UNIVERZITET U BEOGRADU – ELEKTROTEHNIČKI FAKULTET
KATEDRA ZA ELEKTRONIKU

DIGITALNA ELEKTRONIKA 1

Materijali za računske vežbe

ARITMETIČKE OPERACIJE

Vežbe 4

Pripremio:
Haris Turkmanović (haris@etf.bg.ac.rs)

Beograd 2022

Sadržaj

1. Uvod

1.1. Sabiranje i oduzimanje neoznačenih brojeva

Sabiranje neoznačenih brojeva X = $x_n\ x_{n-1}\ x_{n-2} \ldots x_1\ x_0$ i Y = $y_n\ y_{n-1}\ y_{n-2} \ldots y_1\ y_0$ u brojnom sistemu sa osnovom *r* se realizuje sabiranjem svake od cifara brojeva X i Y u brojnom sistemu *r*, počevši od cifre najmanje težine, i dodavanjem cifara ulaznog prenosa . Dakle, cifra s_i zbira S se određuje primenom sledeće formule:

$$s_i = (x_i + y_i + c_i) \bmod \mathrm{r} \tag{1.1.1}$$

gde c_i predstavlja cifru ulaznog prenosa. Cifra izlaznog prenosa c_{i+1} uzima vrednost 1 ako je $x_i + y_i + c_i > r$ dok u suprotnom uzima vrednost 0.

$$\begin{array}{ccccccc}
c_{n+1} & c_n & c_{n-1} & c_{n-2} & \ldots & c_1 & c_0 \\
+ & x_n & x_{n-1} & x_{n-2} & \ldots & x_1 & x_0 \\
 & y_n & y_{n-1} & y_{n-2} & \ldots & y_1 & y_0 \\
\hline
s_{n+1} & s_n & s_{n-1} & s_{n-2} & \ldots & s_1 & s_0
\end{array}$$

U slučaju da brojevi X i Y nisu predstavljeni sa istim brojem cifara, pre realizacije operacije sabiranja potrebno je dodati vodeće nule ispred sabirka koji ima manje cifara kako bi se broj cifara izjednačio. Ukoliko sabirci X i Y sadrže razlomljeni deo, pre realizacije operacije sabiranja neophodno je poravnati decimalne tačke sabiraka.

Oduzimanje neoznačenih brojeva X = $x_n\ x_{n-1}\ x_{n-2} \ldots x_1\ x_0$ i Y = $y_n\ y_{n-1}\ y_{n-2} \ldots y_1\ y_0$ u brojnom sistemu sa osnovom *r* se realizuje oduzimanjem svake od cifara brojeva X i Y u brojnom sistemu *r*, počevši od cifre najmanje težine, i oduzimanjem cifara ulazne pozajmice. Dakle, cifra d_i razlike D se određuje primenom sledeće formule:

$$d_i = (x_i - y_i - b_i) \bmod \mathrm{r} \tag{1.2.1}$$

gde b_i predstavlja cifru ulazne pozajmice. Cifra izlazne pozajmice b_{i+1} uzima vrednost 1 ako je $x_i - y_i - c_i < r$ dok u suprotnom uzima vrednost 0.

$$\begin{array}{ccccccc}
b_{n+1} & b_n & b_{n-1} & b_{n-2} & \ldots & b_1 & b_0 \\
 & x_n & x_{n-1} & x_{n-2} & \ldots & x_1 & x_0 \\
- & y_n & y_{n-1} & y_{n-2} & \ldots & y_1 & y_0 \\
\hline
d_{n+1} & d_n & d_{n-1} & d_{n-2} & \ldots & d_1 & d_0
\end{array}$$

U slučaju da brojevi X i Y nisu predstavljeni sa istim brojem cifara, pre realizacije operacije sabiranja potrebno je dodati vodeće nule ispred broja koji ima manje cifara kako bi se broj cifara izjednačio. Ukoliko brojevi X i Y sadrže razlomljeni deo, pre realizacije operacije oduzimanja neophodno je poravnati decimalne tačke.

1.2. Aritmetičke operacije u predstavi znak i apsolutna vrednost

Prilikom aritmetičkih operacija sa brojevima X = $x_n\, x_{n-1}\, x_{n-2} \ldots x_1\, x_0$ i Y = $y_n\, y_{n-1}\, y_{n-2} \ldots y_1\, y_0$, koji su dati u predstavi znak i apsolutna vrednost, najpre je neophodno odrediti znak rezultata a zatim primeniti neku od odgovarajućih operacija nad apsolutnim vrednostima brojeva X i Y.

U slučaju operacije X - Y, ukoliko je |X| < |Y|, jasno je da će rezultat biti negativan, što znači da će bit najveće težine reultata operacije oduzimanja imati vrednost 1, dok će se apsolutna vrednost rezultata dobiti kao rezultat operacije |Y| - |X|.

1.3. Sabiranje i oduzimanje brojeva u komplementu osnove

Sabiranje brojeva X = $x_n\, x_{n-1}\, x_{n-2} \ldots x_1\, x_0$ i Y = $y_n\, y_{n-1}\, y_{n-2} \ldots y_1$ datih u komplementu osnove se realizuje na skoro identičan način kao i sabiranje neoznačenih brojeva pri čemu je potrebno voditi računa prilikom ekstenzije znaka sabiraka. U okviru dobijenog rezultata se zadržava zahtevani broj cifara *n* dok se ostale cifre odbacuju.

Usled ograničenosti broja cifara, za razliku od realizacije operacije sabiranja u slučaju neoznačenih brojeva, prilikom sabiranja označenih brojeva može doći do prekoračenja (*overflow* - OF). Prekoračenje nastaje ukoliko su sabirci istog a zbir suprotnog znaka. Takođe, prekoračenje (*overflow*) se može identifikovati ukoliko su ulazni i izlazni prenos na poziciji bita znaka različiti. U slučaju binarnog brojnog sistema važi: OF = 1 ako je $c_{n+1} \neq c_n$:

$$
\begin{array}{ccccccc}
c_{n+1} & c_n & c_{n-1} & c_{n-2} & \ldots & c_1 & c_0 \\
 & x_n & x_{n-1} & x_{n-2} & \ldots & x_1 & x_0 \\
- & y_n & y_{n-1} & y_{n-2} & \ldots & y_1 & y_0 \\
\hline
\cancel{s_{n+1}} & s_n & s_{n-1} & s_{n-2} & \ldots & s_1 & s_0
\end{array}
$$

odbacuje se

Oduzimanje brojeva u komplementu osnove se svodi na sabiranje u komplementu osnove gde se umesto umanjioca uzima njegova suprotna vrednost.

1.4. Sabiranje i oduzimanje brojeva u komplementu maksimalne vrednosti

Sabiranje brojeva u komplementu maksimalne vrednosti se realizuje na sličan način kao i sabiranje brojeva u komplementu osnove. Konačan rezultat se dobija dodavanjem cifre izlaznog prenosa na zbir dobijen sabiranjem u

$$
\begin{array}{ccccccc}
c_{n+1} & c_n & c_{n-1} & c_{n-2} & \ldots & c_1 & c_0 \\
 & x_n & x_{n-1} & x_{n-2} & \ldots & x_1 & x_0 \\
+ & y_n & y_{n-1} & y_{n-2} & \ldots & y_1 & y_0 \\
\hline
t_{n+1} & t_n & t_{n-1} & t_{n-2} & \ldots & t_1 & t_0 \\
 & & & & & & t_{n+1} \\
\hline
 & s_n & s_{n-1} & s_{n-2} & \ldots & s_1 & s_0 \\
\hline
\end{array}
$$

Oduzimanje označenih brojeva u komplementu maksimalne vrednosti se najčešće vrši svođenjem na sabiranje brojeva u komplementu maksimalne vrednosti tako što se umanjilac zameni njegovom suprotnom vrednošću u komplementu maksimalne vrednosti a zatim sabere sa umanjenikom.

