

# POPIS INŠTRUKCIÍ

Inštrukčný súbor CPU emulátora obsahuje 53 inštrukcií. Tieto inštrukcie môžeme rozdeliť do piatich základných skupín. Sú to:

- aritmetické a logické inštrukcie
- inštrukcie pre posun a rotáciu
- inštrukcie prenosu dát
- inštrukcie vetvenia
- špeciálne inštrukcie

## 1. ARITMETICKÉ A LOGICKÉ INŠTRUKCIE

### ADD - Sčítanie bez Carry

*Popis:* Sčítanie obsahu dvoch registrov a umiestenie výsledku do registra Rd.

*Operácia:*

$$Rd \leftarrow Rd + Rs$$

*Syntax:*

ADD Rd, Rs

*Parametre:*

Rd, Rs – A, B, C, D

*Čítač inštrukcií:*

$$PC \leftarrow PC + 1$$

*Nastavenie príznakov:* Z, CY

### ADC - Sčítanie s Carry

*Popis:* Sčítanie obsahu dvoch registrov a obsahu príznakového bitu Carry a umiestenie výsledku do registra Rd.

*Operácia:*

$$Rd \leftarrow Rd + Rs + CY$$

*Syntax:*

ADC Rd, Rs

*Parametre:*

Rd, Rs – A, B, C, D

*Čítač inštrukcií:*

$$PC \leftarrow PC + 1$$

*Nastavenie príznakov:* Z, CY

### ADI – Pripočítanie konštanty

*Popis:* Pripočítanie konštanty k obsahu registra a uloženie výsledku do registra.

*Operácia:*

$$Rd \leftarrow Rd + K$$

*Syntax:*

ADI Rd, K

*Parametre:*

Rd – A, B, C, D;  $0 \leq K \leq 255$

*Čítač inštrukcií:*

$$PC \leftarrow PC + 1$$

*Nastavenie príznakov:* Z, CY

### SUB – Odčítanie bez Carry

*Popis:* Odčítanie obsahov dvoch registrov a uloženie výsledku do registra Rd.

*Operácia:*

$$Rd \leftarrow Rd - Rs$$

*Syntax:*

SUB Rd, Rs

*Parametre:*

Rd, Rs – A, B, C, D

*Čítač inštrukcií:*

$$PC \leftarrow PC + 1$$

*Nastavenie príznačov: Z,CY*

### **SUC– Odčítanie s Carry**

*Popis:* Odčítanie obsahu registra Rs a Carry od obsahu registra Rd a uloženie výsledku do registra Rd.

*Operácia:*

$$Rd \leftarrow Rd - Rs - CY$$

*Syntax:*

SBC Rd, Rs

*Parametre:*

Rd, Rs – A, B, C, D

*Čítač inštrukcií:*

$$PC \leftarrow PC + 1$$

*Nastavenie príznačov: Z, CY*

### **SBI – Odčítanie konštanty**

*Popis:* Odčítanie konštanty od obsahu registra a uloženie výsledku do registra.

*Operácia:*

$$Rd \leftarrow Rd - K$$

*Syntax:*

SBI Rd, K

*Parametre:*

Rd – A, B, C, D;  $0 \leq K \leq 255$

*Čítač inštrukcií:*

$$PC \leftarrow PC + 1$$

*Nastavenie príznačov: Z, CY*

### **AND - Logický AND**

*Popis:* Operácia vykoná logický AND medzi registrom Rd a registrom Rs z následným uložením výsledku do registra Rd.

*Operácia:*

$$Rd \leftarrow Rd \bullet Rs$$

*Syntax:*

AND Rd, Rs

*Parametre:*

Rd, Rs – A, B, C, D

*Čítač inštrukcií:*

$$PC \leftarrow PC + 1$$

*Nastavenie príznačov: Z*

### **ANI - Logický AND s konštantou**

*Popis:* Vykoná logický AND medzi obsahom registra Rd a konštantou z následným uložením výsledku do registra Rd.

*Operácia:*

$$Rd \leftarrow Rd \bullet K$$

*Syntax:*

ANI Rd, K

*Parametre:*

Rd – A, B, C, D;  $0 \leq K \leq 255$

*Čítač inštrukcií:*

$$PC \leftarrow PC + 1$$

*Nastavenie príznačov: Z*

### **ORR – Logický OR**

*Popis:* Vykoná logický OR medzi obsahom registra Rd a obsahom registra Rs z následným uložením výsledku do registra Rd.

