

1

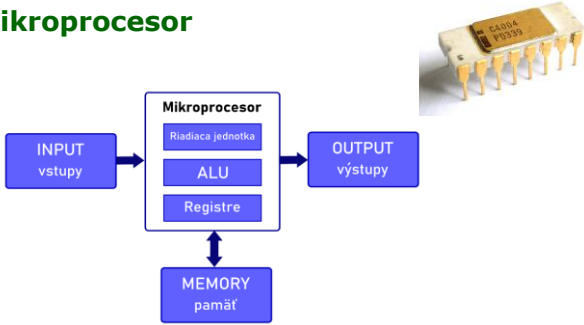
20:00 Bonusy



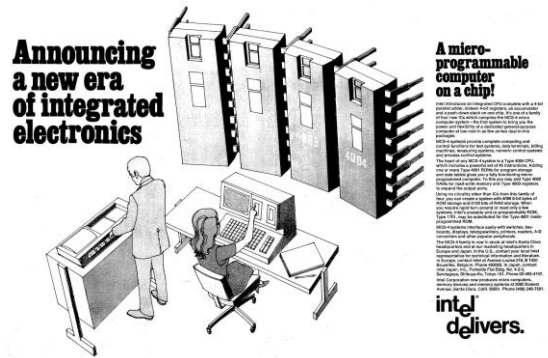
<https://senzor.robotika.sk/zp/bonus2/>

2

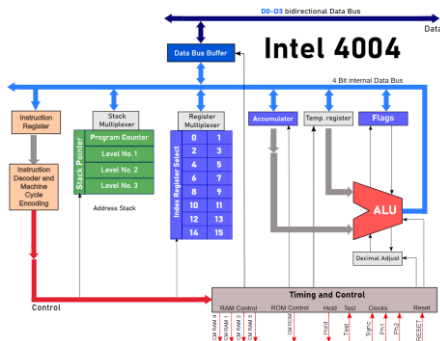
Mikroprocesor



3



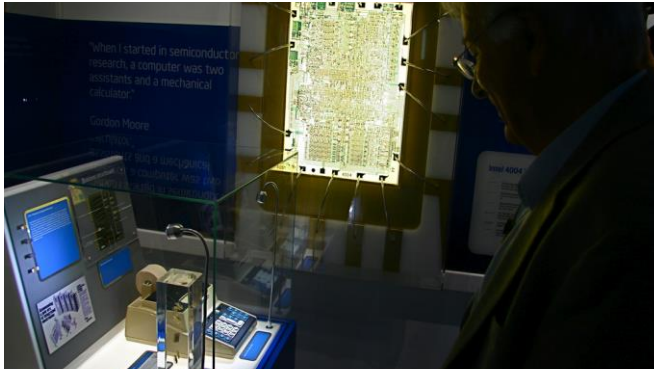
4



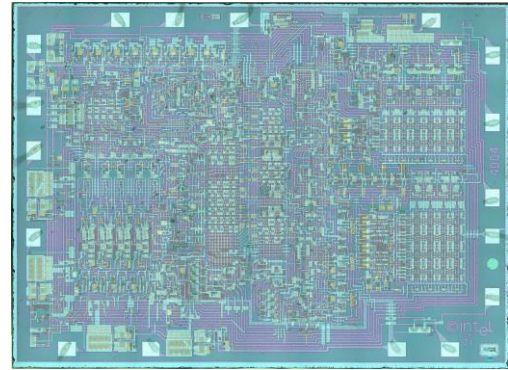
5



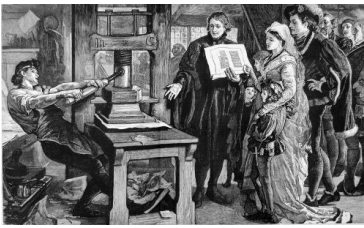
6



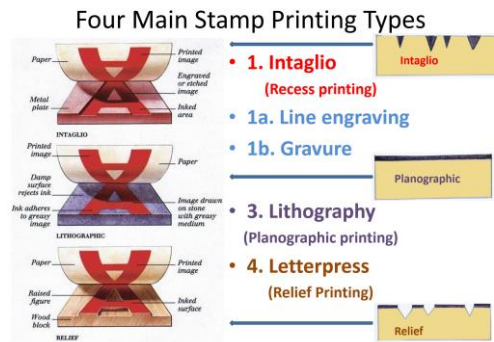
7



8

