ELEKTROTEHNIČKI FAKULTET
UNIVERZITETA U BEOGRADU

predmet: RAČUNARSKI VLSI SISTEMI
Domaći zadatak broj 1

PETOBITNI SABIRAČ SA USLOVNIM SUMAMA

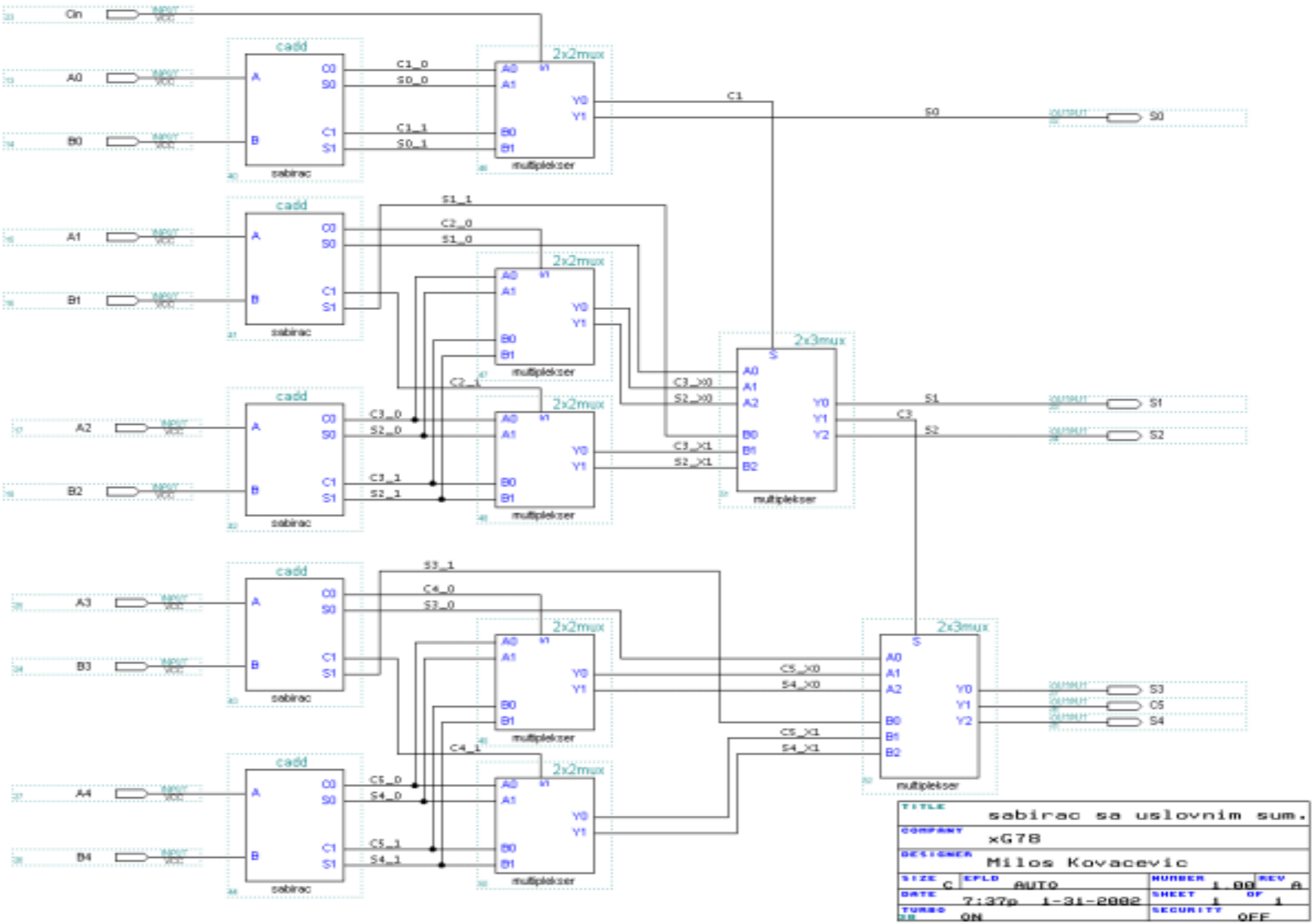
profesor: Veljko Milutinović
asistent: Gvozden Marinković
autor: Miloš Kovačević 250/97

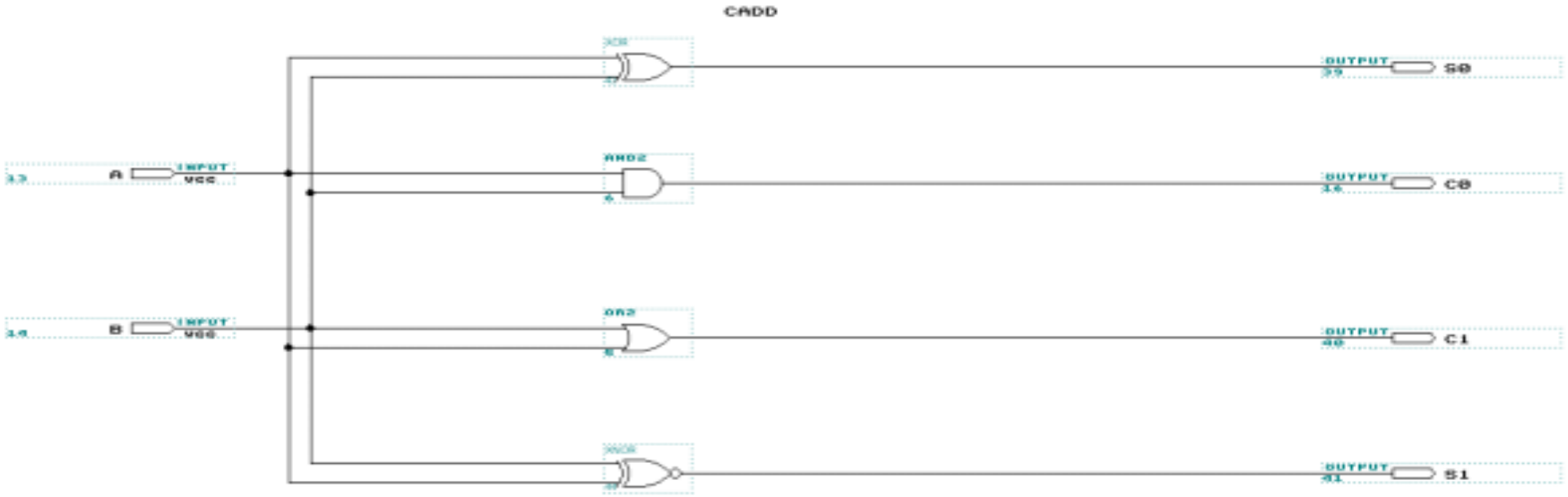
Beograd,
29.01.2002.