*Operácia:*

$$Rd \leftarrow Rd \vee Rs$$

*Syntax:*

ORR Rd, Rs

*Parametre:*

Rd, Rs – A, B, C, D

*Čítač inštrukcií:*

$$PC \leftarrow PC + 1$$

*Nastavenie príznačov: Z*

## **ORI - Logický OR s konštantou**

*Popis:* Vykoná logický OR medzi obsahom registra Rd a konštantou z následným uložením výsledku do registra Rd.

*Operácia:*

$Rd \leftarrow Rd \vee K$

*Syntax:*

ORI Rd,K

*Parametre:*

Rd – A,B,C,D;  $0 \leq K \leq 255$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

*Nastavenie príznačkov:* Z

## **XOR – Exkluzívny OR**

*Popis:* Vykoná logickú operáciu XOR medzi obsahom registra Rd a obsahom registra Rs a uloží výsledok do registra Rd.

*Operácia:*

$Rd \leftarrow Rd \oplus Rs$

*Syntax:*

XOR Rd,Rs

*Parametre:*

Rd,Rs – A,B,C,D

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

*Nastavenie príznačkov:* Z

## **XRI - Exkluzívny OR s konštantou**

*Popis:* Vykoná logickú operáciu XOR medzi obsahom registra Rd a konštantou z následným uložením výsledku do registra Rd.

*Operácia:*

$Rd \leftarrow Rd \oplus K$

*Syntax:*

XRI Rd,K

*Parametre:*

Rd – A,B,C,D;  $0 \leq K \leq 255$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

*Nastavenie príznačkov:* Z

## **INC – Inkrement**

*Popis:* Pripočíta jednotku k obsahu registra a výsledok umiestni do tohto registra.

*Operácia:*

$Rd \leftarrow Rd + 1$

*Syntax:*

INC Rd

*Parametre:*

Rd – A,B,C,D

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

*Nastavenie príznačkov:* Z,CY

## **INX – Inkrement 16-bitového registra**

*Popis:* Pripočíta jednotku k obsahu registra a výsledok umiestni do tohto registra.

*Operácia:*

$Rd \leftarrow Rd + 1$

*Syntax:*

INX Rd

*Parametre:*

Rd – S,M

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

*Nastavenie príznačkov:* Z,CY

## **DEC – Dekrement**

*Popis:* Odpočíta jednotku od obsahu registra a výsledok umiestni do tohto registra.

*Operácia:*

$Rd \leftarrow Rd - 1$

*Syntax:*

DEC Rd

*Parametre:*

Rd – A,B,C,D

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

*Nastavenie príznakov:* Z,CY

## **DCX – Dekrement 16-bitového registra**

*Popis:* Odpočíta jednotku od obsahu registra a výsledok umiestni do tohto registra.

*Operácia:*

$Rd \leftarrow Rd - 1$

*Syntax:*

DCX Rd

*Parametre:*

Rd – S,M

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

*Nastavenie príznakov:* Z,CY

## **CMP – Porovnanie obsahu dvoch registrov**

*Popis:* Nedeštruktívne odčítanie obsahov dvoch registrov a nastavenie príznakov.

*Operácia:*

$Rd - Rs$

*Syntax:*

CMP Rd, Rs

*Parametre:*

Rd, Rs – A,B,C,D

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

*Nastavenie príznakov:* Z,CY

## **CMI – Porovnanie obsahu registra a konštanty**

*Popis:* Nedeštruktívne odčítanie konštanty od obsahu registra a nastavenie príznakov.

*Operácia:*

$Rd - K$

*Syntax:*

CMI Rd, K

*Parametre:*

Rd – A,B,C,D;  $0 \leq K \leq 255$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

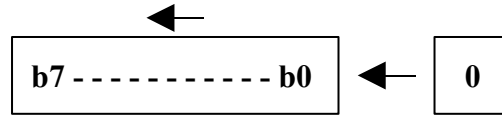
*Nastavenie príznakov:* Z,CY

## 2. INŠTRUKCIE POSUNU A ROTÁCIE

### SHL – logický posun vľavo

*Popis:* Posunie všetky bity registra o daný počet miest doľava. Do  $n$  spodných bitov uloží nulu.

