

Pavel Náplava

naplava@fel.was.cz

naplava@email.cz

11. týden test 20 b.

pozdní odevzotání

dom. práce 10 b. -5 b.

lab. práce 10 b. -5 b.

aktivita až 5 b.

sápočet až po 25 bodů

<http://pitbul.wz.cz>

Číselné soustavy

základ	použité znaky
$Z = 10$	0...9
$= 2$	0...1
$= 16$	0...9, A-F

$$\begin{array}{ccccccc} & 1 & 3 & 9 & , & 1 & 2 \\ & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ z^2 & z^1 & z^0 & , & z^{-1} & z^{-2} \end{array}$$

$$\Rightarrow 1 \cdot 10^2 + 3 \cdot 10^1 + 9 \cdot 10^0 + 1 \cdot 10^{-1} + 2 \cdot 10^{-2}$$

100_2
základ

→ dělení
→ násobení

$$139 : 2 = 69 \dots 1$$

$$69 : 2 = 34 \dots 1$$

$$34 : 2 = 17 \dots 0$$

$$17 : 2 = 8 \dots 1$$

$$8 : 2 = 4 \dots 0$$

$$4 : 2 = 2 \dots 0$$

$$2 : 2 = 1 \dots 0$$

$$1 : 2 = 0 \dots 1$$

$$139 \Rightarrow 1_2 \cdot 1010_2^{10} + 11_2 \cdot 1010_2^1 + 1001_2 \cdot 1010_2^0$$

příbuzné soustavy

$$R_1 = R_2^m$$

$$\begin{array}{c} \leftarrow \quad \rightarrow \\ \hline 0101, 110100 \\ \hline 0 \quad 5, 6 \quad 4_8 \end{array}$$

$8 = 2^3$
 $16 = 4^2$

Aritmetické operace

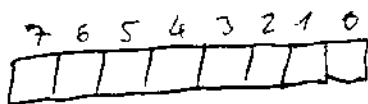
$$\begin{array}{r} 159 \\ - 65 \\ \hline 224 \end{array} \quad \begin{array}{l} / 2 \\ 112 \\ \hline 56 \end{array}$$

$$\begin{array}{r} 133 \\ - 69 \\ \hline 64 \end{array} \quad \begin{array}{l} 133 - 69 = 64 \\ 133 - 69 = 64 \end{array}$$

$$\begin{array}{r} 159 \\ - 13 \\ \hline 146 \end{array}$$

$$635 : 7 = 91,42857142857143$$

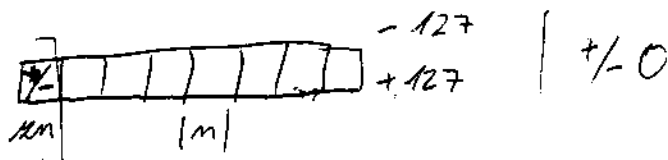
Řádová mřížka



$$Z = 2^{m+1} = 256$$

$$\epsilon = 1$$

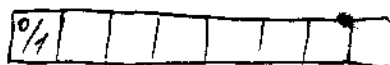
modul. ř. mřížky, udává uložitelný rozsah



