



Univerzitet u Novom Sadu

Fakultet tehničkih nauka

Odsek za računarsku tehniku i
računarske komunikacije



Aritmetičke kombinacione mreže

Podela po vrstama i karakteristični načini implementacije

**There are 10 kinds of people:
those who understand
binary arithmetic and
those who don't**



Vrste aritmetičkih kombinacionih mreža



- ❖ Pomerači
 - ❖ Komparatori,
 - ❖ Sabirači,
 - ❖ Množači,
 - ❖ Delitelji,
 - ❖ Mreže za dobijanje I i II komplementa
- ❖ Moguća je i realizacija aritmetičkih kombinacionih mreža koje izvršavaju više od jedne operacije. Takve mreže se nazivaju **aritmetičko logičke jedinice**.

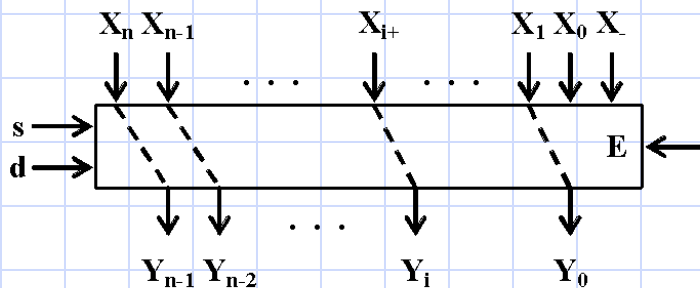
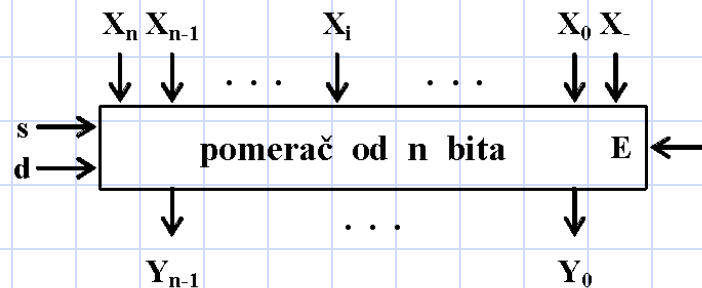
Pomerač

- ❖ Prost pomerač je kombinaciona mreža sa
 - ❖ $(n+2)$ ulaza X ,
 - ❖ n izlaza Y i
 - ❖ dva upravljačka ulaza
 - ❖ za definisanje smeru pomeranja (d) i
 - ❖ definisanje da li ima pomeranja ili nema (s).

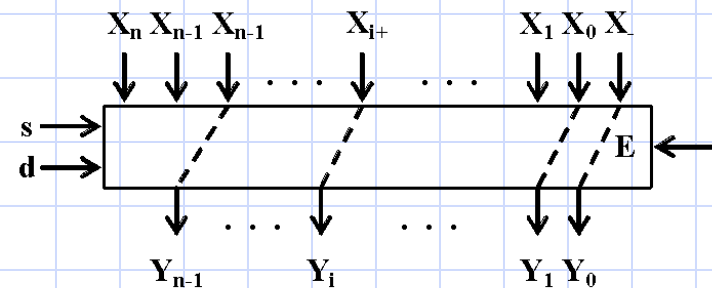
$$Y_i = \begin{cases} X_{i-1} & \text{za } d = 1 \wedge s = 1 \wedge E = 1 \quad (\text{levi pomeraj}) \\ X_{i+1} & \text{za } d = 0 \wedge s = 1 \wedge E = 1 \quad (\text{desni pomeraj}) \\ X_j & \text{za } s = 0 \wedge E = 1 \quad (\text{nema pomeraja}) \\ 0 & \text{za } E = 0 \end{cases}$$

Formiranje izlaza pomerača

Izlaz iz pomerača odgovara ulazu $X = (X_n, X_{n-1}, \dots, X_0, X_{-1})$ pomerenom za jedan bit u levo ili desno, ili nepromenjenom ulazu, kako je to definisano priključkom smera d , odnosno, ulazom s .



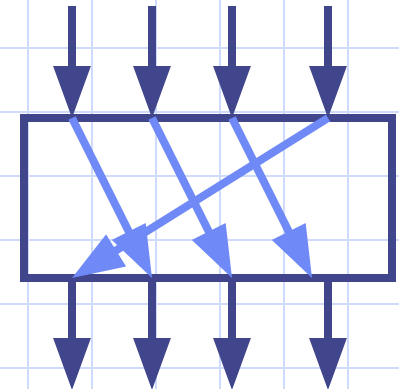
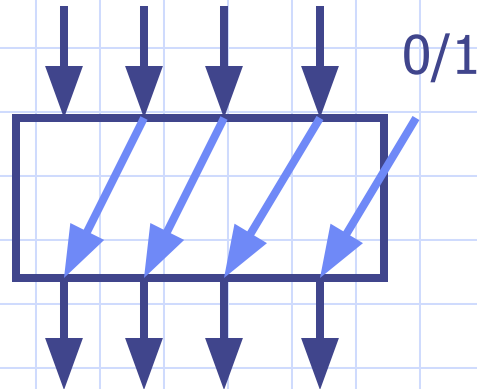
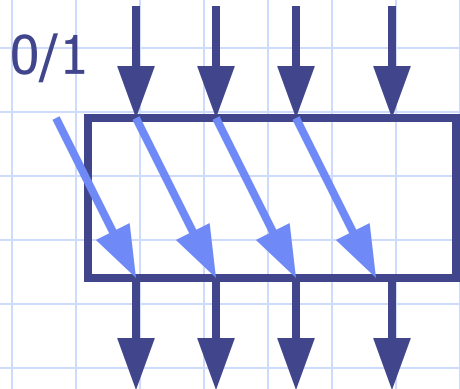
pomeranje u desno



