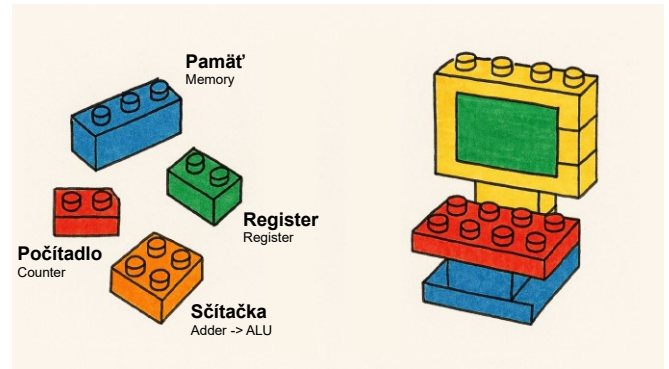
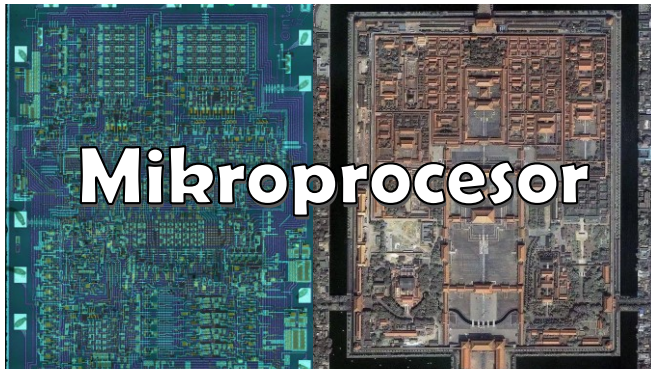
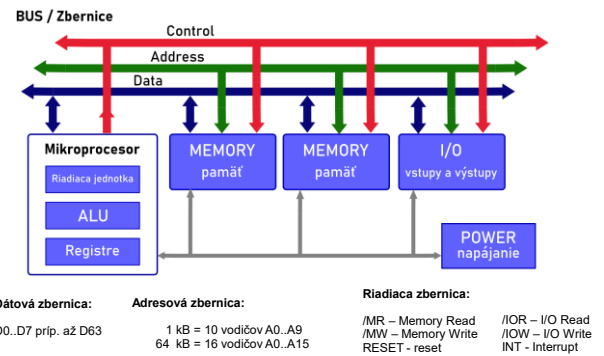
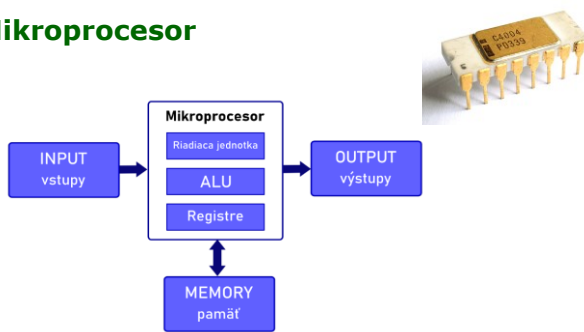