*Operácia:*



*Syntax:*

SHL Rd,n

*Parametre:*

Rd – A,B,C,D;  $1 \leq n \leq 8$

*Čítač inštrukcií:*

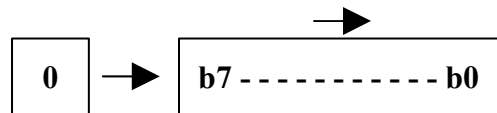
PC  $\leftarrow$  PC + 1

*Nastavenie príznakov:* Z

### SHR – logický posun vpravo

*Popis:* Posunie všetky bity registra o daný počet miest doprava. Do  $n$  horných bitov uloží nulu.

*Operácia:*



*Syntax:*

SHR Rd,n

*Parametre:*

Rd – A,B,C,D;  $1 \leq n \leq 8$

*Čítač inštrukcií:*

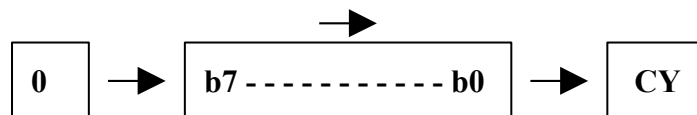
PC  $\leftarrow$  PC + 1

*Nastavenie príznakov:* Z

### SCR – logický posun vpravo s Carry

*Popis:* Posunie všetky bity registra o daný počet miest doprava. Do  $n$  horných bitov uloží nulu. Nultý bit sa pri každom posune uloží do Carry.

*Operácia:*



*Syntax:*

SCR Rd,n

*Parametre:*

Rd – A,B,C,D;  $1 \leq n \leq 8$

*Čítač inštrukcií:*

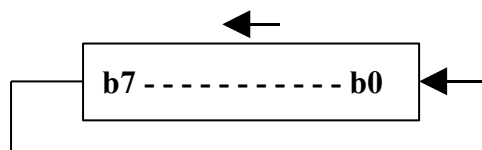
PC  $\leftarrow$  PC + 1

*Nastavenie príznakov:* Z,CY

### RTL – rotácia vľavo

*Popis:* Posunie všetky bity v registri Rd doľava o  $n$  miest. Pri každom posune uloží do nultého bitu obsah siedmeho bitu.

*Operácia:*



*Syntax:*

RTL Rd,n

*Parametre:*

Rd – A,B,C,D;  $1 \leq n \leq 8$

*Čítač inštrukcií:*

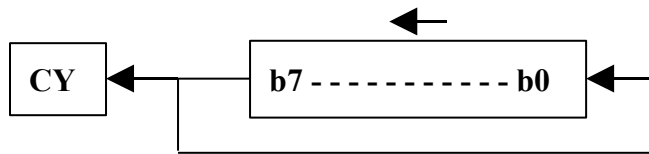
PC  $\leftarrow$  PC + 1

*Nastavenie príznakov:* Z

## RCL – rotácia vľavo s Carry

*Popis:* Posunie všetky bity v registri Rd doľava o  $n$  miest. Pri každom posune uloží do nultého bitu obsah siedmeho bitu. Siedmy bit sa pri každom posune ukladá do Carry.

*Operácia:*



*Syntax:*

RCL Rd,n

*Parametre:*

Rd – A,B,C,D;  $1 \leq n \leq 8$

*Čítač inštrukcií:*

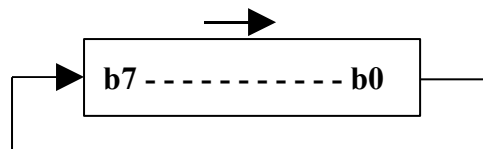
PC  $\leftarrow$  PC + 1

*Nastavenie príznačkov:* Z,CY

## RTR – rotácia vpravo

*Popis:* Posunie všetky bity v registri Rd doprava o  $n$  miest. Pri každom posune uloží do siedmeho bitu obsah nultého bitu.

*Operácia:*



*Syntax:*

RTR Rd,n

*Parametre:*

Rd – A,B,C,D;  $1 \leq n \leq 8$

*Čítač inštrukcií:*

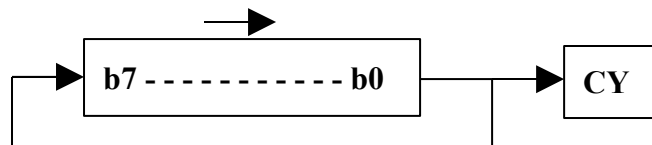
PC  $\leftarrow$  PC + 1

*Nastavenie príznačkov:* Z

## RCR – rotácia vpravo s Carry

*Popis:* Posunie všetky bity v registri Rd doprava o  $n$  miest. Pri každom posune uloží do siedmeho bitu obsah nultého bitu a do Carry sa uloží nultý bit.