SADRŽAJ

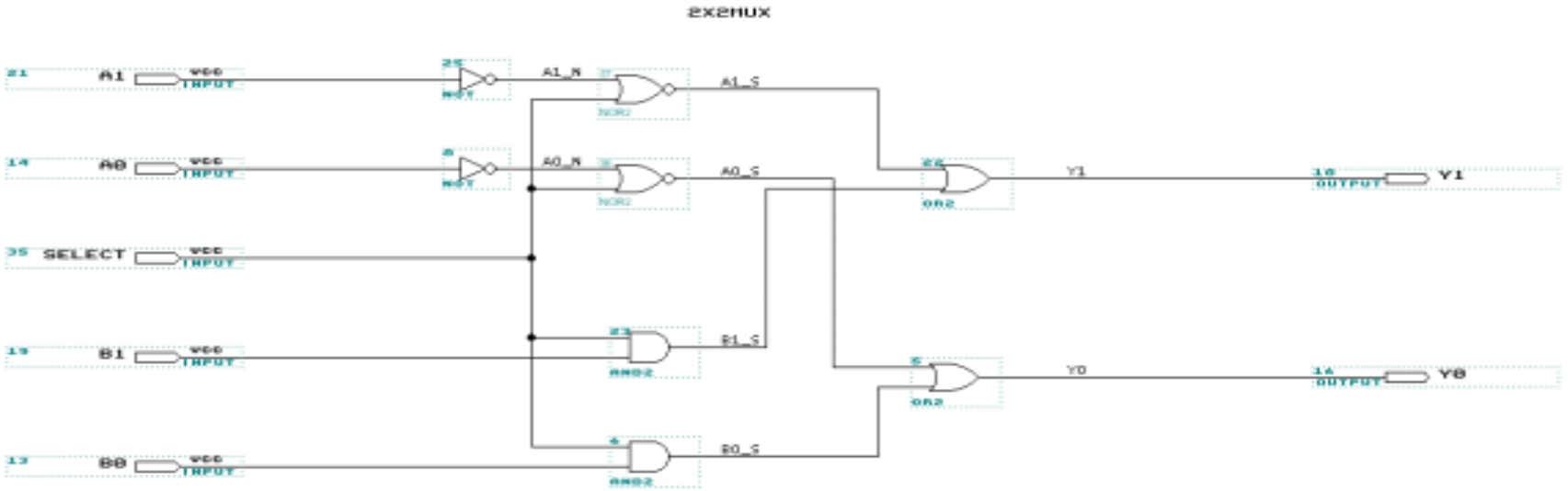
1. Projekat petobitnog sabirača sa uslovnim sumama u programu Altera MAX+plus II

1.1 Električne šeme

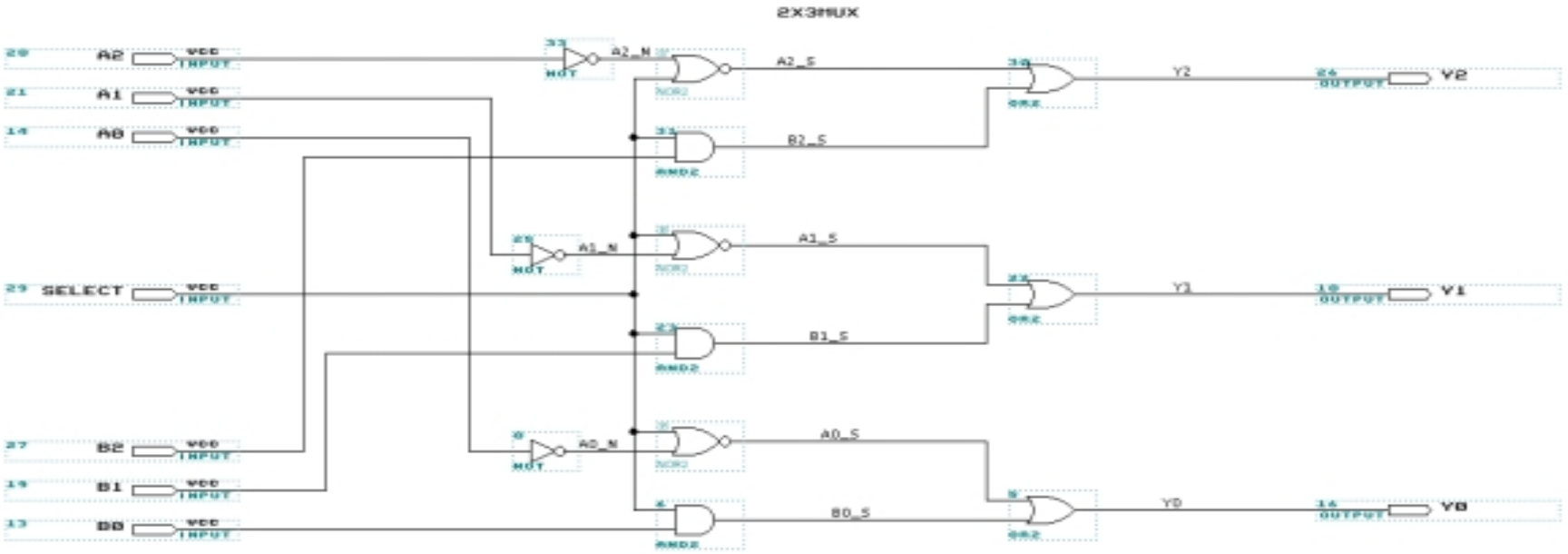

Cin
A0
B0
A1
B1
A2
B2
A3
B3
A4
B4
cadd
A
B
C0
S0
C1
S1
sabirac
C1_0
S0_0
C1_1
S0_1
2x2mux
A0
A1
B0
B1
Y0
Y1
multiplekser
C1
S0
S1_1
C2_0
S1_0
C2_1
C3_0
S2_0
C3_1
S2_1
2x3mux
S
A2
B0
B1
B2
C3_X0
S2_X0
C3_X1
S2_X1
S1
C3
S2
S3_1
C4_0
S3_0
C4_1
C5_0
S4_0
C5_1
S4_1
C5_X0
S4_X0
C5_X1
S4_X1
S3
C5
S4
TITLE
sabirac sa uslovnim sum.
COMPANY
xG78
DESIGNER
Milos Kovacevic
SIZE
C
EPLD
AUTO
NUMBER
1.00
REV
A
DATE
7:37p 1-31-2002
SHEET
1 OF 1
TURBO
ON
SECURITY
OFF



TITLE	Uslovni sabirac		
COMPANY	xG78		
DESIGNER	Milos Kovacevic		
SIZE C	CPLD AUTO	NUMBER 1.00	REV A
DATE 1:53p 1-28-2002		SHEET 1 OF 1	
TURBO ON		SECURITY OFF	



