

Intel 8080

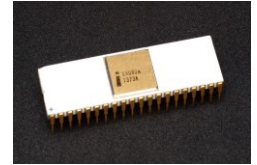
Produced April 1974; 43 years ago(!)

CPU clock: 2 MHz to 3.125 MHz (8080-A)
 Package: 40-pin DIP
 Power: +5, -5 and +12 V, ~ 1,3 Watt
 Military: - 55 °C to +125 °C
 Technology: 6 μm, 6 000 transistors, die size 20 mm²

Common manufacturer(s)

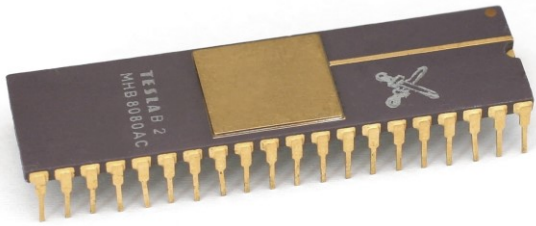
Intel, AMD, NEC, National Semiconductor, OKI,
 Siemens, Texas Instruments,
 Sovietsky zväz KR580VM80A,
 Polsko MCY7880, ČSSR Tesla Piešťany MHB8080A

r. 1979 – 500 000 ks / mesačne za cca \$3 – 4 / ks



By Konstantin Lanzet - CPU collectorCamera
<https://commons.wikimedia.org/wiki/index.php?title=7028099>

TESLA MHB8080A



By Konstantin Lanzet - CPU collectorCamera: Canon EOS 400D, CC BY-SA 3.0,
<https://commons.wikimedia.org/wiki/index.php?curid=7028099>

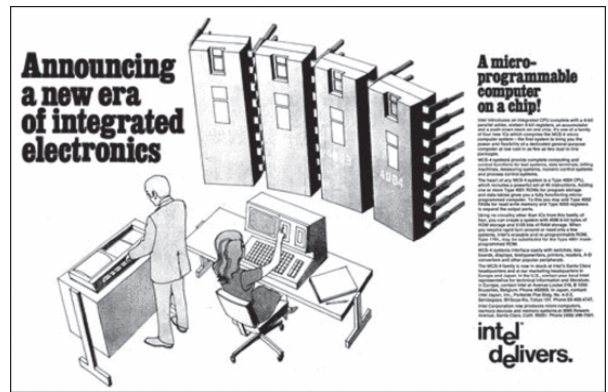
Intel 8080



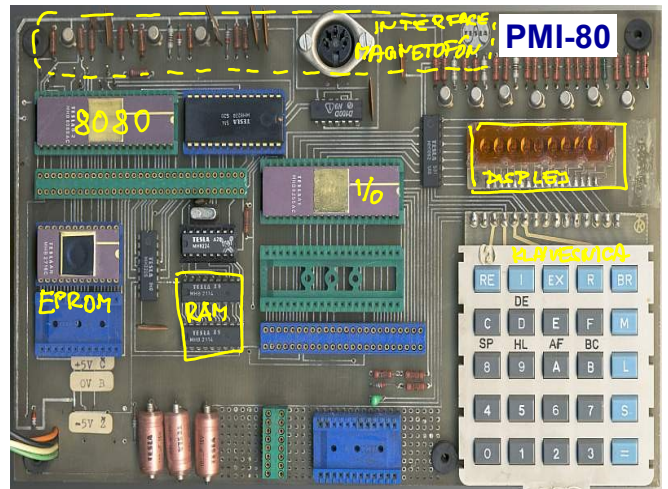
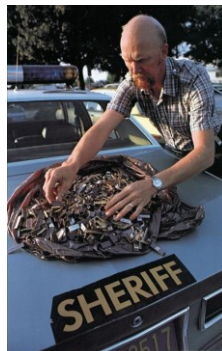
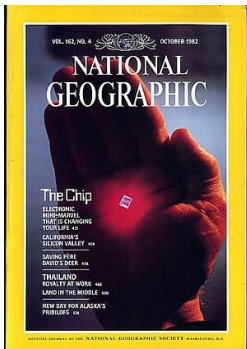
Rare Intel 8080a White Ceramic Processor Produced In the 441

