

Základy počítačov: prednáška č.10

Join at
slido.com
#ZPOC



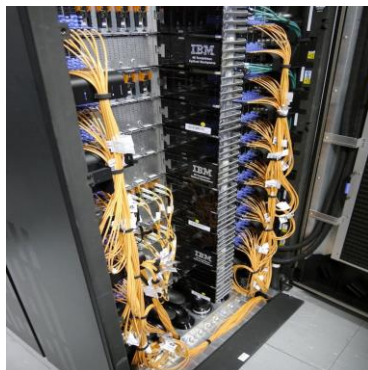
1

SAV: Múzeum počítačov

12.12. piatok
15.12. pondelok
16.12. utorok
17.12. streda

Záujem? **Sli.do**

2



Exkurzia SAV

History



Supercomputer

Aurel ⇒ Devana ⇒ Perun
128 ⇒ 800 TFLOP/s
4096 + 0 ⇒ 9472 + 221184 jadier

3

Náhradné testy

```
1 0010 01 02 03 04 25 .ORG 0010h
2 0010 01 02 03 04 25 .ORG 01,02,03,04,25
3 0000 0000 .ORG 0000h
4
5 0000 3A 10 00 LDA 0010h
6 0003 47 MOV R,A
7 0004 3A 11 00 LDA 0011h
8 0007 85 AND R
9 0008 22 12 00 STA 0012h
10 000B 76 HLT
11
```

Na **poslednom cvičení** budú otvorené znova všetky testy 6-12 pre tých študentov, ktorí ich nepísali.

Druhýkrát sa písať nedá.

4

Doučovanie z Matlabu - STREDA 26.11. / 11:00 / CD-150

Milé študentky a študenti programu Digitálne technológie,

radi by sme vás pozvali na **doučovanie / hromadnú konzultáciu**, ktorá bude zameraná na **výpočtové prostredie Matlab**. Túto konzultáciu organizujeme na základe zistení z cvičení v rámci predmetu **B-UIBE**, kde sa ukázalo, že niektorí z vás majú **problémy s orientáciou** v tomto prostredí alebo s programovaním v jazyku Matlab.

Doučovanie sa bude konať v stredu 26.11. o 11:00 v miestnosti CD150. Doučovanie bude viesť môj mladý kolega **Ing. Richard Schwarz**.



5

Skúška

Riadny termín

16. 12. 2025 – CPU
11:00 – 17:00

Opravný termín

26. 1. 2025 v CPU
11:00 – 14:00

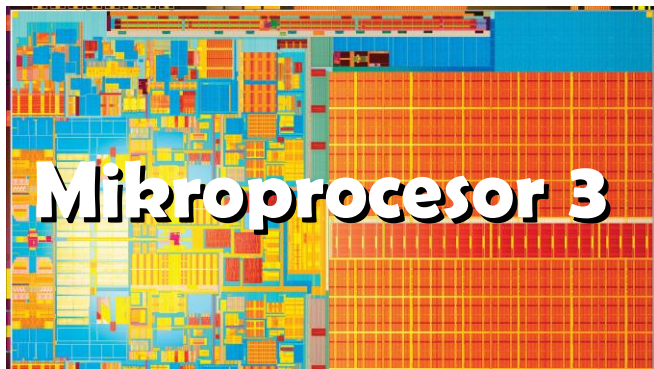


Pred skúškou
pokusná písomka v AIsE

nesprávna odpoveď -1b.

Po skúške
hodnotenie predmetu – dotazník

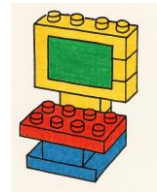
6



7

Dnes:

- Dokončíme vývoj od 8080 po x86_64
- Skoky, podprogramy a prerušenia
- ...pochopiť ďalší jednoduchý program



8

Skokové inštrukcie



A brown hare or European hare (*Lepus europaeus*) (Credit: Fabrice Cahuzat/naturepi.com)



A dog flea (*Ctenocephalides canis*) (Credit: Kim Taylor/naturepi.com)