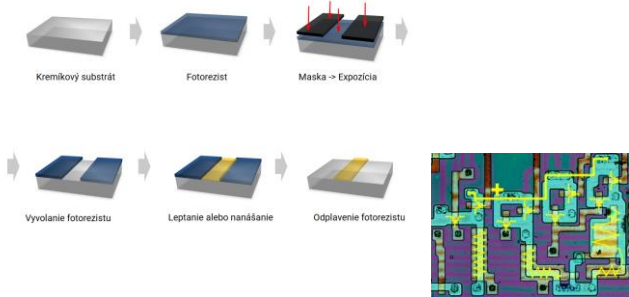


9

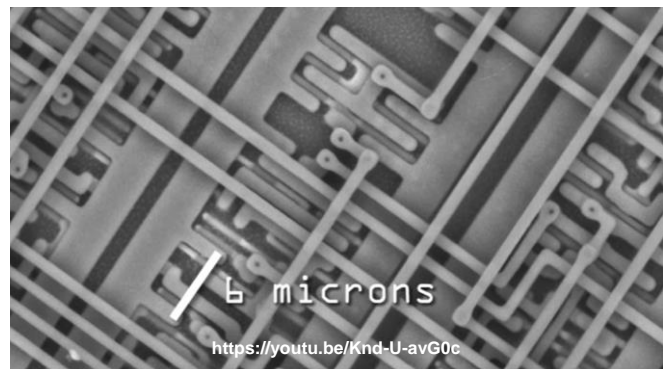


10

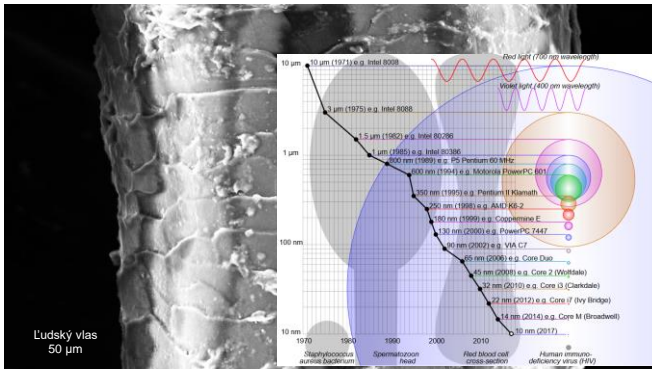
Výroba čipov - fotolitografia



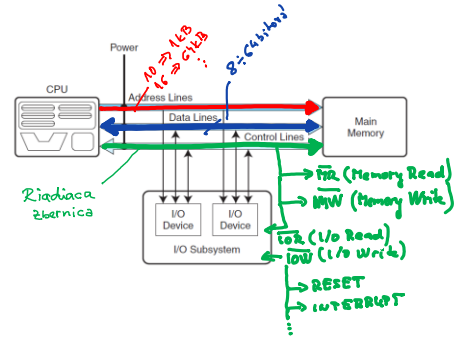
11



12



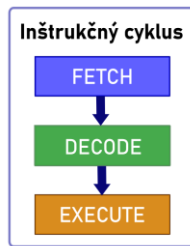
13



14

Inštrukčný cyklus

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



15

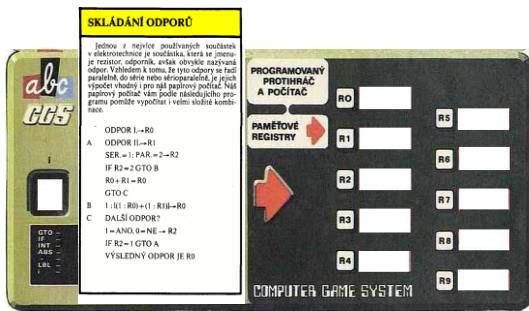
Inštrukcia

Pre človeka? (no...)