pomeranje u levo

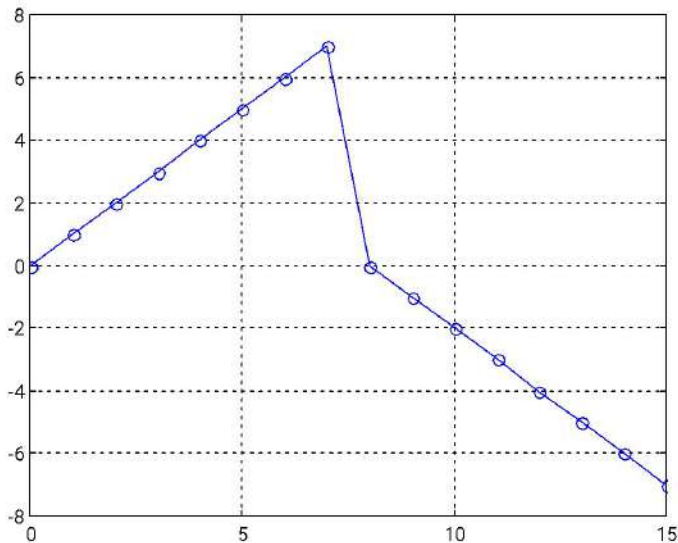
Logičko pomeranje

- ❖ Pomeranje u levo/desno sa ubacivanjem nule $X_n (X_{-1}) = 0$
- ❖ Pomeranje u levo/desno sa ubacivanjem jedinice $X_n (X_{-1}) = 1$
- ❖ Cirkularno pomeranje u levo/desno (rotiranje) $X_0 = X_{n-1} / X_{n-1} = X_0$

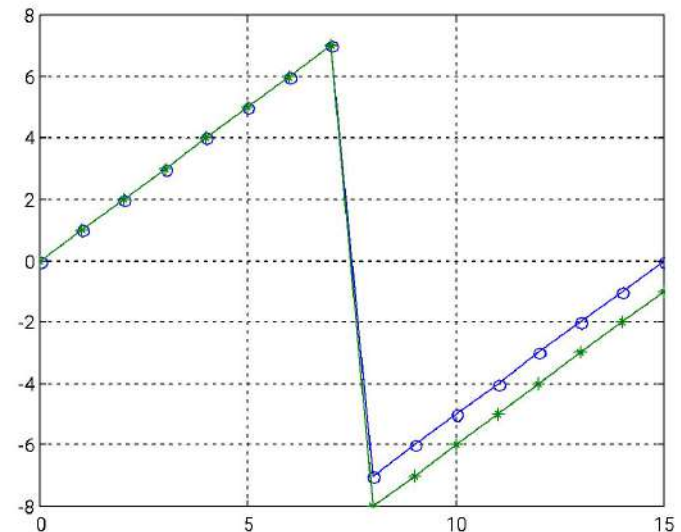


Aritmetičko pomeranje

- ❖ Aritmetičko pomeranje brojeva ulevo/udesno predstavlja množenje/deljenje tih brojeva sa 2
- ❖ Aritmetičko pomeranje (pomeranje označenih celih brojeva)
 - ❖ Predstava negativnih brojeva **znak+moduo**
 - ❖ Predstava negativnih brojeva **u prvom komplementu**
 - ❖ Predstava negativnih brojeva **u drugom komplementu**



