MIE-ARI
(Computer arithmetics)

Pavel Kubalík
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague

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

Introduction I.

- Ing. Pavel Kubalík, Ph.D.
- Office Room: A-1037
- Email: Pavel.Kubalik@fit.cvut.cz
- Research Interests
  - Fault-tolerant design in FPGA
  - Digital design
  - Self testing circuits based on FPGA
  - HW design of networks
  - High-speed wireless networks

Introduction II.

- Course type: 2+1, (lecture+ seminar), course end: assessment + exam
- Lecture: every week
- Seminar: ones per two weeks
- Assessment:
  - homeworks (50 points), minimum is 25 points
- Exam:
  - Test (50 points). (A, B, C, D, E, F)

Motivation

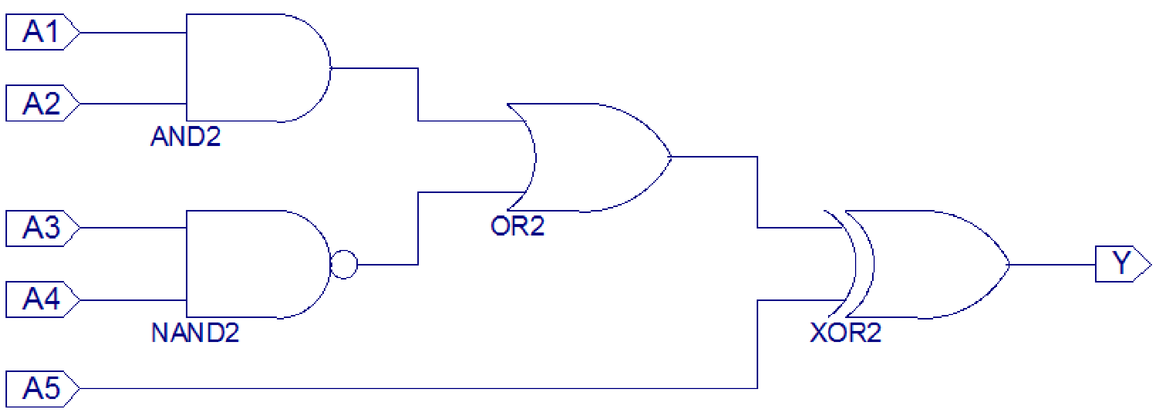
- Arithmetic
  - Every common processor
  - Used in telecommunication
  - Used in embedded processor
  - Data processing (hardware and software)
  - Used in Cryptography

Literature

- Computer Arithmetic
  - Parhami, B.: Computer Arithmetic: Algorithms and Hardware Designs. Oxford University Press, 1999. ISBN 0195125835.
  - Koren, I.: Computer Arithmetic Algorithms (2nd edition). A K Peters, 2001. ISBN 1568811608.
  - Muller, J. M.: Elementary Functions: Algorithms and Implementation (2nd edition). Birkhäuser Boston, 2005. ISBN 0817643729.

First Lecture contents - Recapitulation

- Recapitulation of previous courses (BIE-CAO = Digital and Analog Circuits, BIE-SAP = Computer Structures and Architectures, BIE-JPO=Computer units) – lecture + tutorial



Lecture contents

- Introduction and recapitulation.
- Number systems and basic operations.
- Decimal codes.
- Multiplication I.
- Division I.
- Floating point.
- Problem with carry and its accelerating.
- Multiplication II.
- Division II.
- Elementar functions I.
- Elementar functions II.
- Non standard number system.

Specification levels

- Behavioral (functional) specification
  - Black-box view: describes what is the function of the device (input – output dependence),
  - device implementation is not included.
- Structure specification
  - white-box view: describes the device implementation (interconnection of building elements).
- Physical specification
  - describes physical properties of each partial block (size, power consumption, propagation delays, voltage ranges, available temperature limits,….).

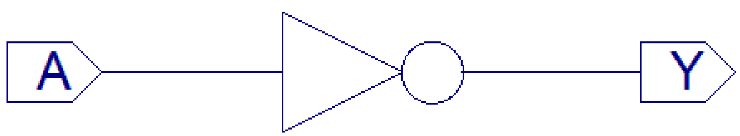
Abstraction levels of various views of digital devices

Abstraction Levels	Functional Description	Structure Description	Physical Description
Transistor Level	Differential equation, transistor volt-ampere characteristics.	Transistors, resistances.	Analog and digital cells; layout.
Gate Level	Boolean equation, finite automaton.	Gates, flip-flops	Module, blocks.
Register Level	Algorithm, flow-chart, set of functions (load, increment, …)	Adders, registers, comparators.	Microchips.
System Level	Functional specification, programs.	Processor, memories, convertors, …	Printed circuit boards, systems on chip, racks.