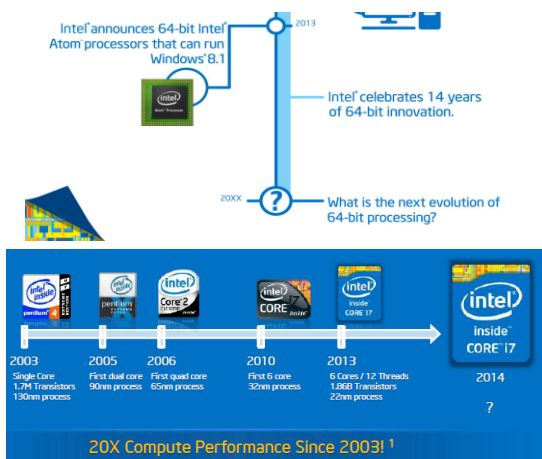
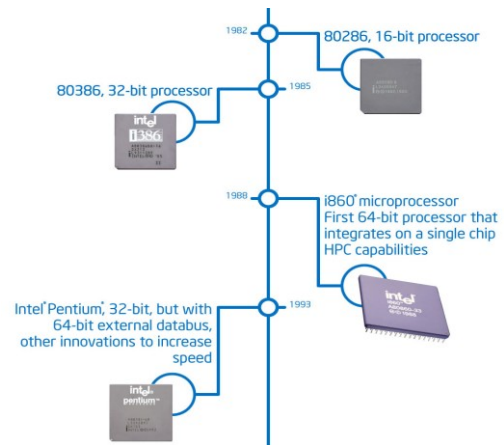
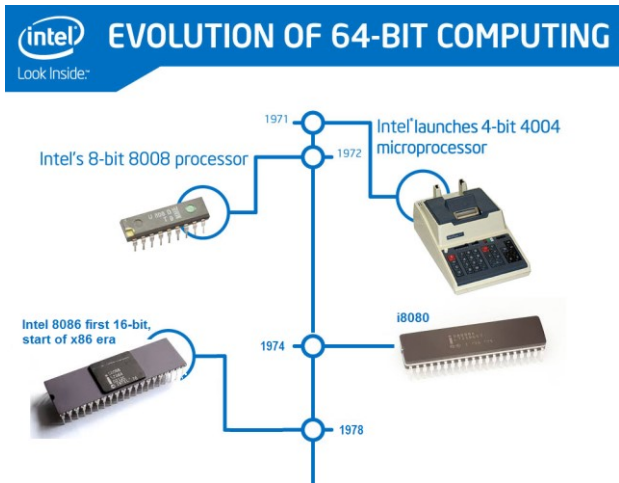
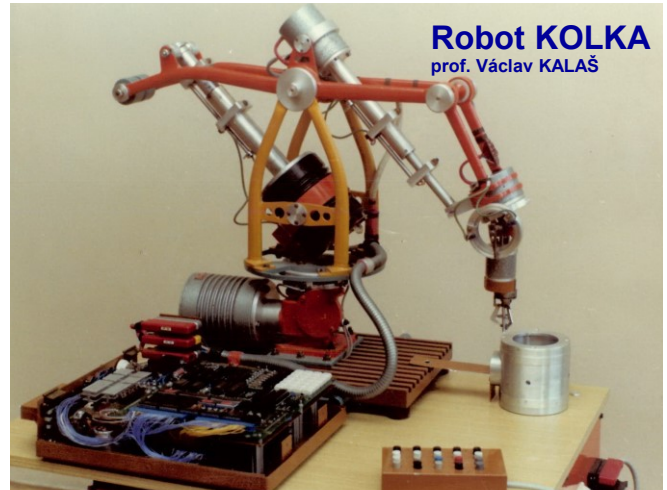
\$359.99

