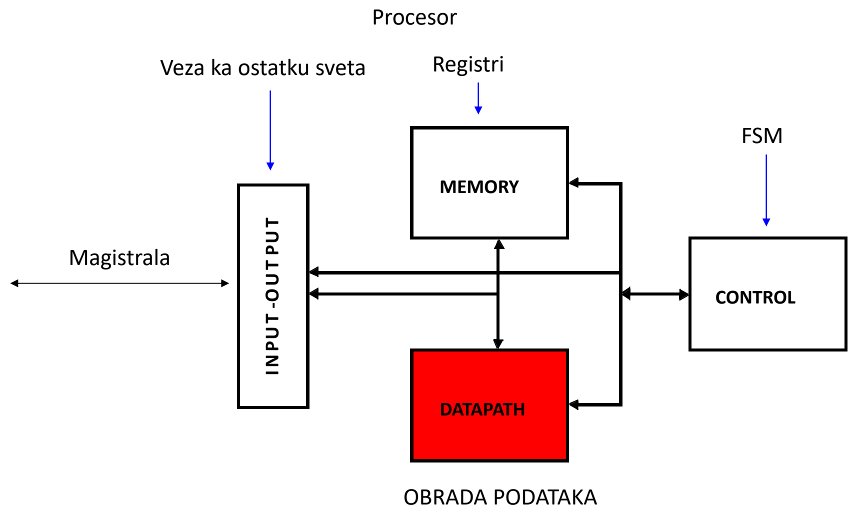
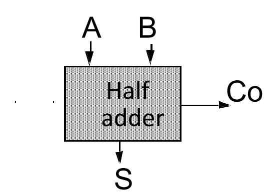
Digitalna elektronika
Aritmetička kola

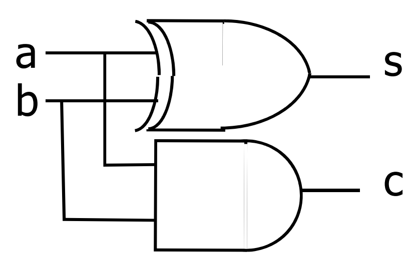


Poređenje, Sabiranje, Množenje, Deljenje, Pomeranje, ...

Digitalna elektronika
Aritmetička kola – Sabirači - Polusabirač

a b	c	s
0 0	0	0
0 1	0	1
1 0	0	1
1 1	1	0

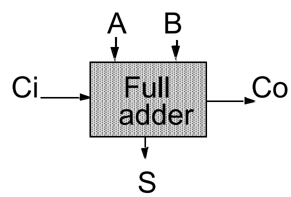




Digitalna elektronika

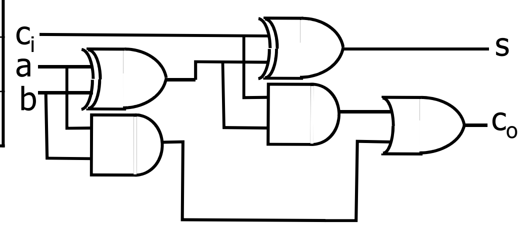
Aritmetička kola – Sabirači – Potpuni sabirač

c_i a b	c_o	s
0 0 0	0	0
0 0 1	0	1
0 1 0	0	1
0 1 1	1	0
1 0 0	0	1
1 0 1	1	0
1 1 0	1	0
1 1 1	1	1



$s = c_i \text{ xor } a \text{ xor } b$

$c_o = ab + c_i(a+b)$



$s = c_i ab + \overline{c_o}(c_i + a + b)$

Digitalna elektronika

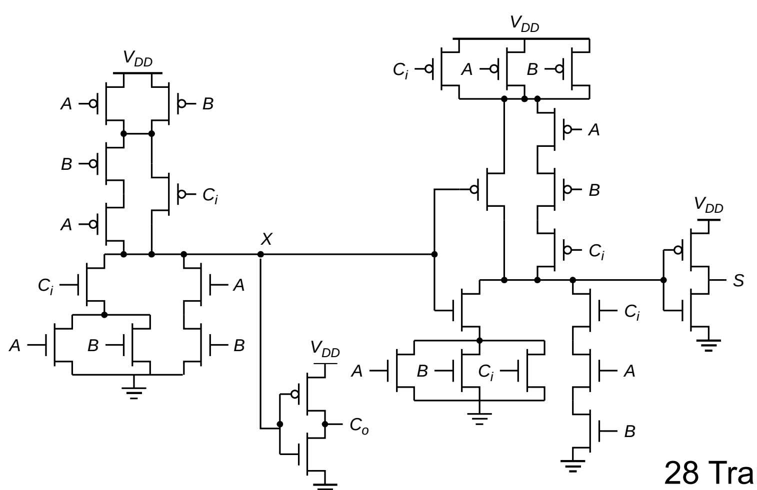
Aritmetička kola – Sabirači – Potpuni sabirač

$c_o = ab + c_i(a+b)$

$s = c_i \text{ xor } a \text{ xor } b$

$= abc_i + \overline{c_o}(c_i + a + b)$

Da bi radili samo sa pravim, a ne i komplementnim, vrednostima

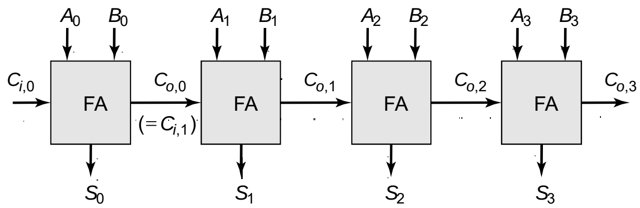


28 Tranzistora

Digitalna elektronika

Aritmetička kola – Sabirači – Ripple Carry

Ideja iz “algoritamskog” sabiranja



Worst case kašnjenje

Kašnjenje za najgori slučaj

$t_{adder} = (N\text{-}1)t_{carry} + t_{sum}$

Raste linearno sa brojem bita

$t_{adder} = O(N)$

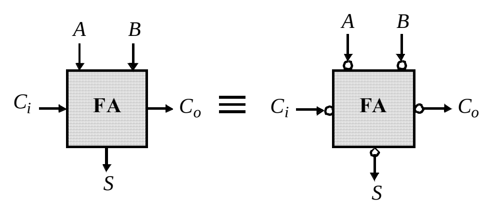
Cilj: Napraviti što brži prenos carry bita

Digitalna elektronika

Aritmetička kola – Sabirači – Princip inverzije

Da li nam trebaju invertori na izlazu u realizaciji sa 28 tranzistora?

c_i a b	c_o	s
0 0 0	0	0
0 0 1	0	1
0 1 0	0	1
0 1 1	1	0
1 0 0	0	1
1 0 1	1	0
1 1 0	1	0
1 1 1	1	1



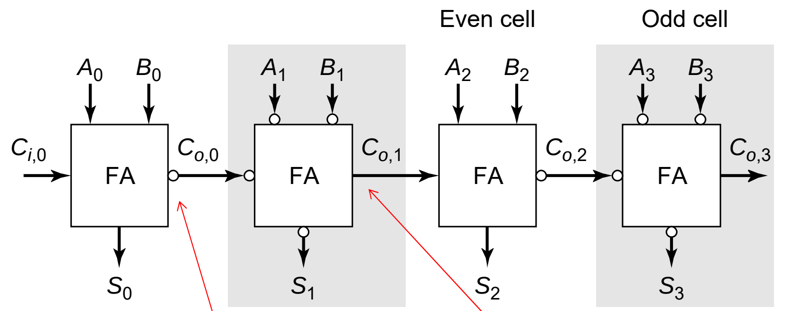
$$\bar{S}(A, B, C_i) = S(\bar{A}, \bar{B}, \bar{C}_i)$$

$$\bar{C}_o(A, B, C_i) = C_o(\bar{A}, \bar{B}, \bar{C}_i)$$

Ako na ulaze već napravljenog sabirača dovedemo invertovane vrednosti, dobićemo invertovane vrednosti izlaza.