0/1

problém s operacemi

doplňkový kód



+/-

$$C \geq 0 \Rightarrow C$$

$$00000100_2 \Rightarrow 00000100_2$$

$$C < 0 \Rightarrow K - C$$

$$2^8 = 100000000_2$$

$$- 00000100_2$$

$$Z + 1$$

$$\begin{array}{r} 11111011_2 \\ + 1 \\ \hline \end{array}$$

$$11111100_2$$

posuv

1 1 0 0 1 1 0 1

aritm. 1 1 1 0 0 1 1 0

Příklad

8 bit mřížka

16 soustava

$7E_{16}$		→ kladná
$E9_{16}$		→ první kód
		→ doplnkový kód

součet
znaménko
přetčení

$$7E_{16} = 0111,1110_2$$

$$E9_{16} = 1110,1001_2$$

Kladná

$$\begin{array}{r} 01111110 \\ 11101001 \\ \hline 101100111 \end{array}$$

p. kód

$$\begin{array}{r} 01111110 \\ 11101001 \\ \hline 00010101 \end{array}$$

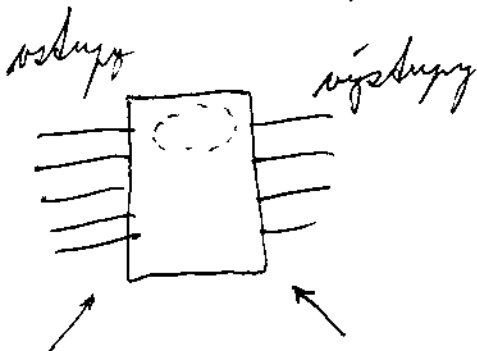
$$\begin{array}{r} 01111110 \\ 11101001 \\ \hline 101100111 \end{array}$$

predekla

Processor DOP

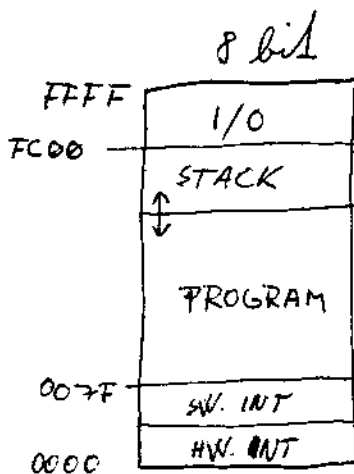
Daněčkův originální procesor

(Daněčkovy opilecké poselství)

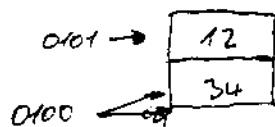


addr. bus $\rightarrow$ 16 bit
data bus
control bus

Paměť



little endian

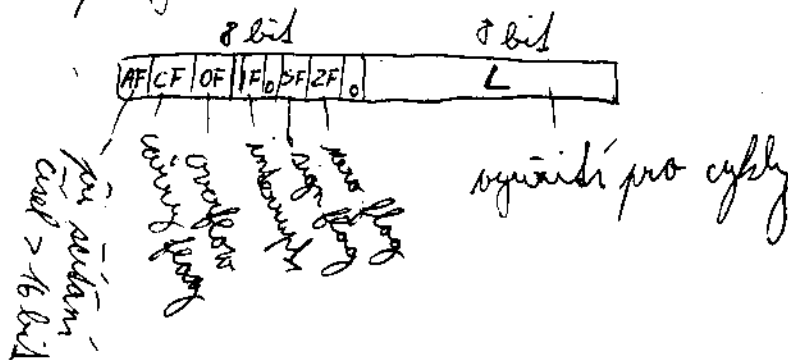


1234h

Registry

- označení

- PC - program counter
- SP - stack pointer
- S - source
- D - destination
- W - work -> strádát
- PSW - program status word



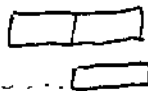
- shrnutí DI, DO, U, T

✓
pomocné
pro přenos
do / ven

✓
pomocné reg

Instrukce

16b
8b



ADD - pracuje s 16 bity

ADDB - čte 8 bit a do horních bitů dá 0

ADDs - sign, čte 8 bitů a doplněje nejvyšší bit kvůli zachování znaménka

nastavují AF

parametry konst

D, S, SP

přímá hodnota

[D], [S], [SP]

ne přímo - adresa do paměti

[D], [S+], [SP-]

[S-]

zvětšuje adresu o 1/2 byty podle typu operandu

LD S, konst

MOV W, S

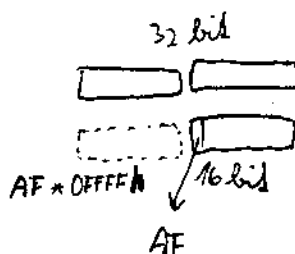
- přesun nemění přímky

ADD S (W := W + S)