*Operácia:*



*Syntax:*

RCR Rd,n

*Parametre:*

Rd – A,B,C,D;  $1 \leq n \leq 8$

*Čítač inštrukcií:*

PC  $\leftarrow$  PC + 1

*Nastavenie príznačkov:* Z,CY

### 3. INŠTRUKCIE PRENOSU DÁT

#### MOV – Kopírovanie obsahu registra

*Popis:* Tato inštrukcia zapíše do registra Rd obsah registra Rs

*Operácia:*

$Rd \leftarrow Rs$

*Syntax:*

MOV Rd, Rs

*Parametre:*

Rd, Rs – A, B, C, D

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

#### MVI – Načítanie konštanty do registra

*Popis:* Načítanie 8-bitovej konštanty do registra

*Operácia:*

$Rd \leftarrow K$

*Syntax:*

MVI Rd, K

*Parametre:*

Rd – A, B, C, D;  $0 \leq K \leq 255$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

#### MXI – Načítanie konštanty do registra

*Popis:* Načítanie 16-bitovej konštanty do registra

*Operácia:*

$Rd \leftarrow K$

*Syntax:*

MXI Rd, K

*Parametre:*

Rd – S, M;  $0 \leq K \leq 65535$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

#### MVX – Kopírovanie dvoch 8-bitových registrov do 16-bitového, alebo naopak

*Popis:* Tato inštrukcia zapíše do 16-bitového registra obsah dvoch 8-bitových registrov, alebo naopak.

*Syntax:*

MVX S, A

MVX M, A

MVX C, S

MVX C, M

*Operácia:*

$SP \leftarrow BA$

$MP \leftarrow BA$

$CD \leftarrow SP$

$CD \leftarrow MP$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

#### MMR – Načítanie konštanty z pamäti programu

*Popis:* Načítanie konštanty z pamäti programu do registra Rd. V registri Rs je uložený index miesta v pamäti. Definovanie konštanty v pamäti programu sa uskutočňuje inštrukciou BYTE.

*Operácia:*

$Rd \leftarrow [Rs]$

*Syntax:*

MMR Rd, Rs

*Parametre:*

Rd, Rs – A, B, C, D

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

#### LMI – Načítanie pamäte do registra s použitím adresy

*Popis:* Načíta 1 byte do registra z miesta v pamäti, ktoré je dané adresou.

*Operácia:*

$Rd \leftarrow [adr]$

*Syntax:*

LMI Rd, adr

*Parametre:*

Rd – A, B, C, D;  $0 \leq adr \leq 65535$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

## LMR – Nepriame načítanie do registra s použitím smerníka MP

*Popis:* Načíta 1 byte do registra z miesta v pamäti, na ktoré ukazuje smerník MP.

*Operácia:*

$Rd \leftarrow [MP]$

*Syntax:*

LMR Rd

*Parametre:*

$Rd - A, B, C, D$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

## SMI – Uloženie obsahu registra do pamäte s použitím adresy

*Popis:* Uloží obsah registra do miesta v pamäti, ktoré je dané adresou.

*Operácia:*

$[adr] \leftarrow Rs$

*Syntax:*

SMI adr, Rs

*Parametre:*

$Rs - A, B, C, D; 0 \leq adr \leq 65535$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

## SMR – Nepriame uloženie obsahu registra do pamäte s použitím smerníka MP

*Popis:* Uloží obsah registra do miesta v pamäti, na ktoré ukazuje smerník MP.

*Operácia:*

$[MP] \leftarrow Rs$

*Syntax:*

SMR Rs

*Parametre:*

$Rs - A, B, C, D$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

## INN – Čítanie z I/O registra

*Popis:* Načíta dáta z I/O registra daného adresou do registra Rd.

*Operácia:*

$Rd \leftarrow [adr]$

*Syntax:*

INN Rd, adr

*Parametre:*

$Rd - A, B, C, D; 0 \leq adr \leq 65535$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

## OUT – Zapísanie do I/O registra

*Popis:* Zapíše obsah registra Rs do I/O registra daného adresou.

*Operácia:*

$[adr] \leftarrow Rs$

*Syntax:*

OUT adr, Rs

*Parametre:*

$Rs - A, B, C, D; 0 \leq adr \leq 65535$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

## PUS – Uloženie obsahu registra do zásobníka

*Popis:* Zapíše obsah registra Rs do zásobníka.