A desert locust (*Schistocerca gregaria*) (Credit: Ingo Arndt/naturepi.com)

9

JMP

Jump to

JMP addr16 ; preskoč na adresu addr16

Operandy: destination, source **Dĺžka:** 3 Byte **Cykly:** 10

Príznačky: S Z A P C

Popis: Presunie addr16 do registra PC, tým sa presunie vykonávanie programu do inej časti pamäti

Priklad:

START: **MOV A,B** ; nactaj hodnotu z B
 RAR ; posun hodnotu vpravo
 JMP START ; vrat sa na zaciatok

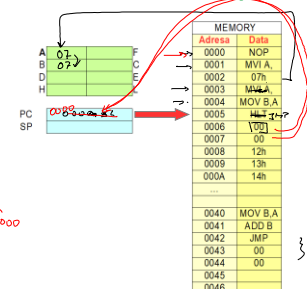
10

JMP

Jump to

START: NOP ✓
 MVI A, 07h ✓
 MOV B,A ✓
 HLT
 ~~JMP 0000~~
 JMP START

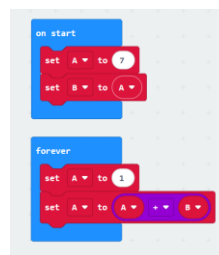
MOV PC, 0000



11

JMP

Jump to



START: MVI A, 07h ✓

MOV B,A ✓

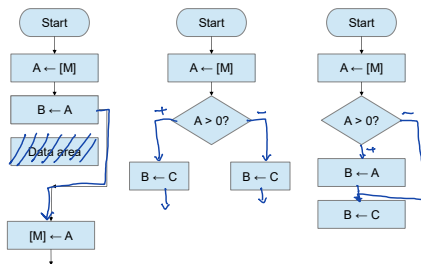
LOOP: MVI A, 01h ✓

ADD B ✓

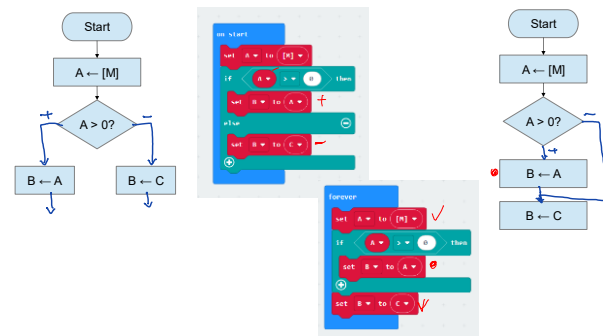
JMP LOOP ✓

12

BRANCH - Vetvenie programu



13



14

JZ, JNZ

Jump if (not) Zero

JNZ addr16	; preskoč na adresu addr16
JZ addr16	; preskoč na adresu addr16

Operandy: destination, source Dĺžka: 1 Byte Cykly: 4(5)

Priznaky: S Z A P C

Popis:

Priklad:

```
SUB B ; odpočítaj B
JZ 01C00h ; ak su rovnake, vysledok je nula a skoc niekam prec
MOV B,A ; vysledok uloz do B, lebo nie su rovnake...
```

15

JC, JNC

Jump if (not) Carry

JNC addr16	; preskoč na adresu addr16
JC addr16	; preskoč na adresu addr16

Operandy: destination, source Dĺžka: 3 Byte Cykly: 10

Priznaky: S Z A P C

Popis: Preskočí na zadanú adresu v prípade, že je podmienka splnená

Priklad:

16

CMP

Compare

CMP reg	; porovnaj A s registrom
CMP addr16	; porovnaj A s obsahom pamäti

Operandy: A, B, C, D, E, H, M Dĺžka: 1 (3) Byte Cykly: 4 (7)

Priznaky: S Z A P C

Popis: Inštrukcia CMP M nastaví F tak, ako keby bol obsah z X odpočítaný od obsahu A. Na rozdiel od SUB však obsah Akumulátora ostáva nezmenený. F je nastavené nasledovne:

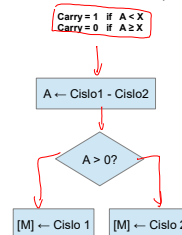
```
Zero = 1 if A = X
Zero = 0 if A ≠ X
Carry = 1 if A < X
Carry = 0 if A ≥ X
```

(A a X chápeme ako neznamienkové binárne čísla)

17

Príklad 4: ktoré z dvoch čísel je väčšie?

