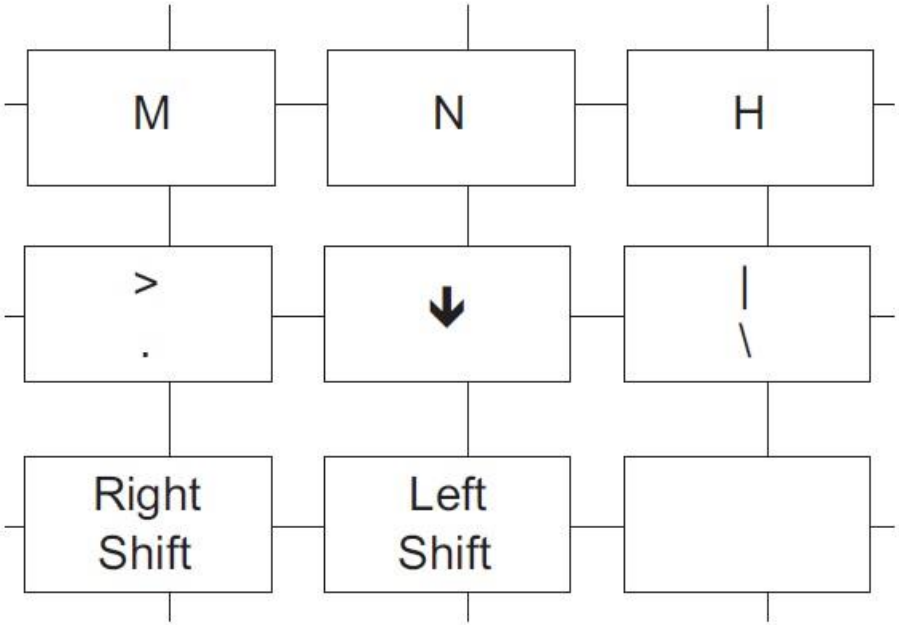
- Klawiatury buduje się zazwyczaj z klawiszy zwiernych połączonych w matryce.
- Identyfikacja naciśniętego klawisza wymaga przeszukania klawiatury poprzez podawanie zera na kolejne wiersze matrycy i badanie za każdym razem czy pojawiło się zero na którejś z kolumn.
- Pojawienie się zera na *i*-tej kolumnie po podaniu zera na *j*-ty wiersz oznacza, że został naciśnięty klawisz leżący na skrzyżowaniu *i*-tej kolumny *j*-tego wiersza.
- Na rysunku pokazano usytuowanie klawiszy w matrycy złożonej z 16 wierszy i 8 kolumn.

POLITECHNIKA GDAŃSKA

Identyfikacja naciśniętego klawisza (2)



