

1. bezprostřední	$X \rightarrow S$	LNA X
- hodnota operandu přímo do S		
2. přímá	$\langle X \rangle \rightarrow S$	LDA X
- X je adresa v HP, obsah adresy X se uloží do S		
3. nepřímá	$\langle \langle X \rangle \rangle \rightarrow S$	LDI X
- X je adresa, na které je uložena adresa operandu = obsah obsahu X se uloží do S		
4. přímá adresace registrem	$\langle R \rangle \rightarrow S$	LDA R
- obsah registru R se uloží do S		
5. nepřímá adresace registrem	$\langle \langle R \rangle \rangle \rightarrow S$	LDI R
- v registru R je adresa v HP – obsah adresy se uloží do S		
6. adresace s posuvem	$\langle X + \langle R \rangle \rangle \rightarrow S$	
- k obsahu R se přičte X, výsledkem je adresa jejíž obsah se uloží do S		
a) <b>relativní</b>	R = CI (čítač instrukcí) – adresa následující instrukce X = číslo v doplňkovém kódu - + - posun vzhledem k R	
b) <b>bázová (segmentová)</b>	R = adresa v HP (báze, segment) X = kladný posuv (číslo)	
c) <b>indexová</b>	X = adresa v HP R = kladný posuv (obsah registru) – obvykle index $\langle X + \langle IND \rangle \rangle \rightarrow S$	LDA X[IND]

**Př. 6.2:**

Výpis obsahu paměti:

obsah IND = 20

Adresa	Obsah
100	140
110	50
120	180
130	90
140	45
150	140
160	30
170	150
180	60

Provedeme následující instrukce. Jaký bude obsah střadače (S) ?

- a) LNA 150                      150 **à** S                      S = 150  
 b) LDA 150                      <150> **à** S                      S = 140  
 c) LDI 150                      <<150>> = <140> **à** S                      S = 45  
 d) LNA 100 [IND]              100 + <IND> = 100 + 20 **à** S                      S = 120  
 e) LDA 100 [IND]              <100 + <IND>> = <100+20> **à** S                      S = 180  
 f) LDI 100 [IND]              <<100 + <IND>> = <<120>> = <180> **à** S                      S = 60

**6.3. Strojové instrukce**

- |                                      |                         |
|--------------------------------------|-------------------------|
| 1. instrukce pro přenos dat          | LNA, LDA, LDI, STA, STI |
| 2. aritmetické instrukce             | ADD, SUB, MLP, MOD, DIV |
| 3. logické instrukce                 | AND, OR, XOR            |
| 4. instrukce posuvu a rotace         | SHR, SHL, RSHR, RSHL    |
| 5. instrukce (ne)podmíněného skoku   | JMP, JZ, JP, JZ         |
| 6. instrukce pro podporu cyklů       | LOOP                    |
| 7. instrukce pro podporu podprogramů | JSR, RET                |
| 8. instrukce pro práci se zásobníkem | PUSH, POP               |
| 9. instrukce pro I/O operace         |                         |
| 10. systémové instrukce              | HLT                     |

**6.3.1. Logické instrukce**

P	Q	non P	P and Q	P or Q	P xor Q	P eq Q
0	0	1	0	0	0	1
0	1	1	0	1	1	0
1	0	0	0	1	1	0
1	1	0	1	1	0	1

**Př. 6.3:**

Mějme dva registry A a B, kde

A = 0 1 0 0 1 1 0 1      (4D)<sub>16</sub>B = 1 1 0 0 0 1 0 1      (C5)<sub>16</sub>

a instrukce:

AND A, B                      <A> AND <B> **à** B                      B = 0 1 0 0 0 1 0 1      ()  
 OR A, B                      <A> OR <B> **à** B                      B = 1 1 0 0 1 1 0 1      ()  
 XOR A, B                      <A> XOR <B> **à** B                      B = 1 0 1 1 0 0 1 0      ()

B = 1 1 1 1 1 1 1 1

XOR A, B                      B = 1 0 1 1 0 0 1 0                      **à**      NON A**Př. 6.4:**

Registry mají následující obsah: A = FE – 11111110

B = 5D – 01011101

C = A1 – 10100001

$$D = 0E - 00001110$$

Jaký bude obsah registrů po provedení těchto instrukcí za sebou ?

