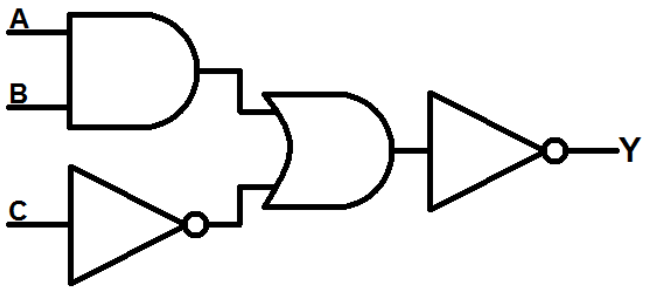
APEKS, SEK, VHDL, Pripremni materijal, 2023

A) Kombinaciona kola (1-30)

1. Za kolo prikazano na slici napisati VHDL kod. Priložiti kod i simulacione dijagrame.



Slika 1.

2. Za kolo punog-sabirača :
   a) Napisati VHDL kod.
   b) Kreirati simbol.
   c) Simulacijom verifikovati ispravnost dizajna punog sabirača. Parametre *End Time* i *Grid Size* postaviti na vrijednosti 2μs i 200ns, respektivno.
   d) Sabirac je aktivan pri ENABLE=1.

3. a) Projektovatki kolo 8_bitnog sabirača koristeći komponentu 1 bitnog punog sabirača. Izvršiti simulaciju i napraviti simbol.
   b) Koristeći 8_bitni simbol projektovati u grafičkom editoru 64_bitni sabirač i izvršiti njegovu simulaciju.

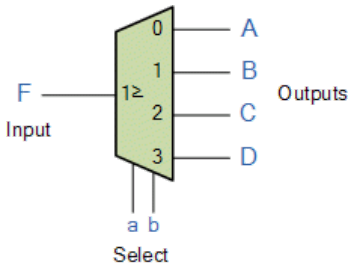
4. a) Projektovati kolo u VHDLu 2-to-1 1-bitnog MUXa.

IO Pins	Description
a, b	Data Inputs
s[0]	MUX selector
o	Data Output +

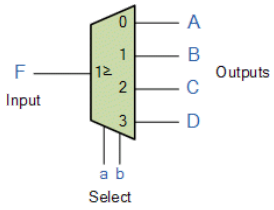
   b) Koristeći kreirani symbol Mux 2-1, projektovati kolo multipleksora 4-1 i simulirati njegov rad.

5. Dizajnirajte sistem glasanja od 16 glasača gdje 0 znači „Ne“, a 1 „Da“. Yes[0..3\ prikazuje broj glasova „da“, a No[0..3\ broj glasova „ne“.

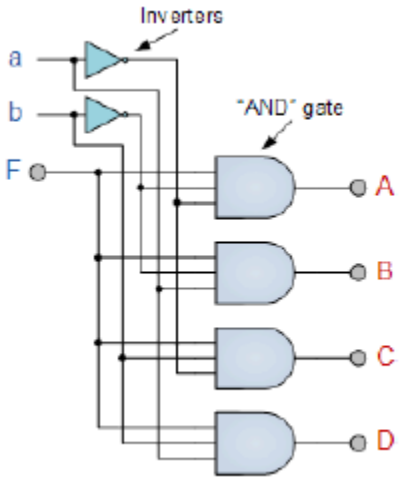
6. Na slici je kolo demultipleksora. Napisati kod i verifikovati ispravnost dizajna (simulacijom). Koristiti formulu $F = abA + abB + abC + abD$.



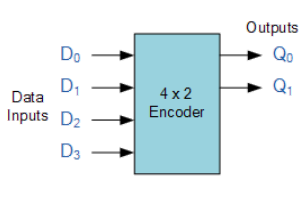
7. Na slici je kolo demultipleksora. Napisati kod i verifikovati ispravnost dizajna. Koristiti CASE tipa naredbu napisati kod i verifikovati ispravnost



8. Na slici je kolo demultipleksora. Koristeci graficki editor i simulator verifikovati dizajn.

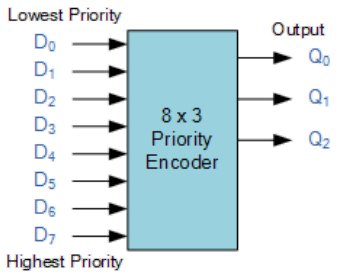


9. Na slici je je kolo 4-2 binarnog encodera. Koristeci VHDL editor i simulator verifikovati dizajn.



Inputs				Outputs	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

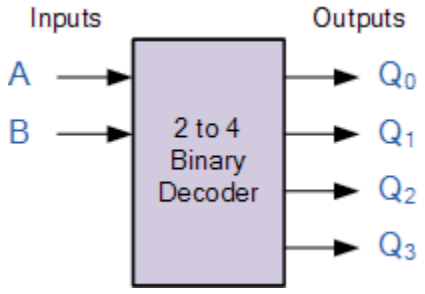
10. Na slici je je kolo 8-3 binarnog encodera. Koristeci VHDL editor i simulator verifikovati dizajn.



Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	x	0	0	1
0	0	0	0	0	1	x	x	0	1	0
0	0	0	0	1	x	x	x	0	1	1
0	0	0	1	x	x	x	x	1	0	0
0	0	1	x	x	x	x	x	1	0	1
0	1	x	x	x	x	x	x	1	1	0
1	x	x	x	x	x	x	x	1	1	1

X = dont care

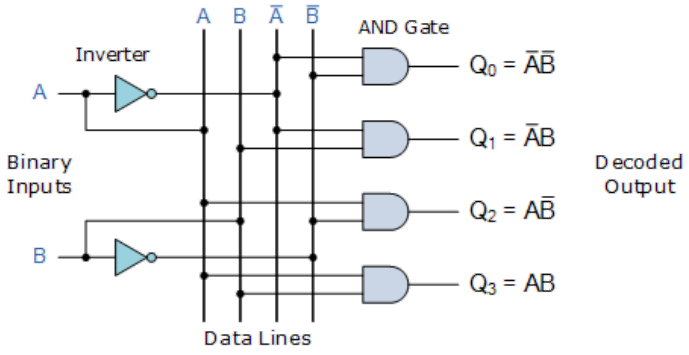
11. Na slici je je kolo 2-4 binarnog decodera. Koristeci VHDL editor i simulator verifikovati dizajn.