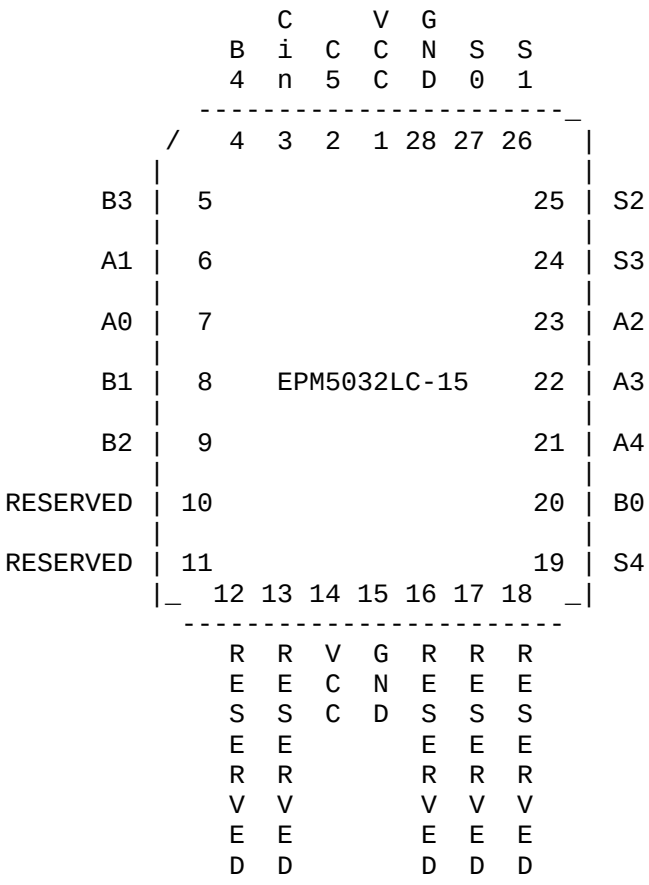
TITLE	2x2 Multiplekser
COMPANY	xG78
DESIGNER	Milos Kovacevic
SIZE C / CPLD AUTO / NUMBER 1.00 / REV A	
DATE 10:24p 1-29-2002 / SHEET 1 OF 1	
TURBO ON / SECURITY OFF	



TITLE	2x3 Multiplekser
COMPANY	xG78
DESIGNER	Milos Kovacevic
SIZE C / EPLD AUTO / NUMBER 1.00 / REV A	
DATE 10:24p 1-29-2002 / SHEET 1 OF 1	
TURBO ON / SECURITY OFF	

1.2 Izveštaj AlteraMAX+plus kompajlera

```
***** Project compilation was successful
Title: sabirac sa uslovnim sum.
Company: xG78
Designer: Milos Kovacevic
Rev: A
Date:  7:37p  1-31-2002

** DEVICE SUMMARY **
Chip/                    Input   Output   Bidir         Shareable
POF       Device         Pins    Pins     Pins     LCs  Expanders  % Utilized
5csadd    EPM5032LC-15    11       6        0      13      30          40 %
User Pins:                11       6        0
-----
***** Logic for device '5csadd' compiled without errors.
Device: EPM5032LC-15
Device Options:
    Security Bit                                = ON
```




```
N.C. = Not Connected.
VCC = Dedicated power pin, which MUST be connected to VCC.
GND = Dedicated ground pin or unused dedicated input, which MUST be connected to
GND.
RESERVED = Unused I/O pin, which MUST be left unconnected.
-----
```

```
** RESOURCE USAGE **
                                                 Shareable
Logic Array Block     Logic Cells    I/O Pins    Expanders
A:     LC1 - LC32    13/32( 40%)   9/16( 56%)  36/64( 56%)
Total dedicated input pins used:                  8/8      (100%)
Total I/O pins used:                              9/16     ( 56%)
Total logic cells used:                          13/32     ( 40%)
Total shareable expanders used:                  30/64     ( 46%)
Total shareable expanders not available (n/a):    6/64     (  9%)
Average fan-in:                                   5.38
Total fan-in:                                     70
Total input pins required:                       11
Total output pins required:                       6
Total bidirectional pins required:                0
Total logic cells required:                      13
Total flipflops required:                         0
Total shareable expanders in database:           30
Synthesized logic cells:                          7/  32   ( 21%)
-----
** INPUTS **
                                        Shareable
                                        Expanders      Fan-In    Fan-Out
 Pin     LC  LAB  Primitive    Code    Total Shared n/a INP  FBK  OUT  FBK  Name
   7      -   -       INPUT               0      0   0    0    0    3    5  A0
   6      -   -       INPUT               0      0   0    0    0    2    4  A1
  23      -   -       INPUT               0      0   0    0    0    1    5  A2
  22      -   -       INPUT               0      0   0    0    0    2    2  A3
  21      -   -       INPUT               0      0   0    0    0    1    2  A4
  20      -   -       INPUT               0      0   0    0    0    3    5  B0
   8      -   -       INPUT               0      0   0    0    0    2    4  B1
   9      -   -       INPUT               0      0   0    0    0    1    3  B2
   5   (31)  (A)      INPUT               2      0   2    0    0    2    2  B3
   4   (29)  (A)      INPUT               2      0   2    0    0    1    2  B4
   3   (27)  (A)      INPUT               2      0   2    0    0    3    5  Cin

Code:
s = Synthesized pin or logic cell
+ = Synchronous flipflop
! = NOT gate push-back
r = Fitter-inserted logic cell
-----
** OUTPUTS **
                                        Shareable
                                        Expanders      Fan-In    Fan-Out
 Pin     LC  LAB  Primitive    Code    Total Shared n/a INP  FBK  OUT  FBK  Name
   2     25    A     OUTPUT              2      0   0    4    1    0    0  C5
  27     23    A     OUTPUT              0      0   0    3    0    0    0  S0
  26     21    A     OUTPUT              7      0   0    5    0    0    0  S1
  25     19    A     OUTPUT              8      0   0    7    2    0    0  S2
  24     17    A     OUTPUT              0      0   0    2    1    0    0  S3
  19     15    A     OUTPUT              0      0   0    0    5    0    0  S4

Code:
s = Synthesized pin or logic cell
+ = Synchronous flipflop
! = NOT gate push-back
r = Fitter-inserted logic cell
```

1.3 Opis rada sabirača

Zbog velikog kašnjenja serijskih sabirača za višebitne brojeve, u cilju ubrzanja, može se koristiti sabirač sa uslovnim sumama. Princip sabirača je da sa paralelno sabiraju bitovi svih razreda i to za obe moguće vrednosti ulaznog prenosa. Dakle kao rezultat sabiranja bitova i. razreda Ai i Bi imamo dve sume - jednu u slučaju ulaznog prenosa 0 (Si_0) i drugu u slučaju ulaznog prenosa 1 (Si_1). Isto važi i za prenose u sledeći razred (Ci+1_0 i Ci+1_1). Blokovi koji obavljaju ove jednostavne operacije označeni su kao cadd - conditional adder.

Ulazni prenos za ceo sabirač C0 određuje koja od dve sume i prenosa iz uslovnog sabirača 0. razreda je validna, što je realizovano korišćenjem dvobitnog dvoulaznog multipleksera (mux2x2) na čije su ulaze vezani rezultati u slučaju ulaznog prenosa 0 i 1, a kao signal selekcije upravo doveden C0. Dakle na izlazu ovog multipleksera imamo već značajne rezultate - S0 i C1.

Pošto već imamo i potencijalne vrednosti za prenos C2 iskoristićemo još dva multipleksera. Na ulaze oba ćemo dovesti iste signale - izlaze uslovnog sabirača 2. razreda - cadd2, ali ćemo kao signal selekcije na prvi multiplekser dovesti C2_0 (prenos u 2. razred pod uslovom C1=0) a na drugi C2_1. Na isti način vezujemo još dva multipleksera na izlaze cadd4, a kao kontrolne signale C4_0 i C4_1.

Pošto smo na osnovu ulaznog prenosa C0 izabrali prave vrednosti za S0 i C1, dakle znamo prenos u 1. razred korišćenjem jednog trobitnog dvoulaznog multipleksera (mux2x3) na čiji je kontrolni ulaz vezan C1, izabraćemo jednu od dve sume S1_0 ili S1_1 i izlaze jednog od dva multipleksera na čije su ulaze dovedeni rezultati sabiranja 2. razreda. Izlazi C3_X0, C3_X1, S2_X0 i S2_X1 ta dva multipleksera su prenos u 3.razred pod uslovom C1=0, C1=1 i sume drugog razreda pod uslovom C1=0 i C1=1, respektivno. Izlali novog trobitnog multipleksera su S1, C3 i S2.

