Budapest University of Technology and Economics

Faculty of Transportation Engineering and Vehicle Engineering

Department of Control for Transportation and Vehicle Systems

ARITHMETIC OPERATIONS, PART 2.

Lecture 4.

Floating-Point Operations

- Operations in floating-point numbers:
  - $A = \pm m_A * 2^{\pm k_A}$
  - $B = \pm m_A * 2^{\pm k_B}$

- Addition/substraction:
  1. it is needed to convert the characteristics to the same value,
  2. addition,
  3. if it is needed, it has to transform the result

  - addition/substraction is executed only with mantissas! (m_A and m_B)

Floating-Point Operations

- Addition/substraction, e.g:
  - $A = 0{,}9 * 10^2$
  - $B = 0{,}993 * 10^4$

- Addition/substraction:
  1. coversion: $0{,}9*10^2=0{,}09*10^3=0{,}009*10^4$
  2. addition: 0,009 + 0,993= 1,002
  3. transformation to normalized form: $1{,}002*10^4=0{,}1002*10^5$

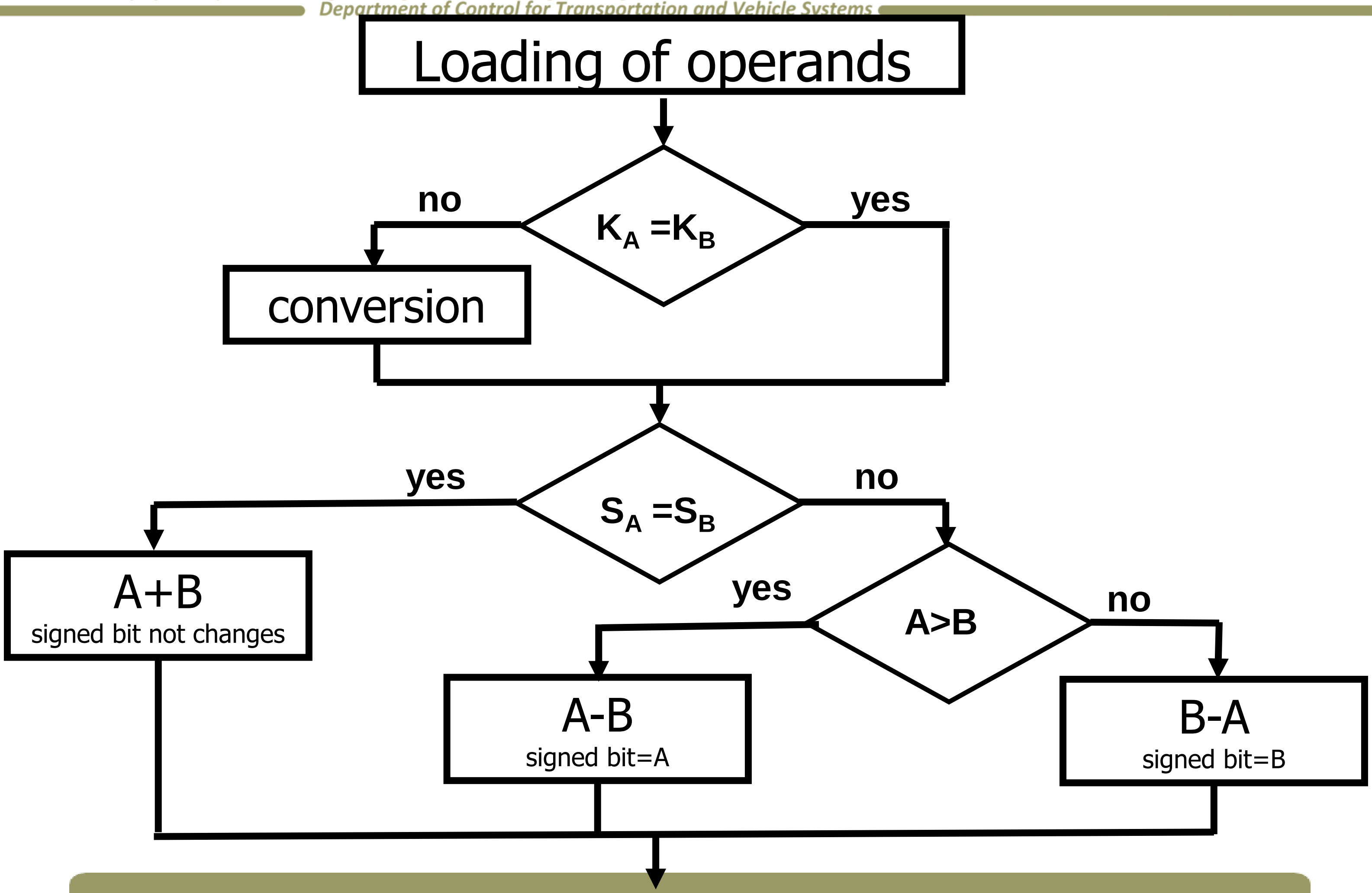
Floating-Point Operations

- Addition/substraction, e.g:
  - $A = 0{,}11001 * 2^4$
  - $B = -0{,}11111 * 2^5$

- Addition/substraction:
  1. coversion: $0{,}11001*2^4=0{,}011001*2^5$
  2. substraction, but A>B, then the substraction=B-A=0,111110-0,011001=0,100101
  3. result:$-0{,}100101*2^5=-18{,}5_{10}$

Floating-Point Operations

- Algorithm of a floating-point adder in the ALU



Floating-Point Operations

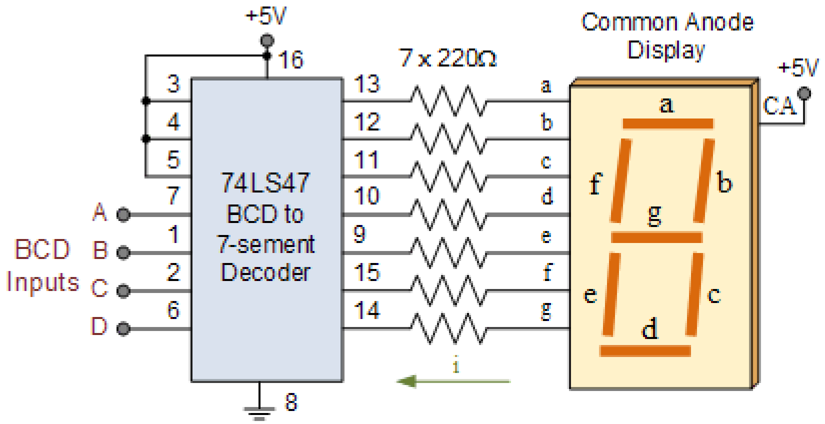
- Multiplication:
  - $A = \pm m_A * 2^{\pm k_A}$
  - $B = \pm m_A * 2^{\pm k_B}$
  - $N = A * B$