znak+moduo



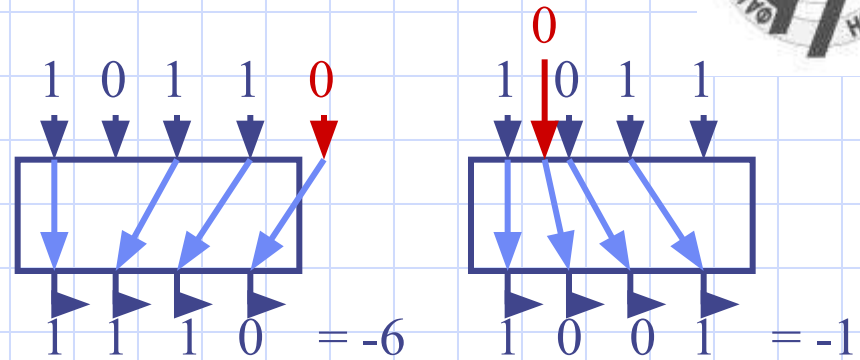
I i II komplement

Binarna predstava celih označenih brojeva

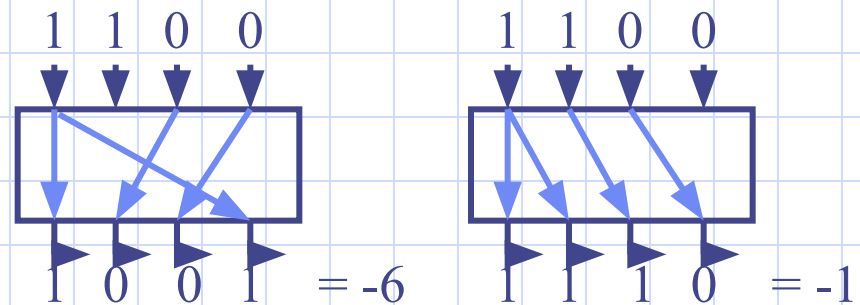
Decimalni broj	II komplement	I komplement	znak+moduo	višak 8
-8	1000	-	-	0000
-7	1001	1000	1111	0001
-6	1010	1001	1110	0010
-5	1011	1010	1101	0011
-4	1100	1011	1100	0100
-3	1101	1100	1011	0101
-2	1110	1101	1010	0110
-1	1111	1110	1001	0111
0	0000	0000 ili 1111	0000 ili 1000	1000
1	0001	0001	0001	1001
2	0010	0010	0010	1010
3	0011	0011	0011	1011
4	0100	0100	0100	1100
5	0101	0101	0101	1101
6	0110	0110	0110	1110
7	0111	0111	0111	1111

Primeri aritmetičkog pomeranja

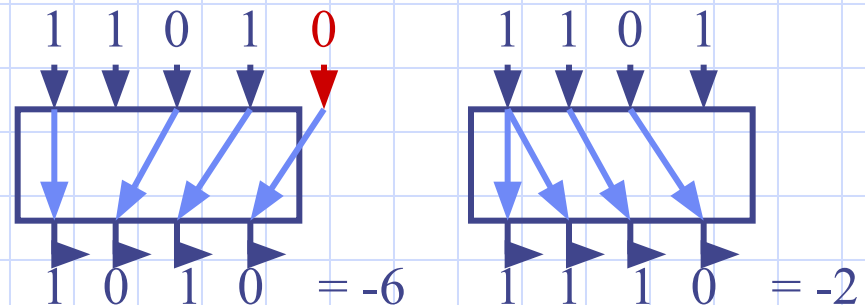
znak + moduo $-3 = 1011$



prvi komplement $-3 = 1100$

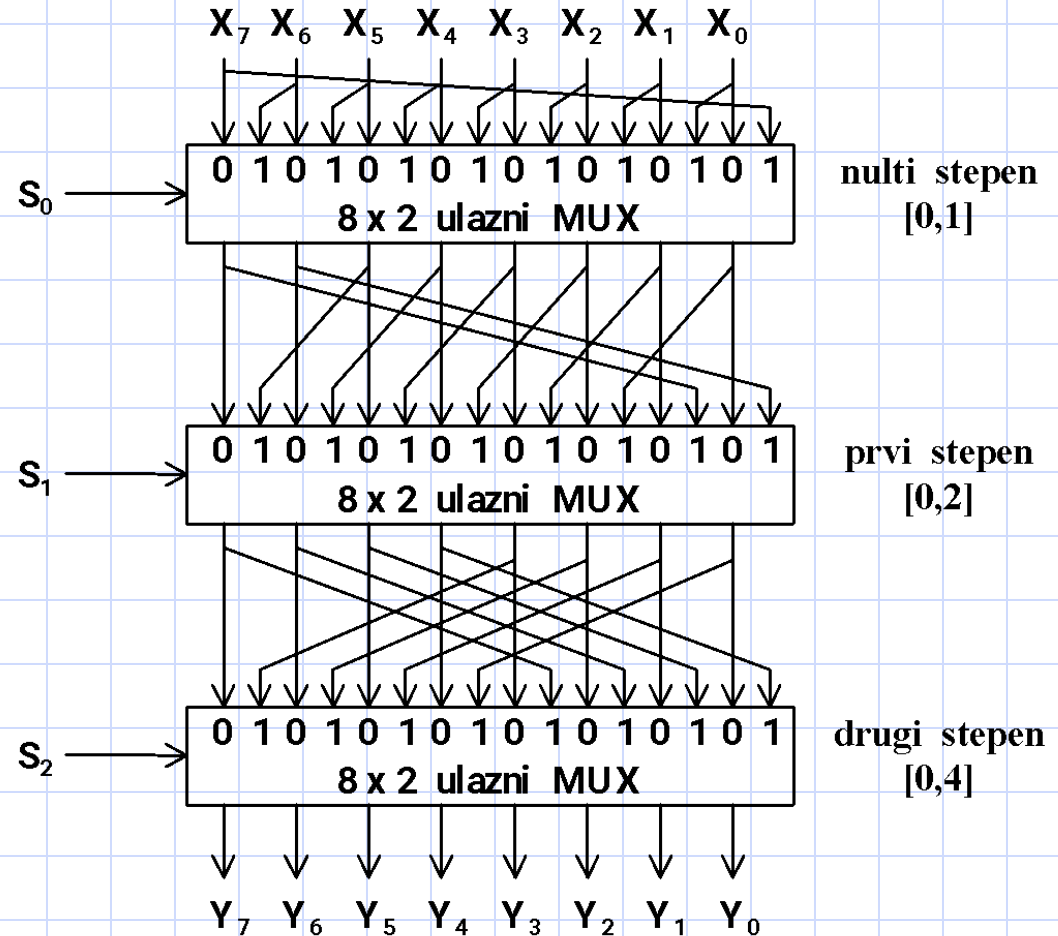


drugi komplement $-3 = 1101$



Barelov pomerač (lančani pomerač)