V pamäti sú za sebou uložené dve čísla. Nájdi väčšie z nich a ulož ho na ďalšiu pozíciu v pamäti.



```
ORG 0010H
DB 03H,02H,00H
```

```
ORG 0000
LDA 0010h
MOV B,A
LDA 0011h
CMP B
JNC HOTOVO
MOV A,B
STA 0012h
HLT
```

HOTOVO:

18

Príklad 4: ktoré z dvoch čísel je väčšie?

Carry = 1 if $A < X$
Carry = 0 if $A \geq X$

V pamäti sú za sebou uložené dve čísla. Nájdi väčšie z nich a ulož ho na ďalšiu pozíciu v pamäti.

```
ORG 0010H
DB 03H,02H,00H

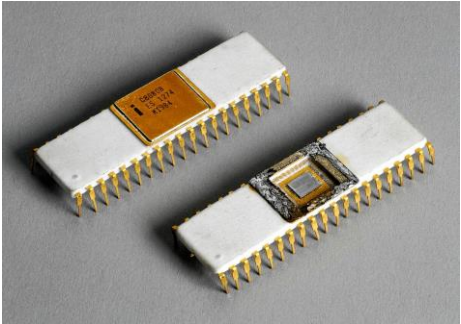
ORG 0000
LDA 0010h ✓
MOV B,A,
LDA 0011h
CMP B
JNC HOTOVO ✗ ✓
MOV A,B,
HOTOVO: STA 0012h ✓
HLT
```

Diagram of the 8080 internal structure showing the ALU, registers (A, B, C, D, E, H, L), and flags (Carry, Zero, Sign, Parity, Overflow). The ALU is performing a subtraction (A - B) and the Zero flag is set (Z=1).

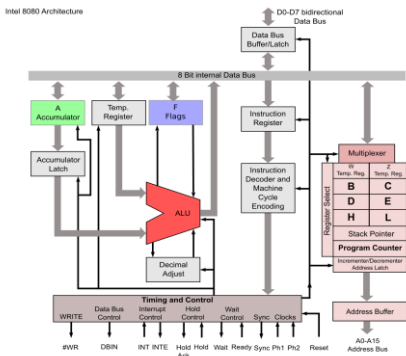
Address	Data
0000	LDA
0001	00
0002	10
0003	MOV B,A
0004	LDA
0005	00
0006	11
0007	CMP B
0008	STA
0009	00
000A	12
000B	HLT
000C	
000D	
000E	
000F	
0010	03
0011	02
0012	00
0013	

