

Prednáška 2

Procesor i8086 z pohľadu programátora

Príklady programov v
Jazyku Symbolických Inštrukcií

Vyšší programovací jazyk

JSI

MOV

A,20

ASSEMBLER

SW

3

E

2

0

VIRTUAL

HW

0011

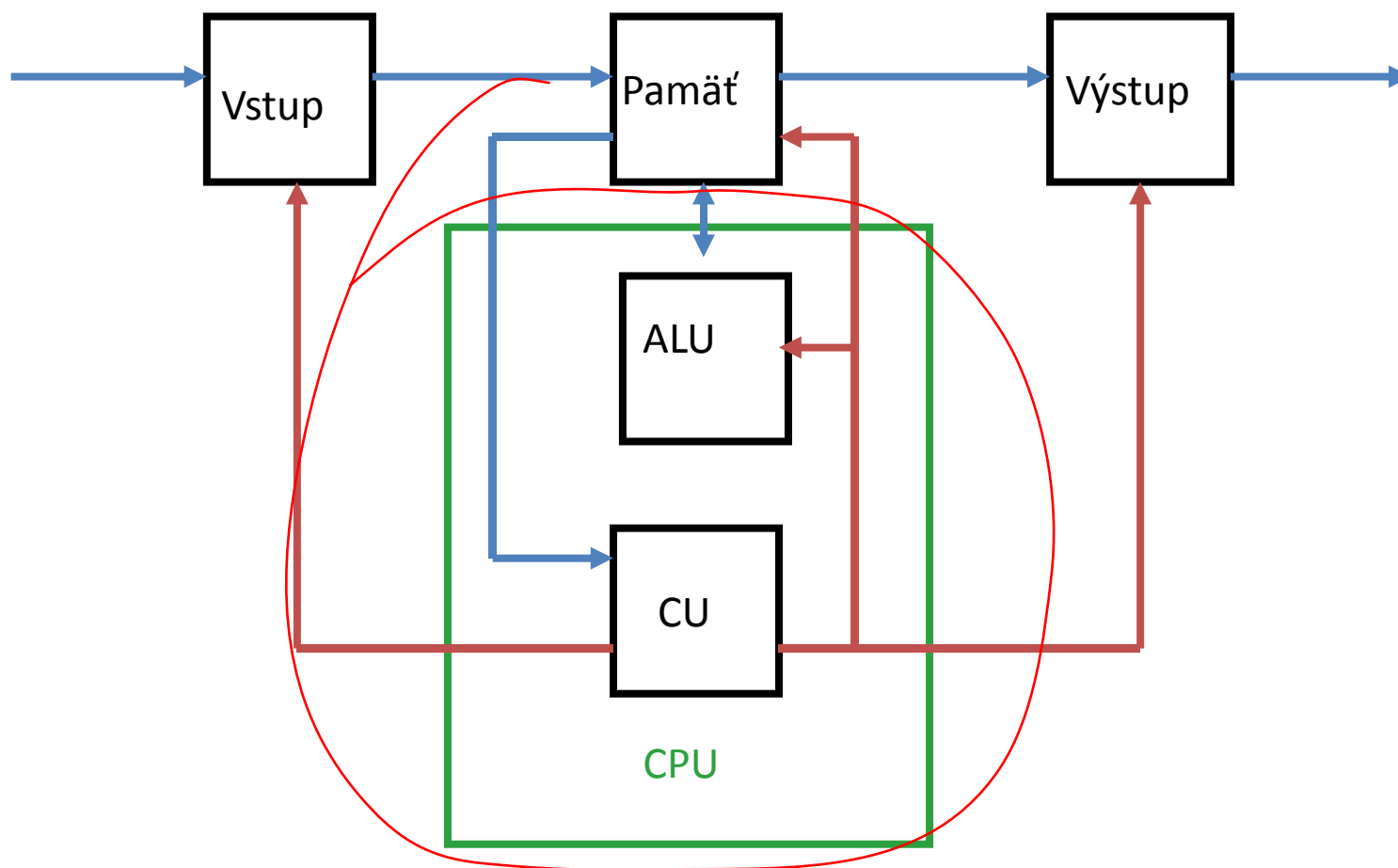
1110

0010

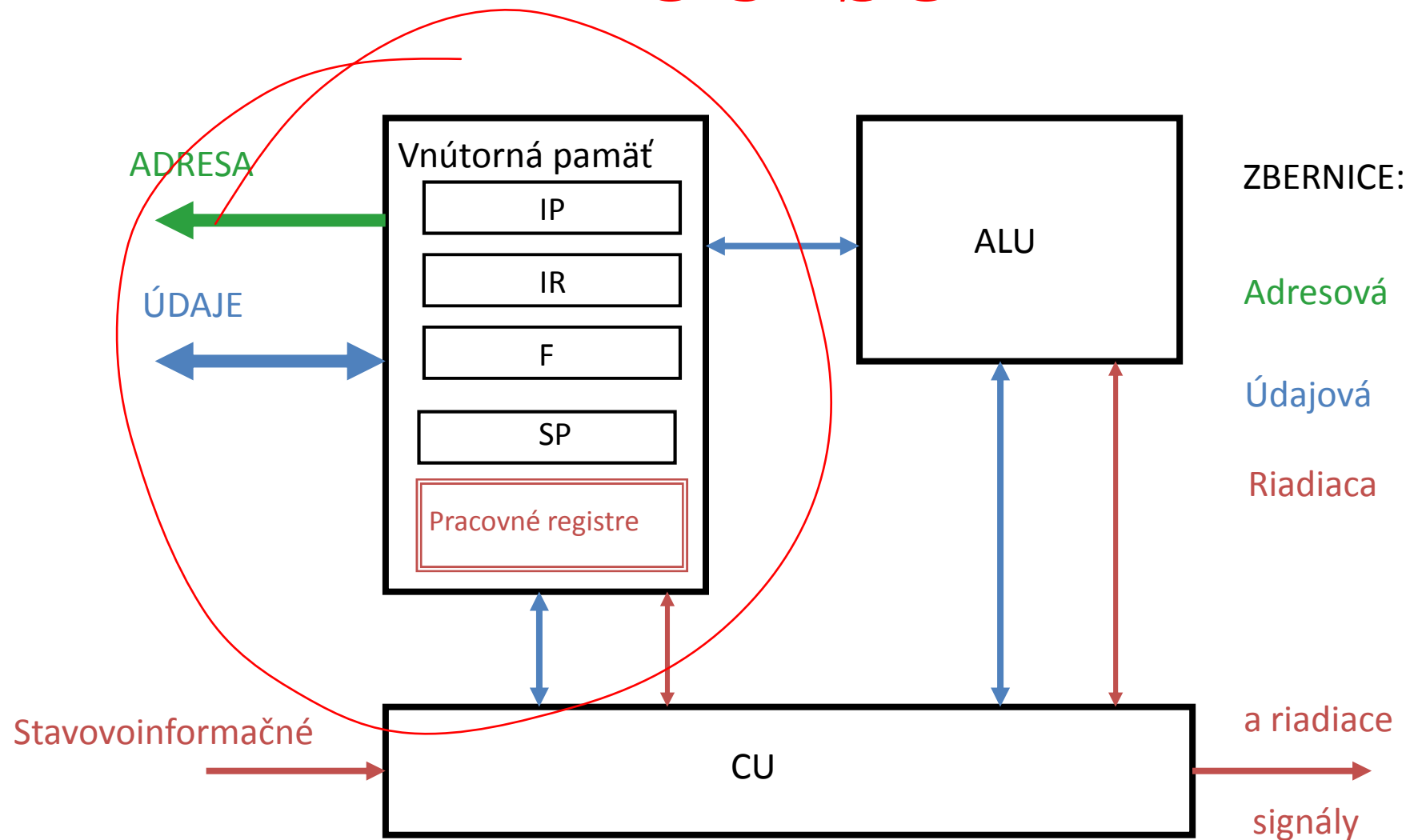
