Budapest University of Technology and Economics
Faculty of Transportation Engineering and Vehicle Engineering
Department of Control for Transportation and Vehicle Systems

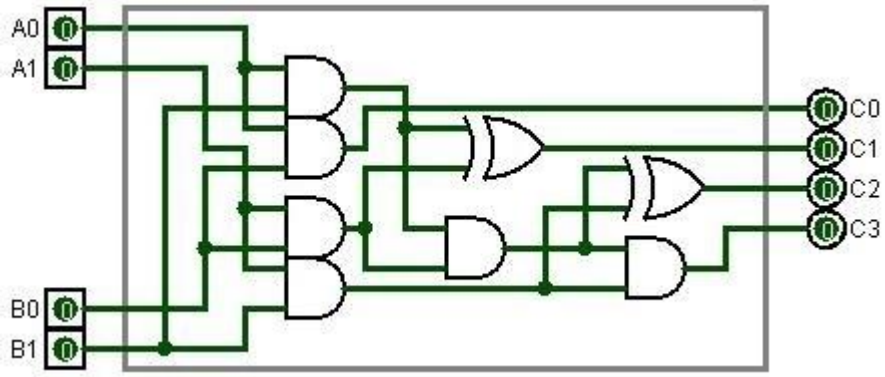
ARITHMETIC OPERATIONS, PART 3.

Lecture 5.

Binary Multiplication

- **Multiplication is usually (generally) realized by repeated addition!**

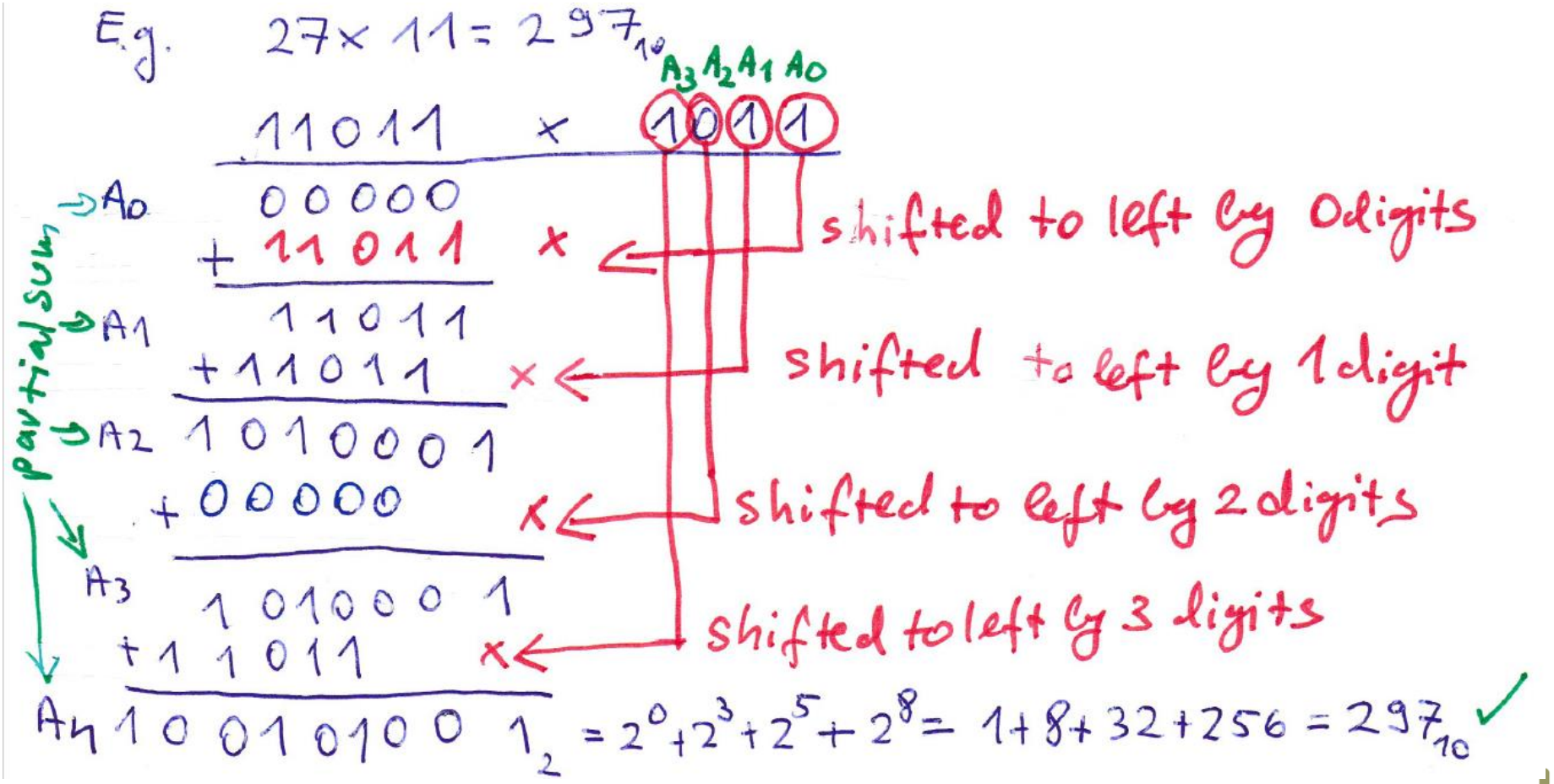
- Methods:
  - Multiplication by leftward shifting of the multiplicand,
  - Multiplication by rightward shifting of the partial sum,
  - Multiplication by using „ones row”,
  - Multiplication by grouping of digits.



other method: 2*2 bit multiplier, source: http://survivalcraftgame.wikia.com/wiki/Binary_Calculator_-_Multiplication

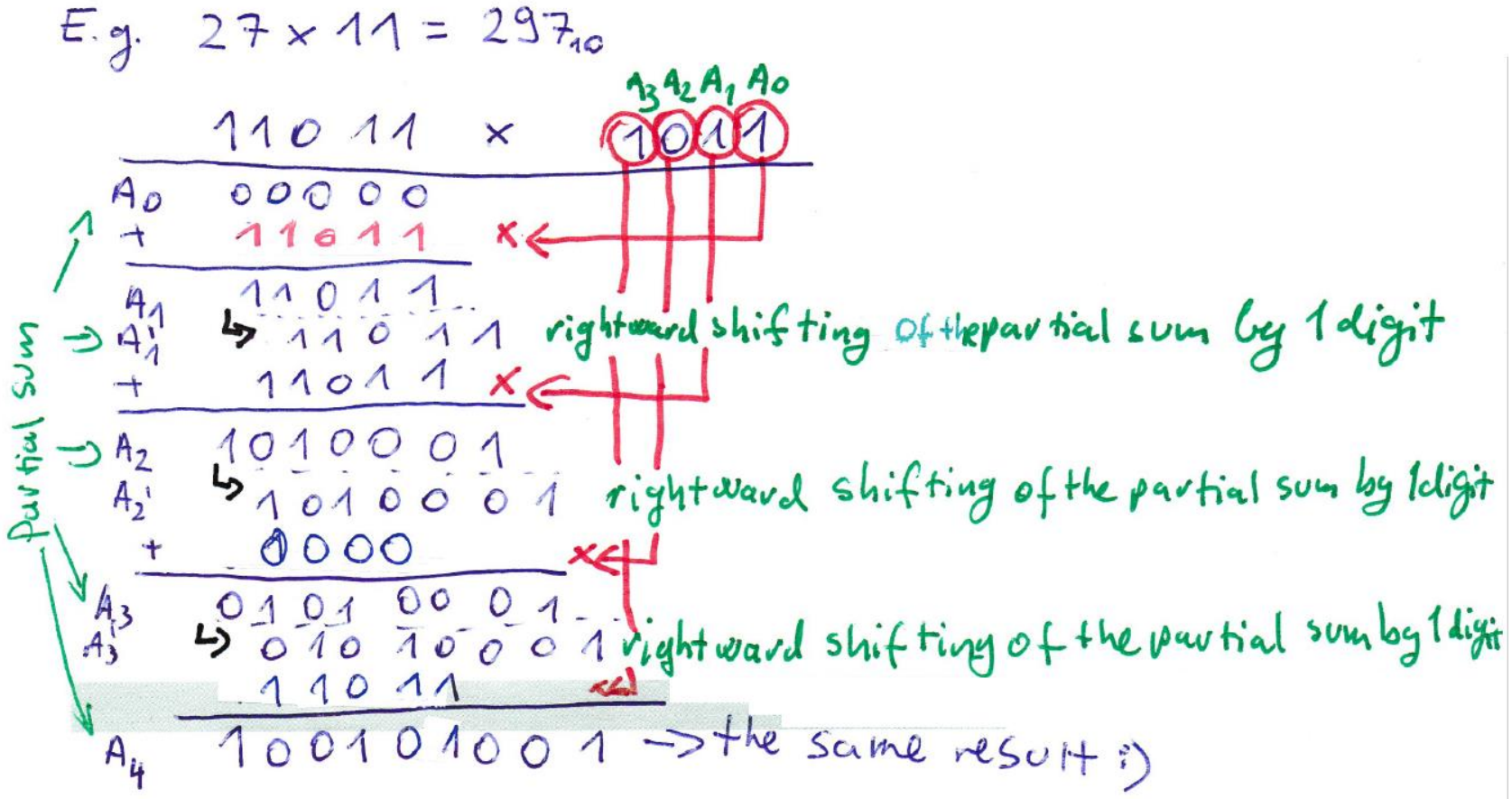
Binary Multiplication

- Multiplication by leftward shifting of the multiplicand:
  - if the length of the operands=n, the result will generates in n steps!