Digitalna elektronika

Aritmetička kola – Sabirači – Princip inverzije



Nema invertora -> t_{carry} manje
Dobili invertovanu vrednost za naredni stepen

Nema invertora -> t_{carry} manje
Dobili pravu vrednost za naredni stepen

…

Potrebni invertori na ulazima A i B i izlazima S "ne utiču"na kašnjenje
Svi A i B se istovremeno dovode. Pojavljuju se na S možda samo na poslednjem sabiranju

Digitalna elektronika

Aritmetička kola – Sabirači – Potpuni sabirač

Najveći problem predstavlja "čekanje" prethodnog carry

A	B	C_i	S	C_o	*Carry status*
0	0	0	0	0	delete
0	0	1	1	0	delete
0	1	0	1	0	propagate
0	1	1	0	1	propagate
1	0	0	1	0	propagate
1	0	1	0	1	propagate
1	1	0	0	1	generate
1	1	1	1	1	generate

G - generate $G=AB$

P - propagate $P=A \text{ xor } B$

D - delete $D=\bar{A}\bar{B}$

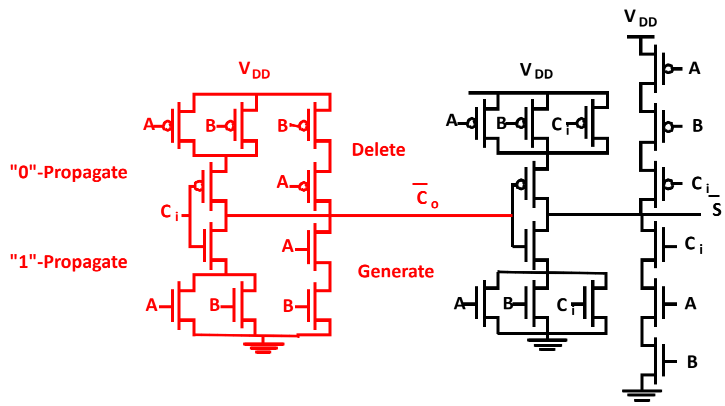
$$C_o = G+PC_i$$

$$S=P \text{ xor } C_i$$

Digitalna elektronika
Aritmetička kola – Sabirači – Mirror adder

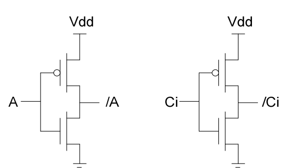
G=AB P=A xor B D=$\overline{A}\overline{B}$

$s=c_i$ xor a xor b
$=abc_i+\overline{c_o}(c_i+a+b)$

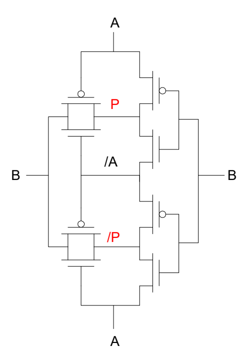


24 tranzistora
t_{carry} još manje

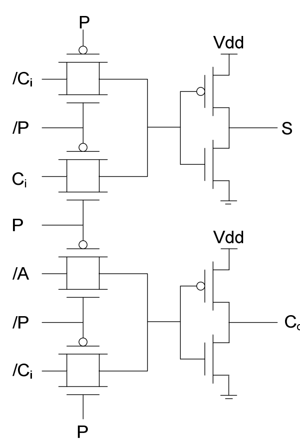
Digitalna elektronika
Aritmetička kola – Sabirači – Potpuni sabirač sa transmisionim gejtovima



Priprema



Generisanje signala P



$S=P$ xor C_i

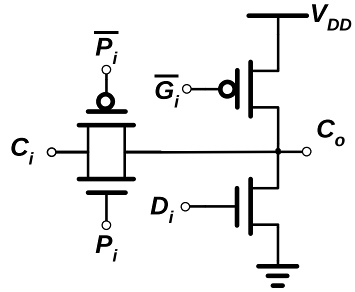
$C_o = G+PC_i$

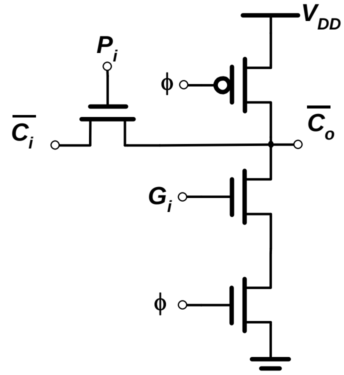
Ako jeste P ...
Ako nije P onda je G=AB.
Nije P samo ako je A=B=0 ili A=B=1 ....

Digitalna elektronika

Aritmetička kola – Sabirači

U višebitnom sabiraču svi biti sabiraka dolaze istovremeno.
Istovremeno se po bitima generišu svi potrebni P, G, D.
Znači samo carry treba da propagira.





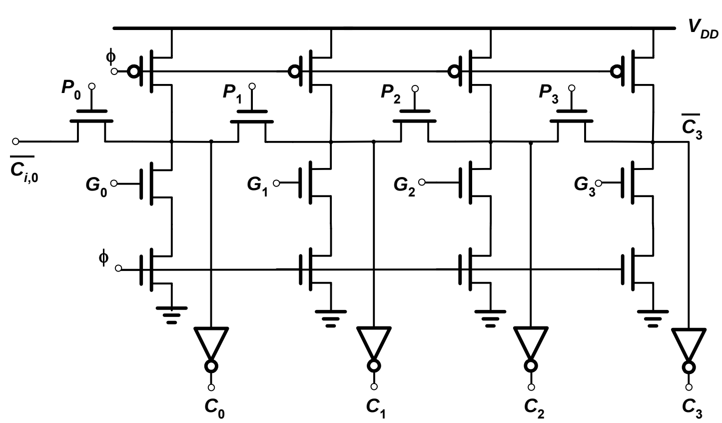
Dinamička logika

G=AB   P=A xor B $D=\bar{A}\bar{B}$

Digitalna elektronika

Aritmetička kola – Sabirači – Manchester Carry Chain

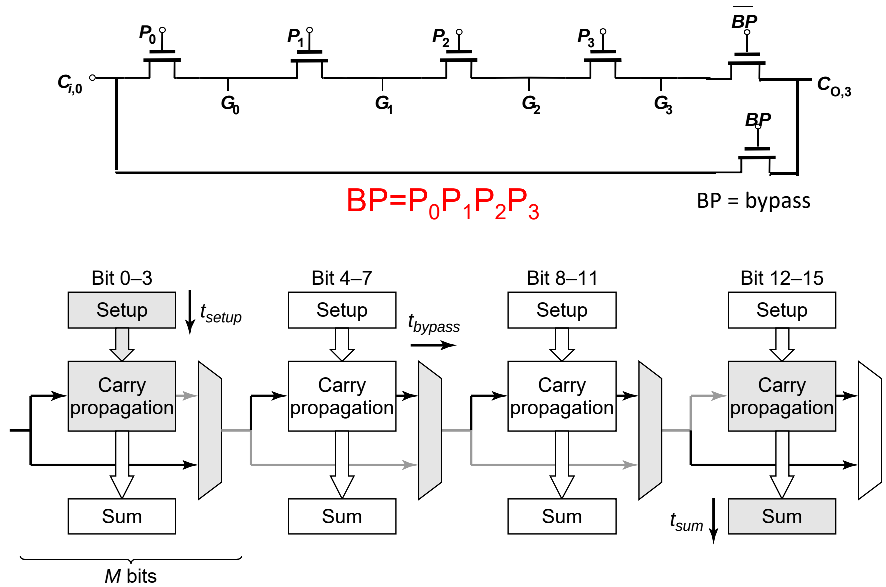
Višebitna dinamička logika



C učestvuju u sumi!
C_3 ili je "odmah" pripremljen D, G ili se u najgorem slučaju kada su svi P=1 čeka propagiranje ulaznog C_i