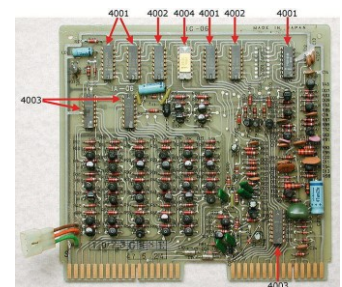
Mikroprocesor

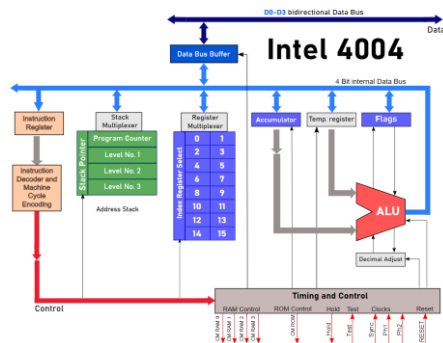


Busicom 141-PF calculator

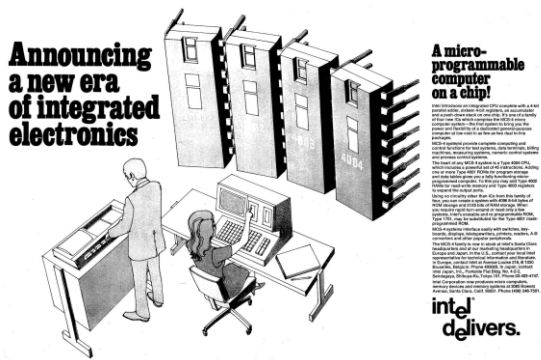
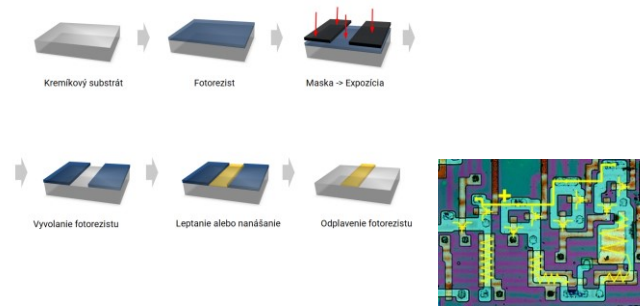


- 4001 2kbit ROM
- 4002 320-bit RAM
- 4003 10-bit shift register
- 4004 4-bit CPU (the microprocessor)

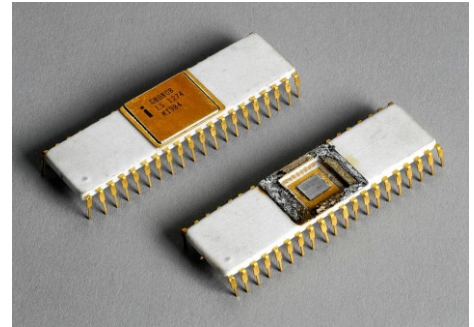




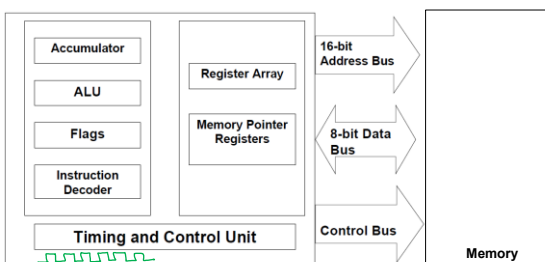
Výroba čipov - fotolitografia



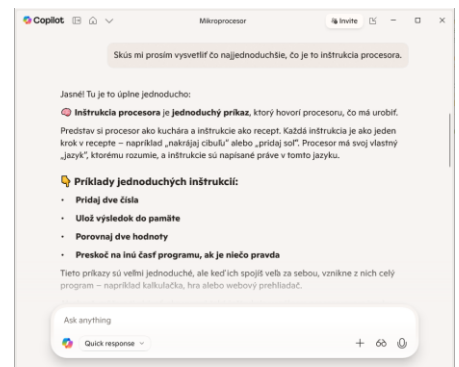
Intel 8080



Zjednodušený model i8080



Inštrukcie



Inštrukcie

Príklad inštrukcie „Pripočítaj B k A“:

```
ADD A, B
```

- **ADD** je príkaz: „sčítaj“
- **A** a **B** sú miesta (registre), kde sú uložené čísla
- Procesor sčíta hodnoty v A a B a výsledok uloží do A

Zložitejší príklad:

```
MVI A, 5
ADD A, 3
```

- **MOV** presunie číslo 5 do miesta A
- **ADD** pripočíta k A číslo 3
→ teraz je v A číslo 8

Inštrukcia

Pre človeka? (no...)