Buy it Now
+\$22.50 shipping



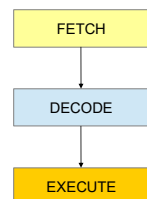
Intel 8080



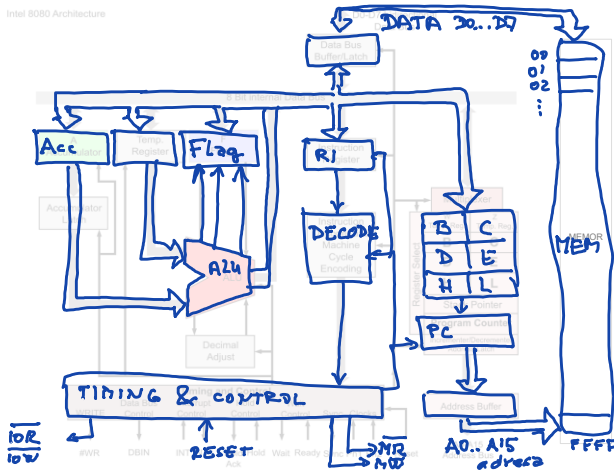


Inštrukčný cyklus

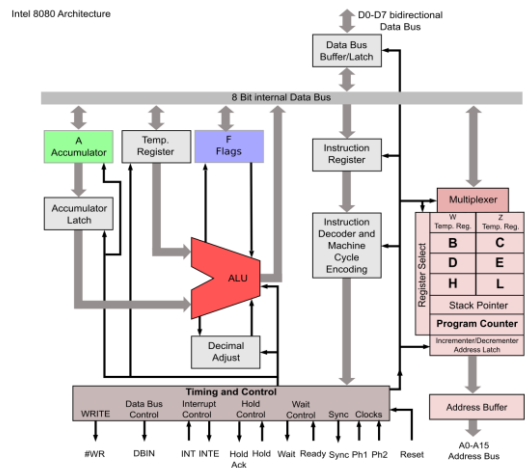
- Výber inštrukcie z pamäte
- Dekódovanie inštrukcie
- Výber operandov
- Vykonanie požadovanej operácie
- Zápis výsledkov do pamäte



Intel 8080 Architecture

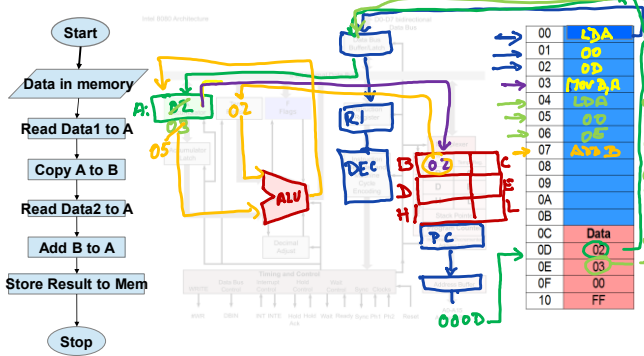


Intel 8080 Architecture



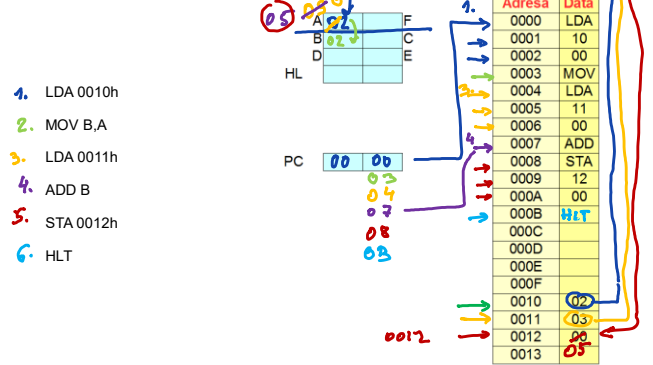
Príklad 2:

