

Przerwania w systemie mikroprocesorowym

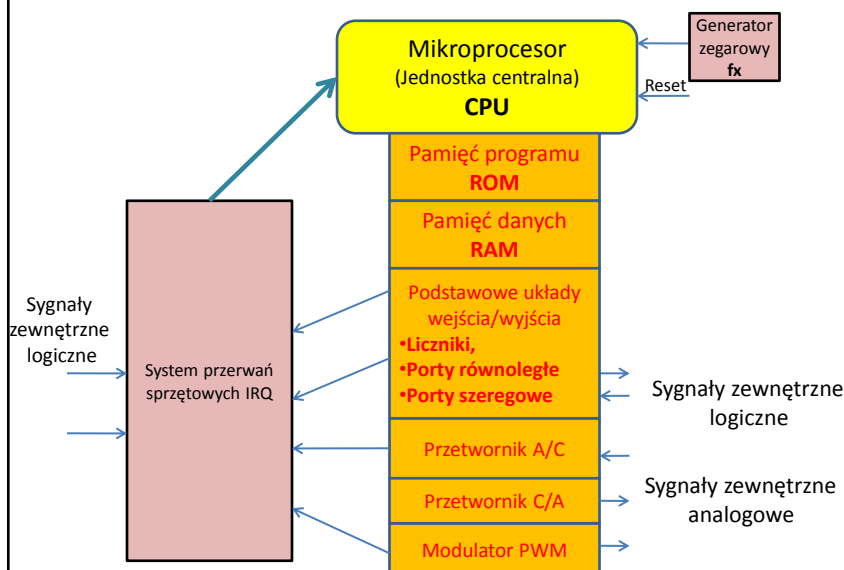
Obsługa urządzeń wejścia/wyjścia

- W każdym systemie mikroprocesorowym oprócz pamięci programu i pamięci danych znajduje się szereg układów lub urządzeń wejścia/wyjścia, które są niezbędne do komunikacji systemu:
 - np. z użytkownikiem - przyciski, klawiatura, myszka, monitor
 - lub do komunikacji z innym systemem mikroprocesorowym, np. porty szeregowo lub równoległe do transmisji danych.
- Oprócz tego każdy system zawiera podstawowe układy wejścia/wyjścia:
 - układy czasowo-licznikowe,
 - Porty równoległe do, których doprowadzone są sygnały zewnętrzne,
 - porty do transmisji szeregowo,
 - przetwornik analogowo-cyfrowy,

Obsługa urządzeń wejścia/wyjścia

- Każde z tych układów wymaga „obsługi”, na co się składa:
 - wstępne zaprogramowanie, np. ustawienie parametrów i trybu pracy, jest to przeważnie realizowane jednokrotnie na początku programu.
 - późniejsza obsługa wymaga już wielokrotnego i cyklicznego wykonywania pewnych operacji (fragmentu programu) przez jednostkę centralną aby efekt działania układu wejścia/wyjścia nie był stracony lub by nie nastąpiło nieprawidłowe działanie programu w systemie, a tym samym nieprawidłowe działanie urządzenia.
- Czas reakcji systemu na obsługę zdarzeń w układach wejścia/wyjścia jest też bardzo ważny. Cza ten powinien być przewidywalny.
- Każdy układ wejścia/wyjścia zawiera przynajmniej jeden bit (tzw. flagę) informującą system o potrzebie przeprowadzenia obsługi.

System mikroprocesorowy



Obsługa sytuacji wyjątkowych

- System mikroprocesorowy musi czasami reagować na sygnały zewnętrzne (zdarzenia zewnętrzne) w ściśle określonym czasie. Sygnały te mogą występować w sposób cykliczny lub mogą mieć charakter przypadkowy (losowy).
- Od czasu do czasu mogą się zdarzyć sytuacje wyjątkowe w systemie mikroprocesorowym:
 - pojawienie się sygnału zerowania (Reset) od jakiegoś układu lub elementu w systemie,
 - błąd programowy polegający na dzieleniu przez zero,
 - próba wykonania skoku do adresu w pamięci programu lub w pamięci danych, którego nie ma w systemie,
 - próba wykonania skoku do zarezerwowanego obszaru adresu w pamięci programu lub w pamięci danych,
 - błąd parzystości przy zapisie/odczytaniu pamięci RAM,
 - błąd działania CPU, i elementów wchodzących w skład CPU, błąd jednostki FPU.
- Na takie zdarzenia dobrze zaprogramowany i zaprojektowany system mikroprocesorowy, powinien tak zareagować aby nie nastąpiło np. zawieszenie systemu i tym samym zaprzestanie działania jakiegoś urządzenia lub nieprzewidywalne działanie urządzenia.

