

LABORATORIUM

PROGRAMOWANIA MIKROKONTROLERÓW

dr inż. Remigiusz Baran
Katedra Informatyki, Elektroniki i Elektrotechniki
PŚK, 2020



LABORATORIUM
PROGRAMOWANIA MIKROKONTROLERÓW

INSTRUKCJA NR: 3

TEMAT: WYKORZYSTANIE UKŁADU CZASOWO-LICZNIKOWEGO
I SYSTEM PRZERWAŃ.

Celem ćwiczenia jest zapoznanie z architekturą i wykorzystaniem układu czasowo-licznikowego mikrokontrolerów PIC16F74. W tym celu, procedura opóźniająca DziesięćMs zostanie zastąpiona przez pomiar czasu rzeczywistego zrealizowany w oparciu o jeden z liczników mikrokontrolera.

Mikrokontrolery PIC16F74 zawierają układ czasowo-licznikowy złożony z trzech modułów: Timer0, Timer1 i Timer2.

3.1. Moduł Timer0

Moduł Timer0 jest 8-bit licznikiem/czasomierzem (może zliczać impulsy wewnętrznego generatora o częstotliwości $f_{osc}/4$ i sygnału zewnętrznego podawanego na wejście RA4/T0CK1) wyposażonym w 8-bit programowy preskaler (układ wstępnego skalowania). Moduł Timer0 powiązany jest z rejestrem TMR0 (01h). Przepełnienie rejestru TMR0

(FFh \rightarrow 00h) ustawia bit T0IF (INTCON<2>) w rejestrze INTCON (0Bh, 8Bh), będący źródłem zgłoszenia przerwania z modułu Timer0. Przerwanie z modułu Timer0 jest maskowalne za pomocą bitu T0IE (INTCON<5>). Bit T0IF jest ustawiany sprzętowo, ale zerowany tylko programowo.

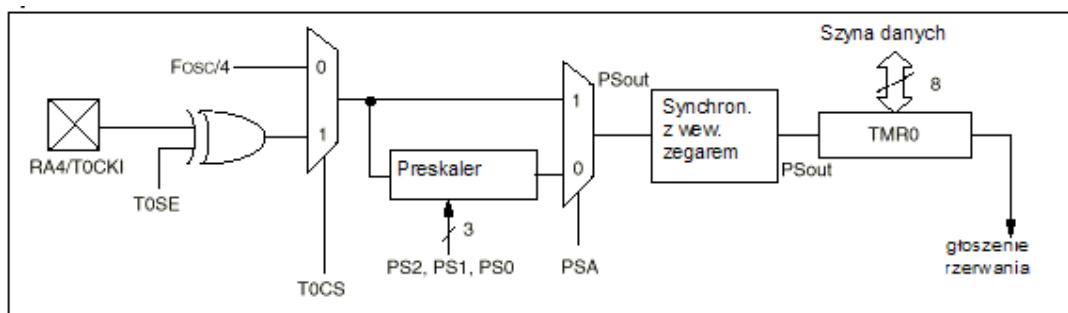
W zależności od trybu pracy Timer0 może zwiększać swoją zawartość w odpowiedzi na rosnące lub opadające zbocze zliczanego sygnału. Za funkcję pełnioną przez moduł Timer0 odpowiada rejestr OPTION_REG (81h), a dokładnie bit T0CS (OPTION_REG<5>). O rodzaju zbocza, na które następuje zwiększenie zawartości modułu decyduje bit T0SE (OPTION_REG<4>). Zawartość i szczegółowy opis bitów rejestru OPTION_REG pokazuje rys. 3.1.

OPTION_REG (81h)							
	RBP	INTEDG	T0CS	T0SE	PSA	PS2	PS0
	bit7						bit0
RBP	- bit uaktywnienia układu podnoszącego napięcie dla portu B: 1 – układ nieaktywny, 0 – układ aktywny						
INTEDG	- bit wyboru sposobu zgłoszenia przerwania na wejściu RB0/INT: 1 – zboczem narastającym, 0 – zboczem opadającym						
T0CS	- bit wyboru źródła zliczanego sygnału (funkcji): 1 – sygnał z wejścia RA4/T0CK1 (licznik), 0 – sygnał z wewn. zegara (czasomierz)						
T0SE	- bit wyboru zliczanego zbocza sygnału zewnętrznego z wejścia RA4/T0CK1: 1 – zbocze opadające, 0 – zbocze narastające						
PSA	- bit przypisania preskalera: 1 – WDT, 0 – Timer0						
PS2:PS0	- zakres preskalera:						
	bity		zakres		zakres		
	PS2:PS0		TMR0		WDT		
	000		1:2		1:1		
	001		1:4		1:2		
	010		1:8		1:4		
	011		1:16		1:8		
	100		1:32		1:16		
	101		1:64		1:32		
	110		1:128		1:64		
	111		1:256		1:128		