0000

REAL

POČÍTAČ



PROCESOR



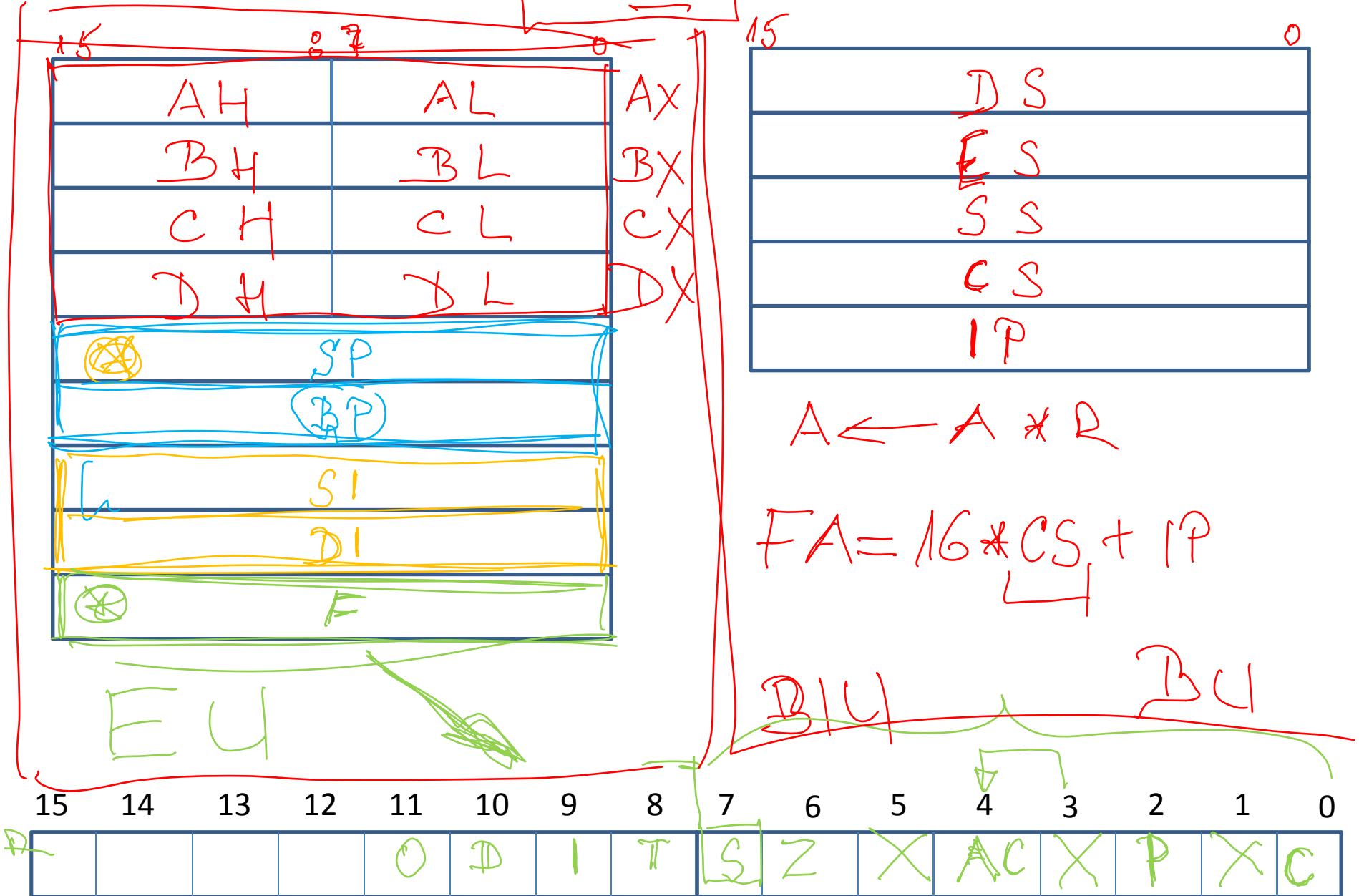
i 8086

15	8	7	0	
AH		AL		AX
BH		BL		BX
CH		CL		CX
DH		DL		DX
SP				
BP				