Digitalna elektronika

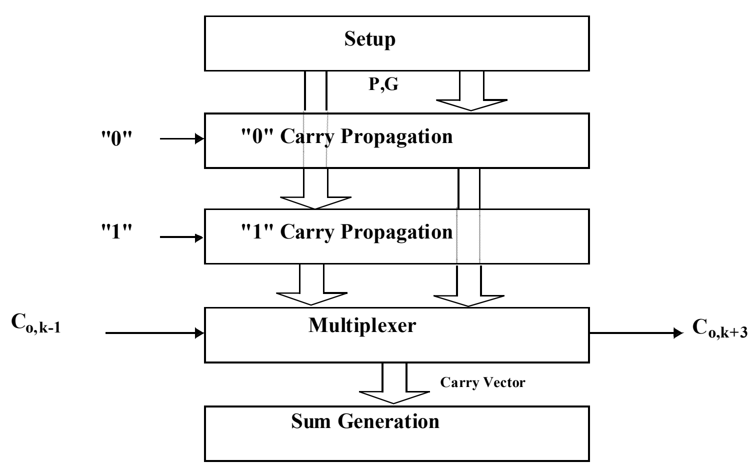
Aritmetička kola – Sabirači – Carry Bypass Adder



$$t_{adder} = t_{setup} + Mt_{carry} + (N/M\text{-}1)t_{bypass} + (M\text{-}1)t_{carry} + t_{sum}$$

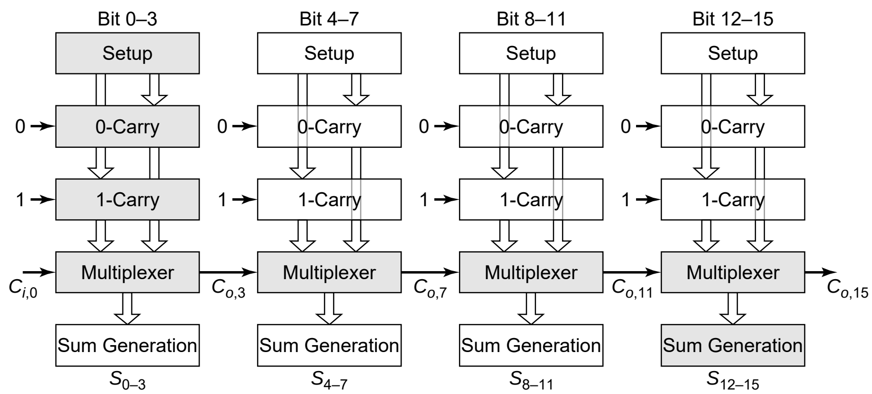
Digitalna elektronika

Aritmetička kola – Sabirači – Carry Select Adder



Digitalna elektronika

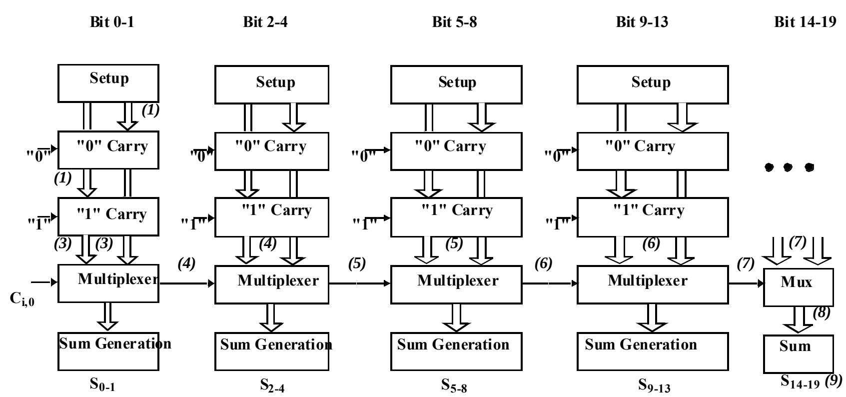
Aritmetička kola – Sabirači – Carry Select Adder



$$t_{adder} = t_{setup} + Mt_{carry} + (N/M)t_{mux} + t_{sum}$$

Digitalna elektronika

Aritmetička kola – Sabirači – Square Root Carry Select Adder



$$t_{adder} = t_{setup} + Mt_{carry} + (\sqrt{2N})\, t_{mux} + t_{sum}$$

M - broj bita prve grupe

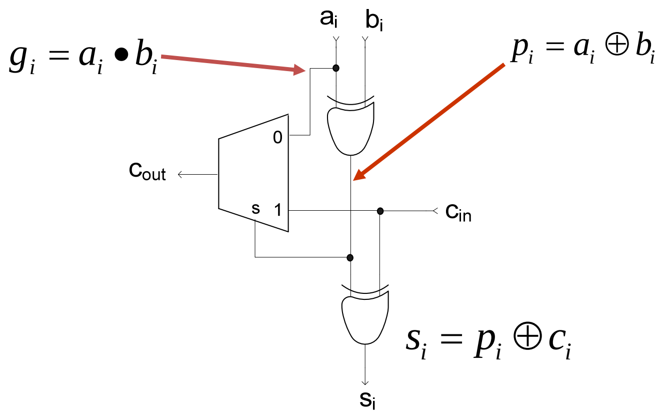
N – ukupan broj bita

Digitalna elektronika

Aritmetička kola – Sabirači – Carry Look Ahead Adder

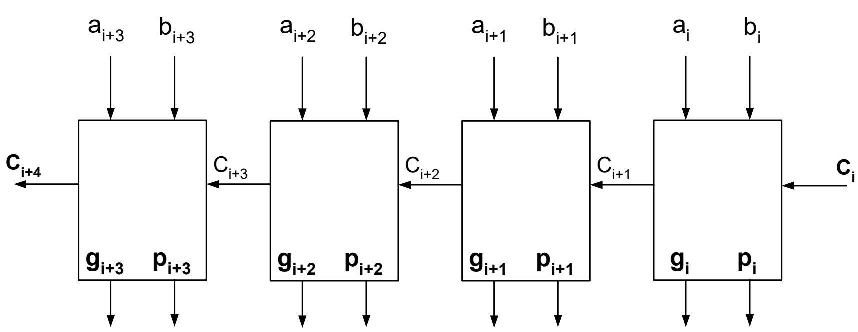
Dizajn – kompromis složenosti i brzine
Brzina po svaku cenu

$$c_{i+1} = g_i + p_i c_i$$



Digitalna elektronika

Aritmetička kola – Sabirači – Carry Look Ahead Adder



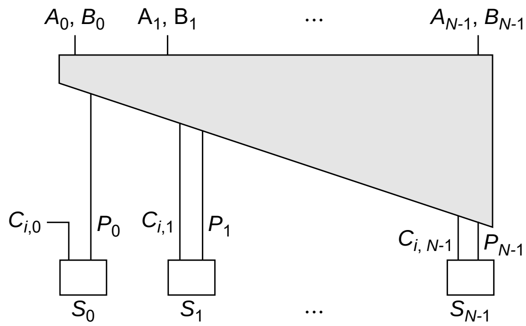
$$c_{i+1} = \bar{a}_i b_i c_i + a_i \bar{b}_i c_i + a_i b_i = g_i + p_i c_i$$

$$c_{i+2} = g_{i+1} + p_{i+1} c_{i+1} = g_{i+1} + p_{i+1}\left(g_i + p_i c_i\right) =$$
$$= g_{i+1} + p_{i+1} g_i + p_{i+1} p_i c_i$$