súčet dvoch čísel v pamäti



Príklad 2:

súčet dvoch 8-bitových čísel



Pseudoinštrukcie

```

0000 ; ADDITION OF TWO 8 BIT NUMBERS
0000 CISLO EQU 0x000D
000D ORG 000DH
000D 02 03 00 DB 02H,03H,00H

0000 ORG 0000
0000 START:
0000 3A 0D 00 LDA 000DH
0003 47 MOV B,A
0004 3A 0E 00 LDA 000EH
0007 80 ADD B
0008 32 0F 00 STA 000FH
000B 7E HLT
    
```

Adresa	Data
00	Program
01	
02	
03	
04	
05	
06	
07	
08	
09	
0A	
0B	
0C	Data
0D	02
0E	03
0F	00
10	FF

Pseudoinštrukcie

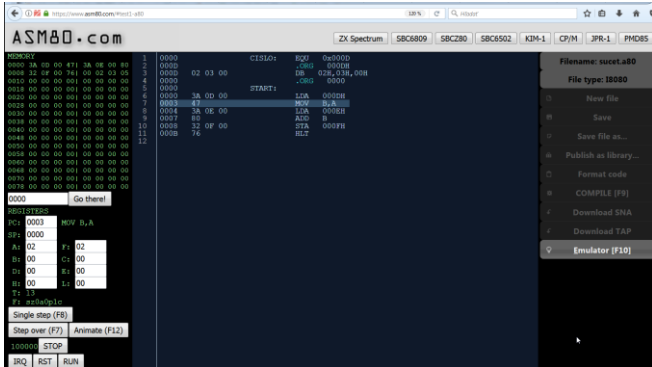
```

0000 ; ADDITION OF TWO 8 BIT NUMBERS
0000 CISLO EQU 0x000D
000D ORG 000DH
000D 02 03 00 DB 02H,03H,00H

0000 ORG 0000
0000 START:
0000 3A 0D 00 LDA 000DH
0003 47 MOV B,A
0004 3A 0E 00 LDA 000EH
0007 80 ADD B
0008 32 0F 00 STA 000FH
000B 7E HLT
    
```

Adresa	Data
00	Program
01	
02	
03	
04	
05	
06	
07	
08	
09	
0A	
0B	
0C	Data
0D	02
0E	03
0F	00
10	FF

Príklad 2: ukážka na ASM80.com



Inštrukčná sada procesora 8080

- Skladá sa z
 - 74 základných inštrukcií, napr. **MOV**
 - 246 konkrétnych inštrukcií, napr. **MOV A,B**

• 8080 inštrukcie delíme na

1. Data Transfer (Copy)
2. Arithmetic
3. Logical and Bit manipulation
4. Branch
5. Machine Control

- Presunové
- Aritmetické
- Logické
- Skokové
- Riadiace

Riadiace inštrukcie

NOP

No operation

NOP ; nerobim nic

Operandy: žiadne
Cykly: 4

Dĺžka: 1 Byte

Príznamy: S Z A P C
- - - - -

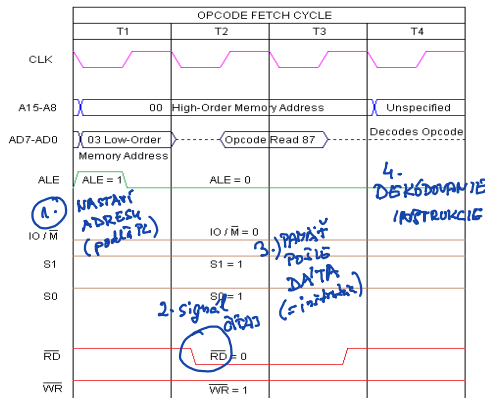
Popis: Inštrukcia, ktorá nerobí vôbec nič. Používa sa napr. na jemné doladenie časových slučiek alebo na spomalenie V/V procesov.