Na identičan način kao što smo na osnovu C1 izabrali S1, C3 i S2, sada pomoću novog trobitnog dvoulaznog multipleksera biramo validne vrednosti za S3, C5 i S4, a na osnovu 'kandidata' - S3_0, C5_X0, S4_X0 i S3_1, C5_X1, S4_X1.

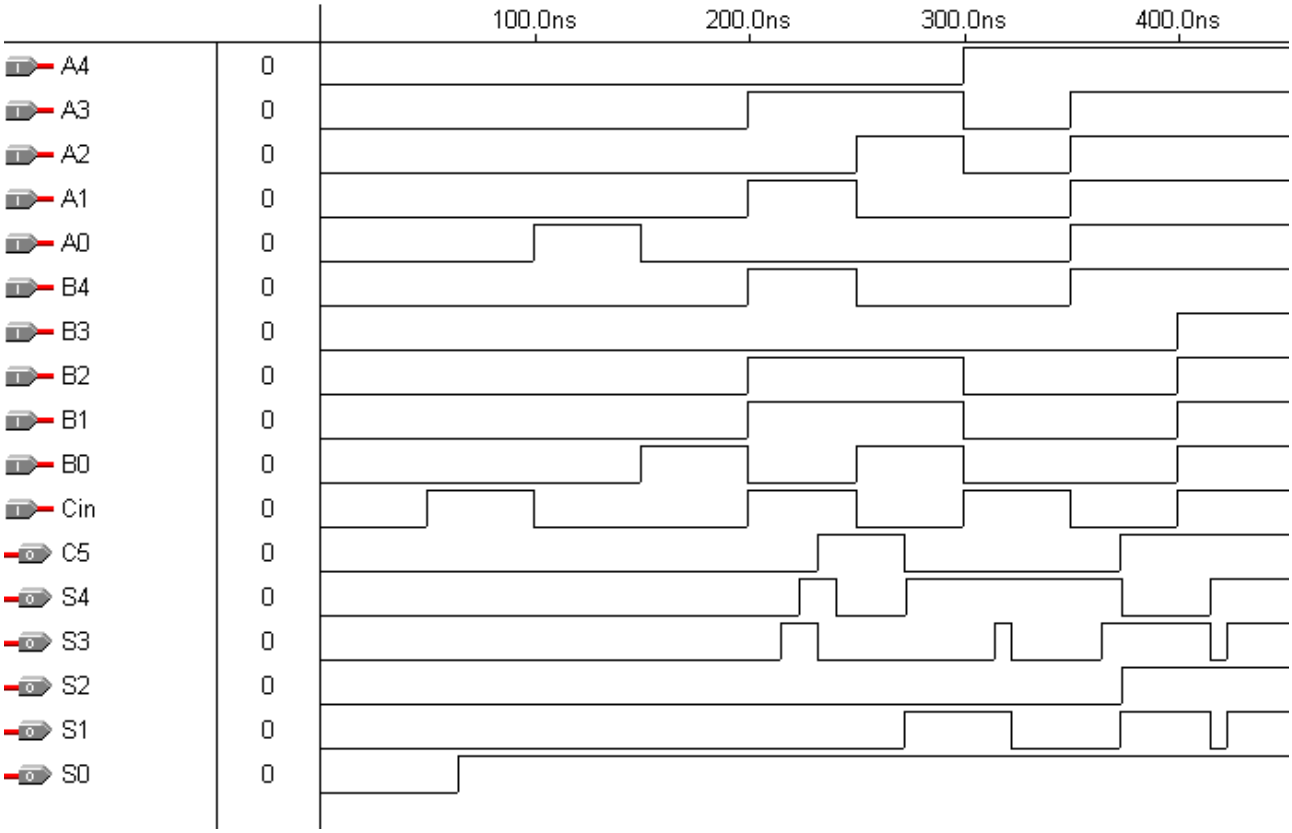
Sada već imamo sve potrebne reezultate - S0..S4 i C5!

1.4 Simulacija sabirača

1.4.1 Izveštaj simulacije - ulazni vektori i rezultati

```
% ULAZNI VEKTORI ZA SIMULACIJU SABIRACA %
START 0 ;   % vreme pocetka simulacije %
STOP 450 ;  % vreme zavrsetka simulacije %
INTERVAL 50 ;     % vreme izmedju 2 uzastopna zadavanja %
            % vrednosti ulaznim signalima %
INPUTS A4 A3 A2 A1 A0   B4 B3 B2 B1 B0   Cin ;
PATTERN           % test vektori %
0>    0 0 0 0 0   0 0 0 0 0   0
50>   0 0 0 0 0   0 0 0 0 0   1
100>  0 0 0 0 1   0 0 0 0 0   0
150>  0 0 0 0 0   0 0 0 0 1   0
200>  0 1 0 1 0   1 0 1 1 0   1
250>  0 1 1 0 0   0 0 1 1 1   0
300>  1 0 0 0 0   0 0 0 0 0   1
350>  1 1 1 1 1   1 0 0 0 0   0
400>  1 1 1 1 1   1 1 1 1 1   1
;
OUTPUTS C5   S4 S3 S2 S1 S0 ;%
PATTERN         %% ocekivani rezultati (sa kasnjenjem) %%
= X   X X X X X
= 0   0 0 0 0 0
= 0   0 0 0 0 1
= 0   0 0 0 0 1
= 0   0 0 0 0 1
= 1   0 0 0 0 1
= 0   1 0 0 1 1
= 0   1 0 0 0 0
= 0   1 0 0 0 0
= 1   1 1 1 1 1
;%
-----
% REZULTATI SIMULACIJE %
INPUTS A4 A3 A2 A1 A0 B4 B3 B2 B1 B0 Cin ;
OUTPUTS C5 S4 S3 S2 S1 S0 ;
UNIT ns ;
RADIX HEX ;
PATTERN
%                          C               %
%      A A A A A B B B B B i   C S S S S S %
%      4 3 2 1 0 4 3 2 1 0 n   5 4 3 2 1 0 %
  0.0> 0 0 0 0 0 0 0 0 0 0 0 = 0 0 0 0 0 0
 50.0> 0 0 0 0 0 0 0 0 0 0 1 = 0 0 0 0 0 0
 65.0> 0 0 0 0 0 0 0 0 0 0 1 = 0 0 0 0 0 1
100.0> 0 0 0 0 1 0 0 0 0 0 0 = 0 0 0 0 0 1
150.0> 0 0 0 0 0 0 0 0 0 1 0 = 0 0 0 0 0 1
200.0> 0 1 0 1 0 1 0 1 1 0 1 = 0 0 0 0 0 1
215.0> 0 1 0 1 0 1 0 1 1 0 1 = 0 0 1 1 1 1
223.0> 0 1 0 1 0 1 0 1 1 0 1 = 0 0 1 0 0 1
224.0> 0 1 0 1 0 1 0 1 1 0 1 = 0 1 1 0 0 1
232.0> 0 1 0 1 0 1 0 1 1 0 1 = 1 0 0 0 0 1
250.0> 0 1 1 0 0 0 0 1 1 1 0 = 1 0 0 0 0 1
273.0> 0 1 1 0 0 0 0 1 1 1 0 = 0 0 0 0 1 1
274.0> 0 1 1 0 0 0 0 1 1 1 0 = 0 1 0 0 1 1
300.0> 1 0 0 0 0 0 0 0 0 0 1 = 0 1 0 0 1 1
315.0> 1 0 0 0 0 0 0 0 0 0 1 = 0 1 1 1 0 1
323.0> 1 0 0 0 0 0 0 0 0 0 1 = 0 1 1 0 0 1
332.0> 1 0 0 0 0 0 0 0 0 0 1 = 0 1 0 0 0 1
```

```
350.0> 1 1 1 1 1 1 0 0 0 0 0 = 0 1 0 0 0 1
365.0> 1 1 1 1 1 1 0 0 0 0 0 = 1 1 1 0 0 1
373.0> 1 1 1 1 1 1 0 0 0 0 0 = 1 1 1 1 1 1
374.0> 1 1 1 1 1 1 0 0 0 0 0 = 1 0 1 1 1 1
400.0> 1 1 1 1 1 1 1 1 1 1 1 = 1 0 1 1 1 1
415.0> 1 1 1 1 1 1 1 1 1 1 1 = 1 0 0 0 0 1
423.0> 1 1 1 1 1 1 1 1 1 1 1 = 1 0 0 1 1 1
424.0> 1 1 1 1 1 1 1 1 1 1 1 = 1 1 0 1 1 1
432.0> 1 1 1 1 1 1 1 1 1 1 1 = 1 0 0 1 1 1
441.0> 1 1 1 1 1 1 1 1 1 1 1 = 1 1 1 1 1 1
450.0> X X X X X X X X X X X = X X X X X X
```