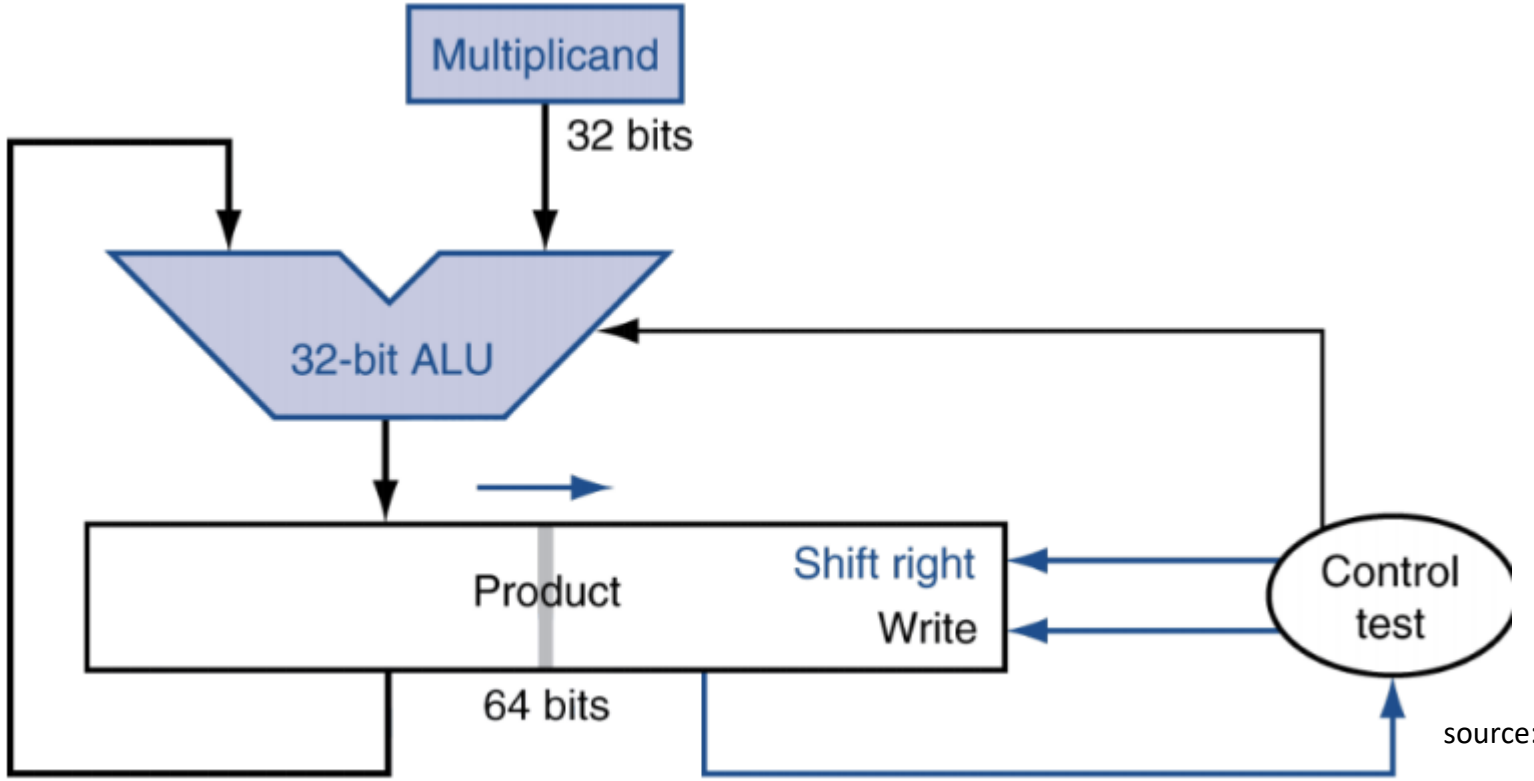
Binary Multiplication

- Multiplication by rightforward shifting of the partial sum:



Binary Multiplication

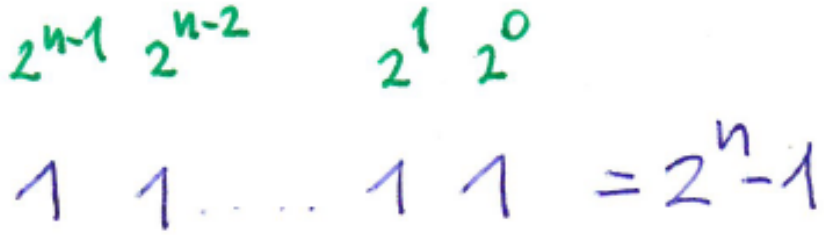
- Multiplication by rightforward shifting of the partial sum:



source: https://i.stack.imgur.com/NOrIk.png

Binary Multiplication

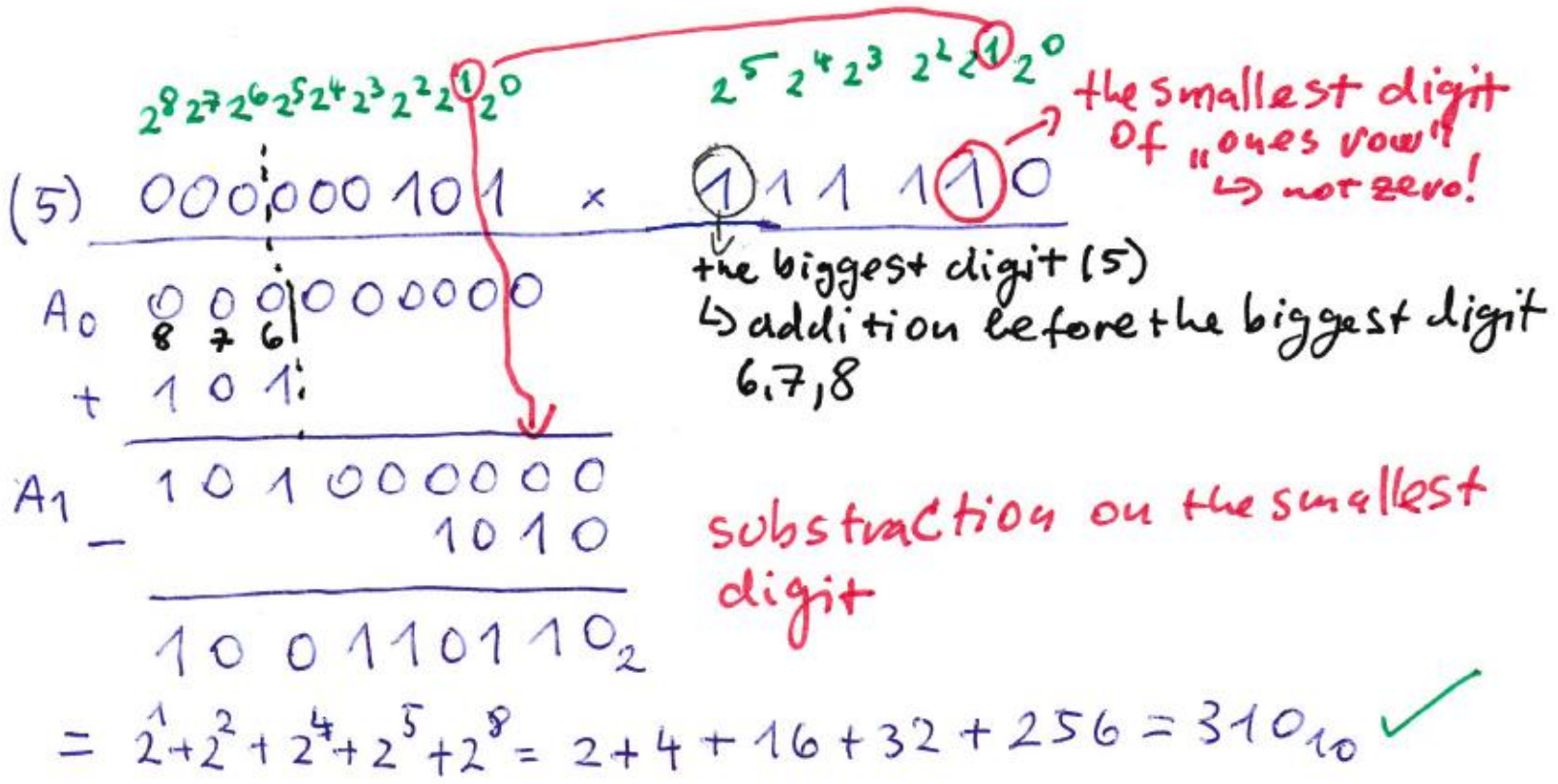
- Multiplication by using „ones row”:
  - ones row can be written as the followings:



- e.g:
  - $5*62 = 310_{10}$
  - $310 = 5*2^6 - 2*5 = 320 - 10$

Binary Multiplication

- Multiplication by using „ones row”:



Binary Multiplication

- Multiplication by using „grouping of digits:
  - Idea: we divide the multiplier to group of digits:
    - 00 ➡ 0
    - 01 ➡ 1
    - 10 ➡ 2
    - 11 ➡ 3
  - the partial sums will be 0…3 times of the multiplicand