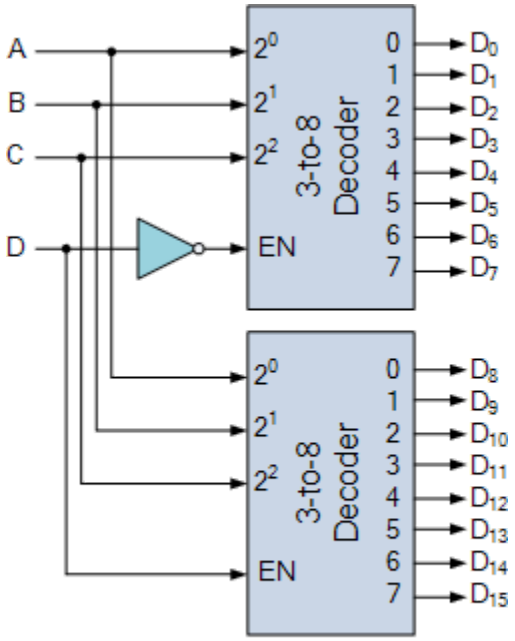
Truth Table

A	B	Q_0	Q_1	Q_2	Q_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

12. Na slici je je kolo 2-4 binarnog decodera. Koristeci graficki editor i simulator verifikovati dizajn.

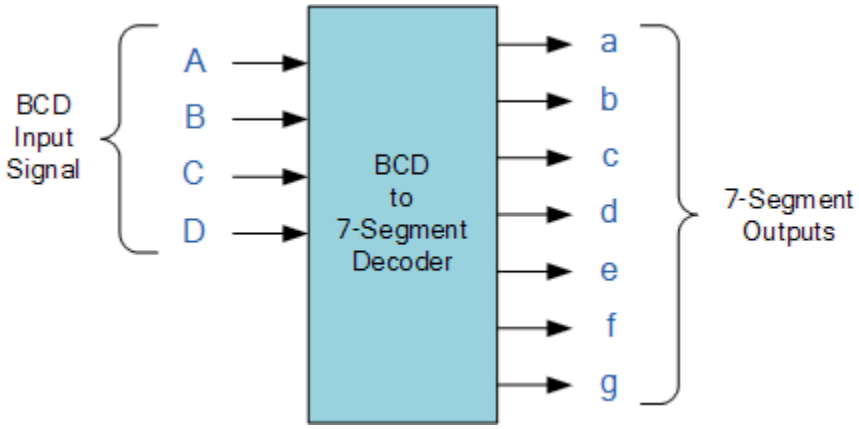


13. Na slici je 4-16 binarni decoder. Projektovati 3-8 decoder, zatim ga sacuvati kao symbol, pa u grafickom editoru i simulatoru verifikovati dizajn 4-16.

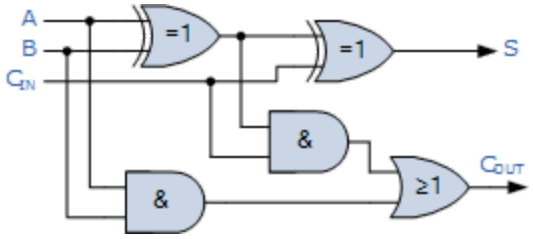


4-to-16 Line Decoder Implemented with two 3-to-8 Decoders

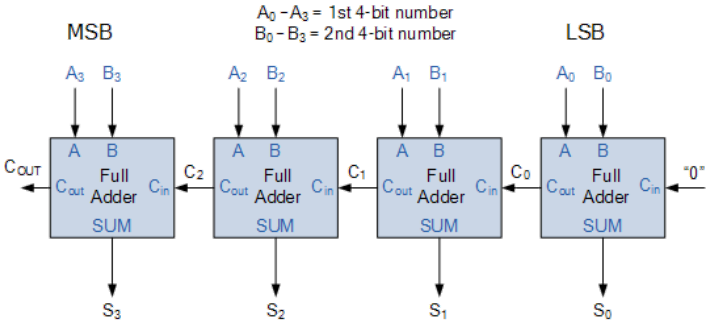
14. Na slici je BCD-7 segment decoder. Koristeci VHDL editor i simulator verifikovati dizajn.



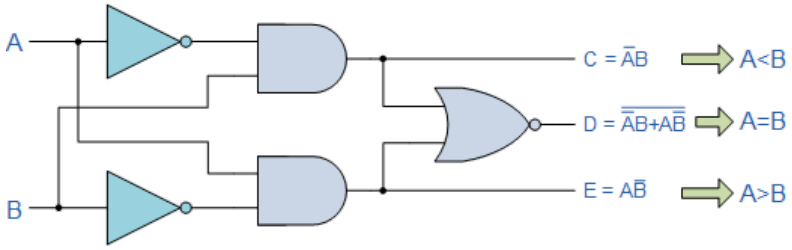
15. Na slici je kolo punog sabiraca. Koristeci VHDL editor i simulator verifikovati dizajn



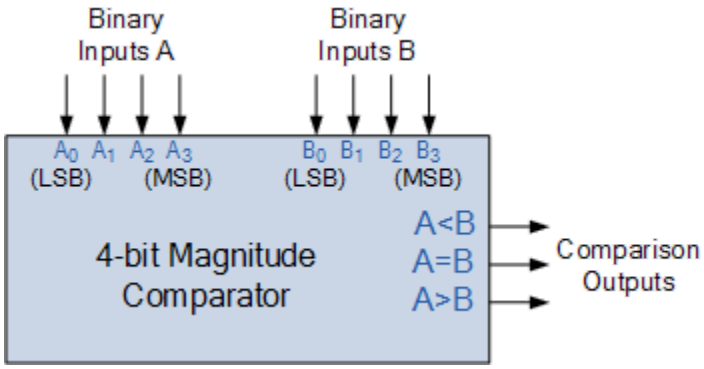
16. Na slici je kolo 4-bitnog punog sabiraca. Koristeci VHDL editor i simulator verifikovati dizajn



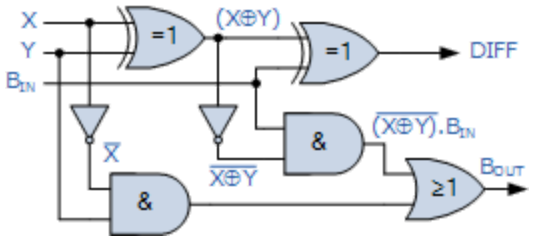
17. Na slici je kolo 1-bitnog komparatora. Koristeci VHDL editor i simulator verifikovati dizajn



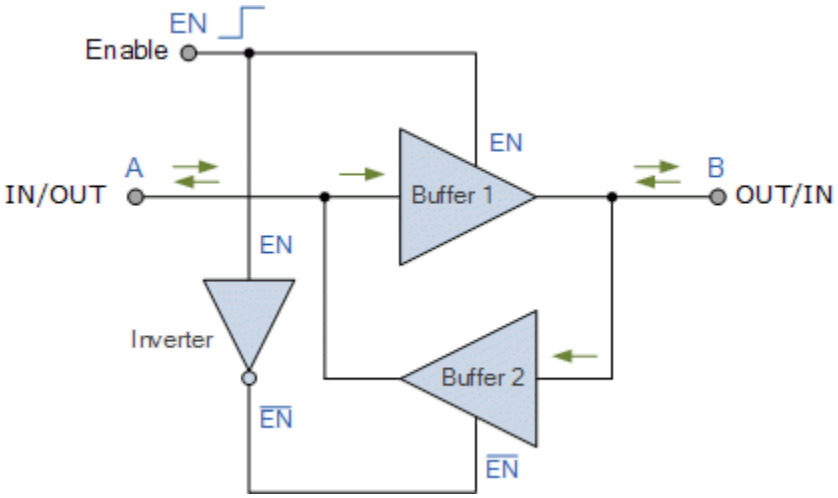
18. Na slici je kolo 4-bitnog komparatora. Koristeci VHDL editor i simulator verifikovati dizajn

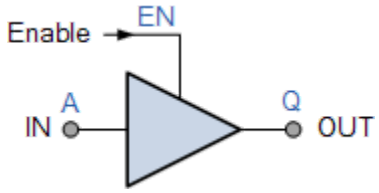


19. Na slici je kolo punog 1-bitnog mnozaca. Koristeci VHDL editor i simulator verifikovati dizajn



20. Na slici je slika bidirekcionog pina kojim se upravlja sa EN (EN=1, OUT=IN; EN=0, IN=OUT). Glavna komponenta je 3-state buffer, cija je tabela stanja data ispod znaka. Koristeci VHDL editor i simulator verifikovati dizajn. Pomoc: https://vlsicoding.blogspot.com/2014/07/vhdl-code-for-bidirectional-bus.html