Príklad:

```
OUT 44, A ; na port 44 zapisem obsah registra A
NOP      ; počkam, lebo viem, ze je pomaly
OUT 44, B ; na port 44 zapisem dalsiu hodnotu z B
```

NOP

No Operation



HLT

Halt

HLT ; zastavim mikroprocesor

Operandy: žiadne
Cykly: 4

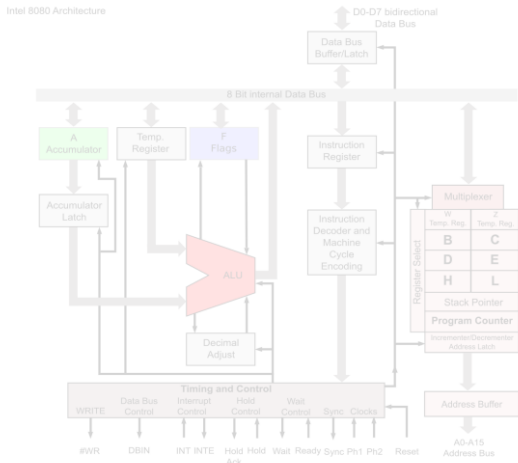
Dĺžka: 1 Byte

Príznamy: S Z A P C
- - - - -

Popis: Inštrukcia, ktorá zastaví činnosť procesora. Zobudí sa až na RESET, alebo INT

Príklad:

```
OUT 44, A ; na port 44 zapisem obsah registra A
HLT      ; koniec, mam hotovo
```

Presunové inštrukcie

MOV

Move Data I.

MOV R1, R2 ; presun obsahu registra R2 do R1
skopiruj

Operandy: destination, source
Cykly: 5

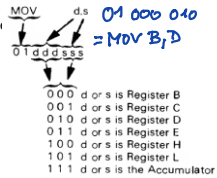
Dĺžka: 1 Byte
= 4 + 1 = 5 cykly

Priznaky: S Z A P C
- - - - -

Popis: Presunie obsah z ľubovôleho registra do in

Príklad:

```
MOV A, B ; nactaj hodnotu z B
RAR      ; posun hodnotu vpravo
MOV B, A ; vysledok uloz do B
```



MOV

Move Data II.

MOV R, M ; presun obsahu pamätového miesta {HL} do R
skopiruj

MOV M, R ; presun obsahu registra do pamätového miesta {HL}
skopiruj

Operandy: destination, HL

Dĺžka: 1 Byte

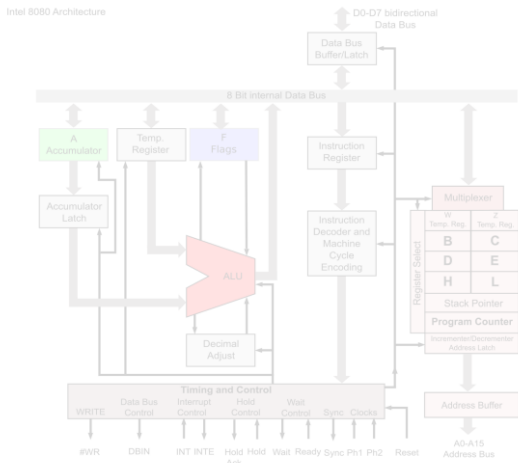
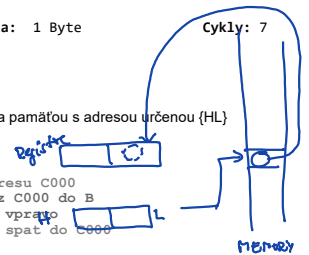
Cykly: 7

Priznaky: S Z A P C
- - - - -

Popis: Presunie obsah medzi registrom a pamätou s adresou uröenou {HL}

Príklad:

```
LXI H, C000H ; uloz do HL adresu C000
MOV B, M     ; presun obsah z C000 do B
RAR          ; posun hodnotu vpravo
MOV M, B     ; vysledok uloz spat do C000
```