E.g. - multiplicand: 6
- multiplier: 99

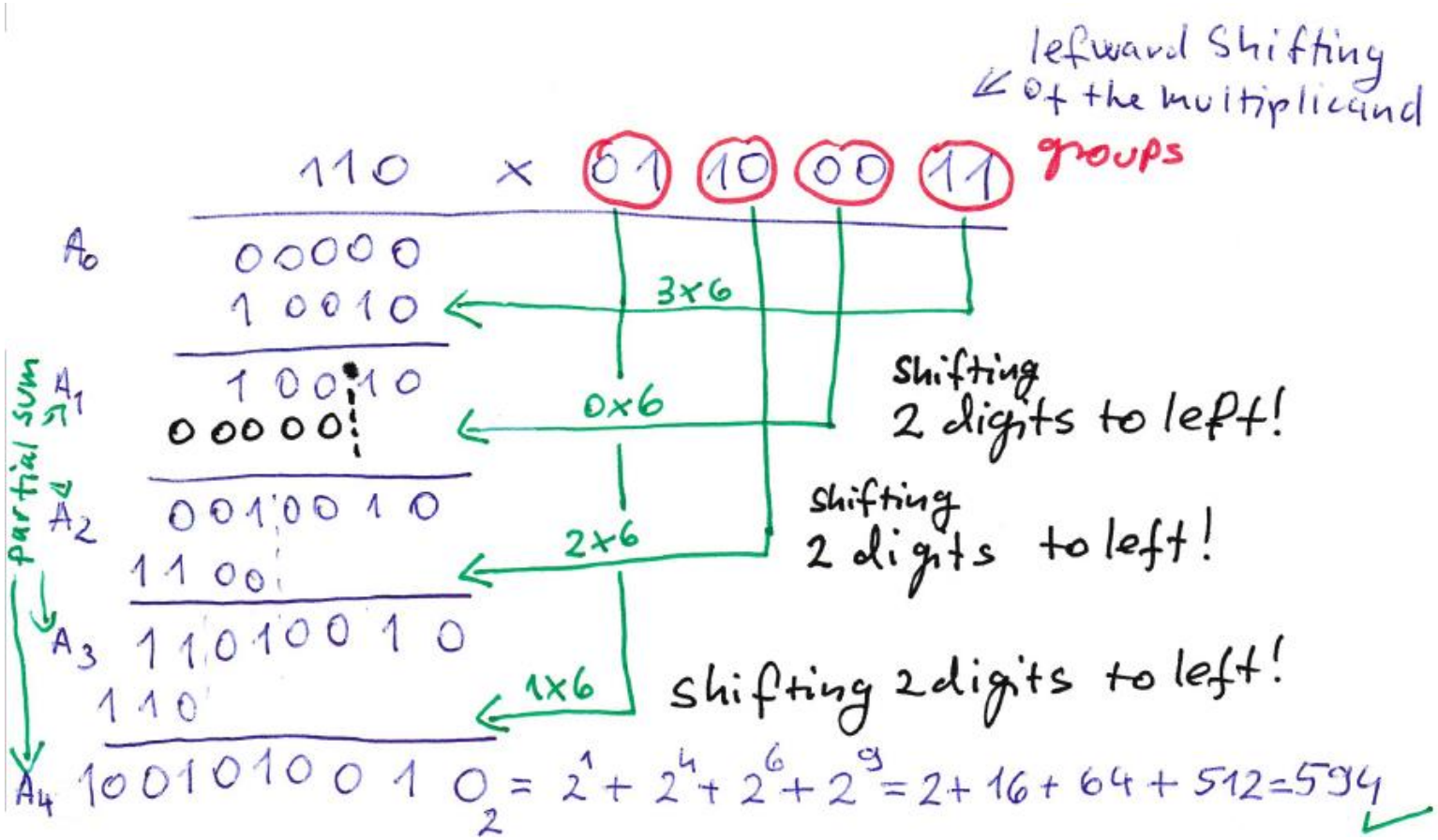
$6 \times 99 = 594_{10}$

$99_{10} = 1100011_2$

0 × 6 = 0 = 00000
1 × 6 = 6 = 00110
2 × 6 = 12 = 01100
3 × 6 = 18 = 10010

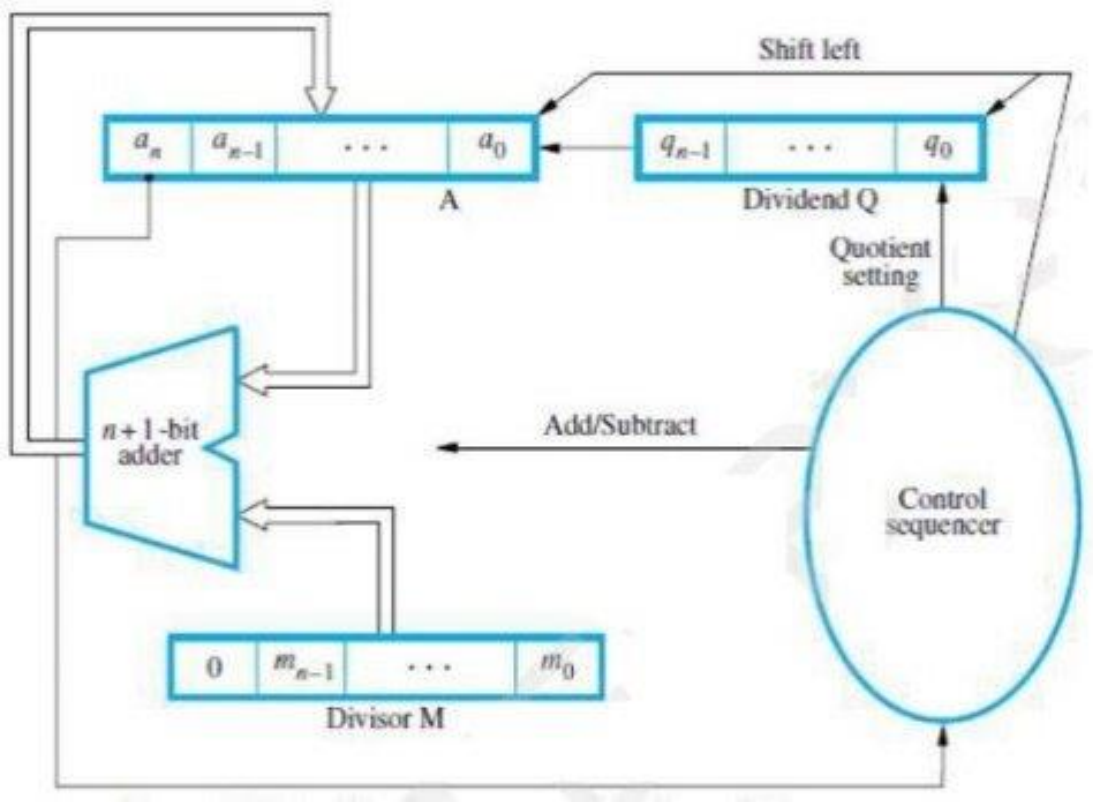
Binary Multiplication

- Multiplication by using „grouping of digits:



Binary Division

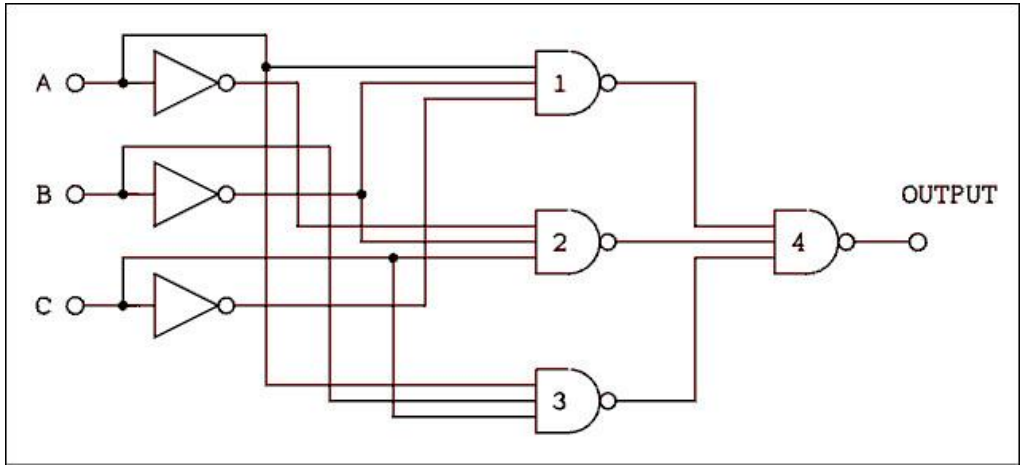
- Methods:
  - Division by repeated substraction,
  - Restoring division,
  - Division in complement code.



source:
https://www.researchgate.net/publication/326669218_Reversible_Signed_Division_for_Computing_Systems/figures?lo=1

Binary Division

- Division by repeated substraction:
  - Algorithm:
  - $A = dividend,\ B = divisor, A_i = partial\ sum$
  - $A_i = 2 * A_{i-1} - q_i * B$
  - $where:\ q_i = \begin{cases} 1, if\ \ 2 * A_{i-1} \geq B \\ 0, if\ 2 * A_{i-1} < B \end{cases}$