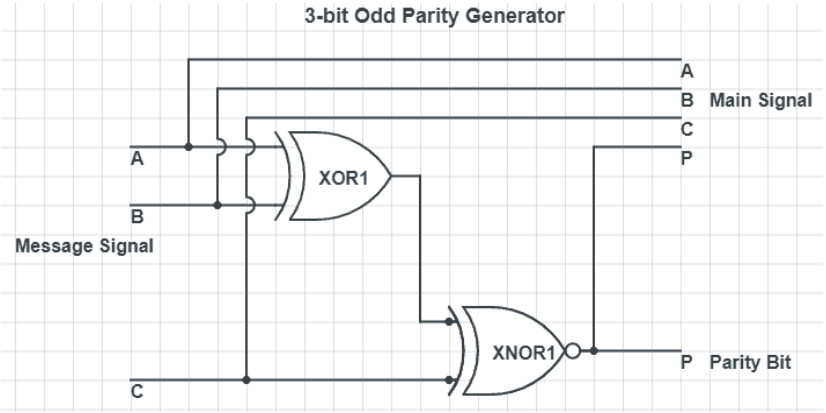




Enable	IN	OUT
0	0	Hi-Z

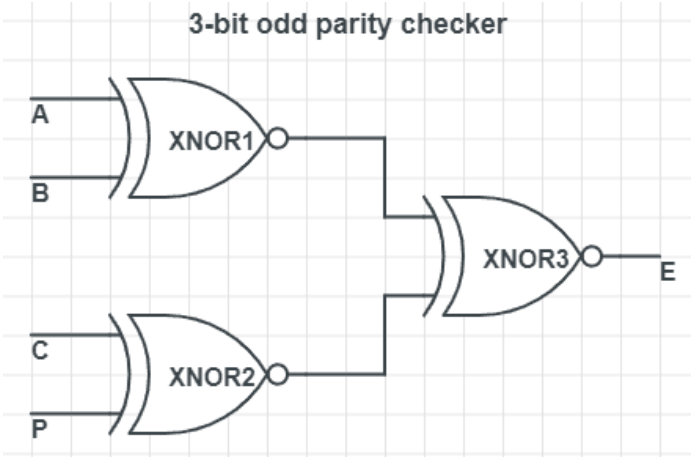
0	1	Hi-Z
1	0	0
1	1	1

21. Na slici je dato kolo za generisanje bita parnosti, tj. ako je paran broj 1-ca na ulazima A,B,C kolo daje na svom izlazu P=1, u suprotnom P=0. Koristeci VHDL editor i simulator verifikovati dizajn.



A	B	C	P
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	X

22. Na slici je dato kolo za provjeru parnosti, tj. ako je paran broj 11 na ulazima A,B,C kolo daje na svom izlazu P=1, u suprotnom P=0. Koristeci VHDL editor i simulator verifikovati dizajn.



A B C P E

0 0 0 0 1

0 0 0 1 0

0 0 1 0 0

0 0 1 1 1

0 1 0 0 0

0 1 0 1 1

0 1 1 0 1

0 1 1 1 0

1 0 0 0 0

1 0 0 1 1

1 0 1 0 1

1 0 1 1 0

1 1 0 0 1

1 1 0 1 0

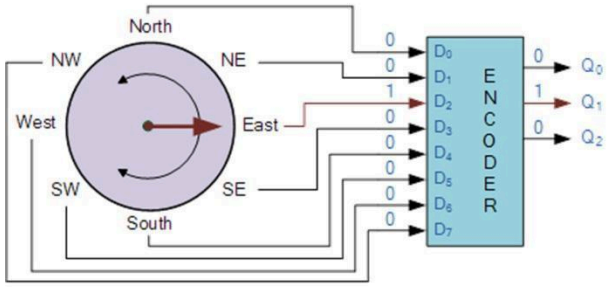
1 1 1 0 0

1 1 1 1 1

23. Projektovati kolo koje će realizovati zadatu fukciju F .Zadatak uraditi na dva načina : a)unošenjem odgovarajućeg koda , b)metodom blok dijagrama.Zadatke pod a i b realizovati kao dva odvojena projekta,zadati iste simulacione dijagrame i uvjeriti se da se dobijaju isti rezultati simulacije.

F=(A+B)*not(C)+(B*not(D)+C*not(A))*D

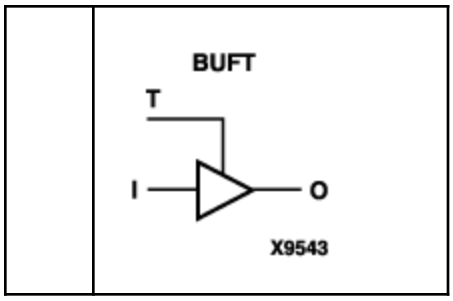
24. Na slici je dat primjer enkodera koji rotacionu poziciju kompasa pretvara u digitalni kod, tj. **8 pozicija** u **3-bitni izlaz Q0,Q1,Q2** prema donjoj tabeli. Projektovati VHDL kod enkodera za date svrhe. Priložiti kod i simulacione dijagram sa istaknutim kursorom na poziciji "SE".



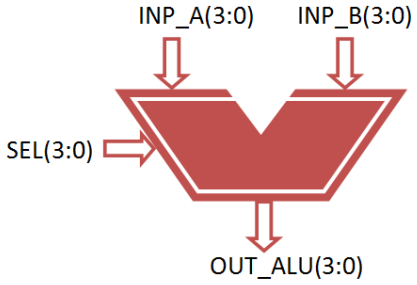
Compass Direction	Binary Output Q0	Q1	Q2
North	0	0	0
North-East	0	0	1
East	0	1	0
South-East	0	1	1
South	1	0	0
South-West	1	0	1
West	1	1	0
North-West	1	1	1

25. a) Projektovati kolo multipleksera 2/1, kreirati odgovarajući paket i simbol. Priložiti kodove. Simulaciju nije potrebno odraditi. b) b) Na osnovu donje slike projektovati multiplekser 8/1 primjenom blok dijagram fajla metode. Priložiti blok dijagram simulacioni dijagram.

IO Pins	Description
I	Data Input
T	Output Enable (active Low)
O	Data Output



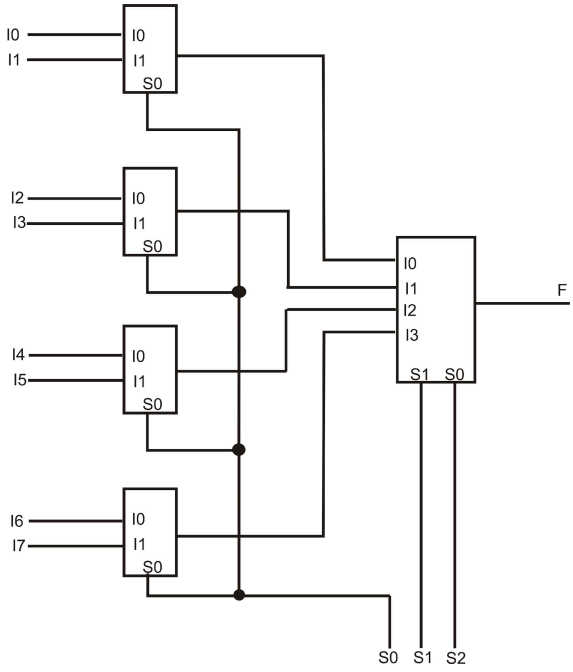
26. Projektovati i simulirati 4-bitnu ALU, 4-bit Arithmetic Logic Unit, sa logikom koja je data dolje.



Selection Input			Operation Performed
0	0	0	A + B
0	0	1	A - B
0	1	0	A - 1
0	1	1	A + 1
1	0	0	A and B
1	0	1	A or B
1	1	0	not A
1	1	1	A xor B

27. a) Projektovati kolo multipleksera 2/1, kreirati odgovarajući paket i simbol. Priložiti kodove. Simulaciju nije potrebno odraditi.

b) Na osnovu donje slike projektovati multiplekser 8/1 primjenom blok dijagram fajla metode. Priložiti blok dijagram simulacioni dijagram.