SUB S W := W - S

ADD XAF

SUB XAF



JMP addr

Adresa: instr1.
instr2
:
jmp adresa

JC addr carry = 1
JNC addr carry = 0

DEC } L = L - 1 - stop if L = 0
DJNZ } - stop if L ≠ 0

Prefixy

[SWW
ADD S] - neuloží výsledek operace

[SWW
SUB S] compare
 SUB S
 W-S nastaví se pouze přenosy

[UCF
ADD S] 10 ADD se použije ještě CF
 W+S+CF
 při operacích s čísly > 16 bit

Prüfung 1, 2, 5

1 LD W, [D]

W = ~~FF01~~ h

ADD [S+]

[S+] = 3482 h

12493 h

	SP	S	D	W	AF	ST	CF	ZF	OF
	2000	1FFC	1FF7	1234	0	0	0	0	0
LD W, [D]				F011					
ADD [S+]		1FFE		2493	0	0	1	0	0
ST W			1FF9						
LD W, [D]				1032					
UCP ADD [S+]				1F43	0	0	0	0	0
ST W			1FFA						

		ST W	ST W
2000	AB		
	OF		
1FFE	10		
	34		
1FFC	82		
	FF		
	10		1F
1FF9	32		43
	F0	24	
1FF7	11	93	

$$\begin{array}{r}
 1032 \\
 [S+] = \text{OF } 10 \\
 \quad \quad 1 \\
 \hline
 1F43
 \end{array}$$