1.5. Množenje neoznačenih brojeva

Množenje neoznačenih brojeva je moguće realizovati na isti način kao i množenje brojeva u decimalnom brojnom sistemu. Ukoliko činioci imaju decimalnu tačku, množenje se realizuje uklanjanjem decimalne tačke dok se u okviru dobijenog proizvoda decimalna tačka ubacuje na poziciju između cifara koja odgovara zbiru pozicija decimalnih tačaka činioca.

Iako je prvi način zastupljeniji u slučaju standardnog množenja u decimalnom brojnom sistemu, takav pristup dovoljno tačno ne oslikava implementaciju množenja u digitalnoj logici. Alternativa, koja je pogodnija za implementaciju u digitalnoj logici, podrazumeva postupno određivanje međurezultata sabiranja.

1.6. Množenje označenih brojeva

Množenje označenih brojeva se realizuje na sličan način kao i množenje neoznačenih brojeva dok se sabiranje međurezultata realizuje po pravilima sabiranja predstave u kojoj su dati brojevi predstavljeni. Prilikom realizacije operacije sabiranja, u okviru pojedinačnih međurezultata, treba voditi računa o ekstenziji znaka sabiraka za jedno mesto i odbacivanju eventualnog viška cifara u poslednjem sabiranju. U dobijenim međurezultatima ne treba odbacivati eventualni višak cifara jer su ta sabiranja samo fragment ukupnog sabiranja na većem broju cifara. Takođe, ukoliko je cifra najveće težine broja sa kojim množimo negativna (pripada skupu {r/2,…,r-1}), poslednji sabirak je potrebno zameniti njegovom suprotnom vrednošću u osnovi u kojoj je predstavljen.

1.7. Deljenje neoznačenih brojeva

Operacija deljenja neoznačenih binarnih brojeva se realizuje na isti način kao operacija deljenja neoznačenih decimalnih brojeva

2. Zadaci sa časova vežbi

Zadatak 2.1.

a) Izvršiti sabiranje neoznačenih brojeva brojnom u sistemu sa osnovom u kome su dati i odrediti sve cifre prenosa:

$100.11_2 + 10.01_2$, $11111_2 + 10011_2$ uz ulazni prenos 1, $294_{10} + 42_{10}$, $5325_7 + 4163_7$, $AF3.50_{16} + FB2.B9_{16}$, $26417_8 + 13140_8$

b) Izvršiti sabiranje neoznačenih brojeva brojnom u sistemu sa osnovom u kome su dati i odrediti sve cifre prenosa:

$110101_2 - 10110_2$, $100010_2 - 10010_2$ uz ulaznu prozajmicu 1, $25.2_{10} - 2.8_{10}$, $436_8 - 627_8$, $FB.2B9_{16} - AF.350_{16}$, $26417_8 - 13140_8$

Rešenje:

a) Na osnovu algoritma opisanog u 1.1 važi:

Izraz	Postupak	Rezultat
$100.11_2 + 10.01_2$:	0 0 1 1 **0** (ulazni prenos 0) 1 0 0 . 1 1 + 0 1 0 . 0 1 **1 1 1 . 0 0**	111.00_2
$11111_2 + 10011_2$	1 1 1 1 1 **1** (ulazni prenos 1) 1 1 1 1 1 + 1 0 0 1 1 **1 1 0 0 1 1**	110011_2
$294_{10} + 42_{10}$	1 0 **0** (ulazni prenos 0) 2 9 4 + 0 4 2 (dodata vodeća nula) **3 3 6**	336_{10}
$5325_7 + 4163_7$	1 0 1 1 **0** (ulazni prenos 0) 5 3 2 5 + 4 1 6 3 **1 2 5 2 1**	12521_7
$AF3.50_{16} + FB2.B9_{16}$	1 1 0 1 0 **0** (ulazni prenos 0) A F 3 . 5 0 + F B 2 . B 9 **1 A A 6 . 0 9**	$1AA6.09_{16}$
$26417_8 + 13140_8$	1 0 0 0 **0** (ulazni prenos 0) 2 6 4 1 7 + 1 3 1 4 0 **4 1 5 5 7**	41557_8

b) Na osnovu algoritma opisanog u 1.1 važi:

Izraz	Postupak	Rezultat
$110101_2 - 10110_2$	1 1 1 1 0 **0** (ulazna pozajmica 0) 1 1 0 1 0 1 - 0 1 0 1 1 0 **0 1 1 1 1 1**	11111_2
$100010_2 - 10010_2$	1 1 1 1 1 **1** (ulazni prenos 1) 1 0 0 0 1 0 - 0 1 0 0 1 0 **0 0 1 1 1 1**	001111_2
$25.2_{10} - 2.8_{10}$	0 1 **0** (ulazni prenos 0) 2 5 . **2** - 0 2 . 8 **2 2 . 4**	22.4_{10}
$436_8 - 627_8$	1 0 1 **0** (ulazni prenos 0) 4 3 6 - 6 2 7 **7 7 7 6 0 7**	777607_8
$FB.2B9_{16} – AF.350_{16}$	0 1 1 0 0 **0** (ulazni prenos 0) F B . 2 B 9 - A F . 3 5 0 **4 B . F 6 9**	$4B.F69_{16}$
$26417_8 - 13140_8$	0 0 1 0 **0** (ulazni prenos 0) 2 6 4 1 7 - 1 3 1 4 0 **1 3 2 5 7**	13257_8

Zadatak 2.2.

c) Izvršiti sledeće operacije u kodu znak i apsolutna vrednost ako je na raspolaganju dovoljan broj bita:

A + B, B - A, A – B, -B - A, C – A, C – D, A – 2B, A – B + C – D

ako je A = 010010, B = 001010, C = 101100, D = 111001.

Rešenje:

Za realizaciju operacija u ovom zadatku, koristimo algoritam opisan u 1.2. Brojevi A i B su pozitivni dok su C i D negativni. Posmatranjem apsolutnih vrednosti brojeva, zaključujemo da je |B| < |C| < |A| < |D| odnosno: 01010 < 01100 < 10010 < 11001

Izraz	Postupak		Rezultat
	Znak	Apsolutna vrednost	

A + B	+ (jer su oba broja pozitivna)	0 0 1 0 0 1 0 0 1 0 + 0 1 0 1 0 1 1 1 0 0	011100_{ZA}
B – A	- (jer je B < A)	1 0 0 0 0 1 0 0 1 0 A - 0 1 0 1 0 B 0 1 0 0 0	101000_{ZA}
A – B	+ (jer je B < A)	1 0 0 0 0 1 0 0 1 0 A - 0 1 0 1 0 B 0 1 0 0 0	001000_{ZA}
–B – A	- (-(B+A))	1 0 1 0 0 1 0 0 1 0 A + 0 1 0 1 0 B 1 1 1 0 0	111100_{ZA}
C – A	Za samostalni rad		
C – D			
A – 2B	- (Množenje sa 2 podrazumeva pomeranje bita za jedno mesto u levo. Zbog toga važi da je 2B > A, odnosno da je rezultat negativan)	0 0 1 0 0 1 0 1 0 0 2B - 1 0 0 1 0 A 0 0 0 1 0	100010_{ZA}
A – B + C – D	Za samostalni rad		

Zadatak 2.3.

a) Izvršiti sabiranje označenih brojeva datih u komplementu osnove ukoliko su na raspolaganju 4 cifre za predstavu rezultata. Operaciju sabiranja realizovati u brojnom sistemu u kome su brojevi i dati. Pored svakog rezultata naznačiti da li prilikom operacije sabiranja dolazi do prekoračenja