Gate level- functional and structure description

INV (NOT)

$\overline{A} = Y$

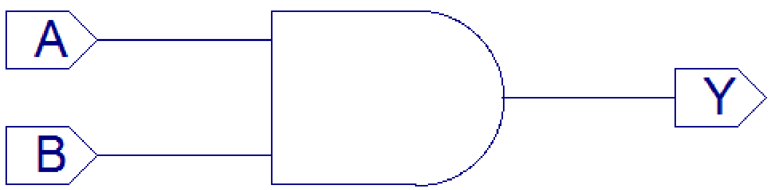


A	Y
0	1
1	0

Gate level- functional and structure description

AND

A B = A & B = Y

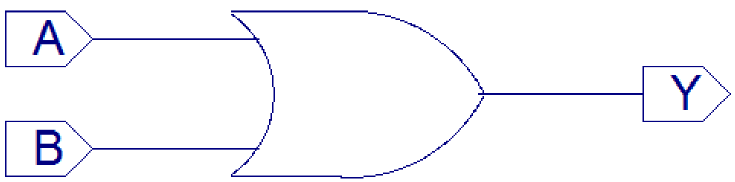


A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Gate level- functional and structure description

OR

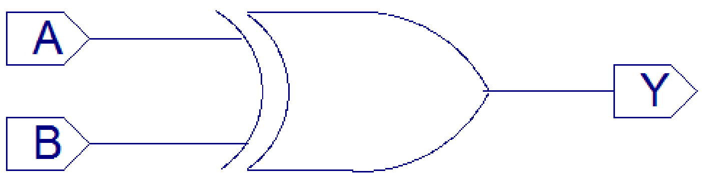
$$A + B = Y$$



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Gate level- functional and structure description

XOR



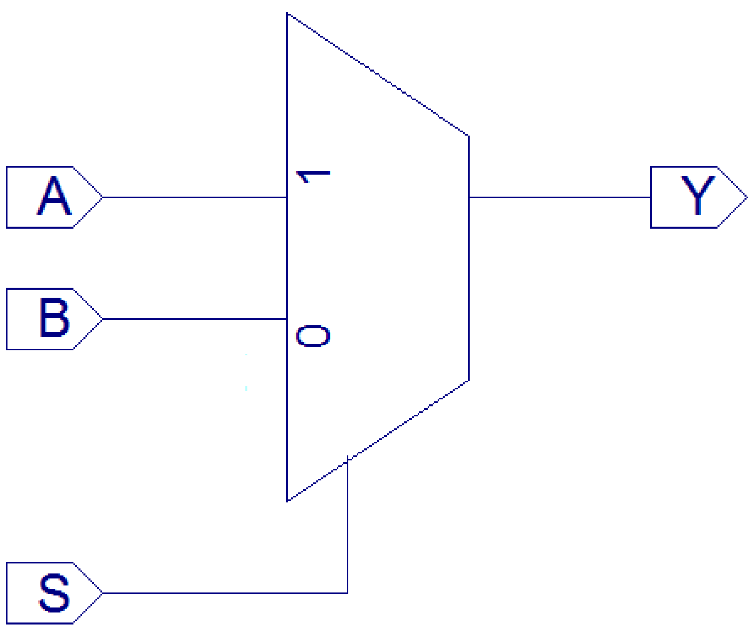
$A \oplus B = Y$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Functional and structure description

Multiplexor

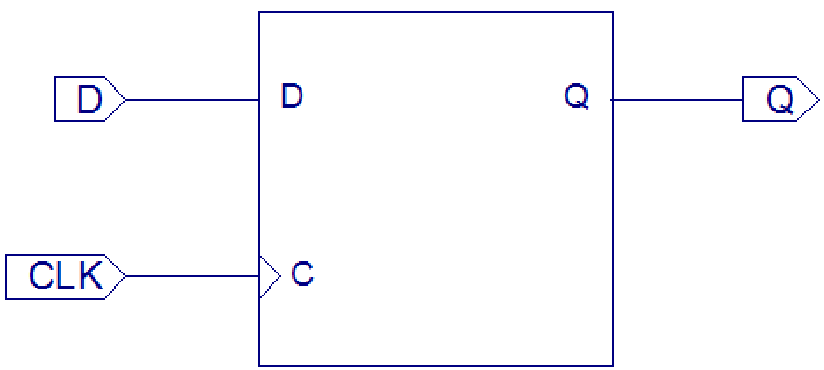
$$A\,S + B\,\overline{S} = Y$$



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Functional and structure description

Register, flip-flop (D type)



D	CLK	Q_{i-1}	Q_i
0	0/1	0	0
1	0/1	1	1
0		X	0
1		X	1

Q_i <= D when rising edge of clk else Q_{i-1}

Other functions

- INV, AND, OR, NOR, XOR, Register, Multiplexor
- NAND, NOR, XNOR, ...