source: http://golfinamigos.com/binary-division-circuit-diagram/

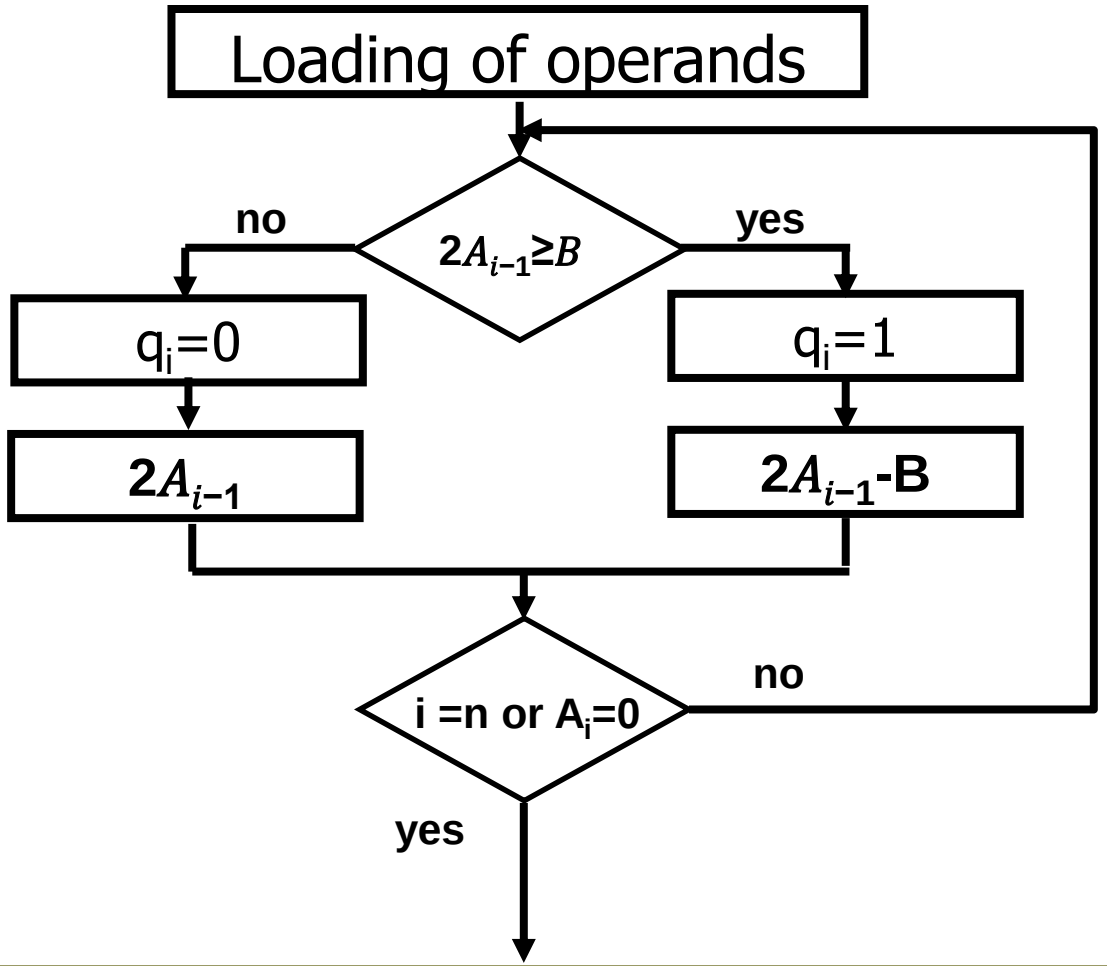
- $\frac{A}{B} = Q + \frac{r_n}{B}$, where Q = quotient, $r_n = remainder\ after\ the\ step\ n.$

Binary Division

- $A_i = 2 * A_{i-1} - q_i * B$
- $A_1 = 2 * A_0 - q_0 * B$
- $A_2 = 2 * A_1 - q_1 * B = 2^2 * A_0 - 2 * q_1 * B - q_2 * B$
- $A_3 = 2 * A_2 - q_3 * B = 2^3 * A_0 - 2^2 * q_1 * B - 2 * q_2 * B - q_3 * B$
- …
- $A_n = 2^n * A_0 - B * (2^{n-1} * q_1 + 2^{n-2} * q_2 + \cdots + q_n)$
- $* \frac{2^{-n}}{B}$
- $\frac{A*2^{-n}}{B} = \frac{A_0}{B} - B * (2^{-1} * q_1 + 2^{-2} * q_2 + \cdots + 2^{-n} * q_n), where\ A_0 = A, and\ (\dots) = local\ values\ of\ the\ quotient$
- $\frac{A}{B} = Q + \frac{A*2^{-n}}{B}$
- $\frac{A}{B} = Q + \frac{r_n}{B}$, where Q = quotient, $r_n = remainder\ after\ the\ step\ n.$

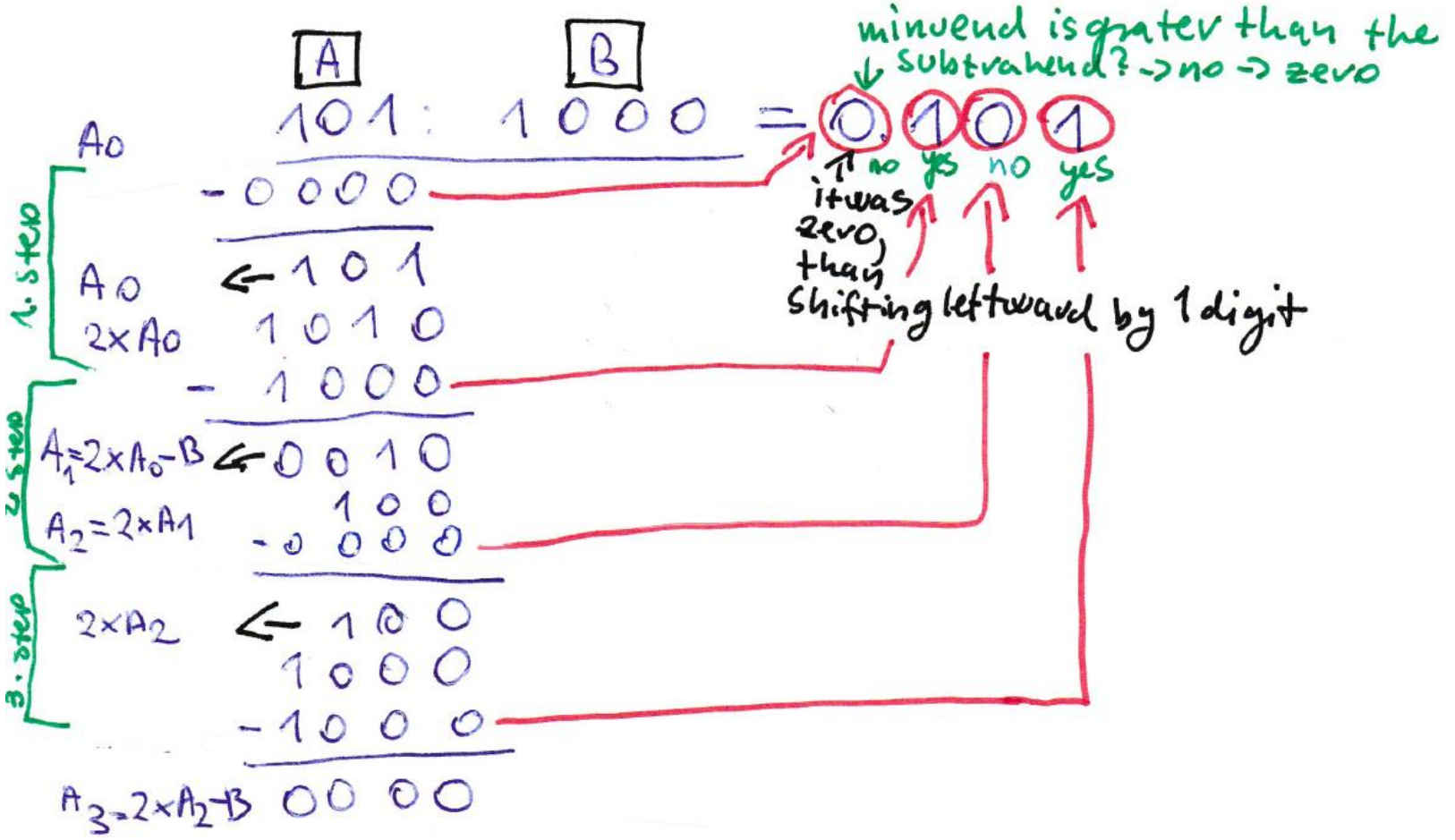
Binary Division

- Algorithm of the division with repeated substraction in the ALU



Binary Division

- Example: 5/8



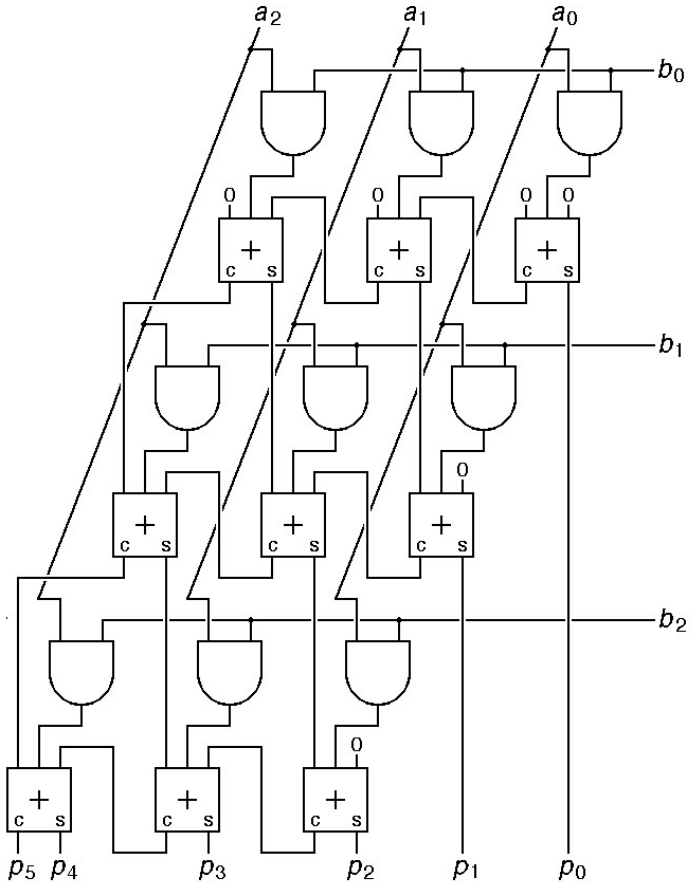
Binary Division

- Division by restoring division:
  - Algorithm:
  - we make the substraction in every step, and if the difference is negative, we have to add the divisor to the partial sum, than it has to shift leftward the partial sum
  - if the difference is positive, the restoring is cancelled
  - the substraction means an addition is 2's complement