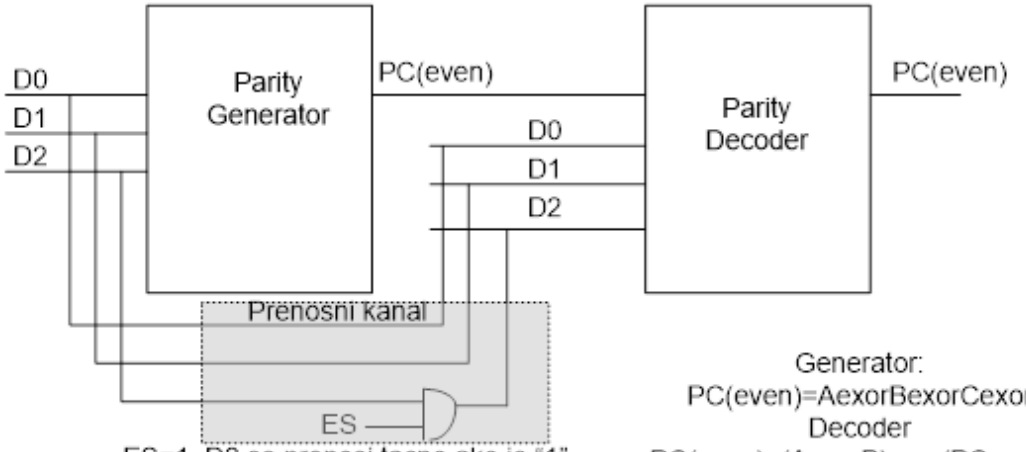
28. Na slici je dat digitalni prenosni sistem koji se sastoji od “parity generatora” , prenosnog kanala i “parity decodera”.

a) Napraviti VHDL cod i simulirati rad parity generatora (neparnog, PC=1 u slucaju neparnog broja “1” u 3 bitnom vektoru [DO…D2].

b) Napraviti VHDL cod i simulirati rad pariry decodera (neparnog, PC=1 u slucaju neparnog broja “1” u vektoru (4bitnog) [DO…D2..PC].

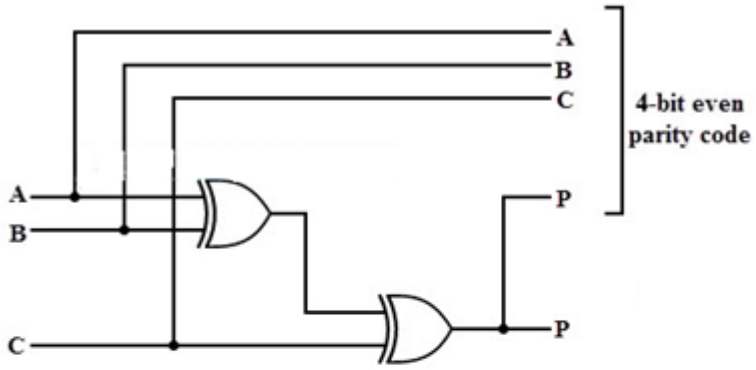
c) Koristeci simbole parity generator i parity decoder simulirati prenos podataka sa detekcijom greske, koristiti ES ulaz za generisanje greske kod bita D2, npr uzeti [D0,D1,D2]=[1,1,1] i ES=0 za simulaciju greske u prenosu i [D0,D1,D2]=[1,1,1] i ES=1 bez greske. Moze se koristiti block diagram pristup (graficki) za konstrukciju i simulaciju citavog sistema.

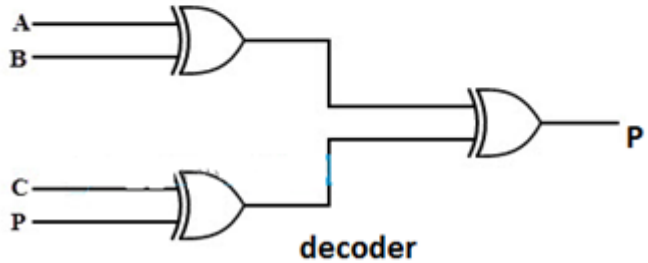
Date su pomocne seme (dolje) parity generatora i parity decodera.



ES=1, D2 se prenosi tacno ako je "1"
ES=0, D2 se prenosi tacno ako je "1"
ES nema uticaja ako je D2=0

Generator:
PC(even)=AexorBexorCexor
Decoder
PC(even)=(AexorB)exor(PCexorD)





decoder

29. Realizovati kolo parity generatora promenljive duzine N pomoću funkcije „parity“ i provjeriti (simulacijom) njegov rad. Koristiti cinjenicu PC(even)=A exor Bexor Cexor……. Koja se moze implementirati u for petlji. Moze se koristiti atribut ‘RANGE.

    function parity (a: std_logic_vector) return std_logic is

    .....

    end parity;

30. Projektovati VHDL kod MAX_MIN koristeći funkciju:

    function MAX_MIN(A, B,C: in integer; D: in bit,) return integer is

    …….

    end MAX_MIN;

    Ako je D=0 funkcija vraca maximalnu vrijednost (najveci od) ulaznih parametara A,B,C, ako je D=1 funkcija vraca minimalnu vrijednost (najmanji od) ulaznih parametara .

    b) Realizovati isti kod koristeći proceduru.

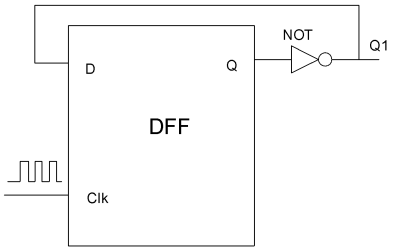
    Priložiti kod i simulacione dijagrame.

B) SEKVENCIJALNA KOLA (1-23)

1. a) Projektovati u VHDL-u D Flip Flop sa ENABLE ulazom i klokom na silaznoj ivici

    b) Formirati simbol.

    c) Koristeci simbol DFF i invertor u grafičkom editoru napravite djelitelj frekvencije kloka sa 4.