- ❖ Mreža od r stepeni, takva da i -ti stepen pomera na rastojanje 0 ili 2^i
- ❖ Njegov svaki stepen se realizuje kao $n \times 2$ ulazni multiplexer, gde se adresiranje ulaza i -tog multiplexera određuje sa bitom S_i .
- ❖ Kašnjenje u mreži proporcionalno je broju stepeni



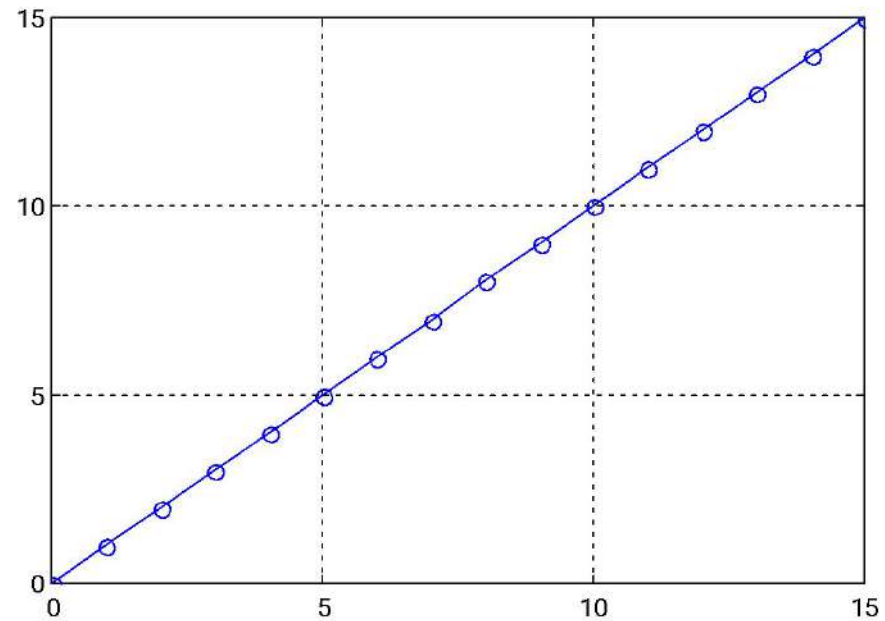
Aritmetičke kombinacione mreže

- ❖ Aritmetičke kombinacione mreže kao ulazne reči koriste cele brojeve, a njihov izlaz predstavljaju cele brojeve ili relacione operatore (jednako, manje, veće).

- ❖ Celi brojevi se predstavljaju u obliku brojeva sa brojnom osnovom dva kod kojih se broj definiše u obliku

$$X = \sum_{i=0}^{n-1} X_i 2^i$$

što omogućava rad nad celim brojevima u opsegu od 0 do $2^n - 1$
(**celi neoznačeni brojevi**)



Komparatori

- ❖ Upoređivanje brojeva se u digitalnim sistemima vrši u obliku:
 - ❖ upoređivanja po modulu,
 - ❖ upoređivanje znakova operanda

- ❖ Najpotpunija operacija upoređivanja se odnosi na ispitivanja ispunjenja jednog od uslova:
 - ❖ $X = Y$,
 - ❖ $X > Y$,
 - ❖ $X < Y$

$$X = \pm \sum_{i=0}^{n-1} X_i 2^i$$

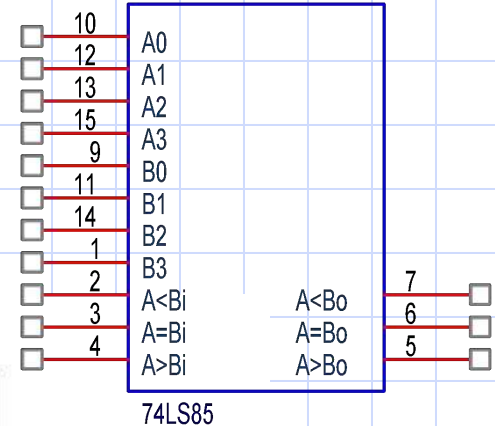
$$Y = \pm \sum_{i=0}^{n-1} Y_i 2^i$$

Primer komparatora: 74LS85

- ❖ Integrirano kolo 74LS85 izvršava funkciju komparatora
- ❖ TTL tehnologija izrade