SI				
DI				
F				

15	0
DS	
ES	
SS	
CS	
IP	

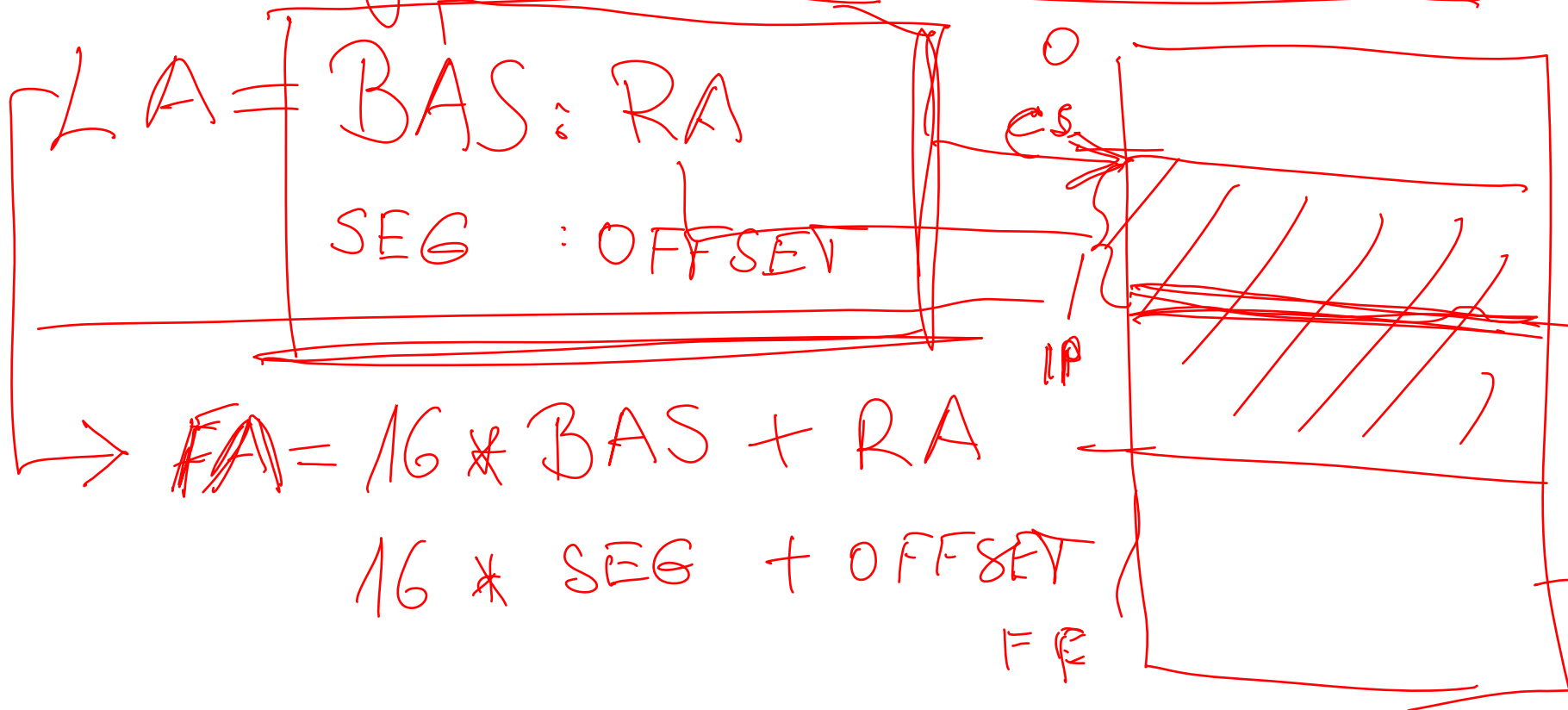
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