  - $A = dividend,\ B = divisor, A_i = partial\ sum$
  - $B_k = 2's\ complement\ code\ of\ B$

  - $A + B_k = A + 2^k - |B| = (A - B) + 2^k$

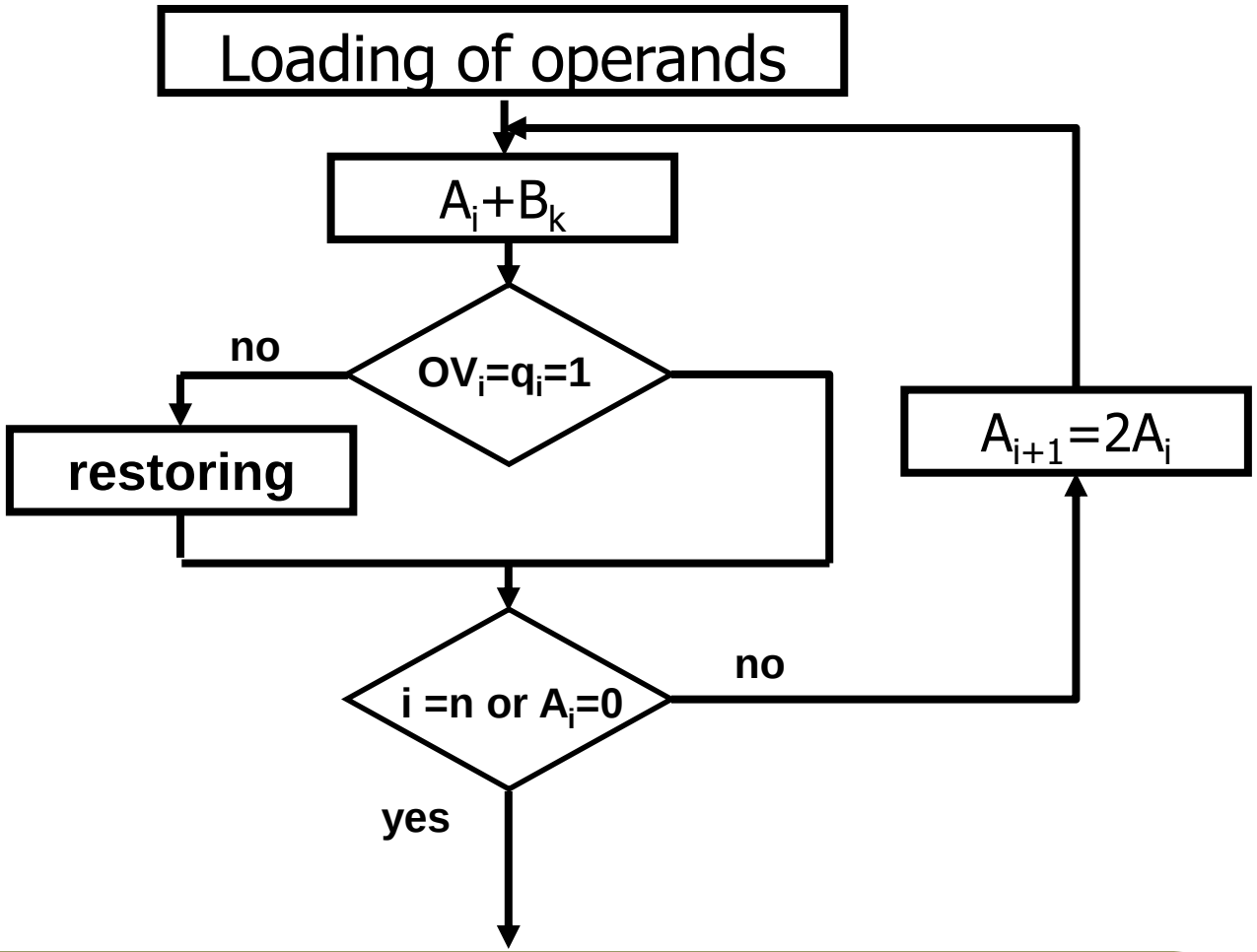
  - $A_i = 2 * A_{i-1} - q_i * B, where\ q_i = overflow$