Pr. 2

dělá to toto

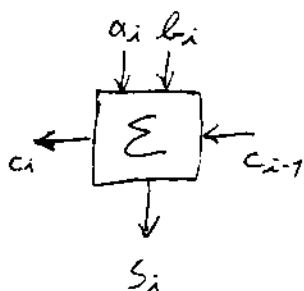
a long

b char sign

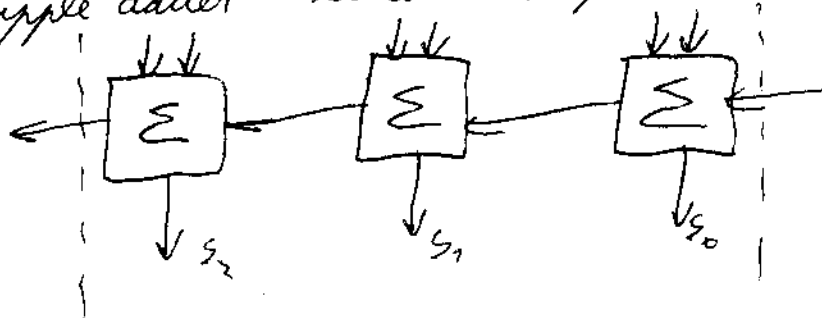
$$a := a + b$$

3

Seřadky



- carry ripple adder - seřadka s postupným přenosem

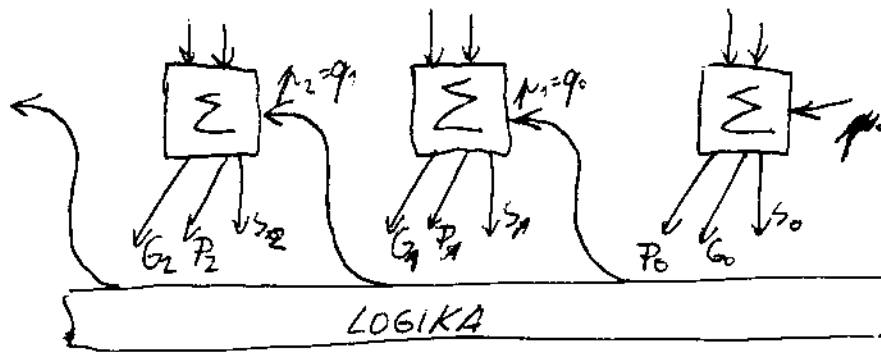


- carry look-ahead adder - seřadka s předikcí přenosu

$$P = \text{PROPAGATE} = a_i \oplus b_i$$

$$G = \text{GENERATE} = a_i b_i$$

zrychlení oproti seř. s postupným přenosem

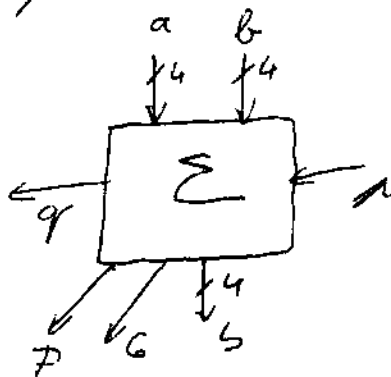


$$q_0 = G_0 + P_0 p_0$$

$$q_1 = G_1 + P_1 p_1 = G_1 + P_1 G_0 + P_1 P_0 p_0$$

$$q_i = G_i + P_i p_i$$

v prípade n-bitového bloku



$$P = P_0 \cdot P_1 \cdot P_2 \cdot P_3$$

$$G = G_0 \overline{P_1} P_2 P_3 + G_1 P_2 P_3 + \dots$$

④ nebylo

⑤

DOP

Datová část

Registry

PE přebírá poslední sig.

Příkladky

kopírovat $S \Rightarrow T$, co doletá

OES = 1	OET = 0
PES = 0	PET = 1
ECS = 0	

SP- $\Rightarrow T$

UDSP = 1	PESP = 0	PET = 1
PESP = 0	$\Rightarrow$ EESP = 0	OET = 0
ECSP = 1	OESP = 1	
OESP = 1	UDSP = X	

ALU

Příklady

- $SP \Rightarrow W$

$OESP = 1, OP = 2, CIN = 0, ENW = 1$