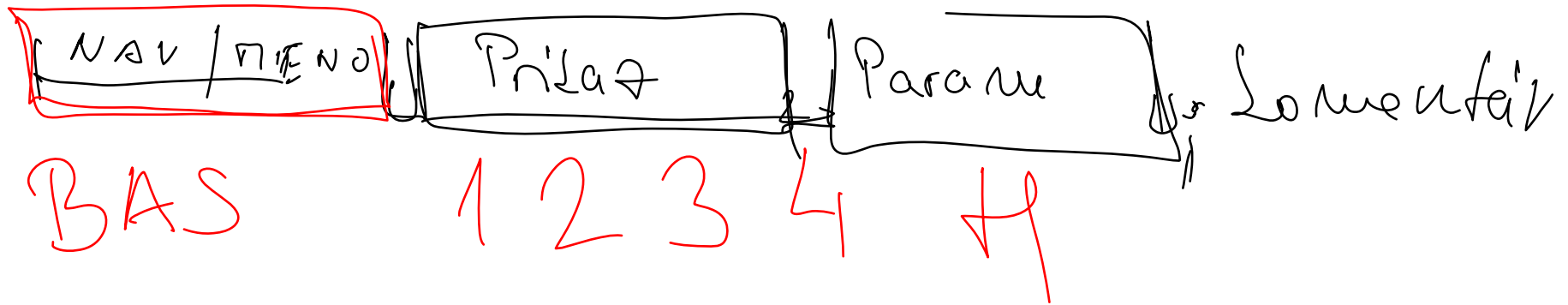
i 8086



PC (IP) \rightarrow ADDR. \rightarrow Effective adr. FA

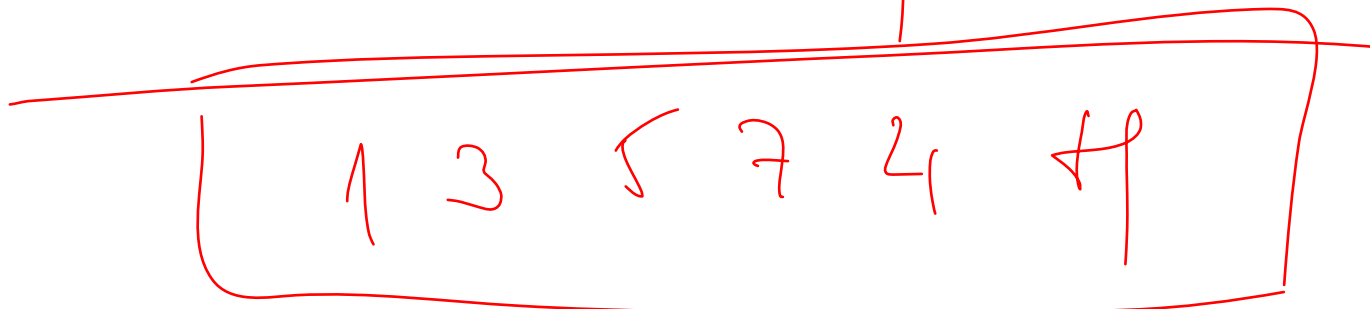
Virtualy prístav — SEGMENT





OFFSET

1 2 3 4



;PROGRAM VYPISE SLOVO 'AHOJ'