LDA Load Acumulator from Memory

LDA addr16 ; presun obsahu pamätového miesta {ADDR} do A

Operandy: adresa

Dĺžka: 3 Byte

Cykly: 16

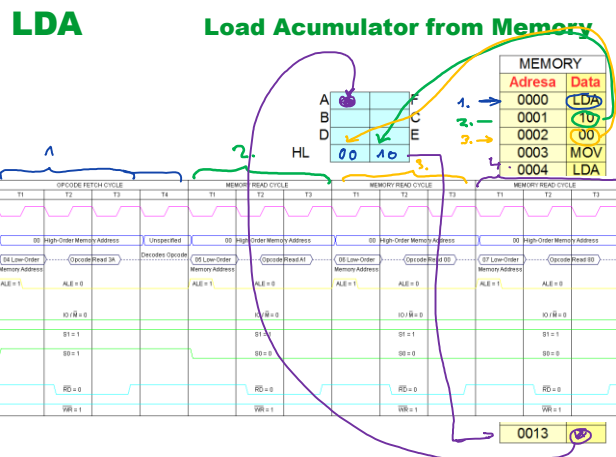
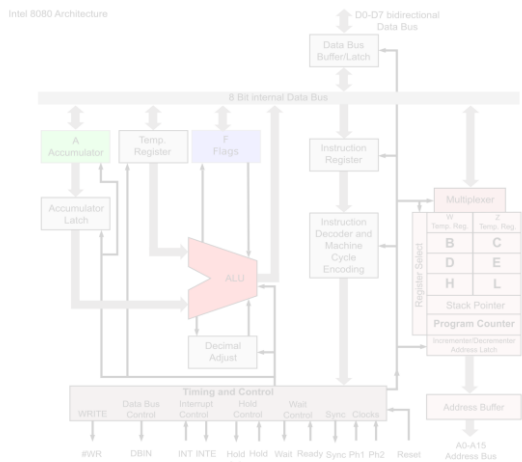
Priznaky: S Z A P C
- - - - -

Popis: Presunie do akumulátora obsah pamätového miesta {addr16}

Príklad:

```
LDA C000H ; uloz do A obsah z adresy C000H
RAR       ; posun hodnotu vpravo
STA C000H ; vysledok uloz spat do C000
```





STA Store Acumulator to Memory

STA addr16 ;presun obsahu pamätového miesta {ADDR} do A

Operandy: adresa **Dĺžka:** 3 Byte **Cykly:** 13

Príznamky: S Z A P C

Popis: Presunie obsah Akumulátora do pamäte {addr16}

Príklad:

```
LDA C000H ; uloz do A obsah z adresy C000h
RAR       ; posun hodnotu vpravo
STA C000H ; vysledek uloz spat do C000
```

LXI Load Immediate 16-bit value into register pair

LXI RP, DATA16 ;nahraj do RP priame data DATA16

Operandy: registrový pár, data **Dĺžka:** 3 Byte **Cykly:** 10

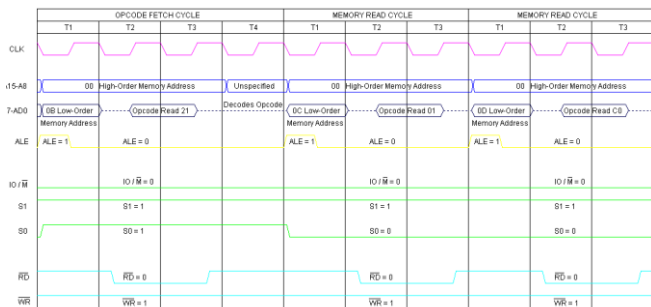
Príznamky: S Z A P C

Popis: Presunie do registrového páru data

Príklad:

```
LXI H,C000H ; uloz do HL adresu C000
MOV B,M     ; presun obsah z C000 do B
RAR         ; posun hodnotu vpravo
MOV M,B     ; vysledek uloz spat do C000
```