$010_2 + 0011_2$, $11_2 + 110_2$, $0110_2 + 1011_2$, $1100_2 + 0101_2$, $0101_2 + 0110_2$, $1101_2 + 1011_2$, $1.0_2 + 10.1_2$, $435_{10} + 834_{10}$, $A32F_{16} + 476_{16}$, $324_7 + 365_7$

b) Izvršiti oduzimanje označenih brojeva datih u komplementu osnove ukoliko su na raspolaganju 4 cifre za predstavu rezultata. Operaciju oduzimanja realizovati u brojnom sistemu u kome su brojevi i dati. Pored svakog rezultata naznačiti da li prilikom operacije oduzimanja dolazi do prekoračenja

$010_2 - 1101_2$, $11_2 - 010_2$, $0110_2 - 0101_2$, $1100_2 - 1011_2$, $01.01_2 - 10.10_2$, $1101_2 - 0101_2$, $10_2 - 011_2$, $435_{10} - 166_{10}$, $A32F_{16} - 524_{16}$, $8135_{16} - FA3B_{16}$, $364_7 - 302_7$

Rešenje:

a) Na osnovu algoritma opisanog u 1.3 važi:

Izraz	Postupak	Rezultat
$010_2 + 0011_2$	0 0 1 0 **0** 0 0 1 0 + 0 0 1 1 ~~0~~ **0 1 0 1**	0101_2 OF = 0
$11_2 + 110_2$	1 1 1 0 **0** 1 1 1 1 + 1 1 1 0 ~~1~~ **1 1 0 1**	1101_2 OF = 0
$0110_2 + 1011_2$	1 1 1 0 **0** 0 1 1 0 + 1 0 1 1 ~~1~~ **0 0 0 1**	0001_2 OF = 0
$1100_2 + 0101_2$	1 1 0 0 **0** 1 1 0 0 + 0 1 0 1 ~~1~~ **0 0 0 1**	0001_2 OF = 0
$0101_2 + 0110_2$	0 1 0 0 **0** 0 1 0 1 + 0 1 1 0 ~~0~~ **1 0 1 1**	1011_2 OF = 1
$1101_2 + 1011_2$	1 1 1 1 **0** 1 1 0 1 + 1 0 1 1 ~~1~~ **1 0 0 0**	1000_2 OF = 0
$1.0_2 + 10.1_2$	1 1 0 0 **0** 1 1 0 . 1 + 1 1 1 . 0 ~~1~~ **1 0 1 . 1**	101.0_2 OF = 0
$435_{10} + 834_{10}$	1 1 0 0 **0** 0 4 3 5 + 9 8 3 4 ~~1~~ **0 2 6 9**	0269_{10} OF = 0
$A32F_{16} + 476_{16}$	0 0 0 1 **0** A 3 2 F + 0 4 7 6 ~~0~~ **A 7 A 5**	$A7A5_{16}$ OF = 0
$3247_7 + 3657_7$	1 1 1 1 **0** 0 3 2 4 + 6 3 6 5 ~~0~~ **0 0 2 2**	0022_7 OF = 0

b) Na osnovu algoritma opisanog u 1.3 važi:

Izraz	Postupak	Rezultat

$010_2 - 1101_2$	0 0 1 0 **0** 0 0 1 0 + 0 0 1 1 -1101_2 ~~0~~ **0 1 0 1**	0101_2 OF = 0
$11_2 - 010_2$	1 1 1 0 **0** 1 1 1 1 + 1 1 1 0 -0010_2 ~~1~~ **1 1 0 1**	1101_2 OF = 0
$0110_2 - 0101_2$	1 1 1 0 **0** 0 1 1 0 + 1 0 1 1 -0101_2 ~~1~~ **0 0 0 1**	0001_2 OF = 0
$1100_2 - 1011_2$	1 1 0 0 **0** 1 1 0 0 + 0 1 0 1 -1011_2 ~~1~~ **0 0 0 1**	0001_2 OF = 0
$01.01_2 - 10.10_2$	1 1 0 0 **0** 0 1 . 0 1 + 0 1 . 1 0 -10.10_2 ~~1~~ **1 0 . 1 1**	10.11_2 OF = 1
$1101_2 - 0101_2$	1 1 1 1 **0** 1 1 0 1 + 1 0 1 1 -0101_2 ~~1~~ **1 0 0 0**	1000_2 OF = 0
$10_2 - 011_2$	1 1 0 0 **0** 1 1 1 0 + 1 1 0 1 -0011_2 ~~1~~ **1 0 1 1**	1011_2 OF = 0
$435_{10} - 166_{10}$	1 1 0 0 **0** 0 4 3 5 + 9 8 3 4 -0166_{10} ~~1~~ **0 2 6 9**	0269_{10} OF = 0
$A32F_{16} - 524_{16}$	1 0 1 1 **0** A 3 2 F + F A D C -524_{16} ~~1~~ **9 E 0 B**	$9E0B_{16}$ OF = 0
$8135_{16} - FA3B_{16}$	0 0 0 0 **0** 8 1 3 5 + 0 5 C 5 $-FA3B_{16}$ ~~0~~ **8 6 F B**	$86FB_{16}$ OF = 0
$3647_7 - 302_7$	0 0 0 0 **0** 6 3 5 4 + 6 3 6 5 -302_7 ~~1~~ **6 0 6 2**	6062_7 OF = 0

Zadatak 2.4

a) Izvršiti sabiranje označenih brojeva datih u komplementu maksimalne vrednosti ukoliko su na raspolaganju 4 cifre za predstavu rezultata. Operaciju sabiranja realizovati u brojnom sistemu u kome su brojevi i dati. Pored svakog rezultata naznačiti da li prilikom operacije sabiranja dolazi do prekoračenja

$010_2 + 0011_2$, $11_2 + 110_2$, $0110_2 + 1011_2$, $1100_2 + 0101_2$, $0101_2 + 0110_2$, $1101_2 + 1011_2$, $1.0_2 + 10.1_2$, $435_{10} + 834_{10}$, $A32F_{16} + 476_{16}$, $324_7 + 365_7$

b) Izvršiti oduzimanje označenih brojeva datih u komplementu maksimalne vrednosti ukoliko su na raspolaganju 4 cifre za predstavu rezultata. Operaciju oduzimanja realizovati u brojnom sistemu u kome su brojevi i dati. Pored svakog rezultata naznačiti da li prilikom operacije oduzimanja dolazi do prekoračenja

$010_2 - 1101_2$, $11_2 - 010_2$, $0110_2 - 0101_2$, $1100_2 - 1011_2$, $01.01_2 - 10.10_2$, $1101_2 - 0101_2$, $10_2 - 011_2$, $435_{10} - 166_{10}$, $A32F_{16} - 524_{16}$, $8135_{16} - FA3B_{16}$, $364_7 - 302_7$

Rešenje:

a) Na osnovu algoritma opisanog u 1.4 važi:

Izraz	Postupak	Rezultat
$010_2 + 0011_2$	0 0 1 0 **0** 0 0 1 0 + 0 0 1 1 0 0 1 0 1 + 0 **0 1 0 1**	0101_2 OF = 0
$11_2 + 110_2$	1 1 1 0 **0** 1 1 1 1 + 1 1 1 0 1 1 1 0 1 + 1 **1 1 1 0**	1110_2 OF = 0
$0110_2 + 1011_2$	1 1 1 0 **0** 0 1 1 0 + 1 0 1 1 1 0 0 0 1 + 1 **0 0 1 0**	0010_2 OF = 0
$1100_2 + 0101_2$	1 1 0 0 **0** 1 1 0 0 + 0 1 0 1 1 0 0 0 1 + 1 **0 0 1 0**	0010_2 OF = 0
$0101_2 + 0110_2$	0 1 0 0 **0** 0 1 0 1 + 0 1 1 0 0 1 0 1 1 + 0 **1 0 1 1**	1011_2 OF = 1