Digitalna elektronika

Aritmetička kola – Sabirači – Carry Look Ahead Adder

Dizajn – kompromis složenosti i brzine
Brzina po svaku cenu



$$C_{o,0} = G_0 + P_0 C_{i,0}$$

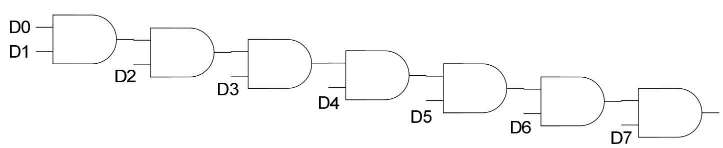
$$C_{o,1} = G_1 + P_1 G_0 + P_1 P_0 C_{i,0}$$

$$C_{o,2} = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{i,0}$$

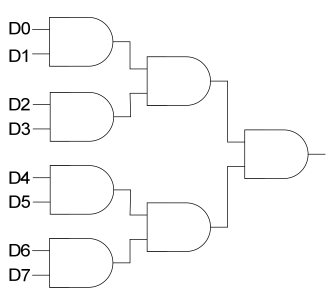
$$= (G_2 + P_2 G_1) + (P_2 P_1)(G_0 + P_0 C_{i,0}) = G_{2:1} + P_{2:1} C_{o,0}$$

Digitalna elektronika

Aritmetička kola – Sabirači – Ideja za veliki broj ulaza u logička kola



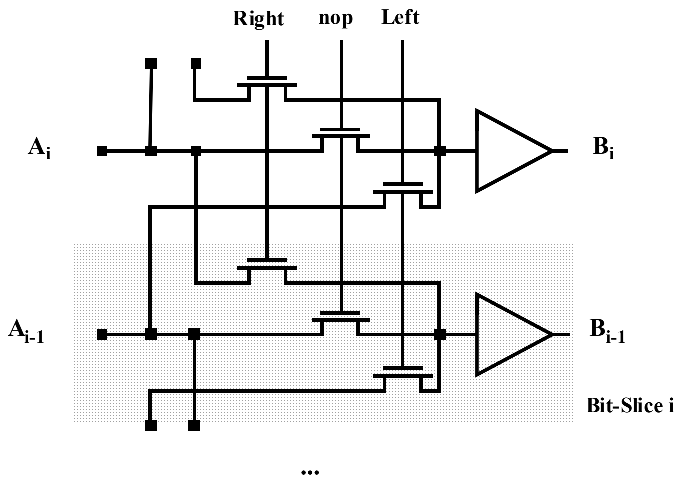
$$t_p \Leftrightarrow N$$



$$t_p \Leftrightarrow \log_2 N$$

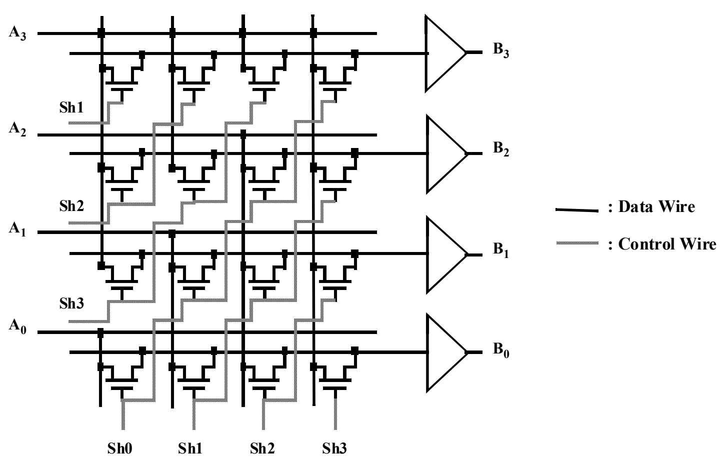
Digitalna elektronika

Aritmetička kola – Pomerač – Shifter – množenje, deljenje sa 2



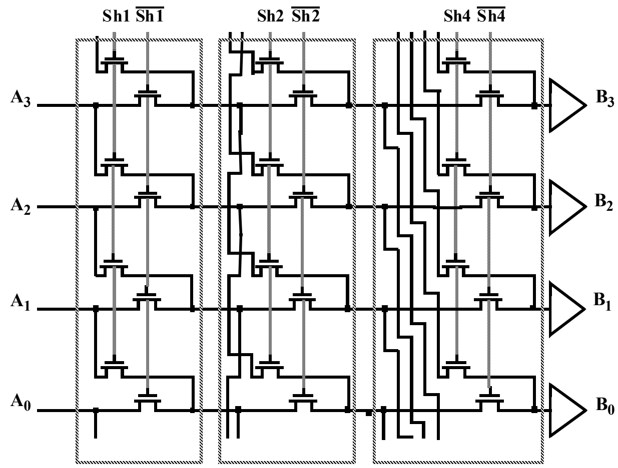
Digitalna elektronika

Aritmetička kola – Pomerač – Barel Shifter – programabilno množenje, deljenje sa 2^n



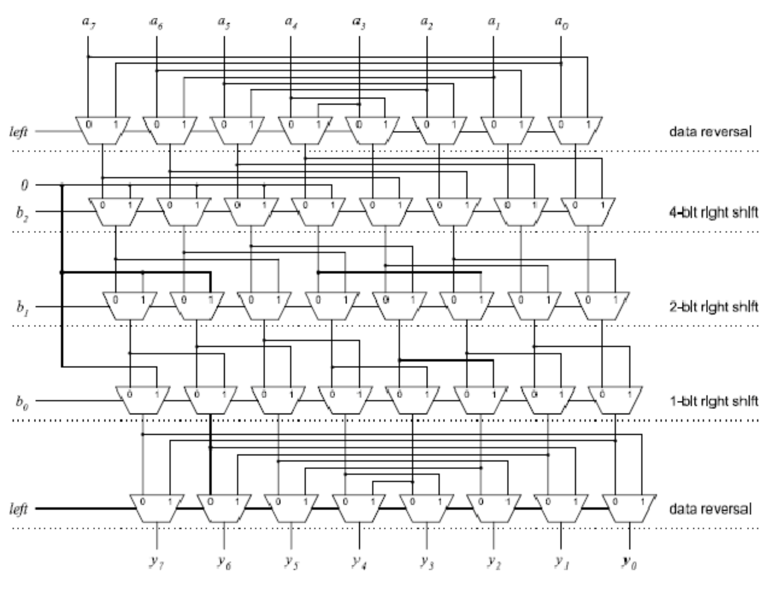
Digitalna elektronika

Aritmetička kola – Pomerač – Logaritmic Shifter



Digitalna elektronika

Aritmetička kola – Pomerač – primer - logičko pomeranje levo ili desno



Digitalna elektronika

Aritmetička kola – Binarno množenje

$$X = \sum_{i=0}^{M-1} X_i 2^i \qquad Y = \sum_{j=0}^{N-1} Y_j 2^j$$

$$XY = \left(\sum_{i=0}^{M-1} X_i 2^i\right)\left(\sum_{j=0}^{N-1} Y_j 2^j\right) = \sum_{j=0}^{N-1} Y_j \left(\sum_{i=0}^{M-1} X_i 2^{i+j}\right)$$