1.4.2 Matrica kašnjenja

Delay Matrix

Source \ Destination	C5	S0	S1	S2	S3	S4
A0	24.0ns/49.0ns	15.0ns	23.0ns	23.0ns/24.0ns	24.0ns/49.0ns	24.0ns/49.0ns
A1	32.0ns/41.0ns		23.0ns	23.0ns/24.0ns	32.0ns/41.0ns	24.0ns/41.0ns
A2	32.0ns/41.0ns			23.0ns/24.0ns	32.0ns/41.0ns	24.0ns/41.0ns
A3	15.0ns/23.0ns				15.0ns	24.0ns
A4	15.0ns/23.0ns					24.0ns
B0	24.0ns/49.0ns	15.0ns	23.0ns	23.0ns/24.0ns	24.0ns/49.0ns	24.0ns/49.0ns
B1	32.0ns/41.0ns		15.0ns/23.0ns	23.0ns/24.0ns	32.0ns/41.0ns	24.0ns/41.0ns
B2	32.0ns/41.0ns			15.0ns	32.0ns/41.0ns	24.0ns/41.0ns
B3	15.0ns/23.0ns				15.0ns	24.0ns
B4	15.0ns/23.0ns					24.0ns
Cin	24.0ns/49.0ns	15.0ns	23.0ns	23.0ns/24.0ns	24.0ns/49.0ns	24.0ns/49.0ns

