



DK – STEP 7 – Posredno naslavljanje

◆ Pomnilniško posredno naslavljanje (16-bitni kazalec)

```
MW 20:  2#0000_0000_0010_1001 = 16#0029 = 41
```

```
...  
CD C[MW 20] // štetje navzdol števca, katerega naslov...  
           // ...je definiran z vsebino MW 20  
...
```

```
MW 40:  2#0000_0000_0000_1010 = 16#000A = 10
```

```
A I 0.1  
SE T[MW 40] // podaljšanje pulza na časovniku z naslovom,  
...       // ... definiranim z vsebino MW 40
```

```
MW 8:   2#0000_0000_0001_0000 = 16#0010 = 16
```

```
A M 14.1  
CC FB[MW 8] // klic funkcijskega bloka z naslovom,  
...       // ... definiranim z vsebino MW 8
```



DK – STEP 7 – Posredno naslavljanje

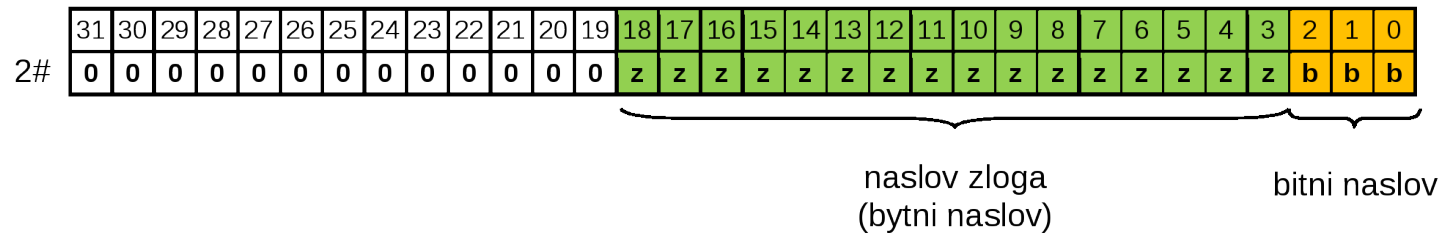
◆ Pomnilniško posredno naslavljanje bitov (32-bitni kazalec)

I 1.6

Q 124.2

M 16.0

P# kazalčni format



P#1.6 = 16#0000_0000_0000_000E = 2#0000_0000_0000_0000_0000_0000_0000_1110

P#124.2 = 16#0000_0000_0000_03E2 = 2#0000_0000_0000_0000_0000_0011_1110_0010

P#16.0 = 16#0000_0000_0000_0080 = 2#0000_0000_0000_0000_0000_0000_1000_0000

I [MD 40]

Q [MD 44]

M [MD 48]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	z	z	z	z	z	z	z	z	z	z	z	z	z	z	z	z	z	b	b	b

naslov zloga
(bytni naslov)

bitni naslov

DK – STEP 7 – Posredno naslavljanje

◆ Pomnilniško posredno naslavljanje bitov (32-bitni kazalec)

```

A   I[MD 16]      // povpraševanje po prvem vhodu           A   I 0.3
A   I[MD 20]     // povpraševanje po drugem vhodu          A   I 1.7
=   Q[MD 24]     // rezultat na izhod                       =   Q 4.4

```

```

MD 16:   P#0.3    = 16#0000_0000_0000_0003 = 2#0000_0000_0000_0000_0000_0000_0000_0011
MD 20:   P#1.7    = 16#0000_0000_0000_000F = 2#0000_0000_0000_0000_0000_0000_0000_1111
MD 24:   P#4.4    = 16#0000_0000_0000_0024 = 2#0000_0000_0000_0000_0000_0000_0010_0100

```

```

L   P#0.3      // priprava kazalca za prvi vhod
T   MD 16
L   P#1.7      // priprava kazalca za drugi vhod
T   MD 20
L   P#4.4      // priprava kazalca za izhod
T   MD 24
A   I[MD 16]    // povpraševanje po prvem vhodu
A   I[MD 20]    // povpraševanje po drugem vhodu
=   Q[MD 24]    // rezultat na izhod

```

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	z	z	z	z	z	z	z	z	z	z	z	z	z	z	z	z	z	b	b	b

naslov zloga
(bytni naslov)

bitni naslov

DK – STEP 7 – Posredno naslavljanje

◆ Pomnilniško posredno naslavljanje B, W, D (32-bitni kazalec)