2. Projektovati RS flip flop sa clear i set ulazima. Uočiti detla delay. Priložiti kod i simulacione dijagrame.

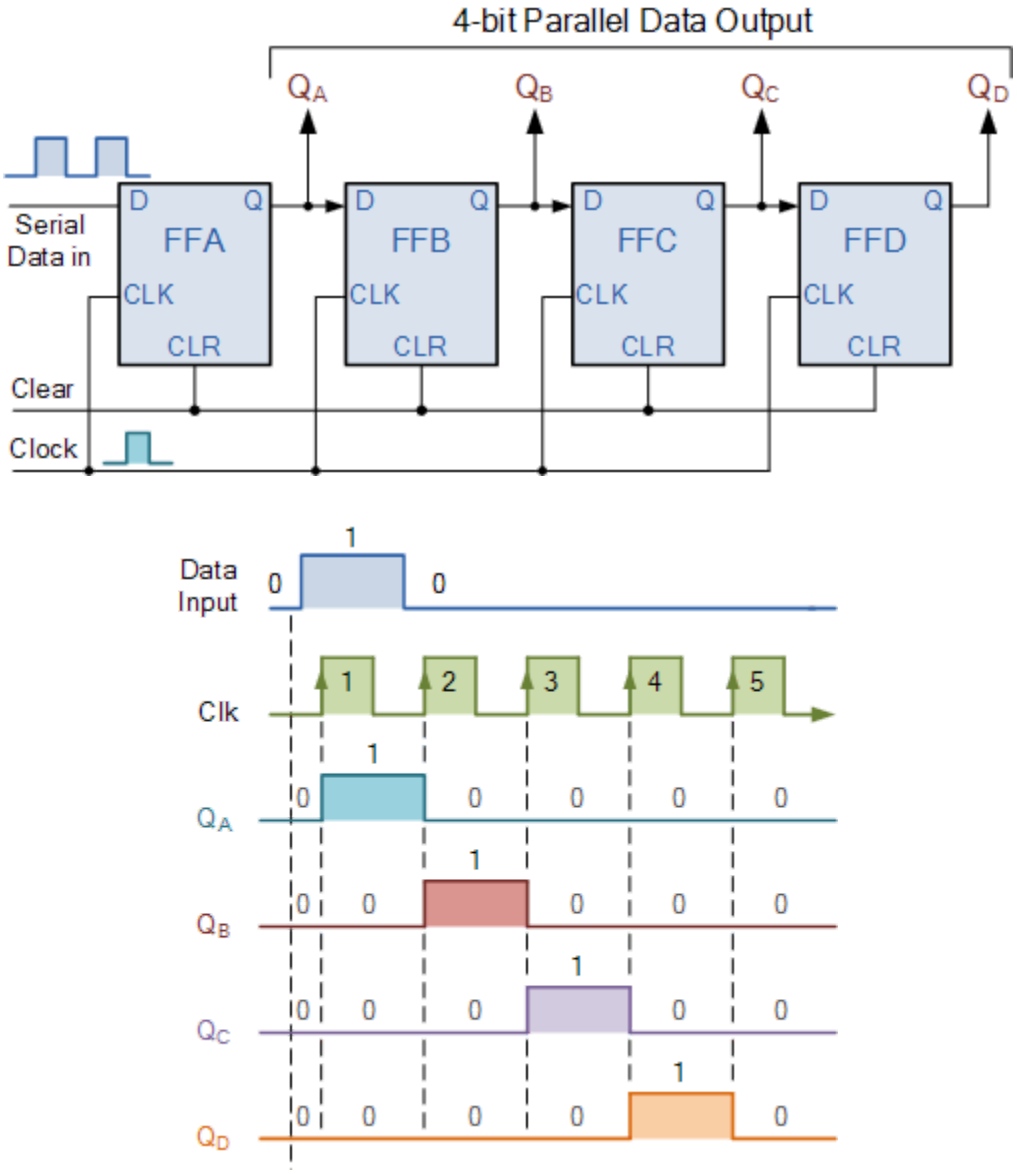
3. Donja tabela prikazuje definiciju pinova N bitnog brojača sa asinhronim clearom. Priložiti VHDL kod i simulacione dijagrame. Dizajnirati kod i sprovesti simulaciju.

IO Pins	Description
Clock	Negative-Edge Clock
Clear	Asynchronous Clear (active High)
Enable	Enable input (active Low)
Q[N:0]	Data Output

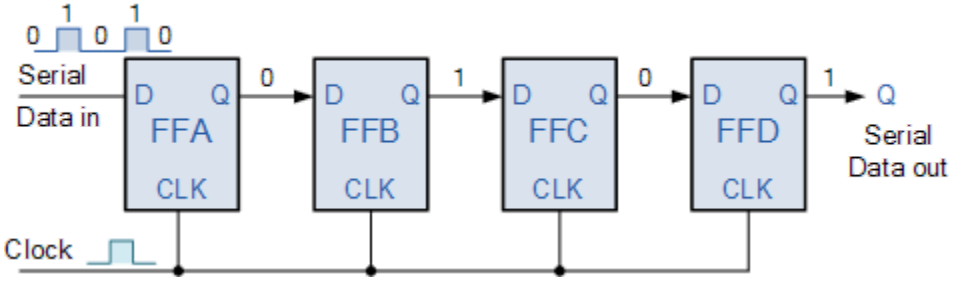
4. Donja tabela prikazuje definiciju pinova N bitnog brojača sa asinhronim clearom koji moze da broi UP/DOWN. Priložiti VHDL kod i simulacione dijagrame. Dizajnirati kod i sprovesti simulaciju.

IO Pins	Description
C	Positive-Edge Clock
CLR	Asynchronous Clear (active High)
UP_DOWN	up/down count mode selector
Q[3:0]	Data Output

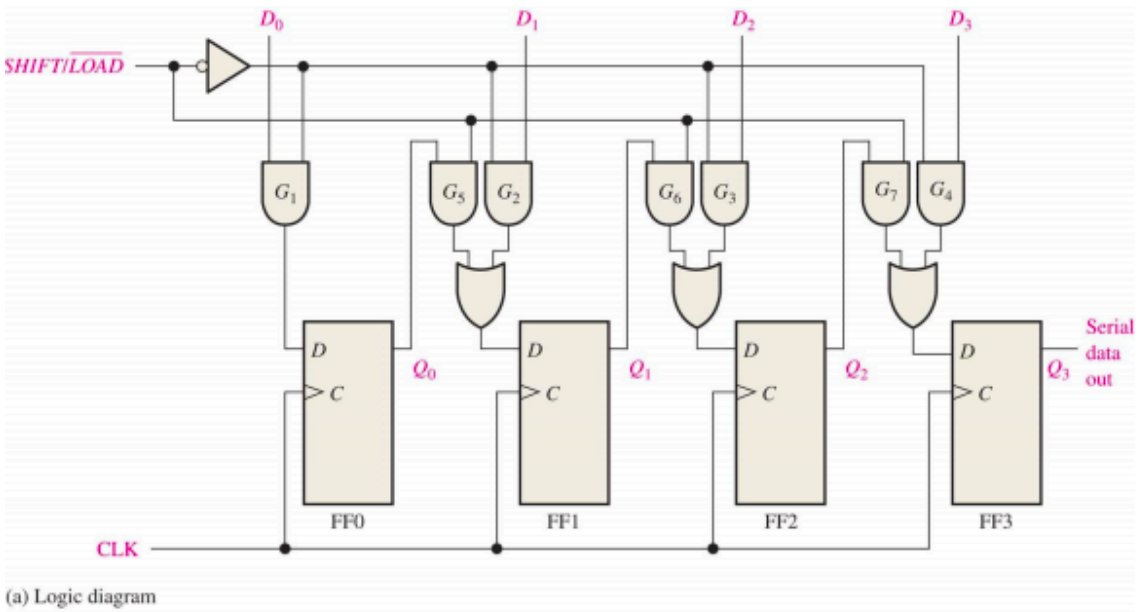
5. Donja slika prikazuje 4 bitni sift registar sa paralelnim izlazima. Priložiti VHDL kod i simulacione dijagrame. Dizajnirati kod i sprovesti simulaciju.



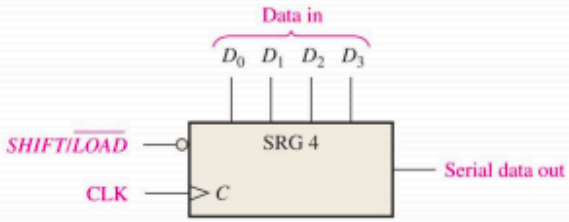
6. Donja slika prikazuje 4 bitni sift registar sa serijskim izlazom. Priložiti VHDL kod i simulacione dijagrame. Dizajnirati kod i sprovesti simulaciju.



7. Donja slika prikazuje 4 bitni sift registar sa serijskim izlazom i paralelnim loadovanjem broja D0,D1,D2,D3. Priložiti VHDL kod i simulacione dijagrame. Dizajnirati kod i sprovesti simulaciju.