INST ciele, zdroj

ZAS SEGMENT STACK
DW 64DUP(?) ;vyhradenie poloziek pre zasobnik
ZAS ENDS

DATA SEGMENT
TEXT DB 'AHOJ\$' ;retazec znakov na zobrazenie
DATA ENDS

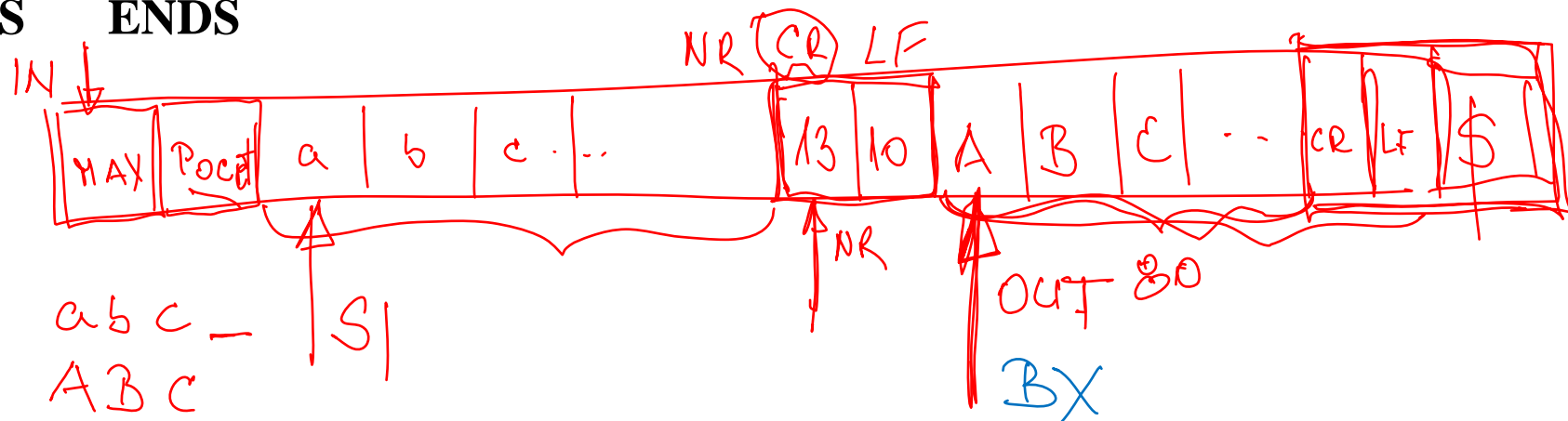
*↑
→ FA = 16 * DS + DX*

CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:~~STACK~~ ZAS

START MOV AX,SEG DATA ;inicializacia
MOV DS,AX ;DS segmentu
MOV DX,OFFSET TEXT ;v DX je offset premennej TEXT
MOV AH,09H ;funkcia 'write string'
INT 21H ;vyvolanie sluzby MS-DOS
MOV AX,4C00*H* ;ukoncenie programu
INT 21H ;odovzдание riadenie riadiacemu prg.
CODE ENDS
END START

;program nacita retazec znakov z klavesnice a vypise tento retazec, v ktorom
 ;su konvertovane male pismena na velke

```
ZAS    SEGMENT    STACK
        DW          64DUP(?)    ;vyhradenie poloziek pre zasobnik
ZAS    ENDS
```