source: http://golfinamigos.com/binary-division-circuit-diagram/

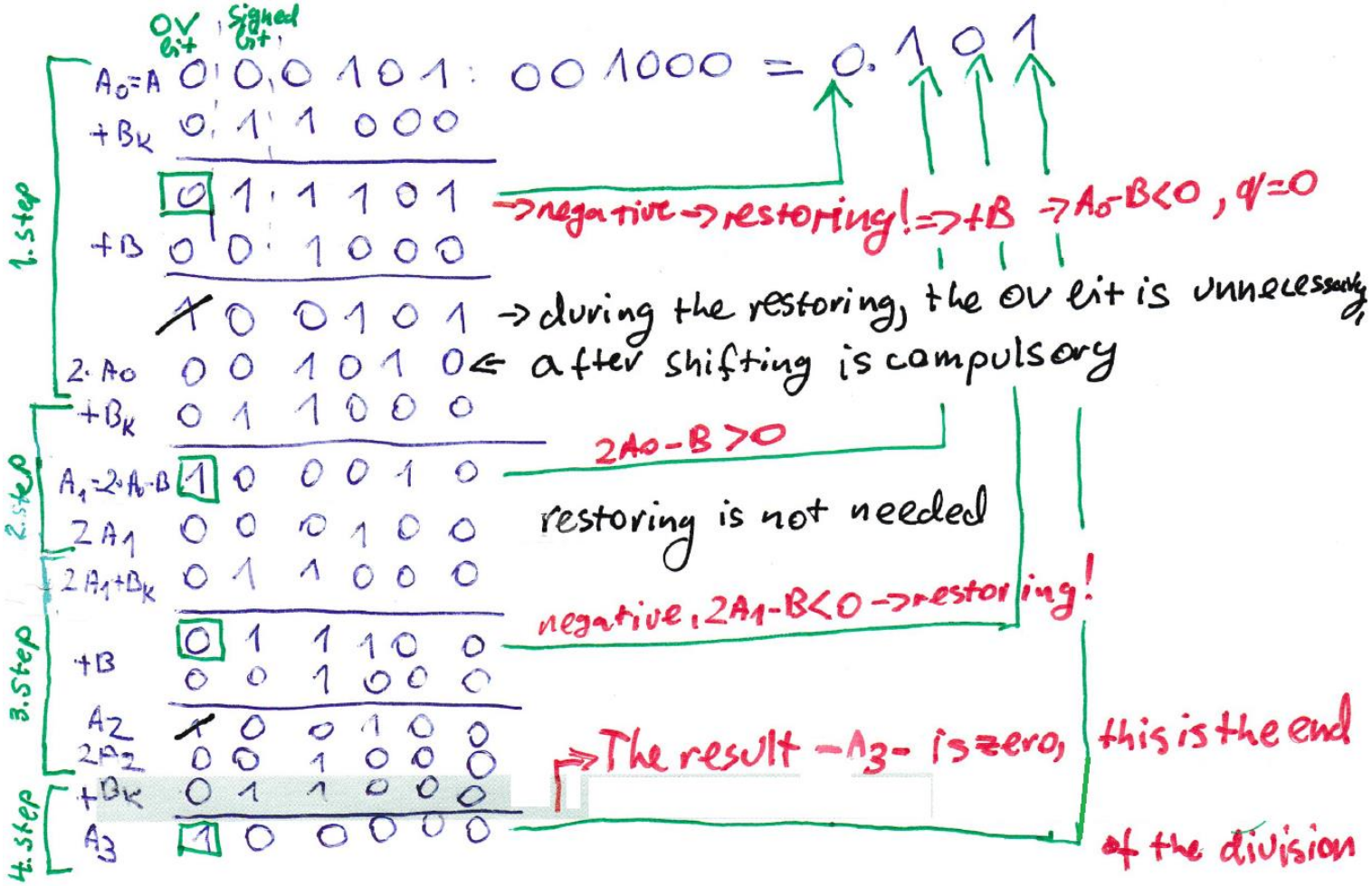
Binary Division

- Algorithm of the restoring division in the ALU



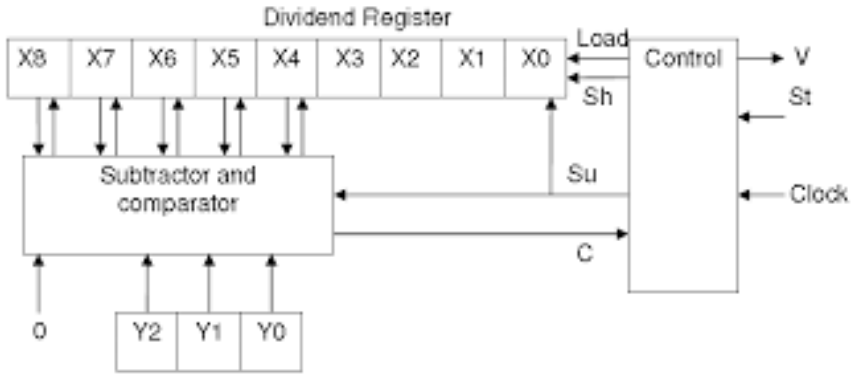
Binary Division

- Example: 5/8



Binary Division

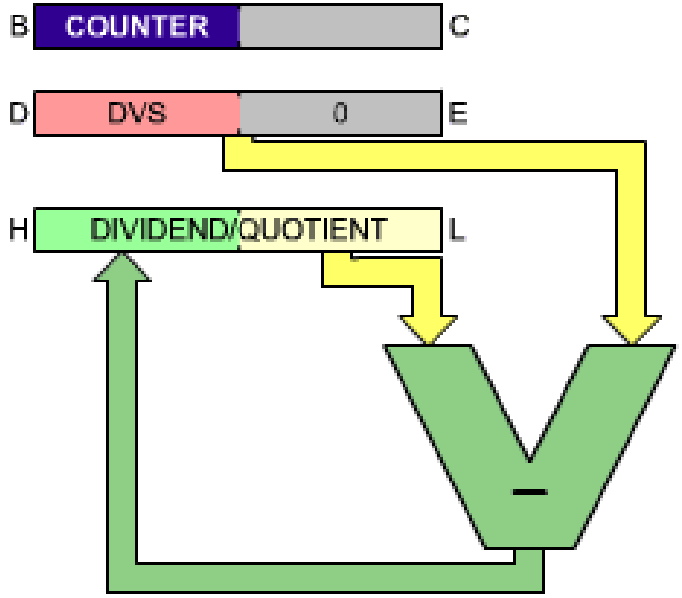
- Division in complement code :
  - Algorithm:
  - we make the addition and substraction in turn, and if the signed bits are the same, then the actual bit of the quotient is 1 and we have to make a substraction, if the signed bit is different, than the actual bit of the quotient is 0, and we have to make an addition in the next step.
  - $A = dividend,\ B = divisor, A_i = partial\ sum$
  - $A_1 = \begin{cases} 2 * A_0 - B\ and\ q_1 = 1, if\ \ sign(A_0) = sign(B) \\ 2 * A_0 + B\ and\ q_1 = 0, if\ \ sign(A_0) = sign(B) \end{cases}$
  - …
  - $A_i = \begin{cases} 2 * A_{i-1} - B\ and\ q_i = 1, if\ \ sign(A_{i-1}) = sign(B) \\ 2 * A_{i-1} + B\ and\ q_i = 0, if\ \ sign(A_{i-1}) = sign(B) \end{cases}$



source: http://needpixies.com/binary-divider-schematic.html

Binary Division

- $A_i = 2 * A_{i-1} + (1 - 2 * q_i) * B$
- $A_1 = 2 * A_0 + (1 - 2 * q_1) * B$
- $A_2 = 2 * A_1 + (1 - 2 * q_2) * B =$
- $= 2^2 * A_0 + 2 * (1 - 2 * q_1) * B + (1 - 2 * q_2) * B$
- $\dots$
- $A_n = 2^n * A_0 + B * [2^{n-1}(1 - 2q_1) + 2^{n-2}(1 - 2q_2) + \cdots + (1 - 2q_n)]$

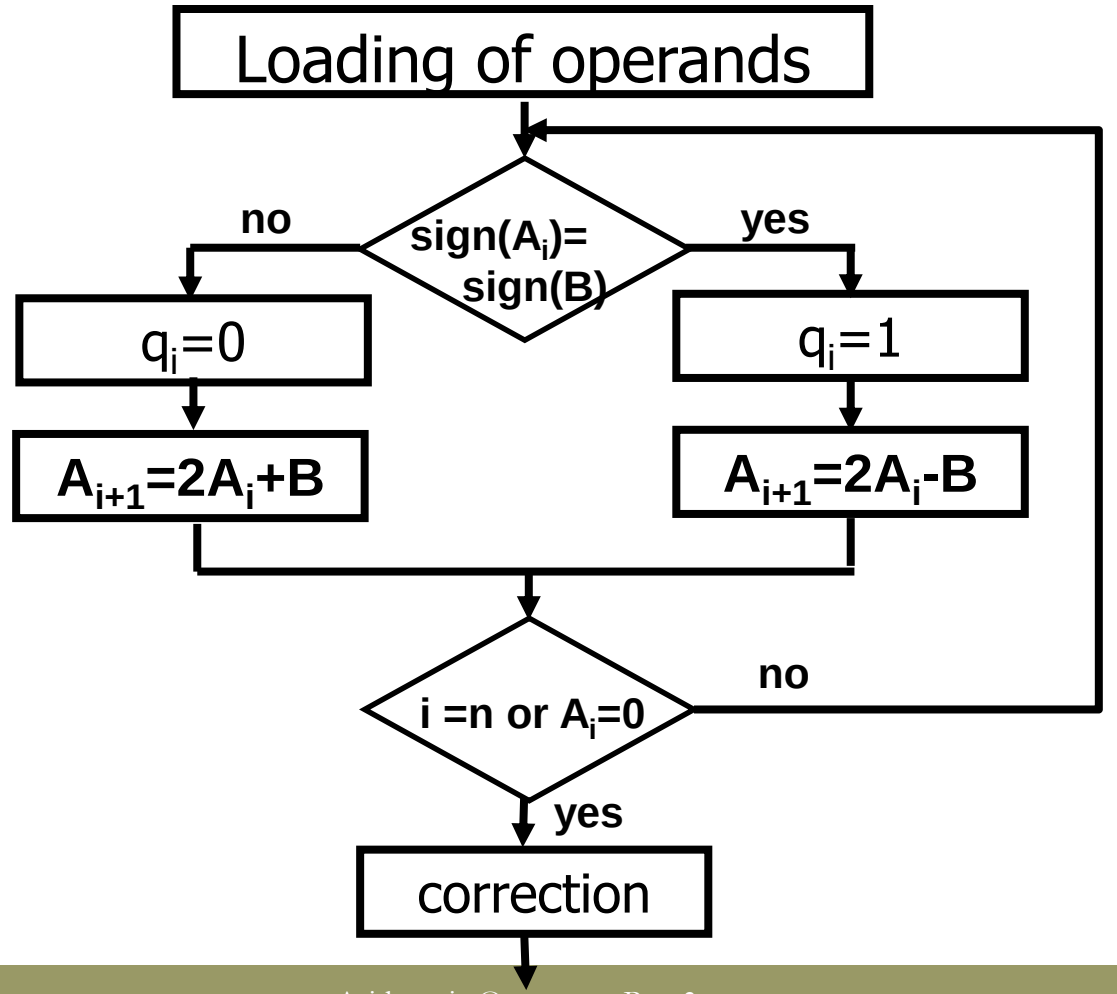


source: http://rvbelzen.tripod.com/z80prgtemp/z80prg03.htm

- $* \frac{2^{-n}}{B}$
- $\frac{A*2^{-n}}{B} = \frac{A_0}{B} + (2^{-1} + 2^{-2} + \cdots + 2^{-n}) - 2(q_1 2^{-1} + q_2 2^{-2} + \cdots + q_n 2^{-n})$
- $\frac{A}{B} = -1 + 2^{-n} + 2\sum_{i=1}^{n} q_i 2^{-i} + \frac{2^{-n} A_n}{B}$