$1101_2 + 1011_2$	1 1 1 1 **0** 1 1 0 1 + 1 0 1 1 1 1 0 0 0 + 1 **1 0 0 1**	1001_2 OF = 0
$1.0_2 + 10.1_2$	1 1 0 0 **0** 1 1 0 . 1 + 1 1 1 . 0 1 1 0 1 . 1 + 1 **1 1 0 . 0**	110.0_2 OF = 0
$435_{10} + 834_{10}$	1 1 0 0 **0** 0 4 3 5 + 9 8 3 4 1 0 2 6 9 + 1 **0 2 7 0**	0270_{10} OF = 0
$A32F_{16} + 476_{16}$	0 0 0 1 **0** A 3 2 F + 0 4 7 6 0 A 7 A 5 + 0 **A 7 A 5**	$A7A5_{16}$ OF = 0
$324_7 + 365_7$	1 1 1 1 **0** 0 3 2 4 + 6 3 6 5 1 0 0 2 2 + 1 **0 0 2 3**	0023_7 OF = 0

b) Na osnovu algoritma opisanog u 1.4 važi:

Izraz	**Postupak**	**Rezultat**
$010_2 - 1101_2$	0 0 1 0 **0** 0 0 1 0 + 0 0 1 0 -1101_2 0 0 1 0 0 0 **0 1 0 0**	0100_2 OF = 0
$11_2 - 010_2$	1 1 1 1 **0** 1 1 1 1 + 1 1 0 1 -0010_2 1 1 1 0 0 1 **1 1 0 1**	1101_2 OF = 0

$0110_2 - 0101_2$	1 1 1 0 **0** 0 1 1 0 + 1 0 1 0 -0101_2 1 0 0 0 0 1 **0 1 0 1**	0001_2 OF = 0
$1100_2 - 1011_2$	1 1 0 0 **0** 1 1 0 0 + 0 1 0 0 -1011_2 1 0 0 0 1 1 **0 0 1 0**	0001_2 OF = 0
$01.01_2 - 10.10_2$	1 1 0 1 **0** 0 1 . 0 1 + 0 1 . 0 1 -10.10_2 0 1 0 . 1 0 0 **1 0 . 1 0**	10.10_2 OF = 1
$1101_2 - 0101_2$	1 0 0 0 **0** 1 1 0 1 + 1 0 1 0 -0101_2 1 0 1 1 1 1 **1 0 0 0**	1000_2 OF = 0
$10_2 - 011_2$	1 1 0 0 **0** 1 1 1 0 + 1 1 0 0 -0011_2 1 1 0 1 0 1 **1 0 1 1**	1011_2 OF = 0
$435_{10} - 166_{10}$	1 1 0 0 **0** 0 4 3 5 + 9 8 3 3 -0166_{10} 1 0 2 6 8 1 **0 2 6 9**	0269_{10} OF = 0
$A32F_{16} - 524_{16}$	1 0 1 1 **0** A 3 2 F + F A D B -524_{16} 1 9 E 0 A 1 **9 E 0 B**	$9E0B_{16}$ OF = 0
$8135_{16} - FA3B_{16}$	0 0 0 0 **0** 8 1 3 5 + 0 5 C 5 $-FA3B_{16}$ 0 8 6 F B 0 **8 6 F B**	$86FB_{16}$ OF = 0
$3647_7 - 302_7$	1 1 1 1 **0**	6062_7

<table>
<tr><td></td><td>6 3 5 4
+ 6 3 6 4 -302_7
1 6 0 6 1
1
6 0 6 2</td><td>OF = 0</td></tr>
</table>

Zadatak 2.5

a) Izvršiti množenje neoznačenih binarnih brojeva:

$$10110_2 \times 01010_2,\ 110.01_2 \times 10.111_2,\ 0110.1_2 \times 1.0110_2$$

b) Izvršiti množenje petobitnih binarnih brojeva datih u komplementu osnove:

$$10110_2 \times 01010_2,\ 110.01_2 \times 10.111_2,\ 0110.1_2 \times 1.0110_2$$

Rešenje:

a) Na osnovu postupka objašnjenog u 1.5 jasno je da postoje dva pristupa za računanje proizvoda neoznačenih brojeva datih u binarnom brojnom sistemu:

I) pristup koji isključuje međuzbirove
II) pristup koji podrazumeva generisanje međurezultata (pogodniji za implementaciju u digitalnoj logici)

<table>
<tr><th>Izraz</th><th colspan="2">Postupak</th><th>Rezultat</th></tr>
<tr><td rowspan="2">$10110_2 \times 01010_2$</td><td>I</td><td>0 0 0 0 0 ($10110_2 \times 0_2$)
1 0 1 1 0 ($10110_2 \times 1_2$)
0 0 0 0 0 ($10110_2 \times 0_2$)
1 0 1 1 0 ($10110_2 \times 1_2$)
+ 0 0 0 0 0 ($10110_2 \times 0_2$)
0 1 1 0 1 1 1 0 0</td><td rowspan="2">011011100_2</td></tr>
<tr><td>II</td><td>0 0 0 0 0 (početni zbir 0)
+ 0 0 0 0 0 ($10110_2 \times 0_2$)
0 0 0 0 0
+ 1 0 1 1 0 ($10110_2 \times 1_2$)
1 0 1 1 0 0
+ 0 0 0 0 0 ($10110_2 \times 0_2$)
0 1 0 1 1 0 0
+ 1 0 1 1 0 ($10110_2 \times 1_2$)
1 1 0 1 1 1 0 0
+ 0 0 0 0 0 ($10110_2 \times 0_2$)
0 1 1 0 1 1 1 0 0</td></tr>
<tr><td>$110.01_2 \times 10.111_2$</td><td>I</td><td>1 1 0 0 1 ($11001_2 \times 1_2$)
1 1 0 0 1 ($11001_2 \times 1_2$)
1 1 0 0 1 ($11001_2 \times 1_2$)
0 0 0 0 0 ($11001_2 \times 0_2$)
+ 1 1 0 0 1 ($11001_2 \times 1_2$)
1 0 0 0 1 1 1 1 1 1</td><td>10001.11111_2</td></tr>
</table>

	II	0 0 0 0 0 (početni zbir 0) + 1 1 0 0 1 ($11001_2 \times 1_2$) 1 1 0 0 1 + 1 1 0 0 1 ($11001_2 \times 1_2$) 1 0 0 1 0 1 1 + 1 1 0 0 1 ($11001_2 \times 1_2$) 1 0 1 0 1 1 1 1 + 0 0 0 0 0 ($11001_2 \times 0_2$) 1 0 1 0 1 1 1 1 + 1 1 0 0 1 ($11001_2 \times 1_2$) **1 0 0 0 1 1 1 1 1 1**	
$0110.1_2 \times 1.0110_2$	I	0 0 0 0 0 ($11001_2 \times 0_2$) 1 1 0 0 1 ($11001_2 \times 1_2$) 1 1 0 0 1 ($11001_2 \times 1_2$) 0 0 0 0 0 ($11001_2 \times 0_2$) + 1 1 0 0 1 ($11001_2 \times 1_2$) **1 0 0 0 1 1 1 1 1 1**	1000.11110_2
	II	0 0 0 0 0 (početni zbir 0) + 0 0 0 0 0 ($0110.1_2 \times 0_2$) 0 0 0 0 0 + 0 1 1 0 1 ($0110.1_2 \times 1_2$) 0 0 1 1 0 1 0 + 0 1 1 0 1 ($0110.1_2 \times 1_2$) 0 1 0 0 1 1 1 0 + 0 0 0 0 0 ($0110.1_2 \times 0_2$) 0 1 0 0 1 1 1 0 + 0 1 1 0 1 ($0110.1_2 \times 1_2$) **1 0 0 0 1 1 1 1 0**	

b) Na osnovu postupka objašnjenog u 1.6 važi:

Izraz	Postupak		Rezultat
$10110_2 \times 01010_2$	I	0 0 0 0 0 0 0 0 0 ($10110_2 \times 0_2$ + EZ[1]) 1 1 1 1 0 1 1 0 ($10110_2 \times 1_2$ + EZ) 0 0 0 0 0 0 0 ($10110_2 \times 0_2$ + EZ) 1 1 0 1 1 0 ($10110_2 \times 1_2$ + EZ) + 0 0 0 0 0 ($10110_2 \times 0_2$ + EZ) ~~1~~ **1 1 0 0 1 1 1 0 0**	110011100_2
	II	0 0 0 0 0 (početni zbir 0) + 0 0 0 0 0 0 ($10110_2 \times 0_2$ + EZ) 0 0 0 0 0 0 0 (EZ) + 1 1 0 1 1 0 ($10110_2 \times 1_2$ + EZ) 1 1 1 0 1 1 0 0 (EZ) + 0 0 0 0 0 0 ($10110_2 \times 0_2$ + EZ) 1 1 1 1 0 1 1 0 0 (EZ) + 1 1 0 1 1 0 ($10110_2 \times 1_2$ + EZ) 1 1 1 0 0 1 1 1 0 0 (EZ)	

[1] EZ – Ekstenzija znaka

		0 0 0 0 0 0 $(10110_2 \times 0_2 + EZ)$ ~~1~~ **1 1 0 0 1 1 1 0 0**	
$110.01_2 \times 10.111_2$	I	1 1 1 1 1 1 0 0 1 $(11001_2 \times 1_2 + EZ)$ 1 1 1 1 1 0 0 1 $(11001_2 \times 1_2 + EZ)$ 1 1 1 1 0 0 1 $(11001_2 \times 1_2 + EZ)$ 0 0 0 0 0 0 $(11001_2 \times 0_2 + EZ)$ + 0 0 1 1 1 $(11001_2 \times 1_2 \rightarrow$ DK[2]) ~~1~~ **0 0 0 1 1 1 1 1 1**	0001.11111_2
	II	0 0 0 0 0 (početni zbir 0) + 1 1 1 0 0 1 $(11001_2 \times 1_2 + EZ)$ 1 1 1 1 0 0 1 (EZ) + 1 1 1 0 0 1 $(11001_2 \times 1_2 + EZ)$ 1 1 1 0 1 0 1 1 (EZ) + 1 1 1 0 0 1 $(11001_2 \times 1_2 + EZ)$ 1 1 1 0 0 1 1 1 1 (EZ) + 0 0 0 0 0 0 $(11001_2 \times 0_2 + EZ)$ 1 1 1 1 0 0 1 1 1 1 (EZ) 0 0 0 1 1 1 $(11001_2 \times 1_2$ -> DK) ~~1~~ **0 0 0 1 1 1 1 1 1**	
$0110.1_2 \times 1.0110_2$	I	0 0 0 0 0 0 0 0 0 $(01101_2 \times 0_2)$ 0 0 0 0 1 1 0 1 $(01101_2 \times 1_2)$ 0 0 0 1 1 0 1 $(01101_2 \times 1_2)$ 0 0 0 0 0 0 $(01101_2 \times 0_2)$ + 1 0 0 1 1 $(01101_2 \times 1_2$ -> DK) ~~1~~ **1 0 1 1 1 1 1 1 0**	1011.11110_2
	II	0 0 0 0 0 (početni zbir 0) + 0 0 0 0 0 0 $(0110.1_2 \times 0_2)$ 0 0 0 0 0 0 0 + 0 0 1 1 0 1 $(0110.1_2 \times 1_2)$ 0 0 0 1 1 0 1 0 + 0 0 1 1 0 1 $(0110.1_2 \times 1_2)$ 0 0 1 0 0 1 1 1 0 + 0 0 0 0 0 0 $(0110.1_2 \times 0_2)$ 0 0 0 1 0 0 1 1 1 0 1 1 0 0 1 1 $(0110.1_2 \times 1_2)$ ~~1~~ **1 0 1 1 1 1 1 1 0**	

Zadatak 2.6.

Izvršiti deljenje neoznačenih binarnih brojeva:

11010111/1011, 1001101001/101, 10001011/1101

[2] DK – Drugi komplement

Rešenje:

Na osnovu postupka opisanog u 1.7 važi:

Izraz	Postupak	Rezultat
11010111/1011	1 1 0 1 0 1 1 1 : 1 0 1 1 = 10011 - 1 0 1 1 $1011_2 \times \mathbf{1}$ 0 0 1 0 0 Dopisana cifra 0 - 0 0 0 0 $1011_2 \times \mathbf{0}$ 1 0 0 1 Dopisana cifra 1 - 0 0 0 0 $1011_2 \times \mathbf{0}$ 1 0 0 1 1 Dopisana cifra 1 - 1 0 1 1 $1011_2 \times \mathbf{1}$ 1 0 0 0 1 Dopisana cifra 1 - 1 0 1 1 $1011_2 \times \mathbf{1}$ 1 1 0 Ostatak	10011_2
1001101001/101	1 0 0 1 1 0 1 0 0 1 : 1 0 1 = 01111011 - 0 0 0 $101_2 \times \mathbf{0}$ 1 0 0 1 Dopisana cifra 1 - 1 0 1 $101_2 \times \mathbf{1}$ 1 0 0 1 Dopisana cifra 1 - 1 0 1 $101_2 \times \mathbf{1}$ 1 0 0 0 Dopisana cifra 0 - 1 0 1 $101_2 \times \mathbf{1}$ 0 1 1 1 Dopisana cifra 1 - 1 0 1 $101_2 \times \mathbf{1}$ 0 1 0 0 Dopisana cifra 0 - 0 0 0 $101_2 \times \mathbf{0}$ 1 0 0 0 Dopisana cifra 0 - 1 0 1 $101_2 \times \mathbf{1}$ 0 1 1 1 Dopisana cifra 1 - 1 0 1 $101_2 \times \mathbf{1}$ 1 0 Ostatak	1111011_2
10001011/1101	1 0 0 0 1 0 1 1 : 1 1 0 1 = 01010 - 0 0 0 0 $1101_2 \times \mathbf{0}$ 1 0 0 0 1 Dopisana cifra 1 - 1 1 0 1 $1101_2 \times \mathbf{1}$ 0 1 0 0 0 Dopisana cifra 0 - 0 0 0 0 $1101_2 \times \mathbf{0}$ 1 0 0 0 1 Dopisana cifra 1 - 1 1 0 1 $1101_2 \times \mathbf{1}$ 0 1 0 0 1 Dopisana cifra 1 - 0 0 0 0 $1101_2 \times \mathbf{0}$ 1 0 0 1 Ostatak	01010_2

3. Zadaci za samostalni rad

Zadatak 3.1.