Used conventions

z	number system radix (base of radix system)
n	most significant position
$-m$	least significant position
$Z = z^{n+1}$	format module
$\varepsilon = z^{-m}$	format unit
MSB	most significant bit
LSB	least significant bit
over	overflow

Binary digit system

- radix (base) r = 2 number record is a sequence of binary digits (zeroes or ones)
- Example:

$$v_i \quad \ldots 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0 \quad 2^{-1} \quad 2^{-2} \quad 2^{-3}$$

$$1 \quad 0 \quad 0 \quad 1 \quad 1 \, , \, 1 \quad 0 \quad 1_2$$

$$v(A) = 2^4 + 2 + 1 + 1/2 + 1/8 = \underbrace{19{,}625}_{\text{this is decimal value of the number}}$$

- there are separate methods for converting the integer part and the fractional part of a number

Conversion of number integer part (from decimal to binary)

- repeated dividing of the integer part of the number by radix 2 and putting together all remainders

Example: convert the number 57_{10} to the binary system.

$$57_{10} \approx A_2$$

$$A_2 = 111001_2$$

$$\begin{cases} 57 : 2 = 28 & \textit{remainder } 1 \; \ldots \; a_0 \\ 28 : 2 = 14 & \textit{remainder } 0 \\ 14 : 2 = 7 & \textit{remainder } 0 \\ 7 : 2 = 3 & \textit{remainder } 1 \\ 3 : 2 = 1 & \textit{remainder } 1 \\ 1 : 2 = {\color{red}0} & \textit{remainder } 1 \; \ldots \; a_5 \end{cases}$$

Note: remainders are recorded in opposite order

Conversion of number fractional part (from decimal to binary)

- repeated multiplying of the fractional part of the number by radix 2

Example: convert the number 0,6562510 into the binary system.

$$0{,}65625_{10} \approx A_2$$

$$A_2 = 0{,}10101_2$$

$$\left\{\begin{array}{llll} 0{,}65625 & \cdot\, 2 = & 1 & ,\,3125 \quad \ldots\, a_{-1} \\ 0{,}3125 & \cdot\, 2 = & 0 & ,\,625 \\ 0{,}625 & \cdot\, 2 = & 1 & ,\,25 \\ 0{,}25 & \cdot\, 2 = & 0 & ,\,5 \\ 0{,}5 & \cdot\, 2 = & 1 & ,\,{\color{red}0} \end{array}\right.$$

$$\ldots\, a_{-5}$$

END. Or is it?
Some numbers
never end.

Examples of number conversion

1. $11010001{,}11_2$ → $209{,}75_{10}$
2. 1111111_2 → 127_{10}
3. $1{,}011001_2$ → $1{,}390625_{10}$

4. $147{,}15625_{10}$ → $1001\ 0011{,}0010\ 1_2$
5. $1345{,}125_{10}$ → $101\ 0100\ 0001{,}001_2$

Most important values of power of 2

n	2^n	Dec.
0	2^0	1
1	2^1	2
2	2^2	4
3	2^3	8
4	2^4	16
5	2^5	32
6	2^6	64
7	2^7	128

n	2^n	Dec.
8	2^8	256
9	2^9	512
10	2^{10}	1 024
11	2^{11}	2 048
12	2^{12}	4 096
13	2^{13}	8 192
14	2^{14}	16 384
15	2^{15}	32 768
16	2^{16}	65 536

n	2^n	Dec.
20	2^{20}	1 M
30	2^{30}	1 G
32	2^{32}	4 G
40	2^{40}	1 T
-1	2^{-1}	0,5
-2	2^{-2}	0,25
-3	2^{-3}	0,125
-4	2^{-4}	0,0625

This is very useful to remember!

Binary addition

- **Basic principle:**
  sum of two one-digit numbers:

+	0	1
0	0	1
1	1	10

Carry to the higher order.