19

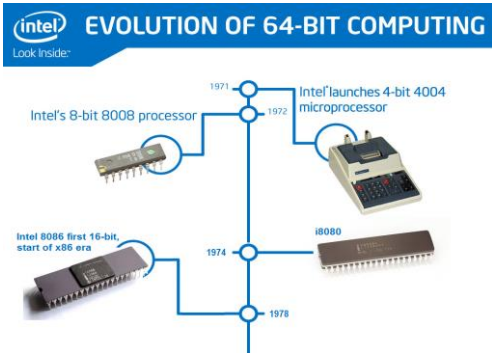
Intel 8080



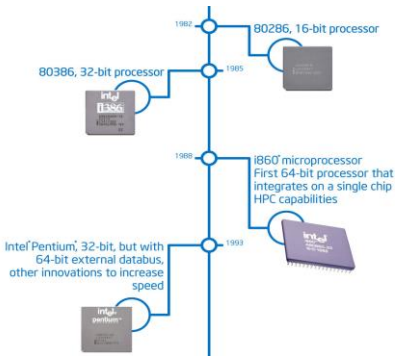
20



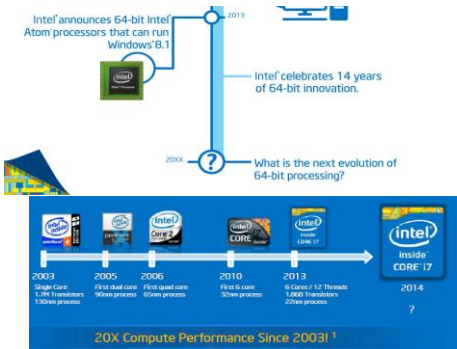
21



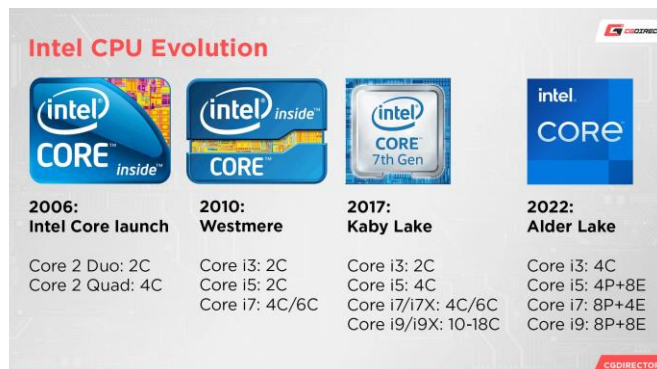
22



23



24



25

Architektúra Intel x86

Intel Instruction Set Architecture (ISA)

=vývoj + spätná kompatibilita

- **8080** (1974) 8-bitový mikroprocesor
Accumulator, 3 registrové páry
- **8086** (1978) 16-bitové rozšírenie 8080
CISC procesor (complex instruction set)
- **8087** (1980) aritmetický (floating-point) koprocessor
Pridané FP inštrukcie a sada registrov
- **80286** (1982) 24-bitové adresovanie, MMU
segmentový prístup k pamäti a jej ochrana
- **80386** (1985) 32-bitové rozšírenie (dnes ozn. IA-32)
pridané režimy adresovania aj inštrukcie
okrem segmentového aj stránkový prístup

26

Architektúra Intel x86

Intel Instruction Set Architecture (ISA)

=ďalší vývoj

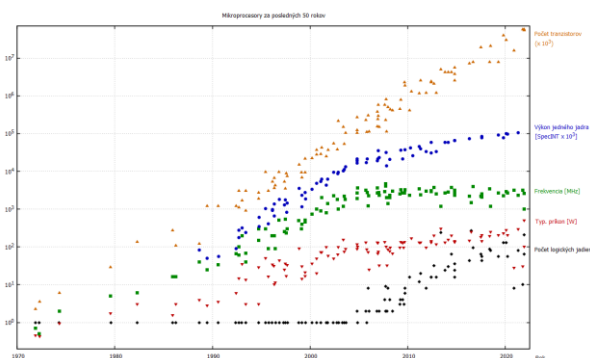
- **i486** (1989) pipeline, cache a FPU priamo na čípe
konkurencia: AMD, Cyrix
- **Pentium** (1993) superskalárny procesor, 64-bit data + nechvalne známa FDIV bug
neskôr pribudli inštrukcie MMX (Multi-Media eXtension)
- **Pentium Pro** (1995), **Pentium II** (1997) -- nová mikroarchitektúra
- **Pentium III** (1999) pribudli inštrukcie SSE (Streaming SIMD Extensions) aj s registrami
- **Pentium 4** (2001) nová mikroarchitektúra, pribudli inštrukcie SSE2

27

Architektúra Intel x86-64

- **AMD64** (2003) rozšírenie architektúry na 64 bitov
- **EM64T** (2004) Extended Memory 64 Technology
Pentium 4 Prescott, Pentium D = Intel prijal AMD64 (s úpravami)
pribudli inštrukcie SSE3
- **Intel Core** (2006) pribudli inštrukcie SSE4, podpora VM (virtual machine)
- **Intel Core i7** (2008) Advanced Vector Extension, dlhšie registre SSE, viac inštrukcií
Core i5 (2009)
Core i3 (2010)
- **Intel Adler Lake** (2021) Hybrid design with performance (Golden Cove)
efficiency cores (Gracemont)
support for PCIe Gen5 and DDR5
updated Gaussian Neural Accelerator (GNA3)