Rys. 3.1. Rejestr OPTION_REG

Jako czasomierz moduł Timer0 zwiększa swoją zawartość w każdym cyklu, chyba, że argumentem docelowym wykonywanej instrukcji jest rejestr TMR0. W takim przypadku kontynuowanie zliczania przez moduł Timer0 jest powstrzymywane na czas kolejnych dwóch cykli.

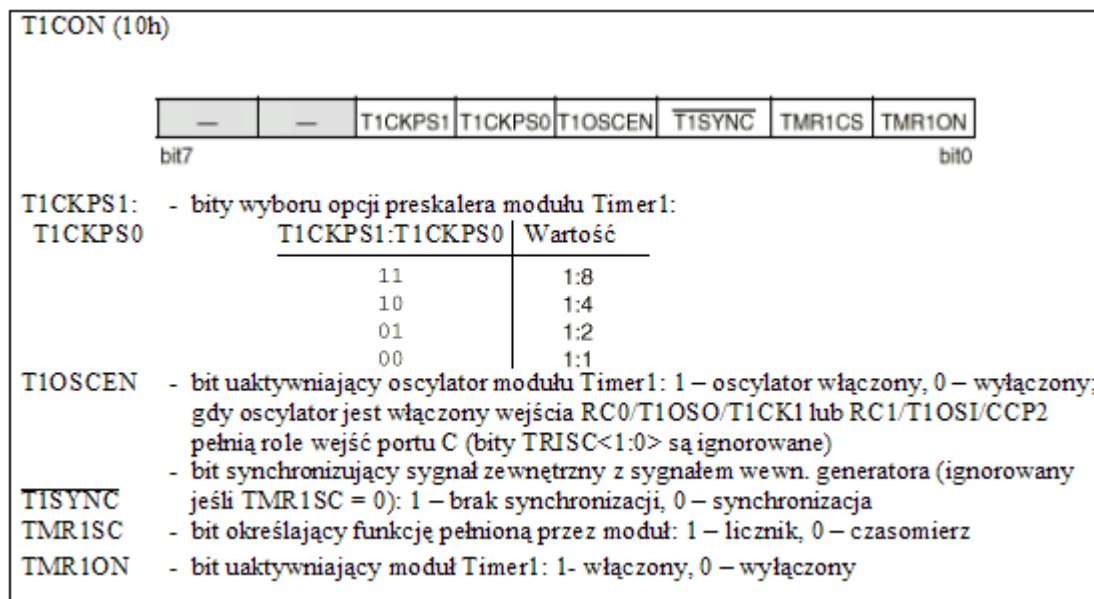
Moduł Timer0 wyposażony jest w opcję programowalnego preskalera. Preskaler może być przypisany albo do modułu Timer0 albo do układu WDT (Watchdog Timer). O przypisaniu preskalera decyduje bit PSA (OPTION_REG<3>). Bity PS2, PS1 i PS0 (OPTION_REG<2:0>) decydują o wartości preskalera. Jak wynika z rys. 2. Timer0 może zwiększać swoją zawartość w odpowiedzi na zmiany sygnału dołączonego poprzez następujące dzielniki częstotliwości: 1:1 (wtedy, gdy preskaler przypisany do układu WDT), 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, i 1:256 (tylko moduł Timer0). Schemat blokowy modułu Timer0 przedstawia rys. 3.2.



Rys. 3.2. Schemat blokowy modułu Timer0.

3.2. Moduł Timer1

Moduł Timer1 jest 16-bit licznikiem/czasomierzem (może zliczać impulsy wewnętrznego generatora o częstotliwości $f_{osc}/4$ i sygnału zewnętrznego podawanego na wejście RC0/T1OSO/T1CK1 lub RC1/T1OSI/CCP2 w zależności od stanu bitu T1OSCEN w rejestrze T1CON) wyposażonym w 2-bit programowy preskaler. Zawartość i szczegółowy opis bitów rejestru T1CON pokazuje rys. 3.3.