Algoritamskim računanjem, korak po korak, izračunati vrednosti brojeva A, B i C a zatim ih sortirati u opadajućem redosledu:

$A = 001011_{KMV} - 100101_{KMV}$
$B = 010110_{KO} \times 100110_{KO}$
$C = 3321_{4KO} - 3320_{4KO}$

Zadatak 3.2.

a) Odrediti vrednosti X, Y i Z:

$$X_{16KO} = -13.3125_{10}$$

$$Y_{3KMV} = 520_{9KO}$$

b) Naznačiti da li je dati iskaz tačan ili netačn, ukoliko je na raspolaganju 5 cifara

$$10011_{KMV} + 11111_{KMV} = 10111_{ZA} - 01011_{ZA}$$

Napomena: Ukoliko dođe do prekoračenja, naznačiti to i nastaviti sa petobitnim dobijenim rezultatom.

c) Naznačiti da li su dati iskazi tačni ili netačni, ukoliko je na raspolaganju proizvoljan broj cifara

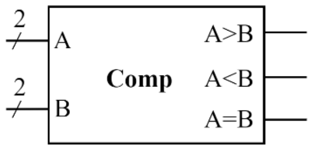
$$243_{5KO} + 304_{5KO} > 10100_{KO} * 111000_{KO}$$

$$F243_{16KMV} - F279_{16KMV} = 111100_{ZA} - 100111_{ZA}$$

Napomena: Ukoliko broj nema oznaku KMV, KO ili ZA u indeksu smatrati da je neoznačen.

Zadatak 3.3.

a) Na slici 3.3.1 je predstavljen jedan dvobitni komparator. Realizovati četvorobitni komparator isključivo korišćenjem komparatora predstavljenog na slici i I, ILI i NI logičkih kola.



Slika 3.3.1

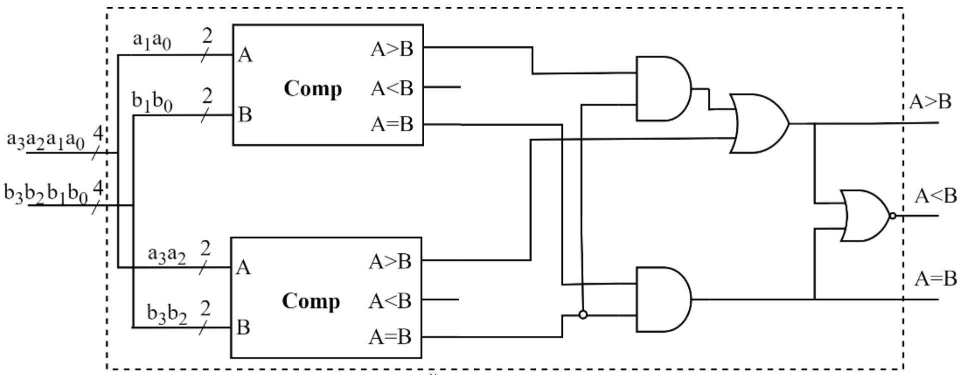
b) Projektovati kombinacionu mrežu koja na izlazu generiše petobitni binarni broj određen funkcijom

$$RES=max(A, 2B, C/2) \quad (3.3.1)$$

gde su A, B i C neoznačeni četvorobitni brojevi. Na raspolaganju su četvorobitni komparatori i potrebna logička kola.

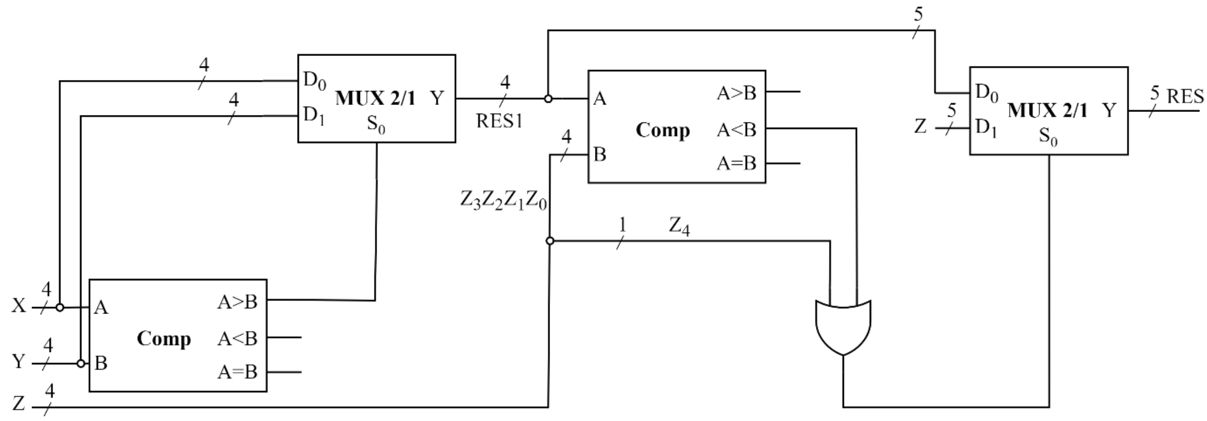
Rešenje:

a) Na slici 3.3.2 je predstavljena realizacija četvorobitnog komparatora korišćenjem dvobitnog komparatora sa slike 7 i I/ILI logičkih kola



Slika 3.3.2 – Četvorobitni komparator

b) Na slici 3.3.3 je prikazana realizacija funkcije RES korišćenjem četvorobitnih komparatora.



Slika 3.3.3 – Realizacija funkcije 3.3.1

Zadatak 3.4.

a) Realizovati sabirač dva dvobitna binarna broja korišćenjem minimalnog broja osnovnih logičkih kola

b) Korišćenjem sabirača iz tačke a) realizovati logičku funkciju:

$$Y = \begin{cases} (A+1)(B+1);\, A < B \\ 2(A+1)(B+1);\, A > B \\ 0;\, A = B \end{cases} \tag{3.4.1}$$

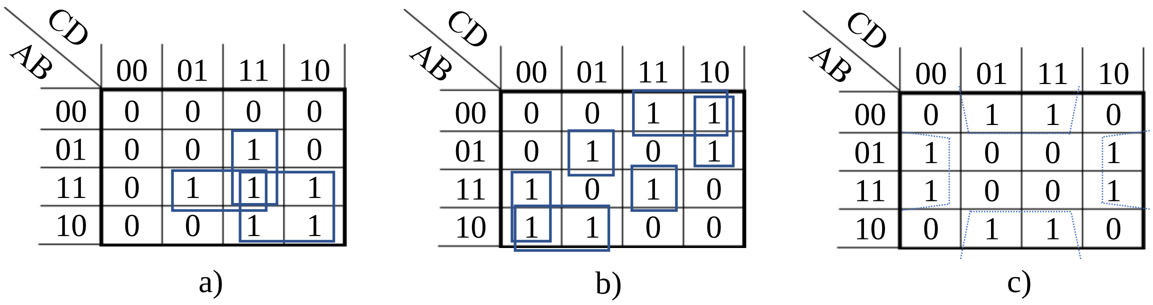
Rešenje:

a) Tabela 3.4.1 predstavlja funkcionalnu tabelu sabirača sa dva dvobitna ulaza.

Tabela 3.4.1 – Funkcionalna tabela dvoulaznog dvobitnog sabirača iz tačke a

a_3	a_2	a_1	a_0	s_2	s_1	s_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Na osnovu tabele 3.4.1 popunjavamo Karnoove karte, za svaki od izlaza posebno, kako bi izveli funkcije izlaza u minimalnoj formi. Na slici 3.4.1 je prikazan sadržaj Karnoove karte za svaki od izlaza dvoulaznog dvobitnog sabirača.