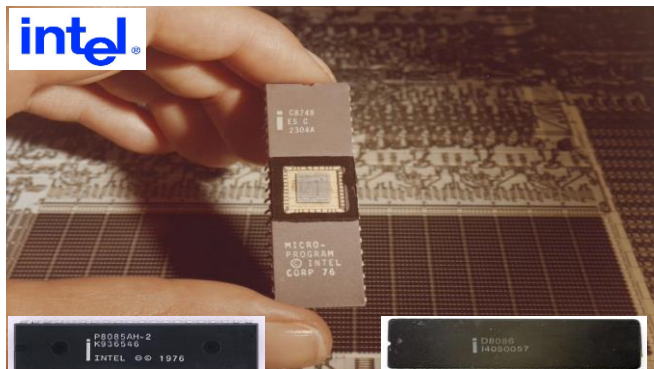
28



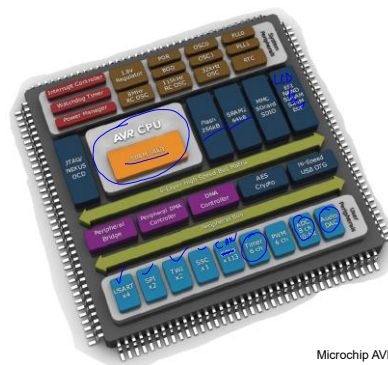
29



30

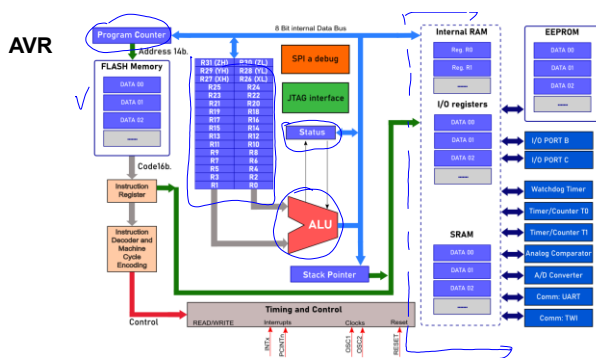


31

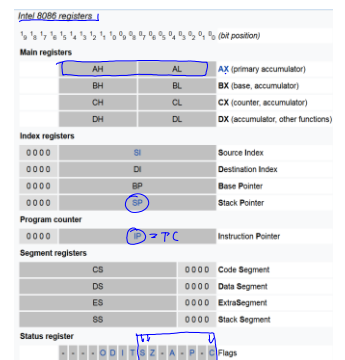
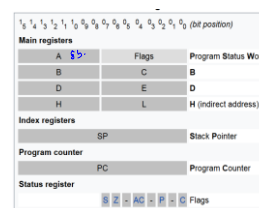