Naloga zahteva, da se ob pritisku na tipko I 0.0 na vse zloge pomožnega pomnilnika od MB 0 do vključno MB 10 zapiše vsebina, ki se nahaja na vhodnem zlogu IB 1.

```

A      I 0.0      // povprašaj po tipki na vhodu I 0.0
FP     M 40.0     // dinamični člen; POZOR: prir. tabela!
JC     SKOK      // če je pritisk, skoči na akcijo
BEU                                         // sicer končaj

SKOK:  L   P#0.0  // začetni kazalec za MB 0
        T   MD 20

        L   +11    // število ponovitev zanke
ZANK:  T   MW 30   // shrani (novo) stanje števca ponovitev

        L   IB 1   // naloži vsebino vhodnega zloga IB 0
        T   MB[MD 20] // in jo prenesi na trenutno izbrani MB

        L   MD 20  // naloži kazalec...
INC    8          // ga ustrezno povečaj !!!
        T   MD 20  // in pripravi za naslednji

        L   MW 30  // naloži števec ponovitev
LOOP   ZANK      // zmanjšaj št. ponovitev in ponovi
BE

```



DK – STEP 7 – Posredno naslavljanje

Registrsko posredno naslavljanje

Uporaba 32-bitnih naslovnih registrov **AR1** in **AR2**

$$= Q [AR1, P\#1.1]$$

identifikator naslova Q je podan v ukazu (**registrsko posredno naslavljanje znotraj področja**)

AR1: P#8.7 = 16#0000_0000_0000_0047 = 2#0000_0000_0000_0000_0000_0000_0100_0111

Offset: P#1.1 = 16#0000_0000_0000_0009 = 2#0000_0000_0000_0000_0000_0000_0000_1001

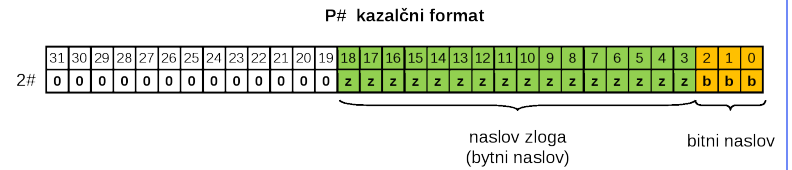
Naslov: P#10.0 = 16#0000_0000_0000_0050 = 2#0000_0000_0000_0000_0000_0000_0101_0000

naslov v AR1 oz. AR2 se po ukazu **NE** spremeni

A I[AR1, P#4.3]

L IB[AR1, P#100.0]

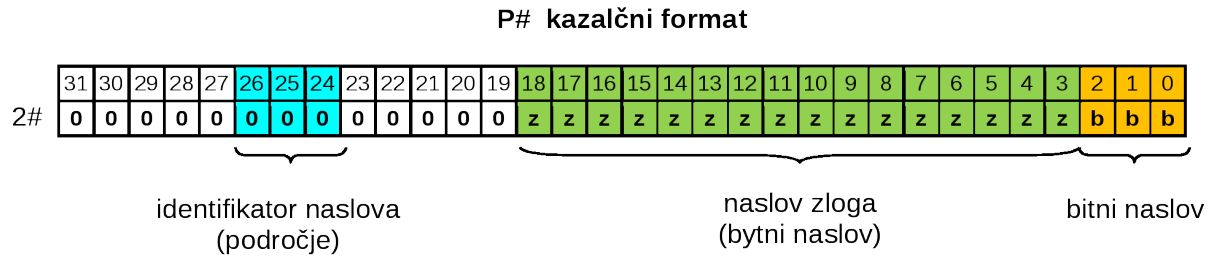
T MD[AR2, P#56.0]





DK – STEP 7 – Posredno naslavljanje

Registrsko posredno naslavljanje preko področja



= [AR1, P#1.1]

identifikator naslova Q NI podan v ukazu, temveč je že vključen v kazalcu v ukazu je določena samo podatkovna širina operanda

L B[AR1, P#100.0]

T D[AR2, P#56.0]

L P#I8.7

LAR1

LAR2 P#Q8.7

A [AR1, P#0.0]

= [AR2, P#1.1]

A I 8.7

= Q 10.0



DK – STEP 7 – Posredno naslavljanje

◆ Ukazi pri delu z naslovnima registroma AR1 in AR2

LAR1

LAR1 P#z.b

LAR1 P#Pz.b

LAR1 MDz

LAR2

LAR2 P#z.b

LAR2 P#Pz.b

LAR2 MDz

TAR1

TAR1 MDz

TAR2

TAR2 MDz

CAR // vzajemno zamenja vsebini obeh AR

+AR1 //k vsebini AR1 prišteje vsebino AKU1

+AR1 P#z.b //k vsebini AR1 prišteje konkretni kazalec

+AR2 //k vsebini AR2 prišteje vsebino AKU1

+AR2 P#z.b //k vsebini AR2 prišteje konkretni kazalec