NOP
No OPeration

MVI A, 55h
MOVe Immediate Data

STA 0010h
STore Accumulator to Memory

HLT
HaLT = Stop

Pre počítač

binárne	hexadecimálne
0000 0000	00 hex
0011 1110	3E
0101 0101	55
0011 0010	32
0001 0000	10
0000 0000	00
0111 0110	76

Strojový kód vs. jazyk symbolických adries

Adresa	Obsah
BIN	BIN
0000	00111010
0001	00001101
0010	00000000
0011	01000111
0100	00111010
0101	00001110
0110	00000000
0111	10000000
1000	00110010
1001	00001111
1010	00000000
1011	01110110
1100	00000000
1101	00000010
1110	00000011
1111	00000000
10000	00000000

Príklady:

MOV – presun medzi registrami
MVI – vlož hodnotu
ADD – sčítaj dve hodnoty
SUB – odíťaj dve hodnoty

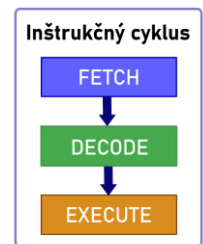
; sucet dvoch cisel A a B

START: **MVI A, 01h** ; vlož 01 do A
MVI B, 02h ; vlož 02 do B
ADD B ; A = A + B

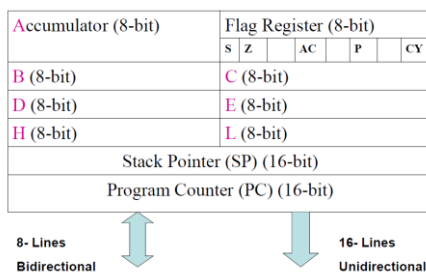
Inštrukčný cyklus

Hlavný rozdiel oproti všetkým dovtedy vybraným súčiastkám je ten, že funkcia obvodu nie je pevná, ale určená inštrukciami programu

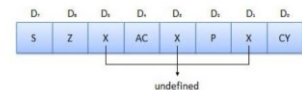
- Výber inštrukcie z pamäte
- Dekódovanie inštrukcie
- Výber operandov
- Vykonanie požadovanej operácie
- Zápis výsledkov do pamäte



Programátorský model 8080/85



Príznamy: Flags



Nedá sa zapisovať, len čítať. Nie všetky inštrukcie ovplyvňujú F.

Sign (S) =1 ak je výsledok záporný

Zero (Z) =1 ak je výsledok nula

Parity (P) = vždy tak, aby doplnil počet 1 vo výsledku na nepárny
(t.j. = 0, ak je nepárny, =1, ak je párny)

Carry (CY) =1 ak nastalo pretečenie, t.j. výsledok vyžaduje prenos do vyššieho rádu

Auxiliary carry (AC) pre prenos medzi spodnou a hornou polovicou bajtu, používa sa pre BCD aritmetiku

Inštrukčná sada procesora 8080

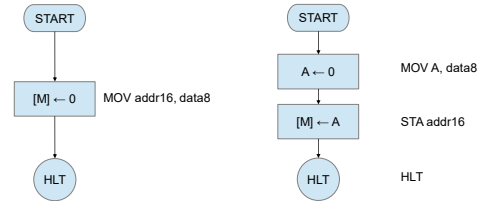
- Skladá sa z
 - 74 základných inštrukcií, napr. **MOV**
 - 246 konkrétnych inštrukcií, napr. **MOV A,B**
- 8080 inštrukcie delíme na

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

Presunové
Aritmetické
Logické
Skokové
Riadiace

Príklad 1: nulovanie pamäťového miesta

POZN.: To nie je to isté ako „vyprázdniť šuflík“



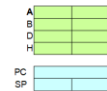
Príklad 1: nulovanie pamäťového miesta

Adresa	Data	Význam
0000	00	
0001	3C	
0002	00	
0003	32	
0004	10	
...		
0005	00	
0006	76	
0007		
0008		
0009		
000A		
...		
0037		
0038		
0039		
003A		
003B		
003C		
003D		
003E		
003F		
0040		