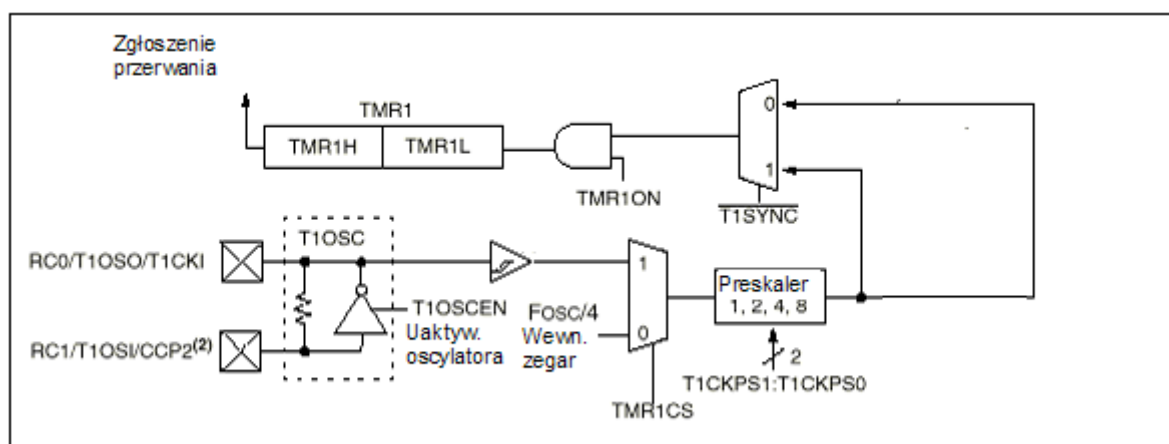


Rys. 3.3. Rejestr T1CON.

Moduł Timer1 składa się z dwóch rejestrów: TMR1H (0Fh) i TMR1L (0Eh). Przepelnienie rejestru TMR1, na który składa się para TMR1H:TMR1L (FFFFh -> 0000h) ustawia bit TMR1IF (PIR1<0>) w rejestrze PIR1 (0Ch), będący źródłem zgłoszenia przerwania z modułu Timer1. Przerwanie z modułu Timer1 jest maskowalne za pomocą bitu TMR1IE (PIE1<0>) w rejestrze PIE1 (8Ch). Za funkcję pełnioną przez moduł Timer1 odpowiada rejestr T1CON (10h), a dokładnie bit TMR1CS (T1CON<1>).

Jako czasomierz moduł Timer1 zwiększa swoją zawartość w każdym cyklu rozkazowym. Jako licznik zdarzeń moduł zwiększa swoją zawartość w odpowiedzi na każde narastające zbocze sygnału zewnętrznego. Moduł Timer1 może być uruchamiany (wstrzymywany) za pomocą bitu TMR1ON (T1CON<0>). Od stanu bitu T1SYNC zależy, czy sygnał zewnętrzny jest synchronizowany z sygnałem wewnętrznego generatora czy nie (tylko jeśli moduł pełni funkcję licznika).

Rys. 3.4. przedstawia schemat blokowy modułu Timer1.

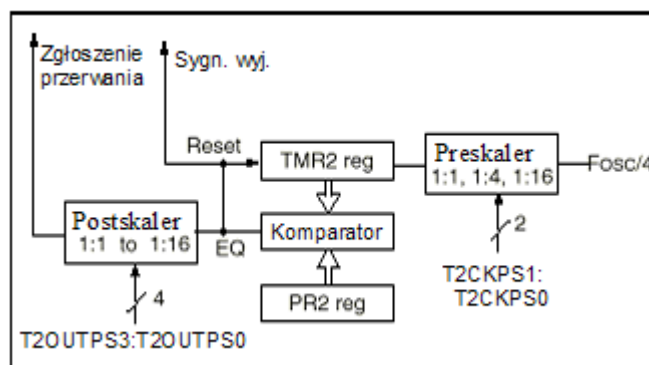


Rys. 3.4. Schemat blokowy modułu Timer1.