Example: adding binary numbers 0101_2 and 1110_2.

$$
\begin{array}{r}
0\,1\,0\,1 \\
+ \quad 1\,1\,1\,0 \\
\hline
{}^{}\,1\,{}^{1}0\,{}^{1}0\,{}^{0}1\,{}^{0}1
\end{array}
$$

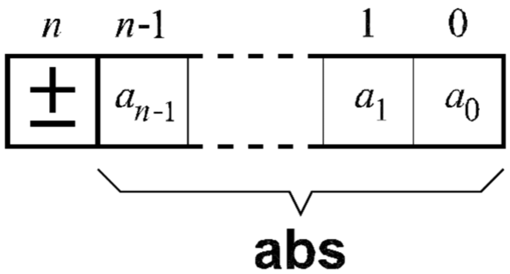
The carry generated during adding digits of the order i is added to digits of the order (i+1).

Note: addition of two n-digit numbers can produce (n+1)-digit number.

Signed numbers

1. sign and magnitude $P(X)$
2. radix complement $D(X)$
   - 2’s complement $z = 2$
   - 10’s complement $z = 10$
3. diminished radix complement $I(X)$
   - 1’s complement $z = 2$
   - 9’s complement $z = 10$
4. biased code $A(X)$

Sign and magnitude code



$$P(X) = \begin{cases} X & for\ X \geq 0 \\ 2^n + |X| & for\ X < 0 \end{cases}$$

$$-\frac{1}{2}Z < X < \frac{1}{2}Z$$

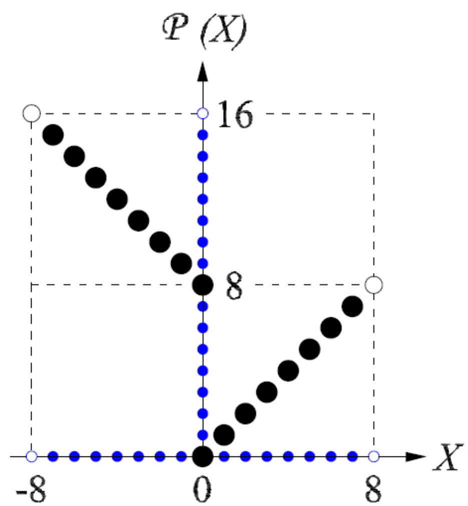
symmetric range

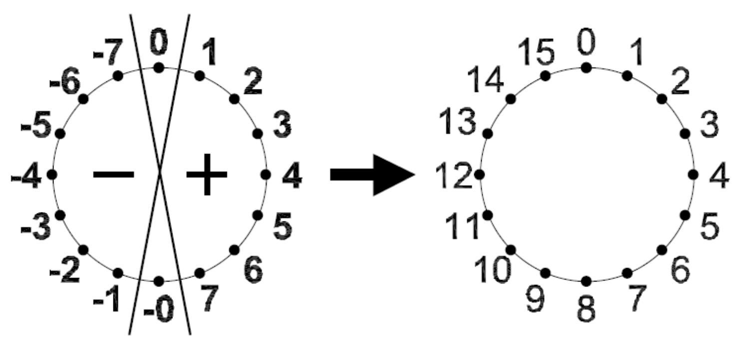
$$MSB = 0 \ \leftrightarrow X \geq 0$$
$$MSB = 1 \ \leftrightarrow X < 0$$

Sign bit= $\begin{cases} 0 & for\ X \geq 0 \\ 1 & for\ X \leq 0 \end{cases}$

$2\ zero\ representations\ (positive\ and\ negative)$

Sign and magnitude - example





X	$\mathcal{P}(X)$
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
-0	1000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111

Biased code – type 0

$$\boxed{A_0(X) = X + \frac{1}{2}Z}$$

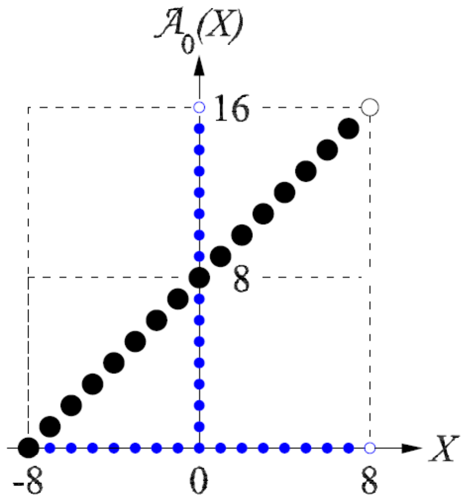
$$-\frac{1}{2}Z \le X < \frac{1}{2}Z$$ asymmetric range

$$\boxed{\begin{array}{l} MSB = 1 \leftrightarrow X \ge 0 \\ MSB = 0 \leftrightarrow X < 0 \end{array}}$$