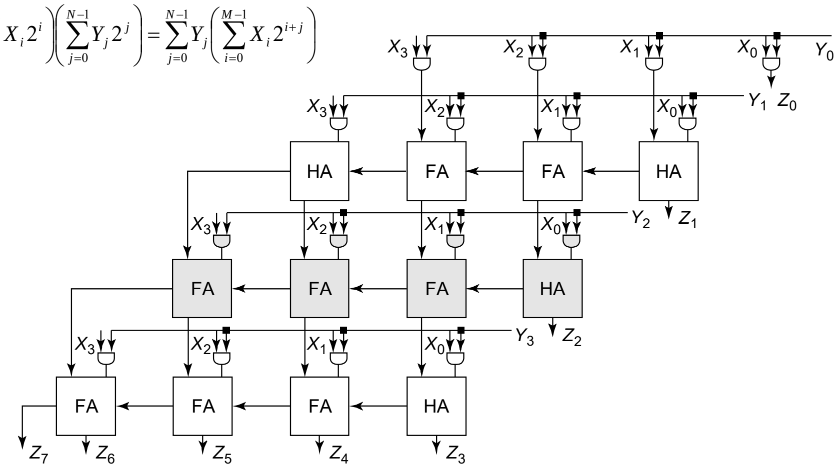
```
          1 0 1 0 1 0   x 1 0 1 1     Operandi
    -------------------
          1 0 1 0 1 0  }
        1 0 1 0 1 0    }
      0 0 0 0 0 0      }  Parcijalni proizvodi
  + 1 0 1 0 1 0        }
    -------------------
    1 1 1 0 0 1 1 1 0     Rezultat
```

Digitalna elektronika

Aritmetička kola – Množač - Array Multiplier

$$X = \sum_{i=0}^{M-1} X_i 2^i \qquad Y = \sum_{j=0}^{N-1} Y_j 2^j$$

$$XY = \left(\sum_{i=0}^{M-1} X_i 2^i\right)\left(\sum_{j=0}^{N-1} Y_j 2^j\right) = \sum_{j=0}^{N-1} Y_j \left(\sum_{i=0}^{M-1} X_i 2^{i+j}\right)$$

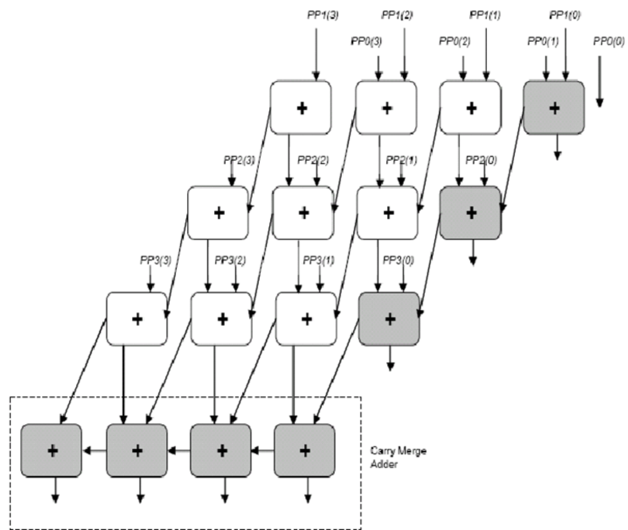


$$t_{mult} = \left(\left(M-1\right)+\left(N-2\right)\right)t_{carry} + \left(N-1\right)t_{sum} + t_{and}$$

Digitalna elektronika
Aritmetička kola – Množač – Carry Save Multiplier

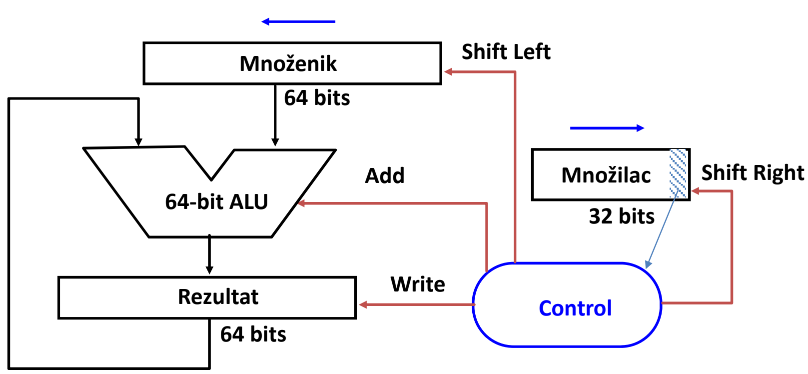
```
 0 0 1
 1 0 1
 -----
 1 1 0

 0 0 1
 1 0 1
 -----
 1 0 0
 0 1 0
 -----
 1 1 0
```

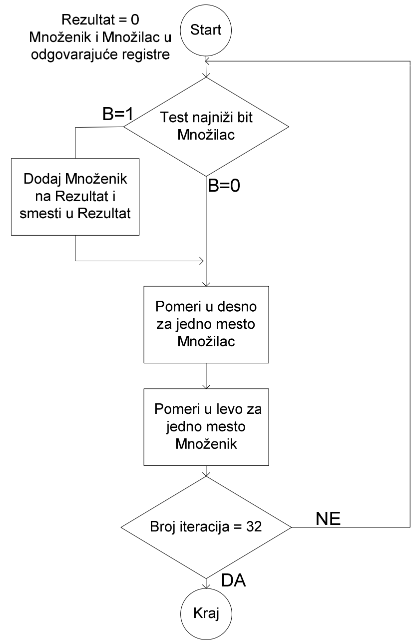


$$t_{mult} = (N-1)t_{carry} + t_{merge} + t_{and}$$

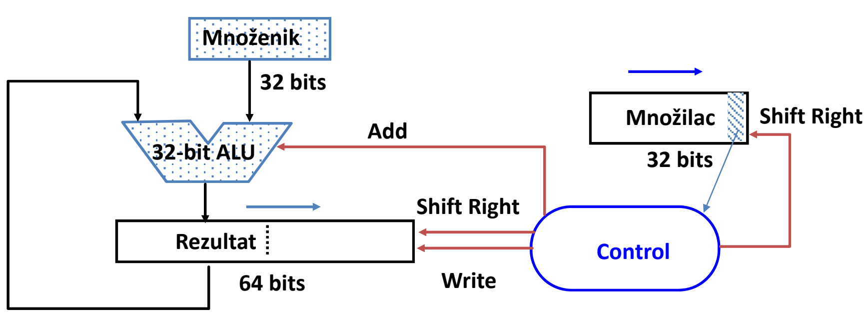
Digitalna elektronika
Aritmetička kola – Množač – Akumulaciono množenje



