

Marcin Stępnia

# Architektura systemów komputerowych

## Laboratorium 12

### Symulator SMS32

### Przerwania sprzętowe

#### 1. Informacje

Przerwania sprzętowe to krótkie fragmenty kodu, dostarczające użyteczne funkcje, które mogą być wywołane przez elementy sprzętu. Mogą to być urządzenia wewnętrzne komputera (np. dysk twardy), jak i urządzenia peryferyjne. Na przykład naciśnięcie klawisza na klawiaturze lub poruszanie myszą wywołuje przerwanie sprzętowe, pozwalające procesorowi odczytać wciśnięte klawisze lub położenie myszy. W odróżnieniu od przerwań programowych, przerwania sprzętowe są asynchroniczne i mogą wystąpić w środku wykonania instrukcji, co wymaga dodatkowej uwagi przy programowaniu (np. zapisywanie rejestrów bazowych i flag na stosie). Akt inicjowania przerwania sprzętowego jest określany jako żądanie przerwania (ang. interrupt request - IRQ). Po odebraniu żądania przerwania procesor przerywa normalne przetwarzanie programu i wykonuje procedurę obsługi przerwania, a po jej zakończeniu wznowia normalne przetwarzanie. Takie podejście sprawia, że przetwarzanie zdarzeń jest bardziej wydajne. Bez przerwania, procesor musiałby odpytywać urządzenia w regularnych odstępach czasu, aby sprawdzić, czy określone zdarzenie zaistniało. Przerwania mogą mieć przypisane priorytety. Na przykład dysk może mieć pierwszeństwo nad drukarką.

##### 1.1. Tablica wektorów przerwań

Wektory przerwań symulatora znajdują się w pamięci pod adresami 00..0F. Ponieważ na początku pamięci musi znaleźć się instrukcja, która

Tabela 1. Przerwania generowane przez urządzenia

Numer przerwania	Urządzenie
02	Zegar
03	Klawiatura
04	Klawiatura numeryczna

przeskoczy tablicę wektorów, to pierwszą użyteczną komórką dla wektora jest [02].

## 1.2. Wywołanie przerwania

Urządzenia wywołują przerwania o odpowiednich numerach. Numer przerwania to w rzeczywistości adres wektora przerwania. Na przykład żądanie przerwania 02 spowoduje odczytanie przez procesor wartości komórki o adresie 02. Następnie wskaźnik instrukcji jest odkładany na stosie, a program kontynuuje pracę od instrukcji znajdującej się pod adresem wskazanym przez odczytaną wartość. W ten sposób wykonywany jest kod przerwania. Jego wykonanie kończy instrukcja *IRET*, która powoduje powrót z przerwania (wskaźnik instrukcji jest przywracany ze stosu).

Gdy flaga przerwania „I” (Interrupt) nie jest ustawiona, to przerwania są ignorowane. Do ustawienia flagi „I” służy instrukcja *STI*, a instrukcja *CLI* czyści tę flagę. Po ustawieniu flagi „I” należy dodać wektory wykorzystywanych przerwań. Zawsze będzie to przynajmniej przerwanie zegara o numerze 02. Oznacza to, że pod adresem 02 w pamięci należy umieścić adres procedury obsługi przerwania zegara. W najprostszej wersji taka procedura będzie zawierała tylko instrukcję *IRET*, czyli powrót z przerwania bez wykonywania żadnych akcji.