Pre počítač

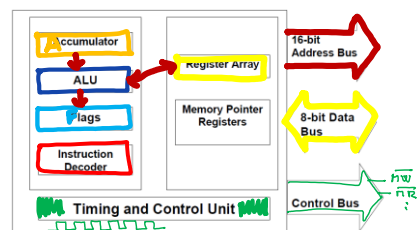
	binárne	hexadecimálne
NOP No Operation	0000 0000	00 hex
MOV A, 55h MOVE Immediate Data	0011 1110 0101 0101	3E 55
STA 0010h STORE Accumulator to Memory	0011 0010 0001 0000 0000 0000	32 10 00
HLT HaLT = Stop	0111 0110	76

16



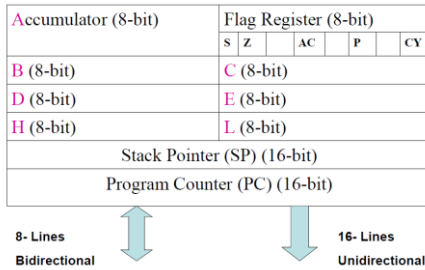
17

Programátorský model 8080/85



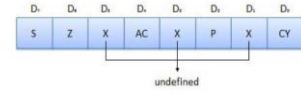
18

Programátorský model 8080/85



19

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 párný)
- 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

20

Inštrukčný cyklus



21

Inštrukčná sada procesora 8080

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

- Data Transfer (Copy)
- Arithmetic
- Logical and Bit manipulation
- Branch
- Machine Control

Presunové
Aritmetické
Logické
Skokové
Riadiace

22

Strojový kód vs. jazyk symbolických adries

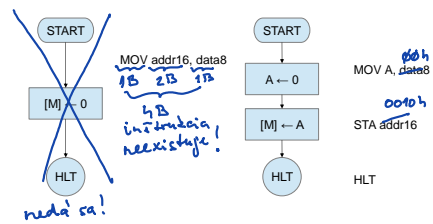
Adresa	Obsah
BIN	BIN
0000	00111010
0001	00001101
0010	00000000
0011	01000111
0100	00111010
0101	00001110
0110	00000000
0111	10000000
1000	00110010
1001	00001111
1010	00000000
1011	01110110
1100	00000000
1101	00000010
1110	00000011
1111	00000000
10000	00000000

Príklady:

- MOV – presun medzi registrami
 - MVI – vlož hodnotu
 - ADD – sčítaj dve hodnoty
 - SUB – odítaj dve hodnoty
- ; sučet dvoch čísel A a B
- START: MVI A, 01h ; vlož 01 do A
 MVI B, 02h ; vlož 02 do B
 ADD B ; A = A + B

23

Príklad 1: nulovanie pamäťového miesta



24

Príklad 1: nulovanie pamäťového miesta

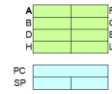
MEMORY		
Adresa	Data	Vyznam
0000	00	
0001	3E	
0002	00	
0003	32	
0004	10	
0005	00	
0006	76	
0007		
0008		
0009		
000A		
...		
0037		
0038		
0039		
003A		
003B		
003C		
003D		
003E		
003F		
0040		

