

## ЛОГИЧКО ПРОЈЕКТОВАЊЕ – РАЧУНСКЕ ВЕЖБЕ АКРЕДИТАЦИЈА 2019

### Консултације

- Заказивање консултација дан унапред на е-mail
  - [milos.radmanovic@elfak.ni.ac.rs](mailto:milos.radmanovic@elfak.ni.ac.rs)
  - [nenad.petrovic@elfak.ni.ac.rs](mailto:nenad.petrovic@elfak.ni.ac.rs)

### Веб

- Сајт предмета: <https://cs.elfak.ni.ac.rs/nastava/>
  - СИП налог
  - Информације о предмету
  - План и програм наставе
  - Литература
  - Приручник за рачунске вежбе
  - Приручник за лабораторијске вежбе
  - Линкови

### Литература

- B. LaMeres, Introduction to Logic Circuits & Logic Design with VHDL, Springer 2017.

### Информације о предмету

- Нотације задатака на рачунским вежбама прате задатке у уџбенику “ B. LaMeres, Introduction to Logic Circuits & Logic Design with VHDL”.
- Оцењивање:
  - 20 поена (20%) – лабораторијске вежбе.
  - 40 поена (40%) – писмени део испита.
  - 40 поена (40%) – усмени део испита.
  - (55-64):6, (65-74):7, (75-84):8 (85-94):9, (95-100):10.
- Писмени део испита:
  - 2 задатка (сваки задатак се оцењује са 20 поена)
  - Резултати испита на сајту предмета
- Усмени део испита:
  - 4 теоријска питања (свако питање се оцењује са 10 поена)
- Лабораторијске вежбе:
  - 5 термина за лаб. вежбе.
  - Свака вежба се бодује са 4 поена.
  - Лаб. вежбе почињу 2. недеље новембра (видети план и програм).
  - Лаб. вежбе се одржавају сваке друге недеље.
  - Последње недеље семестра постоји термин за надокнаду лаб. вежби.
  - У сваком термину се преко Веб сајта предмета добија по 1 задатак за рад (потребно је логовање на сајт СИП налогом).
  - Потребна је претходна припрема вежбе из уџбеника (сајт предмета).
  - Задаци се раде у ModelSim-Intel FPGA Edition
  - Инсталација ModelSim-Intel FPGA Edition (1GB)
  - [http://download.altera.com/akdlm/software/acdsinst/18.1std/625/ib\\_installers/ModelSimSetup-18.1.0.625-windows.exe](http://download.altera.com/akdlm/software/acdsinst/18.1std/625/ib_installers/ModelSimSetup-18.1.0.625-windows.exe)

Молимо вас да уочене грешке и примедбе на садржај предавања и рачунских вежби изнесете предметном професору или асистенту путем email-a,

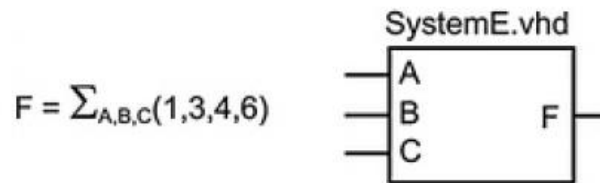
[milos.radmanovic@elfak.ni.ac.rs](mailto:milos.radmanovic@elfak.ni.ac.rs)

[nenad.petrovic@elfak.ni.ac.rs](mailto:nenad.petrovic@elfak.ni.ac.rs)

# 1. Опис комбинационих мрежа у VHDL-у

## Задатак 5.5.1

Написати VHDL модел који описује дигитални систем задат 3-улазном листом минтерма приказаног на слици 5.11. Користити конкурентну доделу сигнала и логичке операторе. Декларисати ентитет тако да описује блок дијаграм са слике. Користити тип bit за портове.