*Operácia:*

$[SP] \leftarrow Rs$

*Syntax:*

PUS Rs

*Parametre:*

$Rs - A, B, C, D, F$

$Rs - M$

*Zásobník:*

$SP \leftarrow SP - 1$

$SP \leftarrow SP - 2$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$



## **POP – Načítanie obsahu registra zo zásobníka**

*Popis:* Načíta obsah registra Rd zo zásobníka.

*Operácia:*

$Rd \leftarrow [SP]$

*Syntax:*

POP Rd

*Parametre:*

Rd – A,B,C,D,F

Rd – M

*Zásobník:*

$SP \leftarrow SP + 1$

$SP \leftarrow SP + 2$

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

## **STR – Zápis obsahu registra do špeciálnej pamäte**

*Popis:* Zapíše obsah registra Rs na adresu špeciálnej 256 bajtovej pamäte, ktorá je uložená v registri Rd.

*Operácia:*

$[Rd] \leftarrow Rs$

*Syntax:*

STR Rd, Rs

*Parametre:*

Rd – A,B,C,D

*Zásobník:*

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

## **LDR – Čítanie obsahu adresy špeciálnej pamäte**

*Popis:* Do registra Rd zapíše obsah adresy špeciálnej 256 bajtovej pamäte, ktorá je uložená v registri Rs.

*Operácia:*

$Rd \leftarrow [Rs]$

*Syntax:*

LDR Rd, Rs

*Parametre:*

Rd – A,B,C,D

*Zásobník:*

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

## 4. Inštrukcie vetvenia

### JMP – Bezpodmienečný skok

*Popis:* Vykoná skok na dané návěstie v programe.

*Operácia:*

$PC \leftarrow [nav]$

*Syntax:*

JMP    nav

*Čítač inštrukcií:*

$PC \leftarrow [nav]$

### JZR – Skok ak Zero = 1

*Popis:* Táto inštrukcia vykoná skok na dané návěstie, ak je príznak Zero nastavený.

*Operácia:*

if  $Z = 1$  then  $PC \leftarrow [nav]$  else  $PC \leftarrow PC + 1$

*Syntax:*

JZR    nav

*Čítač inštrukcií:*

$PC \leftarrow [nav]$     ak  $Z = 1$

$PC \leftarrow PC + 1$     ak  $Z = 0$

### JNZ – Skok ak Zero = 0

*Popis:* Táto inštrukcia vykoná skok na dané návěstie, ak je príznak Zero nulový.

*Operácia:*

if  $Z = 0$  then  $PC \leftarrow [nav]$  else  $PC \leftarrow PC + 1$

*Syntax:*

JNZ    nav

*Čítač inštrukcií:*

$PC \leftarrow [nav]$     ak  $Z = 0$

$PC \leftarrow PC + 1$     ak  $Z = 1$

### JCY – Skok ak Carry = 1

*Popis:* Táto inštrukcia vykoná skok na dané návěstie, ak je príznak Carry nastavený.

*Operácia:*

if  $CY = 1$  then  $PC \leftarrow [nav]$  else  $PC \leftarrow PC + 1$

*Syntax:*

JCY    nav

*Čítač inštrukcií:*

$PC \leftarrow [nav]$     ak  $CY = 1$

$PC \leftarrow PC + 1$     ak  $CY = 0$

### JNC – Skok ak Carry = 0

*Popis:* Táto inštrukcia vykoná skok na dané návěstie, ak je príznak Carry nulový.

*Operácia:*

if  $CY = 0$  then  $PC \leftarrow [nav]$  else  $PC \leftarrow PC + 1$

*Syntax:*

JNC    nav

*Čítač inštrukcií:*

$PC \leftarrow [nav]$     ak  $CY = 0$

$PC \leftarrow PC + 1$     ak  $CY = 1$

### CAL – Bezpodmienečné volanie podprogramu

*Popis:* Volá podprogram, ktorého začiatok je daný návěstím. Návratová adresa (inštrukcia po CAL) je uložená do zásobníka.

*Operácia:*

$PC \leftarrow [nav]$

*Syntax:*

CAL    nav

*Zásobník:*

$STACK \leftarrow PC + 1$

$SP \leftarrow SP - 1$

*Čítač inštrukcií:*

$PC \leftarrow [nav]$

## CZR – Volanie podprogramu ak Zero = 1

*Popis:* Táto inštrukcia zavolá podprogram daný návestím, ak je príznak Zero nastavený. Návratová adresa (inštrukcia po CZR) je uložená do zásobníka.