3.3. Moduł Timer2.

Moduł Timer2 jest 8-bit. czasomierzem, zliczającym impulsy z wewnętrznego generatora o częstotliwości $f_{osc}/4$ i wyposażonym w trzy układy skalujące: 2-bit. preskaler, 8-bit. rejestr PR2 (92h) i 4-bit postskaler (układ skalowania końcowego).

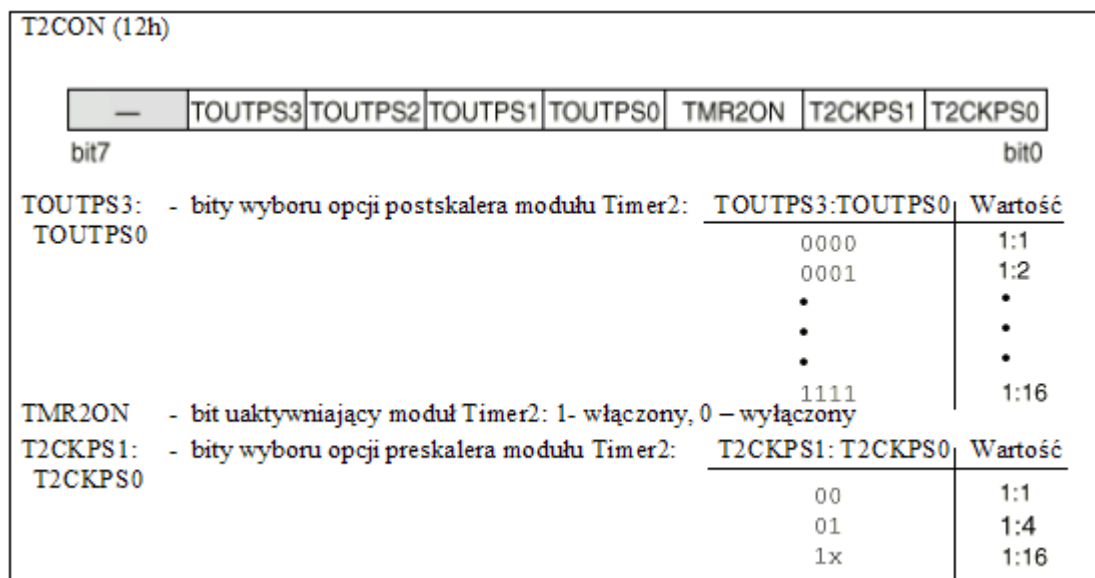
Rys. 3.5. przedstawia schemat blokowy modułu Timer2.



Rys. 3.5. Schemat blokowy modułu Timer2.

Obecność aż trzech układów skalujących daje duże możliwości, szczególnie przy generowaniu zadanych opóźnień. Z tego też względu moduł Timer2 może być wykorzystywany do kontrolowania okresu sygnału o zmodulowanej szerokości impulsu wtedy, gdy moduł CCP1 - jednego z dwóch modułów CCP (ang. Capture/Compare/PWM) pracuje w trybie PWM. Timer2 może być również używany do kontrolowania szybkości transmisji w układzie portu szeregowego SSP (ang. Synchronous Serial Port).

Zliczany sygnał wejściowy, o częstotliwości $f_{osc}/4$, podawany jest najpierw na wejście preskalera o trzech zakresach: 1:1, 1:4 i 1:16. Tak przeskalowany sygnał podawany jest następnie na wejście rejestru TMR2 (11h), z którym powiązany jest moduł Timer2. TMR2 zwiększa swoją zawartość od 00h aż do wartości zapamiętanej w rejestrze PR2 (period register – rejestr okresu), po czym ponownie zaczyna zliczać od 00h. PR2 jest rejestrem dostępnym programowo. Po starcie mikrokontrolera PR2 przyjmuje wartość 0FFh. Sygnał wyjściowy z rejestru TMR2 jest ponownie skalowany przy użyciu 4-bit. postskalera. Układ postskalera pozwala na zliczanie programowo zadaną liczbę razy okresów określonych w rejestrze PR2. Zmiana sygnału na wyjściu postskalera ustawia bit TMR2IF (PIR1<1>), co jest równoznaczne ze zgłoszeniem przerwania z modułu Timer2. Rejestrem kontrolnym modułu Timer2 jest rejestr T2CON (12h) pokazany na rys. 3.6.