  - $N = \left(\pm m_A * 2^{\pm k_A}\right) * \left(\pm m_B * 2^{\pm k_B}\right)$
  - $N = \pm m_A * m_B * 2^{k_A + k_B}$
- Algorithm:
  1. multiplication of the mantissas, (multiplication: see later),
  2. signed bit: $S_N = S_A \oplus S_B - XNOR$,
  3. addition of the characteristics – offset zero point representation!!,
  4. if it is needed, it has to transform the result (e.g. to normalized form).

Floating-Point Operations

- Division:
  - $A = \pm m_A * 2^{\pm k_A}$
  - $B = \pm m_A * 2^{\pm k_B}$
  - $N = \frac{A}{B}$
  - $N = \frac{\pm m_A * 2^{\pm k_A}}{\pm m_B * 2^{\pm k_B}}$
  - $N = \pm \frac{m_A}{m_B} * 2^{k_A - k_B}$
- Algorithm:
  1. division of the mantissas by repeated substraction (see later),
  2. signed bit: $S_N = S_A \oplus S_B - XNOR$,
  3. substraction of the characteristics, dividend – divisor, – offset zero point representation!!,
  4. if it is needed, it has to transform the result (e.g. to normalized form)

BCD Operations

- Binary – Coded Decimal number representation:
  - BCD is a class of binary encodings of decimal numbers,
  - each decimal digit is represented by a fixed number of bits,
    - usually four/eight,
  - 4-bits are equals to $2^4=16$ code words, called tetrades,
    - nibble or tetrade: 0000, 0001, …, 1000, 1001,
    - pseudo-tatrades or pseudo-decimal digits: 1010, 1011, 1100, 1101, 1110, 1111
  - e.g: $93_{10}=10010011_{BCD}$
  - e.g: $10100111000.101_2 = 0101\ 0011\ 0001.1010_{BCD} = 538.625_{10}$
  - advatage: simple coding/decoding,
  - problem: arithmetic operations.