Metoda odpytywania

- Najprostszym sposobem obsługi poszczególnych układów wejścia/wyjścia jest ciągłe sprawdzanie flag (bitów) w kolejnych układach czy nie wymagają obsługi (tzw. metoda odpytywania, z ang. polling).
- Jest to proste rozwiązanie ale zabiera dużo czasu jednostce centralnej i tracone jest dużo czasu na oczekiwanie że flaga zostanie ustawiona przez układ wejścia/wyjścia. Jest bardzo nieefektywna metoda obsługi układów lub urządzeń wejścia/wyjścia.
- W tej metodzie występuje np. niebezpieczeństwo spóźnienia odczytania już odebranego przez port szeregowy znaku. W takim przypadku nie odczytany na czas znak i uległ zapisaniu nowo odebrany znakiem. Aby metoda ta działała poprawnie, program odbierający znaki musi wystarczająco często sprawdzać, czy na wejściu pojawił się nowy znak - jest to strata czasu przez CPU.

Metoda przerwań

- Lepszym rozwiązaniem jest tak skonstruowanie (zaprojektowanie) systemu mikroprocesorowego i jego oprogramowanie aby poszczególne układy zgłaszały potrzebę obsługi do jednostki centralnej tylko wtedy kiedy będzie taka potrzeba.
- Wtedy jednostka centralna przerywa aktualnie wykonywany fragment programu i skacze do innego miejsca w pamięci programu i wykonuje fragment programu, który „obsługuje” układ wejścia/wyjścia, który zgłosił „przerwanie”. Po zakończeniu obsługi, CPU bezpiecznie wraca do miejsca w pamięci programu, w którym nastąpiło „przerwanie”.

Semestr zimowy 2017/2018, WIEIK- PK

7

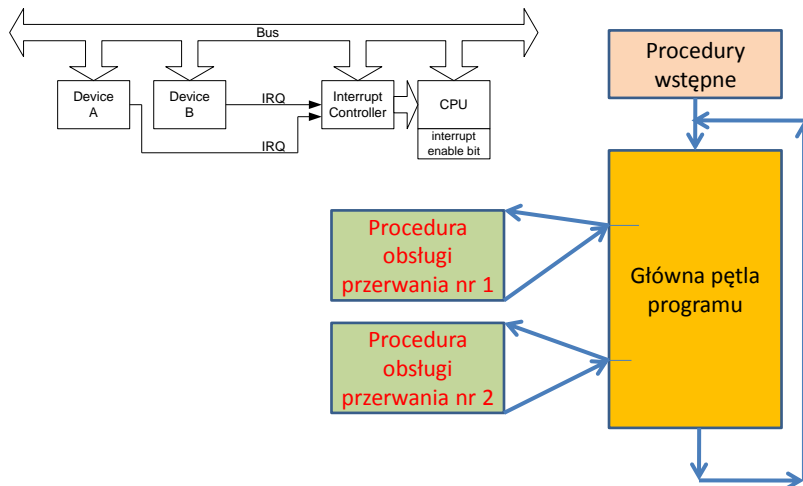
Metoda przerwań

- Ten sposób obsługi urządzeń lub układów wejścia/wyjścia nazywamy systemem przerwań z ang. *Interrupts System*.
- Najczęściej w ten sposób obsługuje się układy wejścia/wyjścia i wtedy mówimy o przerwaniach sprzętowych.
- Program do obsługi „przerwania” związany jest z odczytem/zapisaniem rejestrów układu wejścia/wyjścia, wykonaniem pewnych obliczeń i odczytem/zapisem danych do pamięci RAM lub odczytem/zapisem innych układów wejścia/wyjścia.

Semestr zimowy 2017/2018, WIEIK- PK

8

System przerwań



Semestr zimowy 2017/2018, WIEiK- PK

9

Rodzaje przerwań

Przerwania można podzielić na kilka kategorii:

- sytuacje wyjątkowe (ang. exception interrupts), tzw. **przerwania niemaskowalne (ang. non-maskable interrupt, NMI)**.
- **przerwania maskowalne sprzętowe (ang. event interrupts)**,
- **przerwania programowe (ang. software interrupts)**,
- **pułapki (ang. traps)**,
- **przerwania między-procesorowe (ang. Inter-processor interrupt - IPI)** specjalny typ przerwania wysyłany z jednego procesora do drugiego w systemach multiprocesorowych,
- **przerwania niepożądane (zakłócające) (ang. Spurious interrupt)** – niepożądane przerwanie działania systemu mikroprocesorowego powstające poza systemem, np. poprzez zakłócenia elektromagnetyczne, zakłócenia napięcia zasilania, niepoprawne sygnały elektryczne w systemie, błędne działanie innych urządzeń,