2. Projekat petobitnog sabirača sa uslovnim sumama u programu Active VHDL

2.1 Izvorni kod

```
--    Zaglavlje fajla -------------------------------------------------------
--       Naziv 5csadd - Petobitni sabirac sa uslovnim sumama
--        Opis Sabira dva petobitna broja, tako sto paralelno sabira bit po
--             bit, i to u slucaju ulaznog prenosa 0 i 1, a onda po izracuna-
--             vanju ulaznog prenosa koristi jedan od dva izracunata rezultata
--        Fajl C:\My Documents\etf\ri5vlsi\domaci\mk2002\sabirac\vhdl\sabirac\
--             SRC\5csadd.VHD
--      Namena Domaci zadatak iz predmeta racunarski VLSI sistemi
--    Napomene U realnosti kasnjenje kroz and,or,xor i xnor kola nije isto,ali
--             je ta pretpostavka uvedena u postavci zadatka.
-- Ogranicenja -
--      Greske -
--   Korisceni
--      paketi IEEE.std_logic_1164
--       Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--   Korisceni
--   simulator Active VHDL 3.2 build 207
--     Verzija 1.00 (29/01/2002)
--      Izmene
-------------------------------------------------------------------------------

library IEEE;
use IEEE.std_logic_1164.all;

entity \5csadd\ is
--    Zaglavlje entiteta ---------------------------------------------------
--         Opis Entitet petobitnog sabiraca, sadrzi dva ulazna porta za sabirke
--              ulazni prenos, izlazni petobitni zbir i izlazni prenos.
--  Ogranicenja -
-- Pretpostavke Tk je isto za sva logicka kola, i iznosi 2ns
--      Verzija 1.00
--       Izmene -
--        Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--        Datum 29/01/2002
-------------------------------------------------------------------------------
   generic (
      Tprop:time:=2ns         -- vreme kasnjenja jednog logickog kola
      );
   port (
      A:   in  STD_LOGIC_VECTOR (4 downto 0); -- prvi petobitni sabirak
      Cin: in  STD_LOGIC;                     -- ulazni prenos
      B:   in  STD_LOGIC_VECTOR (4 downto 0); -- drugi petobitni sabirak

      S:   out STD_LOGIC_VECTOR (4 downto 0); -- petobitni zbir
      C5:  out STD_LOGIC                      -- izlazni prenos
   );
end \5csadd\;

architecture mk5csadd001struct of \5csadd\ is
--    Zaglavlje arhitekture ------------------------------------------------
--         Opis Arhitektura strukture petobitnog sabiraca. Brojevi se sabiraju
--              tako sto paralelno sabira bit po bit svakog razreda, i to u
--              slucaju ulaznog prenosa 0 i 1, a onda po izracunavanju tog
--              ulaznog prenosa koristi jedan od dva izracunata rezultata.
--  Ogranicenja Izlaz ima validnu vrednost tek posle kasnjenja od 10*Tk
-- Pretpostavke Kasnjenje je isto za sva logicka kola.
--      Verzija 1.00
```

```
--       Izmene -
--        Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--        Datum 29/01/2002
--------------------------------------------------------------------------------
      signal C1_0:  STD_LOGIC;      -- prenos iz 0. u 1. razred pod uslovom C0=0
      signal S0_0:  STD_LOGIC;      -- zbir 0. razreda pod uslovom C0=0
      signal C1_1:  STD_LOGIC;      -- prenos iz 0. u 1. razred pod uslovom C0=1
      signal S0_1:  STD_LOGIC;      -- zbir 0. razreda pod uslovom C0=1
      signal C2_0:  STD_LOGIC;      -- prenos iz 1. u 2. razred pod uslovom C1=0
      signal S1_0:  STD_LOGIC;      -- zbir 1. razreda pod uslovom C1=0
      signal C2_1:  STD_LOGIC;      -- prenos iz 1. u 2. razred pod uslovom C1=1
      signal S1_1:  STD_LOGIC;      -- zbir 1. razreda pod uslovom C1=1
      signal C3_0:  STD_LOGIC;      -- prenos iz 2. u 3. razred pod uslovom C2=0
      signal S2_0:  STD_LOGIC;      -- zbir 2. razreda pod uslovom C2=0
      signal C3_1:  STD_LOGIC;      -- prenos iz 2. u 3. razred pod uslovom C2=1
      signal S2_1:  STD_LOGIC;      -- zbir 2. razreda pod uslovom C2=1
      signal C4_0:  STD_LOGIC;      -- prenos iz 3. u 4. razred pod uslovom C3=0
      signal S3_0:  STD_LOGIC;      -- zbir 3. razreda pod uslovom C3=0
      signal C4_1:  STD_LOGIC;      -- prenos iz 3. u 4. razred pod uslovom C3=1
      signal S3_1:  STD_LOGIC;      -- zbir 3. razreda pod uslovom C3=1
      signal C5_0:  STD_LOGIC;      -- prenos iz 4. u 5. razred pod uslovom C4=0
      signal S4_0:  STD_LOGIC;      -- zbir 4. razreda pod uslovom C4=0
      signal C5_1:  STD_LOGIC;      -- prenos iz 4. u 5. razred pod uslovom C4=1
      signal S4_1:  STD_LOGIC;      -- zbir 4. razreda pod uslovom C4=1

      signal C3_X0: STD_LOGIC;      -- prenos iz 2. u 3. razred pod uslovom C1=0
      signal S2_X0: STD_LOGIC;      -- zbir 2. razreda pod uslovom C1=0
      signal C3_X1: STD_LOGIC;      -- prenos iz 2. u 3. razred pod uslovom C1=1
      signal S2_X1: STD_LOGIC;      -- zbir 2. razreda pod uslovom C1=1
      signal C5_X0: STD_LOGIC;      -- prenos iz 4. u 5. razred pod uslovom C3=0
      signal S4_X0: STD_LOGIC;      -- zbir 4. razreda pod uslovom C3=0
      signal C5_X1: STD_LOGIC;      -- prenos iz 4. u 5. razred pod uslovom C3=1
      signal S4_X1: STD_LOGIC;      -- zbir 4. razreda pod uslovom C3=1

      signal C1:    STD_LOGIC;      -- prenos iz 0. u 1. razred
      signal C3:    STD_LOGIC;      -- prenos iz 2. u 3. razred

begin
   cadd0: entity cadd(caddbeh)-- kreiranje instance sabiraca 0. razreda
      generic map (
         Tk => Tprop                      -- ovde se precizira kasnjenje Tk
      )
      port map (
         A  => A(0),                      -- veze signala na portove su logicne
         B  => B(0),                      -- na osnovu objasnjenja uloge signala
         C0 => C1_0,
         S0 => S0_0,
         C1 => C1_1,
         S1 => S0_1
      );
   cadd1: entity cadd(caddbeh)-- kreiranje instance sabiraca 1. razreda
      generic map (
         Tk => Tprop
      )
      port map (
         A  => A(1),
         B  => B(1),
         C0 => C2_0,
         S0 => S1_0,
         C1 => C2_1,
         S1 => S1_1
```

```
      );
   cadd2: entity cadd(caddbeh)-- kreiranje instance sabiraca 2. razreda
      generic map (
         Tk => Tprop
      )
      port map (
         A  => A(2),
         B  => B(2),
         C0 => C3_0,
         S0 => S2_0,
         C1 => C3_1,
         S1 => S2_1
      );
   cadd3: entity cadd(caddbeh)-- kreiranje instance sabiraca 3. razreda
      generic map (
         Tk => Tprop
      )
      port map (
         A  => A(3),
         B  => B(3),
         C0 => C4_0,
         S0 => S3_0,
         C1 => C4_1,
         S1 => S3_1
      );
   cadd4: entity cadd(caddbeh)-- kreiranje instance sabiraca 4. razreda
      generic map (
         Tk => Tprop
      )
      port map (
         A  => A(4),
         B  => B(4),
         C0 => C5_0,
         S0 => S4_0,
         C1 => C5_1,
         S1 => S4_1
      );

-- kreiranje instanci dvobitnih dvoulaznih multipleksera koji biraju
-- rezultate uslovnih sabiraca na osnovu vrednosti prenosa u dati razred
   mux2x2_0: entity mux2xn(mux2xnbeh)
      generic map (
         Tk => Tprop,               -- ovde se precizira kasnjenje Tk
         n  => 2                    -- sirina ulaznih signala je 2
      )
      port map (
         A(0) => C1_0,              -- veze signala na portove su logicne
         A(1) => S0_0,              -- na osnovu objasnjenja uloge signala
         B(0) => C1_1,              -- i ovog multipleksera
         B(1) => S0_1,
         s    => Cin,
         Y(0) => C1,
         Y(1) => S(0)
      );
   mux2x2_1: entity mux2xn(mux2xnbeh)
      generic map (
         Tk => Tprop,
         n  => 2
      )
      port map (
         A(0) => C3_0,
```

```
         A(1) => S2_0,
         B(0) => C3_1,
         B(1) => S2_1,
         s    => C2_0,
         Y(0) => C3_X0,
         Y(1) => S2_X0
      );
   mux2x2_2: entity mux2xn(mux2xnbeh)
      generic map (
         Tk => Tprop,
         n  => 2
      )
      port map (
         A(0) => C3_0,
         A(1) => S2_0,
         B(0) => C3_1,
         B(1) => S2_1,
         s    => C2_1,
         Y(0) => C3_X1,
         Y(1) => S2_X1
      );
   mux2x2_3: entity mux2xn(mux2xnbeh)
      generic map (
         Tk => Tprop,
         n  => 2
      )
      port map (
         A(0) => C5_0,
         A(1) => S4_0,
         B(0) => C5_1,
         B(1) => S4_1,
         s    => C4_0,
         Y(0) => C5_X0,
         Y(1) => S4_X0
      );
   mux2x2_4: entity mux2xn(mux2xnbeh)
      generic map (
         Tk => Tprop,
         n  => 2
      )
      port map (
         A(0) => C5_0,
         A(1) => S4_0,
         B(0) => C5_1,
         B(1) => S4_1,
         s    => C4_1,
         Y(0) => C5_X1,
         Y(1) => S4_X1
      );

-- trobitni dvoulazni multiplekser koji bira rezultate 1. i 2. razreda
-- na osnovu vrednosti prenosa u prvi razred - C1.
   mux2x3_0: entity mux2xn(mux2xnbeh)
      generic map (
         Tk => Tprop,               -- ovde se precizira kasnjenje Tk
         n  => 3                    -- sirina ulaznih signala je 2
      )
      port map (
         A(0) => S1_0,              -- veze signala na portove su logicne
         A(1) => C3_X0,             -- na osnovu objasnjenja uloge signala
         A(2) => S2_X0,             -- i ovog multipleksera
```

```
         B(0) => S1_1,
         B(1) => C3_X1,
         B(2) => S2_X1,
         s    => C1,
         Y(0) => S(1),
         Y(1) => C3,
         Y(2) => S(2)
      );
-- trobitni dvoulazni multiplekser koji bira rezultate 3. i 4. razreda
-- na osnovu vrednosti prenosa u treci razred - C3.
   mux2x3_1: entity mux2xn(mux2xnbeh)
      generic map (
         Tk => Tprop,               -- ovde se precizira kasnjenje Tk
         n  => 3                    -- sirina ulaznih signala je 2
      )
      port map (
         A(0) => S3_0,              -- veze signala na portove su logicne
         A(1) => C5_X0,             -- na osnovu objasnjenja uloge signala
         A(2) => S4_X0,             -- i ovog multipleksera
         B(0) => S3_1,
         B(1) => C5_X1,
         B(2) => S4_X1,
         s    => C3,
         Y(0) => S(3),
         Y(1) => C5,
         Y(2) => S(4)
      );

end architecture mk5csadd001struct;
-----
--    Zaglavlje fajla -------------------------------------------------------
--       Naziv cadd - Conditional ADDer
--        Opis Uslovni jednobitni sabirac.Daje zbir i prenos pod pretpostavkom
--             da je ulazni prenos 0 (S0 i C0), odnosno 1 (S1 i C1).
--        Fajl C:\My Documents\etf\ri5vlsi\domaci\mk2002\sabirac\vhdl\sabirac\
--             SRC\cadd.VHD
--      Namena Koristi se kao blok u 5bit sabiracu sa uslovnim sumama.
--    Napomene U realnosti kasnjenje kroz and,or,xor i xnor kola nije isto,ali
--             je ta pretpostavka uvedena u postavci zadatka.
-- Ogranicenja Kasnjenje kroz logicko kolo Tk se mora precizirati pri
--             pravljenju instance ove komponente na visem nivou, u
--             generic mapi.
--      Greske -
--   Korisceni
--      paketi IEEE.std_logic_1164
--       Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--   Korisceni
--   simulator Active VHDL 3.2 build 207
--     Verzija 1.01 (29/01/2002)
--      Izmene Izbacen proces i ubacene konkurentne dodele vrednosti.
-------------------------------------------------------------------------------

library IEEE;
use IEEE.std_logic_1164.all;

entity cadd is
--    Zaglavlje entiteta ----------------------------------------------------
--         Opis Entitet uslovnog jednobitnog sabiraca, sadrzi dva ulazna bita
--              (sabirka) i izlazne, uslovne sume i prenose
--  Ogranicenja Tk - vreme kasnjenja se mora precizirati
-- Pretpostavke Tk je isto za sva logicka kola
```

```
--      Verzija 1.00
--       Izmene -
--        Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--        Datum 29/01/2002
-------------------------------------------------------------------------------
   generic (
      Tk:time                 -- vreme kasnjenja jednog logickog kola
   );
   port (
      A:  in  STD_LOGIC;      -- prvi sabirak (bit)
      B:  in  STD_LOGIC;      -- drugi sabirak (bit)

      C0: out STD_LOGIC;      -- Carry out pod uslovom Cin=0
      S0: out STD_LOGIC;      -- zbir pod uslovom Cin=0
      C1: out STD_LOGIC;      -- Carry out pod uslovom Cin=1
      S1: out STD_LOGIC       -- zbir pod uslovom Cin=1
   );
end cadd;

architecture caddbeh of cadd is
--    Zaglavlje arhitekture ----------------------------------------------------
--         Opis Arhitektura ponasanja uslovnog jednobitnog sabiraca, vrlo
--              pazljivo isprojektovana i testirana.
--  Ogranicenja -
-- Pretpostavke Kasnjenje je isto za and, xor, or i xnor logicka kola.
--      Verzija 1.01
--       Izmene Izbacen proces i ubacene konkurentne dodele vrednosti.
--        Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--        Datum 29/01/2002
-------------------------------------------------------------------------------
begin
   C0 <= A and  B  after Tk;  --  Ovo su konkurentne dodele vrednosti signalima
   S0 <= A xor  B  after Tk;  --  aktiviraju se automatski pri promeni bilo kog
   C1 <= A or   B  after Tk;  --  signala sa desne strane ovih izraza, u ovom
   S1 <= A xnor B  after Tk;  --  slucaju A ili B.
end architecture caddbeh;
-----
--    Zaglavlje fajla ----------------------------------------------------------
--       Naziv mux2xn - Dvoulazni n-bitni multiplekser (2xn ->n)
--        Opis Propusta jedan od dva ulazna vektora od n bita na izlaz,
--             zavisno od vrednosti ulaznog signala s (select).
--        Fajl C:\My Documents\etf\ri5vlsi\domaci\mk2002\sabirac\vhdl\sabirac\
--             SRC\mux2xn.VHD
--      Namena Koristi se kao blok u 5bit sabiracu sa uslovnim sumama.
--    Napomene U realnosti kasnjenje kroz and,or,xor i xnor kola nije isto,ali
--              je ta pretpostavka uvedena u postavci zadatka.
--             U slucaju dovodjenja 'U','X','W','Z' i '-' na s port, na izlaz
--              prosledjuje 2. ulaz - B.
-- Ogranicenja Kasnjenje kroz logicko kolo - Tk i sirina vektora - n se moraju
--             precizirati pri pravljenju instance ove komponente na visem
--             nivou, u generic mapi.
--      Greske -
--   Korisceni
--      paketi IEEE.std_logic_1164
--       Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--   Korisceni
--   simulator Active VHDL 3.2 build 207
--     Verzija 1.01 (29/01/2002)
--      Izmene Izmene u arhitekturi
-------------------------------------------------------------------------------
```

```
library IEEE;
use IEEE.std_logic_1164.all;

entity mux2xn is
--    Zaglavlje entiteta -------------------------------------------------------
--         Opis Entitet dvoulaznog n-bitnog multipleksera, sadrzi dva ulazna,
--              n-bitna porta, port selekcije, i izlazni n-bitni port.
--  Ogranicenja Tk-vreme kasnjenja, i n-sirina ulaza se moraju precizirati.
-- Pretpostavke Tk je isto za sva logicka kola.
--      Verzija 1.00
--       Izmene -
--        Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--        Datum 29/01/2002
--------------------------------------------------------------------------------
   generic (
      n:positive;             -- sirina ulaznih i izlaznog signala, u bitima
      Tk:time                 -- vreme kasnjenja jednog logickog kola
   );
   port (
      A: in  STD_LOGIC_VECTOR (n-1 downto 0); -- 0. ulazni n-bitni signal
      B: in  STD_LOGIC_VECTOR (n-1 downto 0); -- 1. ulazni n-bitni signal
      s: in  STD_LOGIC;                       -- signal selekcije ulaza

      Y: out STD_LOGIC_VECTOR (n-1 downto 0)  -- n-bitni izlazni signal
    );
end mux2xn;

architecture mux2xnbeh of mux2xn is
--    Zaglavlje arhitekture ----------------------------------------------------
--         Opis Arhitektura ponasanja dvoulaznog n-bitnog multipleksera, vrlo
--              pazljivo projektovana i testirana.
--  Ogranicenja -
-- Pretpostavke Kasnjenje je isto za and, not, or i nor logicka kola.
--      Verzija 1.01
--       Izmene Izbacen proces i ubacene uslovne dodele vrednosti.
--        Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--        Datum 29/01/2002
--------------------------------------------------------------------------------
   -- broj nivoa logickih kola od ulaza do izlaza, za odredjivanje kasnjenja
   constant l: positive :=3;
begin
   Y <= A after 3*Tk when (s='0' or s='L') else -- ovo je uslovna dodela vredn.
        B after 3*Tk;         -- aktivira se automatski pri promeni A, B ili s
end architecture mux2xnbeh;
```

