

# **Rozkazy skoków i skoków warunkowych; przykłady sterowania przebiegiem programu**

1. Skoki bezwarunkowe: JMP, IJMP, RJMP
2. Skoki warunkowe:
  - a) rozkazy rozgałęzień BRBC, BRBS  
uwarunkowane stanem flagi w SREG
  - b) rozkazy „omiń” uwarunkowane stanem bitu w:
    - rejestrze roboczym SBRC, SBRS
    - urządzeniu wejściowo/wyjściowym SBIC, SBIS

3. Rozkazy porównujące dwa argumenty CP, CPI

4. Przykłady organizacji wybranych struktur algorytmicznych:

a) rozgałęzienie: if..then..else

b) pętle:

- ze sprawdzaniem warunku wyjścia z pętli na końcu: repeat..until
- ze sprawdzaniem warunku wyjścia z pętli na początku: while

c) rozgałęzienie wielowariantowe: switch

## Przykłady realizacji rozgałęzień *if...then...else*

```
pocz:      ldi      r16,wart_biezaca  
           cpi      r16,prog  
           brbs     0,procedura1    ;gdy (r16)<prog to procedura1
```

```
procedura2:nop                                ;gdy (r16)>=prog to procedura2  
           rjmp     dalej            ;omiń procedurę 1
```

;Uwaga:

;procedura2 może zajmować maksimum 63 komórki

```
procedura1:nop
```

```
dalej:      rjmp     pc
```

```

pocz:  ldi      r16,wart_biezaca
        cpi      r16,prog
        in      r17,sreg
        sbrc    r17,0
        rjmp    procedura1      ;gdy (r16)<prog to procedura1
procedura2:nop                    ;gdy (r16)>=prog to procedura2
        rjmp    dalej            ;omiń procedurę 1

.org $800    ;obecnie procedura2 może mieć do 2047 komórek

procedura1:nop

dalej:      rjmp    pc

```

## Przykład rozgałęzienia na trzy procedury:

[illegible]

## **Przykłady pętli *repeat* z różnymi warunkami wyjścia z pętli sprawdzanymi na jej końcu:**

- ;Kolejne komórki pamięci (min. jedną)
- ;począwszy od tej, której adres jest w X
- ;wypełnić kolejnymi liczbami o tej samej parzystości
- ;co liczba początkowa pozostawiona w r16

;Cztery wersje wyjścia z pętli

```
.equ    liczba_pocz=0x02  
.equ    adres_pocz=0x2fe  
.equ    liczba_komorek=7
```

```
pocz:  ldi  r16,liczba_pocz
        ldi  x1,low(adres_pocz)
        ldi  xh,high(adres_pocz)
        ldi  r17,liczba_komorek      ;w wersji 3 i 4 niepotrzebny
```

```
wpisz:  st   x+,r16
        inc  r16      ;liczby o tej samej parzystości...
        inc  r16      ;...są oddalone od siebie o 2
```

;wyjście z pętli wg liczby komórek do wpisania

```
        dec   r17
wersja1: brbc   1,wpisz
```

;jak w wersji 1, ale gdyby pętla była zbyt długa, tj. >63 komórki

```
wersja2:    dec     r17  
            in      r18,sreg  
            sbrs    r18,1  
            rjmp    wpisz
```

;wyjście z pętli wg ostatniej wartości do wpisania

```
.equ        ostatnia=0x4f  
wersja3:    cpi     r16,ostatnia+2  
            brbc    1,wpisz
```



wyjście z pętli wg ostatniego adresu pod który należy wpisać

.equ        adres\_konc=0x300

wersja4:    cpi        xl,low(adres\_konc+1) ;porównaj adresy

          brbc        1,wpisz

          cpi        xh,high(adres\_konc+1)

          brbc        1,wpisz

koniec:     rjmp      pc

## **Przykład pętli *while* ze sprawdzaniem warunku wyjścia na początku:**

- ;Kolejne komórki pamięci (być może żadnej)
- ;począwszy od tej, której adres jest w X
- ;wypełnić kolejnymi liczbami o tej samej parzystości
- ;co liczba początkowa pozostawiona w r16.
- ;Liczba komórek do wypełnienia - w r17

```
.equ    liczba_pocz=$02  
.equ    liczba_komorek=3  
.equ    adres_pocz=0x200
```

```
pocz:  ldi r16,liczba_pocz    ;załaduj wartości początkowe
      ldi r17,liczba_komerek
      ldi x1,low(adres_pocz)
      ldi xh,high(adres_pocz)
```

```
      tst      r17            ;ustaw flagę Z
wpisz: brbs    1,koniec       ;zakończ jeśli licznik wyzerowany
      st      x+,r16
      inc     r16             ;liczby o tej samej parzystości...
      inc     r16             ;... są od siebie oddalone o 2
      dec     r17             ;czy wpisano już wszystkie?
      rjmp    wpisz          ;rozkaz dec ustawia flagę Z
```

```
koniec: rjmp   pc
```

## Przykład procedury wielowariantowego wyboru *switch*

```
.equ    wartosc_biezaca=0x5  
.equ    wartosc1=0x8  
.equ    wartosc2=0x37  
.equ    wartosc3=0x13
```

```
pocz:   ldi    r16,wartosc_biezaca    ;obliczona wartość wyrażenia
```

```
switch: cpi     r16,wartosc1  
        brbc    1,dalej1             ;jeśli nie ta, to sprawdzaj dalej  
        nop                                           ;procedura1  
        rjmp    koniec                ;omiń pozostałe warianty, tj. break
```

dalej1:	cpi	r16,wartosc2	
	brbc	1,dalej2	;jeśli nie ta, to sprawdzaj dalej
	nop		;procedura2
	rjmp	koniec	;omiń pozostałe warianty, tj. break

dalej2:	cpi	r16,wartosc3	
	brbc	1,default	;jeśli nie ta to wykonaj domyślną
	nop		;procedura3
	rjmp	koniec	;omiń procedurę domyślną, tj. break

default:	nop		;procedura domyślna
----------	-----	--	---------------------

koniec:	rjmp	pc	
---------	------	----	--

