MI-ARI

(Computer arithmetics)

winter semester 2017/18

DK. Decimal codes

© **Alois Pluháček Pavel Kubalík**, 2017
Department of digital design
Faculty of Information technology
Czech Technical University in Prague

https://edux.fit.cvut.cz/courses/MIE-ARI/lectures

DK. **Decimal codes**

- **Decimal codes**
- **One positional decimal adder**
- **Adder in BCD code**
- **Conversion between addition and subtraction**
- **Codes $Ga+F$**
- **Adder in code $+3$**
- **Codes $Ga+F$ — complement into 9 (inversion of digit)**
- **Weight codes**
- **Sign numbers representation**

Decimal codes: Representation of individual decimal digits

k **bits/digits** ... k **bits code**

$$2^k \geq 10 \quad \Longrightarrow \quad \boxed{k \geq 4}$$

- **4 bits codes:** $\binom{10}{16} \cdot 10! = 29\,059\,430\,400$ **codes**
  - **code BCD** or **code 8,4,2,1**
  - **code +3** or **code XS3** or **Stibitz code**
  - **Aiken code** (code 2,4,2,1)
  - **Rubinoff codes** or **code 8,4,−2,−1**
- **5 bits codes — error control codes**
  - **code 3**a**+2** or **Brown code**
  - **code** $\frac{16}{3}, \frac{8}{3}, \frac{4}{3}, -\frac{2}{3}, \frac{1}{3}$
  - **code 2 z 5, eventually 3 z 5**
- **10 bits code — code 1 z 10, eventually code 9 z 10**

One positional decimal adder

analogy (binnary) **full-adder** (for deciaml system):

inputs: $a \rightarrow a^* \sim (\alpha_{k-1}, \ldots, \alpha_0)$ ☜ k **bits**

$b \rightarrow b^* \sim (\beta_{k-1}, \ldots, \beta_0)$ ☜ k **bits**

$p \ldots$ carry for lower order ☜ **1 bit**

outputs: $s \rightarrow s^* \sim (\sigma_{k-1}, \ldots, \sigma_0)$ ☜ k **bitů**

$q \ldots$ carry to higher order ☜ **1 bit**

$$q = (a+b+p) \div 10 = \begin{cases} \mathbf{0}, & \text{je-li } a+b+p < 10 \\ \mathbf{1}, & \text{je-li } a+b+p \geq 10 \end{cases}$$

$$s = (a+b+p) \ \% \ 10 = a+b+p-q \cdot 10$$

possible structure (*! only for some codes !*):

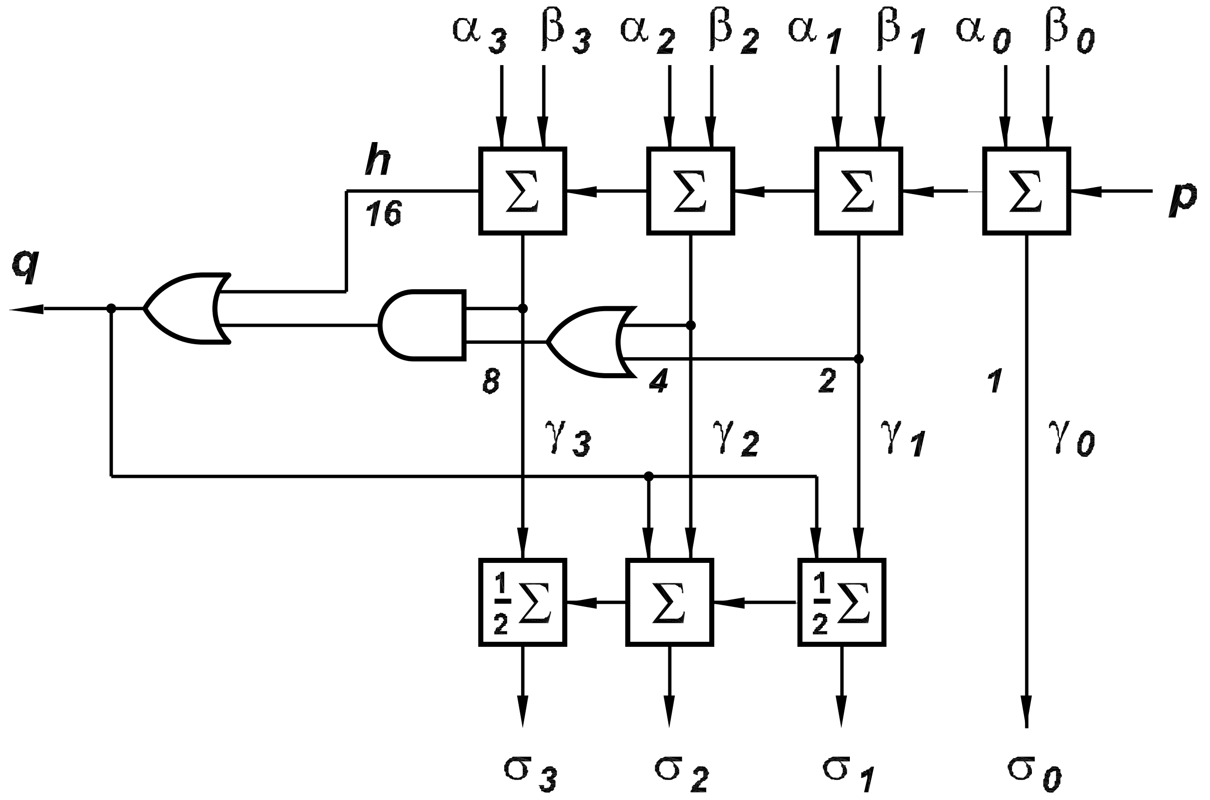
- **basic adder - totals evaluation** $a^* + b^* + p$
  carry h (binnary) **and** (last) k **bits of total**
- **circuit for decimal carry estimation** q
- **circuit for correction estimation** — later denoted kor
- **correction adder — realize correction**