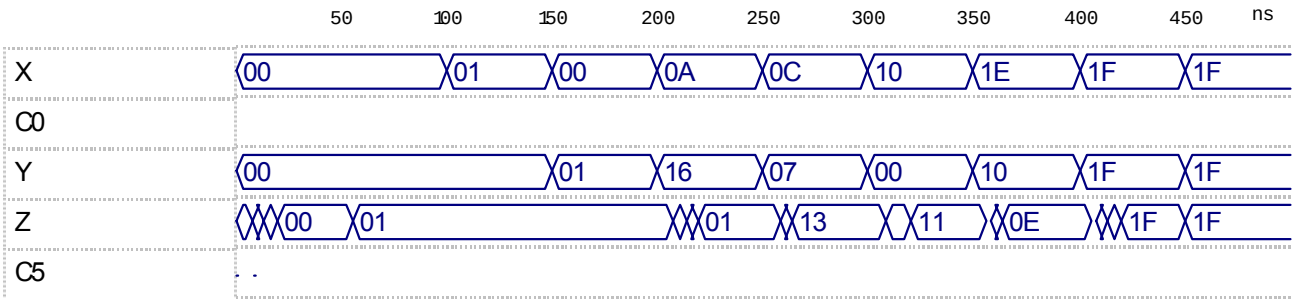
2.2 Simulacija sabirača pomoću TestBench-a u programu ActiveVHDL