Rys. 3.6. Rejestr T2CON.

Sygnał z wewnętrznego generatora o częstotliwości $f_{osc}/4 = 1 \text{ MHz}$ dzielony jest wewnątrz modułu Timer2 kolejno przez C, B i A, gdzie C - wartość preskalera, B – wartość rejestru PR2, a A – wartość postskalera. Należy pamiętać, że wartość wpisywana do rejestru PR2 musi być o 1 mniejsza od wartości celowej. Chcąc dla przykładu ustawić skaler PR2 na wartość 250 należy dokonać wpisu 249. I tak, jeżeli $A = 2$, $B = 250$, $C = 4$ i okres zliczanego sygnału wynosi $1 \mu\text{s}$, to Timer2 może być użyty do wygenerowania opóźnienia o wartości 2 ms ($2 \times 250 \times 4 = 2000 \mu\text{s}$).

3.4. System przerwań.

Zwiększenie zawartości postskalera jest równoznaczne z ustawieniem flagi TMR2IF w rejestrze PIR1. Aby mogło zostać wygenerowane przerwanie z czasomierza Timer2 muszą być ustawione następujące bity:

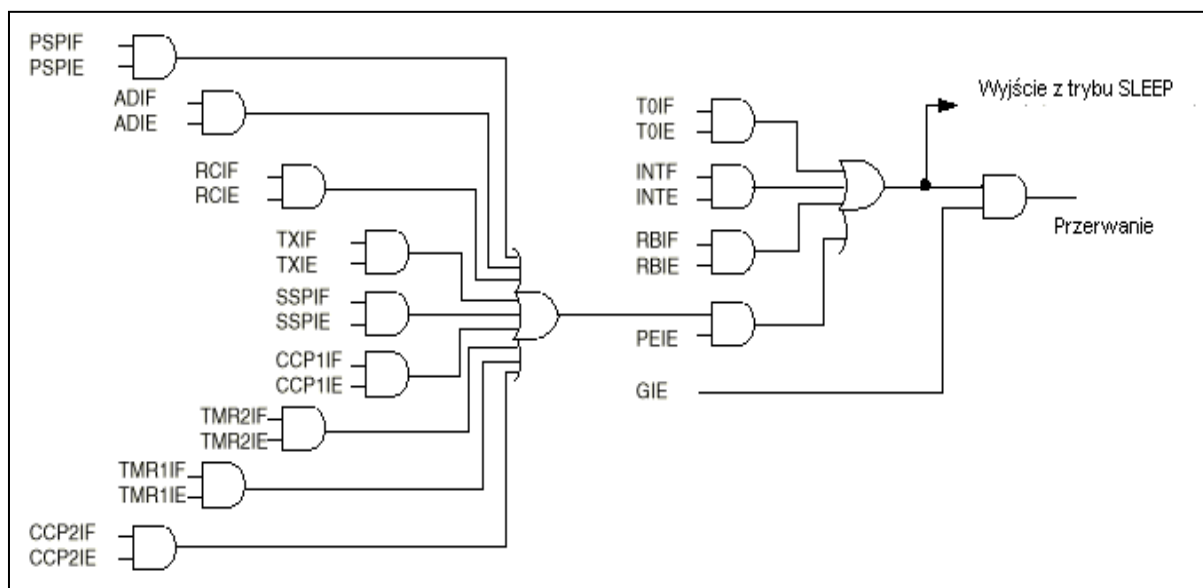
- TMR2IE (maska przerwania z modułu Timer2) w rejestrze PIE1 (8Ch);
- PEIE (maska przerwania zewnętrznych) oraz bit GIE (globalna maska przerwania) w rejestrze INTCON (0Bh, 8Bh).

Bit GIE może być ustawiany i zerowany programowo przy pomocy instrukcji bsf i bcf. Po przyjęciu przerwania bit GIE jest automatycznie, sprzętowo zerowany. GIE jest sprzętowo ustawiany po wykonaniu instrukcji retfie. Ze względu na to, że flaga TMR2IF nie jest zerowana sprzętowo, należy pamiętać o jej programowym wyzerowaniu wewnątrz procedury obsługi przerwania. Można to zrobić za pomocą instrukcji bcf PIR1,TMR2IF. Po restarcie mikrokontrolera bity GIE, PEIE i TMR2IE są zerowane. Należy zatem pamiętać o ich ustawieniu wewnątrz procedury Initial.