source: https://www.electronics-tutorials.ws/binary/binary-coded-decimal.html

BCD Operations

- Addition in BCD:
  - requisite 1: $A_{BCD} + B_{BCD} = (A+B)_{BCD}$,
  - requisite 2: it has to be working for complement code also,

- Case 1:
  - $A = 273_{10}$, $B = 512_{10}$

$$\begin{array}{r} 273_{10} \\ +512_{10} \\ \hline 785_{10} \end{array} \qquad \begin{array}{r} 0010\ 0111\ 0011_{BCD} \\ +0101\ 0001\ 0010_{BCD} \\ \hline 0111\ 1000\ 0101_{BCD} \end{array}$$

  - the result is good.

BCD Operations

- Case 2:
  - A $=273_{10}$, B$=879_{10}$

CYCY CY

$$\begin{array}{r} 273_{10} \\ +879_{10} \\ \hline 1152_{10} \end{array}$$

DA DA DA

$$\begin{array}{r} 0010\ 0111\ 0011_{BCD} \\ +1000\ 0111\ 1001_{BCD} \\ \hline 1010\ 1110\ 1100 \\ 10\quad 14\quad 12\ \text{- not in BCD!} \end{array}$$

DA = decimal adjust (a carry between tetrades)

- solution:
  - if $9<\text{Result}\leq 15$, it is needed to add 6 to the result (in every tetrade, where this condition is true)

DA DA DA

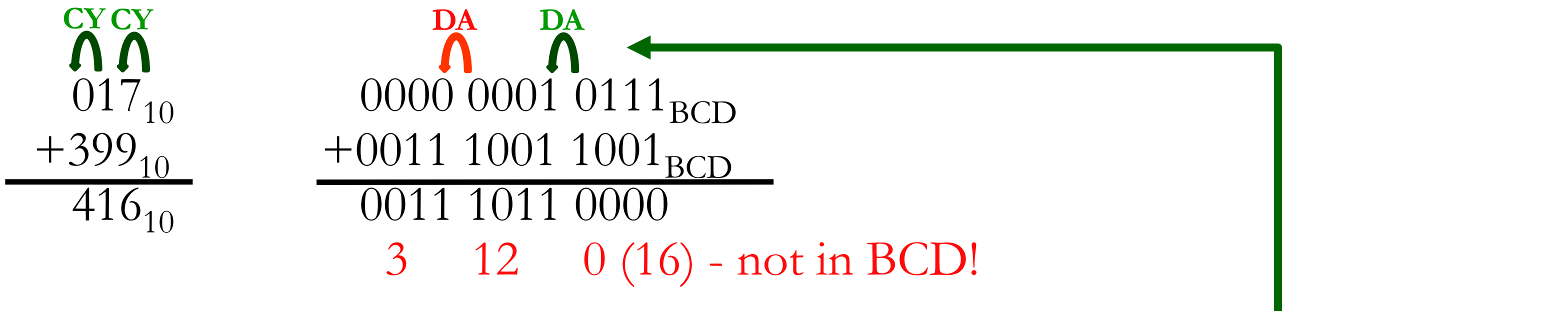
$$\begin{array}{r} 1010\ 1110\ 1100 \\ +0110\ 0110\ 0110_{BCD} \\ \hline 0001\ 0001\ 0101\ 0010_{BCD} \\ 1\quad 1\quad 5\quad 2\ \text{- in BCD!} \end{array}$$

if DA is needed:

Res=Res + correction (+6) +1 (+16)