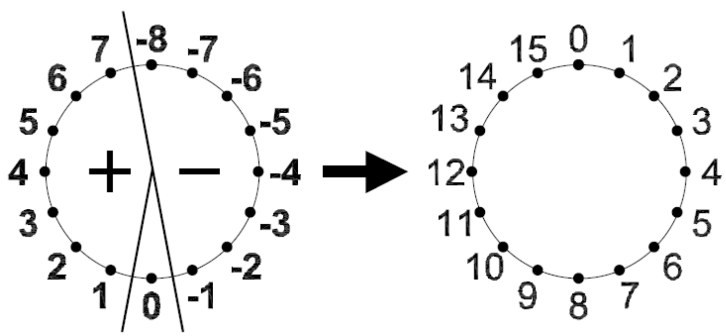
$$A_0(X) \equiv D(X)\ (\,mod\ \frac{1}{2}Z\,)$$

Biased code type 0 – example

$z=2$:



X	$\mathcal{A}_0(X)$
-8	0000
-7	0001
-6	0010
-5	0011
-4	0100
-3	0101
-2	0110
-1	0111
0	1000
1	1001
2	1010
3	1011
4	1100
5	1101
6	1110
7	1111



2's complement code

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

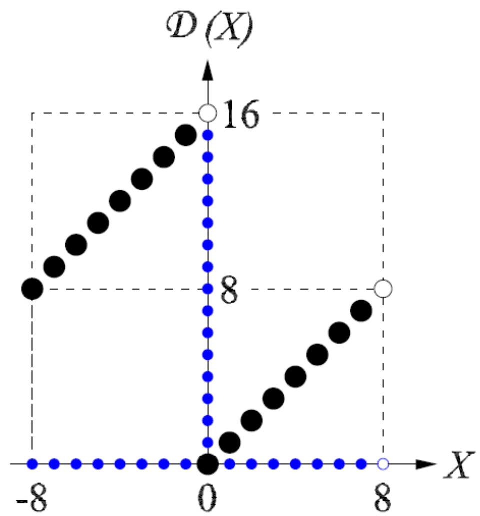
$$-\frac{1}{2}Z \leq X < \frac{1}{2}Z$$ asymmetric range

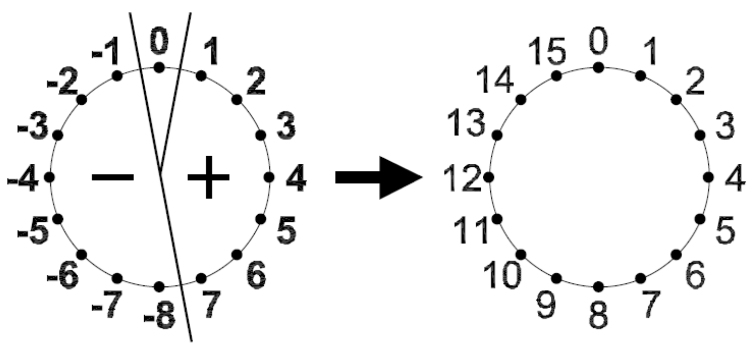
$$MSB = 0 \leftrightarrow X \geq 0$$
$$MSB = 1 \leftrightarrow X < 0$$

$$D(X) \equiv X \ (\ mod\ Z\ )$$

2's complement code - example

z=2:





X	$\mathcal{D}(X)$
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111

Opposite value in 2's complement code

- Algorithm:
  1. Record the number in binary system.
  2. Negate all number bits.
  3. Add the value 1 ("hot one")
- Example: Compute image of the number -5 (r = 2, M = 16, ε = 1).

$$
\begin{array}{r}
\underline{\ 0101} \\
1010 \\
\underline{+\ 0001} \\
1011
\end{array}
$$

Subtraction of signed numbers 2's complement code

- Solution:
  - addition of opposite number
- Example:
  - compute difference of following numbers: $10_{10} - 6_{10}$

$6_{10} = 00110_2$

$10_{10} = 01010_2$

$-6_{10} = 11010_2$

$$
\begin{array}{rr}
-6_{10} = & 11010_2 \\
+10_{10} = & 01010_2 \\
\hline
4_{10} = & {}^{1}00100_2
\end{array}
$$