opcode	instruction	size	flags	function
0x00	NOP	1		no-ops níl
0x01	LXI B, D16	3		B ← byte 3, C ← byte 2
0x02	STAX B	1		(BC) ← A
...				
0x30	---			no-ops
0x31	LXI SP, D16	3		SP, hi ← byte 3, SP, lo ← byte 2
0x32	STA addr	3		(addr) ← A
0x33	INR SP	1		SP ← SP + 1
0x34	INR M	1	Z, S, P, AC	(HL) ← (HL) + 1
0x35	DCR M	1	Z, S, P, AC	(HL) ← (HL) - 1
0x36	MVI M, D8	2		(HL) ← byte 2
0x37	STC	1	CY	CY ← 1
0x38	---			no-ops
0x39	DAD SP	1	CY	HL ← HL + SP
0x3A	LDA addr	3		A ← (addr)
0x3B	DCX SP	1		SP ← SP - 1
0x3C	INR A	1	Z, S, P, AC	A ← A + 1
0x3D	DCR A	1	Z, S, P, AC	A ← A - 1
0x3E	MVI A, D8	2		A ← byte 2
...				
0x76	HLT	1		special

Príklad 1: nulovanie pamäťového miesta

NOP
MVI A, 00h
STA 0010h
HLT

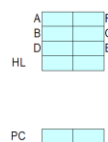


Adresa	Data	Význam
0000	00	NOP
0001	3E	MVI
0002	00	
0003	32	STA
0004	10	
0005	00	
0006	76	HLT
0007		
0008		
0009		
...		
0037		
0038		
0039		
003A		
003B		
003C		
003D		
003E		
003F		
0040		

Inštrukcie aj data sú v tej istej pamäti = von Neumann

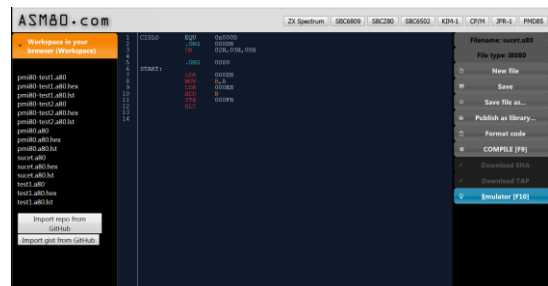
Príklad 1b: presuny sem a tam

MVI A, 07h
MOV B, A
MOV C, D
STA 0010h
MVI A, 00h
STA 0011h
LDA 0010h
HLT



Adresa	Data
0000	MVI A
0001	07
0002	MOV B, A
0003	MOV C, D
0004	STA
0005	10
0006	00
0007	MVI A
0008	00
0009	STA
000A	11
000B	0
000C	LDA
000D	10
000E	0
000F	HLT
0010	
0011	
0012	
0013	

Príklad 1b: ukážka na ASM80.com



Riadiace inštrukcie

NOP

No operation

NOP ; nerobim nic

Operandy: žiadne Dĺžka: 1 Byte Cykly: 4

Príznaky: S Z A P C
- - - - -

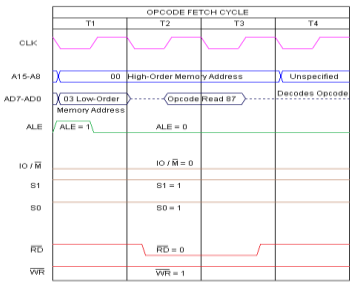
Popis: Inštrukcia, ktorá nerobí vôbec nič. Používa sa napr. na jemné doladenie časových služiek alebo na spomalenie V/V procesov.

Príklad:

OUT 44, A ; na port 44 zapíšem obsah registra A
NOP ; počkam, lebo viem, že je pomaly
OUT 44, B ; na port 44 zapíšem ďalšiu hodnotu z B

NOP

No Operation



HLT

Halt

HLT ; zastavím mikroprocesor

Operandy: žiadne Dĺžka: 1 Byte Cykly: 4

Príznaky: 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 zapíšem obsah registra A
HLT ; koniec, mam hotovo

Presunové inštrukcie

MOV

Move Data I.