Semestr zimowy 2017/2018, WIEiK- PK

10

Rodzaje przerwań

Przerwania można podzielić na źródło pochodzenia:

- przerwania wewnętrzne
 - przerwania pochodzące z jednostki CPU
 - przerwania pochodzące z układów lub urządzeń na płycie głównej,
- przerwania zewnętrzne
 - przerwania pochodzące z zewnętrznych urządzeń dołączonych do systemu mikroprocesorowego
 - przerwania pochodzące z zewnętrznych urządzeń dołączonych do mikrokontrolera

W mikrokontrolerach liczba źródeł przerwań wewnętrznych jest większa od liczby źródeł przerwań zewnętrznych.

Przerwania niemaskowalne

- **Przerwania niemaskowalne** – tzn. użytkownik (programista) nie może je zablokować ani sprzętowo ani programowo. Programista musi uwzględnić programy do obsługi tych przerwań w oprogramowaniu całego systemu. Są to przerwania najszybciej obsługiwane o najwyższym priorytecie.
- Do przerwań niemaskowalnych zaliczymy następujące zdarzenia:
 - sygnały z CPU
 - dzielenie przez zero,
 - błąd parzystości,
 - przepełnienie (ang. overflow),
 - przepełnienie stosu,
 - błędny kod rozkazu (błąd odczytu pamięci programu),
 - błąd w dostępie do pamięci
 - sygnały RESETU pochodzące z układu zerowania (zewnętrznego lub wewnętrznego), układu monitorowania napięcia zasilania, układu tzw. watchdoga, układu monitorowania generatora zegarowego,

Przerwania maskowalne

- **Przerwania maskowalne** programista może odblokować lub zablokować w sposób programowy, poprzez ustawienie lub zerowanie odpowiednich bitów w rejestrach do ustawiania systemu przerwań.
- W mikrokontrolerach system przerwań sprzętowych uwzględnia sygnały przerwań od układów wewnętrznych a także sygnały zewnętrzne pochodzące od urządzeń podłączonych do mikrokontrolera.
- Przerwania maskowalne sprzętowe są zgłaszane przez układy wewnętrzne mikrokontrolera systemu, np.:
 - układy czasowo-licznikowe (Timers, Counters),
 - porty szeregowo (UART, USART, SPI, I2C, CAN..),
 - przetwornik A/C,
- Przerwania maskowalne sprzętowe są też zgłaszane do systemu mikroprocesorowego przez urządzenia lub układy zewnętrzne, np.:
 - klawiaturę,
 - napędy dysków,
 - karty sieciowe,
 - karty dźwiękowe
- Reakcja na sygnały zewnętrzne (przerwania zewnętrzne) może być ustawiona:
 - reakcja na zbocze opadające, lub zbocze narastające,
 - reakcja na zmianę stanu logicznego, na stan logiczny „0” lub „1”

Przerwania programowe

- **Przerwania programowe** są wywoływane przez program (świadome działanie programisty).
- W kodzie programu umieszcza się rozkaz który wywołuje procedurę do obsługi przerwania.
- Przerwania programowe najczęściej wykorzystywane są do komunikacji z systemem operacyjnym, np. w komputerze klasy PC, w procedurze obsługi przerwania (np. w DOS 21h, 2fh, Windows 2fh, Linux x86 przerwanie 80h) umieszcza się kod wywołujący odpowiednie funkcje systemowe w zależności od zawartości rejestrów ustawionych przez program wywołujący.
 - Np. w systemie DOS przerwanie int 21h z parametrem ah=01h
 - Czytanie znaku z echem z standardowego wejścia (klawiatura i ekran monitora). Funkcja oczekuje na znak w standardowym strumieniu wejściowym. Po odebraniu znaku funkcja w rejestrze AL zwraca kod tego znaku.

Pułapki

- **Traps (pułapki)** – sytuacja, która nie jest błędem, jej wystąpienie ma na celu wykonanie określonego kodu. Wykorzystywane przede wszystkim w debuggerach - testowanie oprogramowania, śledzenie wykonywanego programu, krokowe wykonywanie programu.
- Procedura do obsługi pułapki polega najczęściej na wyprowadzeniu zawartości wybranych rejestrów procesora lub zawartości fragmentu pamięci RAM na zewnątrz systemu np. poprzez port szeregowy lub złącze JTAG do komputera typu PC na którym można podglądać zmiany zachodzące w rejestrach i pamięci RAM. Dzięki temu można znaleźć błędy działania w programie.
- Po zakończeniu obsługi pułapki procesor powraca do wykonywania przerwanej kodu.