- $W+ \Rightarrow W$

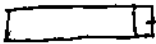
$OP = 7 (8), CIN = 1, ENW = 1$

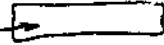
- $W-S \Rightarrow W$ *odčítání*

$OES = 1, OP = 11, CIN = 1, ENW = 1$

↳ k W přičte negaci druhého a CIN

- ~~průměr~~ *roboce* →

1)  CF *neuloží se W*

2) CF 

Příklady

ZF - na odporu může být AND se současnou hodnotou

- při postupném vyhodnocování operandů

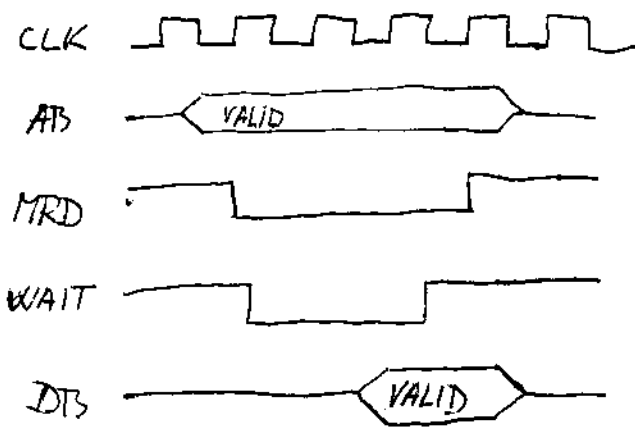
⑥

DOUT vyrovnávací paměť
DIL, DH multiplexace signálů mezi vnější/vnitřní obvodem
IR - složení operačního kroku instrukce

INT. ADDR. vnitřní adresy obsluhy přerušování

čtení paměti / I/O

zápis paměti / I/O



controller

ex
příklad

horizontální mikroprogramový řadič

- ~~10~~ kolik řídicích signálů, tolik bitů paměti

adresa další instrukce - adr + maskování, které
udává velikost bloku

00001 SC=0 MA=00002 , OES *podl. na datl mikroinstrukci*
 00002

cham WAIT

000010₂ SC=10 , MA=00010₂ ; WAIT?
 00011₂ BUT

*potreba premyslet,
 co kee jake noteta
 ;)*

001 | x | x x SC=1 MA=111 | x x x)