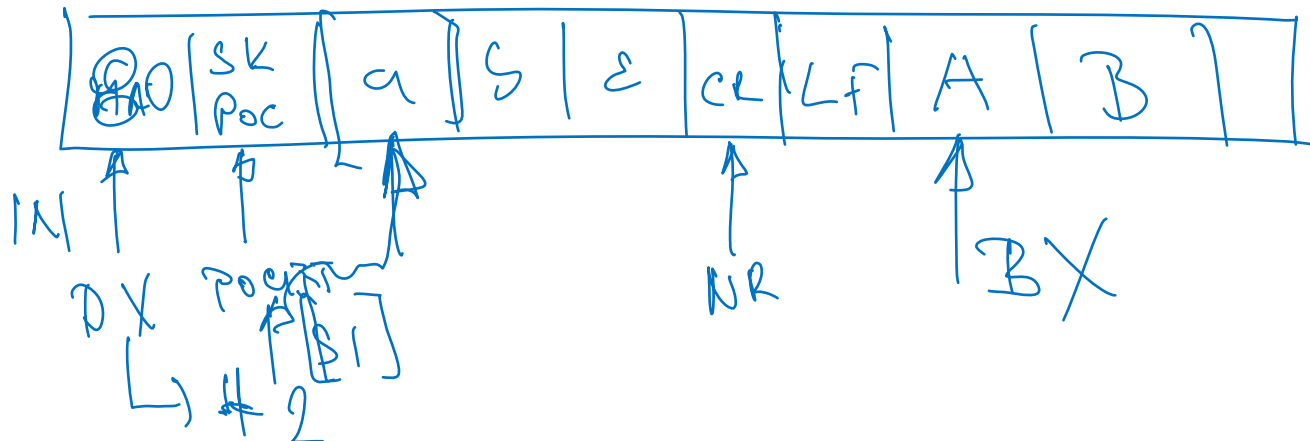
```
DATA    SEGMENT
IN      DB          80          ;max.pocet znakov (=50H)
POCET   DB          0          ;pocet skutocne nacistanych znakov
        DB          80DUP(?)    ;vstupny retazec
NR      DB          13         ;'novy riadok' (=0DH, 'CR')
        DB          10         ;'zaciatok riadku' (=0AH, 'LF')
OUT     DB          83DUP(?)    ;vystupny retazec
DATA    ENDS
```

CODE SEGMENT

ASSUME

CS:CODE, DS:DATA, SS:~~STACK~~ ZAS

START	MOV	AX,SEG DATA	;inicializacia
	MOV	DS,AX	;DS segmentu
	MOV	DX,OFFSET IN	;v DX je offset vstupneho retazca
	MOV	AH,0AH	;funkcia 'read string'
	INT	21H	;vyvolanie sluzby MS DOS-u
	MOV	BX,OFFSET OUT	;v BX je offset vystupneho retazca
	ADD	DX,2	;
	MOV	SI,DX	;SI ukazuje na zaciatok vstupneho retazca
	XOR	CX,CX	;nulovanie reg. CX
	XOR	AX,AX	;nulovanie reg. AX
	MOV	CL,POCET	;nastavenie CX na skutocny pocet znakov



ZNOVA	MOV	AL,[SI]	;v AL je kod nacistaneho znaku
	CMP	AL,60H	;kody nad 60H su male pismena
	JBE	DALEJ	;skok ak v AL nebol kod maleho pismena
	CMP	AL,7BH	;kody nad 7BH su 'ine' znaky
	JNB	DALEJ	;skok ak v AL nebol kod maleho pismena
	SUB	AL,20H	;konverzia maleho na velke
DALEJ	MOV	[BX],AL	;presun do vystupneho retazca
	INC	SI	
	INC	BX	
	LOOPNE	ZNOVA	

A: 41H 0100 0001
 a: 61H 0110 0001
 0010 0000

TLAC	MOV	BYTE PTR [BX],13	
	MOV	BYTE PTR [BX]+1,10	;prechod na novy riadok
	MOV	BYTE PTR [BX]+2,'\$'	;ukoncenie vystupneho retazca
	MOV	DX,OFFSET NR	;v DX je offset premennej NR
	MOV	AH,09H	;funkcia 'write string'
	INT	21H	;vyvolanie sluzby MS-DOS
	MOV	AX,4C00H	;ukoncenie programu
	INT	21H	;odovzdanie riadenie riadiacemu prg.
CODE	ENDS		
	END	START	