Priorytet przerwania

Priorytet przerwania.

- Każde przerwanie ma określony priorytet (ważność). Jeśli zostanie zgłoszonych jednocześnie kilka przerwania, to jako pierwsze zostanie obsłużone przerwanie o najwyższym priorytecie. Po zakończeniu procedury obsługi przerwania obsługiwane jest następnie przerwanie o najwyższym priorytecie.
- W większości przypadków przerwania niemaskowalne mają najwyższy priorytet.
- Priorytet poszczególnych przerwania na ogół jest już ustalony przez projektanta systemu mikroprocesorowego (producenta mikroprocesora lub mikrokontrolera) ale może być modyfikowany w pewnym zakresie przez programistę.
- Np. dla mikrokontrolera 8051 priorytet przerwania maskowalnych:
 1. **Najwyższy dla przerwania zewnętrznego INT0,**
 2. przerwanie z licznika T0,
 3. przerwanie zewnętrznego INT1,
 4. przerwanie z licznika T1,
 5. Najniższy dla przerwania z portu szeregowego,

Wektor przerwań

Wektor przerwań.

- Kiedy wywoływane jest przerwanie, do licznika rozkazów w CPU ładowany jest adres procedury obsługi przerwania (adres pierwszego rozkazu programu do obsługi przerwania).
- Adres ten pobierany jest z tabeli umieszczonej w pamięci programu zwanej wektorem przerwań. Wektor przerwań zawiera adresy procedur obsługi przerwań oraz adres procedury wywoływanej po resecie mikrokontrolera.
- Wektor przerwań (początkowe adresy programów do obsługi przerwań) może być ustalony na stałe przez producenta mikroprocesora/mikrokontrolera lub w niektórych przypadkach programista może narzucić własne adresy procedur do obsługi przerwania.
- W większości przypadków i tak pod pierwszym adresem procedury do obsługi przerwania umieszcza się rozkaz skoku do innego miejsca w pamięci programu.

Wektor przerwań

W zależności od budowy i stopnia skomplikowania systemu mikroprocesorowego po pojawieniu się sygnału przerwania jednostka centralna:

- dokładnie wie z którego urządzenia pojawił się sygnał żądania obsługi przerwania i skacze do pierwszego adresu w pamięci programu obsługującego przerwanie,
- CPU rozpoznaje że jest zgłoszone przerwanie ale nie wie z jakiego miejsca (od jakiego urządzenia). Wtedy program (świadome działanie programisty) musi rozpoznać skąd pochodzi zgłoszenie przerwania i skoczyć do procedury obsługi przerwania.
- Np. dla mikrokontrolera 8051 wektor przerwań:
 - 0000H - przerwanie niemaskowalne RESET
 - 0003H - dla przerwania zewnętrznego INTO,
 - 000BH - dla przerwania z licznika T0,
 - 0013H - dla przerwania zewnętrznego INT1,
 - 001BH - dla przerwania z licznika T1,
 - 0023H - dla przerwania z portu szeregowego,

Table 18. Reset and Interrupt Vectors

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	\$000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	\$002	INT0	External Interrupt Request 0
3	\$004	INT1	External Interrupt Request 1
4	\$006	INT2	External Interrupt Request 2
5	\$008	TIMER2 COMP	Timer/Counter2 Compare Match
6	\$00A	TIMER2 OVF	Timer/Counter2 Overflow
7	\$00C	TIMER1 CAPT	Timer/Counter1 Capture Event
8	\$00E	TIMER1 COMPA	Timer/Counter1 Compare Match A
9	\$010	TIMER1 COMPB	Timer/Counter1 Compare Match B
10	\$012	TIMER1 OVF	Timer/Counter1 Overflow
11	\$014	TIMER0 COMP	Timer/Counter0 Compare Match
12	\$016	TIMER0 OVF	Timer/Counter0 Overflow
13	\$018	SPI, STC	Serial Transfer Complete
14	\$01A	USART, RXC	USART, Rx Complete
15	\$01C	USART, UDRE	USART Data Register Empty
16	\$01E	USART, TXC	USART, Tx Complete
17	\$020	ADC	ADC Conversion Complete
18	\$022	EE_RDY	EEPROM Ready
19	\$024	ANA_COMP	Analog Comparator
20	\$026	TWI	Two-wire Serial Interface
21	\$028	SPM_RDY	Store Program Memory Ready