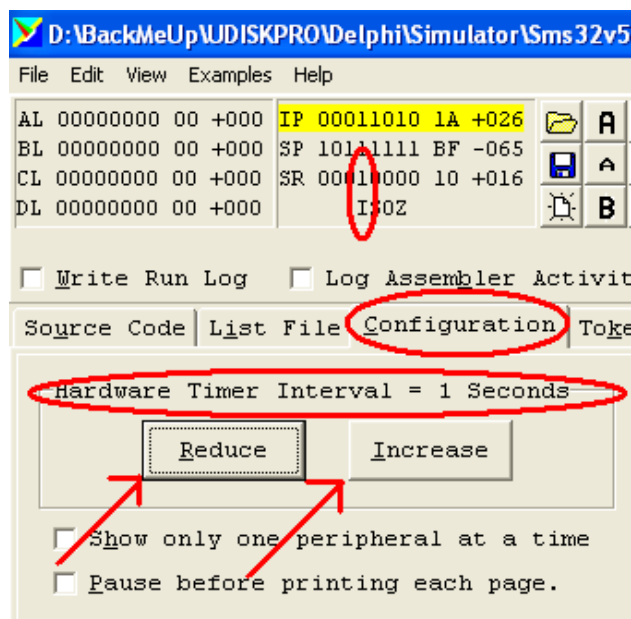
W zrozumieniu działania przerwań powinny pomóc programy przedstawione na listingach 1 i 2.

## 1.3. Zegar

Symulator SMS32 posiada zegar wyzwalający przerwanie 02 w regularnych odstępach czasu, które można ustawić w zakładce *Configuration* (zob. rys. 1).

## 1.4. Klawiatura

W celu wyświetlenia okna reprezentującego klawiaturę (zob. rys. 2) należy wysłać dowolne dane na port 07 (instrukcja *OUT 07*). Po każdym naciśnię-



Rysunek 1. Konfiguracja zegara. Źródło: dokumentacja SMS32



Rysunek 2. Klawiatura generująca przerwanie. Źródło: dokumentacja SMS32

ci klawisza klawiatury, generowane jest sprzętowe przerwanie o numerze 03. Procedura obsługi przerwania powinna odczytać kod naciśniętego klawisza za pomocą instrukcji *IN 07*. Kod zostanie skopiowany do rejestru AL. Naciśnięcie klawisza może generować różne znaki. Zależy to od tego, w którym miejscu klawisza nastąpiło kliknięcie. Po najechaniu myszą nad różne obszary klawisza, generowany znak wyświetli się na dole klawiatury (z prawej strony spacji).

### 1.5. Klawiatura numeryczna

Aby wyświetlić okno reprezentujące klawiaturę numeryczną (zob. rys. 3) należy wysłać dowolne dane na port 08 (instrukcja *OUT 08*). Po każdym naciśnięciu klawisza klawiatury numerycznej, generowane jest sprzętowe przerwanie o numerze 04. Procedura obsługi przerwania powinna odczytać kod



Rysunek 3. Klawiatura numeryczna generująca przerwania. Źródło: dokumentacja SMS32

naciśniętego klawisza za pomocą instrukcji *IN 08*. Kod zostanie skopiowany do rejestru AL.

Listing 1. Szablon programu do obsługi przerwań sprzętowych

```
; -----
; Szablon programu obsługi przerwan sprzętowych
; -----
    jmp    start ; Skok za tablice wektorow przerwan
    db 10 ; Wektor przerwania zegara
    db 00 ; Wektor przerwania klawiatury (nieuzywany)
    db 20 ; Wektor przerwania klawiatury numerycznej

; ===== Procedura obsługi przerwania zegara =====
    org    10
    nop          ; Tutaj nalezy robic cos uzytecznego
    nop
    nop
    nop
    nop
    nop
    nop
    iret

; =====

; ===== Procedura obsługi przerwania klawiatury numerycznej =====
    org    20
    CLI          ; Zapobieganie wielobieznosci (czyszczenie flagi I)
    push    al
    pushf

    in    08
    nop          ; Tutaj nalezy przetworzyc naciśniety klawisz
    nop
    nop
    nop
    nop
    nop

    popf
    pop     al
    STI          ; Ponowne ustawienie flagi I
    iret
```

```

; =====
; ===== Główna petla bezczynności =====
start:
    STI      ; Ustawienie flagi I
    out     08 ; Pokazanie klawiatury numerycznej
idle:
    nop     ; Tutaj należy robić coś użytecznego
    nop
    nop
    nop
    nop
    jmp     idle
; =====
    end
; =====