*podle op. kodu mikroinstrukce musno uciit,
 na jakou adresu se da dostat ; co se mive
 stat*

Zakladni cyklus

UCF Execute
 SWW ———

*rovnou se ke datl instrukce,
 nechceme HW prerušení*

⑦

Symbolické mikroinstrukce
číselný popis mikroinstrukce
mas.pdf

operační znaky instrukcí - základní v dop-oz.pdf

instrukce mov (stv. liná)

op. kód 00

rozhodováno v simulaci a sledovat
diagram v dop-mier.pdf / str. 2

dekódování instrukce str 3

op. znak se dekóduje postupně podle diagramu

- SHR op.km 9A

- $K+5 \rightarrow W$

ADD

není v tab dop-oz.pdf
najít v instr.pdf

je to OPA, oper +, čtení k PC
010

0010 0001

21

- MOV SP, W

- LD4 SP - [PC + 1] → W

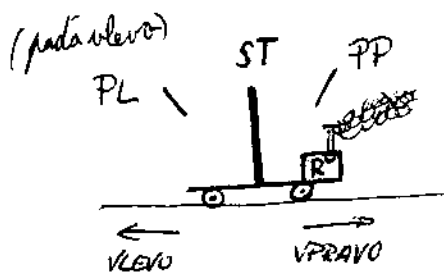
1010 1111

A F

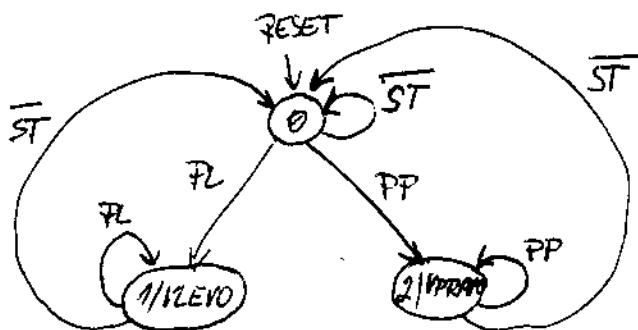
a operand 0

8

Ďalšie

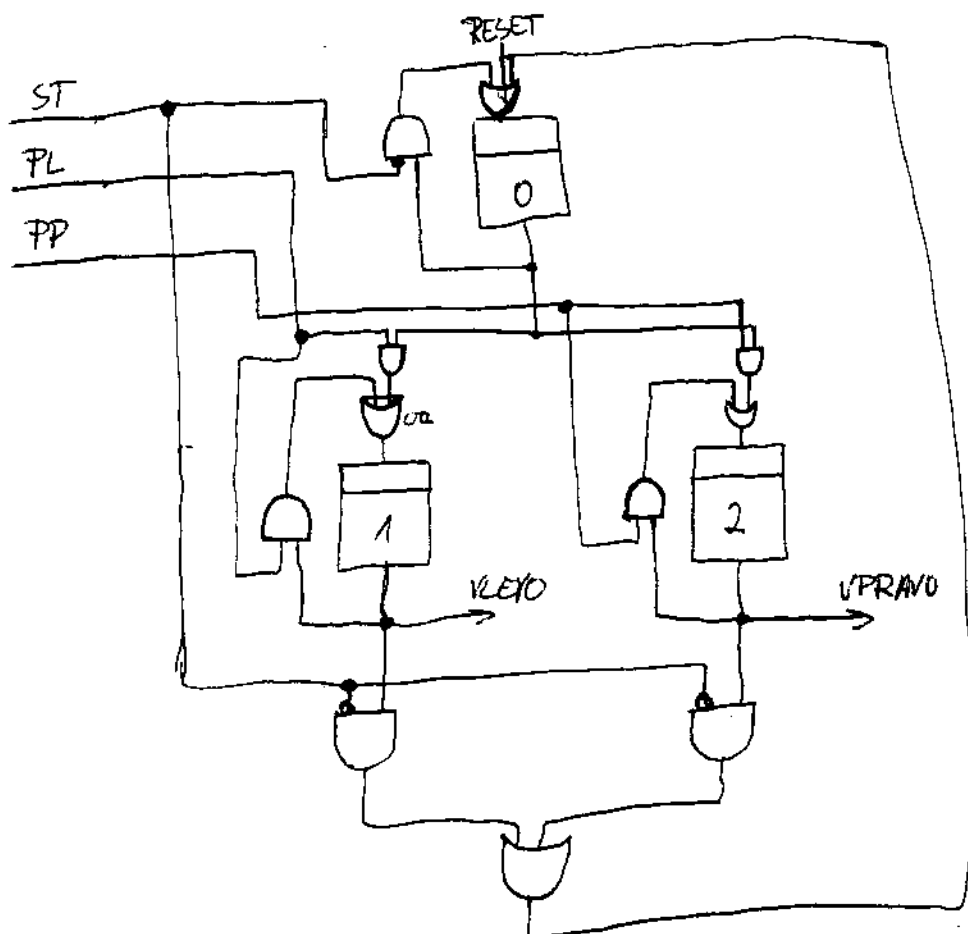


možná bude vyrovnaná
syst., aby nepadala

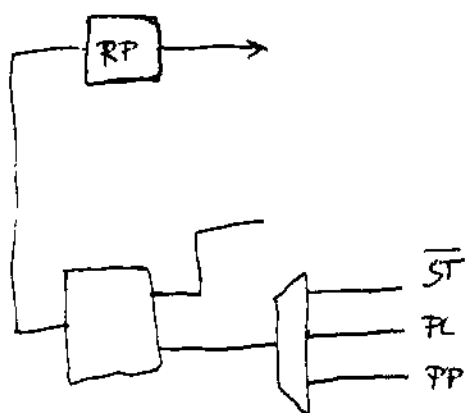


Ďalšie a následní reflexe

každému stavu odpovídá paměťový řetěz
přechody hank. logika



mikroprogramový řadič (horizontální)