Слика. 5.11 System E функционалност

## Решење задатка 5.5.1

A	B	C	F	$F=A'B'C+A'BC+AB'C'+ABC'$
0	0	0	0	
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	1	
1	0	1	0	
1	1	0	1	
1	1	1	0	

```
entity SystemE is
    port (A,B,C : in bit;
          F: out bit);
end entity;

architecture SystemE_arch of SystemE is
    signal An, Bn, Cn : bit;
    signal m1, m3, m4, m6 : bit;
begin
    An <= not A;
    Bn <= not B;
    Cn <= not C;

    m1 <= An and Bn and C;
    m3 <= An and B and C;
    m4 <= A and Bn and Cn;
    m6 <= A and B and Cn;

    F <= m1 or m3 or m4 or m6;
end architecture;
```

## Тестирање задатка 5.5.1 (лабораторијске вежбе)

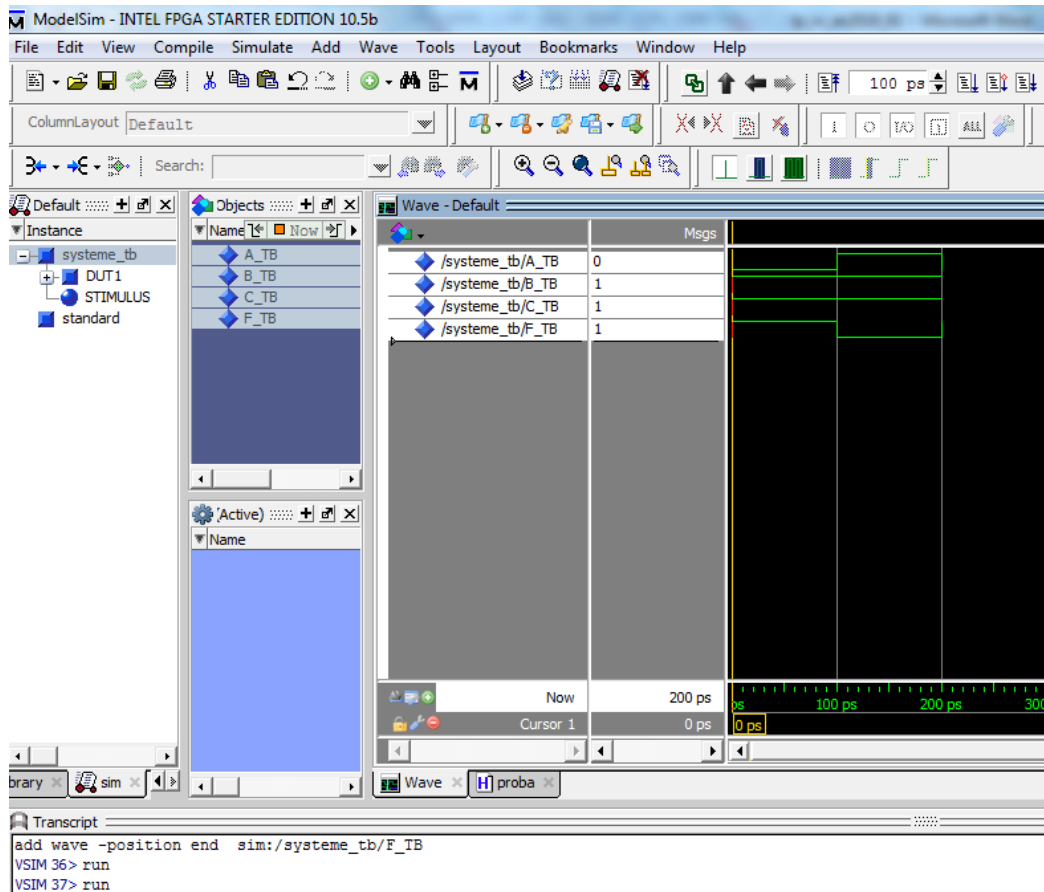
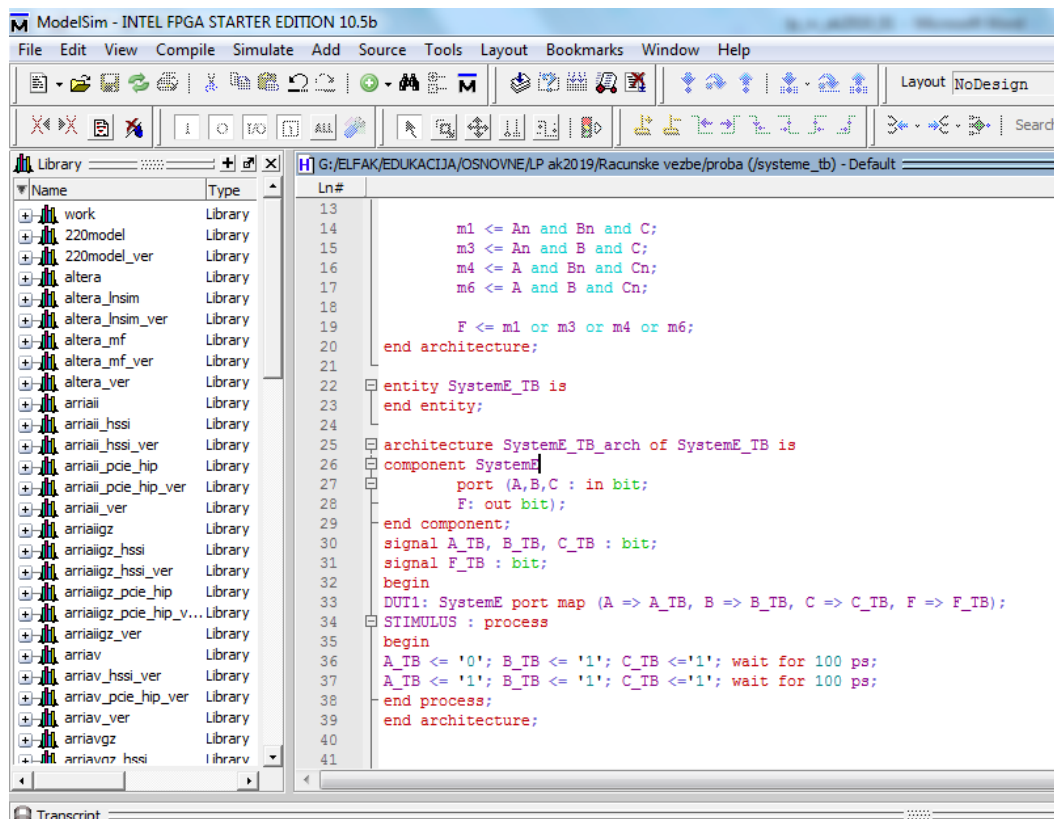
```
entity SystemE_TB is
end entity;

architecture SystemE_TB_arch of SystemE_TB is
    component SystemE
        port (A,B,C : in bit;
              F: out bit);
    end component;
    signal A_TB, B_TB, C_TB : bit;
    signal F_TB : bit;
begin
    DUT1: SystemE port map (A => A_TB, B => B_TB, C => C_TB, F => F_TB);
    STIMULUS : process
begin
```

```

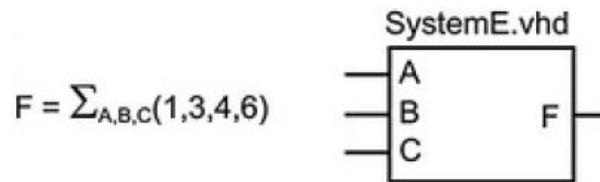
A_TB <= '0'; B_TB <= '1'; C_TB <='1'; wait for 100 ps;
A_TB <= '1'; B_TB <= '1'; C_TB <='1'; wait for 100 ps;
end process;
end architecture;