Fragment w powiększeniu



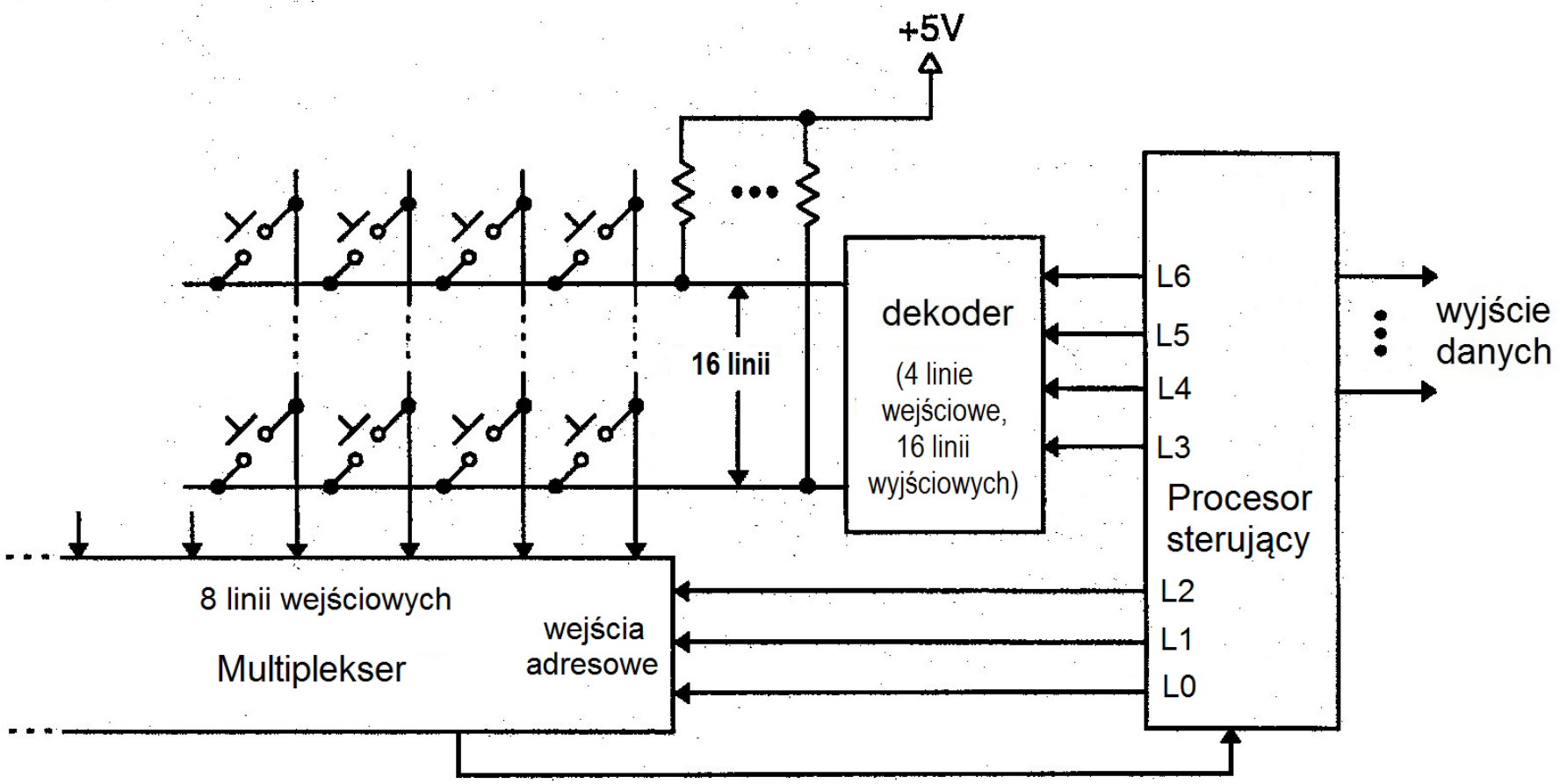
POLITECHNIKA GDAŃSKA

Identyfikacja naciśniętego klawisza (3)

- Realizację techniczną dekodowania naciśniętych i (zwalnianych) klawiszy wyjaśnia rysunek na następnym slajdzie.
- Do wejść dekodera doprowadzono 4 linie z procesora sterującego klawiatury. W zależności od wartości sygnałów podawanych na wejścia dekodera, na jednej, spośród 16 linii wyjściowych, pojawia się stan zero – wszystkie pozostałe wyjścia dekodera są ustawiane w stan wysoki (linie te możemy traktować jako wiersze matrycy klawiatury).

POLITECHNIKA GDAŃSKA

Identyfikacja naciśniętego klawisza (4)



POLITECHNIKA GDAŃSKA

Identyfikacja naciśniętego klawisza (5)

- Naciśnięcie klawisza powoduje połączenie linii wiersza z linią kolumny (8 linii kolumn doprowadzono do multipleksera).
- W celu zidentyfikowania naciśniętego klawisza procesor sterujący generuje na swoich wyjściach liczby adresujące dekoder i multiplekser.
- Część adresu podawana na dekoder uaktywnia jedną z jego linii (wprowadzając ją w stan zero).
- Przy ustalonej wartości na dekoderze zmienia się wartość liczb podawanych na multiplekser (są to bowiem najmłodsze bity).

POLITECHNIKA GDAŃSKA

Identyfikacja naciśniętego klawisza (6)

- Jeżeli żaden z klawiszy nie jest naciśnięty, to na wyjściu multipleksera będzie występował stan wysoki.
- W przypadku wciśniętego klawisza na wyjściu multipleksera pojawi się stan zero, ale tylko dla kombinacji powodującej połączenie wyjścia multipleksera z linią, do której dołączony jest wciśnięty klawisz.
- Zero na wyjściu multipleksera sygnalizuje więc, że odnaleziono naciśnięty klawisz, a jego położenie na klawiaturze można łatwo określić na podstawie numeru wiersza i kolumny.

