

Korrutamisalgoritmid

Täisarvude korrutamisel annavad N-bitised operandid $2*N$ -bitise tulemuse. Sellest ka vajadus jätta alles noorimad järgud.

Murdarvude korrutamisel võib noorimad järgud ära lõigata (või kasutada ümardamist).

Alates noorematest järkudest (pos. arvud), sobib hästi täisarvudele.

Rg1 liidetakse Rg3 ülemisse ossa (NB! Rg3 on $2*N$ bitti!)
Rg2 & Rg3 igas tsükklis $\gg 1$

Rg1 = 13 = 00001101, Rg2 = 24 == 00011000

	Rg2		Rg3
1.	00011000		00000000
	^	00000000
2.	00001100		00000000
	^	00000000
3.	00000110		00000000
	^	00000000
4.	00000011		00000000
	^	+Rg1 [00001101...]	00001101000
5.	00000001		000001101000
	^	+Rg1 [00001101...]	000100111000
6.	00000000		0000100111000
	^	0000100111000
7.	00000000		00000100111000
	^	00000100111000
8.	00000000		000000100111000
	^	000000100111000
			0000000100111000 == 312 (=24*13)

Rg1 = 24 = 00011000, Rg2 = 13 = 00001101

	Rg2		Rg3
1.	00001101		00000000
	^	+Rg1 [00011000]	00011000
2.	00000110		000011000
	^	000011000

```

3. 00000011      0000011000
   ^ +Rg1 [00011000..]
      0001111000

4. 00000001      00001111000
   ^ +Rg1 [00011000...]
      00100111000

5. 00000000      000100111000
   ^ .... 000100111000

6. 00000000      0000100111000
   ^ .... 0000100111000

7. 00000000      00000100111000
   ^ .... 00000100111000

8. 00000000      000000100111000
   ^ .... 000000100111000

                                0000000100111000 == 312 (=13*24)

```

Murdarvud, lisabitid ümardamiseks

Rg1 = 0.6875000 = 0.1011000, Rg2 = 0.7812500 = 0.1100100

	Rg2		Rg3	
1.	01100100		00000000	
	^	00000000	
2.	00110010		000000000	
	^	000000000	
3.	00011001		0000000000	
	^	+Rg1 [01011000..]	0101100000	
4.	00001100		00101100000	
	^	00101100000	
5.	00000110		000101100000	
	^	000101100000	
6.	00000011		0000101100000	
	^	+Rg1 [01011000.....]	0110001100000	
7.	00000001		00110001100000	
	^	+Rg1 [01011000.....]	10001001100000	
8.	00000000		010001001100000	NB! Loogiline nihe - positiivsed arvud!
	^	010001001100000	

Rg3 = 010001001100000 = 0.1000100_1100000 - lisanihet pole, on ümardamine

0.6875000*0.7812500=0.5371093

0.1000100 == 0.5312500 (-1.1%)

0.1000101 == 0.5390625 (+0.4%)

Alates vanematest järkudest (pos. arvud), sobib hästi murdarvudele.

Rg1 liidetakse otse Rg3 (mõlemad N bitti)
 Rg1 igas tsükli >>1, Rg2 igas tsükli <<1

$$Rg1 = 0.6875 = 0.1011000, \quad Rg2 = 0.75 = 0.1100000$$

	Rg1	Rg2		Rg3
1.	01011000	01100000 ^	00000000
2.	00101100	11000000 ^	+Rg1	[00101100] 00101100
3.	00010110	10000000 ^	+Rg1	[00010110] 01000010
4.	00001011	00000000 ^	01000010
5.	00000101	00000000 ^	01000010
6.	00000010	00000000 ^	01000010
7.	00000001	00000000 ^	01000010
8.	00000000	00000000 ^	01000010 == 0.1000010 = 0.515625

$$Rg1 = 0.75 = 0.1100000, \quad Rg2 = 0.6875 = 0.1011000$$

	Rg1	Rg2		Rg3
1.	01100000	01011000 ^	00000000
2.	00110000	10110000 ^	+Rg1	[00110000] 00110000
3.	00011000	01100000 ^	00110000
4.	00001100	11000000 ^	+Rg1	[00001100] 00111100
5.	00000110	10000000 ^	+Rg1	[00000110] 01000010
6.	00000011	00000000 ^	01000010
7.	00000001	00000000 ^	01000010
7.	00000000	00000000 ^	01000010 == 0.1000010 = 0.515625

Negatiivsete arvude vahetu korrutamine

Alates noorematest järkudest (pos. arvud), sobib hästi täisarvudele.