Slika 3.4.1 - Sadržaj Karnoovih karti za izalzne signale a) S_2, b) S_1 i c) S_0

Za realizaciju funkcija izlaznih signala ćemo koristiti formu ZP koja se generiše preklapanjem oblasti koje su na slici 10 uokvirene plavim pravougaonicima. Dobijene funkcije su:

$$s_0 = b_0\overline{a_0} + a_0\overline{b_0} \tag{3.4.2}$$

$$s_1 = a_1\overline{a_0}\overline{b_1} + a_1\overline{b_1}\overline{b_0} + \overline{a_1}\,\overline{a_0}b_1 + b_1\overline{b_0}\,\overline{a_1} + \overline{a_1}a_0\overline{b_1}b_0 + a_1a_0b_1b_0 \tag{3.4.3}$$

$$s_2 = b_1a_1 + a_1a_0b_0 + b_1b_0a_0 \tag{3.4.4}$$

Funkcije (3.4.2 – 3.4.4) se mogu realizovati u još minimalnijoj formi ukoliko su na raspolaganju XOR kola. Uprošćene funkcije su

$$s_0 = a_0 \oplus b_0 \tag{3.4.5}$$

$$s_1 = a_0b_0 \oplus (a_1b_1) \tag{3.4.6}$$

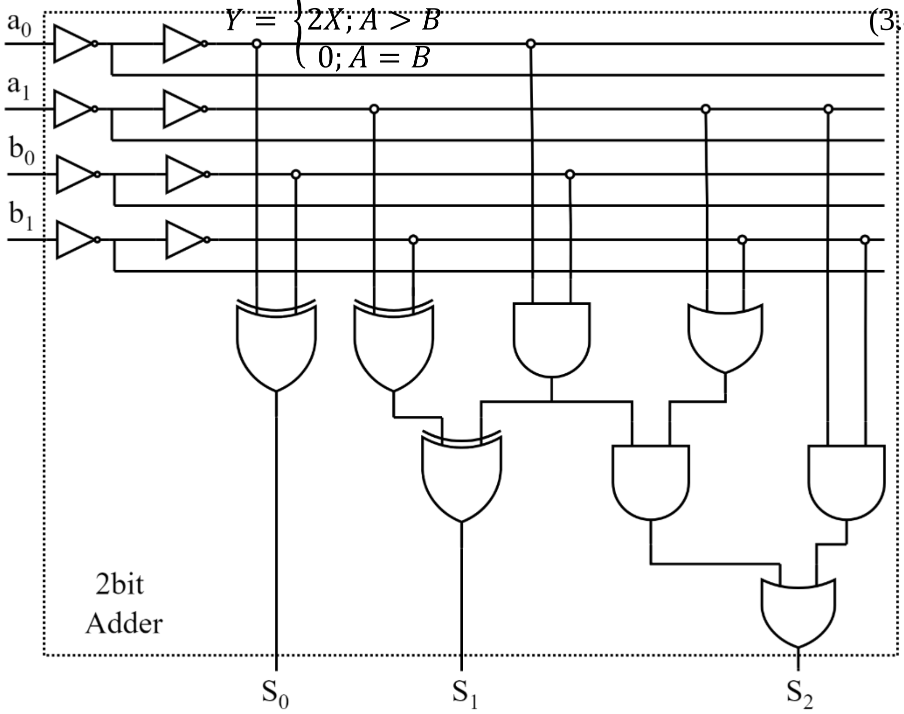
$$s_2 = b_1a_1 + a_0b_0(b_1 + a_1) \tag{3.4.7}$$

Na slici 3.4.2 je prikazana realizacija dvoulaznog dvobitnog sabirača, sa ulazima A(a_1a_0) i B(b_1b_0), dobijena na osnovu funkcija (3.4.5– 3.4.7).

Slika 3.4.2 - Implementacija dvoulaznog dvobitnog sabirača

b) Funkciju (3.4.1) možemo zapisati kao

$$Y = \begin{cases} X; A < B \\ 2X; A > B \\ 0; A = B \end{cases} \qquad (3.4.8)$$



gde je X jednako:

$$X=S(B+1) \qquad (3.4.9)$$

Dakle, funkcija X se koristi i u slučaju $A > B$ i u slučaju $A < B$. U slučaju $A < B$ se koristi neizmenjena vrednost funkcije a u slučaju $A > B$ koristi se vrednost funkcije pomerena za 1 bit u levo dok se bit najmanje težine setuje na logičku nulu.

Postoje mnogobrojni načini za realizaciju funkcije X, međutim, u ovom primeru će biti predstavljena realizacije funkcije X dobijena korišćenjem kola srednjeg stepena integracije.

Funkciju X možemo posmatrati na sledeći način:

ako je B = 0, tada je X=S (3.4.10)

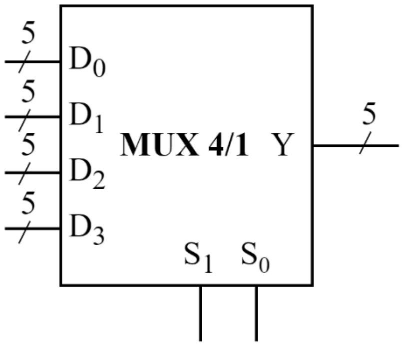
ako je B = 1, tada je X = 2S (3.4.11)

ako je B = 2, tada je X = 3S (3.4.12)

Ako je B = 3, tada je X = 4S (3.4.13)

Dakle, vrednost dvobitnog broja B(b_1b_0) određuje koja od funkcija (S, 2S, 3S ili 4S) će biti dodeljena funkciji X. Ovakva (selekciona) logika se najlakše realizuje korišćenjem multipleksera 4/1. Pre nego predstavimo način povezivanja multipleksera, neophodno je prodiskutovati o širini ulaznih i izlaznih linija multipleksera. U prethodnoj tački smo videli da je za predtavljanje rezltata sabiranja dva dvobitna broja potrebno minimum 3 bita zbog toga što je najveći broj koji se dobija sabiranjem dva neoznačena dvobitna broja jednak 6. Pošto je u slučaju realizacije ove tačke zadatka na jedan ulaz sabirača (ulaz A) doveden neoznačeni dvobitni broj A(a_1a_0) a na drugi ulaz (ulaz B) konstanta 1 ($b_1b_0 = 01$), jasno je da se kao najveća vrednost rezultata sabiranja može obiti broj 4. Množenjem ovog rezultata sa brojem 2 (2.24), najveći broj koji se može dobiti u slučaju B=1 je 8 i za predstavu ovog rezltata nam je potrebno 4 bita. U slučaju množenja izlaza sabirača S sa 3, najveći rezultat koji možemo dobiti je 12, što se takođe može predstaviti sa 4 bita, dok je u slučaju množenja izlaza sabirača sa brojem 4 najveći očekivani rezultat 16 za čiju predstavu nam je potrebno 5 bita. Pošto su širine ulaznih i izlaznih linija određene rezultatom za čiju predstavu nam je potrebno najviše bita, jasno je da moramo uzeti multiplekser čija je širina ulaznih i izlaznih linija jednaka 5. Opisani multiplekser je predstavljen na slici 3.4.3

Slika 3.4.3 - Multiplekser 4/1 sa ulaznim i izlaznim linijama širine 5 bita



Pre nego predstavimo način na koji je multiplekser iskorišćen za realizaciju logike opisane sa (3.4.8 do 3.4.9) potrebno je izvršiti analizu načina za generisanje ulaznih signala D_0, D_1, D_2 i D_3. Generisanje signala opisanih u (3.4.10), (3.4.11) i (3.4.13) se realizuje jednostavnim pomeranjem za 0, 1 i 2 mesta u levo, respektivno. Međutim, za generisanje ulaznog signala D_2 (za slučaj B = 2) nije moguće koristiti vrednost dobijenu na izlazu sabirača već je potrebno realizovati kombinacionu mrežu koja se dobija na osnovu sadržaja kobinacione tabele 3.4.2

Tabela 3.4.2 - Kombinaciona tabela za funkciju 3(A+1)

a_1	a_0	D_{2-4}	D_{2-3}	D_{2-2}	D_{2-1}	D_{2-0}
0	0	0	0	0	1	1
0	1	0	0	1	1	0
1	0	0	1	0	0	1
1	1	0	1	1	0	0