Digitalna elektronika
Aritmetička kola – Množač – Akumulaciono množenje

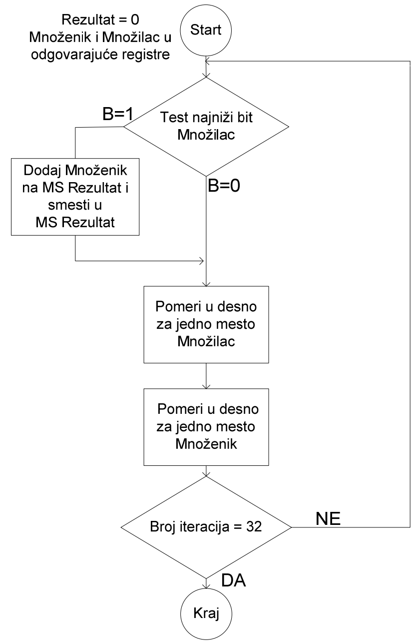


Digitalna elektronika
Aritmetička kola – Množač – Akumulaciono množenje



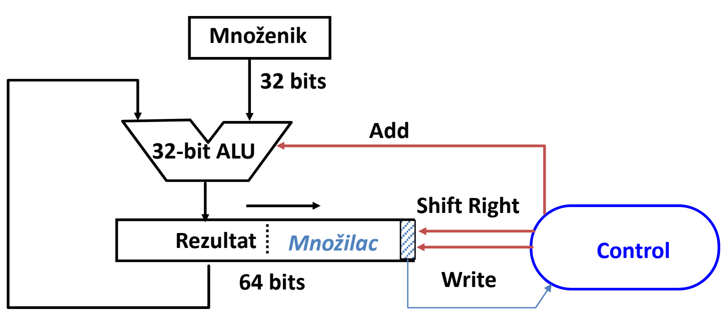
Digitalna elektronika

Aritmetička kola – Množač – Akumulaciono množenje

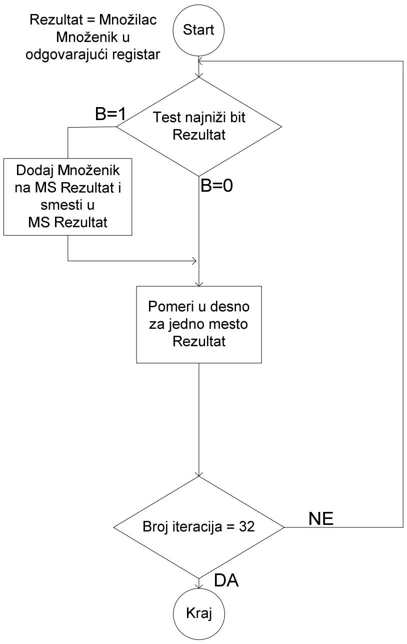


Digitalna elektronika

Aritmetička kola – Množač – Akumulaciono množenje



Digitalna elektronika
Aritmetička kola – Množač – Akumulaciono množenje



Digitalna elektronika
Aritmetička kola – Označeno množenje

1. način

- Zapamtiti znak
- Učiniti sve pozitivnim
- Uraditi množenje
- Definisati znak

```
if sign(a)!=sign(b) then s = true, else s=false
      a = abs(a)
      b = abs(b)
      p = a*b
      negate p if s = true
```

Digitalna elektronika
Aritmetička kola – Označeno množenje

2. način

Primeniti pravila označenih brojeva:

- Ekstenzija znaka parcijalnih proizvoda
- Oduzimanje parcijalnog proizvoda nastalog množenjem bita znaka

Digitalna elektronika
Aritmetička kola – Označeno množenje

-6*3

				1	0	1	0	x 0 0 1 1 Operandi
	1	1	1	1	0	1	0	Parcijalni proizvodi
	1	1	1	0	1	0		
	0	0	0	0	0			
	0	0	0	0				
1	1	1	0	1	1	1	0	Rezultat

Digitalna elektronika
Aritmetička kola – Označeno množenje

3*-6

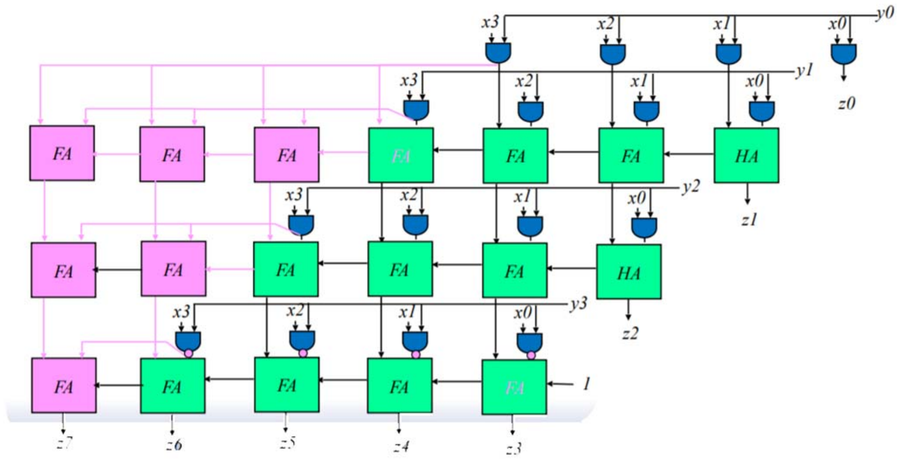
```
              0  0  1  1   x  1  0  1  0   Operandi
         ----------------------------------
              0  0  0  0  0  0  0  \
              0  0  0  0  1  1      |
              0  0  0  0  0         |  Parcijalni proizvodi
       -      0  0  1  1            /
         ----------------------------------
     1  1  1  0  1  1  1  0                Rezultat
```

Digitalna elektronika
Aritmetička kola – Označeno množenje

-3*-6

```
              1  1  0  1   x  1  0  1  0   Operandi
         ----------------------------------
              0  0  0  0  0  0  0  \
              1  1  1  1  0  1      |
              0  0  0  0  0         |  Parcijalni proizvodi
       -      1  1  0  1            /
         ----------------------------------
     0  0  0  1  0  0  1  0                Rezultat
```

Digitalna elektronika
Aritmetička kola – Označeno množenje



Digitalna elektronika
Aritmetička kola – Označeno množenje

Korak 1 – formiranje parcijalnih proizvoda

					X3	X2	X1	X0
*					Y3	Y2	Y1	Y0
	X3Y0	X3Y0	X3Y0	X3Y0	X3Y0	X2Y0	X1Y0	X0Y0
+	X3Y1	X3Y1	X3Y1	X3Y1	X2Y1	X1Y1	X0Y1	
+	X3Y2	X3Y2	X3Y2	X2Y2	X1Y2	X0Y2		
-	X3Y3	X3Y3	X2Y3	X1Y3	X0Y3			
	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0

Korak 2 – dodavanjem konstanti na odgovarajućim mestima eliminiše se višak kola nastalih zbog ekstenzije znaka. Potrebno je na kraju oduzeti ove konstante.