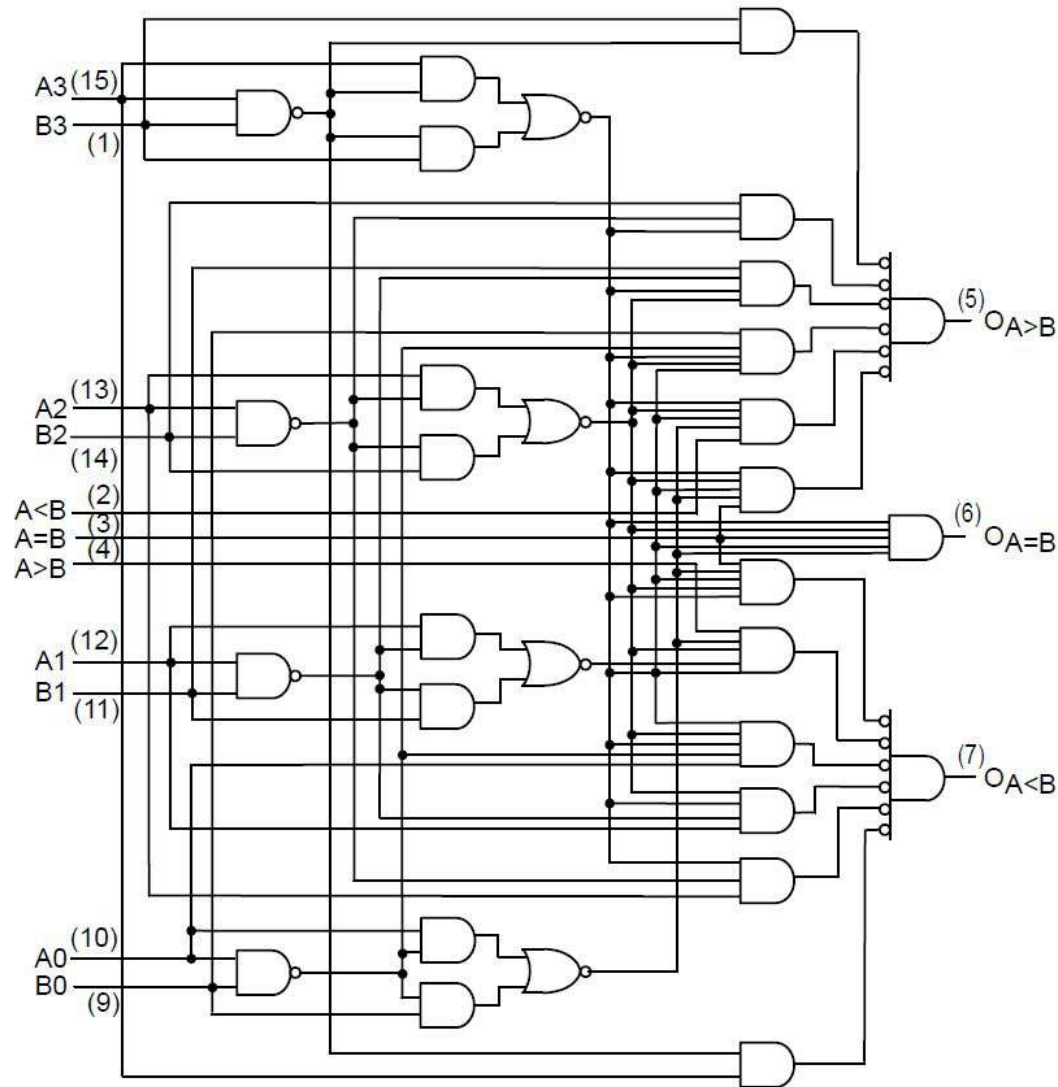
TRUTH TABLE

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A ₃ ,B ₃	A ₂ ,B ₂	A ₁ ,B ₁	A ₀ ,B ₀	I _{A>B}	I _{A<B}	I _{A=B}	O _{A>B}	O _{A<B}	O _{A=B}
A ₃ >B ₃	X	X	X	X	X	X	H	L	L
A ₃ <B ₃	X	X	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ >B ₂	X	X	X	X	X	H	L	L
A ₃ =B ₃	A ₂ <B ₂	X	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ >B ₁	X	X	X	X	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ <B ₁	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ >B ₀	X	X	X	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ <B ₀	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	H	L	L	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	H	L	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	X	X	H	L	L	H
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	H	H	L	L	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	L	L	H	H	L