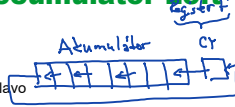
LXI H,C001



Aritmetické inštrukcie

RAL Rotate Accumulator Left (+carry)

RAL ; posun obsahu registra A vľavo



Operandy: žiadne Dĺžka: 1 Byte Cykly: 4

Príznyky: S Z A P C

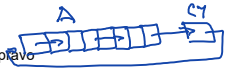
Popis: Posunie akumulátor vľavo a najvyšší bit preniesie do Carry. Do najnižšieho bitu doplní Carry.

Príklad:

```
MOV A,M ; nactaj hodnotu z pamate
RAL     ; posuniem hodnotu doľava
MOV M,A ; vysledok uloz do pamate
```

RAR Rotate Accumulator Right (+carry)

RAR ; posun obsahu registra A vpravo



Operandy: žiadne Dĺžka: 1 Byte Cykly: 4

Príznyky: S Z A P C

Popis: Posunie akumulátor vpravo a najnižší bit preniesie do Carry. Do najvyššieho bitu doplní Carry.

Príklad:

```
MOV A,M ; nactaj hodnotu z pamate
RAR     ; posuniem hodnotu doprava
MOV M,A ; vysledok uloz do pamate
```

ADD Addition

ADD R1 ;pripočítaj obsah R1 k Akumulátoru

ADD M ;pripočítaj obsah M k Akumulátoru

Operandy: destination, source Dĺžka: 1 Byte Cykly: 4(7)

Príznyky: S Z A P C

Popis: Pripočítaj obsah registra alebo pamätového miesta M k Akumulátoru. Výsledok je vždy v Akumulátore.

Príklad:

```
MOV A,B ; nactaj hodnotu z B
ADD C   ; pripocitaj C, vysledok A = B + C
```

SUB Subtraction

SUB R1 ;odpočítaj obsah R1 od Akumulátora

SUB M ;odpočítaj obsah pamätového miesta M od Akumulátora

Operandy: register Dĺžka: 1 Byte Cykly: 4(7)

Príznyky: S Z A P C

Popis: Spraví rozdiel a výsledok uloží späť do akumulátora.

Príklad:

```
MOV A,B ; nactaj hodnotu z B
SUB C   ; odpočitaj C, vysledok A = B - C
```

INR Increment

INR B ; B <-- B+1

Operandy: A,B,...M Dĺžka: 1 Byte Cykly: 5(10)

Príznyky: S Z A P C

Popis: Pripočíta 1 k obsahu ľubovoľného registra alebo pamätového miesta

Príklad:

```
MOV A,B ; nactaj hodnotu z B
INR A   ; k akumulátoru pripocitaj 1
```

INX Increment

INX H ; HL <-- HL+1

Operandy: B, D, H, SP Dĺžka: 1 Byte Cykly: 5

Príznyky: S Z A P C

Popis: Presunie obsah z ľubovoľného registra do iného Prvý je vždy destination

Príklad:

```
MOV A,M ; nactaj hodnotu z pamate {HL}
INX H   ; pripocitaj jednotku k HL
ADD M   ; pripocitaj dalsiu hodnotu z pamate {HL}
```

DCR**Decrement****DCX****Decrement**

DCR B ; B ← B-1

DCX H ; HL ← HL-1

Operandy: A, B, C, ... **Dĺžka:** 1 Byte **Cykly:** 5**Operandy:** B, D, H, SP **Dĺžka:** 1 Byte **Cykly:** 5**Príznaky:** S Z A P C
+ + + + -**Príznaky:** S Z A P C
- - - - (1)**Popis:** Dekrementuje (odpočíta 1) od registra. Ovplyní niektoré príznaky.**Popis:** Odpočíta jednotku z registrového páru**Príklad:****Príklad:**