Microchip AVR® 32-bit AVR UC3

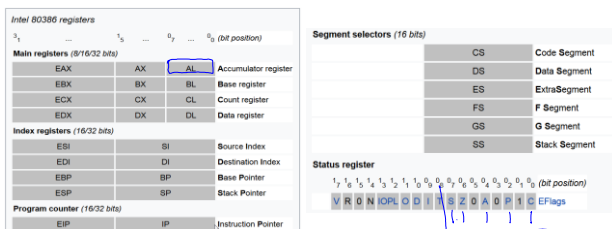
32



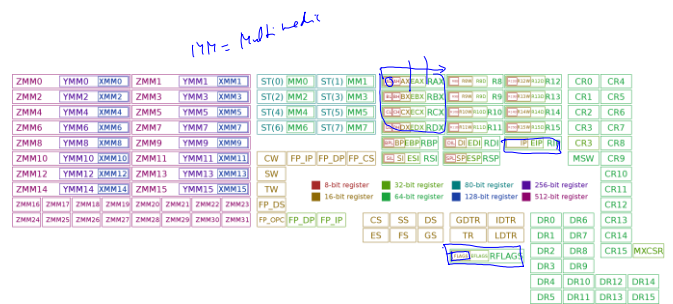
33



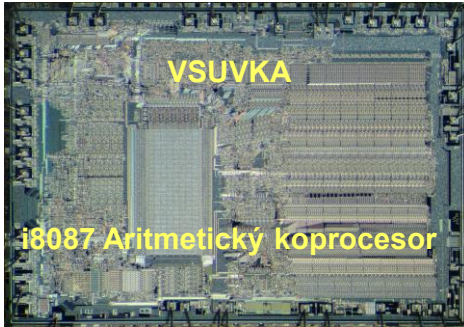
34



35



36

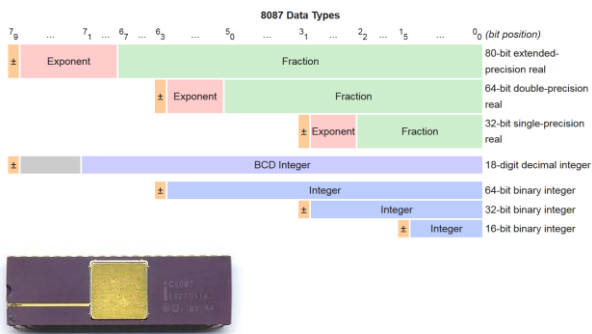


37

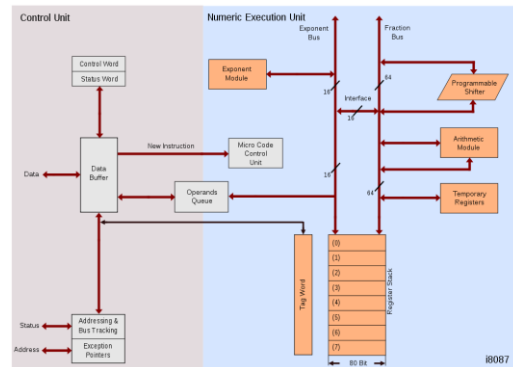
8086 inštrukcie

Data Transfer Instructions	Multi. And Div. Arithmetic Instructions
MOV Move byte or word	MUL Multiply byte or word, unsigned
XCHG Exchange byte or word	DIV Integer divide byte or word, unsigned
XLAT Translate byte	IMUL Multiply byte or word, signed
LEA Load effective address	IDIV Integer divide byte or word, signed
LDS Load pointer using DS	AAM ASCII adjust for multiply
LES Load pointer using ES	AAD ASCII adjust for division
	CBW Convert byte to word
	CWD Convert word to doubleword
Addition Arithmetic Instructions	Logic Instructions
ADD Add byte or word	AND "AND" byte or word
ADC Add byte or word with carry	OR Inclusive "OR" byte or word
INC Increment byte or word by 1	XOR Exclusive "OR" byte or word
AAA ASCII adjust for addition	NOT Complement or negation of byte or word
DAA Decimal adjust for addition	CMP Compare word or byte
Subtraction Arithmetic Instructions	Shift and Rotate Instructions
SUB Subtract byte or word	SAL/SHL Shift arithmetic/logical left byte or word
SBB Subtract byte or word with borrow	SHR Shift logical right byte or word
DEC Decrement byte or word by 1	SAR Shift arithmetic right byte or word
NEG Negate byte or word	ROL Rotate left byte or word
DAS Decimal adjust for subtraction	ROR Rotate right byte or word
AAS ASCII adjust for subtraction	RCL Rotate left through carry, byte or word
CMP Compare operands	

38



39



40

8087 inštrukcie

Class	Instructions
Data Transfer	Load (all data types), Store (all data types), Exchange
Arithmetic	Add, Subtract, Multiply, Divide, Subtract Reversed, Divide Reversed, Square Root, Scale, Remainder, Integer Part, Change Sign, Absolute Value, Extract
Comparison	Compare, Examine, Test
Transcendental	Tangent, Arc tangent, 2^x-1 , $Y*\text{Log}_2(X+1)$, $Y*\text{Log}_2(X)$
Constants	.0, 1, π , $\text{Log}_{10}(2)$, $\text{Log}_e(2)$, $\text{Log}_2(10)$, $\text{Log}_2(e)$
Processor Control	Load Control Word, Store Control Word, Store Status Word, Load Environment, Store Environment, Save, Restore, Enable Interrupts, Disable Interrupts, Clear Exceptions, Initialize