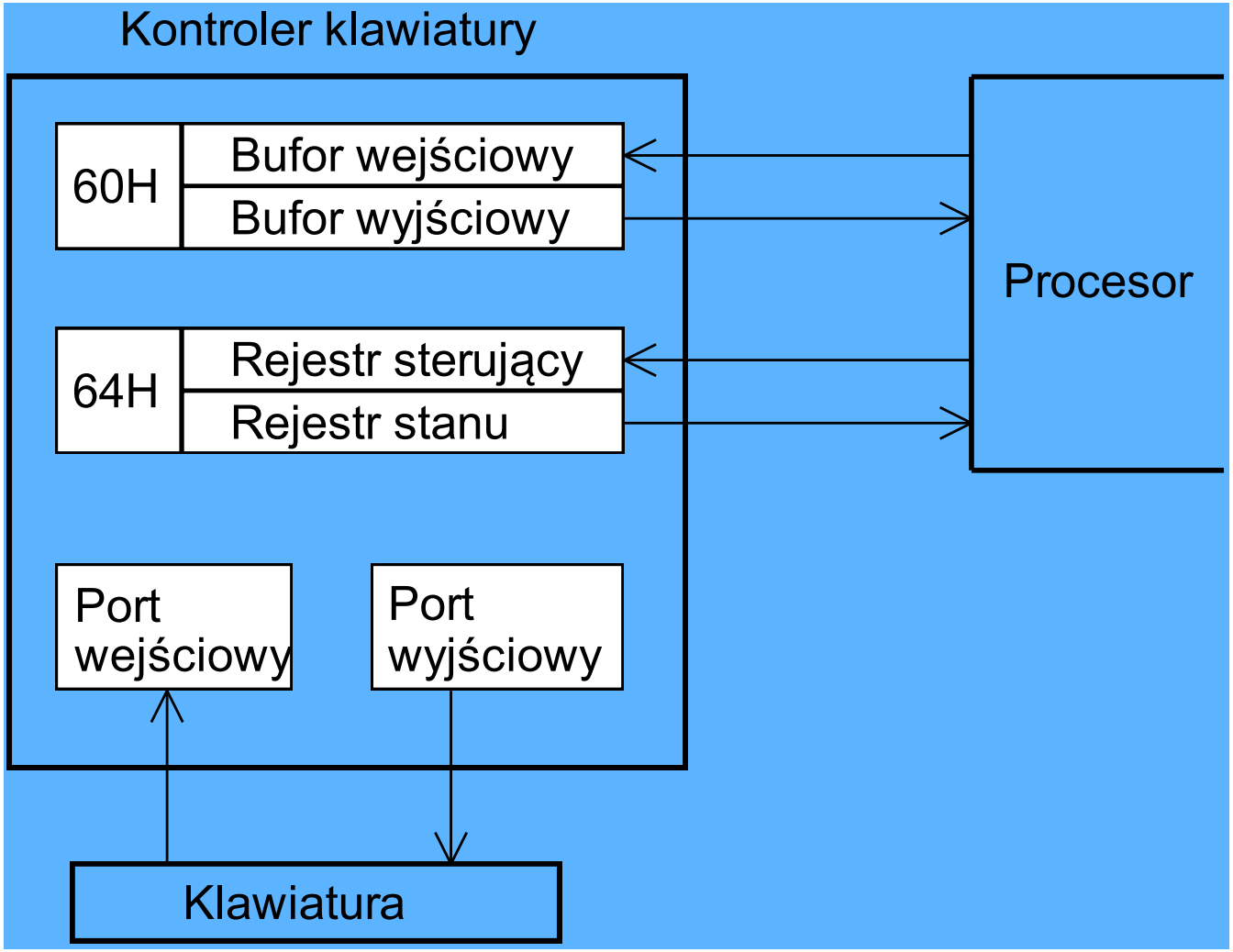
POLITECHNIKA GDAŃSKA

Obsługa klawiatury na poziomie programowania (1)

- Sygnały przesyłane z klawiatury docierają do układu na płycie głównej komputera (mikrokontroler, np. 8042 lub 8741 lub 8742), który odtwarza oryginalny kod pozycji (w przypadku zwolnienia klawisza: kod pozycji +128) — odtworzony kod pozycji zostaje udostępniony w porcie 60H.
- Zarówno mikrokontroler na płycie głównej jak też mikrokontroler klawiatury mogą być programowane za pomocą rozkazów wysyłanych przez główny procesor poprzez porty 60H i 64H.

POLITECHNIKA GDAŃSKA

Obsługa klawiatury na poziomie programowania (2)



POLITECHNIKA GDAŃSKA

Obsługa klawiatury na poziomie programowania (3)

- Istnieje kilkanaście rozkazów, które mogą być przesyłane do klawiatury (łącze między komputerem a klawiaturą umożliwia transmisję dwukierunkową) — m.in. w ten sposób można określić parametry tzw. autorepetycji, czyli samoczynnego powtarzania wysyłania kodu odpowiadającego przytrzymywanemu dłużej klawiszowi.
- Poniższy przykład przedstawia jeden ze sposobów odczytywania numeru naciśniętego klawisza (kodu pozycji) — w podanym przykładzie mechanizm przerwań (opisany dalej) nie jest używany.

POLITECHNIKA GDAŃSKA

Obsługa klawiatury na poziomie programowania (4)

```
; zablokowanie przerwań z klawiatury
  mov    al, 2
  out        21H, al
czekaj:
  in     al, 64H ; odczyt rejestru stanu
                 ; klawiat.
; sprawdzenie czy kod pozycji dostępny jest
; w buforze wyjściowym
  test   al, 1
  jz     czekaj  ; oczekiwanie w pętli
; odczytywanie kodu pozycji naciśniętego
  klawisza
  in     al, 60H
```

POLITECHNIKA GDAŃSKA

Sterowanie pracą urządzeń zewnętrznych (1)

- Zlecenie by urządzenie zewnętrzne dołączone do komputera wykonało pewną operację wymaga podjęcia następujących działań:
  - sprawdzenie stanu urządzenia;
  - wysłanie odpowiednich poleceń do urządzenia, o ile znajduje się ono w stanie gotowości;
  - przesłanie (lub odczytanie) danych;
  - oczekiwanie na zakończenie operacji.

POLITECHNIKA GDAŃSKA

Sterowanie pracą urządzeń zewnętrznych (2)

- Przesyłanie danych do omawianej wcześniej pamięci ekranu może zachodzić w dowolnym momencie, ale w przypadku wielu innych urządzeń przesyłanie danych może nastąpić dopiero wtedy, gdy:
  - urządzenie jest gotowe do przyjęcia danej,
  - lub gdy dana jest już przygotowana do udostępnienia komputerowi.

POLITECHNIKA GDAŃSKA

Metoda aktywnego oczekiwania (odpytywania) (1)

- Zatem operacja przesłania danej z urządzenia do komputera lub odwrotnie musi być poprzedzona sprawdzeniem czy dana jest dostępna, lub czy urządzenie jest w stanie przyjąć daną.
- Jeśli sprawdzenie da wynik pozytywny, to następuje przesłanie danej, w przeciwnym razie trzeba wielokrotnie powtarzać operację sprawdzenia aż do chwili, gdy dana będzie dostępna do odczytu lub gdy urządzenie będzie w stanie przyjąć daną.