Adder in BCD code

$$a^* = a$$

a	a^*
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

one positional decimal adder in BCD code:



$$kor \equiv \begin{cases} 0 \text{ for } q = 0 \\ -10 \equiv +6 \text{ for } q = 1 \end{cases} \pmod{16}$$

Conversion between addition and subtraction

$A - B \equiv A + (\mathcal{Z} - B) \pmod{\mathcal{Z}}$

$\mathcal{Z} - B = (\mathcal{Z} - 1) - B + 1$

$\mathcal{Z} - 1$ ... all digits are 9 ☛ $\boxed{99\ldots99}$

$\mathcal{Z} - B = \boxed{99\ldots99} - B + 1$

digit inversion b: $\widetilde{b} = (z - 1) - b$

for z=10: $\widetilde{b} = 9 - b$ ☛ *complement to 9*

number inversion $B\widetilde{B}$ ... inversion of all digits

$\widetilde{B} = (\mathcal{Z} - 1) - B + 1$

$\boxed{\mathcal{Z} - B = \widetilde{B} + 1}$ inversion of all digits + hot 1

ex.: 532 − 127 = 405

$\widetilde{127} = \widetilde{1}\widetilde{2}\widetilde{7} = 872$

532 + 872 = 1 405

code BCD:

	b	$\widetilde{b}$
0	0 0 0 0	1 0 0 1
1	0 0 0 1	1 0 0 0
2	0 0 1 0	0 1 1 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 0 1
5	0 1 0 1	0 1 0 0
6	0 1 1 0	0 0 1 1
7	0 1 1 1	0 0 1 0
8	1 0 0 0	0 0 0 1
9	1 0 0 1	0 0 0 0

$$b \sim (\beta_3, \beta_2, \beta_1, \beta_0)$$
$$\widetilde{b} \sim (\gamma_3, \gamma_2, \gamma_1, \gamma_0)$$

$$\gamma_0 = \overline{\beta_0}$$
$$\gamma_1 = \beta_1$$
$$\gamma_2 = \beta_2 \cdot \overline{\beta_1} + \overline{\beta_2} \cdot \beta_1$$
$$\gamma_3 = \overline{\beta_3} \cdot \overline{\beta_2} \cdot \overline{\beta_1}$$

Codes $Ga+F$

$$a^* = G \cdot a + F$$

	BCD G=1, F=0	+3 G=1, F=3	$3a$+2 G=3, F=2
0	0 0 0 0	0 0 1 1	0 0 0 1 0
1	0 0 0 1	0 1 0 0	0 0 1 0 1
2	0 0 1 0	0 1 0 1	0 1 0 0 0
3	0 0 1 1	0 1 1 0	0 1 0 1 1
4	0 1 0 0	0 1 1 1	0 1 1 1 0
5	0 1 0 1	1 0 0 0	1 0 0 0 1
6	0 1 1 0	1 0 0 1	1 0 1 0 0
7	0 1 1 1	1 0 1 0	1 0 1 1 1
8	1 0 0 0	1 0 1 1	1 1 0 1 0
9	1 0 0 1	1 1 0 0	1 1 1 0 1

note.: Codes $Ga+F$ are more usually denote as codes $aN+b$.