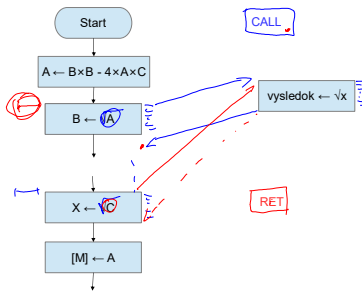
Kompilátor musí vedieť, pre aký počítač prekladá, ak nemá FPU, tak je potrebné chýbajúce inštrukcie nahradiť príslušným podprogramom - emulácia 8087.

41

8087 Emulator Speed Comparison			
Approximate Execution Time in μs @ 5 MHz Clock			
Instruction	8087	8086 Emulation	
Multiply (single precision)	19	1 600	✓
Multiply (double precision)	27	2 100	
Add	17	1 600	
Divide (single precision)	39	3 200	
Compare	9	1 300	
Load (single precision)	9	1 700	
Store (single precision)	18	1 200	
Square root	36	19 600	
Tangent	90	13 000	
Exponentiation	100	17 100	

42

Podprogramy



43

CALL

Call

CALL addr16 ; zavolaj funkciu na adrese addr16

Operandy: Dĺžka: 3 Byte Cykly: 10+7

Priznaky: S Z A P C

Popis: Presunie addr16 do registra PC, tým sa presunie vykonávanie programu do inej časti pamäti. Navyše oproti JMP odloží do **zásobníka** návratovú adresu.

Příklad:

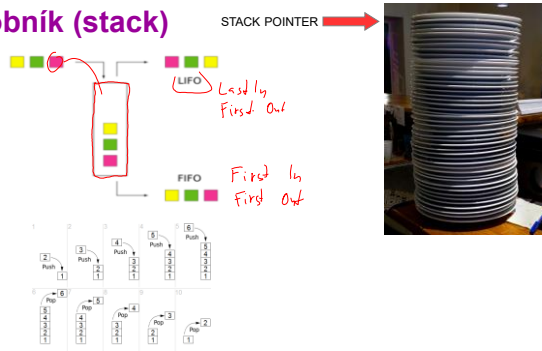
START: **MOV A,B** ; načítaj hodnotu z B
CALL Odmocni ; choď vypočítat odmocninu

RET

Return

44

Zásobník (stack)



45

Zásobník (stack)

Inštrukcie PUSH a POP

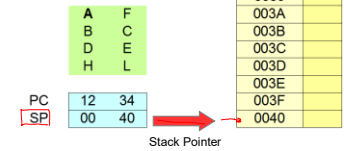
Ukladajú sa vždy dva bajty (adresa alebo RP)

PUSH HL

SP ← SP - 1
[[SP]] ← reg. H
SP ← SP - 1
[[SP]] ← reg. L

POP BC

Reg. B ← [[SP]]
SP ← SP + 1
Reg. C ← [[SP]]
SP ← SP + 1



46

Zásobník (stack)

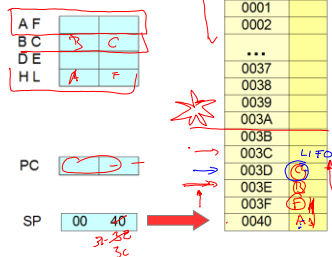
```
.ORG 0
MVI A,011h
LXI B,2233h
LXI D,4455h
LXI H,6677h
LXI SP,0040h

;ukladame do zasobnika:
PUSH PSW ;PSW = A + F
PUSH B
PUSH D
PUSH H

;robime nieco ine:
MVI A, 3
MVI B, 4
ADD B

;vyberame zo zasobnika:
POP B
POP D
POP H
POP PSW

HLT
```