```

Listing 2. Program demonstrujący działanie przerwań sprzętowych

```

; -----
; Przykład wykorzystanie przerwań sprzętowych
; Program obraca silnikiem krokowym w sposób ciągły
; i zmienia światła drogowe po każdym wystąpieniu zdarzenia zegara
; -----
    JMP     Start ; Skok za tablice wektorów przerwań
    DB 50 ; Wektor pod adresem 02 wskazujący na adres 50

Start:
    STI      ; Ustawienie flagi I. Włączenie obsługi przerwań sprzętowych
Rep:
    MOV     AL,1
    OUT     05 ; silnik krokowy
    MOV     AL,2
    OUT     05
    MOV     AL,4
    OUT     05
    MOV     AL,8
    OUT     05
    JMP     Rep

; -----
    ORG     50

    PUSH    al ; Zapisanie AL na stos.
    PUSH    bl ; Zapisanie BL na stos.
    PUSHF   ; Zapisanie flag na stos.

    JMP     PastData

    DB 84 ; Red      Green
    DB c8 ; Red+Amber Amber
    DB 30 ; Green    Red
    DB 58 ; Amber    Red+Amber
    DB 57 ; Wskazuje na początek tablicy
PastData:
    MOV     BL,[5B] ; BL wskazuje na tablice danych

```

```

MOV    AL,[BL]    ; Kopiowanie danych z pamieci pod adresem BL
OUT     01 ; Wyslanie danych do swiatel
CMP     AL,58 ; Sprawdzenie czy to ostatni wpis w tablicy
JZ Reset ; Jezeli ostatni, to resetuje wskaznik

INC     BL ; BL wskazuje do nastepnego elementu tablicy
MOV     [5B],BL ; Zapisanie wskaznika w pamieci
JMP     Stop

Reset:
MOV     BL,57 ; Ustawienie wskaznika na poczatek tablicy
MOV     [5B],BL ; Zapisanie wskaznika do pamieci pod adresem 5B
Stop:
POPF    ; Przywrocenie flag
POP     bl ; Przywrocenie BL
POP     al ; Przywrocenie AL

IRET    ;powrot z procedury obslugi przerwania
; -----
END

```

## 1.6. Dodatkowe informacje

1. [http://www.softwareforeducation.com/sms32v50/sms32v50\\_manual/210-11hwint.htm](http://www.softwareforeducation.com/sms32v50/sms32v50_manual/210-11hwint.htm)
2. <https://pl.wikipedia.org/wiki/Wielobieźność>

## 2. Zadania

### 2.1. Zadanie 1

Napisać program, który steruje grzałką w taki sposób, aby uzyskać stałą temperaturę. Jednocześnie program odlicza w sposób ciągły („przekręcenie licznika”) od 9 do 0 na wyświetlaczu VDU. Zmiana cyfry na wyświetlaczu ma być zrealizowana jako procedura obsługi przerwania zegara.

### 2.2. Zadanie 2

Napisać program wykorzystujący przerwania klawiatury do sterowania ruchem węża w labiryncie.

### 2.3. Zadanie 3

Napisać program obsługujący przerwania sprzętowe klawiatury znakowej i numerycznej, i wstawiający odczytane znaki na kolejną wolną pozycję wyświetlacza VDU. Po naciśnięciu przycisku „Enter” dodawanie rozpoczyna się od nowej linii (początku kolejnej linii), a naciśnięcie przycisku „Backspace”

powoduje usunięcie ostatnio wprowadzonego znaku lub przejście na koniec linii wyżej. Należy zabezpieczyć program przed wyjściem poza komórki pamięci przeznaczone dla wyświetlacza. Opcjonalnie można dodać wyświetlanie kursora.