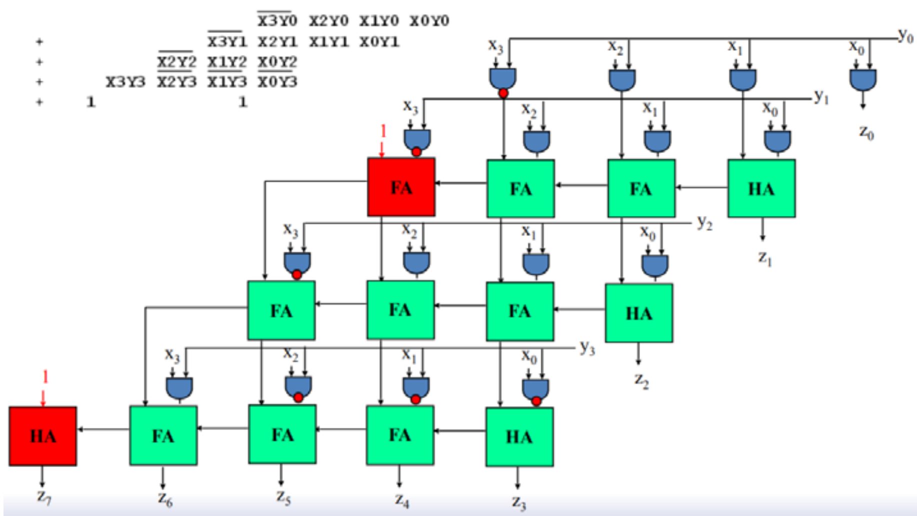
	X3Y0	X3Y0	X3Y0	X3Y0	X3Y0	X2Y0	X1Y0	X0Y0
+					1			
+	X3Y1	X3Y1	X3Y1	X3Y1	X2Y1	X1Y1	X0Y1	
+				1				
+	X3Y2	X3Y2	X3Y2	X2Y2	X1Y2	X0Y2		
+			1					
+	$\overline{X3Y3}$	$\overline{X3Y3}$	$\overline{X2Y3}$	$\overline{X1Y3}$	$\overline{X0Y3}$			
+					1			
+		1						
-		1	1	1	1			

} -B = ~B + 1

					$\overline{X3Y0}$	X2Y0	X1Y0	X0Y0
+				$\overline{X3Y1}$	X2Y1	X1Y1	X0Y1	
+			$\overline{X2Y2}$	X1Y2	X0Y2			
+		X3Y3	$\overline{X2Y3}$	$\overline{X1Y3}$	$\overline{X0Y3}$			
+								
+					1			
-		1	1	1	1			

					$\overline{X3Y0}$	X2Y0	X1Y0	X0Y0
+				$\overline{X3Y1}$	X2Y1	X1Y1	X0Y1	
+			$\overline{X2Y2}$	X1Y2	X0Y2			
+		X3Y3	$\overline{X2Y3}$	$\overline{X1Y3}$	$\overline{X0Y3}$			
+					1			
+	1				1			

Digitalna elektronika
Aritmetička kola – Označeno množenje



Digitalna elektronika
Aritmetička kola – Označeno množenje – Booth's Algorithm

3. način

Booth's Algorithm

Digitalna elektronika
Aritmetička kola – Označeno množenje – Booth's Algorithm

$$D = D_{n-1}D_{n-2}...D_1D_0 = D_{n-1}2^{n-1} + D^*$$

$$D_{n-1} = 0 \quad \Rightarrow \quad D_t = D^*$$

$$D_{n-1} = 1 \quad \Rightarrow \quad D_t = -(2^n - D) = -(2^n - D_{n-1}2^{n-1} - D^*) = -D_{n-1}2^{n-1} + D^*$$

Formirajmo ovako parcijalne proizvode za proizvod b*a

$a=(a_{31}a_{30}a_{29}a_{28} \dots a_3a_2a_1a_0)_2$

$a_{-1}=0$

$(a_{-1} - a_0) \times b \times 2^0$
$(a_0 - a_1) \times b \times 2^1$
$(a_1 - a_2) \times b \times 2^2$
…
$(a_{29} - a_{30}) \times b \times 2^{30}$
$(a_{30} - a_{31}) \times b \times 2^{31}$

$=b \times (- a_{31} 2^{31} + a_{30} 2^{30} + \dots + a_1 2^1 + a_0 2^0)$

Digitalna elektronika
Aritmetička kola – Označeno množenje – Booth's Algorithm

$a=a_{31}a_{30}a_{29}a_{28} \dots a_3a_2a_1a_0$ $\qquad a_{-1}=0$

$(a_{-1} - a_0) \times b \times 2^0$
$(a_0 - a_1) \times b \times 2^1$
$(a_1 - a_2) \times b \times 2^2$
…
$(a_{29} - a_{30}) \times b \times 2^{30}$
$(a_{30} - a_{31}) \times b \times 2^{31}$

$=b \times (- a_{31} 2^{31} + a_{30} 2^{30} + \dots + a_1 2^1 + a_0 2^0)$

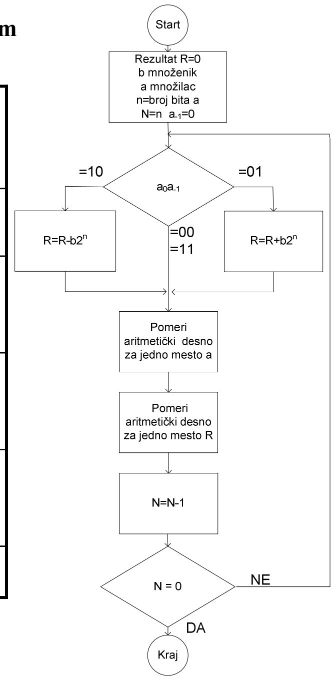
Ako posmatramo dva susedna bita

- **00: 0-0 = nop**
- **01: 1-0 = add**
- **10: 0-1 = sub**
- **11: 1-1 = nop**

Digitalna elektronika

Aritmetička kola – Označeno množenje – Booth's Algorithm

Iteracija	Operand **b**=0010 **a**=0110	Korak	Parcijalni proizvod 0000 0000
N=4	0010 01100	00: no op arith>> 1	0000 0000 0000 0000
N=3	0010 00110	10: R=R-b2ⁿ arith>> 1	1110 0000 1111 0000
N=2	0010 00011	11: no op arith>> 1	1111 0000 1111 1000
N=1	0010 00001	01: R=R+b2ⁿ arith>> 1	0001 1000 0000 1100
N=0		kraj	0000 1100



Digitalna elektronika

Aritmetička kola – Označeno množenje – Booth's Algorithm

Ukoliko se posmatraju 3 susedna bita, broj parcijalnih proizvoda se dvostruko smanjuje (spajanje susednih parcijalnih proizvoda) – Modifikovani Booth-ov algoritam

- **000: 0**
- **001: +b**
- **010: +b**
- **011: +2b**
- **100: -2b**
- **101: -b**
- **110: -b**
- **111: 0**

Sada je za formiranje parcijalnih proizvoda potrebna i operacija pomeranja (množenje sa 2)

```
              0 1  0 1 1 1          23
          ×  -1   +2  -2           ×-10  (Booth recoding 0 -1 | +1 0 | -1 0)
 1 1 1 1 1 1 0 1  0 0  1 0          -10 = 1 1 0 1 1 0 0
 0 0 0 0 1  0 1 1  1 0
 1 1 1  0 1  0 0 1
 1 1 1 1 0 0 0 1 1 0 1 0           = -230
```

Digitalna elektronika
Aritmetička kola – Deljenje

74:8

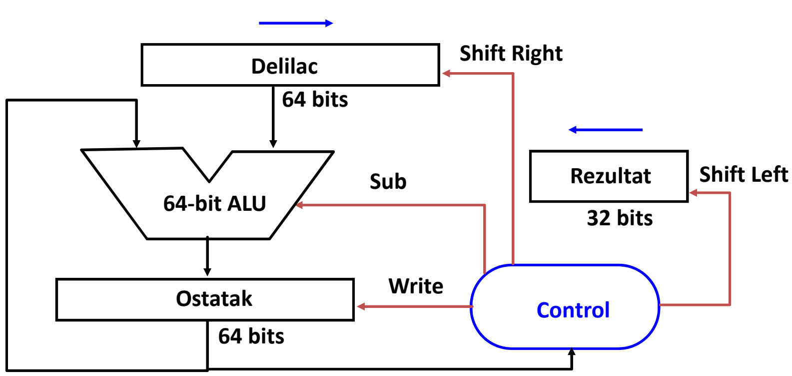
	1	0	0	1	0	1	0	:	1	0	0	0	=	1	0	0	1
-	1	0	0	0	0	0	0	1									
	0	0	0	1	0	0	0	R									
-	0	1	0	0	0	0	0	0									
	0	0	0	1	0	0	0	R									
-	0	0	1	0	0	0	0	0									
	0	0	0	1	0	1	0	R									
-	0	0	0	1	0	0	0	1									
	0	0	0	0	0	1	0	R									