POLITECHNIKA GDAŃSKA

Metoda aktywnego oczekiwania (odpytywania) (2)

- Omawiana metoda wielokrotnego sprawdzenia urządzenia nazywana jest metodą *aktywnego oczekiwania* lub *odpytywania*.
- Metoda aktywnego oczekiwania jest nieefektywna i jałowo pochłania czas pracy procesora (program spędza czas w pętli oczekiwania).
- Trzeba też brać pod uwagę możliwość, że oczekiwane zdarzenie może wystąpić po bardzo długim czasie lub w ogóle nie wystąpić.

POLITECHNIKA GDAŃSKA

Metoda aktywnego oczekiwania (odpytywania) (3)

- Jeśli nawet sprawdzenie urządzenia wykonywane jest w pewnych odstępach czasu, to:
  - występują przerwy w obsłudze urządzenia, które zakłócają płynność jego pracy — urządzenie musi czekać na obsługę, co nie zawsze jest dopuszczalne (np. nie odczytany bajt zostaje zamazany przez kolejny przyjęty);
  - z kolei zwiększenie częstotliwości sprawdzania zwiększa straty czasu procesora — zazwyczaj dobór optymalnej częstotliwości sprawdzania jest trudny.

POLITECHNIKA GDAŃSKA

Metoda aktywnego oczekiwania (odpytywania) (4)

- Metoda aktywnego oczekiwania jest odpowiednia wyłącznie dla nieskomplikowanych urządzeń, w których zainstalowany procesor jest mało obciążony (np. urządzenia domowe, sterowniki oświetlenia lub ogrzewania, itp.); metoda stosowana była także w początkowym okresie rozwoju komputerów osobistych.
- Omawiana metoda nie wymaga rozbudowywania sprzętu komputerowego — synchronizacja transmisji jest osiągana na drodze programowej, poprzez testowania stanu sygnałów gotowości i ustawianie sygnałów sterujących transmisją danych.

POLITECHNIKA GDAŃSKA

Metoda przerwaniowa (1)

- We współczesnych komputerach mechanizmy obsługi przerwań należą do podstawowych elementów funkcjonalnych, niezbędnych dla poprawnej pracy całego systemu komputerowego.
- Sygnały przerwań wysyłane są przez różne urządzenia zewnętrzne komputera (klawiatura, drukarka, mysz, dyski, itd.) do procesora — sygnalizują one wystąpienie pewnych zdarzeń (np. naciśnięcie przycisku myszki), które wymagają podjęcia niezwłocznej obsługi przez system operacyjny.

POLITECHNIKA GDAŃSKA

Metoda przerwaniowa (2)

- W wielu przypadkach sygnały przerwań wysyłane są w celu poinformowania systemu operacyjnego, że dana jest dostępna do odczytania i przesłania do komputera, lub że urządzenie jest w stanie przyjąć daną przesłaną z komputera.

POLITECHNIKA GDAŃSKA

Metoda przerwaniowa (3)

- Procesor po otrzymaniu sygnału przerwania przerywa wykonywanie bieżącego programu i rozpoczyna wykonywanie innego programu (zazwyczaj wchodzącego w skład systemu operacyjnego) — program ten identyfikuje otrzymany sygnał przerwania i podejmuje odpowiednie działania w stosunku do urządzenia, które wysłało przerwanie.
- Po wykonaniu tych działań procesor powraca do dalszego wykonywania przerwanego programu.

POLITECHNIKA GDAŃSKA

Zalety i wady metody przerwaniowej (1)

- Zalety obsługi urządzeń przy użyciu przerwań są szczególnie dobrze widoczne w systemach wielozadaniowych (wieloprocesowych), w których w trakcie oczekiwania na gotowość urządzenia procesor może wykonywać inne czynności (inne programy).
- Przy szybkiej transmisji danych i dużej liczbie zgłaszanych przerwań może wystąpić znaczne obciążenie procesora spowodowane koniecznością przełączania kontekstu (program użytkowy / system operacyjny) — w takich przypadkach lepszym rozwiązaniem może być metoda aktywnego oczekiwania.

POLITECHNIKA GDAŃSKA

Zalety i wady metody przerwaniowej (2)

- Przyjmuje się, że metoda przerwaniowa jest odpowiednia dla niezbyt szybkich urządzeń (do kilku tysięcy przerwań na sekundę).
- Realizacja metody przerwaniowej wymaga tworzenia dość skomplikowanego kodu wchodzącego w skład systemu operacyjnego (obsługa przerwań, przełączanie kontekstu, operacje na procesach).