**Przerwania i wektory
przerwań
w mikrokontrolerze
ATMEGA32, firmy Atmel**

19

Mechanizm obsługi przerwania

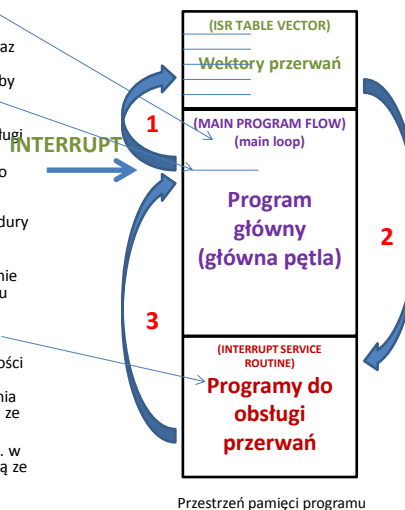
- Pierwszą czynnością wykonywaną po wystąpieniu przerwania jest zachowanie obecnego stanu programu, tak by było możliwe jego odtworzenie po zakończeniu obsługi przerwania i bezpieczny powrót do przerwonego programu.
- Dokonywane jest to poprzez automatyczne zachowanie na stosie: licznika rozkazów, akumulatora, odpowiednich rejestrów oraz rejestru stanu (rejestru kontrolnego).
- Niektóre rejestry i fragmenty pamięci RAM muszą być odłożone na stos samodzielnie przez programistę (lub zapewnia to kompilator np. języka C).
- Po zachowaniu stanu procesora procedura obsługi przerwania może korzystać z wyżej wymienionych rejestrów bez niebezpieczeństwa zakłócenia pracy programu głównego.
- Aby zapobiec przerywaniu procedury obsługi przerwania przez inne przerwania można zablokować inne przerwania. Czynności wykonywane przez procedurę obsługi przerwania zależą od źródła przerwania. W większości przerwania od urządzeń peryferyjnych konieczne jest wyzerowanie znacznika przerwania. Zerowanie tego znacznika najczęściej wykonywane jest poprzez odczyt odpowiedniego rejestru statusowego, a następnie zapis lub odczyt rejestru stowarzyszonego z danym urządzeniem peryferyjnym. Na przykład skasowanie flagi przerwania od portu szeregowego wykonywane jest przez odczyt rejestru danych.

Semestr zimowy 2017/2018, WIEIK- PK

20

Prosty mechanizm obsługi przerwania

- Rysunek przedstawia prosty sposób obsługi przerwania przez system i rozmieszczenie poszczególnych fragmentów programu w pamięci programu ROM.
- Główny program (główna pętla programowa)
- Kiedy przerwanie (1) pojawi się w dowolnym momencie, program w pętli głównej jest zatrzymany (pojedynczy rozkaz procesora jest kończony) i zapamiętywane na stosie (w pamięć RAM) jest miejsce pobrania następnego rozkazu aby bezpiecznie powrócić po obsłudze przerwania.
- Do licznika rozkazów jest ładowany adres (wektor) z pod którego będzie pobrany pierwszy rozkaz programu do obsługi przerwania.
- Od tego momentu jest wykonywany przez CPU program do obsługi przerwania. Najczęściej pierwszy rozkaz, który jest pod adresem wektora, zawiera rozkaz skoku do innego miejsca w pamięci programu, gdzie są umieszczone procedury do obsługi różnych przerw (ang. INTERRUPT SERVICE ROUTINE) (2).
- Po zakończeniu ostatniego rozkazu z przerwania (przeważnie jest to specjalny rozkaz powrotu z przerwania do programu głównego), do licznika rozkazów wczytywany jest adres rozkazu pobranego ze stosu (3).
- Oprócz zapisu na stosie adresu do powrotu do programu głównego, programista powinien zapisać na stosie zawartości najważniejszych rejestrów procesora, np. akumulatora, rejestru stanu, rejestrów roboczych. Po wyjściu z przerwania te informacje są z powrotem pobierane przez programistę ze stosu.
- Przy pisaniu programu w językach wysokiego poziomu, np. w języku C, kompilator automatycznie, odkłada na stos i ściąga ze stosu dane po powrocie do programu głównego.