ADR	KP	ADRN	^{RS} VLEVO	VPRAVO
00	01	0x	0	0
01	10	1x	0	0
10	00	x0	0	1
11	11	xx	1	0

KP 11 na oba dle PL
 00 na první dle PP
 01 na první dle ST
 10 na ——— PL

jinak - jen jedno x v adce

00	01	0x	0	0
01	10	1x	0	0
10	00	x0	0	1
11	11	1x	1	0

- přeskočí na druhou stranu

jinak - větší paměť

000	01	00x	0	0
001	10	x10	0	0
010	00	0x0	0	1
100	PR-SK	000	0	0
110	11	1x0	1	0

Radič - člení paměť



ATS

MRD

WAIT

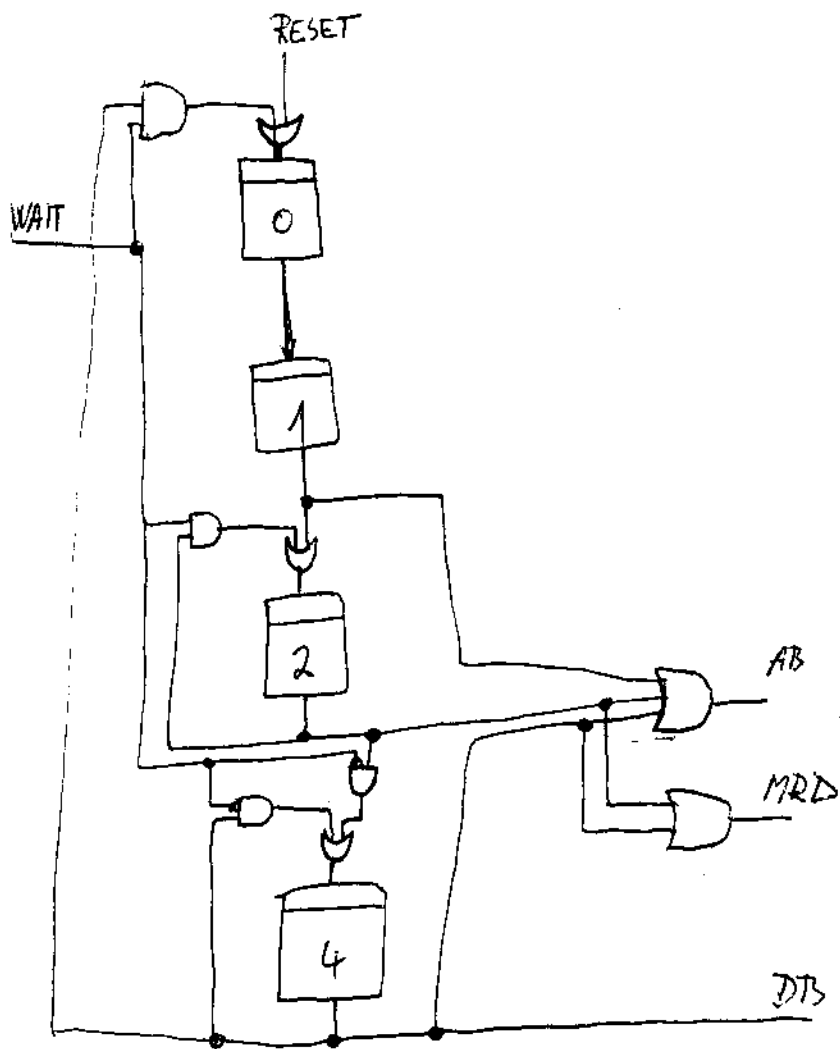
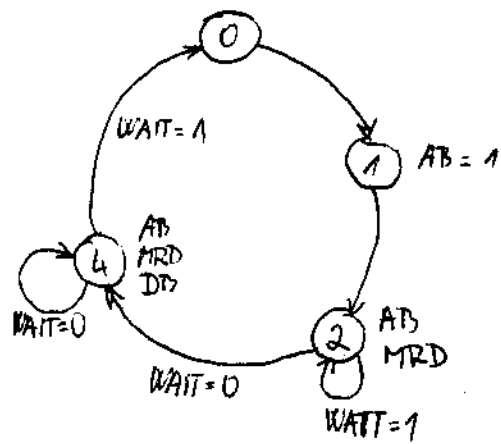
DB

WAB

MRD

WAIT

WDB



A	KF	ADRV	AB	MRD	DB
00	00	01	0	0	0
01	00	10	1	0	0
10	01	$1\bar{w}$	1	1	0
11	11	$\bar{w}\bar{w}$	1	1	1

Dělení

- s návratem přes nulu

$$\begin{array}{r}
 11001_2 : 111_2 = 11,1001_2 \\
 \underline{- 111} \\
 1011 \\
 \underline{- 111} \\
 1000 \\
 \underline{- 111} \\
 1000 \\
 \underline{- 111} \\
 1
 \end{array}$$

← odčítá se - kladně - do výsl 1
záporně - výsl 0

⇒ $+11_2$ } opětná korekce
- ⇒ $+x/2$
+ ⇒ $-x/2$

9

- bez návratu přes nulu

přenos - rychlejší

$$\begin{array}{r}
 11001_2 : 111_2 = 11,1001 \\
 \underline{- 111} \\
 01011 \\
 \underline{- 111} \\
 01000 \\
 \underline{- 111} \\
 0010 \\
 \underline{- 111} \\
 ? -1010 \leftarrow \text{přidá se } 0 \\
 \quad + 111 \quad \text{a přičtení} \\
 ? -00110 \\
 \quad + 111 \\
 \quad + 1
 \end{array}$$