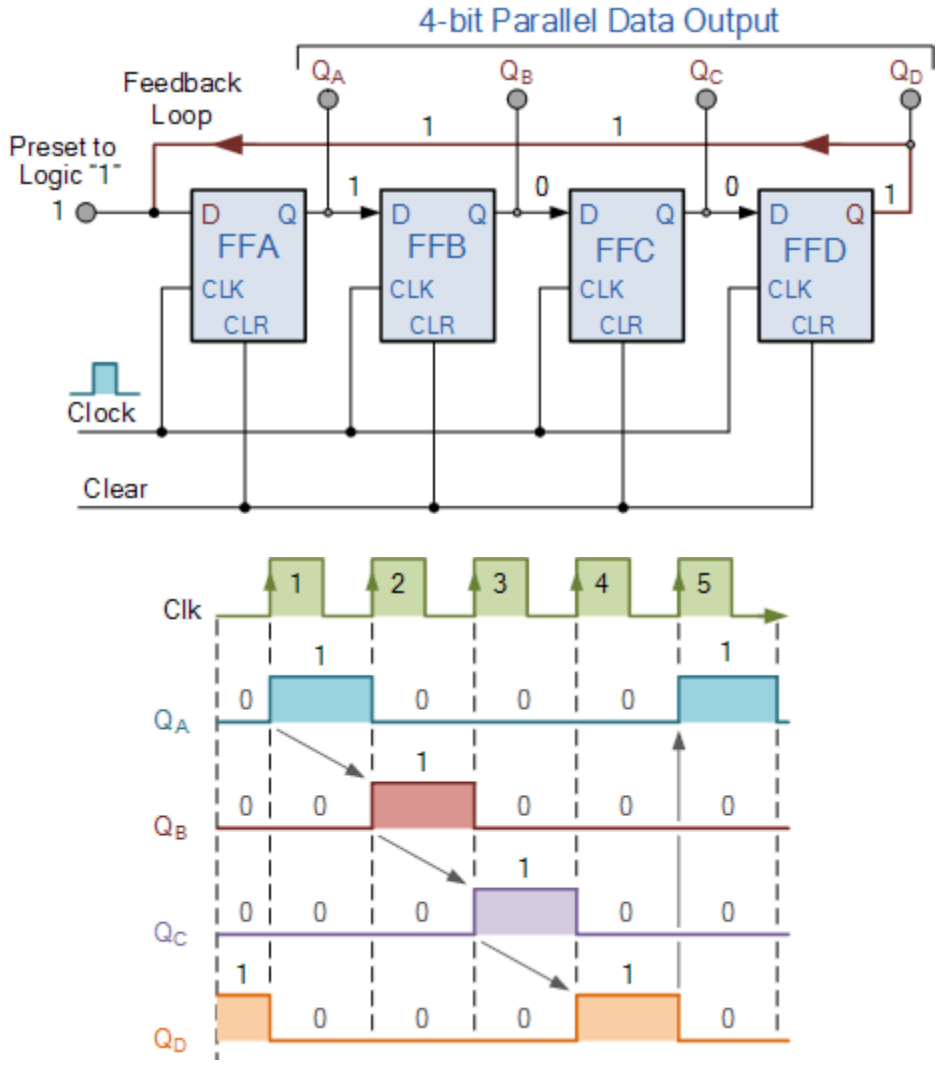


(a) Logic diagram

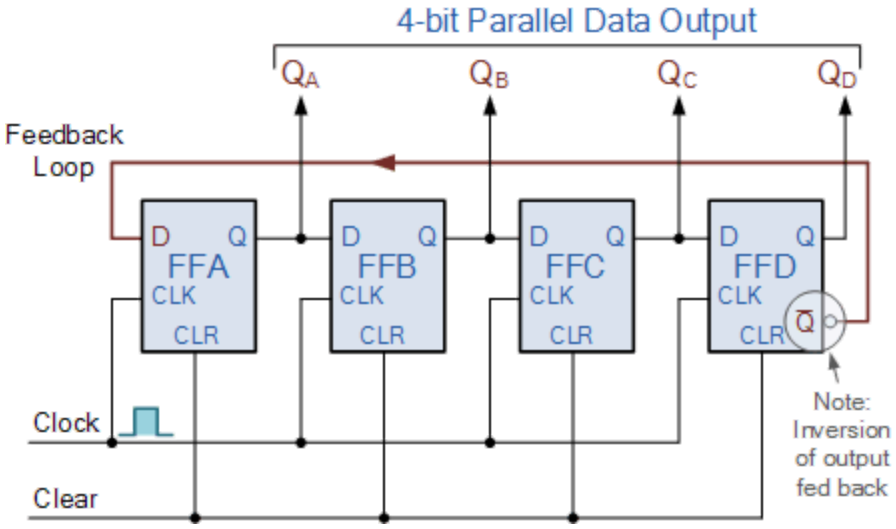


(b) Logic symbol

8. Donja slika prikazuje 4-bitni ring counter. Priložiti VHDL kod i simulacione dijagrame. Dizajnirati kod i sprovesti simulaciju.



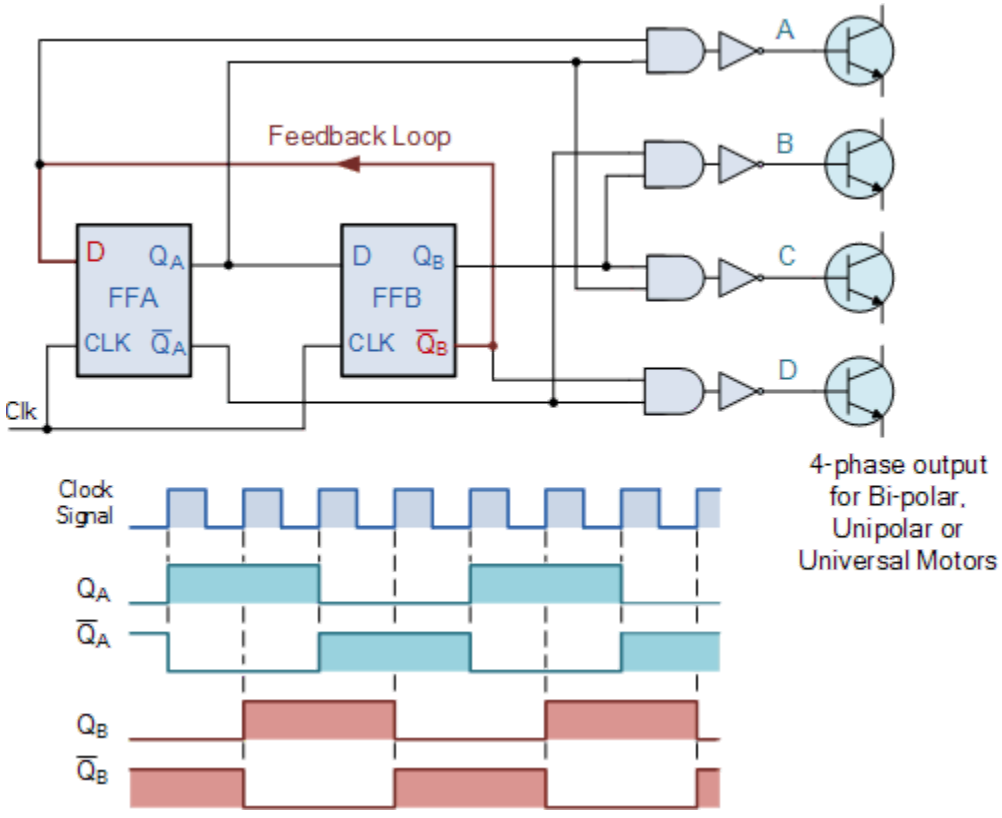
9. Donja slika prikazuje 4-bitni Jonsonov counter/brojac. Priložiti VHDL kod i simulacione dijagrame. Dizajnirati kod i sprovesti simulaciju.



Clock Pulse No	FFA	FFB	FFC	FFD
0	0	0	0	0
1	**1**	0	0	0
2	**1**	**1**	0	0
3	**1**	**1**	**1**	0
4	**1**	**1**	**1**	**1**

5	0	**1**	**1**	**1**
6	0	0	**1**	**1**
7	0	0	0	**1**

10. Donja slika prikazuje 2-bitni Quadrature Generator. Priložiti VHDL kod i simulacione dijagrame. Dizajnirati kod i sprovesti simulaciju.