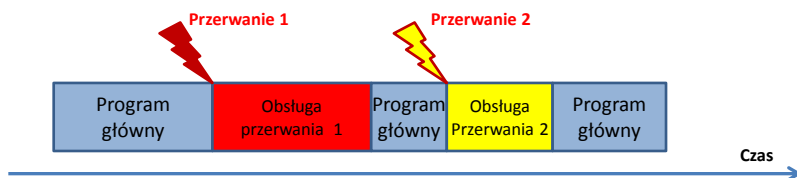


Semestr zimowy 2017/2018, WIEIK- PK

21

Mechanizm obsługi przerwania

- Czasami jedna procedura obsługi przerwania wywoływana jest przez kilka źródeł (np. przerwanie układu czasowego może być wywołane przez 5 różnych źródeł). W takim przypadku na początku procedury obsługi należy wykryć, które źródło wywołało przerwanie. Dokonujemy tego przez testowanie odpowiednich bitów w rejestrach statusowych urządzeń peryferyjnych.
- Na końcu procedury obsługi przerwania programista musi odtworzyć stan wybranych rejestrów i fragmentu pamięci RAM.
- Ostatnią instrukcją procedury obsługi jest specjalna instrukcja powrotu z przerwania. Instrukcja ta powoduje przywrócenie stanu z przed wywołania przerwania. To znaczy, zdejmuję ze stosu akumulator, wybrane rejestry i rejestr stanu oraz licznik rozkazów. Zerowany jest bit maski przerwania.

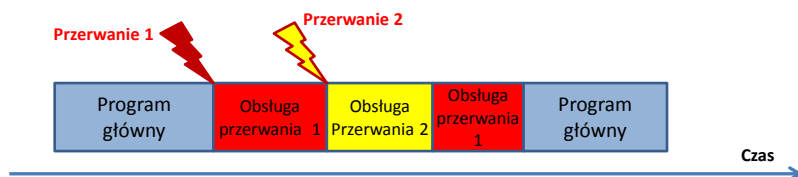


Semestr zimowy 2017/2018, WIEIK- PK

22

Czas obsługi przerwania

- Obsługa urządzeń wejścia/wyjścia i sytuacji wyjątkowych w systemie mikroprocesorowym jest jednym z najważniejszych zadań dla programisty.
- Oprogramowanie układów mikroprocesorowych przewidzianych do sterowania i kontroli w głównej mierze składa się z programów do obsługi przerwania. Program główny w takim przypadku może być nawet pustą pętlą, która nic nie robi.
- W zależności od budowy i stopnia skomplikowania systemu mikroprocesorowego czas przejścia CPU do obsługi przerwania może być stały i taki sam dla wszystkich przerwania lub w bardziej zaawansowanych systemach można wyróżnić przerwanie które są „wolniej” albo „szybciej” obsługiwane po pojawieniu się sygnału zgłoszenia (np. mikrokontrolery typu ARM).