+ $x/2$
- $x/2$

Násobení

v doplnkovém kódu

$$A = -1101_2 \Rightarrow D(A) = 10011_2 \quad -13$$

$$B = 101_2 \Rightarrow D(B) = 00101_2 \quad 5$$

$$\begin{array}{r} 10011 \\ \cdot 00101 \\ \hline 10011 \\ 100110 \\ \hline 1011111 \end{array}$$

$$-0100001$$

$$= -33$$

to není ono

$$\begin{array}{r} 10011 \\ \cdot 00101 \\ \hline 11110011 \\ 00000000 \\ 110011 \\ \hline 11011111 \end{array}$$

$$-01000001$$

$$= -65 \checkmark$$

5 pozic
3 pozice
musno rozšířit
na počet
míst
↓
výsledek 8 poz

Takže možno rozšířit na 8 míst na delku

$$\begin{array}{r} 111110011 \\ 000100101 \end{array}$$

$$\begin{array}{r} 10011 \\ 00101 \end{array}$$

$$\begin{array}{r} \text{výsledek} \\ 000000 \\ 110011 \\ \hline 1110011 \\ 000000 \\ \hline \end{array}$$

$$B = -101_2$$

$$D(B) = 1101_2$$

$$\begin{array}{r} 10011 \\ 11101 \\ \hline 111110011 \\ 11100110 \\ 110011 \\ 10011 \\ \hline 00111 \\ 11 \end{array}$$

$$\begin{array}{r} 10011 \\ - 11011 \\ \hline \end{array}$$

$$\begin{array}{r} 00000 \\ + 10011 \\ \hline 110011 \\ + 10011 \\ \hline 11011001 \\ + 00000 \\ \hline 11011001 \\ + 10011 \\ \hline 1101110001 \\ + 101110001 \\ \hline 0001000001 \end{array}$$

$$= 001101$$

pro 2 operandu
kapsenem
se posledni
odbita
→ prevod do D
a priceni

-1

Relativni cislice

$$\begin{array}{ccc} \hat{1} & 0 & 1 \\ / & / & / \\ -1 & 0 & 1 \end{array}$$

$$11 \Rightarrow 10\hat{1}$$

	a_{i+1}	
a_i	0	1
0	0	1
1	1	0

$$\begin{array}{c} a_i \ a_{i+1} \\ \diagup \ \diagdown \\ 11011, 0_2 \Rightarrow 0\hat{1}10\hat{1}_2 \\ -814 -7 = -5 \end{array}$$

$$10011_2 \times \hat{1}10\hat{1}_2$$

$$\begin{array}{r} 00000 \\ 001101 \\ \hline 001101 \\ 00000 \\ \hline 0001101 \\ 10011 \\ \hline \end{array}$$

$$\begin{array}{r} 11011001 \\ + 01101 \\ \hline 001000001_2 \end{array}$$

Pohyblivá řádová čárka

π, ϵ

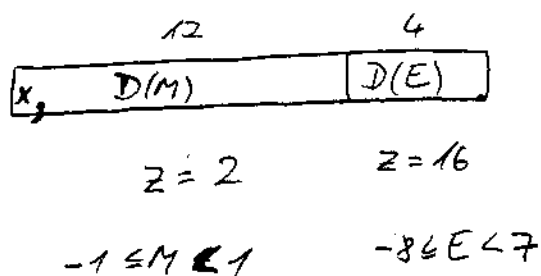
$$\Rightarrow M \times 2^E$$

$$1,6 \cdot 10^3 = 1600$$

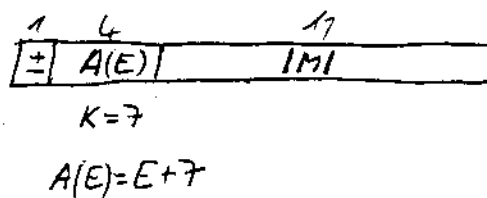
$$16 \cdot 10^2 = 1600$$

$$0,16 \cdot 10^4 = 1600$$

PSEUDO MILITARY



PSEUDO IEEE



PSEUDO MIL

$$BE53_{16} = 1011 \ 1110 \ 0101 \ 0011_2$$

$$-0,10000011011 \times 2^3 = -100,00011011$$

PSEUDO IEEE

$$1011 \ 1110 \ 0101 \ 0011$$

$$-0,110 \ 01010011 \times 2^0$$