Rg1 liidetakse Rg3 ülemisse ossa (NB! Rg3 on 2*N bitti!)

Rg2 & Rg3 igas tsükli >>1

13*24=312

13 – 00001101 / -13 – 11110011

24 – 00011000 / -24 – 11101000

Rg1 = 24 = 00011000, Rg2 = -13 = 11110011

	Rg2		Rg3	
1.	11110011		00000000	
	^	+Rg1	[00011000]	
			00011000	
2.	11111001		000011000	
	^	+Rg1	[00011000.]	
			001001000	
3.	11111100		0001001000	
	^	0001001000	
4.	11111110		00001001000	
	^	00001001000	
5.	11111111		000001001000	
	^	+Rg1	[00011000....]	
			000111001000	
6.	11111111		0000111001000	
	^	+Rg1	[00011000.....]	
			0010011001000	
7.	11111111		00010011001000	
	^	+Rg1	[00011000.....]	
			00101011001000	
8.	11111111		000101011001000	korrektsioon?
	^	-Rg1	[11101000.....]	
			11111011001000	
			111111011001000 == -0000000100111000 = -312	

Rg1 = -13 = 11110011, Rg2 = -24 = 11101000

	Rg2		Rg3
1.	11101000		00000000
	^	00000000

```

2.  11110100      000000000
    ^      ....  000000000

3.  11111010      0000000000
    ^      ....  0000000000

4.  11111101      00000000000
    ^ +Rg1 [11110011...]
      11110011000

5.  11111110      111110011000
    ^      ....  111110011000

6.  11111111      1111110011000
    ^ +Rg1 [11110011.....]
      1110111111000

7.  11111111      11110111111000
    ^ +Rg1 [11110011.....]
      11101010111000

8.  11111111      111101010111000   korrektsioon?
    ^ -Rg1 [11110011.....]
      000000100111000

      0000000100111000 = 312

```

Radix-4 (2 järku korruga)

Rg1 liidetakse/lahutatakse Rg3 ülemisse ossa (Rg3 2*N bitti)
Rg2 & Rg3 igas tsükliis >>2 (aritmeetiline nihe!)

Rg1 = 24 = 00011000, Rg2 = 13 = 00001101

	Rg2		Rg3	
1.	00001101 ^^	+Rg1	00000000 [00011000] 00011000	
2.	00000011 ^^	-Rg1	0000011000 [11101000..] 1110111000	NB! Vahetulemus negatiivne
3.	00000000 ^^	+Rg1	111110111000 [00011000....] 000100111000	Aritmeetiline nihe 3==4-1
4.	00000000 ^^	00000100111000 00000100111000	
			0000000100111000 == 312 (=13*24)	

Rg1 = 13 = 00001101, Rg2 = -24 = 11101000

	Rg2		Rg3
1.	11101000 ^^	00000000 00000000
2.	11111010 ^^	+2*Rg1	0000000000 [00011010..] 0001101000
3.	11111110 ^^	+2*Rg1	000001101000 [00011010....] 001000001000 001000001000
4.	11111111 ^^	-Rg1	00001000001000 [11110011.....] 1111011001000 111111011001000 = -0000000100111000 = -312

Rg1 = -13 = 11110011, Rg2 = -24 = 11101000

	Rg2		Rg3
1.	11101000 ^^	00000000 00000000
2.	11111010 ^^	+2*Rg1	0000000000 [11100110..] 1110011000
3.	11111110 ^^	+2*Rg1	111110011000 [11100110....] 110111111000
4.	11111111 ^^	-Rg1	11110111111000 [00001101.....] 00000100111000 0000000100111000 = 312