XOR A, B	A = 11111110 = FE	B = 10100011
AND B, C	B = 10100011 = A3	C = 10100001
OR C, D	C = 10100001 = A1	D = 10101111 = AF

### 6.3.2. Obecná pravidla

x AND 0 → 0	x OR 0 → x	x XOR 0 → x
x AND 1 → x	x OR 1 → 1	x XOR 1 → negace x

### Pr. 6.5:

Uvažujme, že jsou k dispozici instrukce z př.6.3 (AND A,B; OR A,B; XOR A,B), dále máme 2B registry A, B a konstanty m, n (nazývané masky). Navrhněte obsah masek m a n a sled instrukcí tak, aby:

Pozn.: Instrukce je možné použít i ve formě: INSTR m, A      <m> INSTR <A> → A

#### a) v registru A:

- vynulování bitu číslo (číslováno zleva počínaje 0): 1, 5, 12
- ostatní zůstanou zachovány

Řešení: m = 101110111110111  
AND m, A

#### b) v registru B:

- bity č. 0, 3, 6 nastavit na 1
- bity č. 8 – 15 znegovat

Řešení: m = 1001001000000000  
n = 0000000011111111  
OR m, B  
XOR n, B

#### c) nahrazení nultého bytu registru A za nultým bytem registru B (původní hodnoty A, B není třeba zachovat)

Řešení: m = 0000000011111111  
n = 1111111100000000  
AND m, A  
AND n, B  
OR A,B

#### d) vzájemná záměna obsahů registrů A a B – bez pomoci dalšího pomocného registru!

Řešení: XOR A, B      nebo      XOR B, A  
XOR B, A      XOR A, B  
XOR A, B      XOR B, A

## Strojový kód procesoru

<b>ADD x</b>	$\langle S \rangle + \langle x \rangle \rightarrow S$
<b>AND x</b>	$\langle S \rangle \text{ AND } \langle x \rangle \rightarrow S$
<b>DIV x, y</b>	$\langle x \rangle \text{ DIV } \langle y \rangle \rightarrow S$
<b>HLT</b>	zastavení procesoru
<b>JMP x</b>	$x \rightarrow \text{CI}$
<b>JN x</b>	je-li $\langle S \rangle < 0$ , pak $x \rightarrow \text{CI}$
<b>JP x</b>	je-li $\langle S \rangle > 0$ , pak $x \rightarrow \text{CI}$
<b>JSR x</b>	$\langle \text{SP} \rangle - 1 \rightarrow \text{SP}$ , $\langle \text{CI} \rangle \rightarrow \langle \text{SP} \rangle$ , $x \rightarrow \text{CI}$
<b>JZ x</b>	je-li $\langle S \rangle = 0$ , pak $x \rightarrow \text{CI}$
<b>LDA x</b>	$\langle x \rangle \rightarrow S$
<b>LDA x [IND]</b>	$\langle x + \langle \text{IND} \rangle \rangle \rightarrow S$
<b>LDI x</b>	$\langle \langle x \rangle \rangle \rightarrow S$
<b>LDI x [IND]</b>	$\langle \langle x + \langle \text{IND} \rangle \rangle \rangle \rightarrow S$
<b>LNA x</b>	$x \rightarrow S$
<b>LNA x [IND]</b>	$x + \langle \text{IND} \rangle \rightarrow S$
<b>LOOP x</b>	$\langle \text{IND} \rangle - 1 \rightarrow \text{IND}$ , je-li $\langle \text{IND} \rangle > 0$ , pak $x \rightarrow \text{CI}$
<b>MLP x</b>	$\langle S \rangle * \langle x \rangle \rightarrow S$
<b>MOD x, y</b>	$\langle x \rangle \text{ MOD } \langle y \rangle \rightarrow S$
<b>OR x</b>	$\langle S \rangle \text{ OR } \langle x \rangle \rightarrow S$
<b>POP x</b>	$\langle \langle \text{SP} \rangle \rangle \rightarrow x$ , $\langle \text{SP} \rangle + 1 \rightarrow \text{SP}$
<b>PUSH x</b>	$\langle \text{SP} \rangle - 1 \rightarrow \text{SP}$ , $\langle x \rangle \rightarrow \langle \text{SP} \rangle$
<b>RET</b>	$\langle \langle \text{SP} \rangle \rangle \rightarrow \text{CI}$ , $\langle \text{SP} \rangle + 1 \rightarrow \text{SP}$
<b>RSHL R, n</b>	rotace $\langle R \rangle$ o $n$ bitů vlevo
<b>RSHR R, n</b>	rotace $\langle R \rangle$ o $n$ bitů vpravo
<b>SHL R, n</b>	posun $\langle R \rangle$ o $n$ bitů vlevo
<b>SHR R, n</b>	posun $\langle R \rangle$ o $n$ bitů vpravo
<b>STA x</b>	$\langle S \rangle \rightarrow x$
<b>STA x [IND]</b>	$\langle S \rangle \rightarrow x + \langle \text{IND} \rangle$
<b>STI</b>	$\langle S \rangle \rightarrow \text{IND}$
<b>SUB x</b>	$\langle S \rangle - \langle x \rangle \rightarrow S$
<b>XOR x</b>	$\langle S \rangle \text{ XOR } \langle x \rangle \rightarrow S$