*Operácia:*

if  $Z = 1$  then  $PC \leftarrow [nav]$  else  $PC \leftarrow PC + 1$

*Syntax:*

CZR    nav

ak  $Z = 1$

*Zásobník:*

STACK  $\leftarrow PC + 1$

SP  $\leftarrow SP - 1$

ak  $Z = 0$

–

*Čítač inštrukcií:*

PC  $\leftarrow [nav]$

PC  $\leftarrow PC + 1$

## CNZ – Volanie podprogramu ak Zero = 0

*Popis:* Táto inštrukcia zavolá podprogram daný návestím, ak je príznak Zero nulový. Návratová adresa (inštrukcia po CNZ) je uložená do zásobníka.

*Operácia:*

if  $Z = 0$  then  $PC \leftarrow [nav]$  else  $PC \leftarrow PC + 1$

*Syntax:*

CNZ    nav

ak  $Z = 0$

*Zásobník:*

STACK  $\leftarrow PC + 1$

SP  $\leftarrow SP - 1$

ak  $Z = 1$

–

*Čítač inštrukcií:*

PC  $\leftarrow [nav]$

PC  $\leftarrow PC + 1$

## CCY – Volanie podprogramu ak Carry = 1

*Popis:* Táto inštrukcia zavolá podprogram daný návestím, ak je príznak Carry nastavený. Návratová adresa (inštrukcia po CCY) je uložená do zásobníka.

*Operácia:*

if  $CY = 1$  then  $PC \leftarrow [nav]$  else  $PC \leftarrow PC + 1$

*Syntax:*

CCY    nav

ak  $CY = 1$

*Zásobník:*

STACK  $\leftarrow PC + 1$

SP  $\leftarrow SP - 1$

ak  $CY = 0$

–

*Čítač inštrukcií:*

PC  $\leftarrow [nav]$

PC  $\leftarrow PC + 1$

## CNC – Volanie podprogramu ak Carry = 0

*Popis:* Táto inštrukcia zavolá podprogram daný návestím, ak je príznak Carry nulový. Návratová adresa (inštrukcia po CNC) je uložená do zásobníka.

*Operácia:*

if  $CY = 0$  then  $PC \leftarrow [nav]$  else  $PC \leftarrow PC + 1$

*Syntax:*

CNC    nav

ak  $CY = 0$

*Zásobník:*

STACK  $\leftarrow PC + 1$

SP  $\leftarrow SP - 1$

ak  $CY = 1$

–

*Čítač inštrukcií:*

PC  $\leftarrow [nav]$

PC  $\leftarrow PC + 1$

## RET – Návrat z podprogramu

*Popis:* Vykoná návrat z podprogramu. Návratová adresa je načítaná zo zásobníka.

*Operácia:*

PC  $\leftarrow$  STACK

*Syntax:*

RET

*Zásobník:*

SP  $\leftarrow SP + 1$

*Čítač inštrukcií:*

PC  $\leftarrow$  STACK

## 5. ŠPECIÁLNE INŠTRUKCIE

### EIT – Povolenie prerušenia

*Popis:* Táto inštrukcia povolí prerušenie procesora, ale až po vykonaní inštrukcie nasledujúcej po inštrukcii EIT.

*Operácia:*

$EI \leftarrow 1$

*Syntax:*

EIT

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

### DIT – Zakázanie prerušenia

*Popis:* Táto inštrukcia zakáže prerušenie procesora.

*Operácia:*

$EI \leftarrow 0$

*Syntax:*

DIT

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

### SCALL – Špeciálna inštrukcia pre interakciu programu s užívateľom

*Popis:* Táto inštrukcia umožňuje načítať do registra D klávesu stlačenú užívateľom, alebo zobraziť obsah registra D na displeji.

*Operácia:*

1. SCALL KEY :  $D \leftarrow \text{klávesa}$

2. SCALL DSP :  $\text{displej} \leftarrow D$

*Syntax:*

1. SCALL KEY

2. SCALL DSP

*Čítač inštrukcií:*

$PC \leftarrow PC + 1$

$PC \leftarrow PC + 1$

### BYTE – Definícia konštanty v pamäti programu

*Popis:* Táto pseudoinštrukcia umožňuje definovať konštanty umiestnené v pamäti programu. Ich čítanie zabezpečuje inštrukcia MMR.

*Syntax:*

BYTE K

*Parametre:*

$0 \leq K \leq 255$