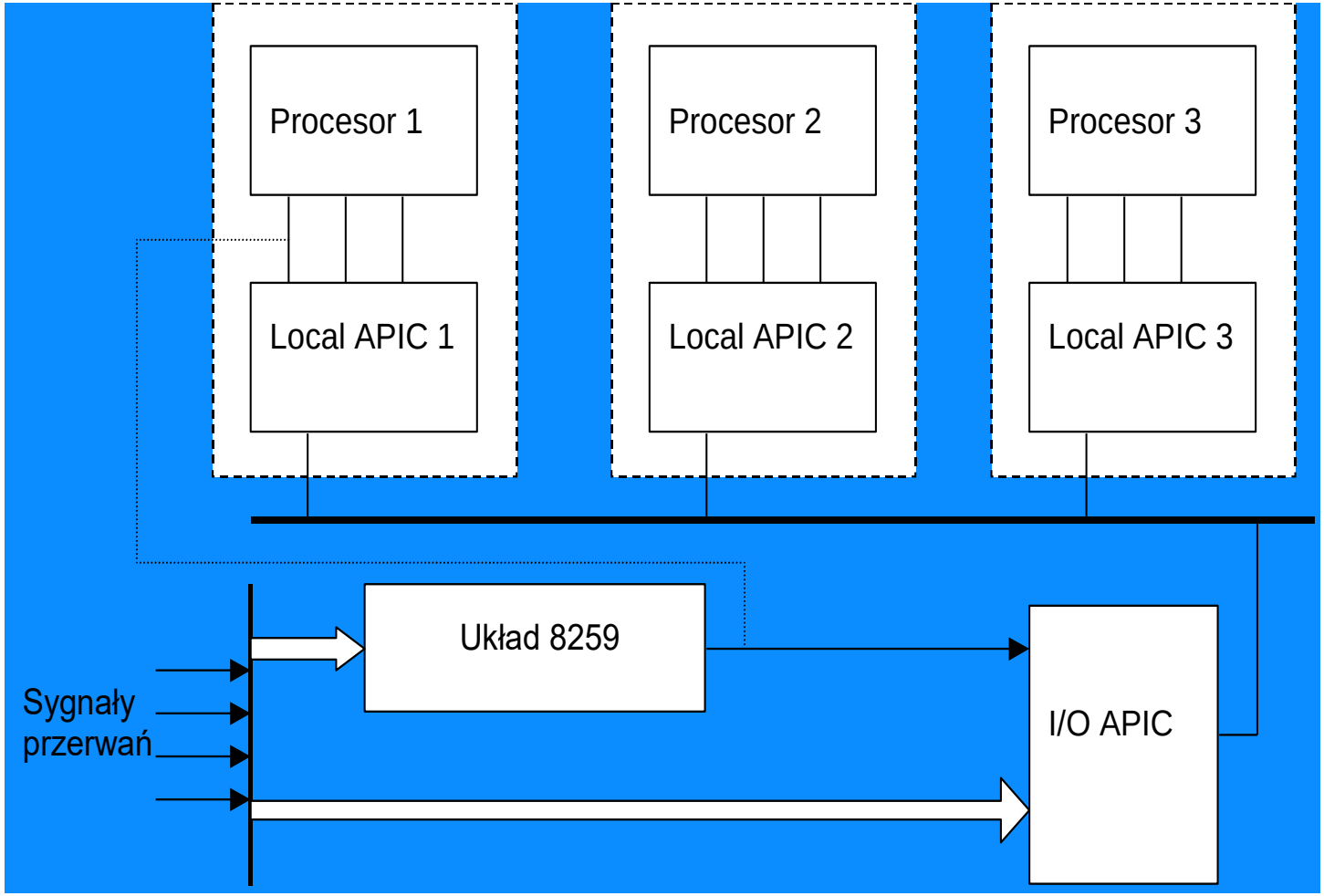
POLITECHNIKA GDAŃSKA

Obsługa przerwań w architekturze x86 (1)

- Procesor po otrzymaniu sygnału przerwania przerywa wykonywanie bieżącego programu i rozpoczyna wykonywanie innego programu, związanego z obsługą zdarzenia.
- Ponieważ po pewnym czasie procesor powróci do wykonywania przerwanego programu, obsługa przerwania musi być tak przeprowadzona, by możliwe było wznowienie pierwotnego programu.
- W tym celu procesor zapisuje na stosie *ślad* zawierający adres rozkazu, który miał wykonany jako następny, ale na razie nie został wykonany ze względu na przerwanie.

POLITECHNIKA GDAŃSKA

Obsługa przerwań w architekturze x86 (2)