H = HIGH Level
L = LOW Level
X = IMMATERIAL

Realizacija sa logičkim kapijama



VHDL realizacija 1

```
PROCESS (iA, iB, iH1, iH2, iH3) BEGIN
    IF      (iA(3) > iB(3)) THEN oH3 <= '1'; oH2 <= '0'; oH1 <= '0';
    ELSIF   (iA(3) < iB(3)) THEN oH3 <= '0'; oH2 <= '1'; oH1 <= '0';
    ELSIF   (iA(2) > iB(2)) THEN oH3 <= '1'; oH2 <= '0'; oH1 <= '0';
    ELSIF   (iA(2) < iB(2)) THEN oH3 <= '0'; oH2 <= '1'; oH1 <= '0';
    ELSIF   (iA(1) > iB(1)) THEN oH3 <= '1'; oH2 <= '0'; oH1 <= '0';
    ELSIF   (iA(1) < iB(1)) THEN oH3 <= '0'; oH2 <= '1'; oH1 <= '0';
    ELSIF   (iA(0) > iB(0)) THEN oH3 <= '1'; oH2 <= '0'; oH1 <= '0';
    ELSIF   (iA(0) < iB(0)) THEN oH3 <= '0'; oH2 <= '1'; oH1 <= '0';
    ELSIF   (iH3 = '1')      THEN oH3 <= '1'; oH2 <= '0'; oH1 <= '0';
    ELSIF   (iH2 = '1')      THEN oH3 <= '0'; oH2 <= '1'; oH1 <= '0';
    ELSE
        oH3 <= '0'; oH2 <= '0'; oH1 <= '1';
    END IF;
END PROCESS;
```

$H1 \Leftrightarrow A=B$

$H2 \Leftrightarrow A<B$

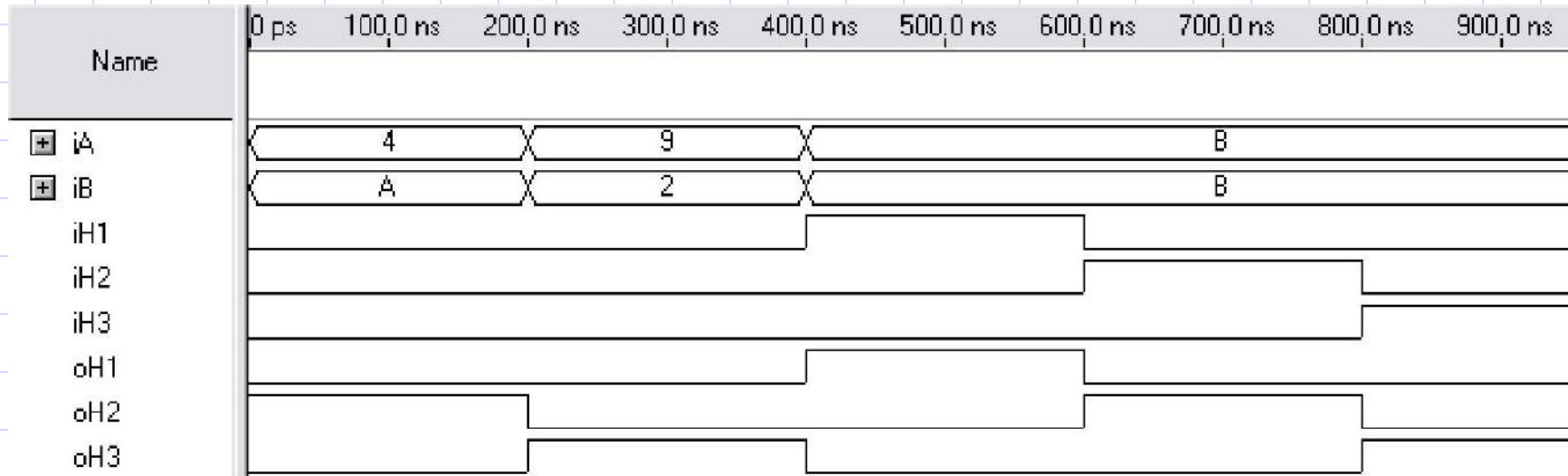
$H3 \Leftrightarrow A>B$

VHDL realizacija 2

```

PROCESS (iA, iB, iH1, iH2, iH3) BEGIN
    IF      (iA > iB)      THEN oH3 <= '1'; oH2 <= '0'; oH1 <= '0';
    ELSEIF  (iA < iB)      THEN oH3 <= '0'; oH2 <= '1'; oH1 <= '0';
    ELSEIF  (iH3 = '1')    THEN oH3 <= '1'; oH2 <= '0'; oH1 <= '0';
    ELSEIF  (iH2 = '1')    THEN oH3 <= '0'; oH2 <= '1'; oH1 <= '0';
    ELSE
        oH3 <= '0'; oH2 <= '0'; oH1 <= '1';
    END IF;
END PROCESS;

```



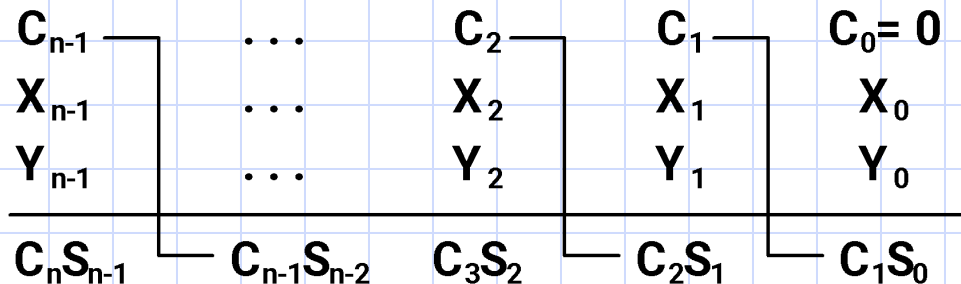
Sabirači

- ❖ Sabirač predstavlja kombinacionu mrežu koja vrši aritmetičko sabiranje brojeva