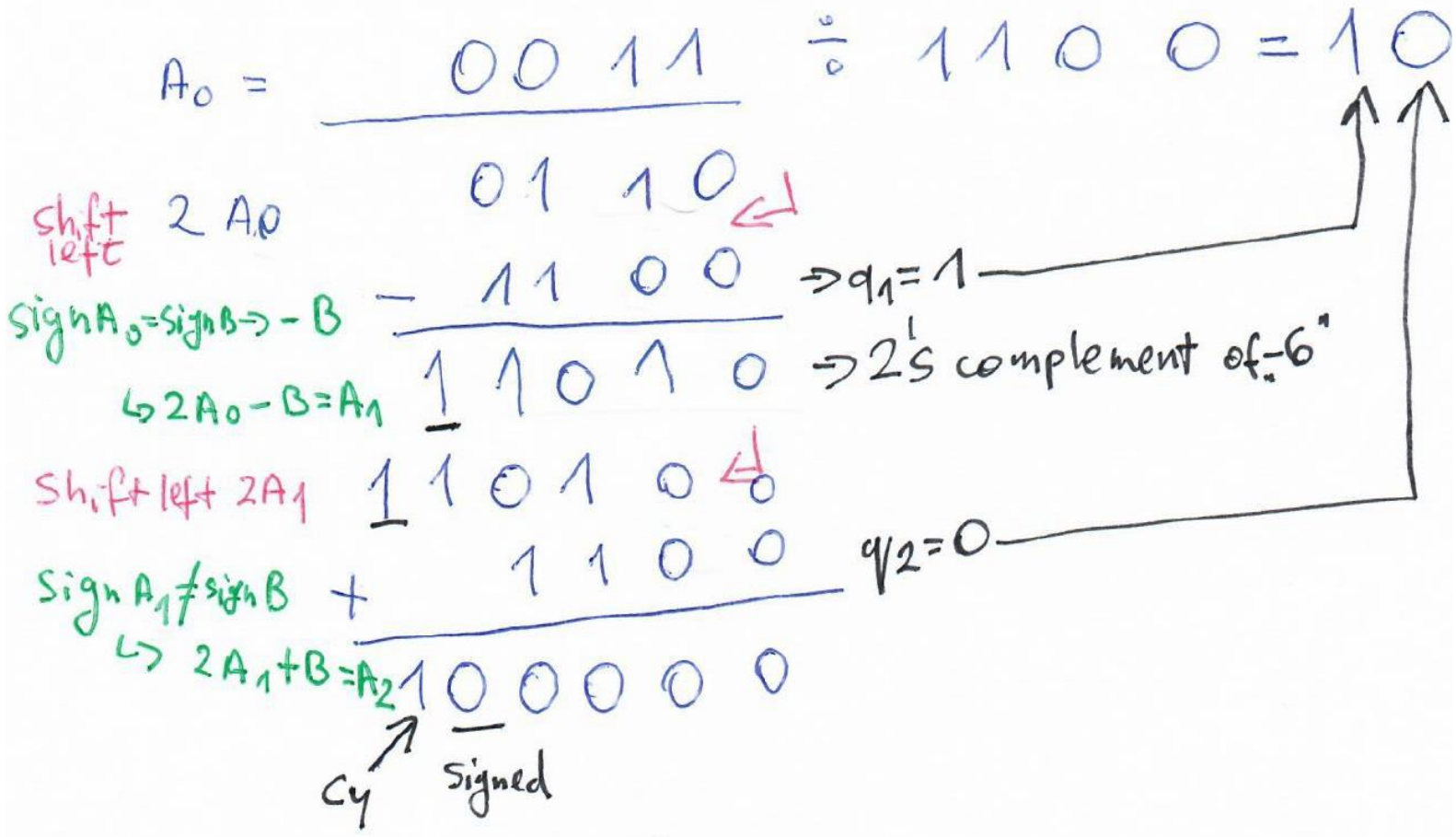
Binary Division

- Algorithm of the complement coded division in the ALU



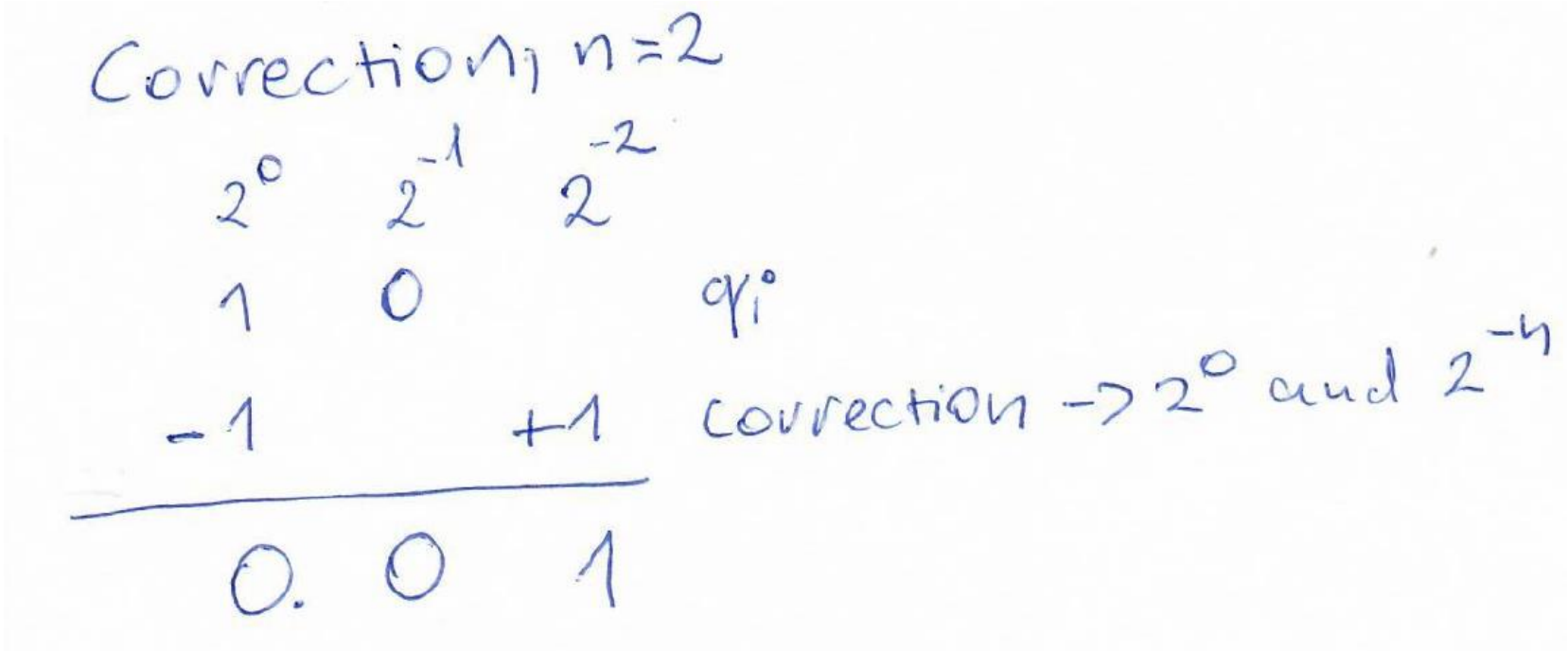
Binary Division

- Example: 3/12



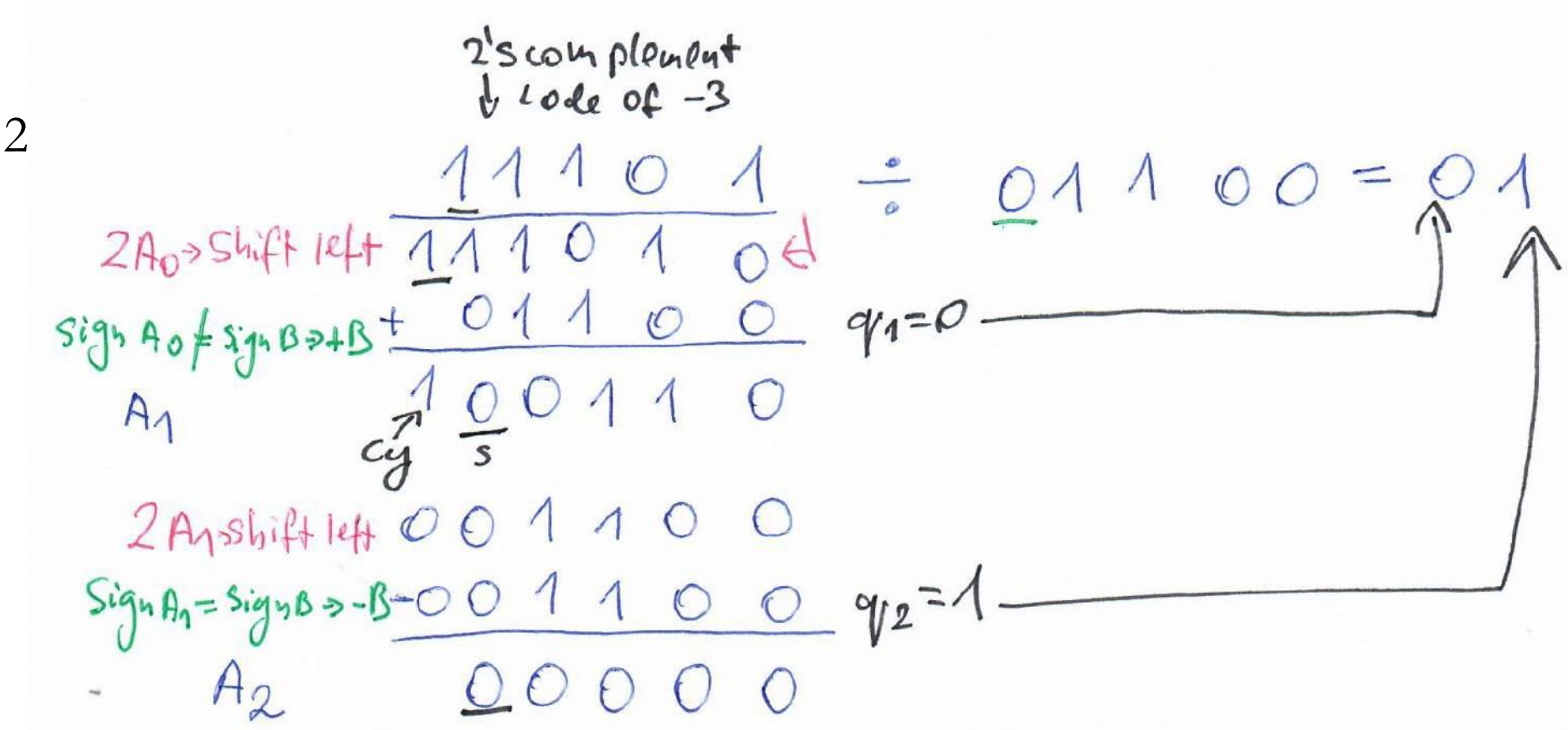
Binary Division

- Example: 3/12



Binary Division

- Example: -3/12



Binary Division

- Example: -3/12

correction $n=2$

2^0	2^{-1}	2^{-2}
0	1	
-1		+1
1.1		1

$\rightarrow$ qi correction

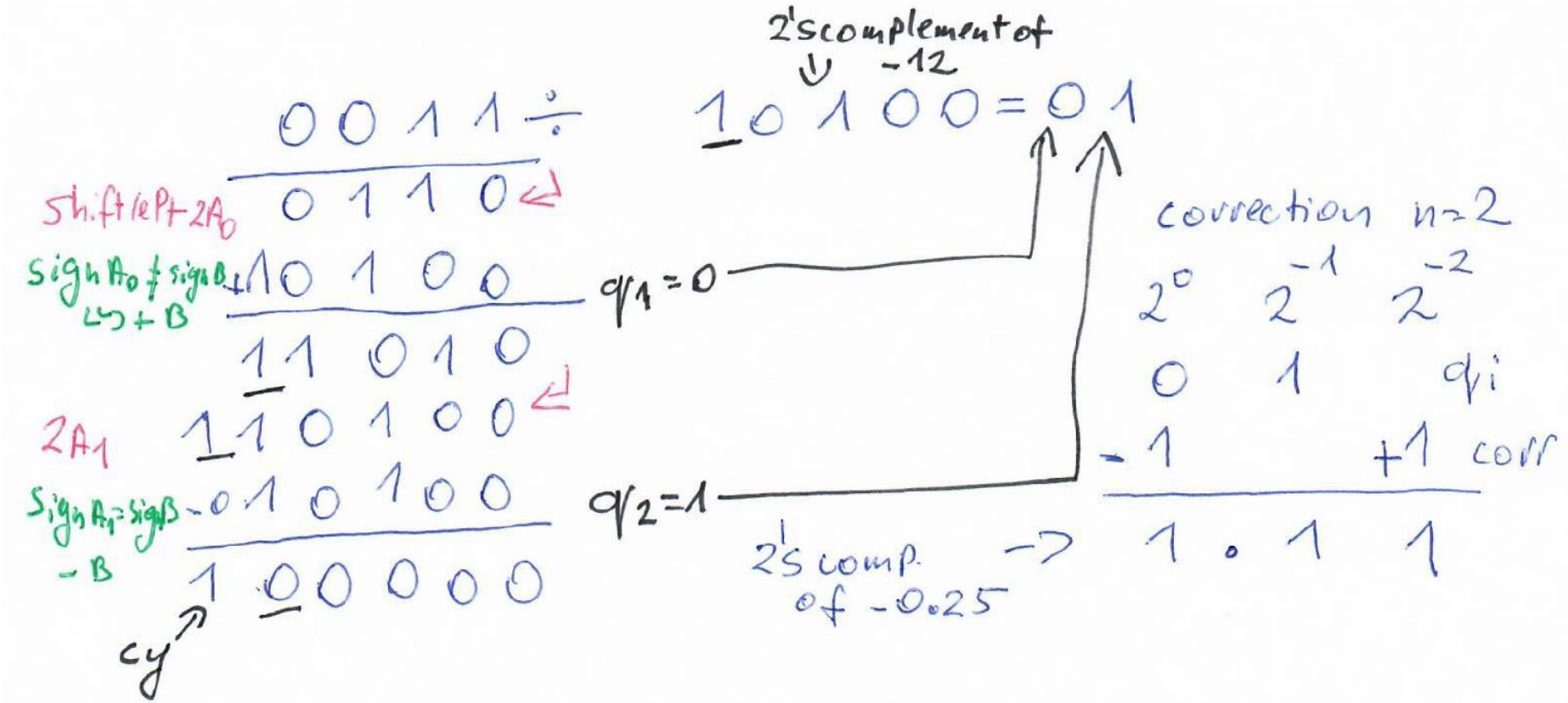
2^1_s complement of -0.25

$= 2 - 0,25 = 1.75$

$$\begin{array}{r} 1\,0.0\,0\,0\,0\ (2) \\ -\quad 0.0\,1\ (0.25) \\ \hline 0\,1.1\,1\,0 \end{array}$$