System przerwania mikrokontrolera PIC16C74B obsługuje 12 źródeł przerwania:

- z układu portu równoległego PSP: bity PSPIF (PIR1<7>), PSPIE (PIE1<7>);
- z przetwornika A/C: bity ADIF (PIR1<6>), ADIE (PIE1<6>);
- z odbiornika USART: bity RCIF (PIR1<5>), RCIE (PIE1<5>);
- z nadajnika USART: bity TXIF (PIR1<4>), TXIE (PIE1<4>);
- z portu szeregowego SSP: bity SSPIF (PIR1<3>), SSPIE (PIE1<3>);
- z modułu CCP1: bity CCP1IF (PIR1<2>), CCP1IE (PIE1<2>);
- z modułu Timer2: bity TMR2IF (PIR1<1>), TMR2IE (PIE1<1>);
- z modułu Timer1: bity TMR1IF (PIR1<0>), TMR1IE (PIE1<0>);
- z modułu CCP2: bity CCP2IF (PIR2<1>) (0Dh), CCP2IE (PIE2<0>) (8Dh);
- z modułu Timer0: bity T0IF (INTCON<2>), T0IE (INTCON<5>);
- z wejścia RB0/INT: bity INTF (INTCON<1>), INTE (INTCON<4>);
- z portu B (zmiana stanu portu): bity RBIF (INTCON<0>), RBIE (INTCON<3>).

Strukturę logiczną systemu przerwania mikrokontrolera PIC16F74 prezentuje rys. 3.7.



Rys. 3.7. Struktura logiczna systemu przerwania mikrokontrolera PIC16F74.

Przykładową postać procedury obsługi wielu różnych źródeł przerwań w architekturze mikrokontrolera PIC16F74 wraz z elementami tzw. prologu oraz epilogu, przedstawia Przykład 3.

Przykład 3.

```

; /////////////////////////////////// Procedura ProcObsPrzerwan ///////////////////////////////////
ProcObsPrzerwan
    movwf W_TEMP          ;zachowywanie rejestru roboczego
    swapf STATUS,W        ;oraz rejestru STATUS
    movwf STATUS_TEMP

    btfsc PIR1,TMR2IF      ;sprawdzenie czy źródłem przerwania jest
    call Timer2            ;Timer2, jeśli tak to wywołanie obsługi tego
    ...                   ;przerwania
;    btfsc ...             ;sprawdzenie innych źródeł przerwań
;    call ...
    ...
    swapf STATUS_TEMP,W    ;odtworzenie rejestrów W i STATUS
    movwf STATUS          ;wykorzystanie instrukcji swapf pozwala
    swapf W_TEMP,F        ;na zachowanie znacznika Z bez zmian
    swapf W_TEMP,W

    retfie                ;wyjście z procedury obsługi przerwania

```

3.5. Zadania dla studentów.

- Kontynuując prace rozwojowe projektu z poprzednich zajęć, dodać funkcję odmierzenia czasu rzeczywistego w oparciu o układ czasowo-licznikowy Timer0, skonfigurowany tak aby sygnał zgłoszenia przerwania z tego modułu następował co 2 ms (przy zał. $F_{osc} = 4$ MHz).
- Napisać procedurę obsługi przerwania z modułu Timer0 tak aby, aby w oparciu o dodatkowe zmienne, pełniące funkcje liczników, czas T_{pT0} odmierzany za pomocą tego modułu wynosił 1s.
- Zamienić dotychczasowy sposób odmierzenia czasu, przez który obecne są kolejne, ustalone za pomocą procedury TablicaPrzełącz stany na porcie D, procedurą obsługi przerwania z modułu Timer0.
- Zmodyfikować projekt tak aby kontrola zmian stanów na porcie D odbywała się za pomocą modułu Timer2, który zgłasza przerwania co 10 ms.
- Dodać do projektu, zmodyfikowanego wg. poprzedniego punktu, funkcję zerowania zmiennej INDEKS tak aby kolejne wywołania procedury TablicaPrzełącz następowały w powtarzającym się cyklu odczytu wszystkich kolejnych elementów tablicy, bez względu na długość tablicy (liczbę jej elementów).