2.2.1 Izvorni kod za TestBench

```
--    Zaglavlje fajla -------------------------------------------------------
--       Naziv adderTestBench - Test za sabirac
--        Opis Proverava izlaze sabiraca za granicne slucajeve ulaza.
--        Fajl C:\My Documents\etf\ri5vlsi\domaci\mk2002\sabirac\vhdl\sabirac\
--             SRC\adderTestBench.VHD
--      Namena Za testiranje 5csadd sabiraca.
--    Napomene -
-- Ogranicenja -
--      Greske -
--   Korisceni
--      paketi IEEE.std_logic_1164
--       Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--   Korisceni
--   simulator Active VHDL 3.2 build 207
--     Verzija 1.00 (31/01/2002)
--      Izmene
--------------------------------------------------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;

entity adderTestBench is
--    Zaglavlje entiteta ----------------------------------------------------
--         Opis Kao i svi entiteti za TestBench i ovaj nema portove
--  Ogranicenja -
-- Pretpostavke -
--      Verzija 1.00
--       Izmene -
--        Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--        Datum 31/01/2002
--------------------------------------------------------------------------------
end entity adderTestBench;

architecture test1 of adderTestBench is
--    Zaglavlje arhitekture -------------------------------------------------
--         Opis Na ulaze sabiraca dovodima iste test vektore koji su korisceni
--              i pri testiranju u alteri.
--  Ogranicenja
-- Pretpostavke
--      Verzija 1.00
--       Izmene -
--        Autor Milos Kovacevic, pirate@bitsyu.net, http://pirate.jump.to
--        Datum 31/01/2002
--------------------------------------------------------------------------------
   signal X : std_logic_vector (4 downto 0);    -- prvi test sabirak
   signal C0: std_logic;                        -- ulazni test prenos
   signal Y : std_logic_vector (4 downto 0);    -- drugi test sabirak

   signal Z : std_logic_vector (4 downto 0);    -- test zbir
   signal C5: std_logic;                        -- izlazni test prenos

begin
   -- kreiranje instance sabiraca koji se testira
   adder: entity \5csadd\ (mk5csadd001struct)
      port map (
         A   => X,
         Cin => C0,
         B   => Y,
```

```
         S   => Z,
         C5  => C5
      );
   granicniTest: process is
      begin
      -- sada sledi primenjivanje istih test vektora kao i u alteri, i
      -- provera dobijenih rezultata
         X <= "00000";
         Y <= "00000";
         C0 <= '0';
         wait for 50ns;
         assert (Z="00000") and (C5='0') -- ocekivani rezultati
         -- poruka za ispis u slucaju greske:
            report "Greska pri sabiranju 0 i 0, uz C0=0";
         C0 <= '1';
         wait for 50ns;
         assert (Z="00001") and (C5='0')
            report "Greska pri sabiranju 0 i 0, uz C0=1";
         X  <= "00001";
         C0 <= '0';
         wait for 50ns;
         assert (Z="00001") and (C5='0')
            report "Greska pri sabiranju 1 i 0, uz C0=0";
         X <= "00000";
         Y <= "00001";
         wait for 50ns;
         assert (Z="00001") and (C5='0')
            report "Greska pri sabiranju 0 i 1, uz C0=0";
         X <= "01010";
         Y <= "10110";
         C0<= '1';
         wait for 50ns;
         assert (Z="00001") and (C5='1')
            report "Greska pri sabiranju 10 i 22, uz C0=1";
         X <= "01100";
         Y <= "00111";
         C0<= '0';
         wait for 50ns;
         assert (Z="10011") and (C5='0')
            report "Greska pri sabiranju 12 i 7, uz C0=0";
         X <= "10000";
         Y <= "00000";
         C0<= '1';
         wait for 50ns;
         assert (Z="10001") and (C5='0')
            report "Greska pri sabiranju 16 i 0, uz C0=1";
         X <= "11110";
         Y <= "10000";
         C0<= '0';
         wait for 50ns;
         assert (Z="01110") and (C5='1')
            report "Greska pri sabiranju 30 i 16, uz C0=0";
         X <= "11111";
         Y <= "11111";
         C0<= '1';
         wait for 50ns;
         assert (Z="11111") and (C5='1')
            report "Greska pri sabiranju 31 i 31, uz C0=1";
         wait;
      end process granicniTest;
end architecture test1;
```

2.2.2 Rezultati simulacije



NAPOMENA: Kašnjenje od ulaza sabirača do njegovog izlaza po proračunu treba da iznosi **10*Tp** (i to kroz cadd Tp, a kroz multipleksere 3*Tp. Pošto od ulaza do izlaza imamo 3 nivoa multipleksera - u prvom su dvobitni a u drugom i trecem trobitni sledi da je ukupno kašnjenje 10*Tp) gde je Tp vreme propagacije, odnosno kašnjenja kroz jedno logičko kolo koje je po postavci zadatka isto za sva standardna kola. Pošto je za vrednost Tp uzeto 2ns ukupno kašnjenje treba da iznosi **20ns** što je i potvrđeno simulacijom.

Međutim iz tablice kašnjenja dobijene u programu AlteraMAX+plus vidimo da je maksimalno kašnjenje čak 39ns! Postoje dva razloga za ove razlike u kašnjenjima. Prvo u programu AlteraMAX+plus izabrana je nešto starija serija čipova MAX5000 (za seriju MAX7000 dobijeno je kašnjenje od 20.3ns) kod koje je i vreme propagacije kroz jedan gejt verovatno lošije. Takođe uvedena preetpostavka (pri računskom određivanju) da je kašnjenje kroz sva kola (recimo invertor i nili kolo) jednako ne odgovara realnosti.

Drugi razlog je taj što Alterin kompajler, ako se ne koristi u wysiwyg modu rada, vrši određene minimizacije i zamene određenih kola sa više drugih logičkih kola (na primer exnili kolo korišćeno u bloku cadd se sigurno ne ostavlja kao tako, nego se verovatno menja sa nili, ni kolima i invertorima, ili sl.)