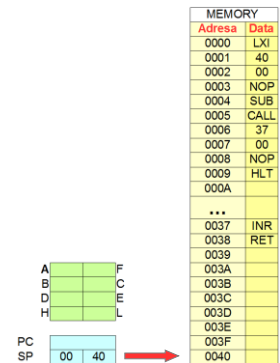
47

CALL addr

((SP) - 1) ← PCH
((SP) - 2) ← PCL
(SP) ← (SP) - 2
(PC) ← addr

```
0000 31 40 00 LXI SP,0040h
0003 00 NOP
0004 97 SUB A
0005 CD 37 00 CALL 0037h
0008 00 NOP
0009 76 HLT

0037 3C INR A
0038 C9 RET
```



48



49

Prerušená (interrupt)



50

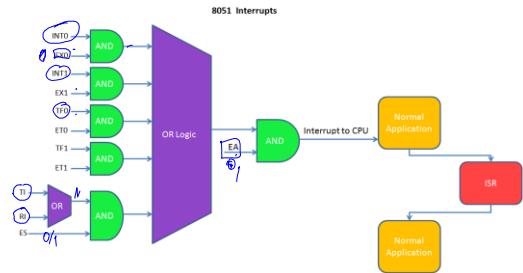
Prerušená – udalosti

- zakázat'
povolit'
maskovat'

EI
DI
- HW:
 - vstup z klávesnice
 - prišla pošta (znak)
 - uplynul čas
 - pomalá periféria je hotová
 - SW:
 - delenie nulou

51

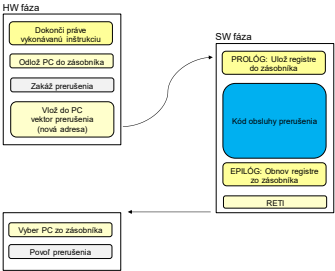
Prerušená – viacdrojové, prioritá



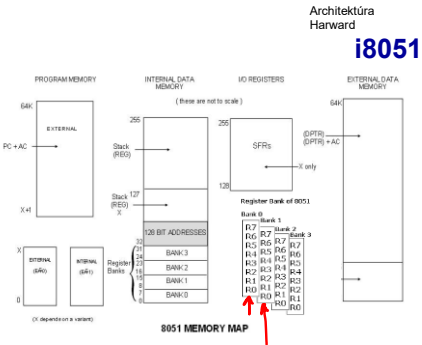
52

MEMORY	
Adresa	Data
0000	LDA
0001	10
0002	00
0003	MOV
0004	LDA
0005	11
0006	00
0007	ADD
0008	STA
0009	12
000A	00
000B	
000C	
000D	
000E	
000F	
0010	02
0011	03
0012	00
0013	

Prerušená (interrupt)

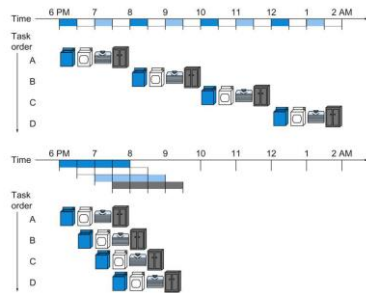


53



54

Pipelining (reťazenie)



Zhrnutie

- Vývoj 8080 až iCore (x86-64)
- Inštrukcie skokov a volania podprogramov
JMP, CALL, RET
- Podmienené skoky
CMP, JZ, JNZ, JC, JNC
- Prerušenie (interrupt)
- Zásobník (stack) – FIFO, LIFO

55

56

Chcem vedieť viac...



David PATTERSON a John HENNESSY:
Computer Organization and Design. ARM Edition.
1st Ed. Morgan Kaufmann, 2016. ISBN: 9780128017333



Jean-Michel BERNARD, Jean HUGON a Robert Le CORVEC:
Od logických obvodů k mikroprocesorům.
SNTL : Praha, 1982



Richard Balogh: **Vybrané kapitoly z mikropočítačů**
I. část: Hardware...
STU, Bratislava, 2023. Skriptá dostupné v knižnici FEI a v AIS



<https://senzor.robotika.sk>

57



58