BCD Operations

- Case 3:
  - A $=017_{10}$, B$=399_{10}$



CY CY

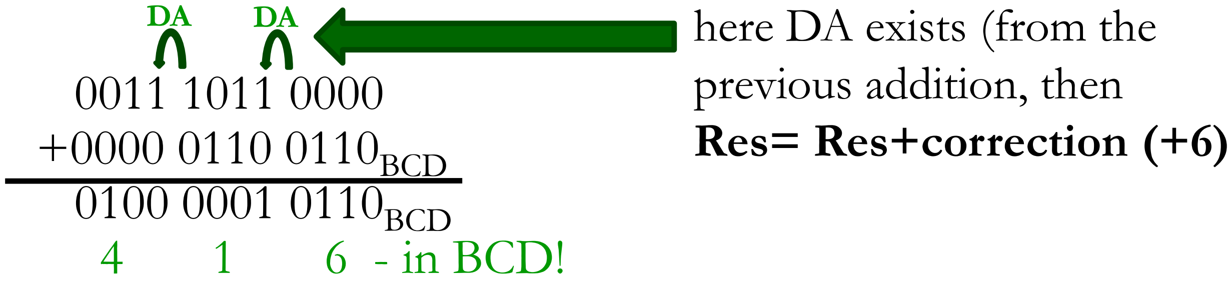
$$\begin{array}{r} 017_{10} \\ +399_{10} \\ \hline 416_{10} \end{array}$$

DA DA

$$\begin{array}{r} 0000\ 0001\ 0111_{BCD} \\ +0011\ 1001\ 1001_{BCD} \\ \hline 0011\ 1011\ 0000 \end{array}$$

3 12 0 (16) - not in BCD!

  - solution:
    - if it was a decimal adjust, it has to be add 6 to the result



DA DA

$$\begin{array}{r} 0011\ 1011\ 0000 \\ +0000\ 0110\ 0110_{BCD} \\ \hline 0100\ 0001\ 0110_{BCD} \end{array}$$

4 1 6 - in BCD!

here DA exists (from the previous addition, then

Res= Res+correction (+6)

BCD Operations

- Summation:

A_i+B_i	wrong Res	good Res	DA	correction
0	0000	0000	No DA	Not necessary
1	0001	0001		
2	0010	0010		
3	0011	0011		
4	0100	0100		
5	0101	0101		
6	0110	0110		
7	0111	0111		
8	1000	1000		
9	1001	1001		
10	1010	0000	No DA, it has to generate it	+6 (+0110)
11	1011	0001		
12	1100	0010		
13	1101	0011		
14	1110	0100		
15	1111	0101		
16	(1)0000	0110	it generates	
17	(1)0001	0111		
18	(1)0010	1000		

BCD Operations

- Substraction: addition in 10’s complement code (in binary form)
- Case 1:
  - A >0, B<0, A =763, B=-948, k=4, Res=$-185_{10}=9815_{10'c}$
  - 10’s complement code of -948 $=10^4-948=9052_{10'c}$

$$\begin{array}{r} \overset{\text{CY}}{0}763_{10'c} \\ +9052_{10'c} \\ \hline 9815_{10'c} \end{array} \qquad \begin{array}{r} 0000\ 0111\ \overset{\text{DA}}{0}110\ 0011_{BCD} \\ +1001\ 0000\ 0101\ 0010_{BCD} \\ \hline 1001\ 0111\ 1011\ 0101 \\ 9 \quad\ 7 \quad\ 11 \quad\ 5\text{ - not in BCD!} \end{array}$$

- the result is not good, DA is missing, we have to add six to the second tetrade!

$$\begin{array}{r} 1001\ 0111\ \overset{\text{DA}}{1}011\ 0101 \\ +0000\ 0000\ 0110\ 0000_{BCD} \\ \hline 1001\ 0000\ 0001\ 0101_{BCD} \\ 9 \quad\ 8 \quad\ 1 \quad\ 5\text{ - in BCD!} \end{array}$$

BCD Operations

- Case 2:
  - A <0, B>0, A =-763, B=948, k=4, Res=$185_{10}=185_{10\text{'c}}$
  - 10's complement code of -763 $=10^4-763=9237_{10\text{'c}}$

$$\begin{array}{r} \text{CY CY CY} \\ 9237_{10\text{'c}} \\ +0948_{10\text{'c}} \\ \hline 10185_{10\text{'c}} \end{array} \qquad \begin{array}{r} \text{DA} \quad \text{DA} \qquad \text{DA} \quad \\ 1001\ 0010\ 0011\ 0111_{BCD} \\ +0000\ 1001\ 0100\ 1000_{BCD} \\ \hline 1001\ 1011\ 0111\ 1111 \end{array}$$

unnecessary bit

9 11 7 15 - not in BCD!