```



### Задатак 5.5.2

Написати VHDL модел који описује дигитални систем задат 3-улазном листом минтерма приказаног на слици 5.11. Користити условну доделу сигнала. Декларисати ентитет тако да описује блок дијаграм са слике. Користити тип `bit` за портове.



Слика. 5.11 System E функционалност

### Решење задатка 5.5.2

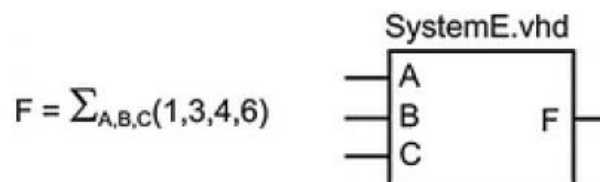
A	B	C	F	$F=A'B'C+A'BC+AB'C'+ABC'$
0	0	0	0	
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	1	
1	0	1	0	
1	1	0	1	
1	1	1	0	

```
entity SystemE is
    port (A,B,C : in bit;
          F: out bit);
end entity;

architecture SystemE_arch of SystemE is
begin
    F <= '1' when (A='0' and B='0' and C='1') else
              when (A='0' and B='1' and C='1') else
              when (A='1' and B='0' and C='0') else
              when (A='1' and B='1' and C='0') else
              '0';
end architecture;
```