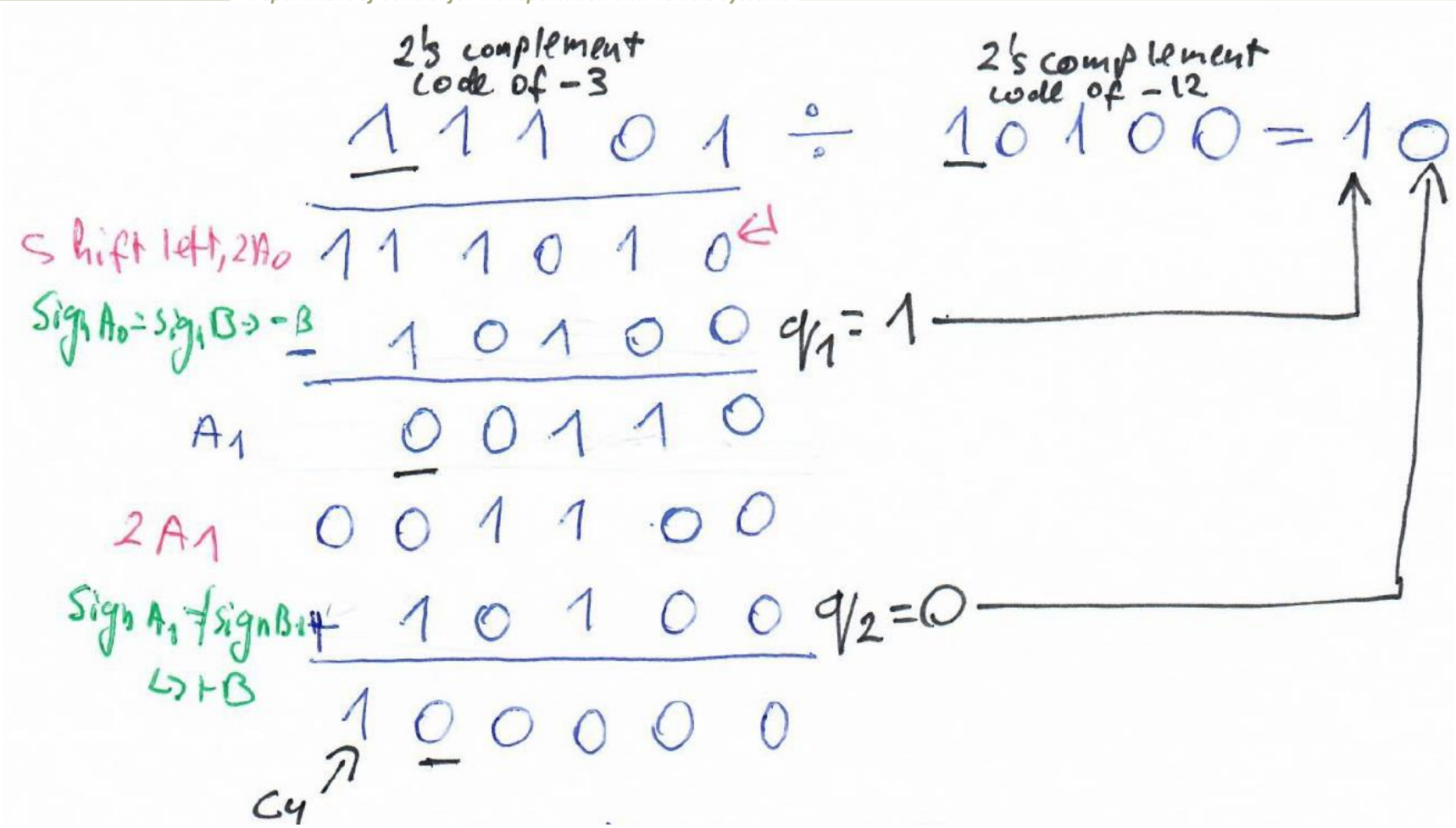
Binary Division

- Example: 3/-12



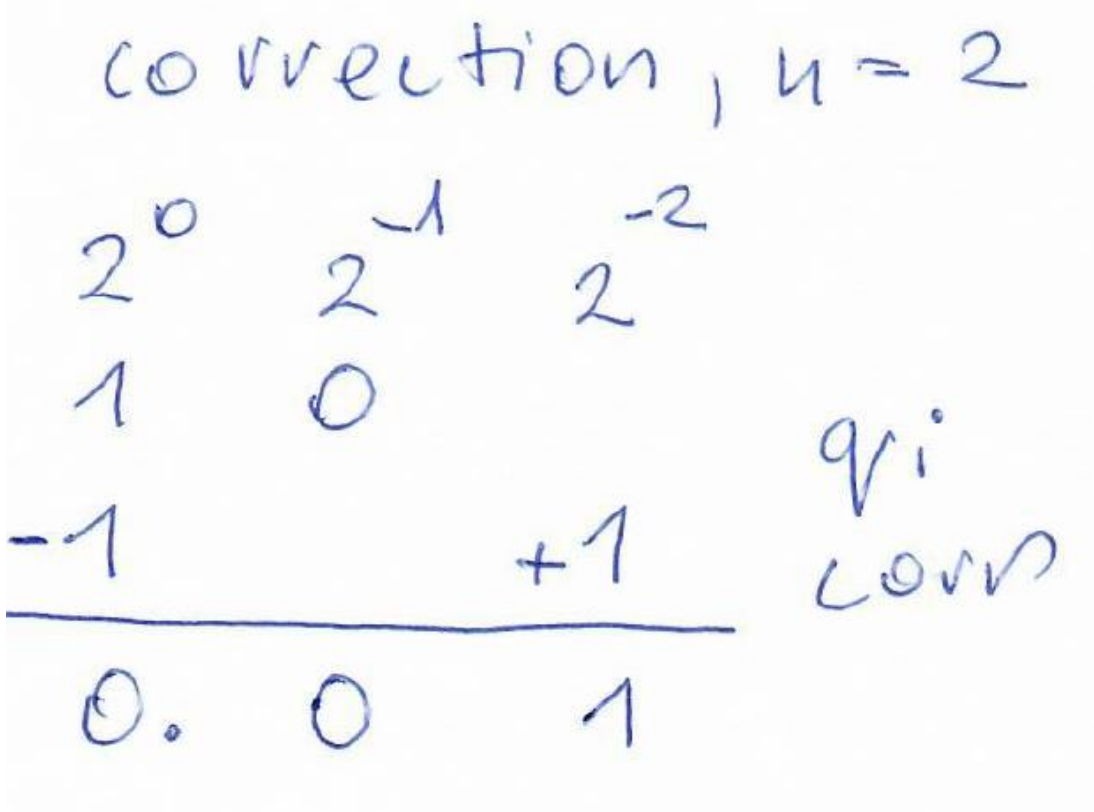
Binary Division

- Example: -3/-12



Binary Division

- Example: -3/-12



End of Lecture 5.

Thank you for your attention!

Budapest University of Technology and Economics
Faculty of Transportation Engineering and Vehicle Engineering
Department of Control for Transportation and Vehicle Systems

REALIZATION OF BINARY OPERATIONS

Lecture 6.

Binary Division

- https://www.youtube.com/watch?v=K79wfflmLNo

End of Lecture 6.

Thank you for your attention!