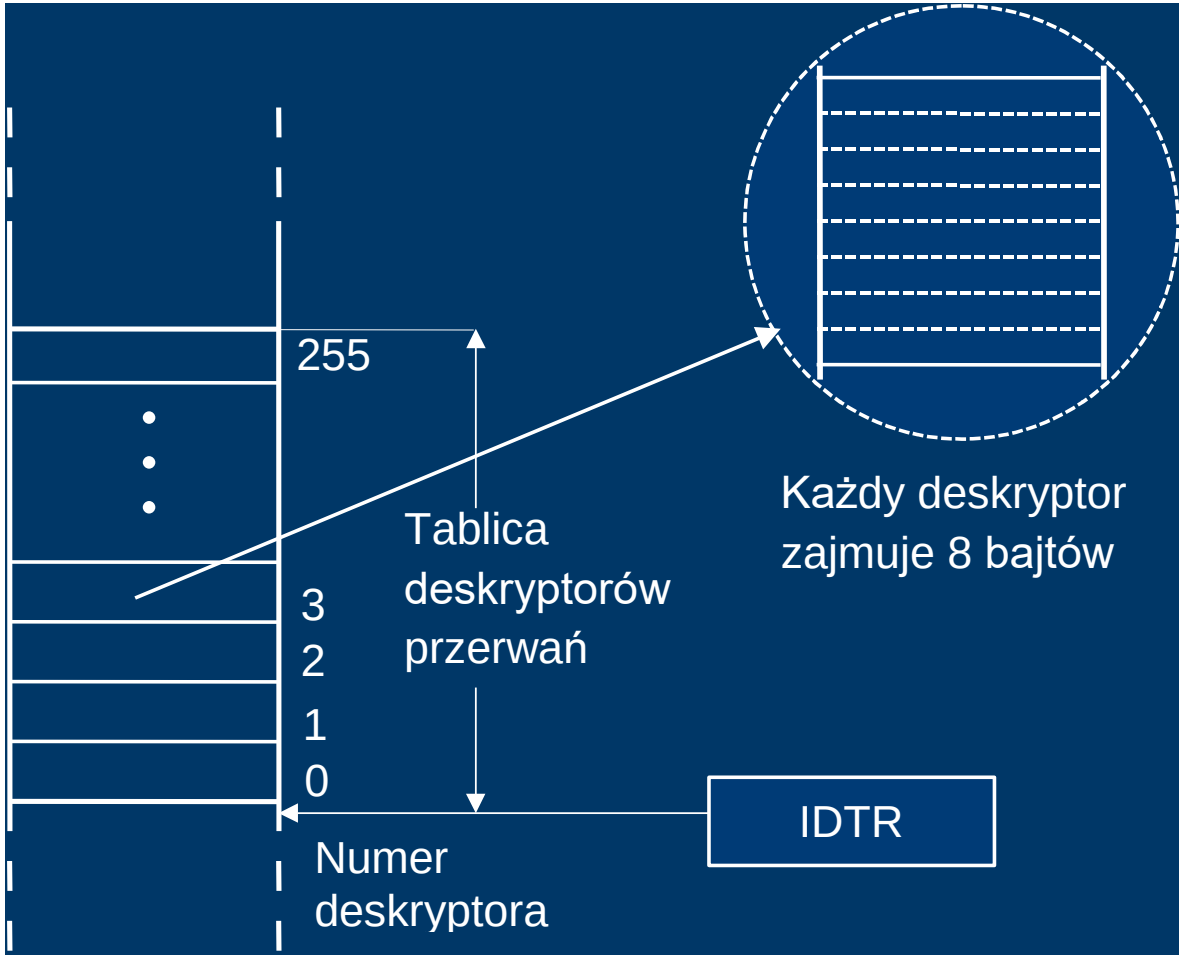
POLITECHNIKA GDAŃSKA

Obsługa przerwań w architekturze x86 (3)

- Sygnał przerwania, poprzez jedną z linii IRQ 0, IRQ 1, ... kierowany jest do układu APIC, który wspomaga procesor w obsłudze przerwań.
- Na podstawie numeru linii IRQ wyznaczany jest odpowiedni wiersz w tablicy adresowej nazywanej *tablicą deskryptorów przerwań*.
- Skojarzenie numerów wierszy i numerów linii wykonywane jest przez system operacyjny w trakcie inicjalizacji, np. w systemie Linux linia IRQ 1, przez którą przesyłane są sygnały przerwań z klawiatury, jest skojarzona w wierszem nr 33 tablicy deskryptorów przerwań.

POLITECHNIKA GDAŃSKA

Obsługa przerwań w architekturze x86 (4)



POLITECHNIKA GDAŃSKA

Obsługa przerwań w architekturze x86 (5)

- Typowe przyporządkowanie linii IRQ i numerów deskryptorów przerwań stosowane w systemie Linux:

IRQ	Nr deskr.	Urządzenie
0	32	zegar systemowy
1	33	klawiatura
3	35	drugi port szeregowy
4	36	pierwszy port szeregowy
5	37	karta dźwiękowa
6	38	napęd dyskietek
7	39	port równoległy
8	40	zegar czasu rzeczywistego (RTC)
11	43	interfejs sieciowy
12	44	mysz PS/2
13	45	koprocesor arytmetyczny
14	46	pierwszy sterownik dysków
15	47	drugi sterownik dysków

POLITECHNIKA GDAŃSKA

Obsługa przerwań w architekturze x86 (6)

- W kolejnym kroku procesor odczytuje odpowiedni adres zawarty w tablicy deskryptorów przerwań.
- Adres zawarty w deskryptorze wskazuje położenie podprogramu obsługi przerwania — adres ten wpisywany jest do rejestru EIP.
- Ponadto zerowany jest znacznik IF, co blokuje przyjmowanie innych przerwań.
- Na tym kończy się operacja przyjęcia przerwania, procesor kontynuuje pracę wg zwykłych reguł, tj. pobiera rozkaz do wykonania z komórki pamięci wskazanej przez wskaźnik instrukcji EIP.

POLITECHNIKA GDAŃSKA

Podprogramy obsługi przerwań (1)

- Program obsługi przerwania, którego adres odczytywany jest z tablicy deskryptorów przerwań, musi być opracowany bardzo starannie — podstawowym wymaganiem jest pozostawienie rejestrów procesora na końcu podprogramu obsługi w takim samym stanie, w jakim znajdowały się na początku podprogramu.
- W przeciwnym razie rejestry przerwanej aplikacji mogłyby zostać zmienione, wskutek czego jej dalsze działanie byłoby błędne.
- Błędy takie mają różne objawy i ujawniają się niekiedy po wielu miesiącach eksploatacji.

POLITECHNIKA GDAŃSKA

Podprogramy obsługi przerwań (2)

- Na końcu podprogramu obsługi przerwania umieszczony jest rozkaz IRET, który pobiera ślad wcześniej zapamiętany na stosie i wpisuje go do rejestru EIP — w rezultacie następuje wznowienie wykonywania przerwanego programu.