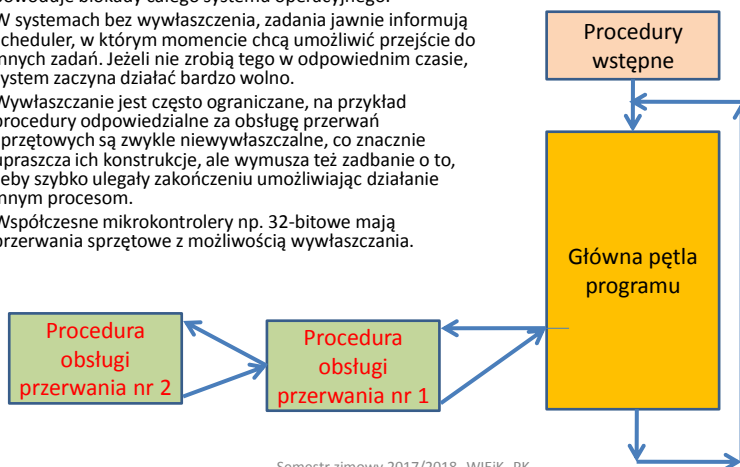


Semestr zimowy 2017/2018, WIEIK- PK

23

Przerwania sprzętowe z wyłączeniem

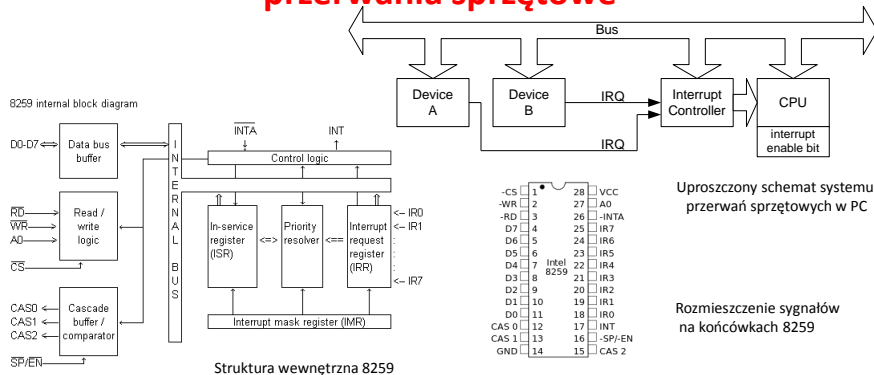
- **Wyłączenie** – technika używana w środowiskach wielozadaniowych, w której algorytm szeregujący może wstrzymać aktualnie wykonywane zadanie (np. proces lub wątek), aby umożliwić działanie innemu.
- Dzięki temu rozwiązaniu zawieszenie jednego procesu nie powoduje blokady całego systemu operacyjnego.
- W systemach bez wyłączenia, zadania jawnie informują scheduler, w którym momencie chcą umożliwić przejście do innych zadań. Jeżeli nie zrobią tego w odpowiednim czasie, system zaczyna działać bardzo wolno.
- Wyłączenie jest często ograniczane, na przykład procedury odpowiedzialne za obsługę przerwania sprzętowych są zwykle niewyłączalne, co znacznie upraszcza ich konstrukcję, ale wymusza też zadbanie o to, żeby szybko ulegały zakończeniu umożliwiając działanie innym procesom.
- Współczesne mikrokontrolery np. 32-bitowe mają przerwanie sprzętowe z możliwością wyłączenia.



Semestr zimowy 2017/2018, WIEIK- PK

24

Przerwania w komputerze klasy PC XT/AT – przerwania sprzętowe



- w pierwszych komputerach klasy PC XT za system przerwań sprzętowych był odpowiedzialny jeden specjalny układ (*ang. P.I.C. Programmable Interrupt Controller*) typu 8259A, firmy Intel,
- układ obsługiwał 8 źródeł przerwań (IRQ0 do IRQ7)
- w komputerze klasy AT, były dwa układy 8259 połączone kaskadowo

Semestr zimowy 2017/2018, WIEIK- PK

25

Przerwania w komputerze klasy PC XT/AT – przerwania sprzętowe

- W komputerze klasy PC/XT do szyny typu ISA był podłączony jeden kontroler 8259, w komputerze PC/AT z szyną typu ISA AT i późniejszych są dwa kontrolery 8259, jeden master drugi slave. Przerwania o numerach IRQ0 do IRQ7 są podłączone do układu mastera, pozostałe IRQ8 do IRQ15 slave.
- Układ master 8259 (nadrzędny) – układy i urządzenia podłączone do kontrolera przerwań sprzętowych:
 - IRQ0 – Intel 8253 lub Intel 8254 Programowalny układ czasowy (Programmable Interval Timer, system timer)
 - IRQ1 – Intel 8042 keyboard – kontroler klawiatury
 - IRQ2 – w PC/XT nie używany; w PC/AT jako połączenie kaskadowe do układu slave 8259
 - IRQ3 – 8250 UART porty szeregowo COM 2 COM 4
 - IRQ4 – 8250 UART porty szeregowo COM1 i COM3
 - IRQ5 – kontroler dysku twardego w PC/XT; Intel 8255 programowalny port równoległy 2 i 3 w PC/AT
 - IRQ6 – Intel 82072A – kontroler stacji dyskietek
 - IRQ7 – Intel 8255 programowalny port równoległy port 1
- Układ Slave 8259 (PC/AT) (podrzędny)
 - IRQ8 – zegar czasu rzeczywistego (RTC)
 - IRQ9 – nie przydzielony ale dodatkowe karty 8-bitowe wykorzystywały IRQ2
 - IRQ10 – nie przydzielony ale do wykorzystania przez inne urządzenia
 - IRQ11 – nie przydzielony ale do wykorzystania przez inne urządzenia
 - IRQ12 – Intel 8042 kontroler myszki w standardzie PS/2
 - IRQ13 – koprocesor arytmetyczny
 - IRQ14 – kontroler dysku twardego nr 1
 - IRQ15 – kontroler dysku twardego nr 2