Na osnovu Tabele 3.4.3 dobijamo funkcije izlaznih signala kombinacione mreže (ulazni signali multipleksera)

$$D_{2-4} = 0 \qquad (3.4.14)$$

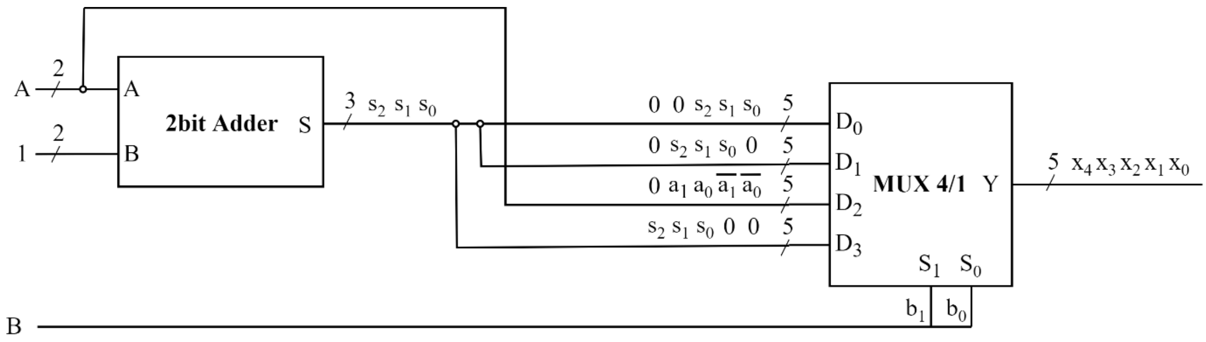
$$D_{2-3} = a_1 \quad (3.4.15)$$

$$D_{2-2} = a_0 \quad (3.4.16)$$

$$D_{2-1} = \overline{a_1} \quad (3.4.17)$$

$$D_{2-0} = \overline{a_0} \quad (3.4.18)$$

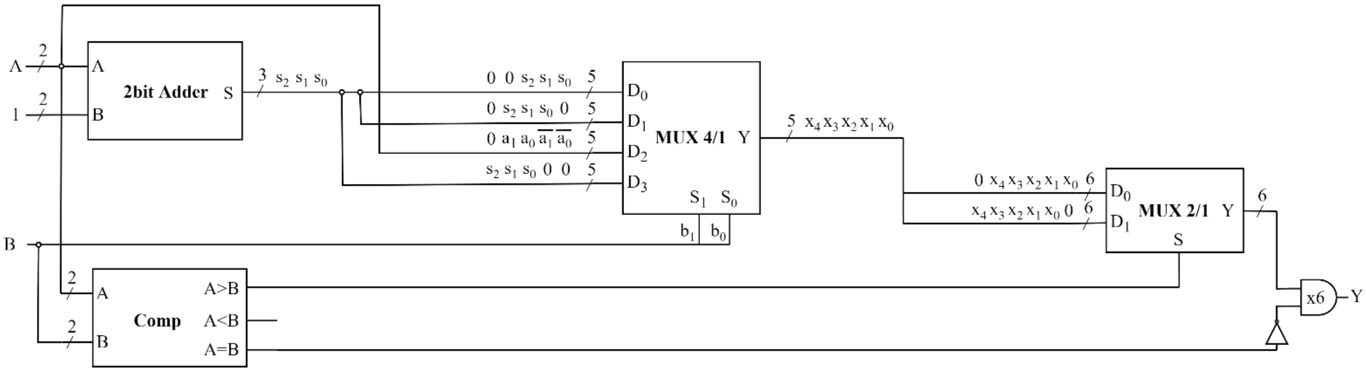
Na slici 3.4.4 je je predstavljen deo kombinacione mreže koji realizuje funkciju X.



Slika 3.4.4 - Realizacija funkcije X

Za realizaciju kompletne funkcije (3.4.1) neophodno je implementirati logiku koja na osnovu poređenja dva neoznačena dvobitna broja A i B generiše izlaznu funkciju Y, tj. ukoliko je $A < B$ funkcija Y uzima vrednost X dok u suprotnom uzima vrednost 2X. Za realizaciju logike proeđenja dva neoznačena binarna broja, koristićemo komparator (kao u prethodnom zadatku) koji realizuje funkcionalnost poređenja dva neoznačena broja. Rezultat poređenja ($A > B$) ćemo iskoristiti kao selekcioni signal multipleksera 2/1 kojim se definiše vrednost funkcije Y. Množenje sa 2 smo realizovali koristeći istu logiku kao i u slučaju generisanja signala X.

Na slici 3.4.4 je predstavljena implementirana kombinaciona mreža.



Slika 3.4.4 – Kompletna realizacija funkcije 3.4.1

Zadatak 3.5.

c) Izvršiti sabiranje označenih brojeva datih u komplementu osnove ukoliko je na raspolaganju 5 cifara za predstavu rezultata. Operaciju sabiranja realizovati u brojnom sistemu u kome su brojevi i dati. Pored svakog rezultata naznačiti da li prilikom operacije sabiranja dolazi do prekoračenja

$010_2 + 0011_2$, $11_2 + 110_2$, $0110_2 + 1011_2$, $1100_2 + 0101_2$, $0101_2 + 0110_2$, $1101_2 + 1011_2$, $1.0_2 + 10.1_2$, $435_{10} + 834_{10}$, $A32F_{16} + 476_{16}$, $324_7 + 365_7$

d) Izvršiti oduzimanje označenih brojeva datih u komplementu osnove ukoliko je na raspolaganju 5 cifara za predstavu rezultata. Operaciju oduzimanja realizovati u brojnom sistemu u kome su brojevi i dati. Pored svakog rezultata naznačiti da li prilikom operacije sabiranja dolazi do prekoračenja

$010_2 - 1101_2$, $11_2 - 010_2$, $0110_2 - 0101_2$, $1100_2 - 1011_2$, $01.01_2 - 10.10_2$, $1101_2 - 0101_2$, $10_2 - 011_2$, $435_{10} - 166_{10}$, $A32F_{16} - 524_{16}$, $8135_{16} - FA3B_{16}$, $364_7 - 302_7$

Zadatak 3.6.

e) Izvršiti sabiranje označenih brojeva datih u komplementu maksimalne vrednosti ukoliko je na raspolaganju 5 cifara za predstavu rezultata. Operaciju sabiranja realizovati u brojnom sistemu u kome su brojevi i dati. Pored svakog rezultata naznačiti da li prilikom operacije sabiranja dolazi do prekoračenja

$010_2 + 0011_2$, $11_2 + 110_2$, $0110_2 + 1011_2$, $1100_2 + 0101_2$, $0101_2 + 0110_2$, $1101_2 + 1011_2$, $1.0_2 + 10.1_2$, $435_{10} + 834_{10}$, $A32F_{16} + 476_{16}$, $324_7 + 365_7$

f) Izvršiti oduzimanje označenih brojeva datih u komplementu maksimalne vrednosti ukoliko je na raspolaganju 5 cifara za predstavu rezultata. Operaciju oduzimanja realizovati u brojnom sistemu u kome su brojevi i dati. Pored svakog rezultata naznačiti da li prilikom operacije sabiranja dolazi do prekoračenja

$010_2 - 1101_2$, $11_2 - 010_2$, $0110_2 - 0101_2$, $1100_2 - 1011_2$, $01.01_2 - 10.10_2$, $1101_2 - 0101_2$, $10_2 - 011_2$, $435_{10} - 166_{10}$, $A32F_{16} - 524_{16}$, $8135_{16} - FA3B_{16}$, $364_7 - 302_7$