POLITECHNIKA GDAŃSKA

Priorytety przerwań (1)

- W sytuacji, gdy nadeszło kilka sygnałów przerwań, układ obsługi wybiera przerwanie o najwyższym priorytecie.
- Zazwyczaj sygnały przerwań dochodzące z urządzeń szybkich mają wyższy priorytet.
- Obsługa przerwań w odpowiedniej kolejności jest kluczowym problemem w systemach czasu rzeczywistego.
- Możliwe jest również przerwanie programu obsługi przerwania, jeśli nadejdzie przerwanie o wyższym priorytecie i podjęcie obsługi tego "ważniejszego”.

Priorytety przerwań (2)

- W architekturze x86 przyjęcie przerwania powoduje wyzerowanie znacznika IF, co blokuje przyjmowanie dalszych przerwań.
- Podprogram obsługi przerwań może jednak ustawić znacznik IF w stan 1 (za pomocą rozkazu STI), co otwiera możliwość przerwania podprogramu obsługi przez przerwanie o wyższym priorytecie.

POLITECHNIKA GDAŃSKA

Przerwania maskowalne i niemaskowalne

- Omówione wyżej przerwania mogą być blokowane poprzez wyzerowanie znacznika IF, wobec czego zaliczane są do klasy przerwań *maskowalnych*.
- Procesor może też przyjmować przerwania *niemaskowalne*, które nie mogą być blokowane.
- Przerwania niemaskowalne (ang. NMI — non-maskable interrupt) stosuje do sygnalizacji zdarzeń wymagających natychmiastowej obsługi niezależnie od stanu systemu.

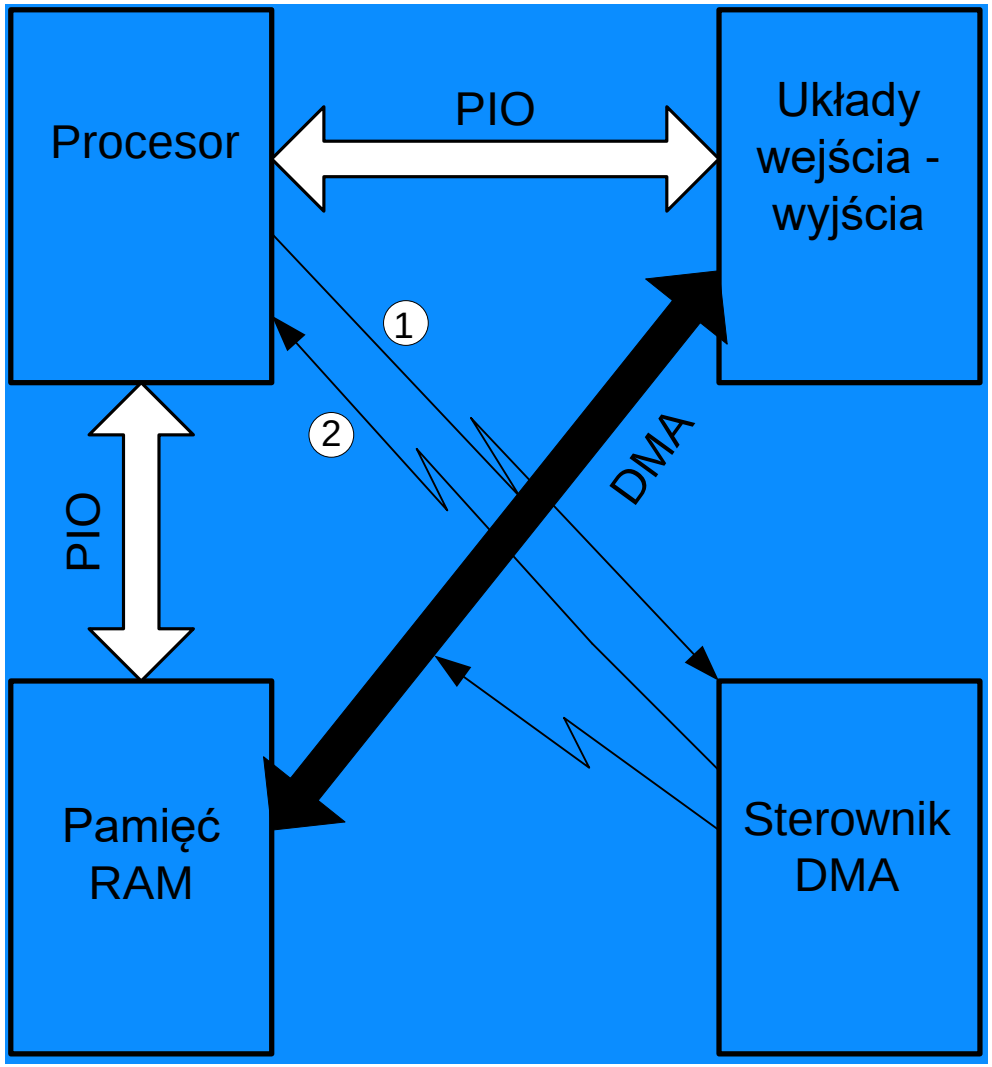
POLITECHNIKA GDAŃSKA

Bezpośredni dostęp do pamięci — układy DMA (1)

- Sterowniki DMA (ang. Direct Memory Access) umożliwiają bezpośrednie przesyłanie danych z urządzenia do pamięci głównej (operacyjnej) lub z pamięci do urządzenia.
- Przesyłanie odbywa się bez udziału procesora – trzeba jedynie odpowiednio zainicjalizować układ DMA (strzałka ① na rysunku).
- Po przesłaniu wszystkich bajtów sterownik DMA generuje przerwanie sprzętowe sygnalizujące koniec przesyłania (strzałka ② na rysunku).