  - the result is not good, DA is missing, we have to add 6 to the adequate tetrades!

$$\begin{array}{r} \text{DA} \quad \text{DA} \qquad \text{DA} \quad \\ 1001\ 1011\ 0111\ 1111 \\ +0000\ 0110\ 0000\ 0110_{BCD} \\ \hline 1010\ 0001\ 1000\ 0101 \end{array} \qquad \begin{array}{r} \text{DA} \quad \text{DA} \qquad \text{DA} \quad \\ 1010\ 0001\ 1000\ 0101 \\ +0110\ 0000\ 0000\ 0000_{BCD} \\ \hline 0001\ 0000\ 0001\ 1000\ 0101_{BCD} \end{array}$$

$\boxed{10}$ 1 8 5 – not in BCD!

$\boxed{1}$ 0 1 8 5 – in BCD!

unnecessary tetrade

BCD Operations

- Case 3:
  - A <0, B<0, A =-763, B=-948, k=4, Res=-1711_{10}
  - 10's complement code of -1711 =10^4-1711=$8289_{10'c}$

CY

$$\begin{array}{r} 9237_{10'c} \\ +9052_{10'c} \\ \hline {\color{red}1}8289_{10'c} \end{array}$$

unnecessary bit

DA

$$\begin{array}{r} 1001\ 0010\ 0011\ 0111_{BCD} \\ +1001\ 0000\ 0101\ 0010_{BCD} \\ \hline 0001\ 0010\ 0010\ 1000\ 1001_{BCD} \end{array}$$

1 2 2 8 9 - in wrong BCD!

here DA exists (from the previous addition, then **Res= Res+correction (+6)**

- the result is not good, DA is missing, we have to add 6 to the adequate tetrades!

DA

$$\begin{array}{r} 0001\ 0010\ 0010\ 1000\ 1001_{BCD} \\ +0000\ 0110\ 0000\ 0000\ 0000_{BCD} \\ \hline 1010\ 1001\ 0010\ 1000\ 1001 \end{array}$$

$\boxed{10}$ 8 2 8 9– not in BCD!

DA

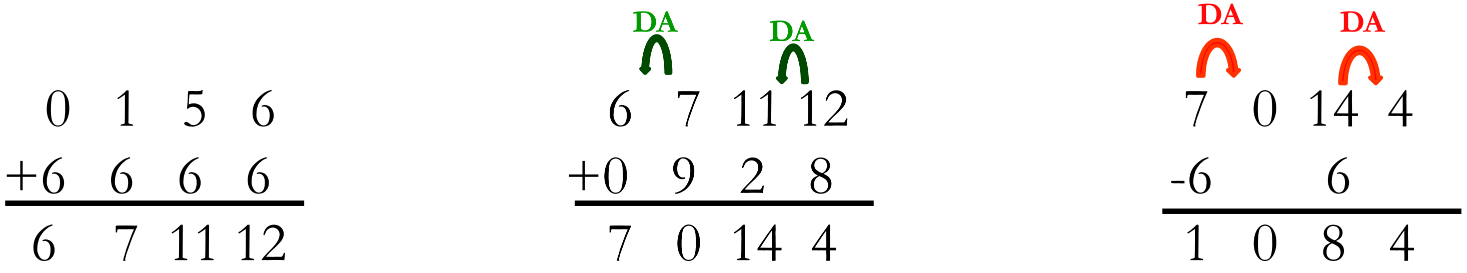
$$\begin{array}{r} 1010\ 1001\ 0010\ 1000\ 1001_{BCD} \\ +0110\ 0000\ 0000\ 0000\ 0000_{BCD} \\ \hline 0001\ 0000\ 1001\ 0010\ 1000\ 1001_{BCD} \end{array}$$

$\boxed{1}$ 0 8 2 8 9 – in BCD!

unnecessary tetrade

BCD Operations

- Other method in BCD addition:
  - correction before the addition, then substraction -6, where no was decimal adjust
  - e.g: A=156_{10}, B=928_{10}, Res=1084_{10}



```
                            DA    DA                 DA     DA
  0   1   5   6          6   7  11 12            7   0  14   4
+6   6   6   6        +0   9   2   8            -6         6
-------------------    -------------------     -------------------
 6    7  11 12          7   0  14   4            1   0   8    4
```

End of Lecture 4.

Thank you for your attention!