$$X = \sum_{i=1}^n x_i 2^{i-1} \quad i \quad Y = \sum_{i=1}^n y_i 2^{i-1}$$

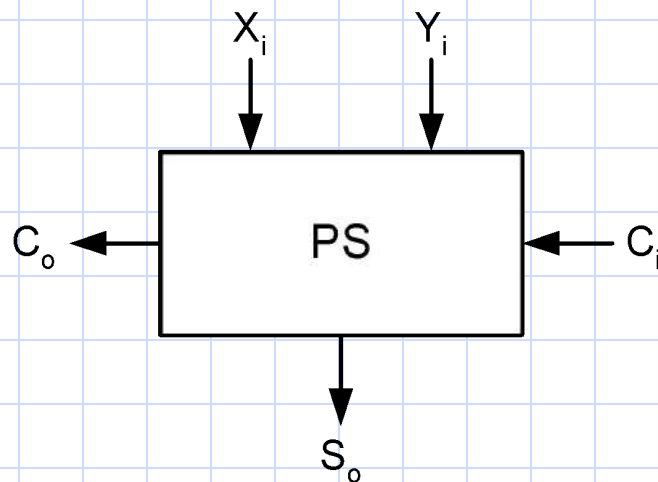
$$x_i \in \{0, 1\} \quad y_i \in \{0, 1\}$$

- ❖ Proces sabiranja dva n-bitna binarna broja može se pokazati na sledeći način:



Pun sabirač

- Punim sabiračem naziva se kombinaciona mreža sa tri ulaza X_i , Y_i , C_i i dva izlaza S_i , C_{i+1} koji predstavljaju vrednost zbira S_i i izlaznog prenosa C_{i+1}



- Koristi se za realizaciju sabiranja jednog razreda binarnih brojeva X i Y

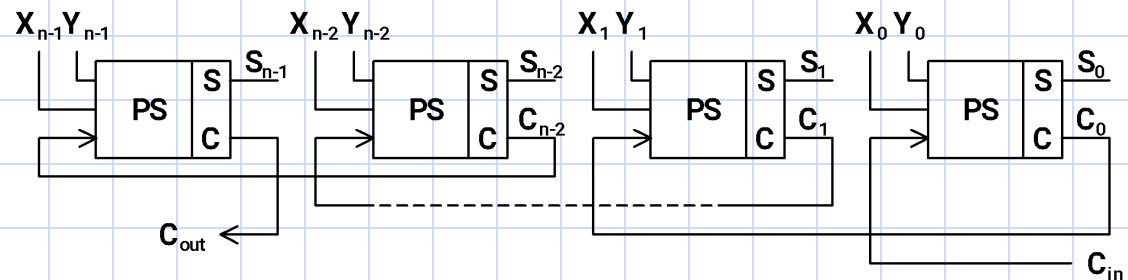
X_i	Y_i	C_i	S_o	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S_o = X_i \cdot \overline{Y_i} \cdot \overline{C_i} + \overline{X_i} \cdot \overline{Y_i} \cdot C_i + X_i \cdot Y_i \cdot C_i + \overline{X_i} \cdot Y_i \cdot \overline{C_i} = X_i \oplus Y_i \oplus C_i$$

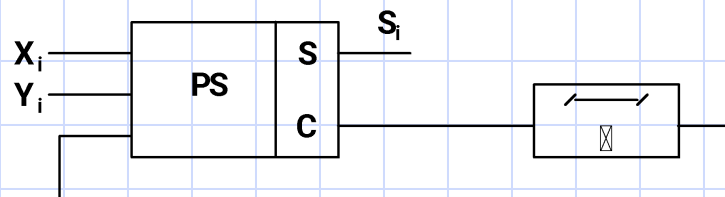
$$C_o = X_i \cdot Y_i + X_i \cdot C_i + Y_i \cdot C_i$$

Višerazredni sabirači

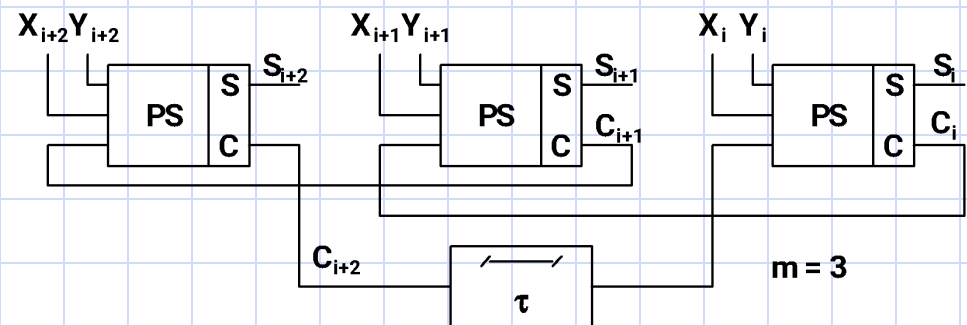
❖ Paralelni sabirač



❖ Serijski sabirač (nije kombinatorni)