Output	A	B	C	D
Q_A+Q_B	1	0	0	0
Q_A+Q_B	0	1	0	0
Q_A+Q_B	0	0	1	0
Q_A+Q_B	0	0	0	1

11. 4-bit Unsigned Up Counter with Asynchronous Clear

The following table shows pin definitions for a 4-bit unsigned up counter with asynchronous clear. Perform coding and simulation.

IO Pins	**Description**
C	Positive-Edge Clock
CLR	Asynchronous Clear (active High)
Q[3:0]	Data Output

12. 4-bit Unsigned Up/Down counter with Asynchronous Clear

The following table shows pin definitions for a 4-bit unsigned up/down counter with asynchronous clear. Perform coding and simulation.

IO Pins	Description
C	Positive-Edge Clock
CLR	Asynchronous Clear (active High)
UP_DOWN	up/down count mode selector
Q[3:0]	Data Output

13. 4-bit Unsigned Up Counter with Asynchronous Load from Primary Input

The following table shows pin definitions for a 4-bit unsigned up counter with asynchronous load from primary input. Perform coding and simulation.

IO Pins	Description
C	Positive-Edge Clock
ALOAD	Asynchronous Load (active High)
D[3:0]	Data Input
Q[3:0]	Data Output

14. 8-bit Shift-Left Register with Positive-Edge Clock, Asynchronous Clear, Serial In, and Serial Out

Note Because this example includes an asynchronous clear, XST will not infer SRL16.

The following table shows pin definitions for an 8-bit shift-left register with a positive-edge clock, asynchronous clear, serial in, and serial out. Perform coding and simulation.

IO Pins	Description
C	Positive-Edge Clock
SI	Serial In
CLR	Asynchronous Clear (active High)
SO	Serial Output

15. 8-bit Shift-Left/Shift-Right Register with Positive-Edge Clock, Serial In, and Parallel Out

Note For this example XST will not infer SRL16.

The following table shows pin definitions for an 8-bit shift-left/shift-right register with a positive-edge clock, serial in, and serial out. Perform coding and simulation.

IO Pins	Description
C	Positive-Edge Clock
SI	Serial In
LEFT_RIGHT	Left/right shift mode selector
PO[7:0]	Parallel Output

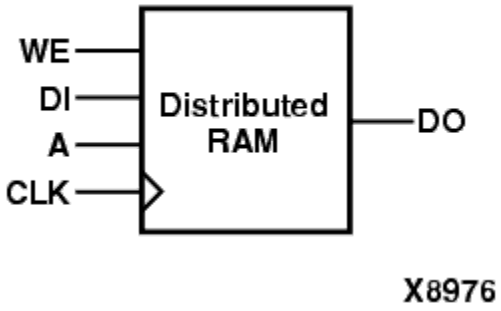
16. Logical shifter

The following table shows pin descriptions for a logical shifter. Perform coding and simulation.

IO pins	Description
D[7:0]	Data Input
SEL	shift distance selector
SO[7:0]	Data Output

17. Single-Port RAM with Asynchronous Read

The following descriptions are directly mappable onto *distributed RAM only*.



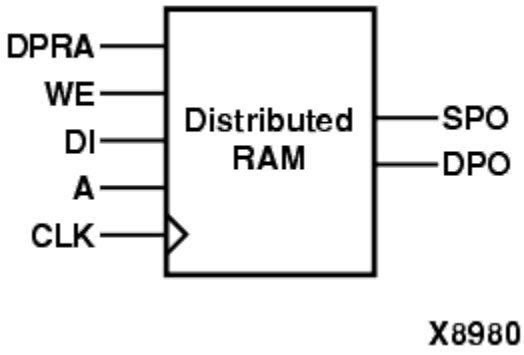
The following table shows pin descriptions for a single-port RAM with asynchronous read. Perform coding and simulation.

IO Pins	Description
clk	Positive-Edge Clock
we	Synchronous Write Enable (active High)
a	Read/Write Address
di	Data Input

do	Data Output

18. Dual-Port RAM with Asynchronous Read

The following example shows where the two output ports are used. It is directly mappable onto *Distributed RAM only*.



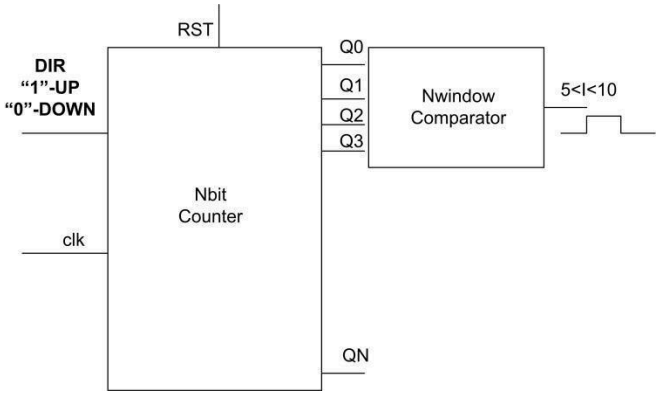
The following table shows pin descriptions for a dual-port RAM with asynchronous read. Perform coding and simulation.

IO pins	Description
clk	Positive-Edge Clock
we	Synchronous Write Enable (active High)
a	Write Address/Primary Read Address
dpra	Dual Read Address
di	Data Input
spo	Primary Output Port
dpo	Dual Output Port

19. a) Projektovati u VHDLu kolo **N**bitnog **UP/DOWN** brojača koji se pogoni klokom **clk** na silaznoj ivici. Osnova brojanja se odredjuje generičkom konstantom **N,** a **RST** je aktivan logičkom 0.

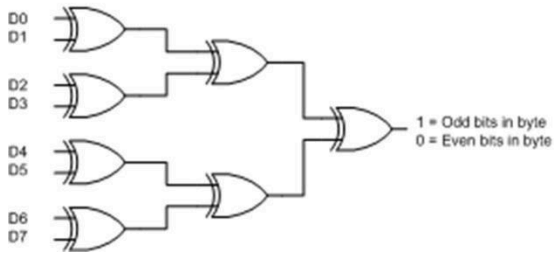
b) Dodati kolo Binarnog N window comparatora čiji je izlaz **I** na stanju logičke „1“ ako je **5<Q0,Q1,Q2,Q3<10**.

c) za slucaj pod b) i frekvenciju clk-a od 100 KHz priložiti simulacione dijagrame sa istaknutim kursorom na poziciji (**Q0,Q1,Q2,Q3)=7**. Takodje priložiti VHDL kod sistema Nbit Counter + Nwindow Comparator.

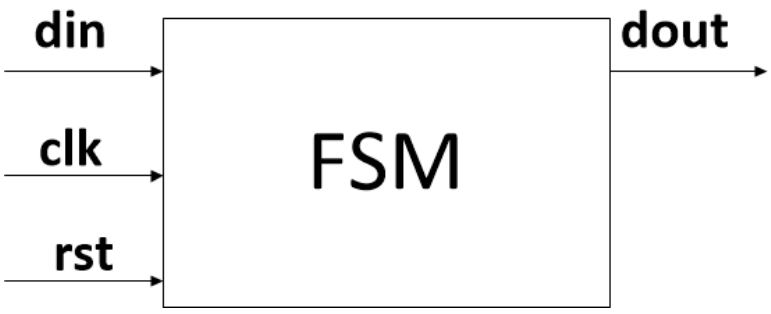


20. a) Projektovati funkciju **parity generatora** koja je smeštena u package-u **my_pcg**. Projektovati kod za provjeru rada funkcije (pod imenom **test_parity**) na primjeru 8 bitnog vektora