POLITECHNIKA GDAŃSKA

Bezpośredni dostęp do pamięci — układy DMA (2)



POLITECHNIKA GDAŃSKA

Bezpośredni dostęp do pamięci — układy DMA (3)

- Symbol PIO (ang. Programmed Input–Output) na rysunku oznacza zwykłą transmisję za pośrednictwem procesora.
- Inicjalizacja układu DMA wymaga:
  - wpisania adresu początkowego przesyłanego obszaru pamięci do rejestru adresu sterownika DMA,
  - wpisania długości bloku danych do licznika przesyłanych danych.

POLITECHNIKA GDAŃSKA

Bezpośredni dostęp do pamięci — układy DMA (4)

- Procedura obsługi przerwania z DMA wywoływana jest po przesłaniu bloku danych, a nie przy każdej transmisji — obsługa transmisji zajmuje więc bardzo mało czasu procesora (procesor nie nadzoruje przesyłania każdej danej).
- Układy DMA są powszechnie używane do przesyłania danych do/z szybkich urządzeń, w szczególności w komputerach PC do/z: dysków, sterowników sieci lokalnej, sterowników dźwiękowych i graficznych, sterowników USB.
- Istotne trudności w korzystaniu z DMA pojawiają się w komputerach, w których używana jest (omawiana później) pamięć podręczna.

POLITECHNIKA GDAŃSKA

Wyjątki procesora (1)

- W trakcie wykonywania programu przez procesor występują sytuacje uniemożliwiające dalsze wykonywanie programu, np. niezidentyfikowany kod rozkazu, próba zmiany zawartości lokacji poza dozwolonym adresem, itd. — wystąpienie takich sytuacji powoduje wygenerowanie *wyjątku* (ang. exception) przez procesor.
- O ile przerwania powstają wskutek zdarzeń zewnętrznych w stosunku do procesora, to wyjątki związane są z wykonywaniem rozkazów przez procesor — niekiedy wyjątki są klasyfikowane jako przerwania sprzętowe wewnętrzne.

Wyjątki procesora (2)

- Wyróżnia się trzy rodzaje wyjątków:
  1. niepowodzenia (ang. faults) – aktualnie wykonywany rozkaz spowodował błąd; można ewentualnie ponowić wykonywanie tego rozkazu na podstawie śladu przechowywanego na stosie;
  2. pułapki (ang. traps) – używane m.in. w debuggerach, np. gdy procesor powraca do wykonywania przerwanego kodu wykonuje rozkaz następny po tym, który spowodował wyjątek;
  3. błędy nienaprawialne (ang. aborts) — nie można zlokalizować błędnego rozkazu ani kontynuować programu.

POLITECHNIKA GDAŃSKA

Wyjątki procesora (3)

- W architekturze x86 poszczególne typy wyjątków są na stałe skojarzone z ustalonymi wierszami tablicy deskryptorów przerwań, np. deskryptor nr 0 zawiera adres podprogramu obsługi wyjątku generowanego po wystąpieniu nadmiaru przy dzieleniu, a deskryptor nr 6 zawiera adres podprogramu uruchamianego po napotkaniu niezidentyfikowanego kodu rozkazu.
- Obsługa wyjątku przebiega podobnie jak obsługa przerwania: zapamiętanie śladu na stosie, wyzerowanie znacznika IF, uruchomienie podprogramu obsługi wyjątku.

Wyjątki procesora (4)

- Przykładowe wyjątki:
  - wyjątek nr 0 błąd dzielenia generowany, jeśli w trakcie wykonywania rozkazu DIV lub IDIV wystąpił nadmiar lub dzielnik był równy zero;
  - wyjątek nr 6 niedozwolony kod — generowany przy próbie wykonania rozkazu o kodzie nierozpoznanym przez procesor;
  - wyjątek nr 7 urządzenie niedostępne — generowany przy próbie wykonania rozkazu odnoszącego się do niedostępnego urządzenia; przykładowo, w pewnych sytuacjach koprocesor arytmetyczny może być tymczasowo niedostępny (bit TS=1 w rejestrze CR0) — próba wykonania rozkazu koprocesora powoduje wygenerowanie tego wyjątku;

Wyjątki procesora (5)

- Przykładowe wyjątki:
  - wyjątek nr 11 — brak segmentu — generowany w przypadku załadowania selektora do jednego z rejestrów segmentowych, wskazującego deskryptor, w którym bit P = 0;
  - wyjątek nr 13 — błąd ochrony (ang. general protection) — generowany w przypadku próby naruszenia niedostępnych zasobów; wyjątek ten jest używany w sytuacji jeśli próba naruszenia nie może być zaklasyfikowana bardziej precyzyjnie;
  - wyjątek nr 14 — błąd stronicowania — generowany, jeśli odwołanie dotyczy strony aktualnie nieobecnej w pamięci operacyjnej.

POLITECHNIKA
GDAŃSKA

HISTORIA MĄDROŚCIĄ
PRZYSZŁOŚĆ WYZWANIEM