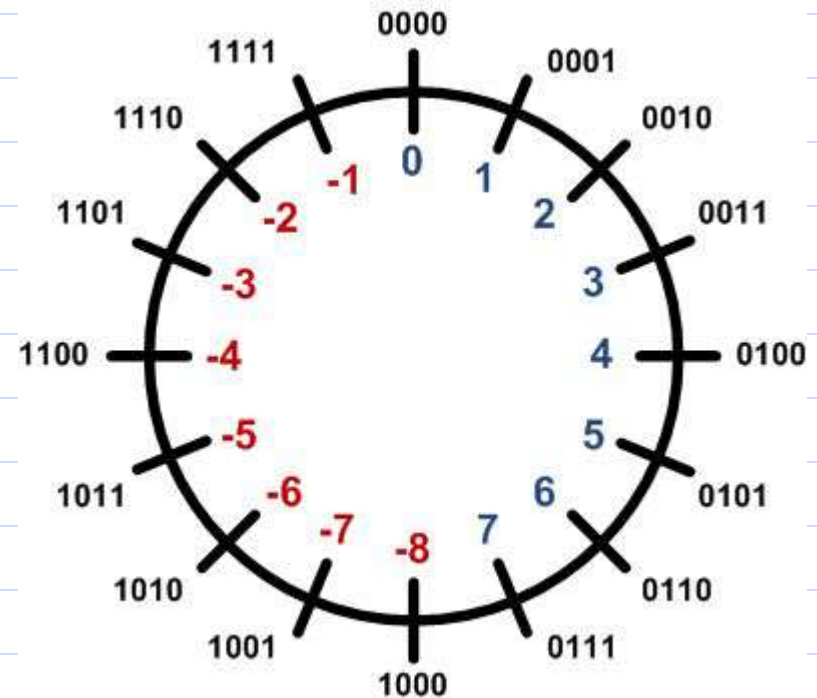
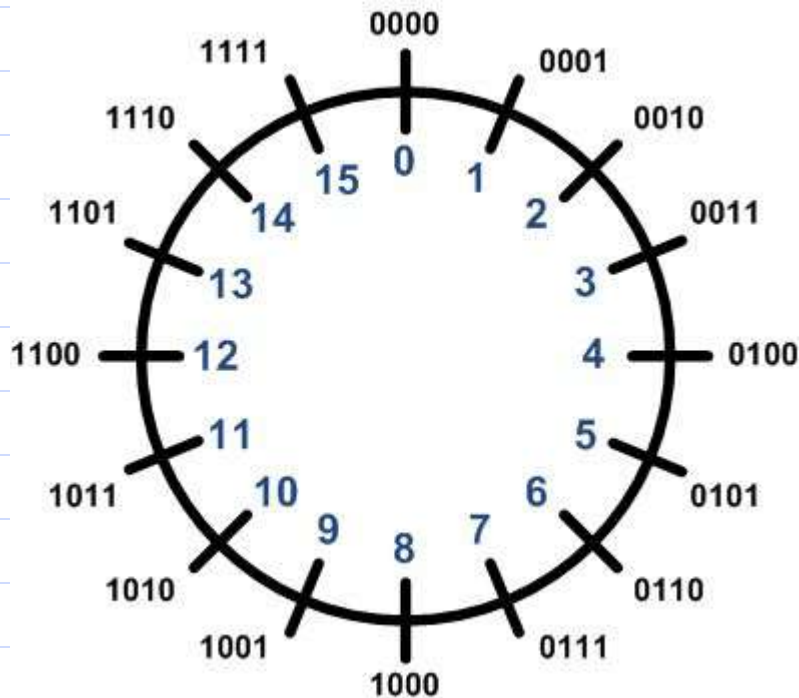
❖ Serijsko-paralelni sabirač (nije kombinatorni)



Krug brojeva

krug neoznačenih brojeva

krug označenih brojeva



Operandi i rezultat su uvek iste bitske širine

Prekoračenje (Overflow)

$$\begin{array}{r}
 -3 \\
 + \quad -6 \\
 \hline
 -9
 \end{array}
 \quad
 \begin{array}{r}
 1101 \\
 + \quad 1010 \\
 \hline
 10111 = +7
 \end{array}$$

$$\begin{array}{r}
 -8 \\
 + \quad -8 \\
 \hline
 -16
 \end{array}
 \quad
 \begin{array}{r}
 1000 \\
 + \quad 1000 \\
 \hline
 10000 = +0
 \end{array}$$

$$\begin{array}{r}
 +5 \\
 + \quad +6 \\
 \hline
 +11
 \end{array}
 \quad
 \begin{array}{r}
 0101 \\
 + \quad 0110 \\
 \hline
 1011 = -5
 \end{array}$$

$$\begin{array}{r}
 +7 \\
 + \quad +7 \\
 \hline
 +14
 \end{array}
 \quad
 \begin{array}{r}
 0111 \\
 + \quad 0111 \\
 \hline
 1110 = -2
 \end{array}$$

Prekoračenje kod sabiranja se javlja ako su znakovi operandata identični ali znak rezultata je različit

Primena sabirača za realizaciju drugih aritmetičkih funkcija