8080 By Opcode				
opcode	Instruction	size	flags	function
0x00	NOP	1		nerobí nič
0x01	LXI B, D16	3		B <- byte 3, C <- byte 2 (BC) <- A
0x02	STA B	1		
...
0x30	---	---	---	neschádza
0x31	LXI SP, D16	3		SP.HI <- byte 3, SP.Lo <- byte 2
0x32	STA adr	3		(adr) <- A
0x33	INX SP	1		SP = SP + 1
0x34	INX H	1	Z, S, P, AC	(HL) <- (HL)+1
0x35	DCR H	1	Z, S, P, AC	(HL) <- (HL)-1
...
0x36	MVI M, D8	2		(HL) <- byte 2
0x37	STC	1	CY	CY = 1
0x38	---	---	---	neschádza
0x39	DAD SP	1	CY	HL = HL + SP
0x3a	LDA adr	3		A <- (adr)
0x3b	DCX SP	1		SP = SP-1
0x3c	INX 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

25

Príklad 1: nulovanie pamäťového miesta

```
NOP
MVI A, 00h
STA 0010h
HLT
```

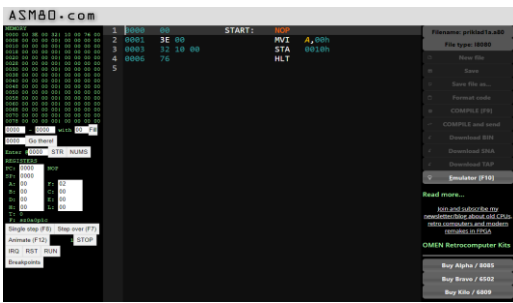


MEMORY		
Adresa	Data	Vyznam
0000	00	NOP
0001	3E	MVI
0002	00	
0003	32	STA
0004	10	
0005	00	
0006	76	HLT
0007		
0008		
0009		
000A		
...		
0037		
0038		
0039		
003A		
003B		
003C		
003D		
003E		
003F		
0040		

Inštrukcie aj data sú v tej istej pamäti = von Neumann

26

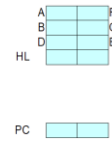
Príklad 1a: ukážka na ASM80.com



27

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
```

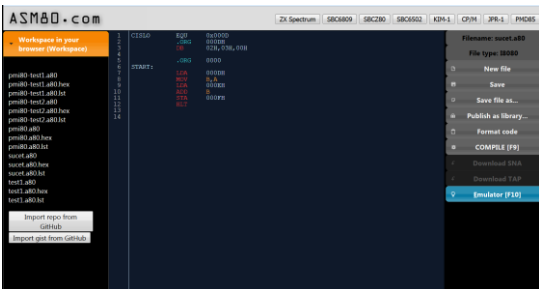


MEMORY		
Adresa	Data	
0000	MVI A	
0001	07	
0002	MOV B, A	
0003	MOV C, D	
0004	STA	
0005	10	
0006	00	
0007	MVI A	
0008	00	
0009	STA	
000A	11	
000B	0	
000C	LDA	
000D	10	
000E	0	
000F	HLT	
0010		
0011		
0012		
0013		

slido

28

Príklad 1b: ukážka na ASM80.com



Riadiace inštrukcie

29

30

NOP No operation

NOP ; nerobim nic

Operandy: žiadne Dĺžka: 1 Byte Cykly: 4

Príznamy: S Z A P C
- - - - -

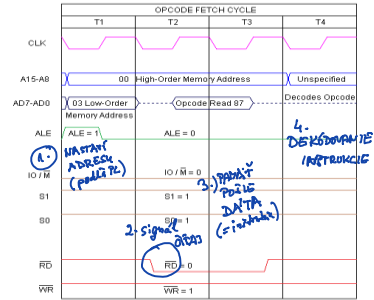
Popis: Inštrukcia, ktorá nerobí vôbec nič. Používa sa napr. na jemné doladenie časových slučiek alebo na spomalenie V/V procesov.

Príklad:

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

31

NOP No Operation



32

HLT Halt

HLT ; zastavím mikroprocesor

Operandy: žiadne Dĺžka: 1 Byte Cykly: 4

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 zapíšem obsah registra A
HLT ; koniec, mam hotovo

33

Zhrnutie

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

34