Semestr zimowy 2017/2018, WIEIK- PK

26

Przerwania w komputerze klasy PC XT/AT – przerwania programowe

- Wraz pojawieniem się systemu operacyjnego wprowadzono przerwania programowe do komputerów klasy PC XT/ AT.
- Przerwanie programowe polega na tym, że z kodu programu głównego, wywoływana jest procedura obsługi przerwania (świadomie działanie programisty).
- Tego typu przerwania najczęściej wykorzystywane są do komunikacji z systemem operacyjnym, który w procedurze obsługi przerwania umieszcza kod wywołujący odpowiednie funkcje systemowe (podprogramy, procedury) w zależności od zawartości rejestrów ustawionych przez program wywołujący.
- Przerwania programowe np. zawierają gotowe procedury (podprogramy, funkcje) np. do obsługi urządzeń wejścia/wyjścia w systemie (na płycie głównej i do niej podłączonych).
- Podstawowe przerwania programowe są zapisane w BIOS-ie (ang. *Basic Input/Output System* – podstawowy system wejścia-wyjścia), a następne są ładowane przy starcie systemu DOS (ang. *Disk Operating System* – pierwszy przenośny (Disk) system operacyjny komputerów PC i mikrokomputerów lat 80.
- Np. przerwanie 10h jest przerwaniem należącym do systemu BIOS obsługującym ekran monitora, natomiast przerwanie 21h jest przerwaniem należącym do systemu operacyjnego DOS.
- Obecnie w komputerach klasy PC, korzysta się z funkcji API w systemie Windows.

Semestr zimowy 2017/2018, WIEIK- PK

27

Przerwania w komputerze klasy PC XT/AT – przerwania programowe

Opis wybranych funkcji przerwania o numerze int 21h z systemu DOS, wywołanie przerwania polega na wpisaniu do rejestru AH w procesorze 80x86 odpowiedniej liczby i wywołania przerwania, np. w assemblerze:

```
mov AH, 01h
int 21h
```

przerwanie	wartość w AH w hex	opis działania
INT 21	0	Zakończenie programu
INT 21	1	Wczytanie znaku z klawiatury z echem, wyprowadzenie na ekran
INT 21	2	Wyświetl znak
INT 21	3	Czekaj na dane z urządzenia pomocniczego
INT 21	4	Wypisz znak na urządzeniu pomocniczym
INT 21	5	Wypisz znak na drukarce
INT 21	6	Czytaj lub pisz na konsolę
INT 21	7	Wczytaj znak z klawiatury bez echa
INT 21	8	Wczytaj dane z konsoli bez echa
INT 21	9	Wypisz łańcuch znaków na ekran
INT 21	A	Buforowane wczytanie z klawiatury
INT 21	B	Sprawdź status STDIN
INT 21	C	Czyść bufor klawiatury, wywołaj funkcję wczytania
INT 21	D	Reset dysku

Semestr zimowy 2017/2018, WIEIK- PK

28

Przerwania w komputerze klasy PC XT/AT – przerwanie programowe INT 21h

INT 21h,01 - Wczytanie z klawiatury z echem

AH = 01

przerwanie zwraca: AL = Znak z standardowego urządzenia wejścia, czeka na dane z STDIN i wyświetla na STDOUT zwraca 0 dla rozszerzonych klawiszy, później funkcja musi być wywołana ponownie by odczytać skankod, jeśli wciśnięto Ctrl-Break, uruchamiane jest INT 23

INT 21h,02 - Wyświetl znak

AH = 02

DL = znak do wyświetlenia ,

przerwanie nie zwraca nic, wyświetla znak na STDOUT, backspace jest traktowany jak niedestruktywne przesunięcie kursora w lewo, jeśli wciśnięto Ctrl-Break, uruchamiane jest INT 23.

INT 21h,09 - Wypisz łańcuch

AH = 09

DS:DX = wskaźnik do łańcucha zakończony znakiem "\$,

przerwanie nie zwraca nic, wyświetla łańcuch znaków na STDOUT, backspace jest traktowane jako znak niedestruktywny, jeśli wciśnięto Ctrl-Break, uruchamiane jest INT 23

Przerwania w komputerze klasy PC XT/AT – przerwania programowe

```
msgZ1: db "czas systemowy: $"
msgZ2: db "początkowa wartosc DX = $"
msgX2: db "koncowa wartosc DX = $"
msgW2: db "roznica wartosc DX = $"
msg1: db "t $"
msgZ3: db "AL = $"
msgZ4: db "CX = $"
msgZwieksza: db "a > b: $,"
start:
call nowa_linia ; nowa linia
mov dx, msgZ1 ;wpisz do dx adres komunikatu
mov ah, 09h ;wpisz do ah wartość 09h
int 21h ;wywołanie przerwania int21h
mov ah, 00h ;wpisz do ah wartość 00h
int 1Ah ;wywołanie przerwania int 1Ah, odczytanie licznika zegara systemowego
mov licznik1, cx
mov licznik2, dx
mov licznik3, al
```

Przykładowy fragment kodu programu w asemblerze 80x86 do wyprowadzenia tekstu na ekran