Rezultat je 1 ako Delilac <= Deljenik, inače 0
- Kako ALU zna da li je ovo tačno?
- Oduzmi i ako je “rezultat” manji od nule rezultat je nula
- Pomeri i probaj ponovo

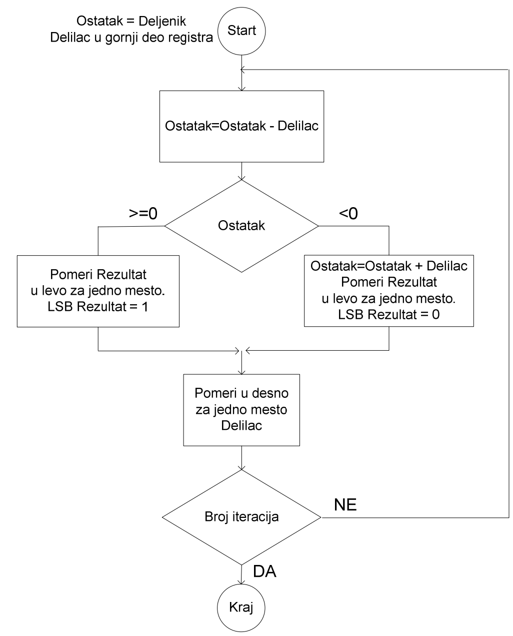
Digitalna elektronika
Aritmetička kola – Deljenje

Deljenik = Rezultat x Delilac + Ostatak

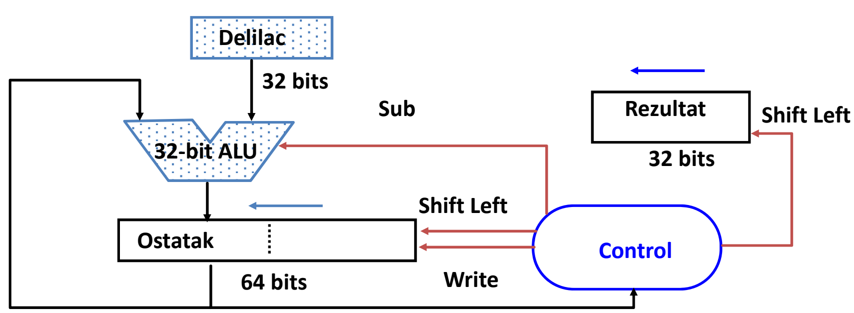
64 bita sa 32 bita



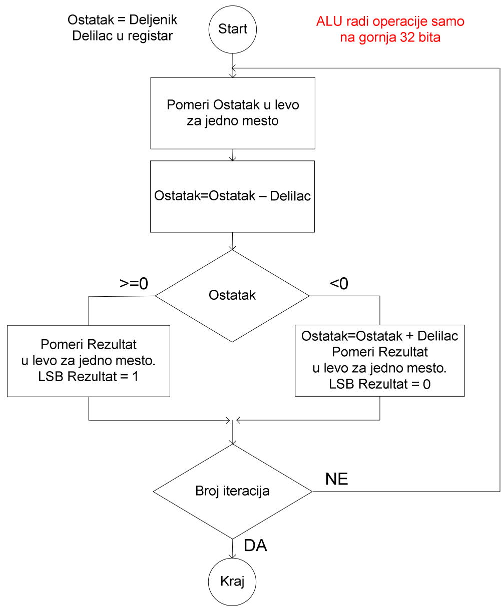
Digitalna elektronika
Aritmetička kola – Deljenje



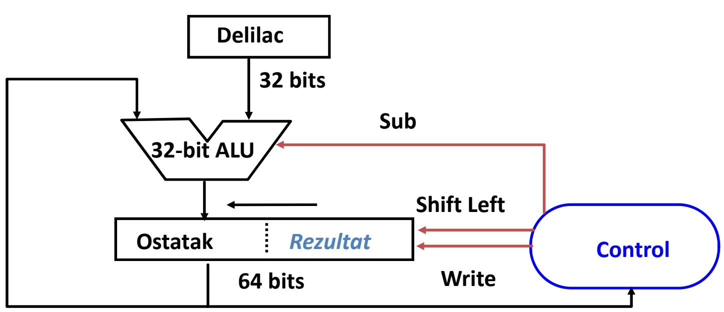
Digitalna elektronika
Aritmetička kola – Deljenje

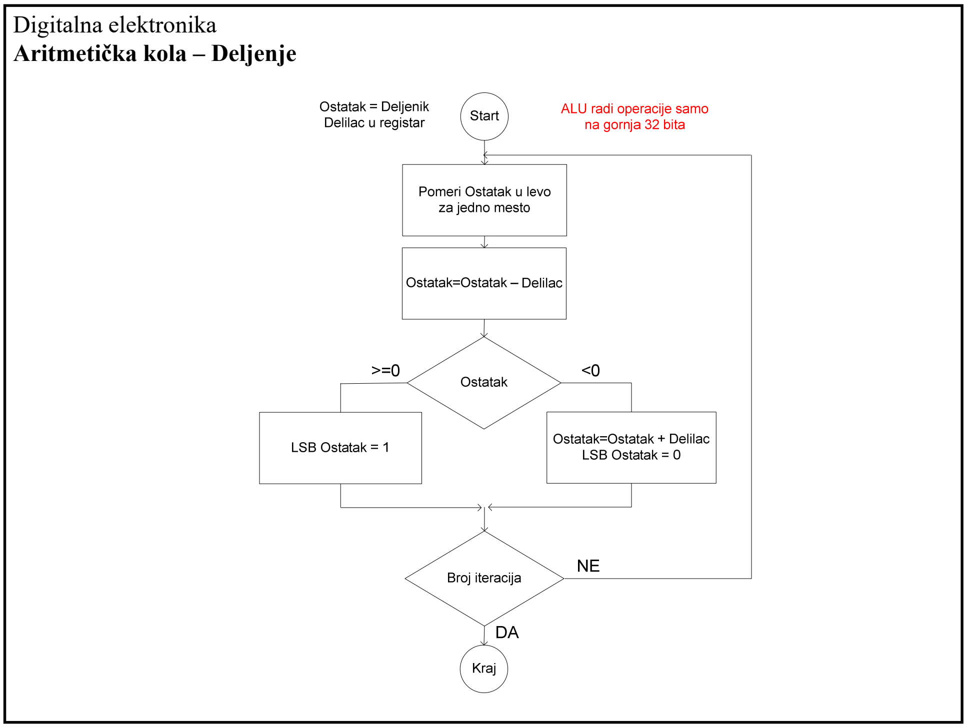


Digitalna elektronika
Aritmetička kola – Deljenje



Digitalna elektronika
Aritmetička kola – Deljenje




Digitalna elektronika
Aritmetička kola – Deljenje
Ostatak = Deljenik
Delilac u registar
Start
ALU radi operacije samo
na gornja 32 bita
Pomeri Ostatak u levo
za jedno mesto
Ostatak=Ostatak – Delilac
>=0
Ostatak
<0
LSB Ostatak = 1
Ostatak=Ostatak + Delilac
LSB Ostatak = 0
Broj iteracija
NE
DA
Kraj