Addition:

$$a^* = Ga+F, \quad b^* = Gb+F, \quad s^* = Gs+F$$

$$\begin{aligned} a^* + b^* + p &= G(a+b) + 2F + p && \textit{(have)} \\ s^* &= G(a+b+p-10q) + F && \textit{(want)} \\ cor &= (G-1)\cdot p - F - 10G\cdot q && \textit{(want - have)} \end{aligned}$$

code +3 (XS3, Stibitzs code ☞ G=1, F=0):

$$cor = \begin{cases} -3 & \text{for } q = 0 \quad (-3 \equiv +13 \bmod 16) \\ -30 & \text{for } q = 1 \quad (-30 \equiv +3 \bmod 16) \end{cases}$$

code $3a$+2 (Browns code ☞ G=3, F=2): $cor = 2p - 2 - 30q$

p	q	cor
0	0	-2 $\equiv$ 30 (mod 32)
0	1	-32 $\equiv$ 0 (mod 32)
1	0	0 $\equiv$ 0 (mod 32)
1	1	-30 $\equiv$ 2 (mod 32)

Addition (continue) — **carries**

$\boldsymbol{Ga+F}$ **is for** $G > 0$ **growing** $\Rightarrow$

$$\Rightarrow \quad a > b \iff a^* > b^* \; \ldots \; \textbf{\textit{monotonous code}}$$

code +3 (XS3, Stibitzs code):

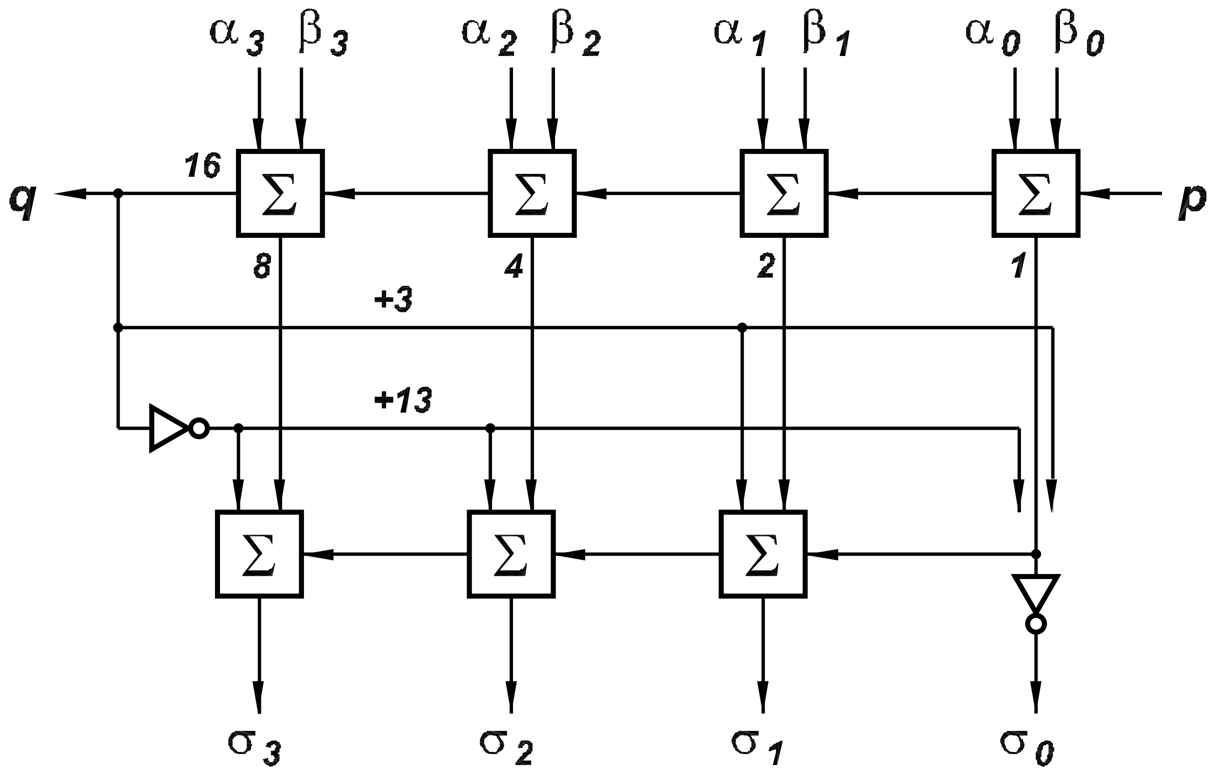
$$\left.\begin{array}{lcl} a+b+p = 9 & \Rightarrow & a^*+b^*+p = 15 \\ a+b+p = 10 & \Rightarrow & a^*+b^*+p = 16 \end{array}\right\} \Longrightarrow q = h$$