MOV R1,R2 ;presun obsahu registra R2 do R1

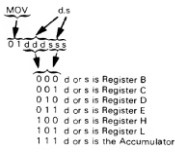
Operandy: destination, source Dĺžka: 1 Byte Cykly: 5

Príznaky: S Z A P C
- - - - -

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

Príklad:

MOV A,B ; nacistaj hodnotu z B
CMA ; sprav doplnok - komplement A
MOV B,A ; vysledok uloz do B



MOV**Move Data II.**

```
MOV R, M ;presun obsahu pamäťového miesta {HL} do R
MOV M, R ;presun obsahu registra do pamäťového miesta {HL}
```

Operandy: destination, HL **Dĺžka:** 1 Byte **Cykly:** 7

Príznyaky: S Z A P C
- - - - -

Popis: Presunie obsah medzi registrom a pamäťou s adresou určenou {HL}

Priklad:

```
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 pamäťového miesta {ADDR} do A
```

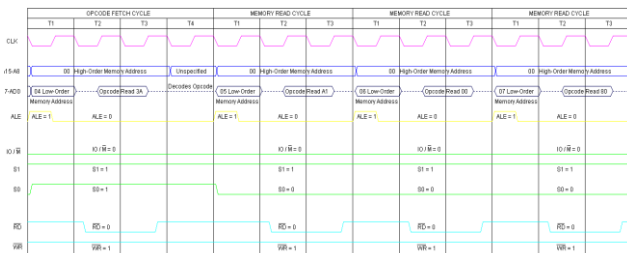
Operandy: adresa **Dĺžka:** 3 Byte **Cykly:** 13

Príznyaky: S Z A P C
- - - - -

Popis: Presunie do akumulátora obsah pamäťového miesta {addr16}

Priklad:

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

LDA**Load Acumulator from Memory****STA****Store Acumulator to Memory**

```
STA addr16 ;presun obsahu pamäťového miesta {ADDR} do A
```

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

Príznyaky: S Z A P C
- - - - -

Popis: Presunie obsah Akumulátora do pamäte {addr16}

Priklad:

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

Aritmetické inštrukcie**ADD****Addition**

```
ADD R1 ;pripocítaj obsah R1 k Akumulátoru
ADD M ;pripocítaj obsah [M] k Akumulátoru
```

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

Príznyaky: S Z A P C
+ + + + +

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

Priklad:

```
MOV A,B ; nacitaj hodnotu z B
ADD C ; pripocitaj C, vysledok A = B + C
ADD M ; pripocitaj [HL], vysledok A = A + [HL]
```

SUB

Subtraction

SUB R1 ;odpočítaj obsah R1 od Akumulátora
SUB M ;odpočítaj obsah pamäťového miesta M od Akumulátora

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

Príznaky: S Z A P C
 + + + + +

Popis: Spraví rozdiel a výsledok uloží späť do akumulátora.

Príklad:

```
MOV A,B ; nacistaj hodnotu z B
SUB C ; odpočítaj C, výsledok A = B - C
```

INR

Increment

INR B ; B <-- B+1

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

Príznaky: S Z A P C
 + + + + -

Popis: Pripočíta 1 k obsahu ľubovoľného registra alebo pamäťového miesta

Príklad:

```
MOV A,B ; nacistaj hodnotu z B
INR A ; k akumulátoru pripočítaj 1
```

DCR

Decrement

DCR B ; B <-- B-1

Operandy: A,B,C,... Dĺžka: 1 Byte Cykly: 5

Príznaky: S Z A P C
 + + + + -

Popis: Dekrementuje (odpočíta 1) od registra. Ovplyvní niektoré príznaky.

Príklad:

Zhrnutie

- Inštrukčný cyklus fetch-decode-execute
- Činnosť mikroprocesora
- Akumulátor, Flags Príznaky Z,S,P,C,AC
- Inštrukcie presunu dát MOV, MVI, STA, LDA
- Aritmetické inštrukcie ADD, SUB
- Riadiace inštrukcie HLT, NOP