### Задатак 5.5.3

Написати VHDL модел који описује дигитални систем задат 3-улазном листом минтерма приказаног на слици 5.11. Користити селецијску доделу сигнала. Декларисати ентитет тако да описује блок дијаграм са слике. Користити тип `bit` за портове.



Слика. 5.11 System E функционалност

### Решење задатка 5.5.3

A	B	C	F	$F=A'B'C+A'BC+AB'C'+ABC'$
0	0	0	0	
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	1	
1	0	1	0	
1	1	0	1	
1	1	1	0	

```
entity SystemE is
    port (A,B,C : in bit;
          F: out bit);
```

```

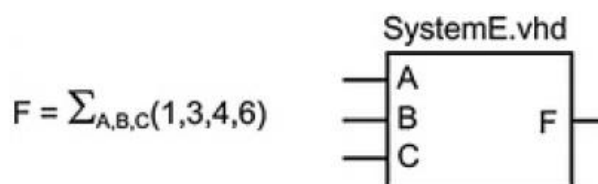
end entity;

architecture SystemE_arch of SystemE is
signal ABC <= bit_vector(2 downto 0);
begin
    ABC <= A & B & C;
    with (ABC) select
        F <= '1' when '001',
            '1' when '011',
            '1' when '100',
            '1' when '110',
            '0' when others;
end architecture;

```

### Задатак 5.6.1

Написати VHDL модел који описује дигитални систем задат 3-улазном листом минтерма приказаног на слици 5.11. Користити структурно пројектовање и базична кола. Потребно је креирати сва потребна базична кола (нпр. INV1, AND2, OR4), која се инстанцирају у архитектури на вишем нивоу. Кола на нижем нивоу треба имплементирати коришћењем конкурентне доделе сигнала и логичких оператора (нпр.  $F \leq \text{not } A$ ). Декларисати ентитет тако да описује блок дијаграм са слике. Користити тип bit за портове.



Слика. 5.11 System E функционалност

### Решење задатка 5.6.1

A	B	C	F	$F = A'B'C + A'BC + AB'C' + ABC'$
0	0	0	0	
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	1	
1	0	1	0	
1	1	0	1	
1	1	1	0	

```

entity INV1 is
port (A : in bit;
      F : out bit);
end entity;

architecture INV1_arch of INV1 is
begin
    F <= not A;
end architecture;

entity AND3 is
port (A,B,C : in bit;
      F      : out bit);
end entity;

architecture AND3_arch of AND3 is
begin
    F <= A and B and C;
end architecture;

entity OR4 is
port (A,B,C,D : in bit;
      F       : out bit);

```

```

end entity;

architecture OR4_arch of OR4 is
begin
    F <= A or B or C or D;
end architecture;

architecture SystemE_arch of SystemE is
    signal An, Bn, Cn : bit;
    signal m1, m3, m4, m6 : bit;

    component INV1 is
        port (A : in bit;
              F : out bit);
    end component;

    component AND3 is
        port (A,B,C : in bit;
              F      : out bit);
    end component;

    component OR4 is
        port (A,B,C,D : in bit;
              F        : out bit);
    end component;
begin
    U1 : INV1 port map (A=>A, F=>An);
    U2 : INV1 port map (A=>B, F=>Bn);
    U3 : INV1 port map (A=>C, F=>Cn);

    U4 : AND3 port map (A=>An, B=>Bn, C=>C, F=>m1);
    U5 : AND3 port map (A=>An, B=>B, C=>C, F=>m3);
    U6 : AND3 port map (A=>A, B=>Bn, C=>Cn, F=>m4);
    U7 : AND3 port map (A=>A, B=>B, C=>Cn, F=>m6);

    U8 : OR4 port map (A=>m1, B=>m3, C=>m4, D=>m6, F=>F);
end architecture;

```