code 3a+2 (Browns code):

$$\left.\begin{array}{lcl} a+b \phantom{{}+1} = 9 & \Rightarrow & a^*+b^*+p = 29 \\ a+b+1 = 9 & \Rightarrow & a^*+b^*+p = 30 \\ a+b \phantom{{}+1} = 10 & \Rightarrow & a^*+b^*+p = 32 \\ a+b+1 = 10 & \Rightarrow & a^*+b^*+p = 33 \end{array}\right\} \Longrightarrow q = h$$

Decimal carry for both codes is q
equal to (binnary) **carry** h **using basic adder.**

Adder in code +3



$$cor \equiv \begin{cases} +13 & \text{for } q = 0 \\ +3 & \text{for } q = 1 \end{cases} \quad \text{mod } 16$$

Codes Ga+F — complement to 9 (inversion of digit)

$$c = \widetilde{b} = 9 - b$$

$$\begin{aligned} b^* &= Gb + F & \sim \quad & (\beta_{k-1}, \ldots, \beta_0) \\ c^* &= G(9-b) + F & \sim \quad & (\gamma_{k-1}, \ldots, \gamma_0) \end{aligned}$$

$$c^* = 9G + 2F - b^*$$

$$\left.\begin{array}{ll} \textbf{code } +3\textbf{:} & c^* = 15 - b^* \\ \textbf{code } 3a+2\textbf{:} & c^* = 31 - b^* \end{array}\right\} \Longrightarrow \boxed{\gamma_i = \overline{\beta_i}}$$

For both codes we can obtain representation of $\widetilde{b}$ as an digit complement b into 9 using simple negation of all bits of digit representation b.
Such as code is called *complementary code*.

If the code is monotonous and complementary, the decimal carry q is equal to carry h from basic adder.

Weight codes

weights — orderly k-tuple of numbers: $(v_{k-1},\ldots,v_0)$

$a \rightarrow a^* \sim (\alpha_{k-1},\ldots,\alpha_0)$

$a = \alpha_{k-1} \cdot v_{k-1} + \ldots + \alpha_0 \cdot v_0$

Code can be fully determined with all theirs weights

(for example. 8,4,2,1 or 8,4,-2,-1 or $\frac{16}{3}, \frac{8}{3}, \frac{4}{3}, -\frac{2}{3}, \frac{1}{3}$)

or not

(for example 2,4,2,1 or 5,4,2,1 or 6,3,2,1).

Every weight code is „very strong individuality".

example of weights codes:

- **Aikens code — code (2,4,2,1) described in following table monotonous and complementary code**
- **Rubinoffs code (8,4,-2,-1)** $\Leftrightarrow$ **code +3**
  negation of 2 last bits
- **code** $\frac{16}{3}, \frac{8}{3}, \frac{4}{3}, -\frac{2}{3}, \frac{1}{3}$ $\Leftrightarrow$ **Browns code**
  negation of last but one bit

	BCD 8 4 2 1	Aiken 2 4 2 1	Rubinoff 8 4 -2 -1	$\frac{16}{3}$ $\frac{8}{3}$ $\frac{4}{3}$ $-\frac{2}{3}$ $\frac{1}{3}$
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0 0
1	0 0 0 1	0 0 0 1	0 1 1 1	0 0 1 1 1
2	0 0 1 0	0 0 1 0	0 1 1 0	0 1 0 1 0
3	0 0 1 1	0 0 1 1	0 1 0 1	0 1 0 0 1
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 0 0
5	0 1 0 1	1 0 1 1	1 0 1 1	1 0 0 1 1
6	0 1 1 0	1 1 0 0	1 0 1 0	1 0 1 1 0
7	0 1 1 1	1 1 0 1	1 0 0 1	1 0 1 0 1
8	1 0 0 0	1 1 1 0	1 0 0 0	1 1 0 0 0
9	1 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1 1

Sign and magnitude

sign & absolute value:

sign: only 1 bit is sufficient

4bits code and 2 digits in one byte:

1 nibble (half of byte) = 4 bits for sign

it is possible to use four bits for sign,

which are not used for digits

(there are 6 possibilities)

representable numbers X:

$\mathcal{Z}$' ... module of the format for absolute value

$$-\mathcal{Z}' < X < \mathcal{Z}'$$

☞ symmetrical range

two zero representation: $\begin{cases} \text{so called „positive zero“} \\ \text{so called „negative zero“} \end{cases}$

Radix complement

$$\mathcal{D}(X) = \begin{cases} X & \text{for } X \geq 0 \\ \mathcal{Z} + X = \mathcal{Z} - |X| & \text{for } X < 0 \end{cases}$$

$-\frac{1}{2}\mathcal{Z} \leq X < \frac{1}{2}\mathcal{Z}$ ☞ **asymetrical range**

MSD . . . digits in higher order (first from left)

MSD $< 5 \iff X \geq 0$

MSD $\geq 5 \iff X < 0$

codes monotonous and complementary (and codes derived from them)**:**

$a \rightarrow a^* \sim (\alpha_{k-1}, \ldots, \alpha_0)$

$$\boxed{a \geq 5 \iff \alpha_{k-1} = 1}$$

monotonous and complementary is

for example code +3, Aikens or Browns code

Radix complement (continue)

code BCD is not complementary!

$$a \;\rightarrow\; a^* \sim (\alpha_{k-1}, \ldots, \alpha_0)$$

$$a < 5 \implies \overline{\alpha_3}\,\overline{\alpha_2} + \overline{\alpha_3}\,\overline{\alpha_1}\,\overline{\alpha_0} = 1$$

$$a \geq 5 \implies \alpha_3 + \alpha_2\alpha_1 + \alpha_2\alpha_0 = 1$$

addition : $\mathcal{D}(A+B) \equiv \mathcal{D}(A) + \mathcal{D}(B) \pmod{\mathcal{Z}}$

overflow: **signs differ**

$$\boxed{+}\ \boxed{+} \rightarrow \boxed{-}$$

$$\boxed{-}\ \boxed{-} \rightarrow \boxed{+}$$

subtraction :

$$A - B = A + (\mathcal{Z} - B) + \mathcal{Z}$$

$$\mathcal{Z} - B = \widetilde{B} + \varepsilon$$

$$A - B = A + \widetilde{B} + \varepsilon - \mathcal{Z}$$

diminished radix complement

$$\mathcal{I}(X) = \begin{cases} X & \text{pro } X \geq 0 \\ \widetilde{|X|} & \text{pro } X \leq 0 \end{cases}$$

$-\frac{1}{2}\mathcal{Z} < X < \frac{1}{2}\mathcal{Z}$ ☞ symetrical range

two zero representation: $\begin{cases} \text{so called „pozitive zero“} & 0\,0 \ldots 0 \\ \text{so called „negative zero“} & 1\,1 \ldots 1 \end{cases}$

MSD $< 5 \implies X \geq 0$
MSD $\geq 5 \implies X \leq 0$

addition: $\mathcal{I}(A + B) \equiv \mathcal{I}(A) + \mathcal{I}(B) \pmod{\mathcal{Z}-1}$
$\mathcal{I}(A) + \mathcal{I}(B) \geq \mathcal{Z} \quad \Rightarrow \quad q=1$
necessary subtract $\mathcal{Z}$ and add 1 $\Rightarrow$ circular carry

overflow detection and subtraction: similarly as in radix complement code

Biased codes

$$\mathcal{A}(X) = X + K$$

$K = \frac{1}{2}\mathcal{Z}$ ☞ $\mathcal{A}_0(X)$ ... **biased code type 0**

$K = \frac{1}{2}\mathcal{Z}-1$ ☞ $\mathcal{A}_1(X)$ ... **biased code type 1**

addition: $\boxed{\mathcal{A}(A+B) = \mathcal{A}(A) + \mathcal{A}(B) - K}$

subtraction: $\boxed{\mathcal{A}(A-B) = \mathcal{A}(A) - \mathcal{A}(B) + K}$

$\mathcal{A}_0(X)$: $-\frac{1}{2}\mathcal{Z} \leq X < \frac{1}{2}\mathcal{Z}$

$\mathcal{A}_1(X)$: $-\frac{1}{2}\mathcal{Z} < X \leq \frac{1}{2}\mathcal{Z}$

sign of overflow detection and subtraction:

similarly to previous cases