Na slici je dat gate opis parity generatora



21. Koristeći FSM, Mooreovog (Moore) tipa, detektovati binarnu sekvencu “101”. Simulirati process.



22. Koristeći FSM, Mielovog (Moore) tipa, detektovati binarnu sekvencu “101”. Simulirati process.

23. Koristeći Shift Registar detektovati binarnu sekvencu “1010”. Simulirati process.

C dio, Test Pitanja

1. Sta je pravo ime za VHDL?

a) Verilog Hardware Description Language

b) Very High speed Description Language

c) Variable Hardware Description Language

d) Very high speed Hardware Description Language

2. Koji je su od sledecih HDL jezika IEEE standardi?
a) VHDL i Verilog
b) C i C++
c) Altera i Xilinx
d) Quartus II i MaxPlus I

3. Sta je od donjeg karakteristika VHDL-a?
a) Case sensitive
b) Use of simple data types
c) Based on C programming language
d) Strongly typed language

4. Koje od sledecih blokova sadrzi osnovni VHDL program?
a) Architecture
b) Entity
c) Process
d) Package

5. Opis kola je dat u:
a) Architecture
b) Entity
c) Library
d) Configurations

6. Jedan entitet moze da sadrzi vise od 1-dne arhitekture?
a) True
b) False

7. U VHDL-u, Bus tip je?
a) Signal
b) Constant
c) Variable
d) Drive

8. Sta od sledeceg nije definisano u entitetu?

a) Direction of any signal

b) Names of signal

c) Different ports

d) Behavior of the signals

9. Sta od sledecega moze biti ime entiteta?

a) Nand_gate

b) NAND

c) AND

d) IF

10. Koja je od sledecih korektna sintaksa za deklaraciju entiteta?

a)

```
ENTITY entity_name IS
   PORT( signal_names : signal_modes;
   signal_names : signal_modes);
  END entity_name;
```

b)

```
 ENTITY entity_name
   PORT( signal_names : signal_modes;
   signal_names : signal_modes);
  END ENTITY
```

11. Sta je tacna deklaracija i definicija arhitekture?

a)

```
ARCHITECTURE architecture_type OF entity_name IS
    Declarations_for_architecture;
    BEGIN
    Code;
    ….
   END architecture_name;
```

b)

```
ARCHITECTURE architecture_name OF entity_name IS
    BEGIN
```

```
  Declarations_for_architecture;
  Code;
  ….
 END architecture_name;
```

c)

```
ARCHITECTURE architecture_type OF entity_name IS
  BEGIN
  Declarations_for_architecture;
  Code;
  ….
 END architecture_type;
```

d)

```
ARCHITECTURE architecture_name OF entity_name IS
  Declarations_for_architecture
  BEGIN
  Code;
  ….
  END architecture_name;
```

12. SIGNED i UNSIGNED data tipovi su definisani u?
    a) std_logic_1164 package
    b) std_logic package
    c) std_logic_arith package
    d) standard package

13. Koju vrijednost poprima x u sledecem kodu?

```
SIGNAL x : IN UNSIGNED (3 DOWNTO 0 );
x <= “1100”;
```

a) 12
b) 5
c) -5
d) 14

14. SIGNAL a : REAL; sta je od donjeg nedozvoljeno pridruzivanje za a)?

a) a <= 1.8
b) a <= 1.0 E10
c) a <= 1.0 E-10
d) a <=1.0 ns

15. Vise dimenzionalni nizovi se mogu upotrijebiti za implementaciju memorije?

a) True.

b) False.

16. Sta od sledeceg ne moze biti vrijednost za x? Pogledaj donji kod?

```
TYPE color IS (red, green, blue, black, white, gray);
SUBTYPE primary IS color RANGE red to blue;
VARIABLE x: primary;
```

a) White
b) Red
c) Green
d) Blue

17. Koliko bitova moze biti smjesteno u promenljivu array2

```
TYPE array1 IS ARRAY ( 0 TO 3 ) OF BIT_VECTOR (3 DOWNTO 0 );
TYPE array2 IS ARRAY ( 0 TO 3 ) OF array1;
```

a) 16
b) 9
c) 64
d) 27

18. Korisnik moze definisati svoj integer tip?

a) True

b) False

19. U VHDL mozete primijeniti osnovne aritmeticke operacije nad razlicitim tipovima podataka?
    a) True
    b) False

20. U donjem kodu sta ce biti greska pri kompalaciji?

```
TYPE my_int IS INTEGER RANGE -32 TO 32;
TYPE other_int IS INTEGER RANGE 0 TO 100;
SIGNAL x : my_int;
SIGNAL y : other_int;
y <= x + 2;
…
```

a) Type mismatch
b) Syntax problem
c) No declaration
d) Can’t compile

21. Kojeg tip ce biti vrijednost y poslije izvrsavanja sledeceg koda, koliko bita?

```
Library ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
…
SIGNAL m : UNSIGNED (3 DOWNTO 0);
SIGNAL n : UNSIGNED (3 DOWNTO 0);
SIGNAL y : STD_LOGIC_VECTOR (7 DOWNTO 0);
y <=CONV_STD_LOGIC_VECTOR ((m+n), 8);
…
```

a) 8- bit STD_LOGIC_VECTOR m+n
b) 8- bit UNSIGNED m+n
c) 4- bit STD_LOGIC m+n
d) Error

22. Koji od donjih operatora nije operator pridruzivanja?

a) <=
b) :=
c) =>
d) =

23. A VARIABLE y je deklarisana kao STD_LOGIC_VECTOR tip, 4 bita, ako zelis da pridruzis 1001 za y, koji je pravi izraz ?

a) y <= “1001”
b) y := “1001”
c) y <= ‘1’, ‘0’, ‘0’, ‘1’
d) y => “1001”

24. Sta je funkcija shift operatora?

a) To shift the data
b) To shift the identifiers
c) To shift the operators
d) To shift the STD_LOGIC_VECTOR

25. a <= b after 10ns; u ovoj recenici se definise delay?

a) True
b) False

26. Koje je kolo definisano donjim kodom?

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
ENTITY my_func IS
PORT(x, a, b : IN std_logic;
q : OUT std_logic);
END my_func;
```

```
ARCHITECTURE behavior OF my_func IS
SIGNAL s : INTEGER;
BEGIN
WITH s SELECT
q <= a AFTER 10 ns WHEN 0;
       b AFTER 10 ns WHEN 1;
s <= 0 WHEN x = '0' ELSE
      1 WHEN x = '1';
END behavior;
```

a) AND gate
b) OR gate
c) MUX 2:1
d) DEMUX 1:2

27. U kojem dijelu VHDL koda se definise generic, dati primjer?
a) Package declaration
b) Entity
c) Architecture
d) Configurations

28. Ako postoji vise procesa u VHDL kodu kako se oni izvrsavaju?
a) One after the other
b) Concurrently
c) According to sensitivity list
d) Sequentially

29. Local varijable u procesu mogu biti definisane na kojem mjestu?
a) Anywhere within the process
b) After a sequential statement
c) Before the BEGIN keyword
d) After the BEGIN keyword

30. Koja je o donjih ispravna sintaksa za deklaraciju procesa ?

a)

```
{Label :} PROCESS
{process_declaration_part};
 sensitivity_list;
BEGIN
sequential_statements;
END PROCESS {Label};
```

b)

```
PROCESS {sensitivity_list}
{process_declaration_part}
BEGIN
sequential_statements;
END PROCESS {Label};
```

c)

```
{Label :} PROCESS
{process_declaration_part}
BEGIN
sensitivity_list;
sequential_statements;
END PROCESS;
```

d)

```
{Label :} PROCESS {sensitivity_list}
{process_declaration_part}
BEGIN
sequential_statements;
END PROCESS {Label};
```

31. Koja od donjih kljucnih rijeci ne pripada If naredbi?

a) ELSE

b) THEN

c) WHEN

d) ELSIF

32. Koliko iteracija ce se odraditi u donjoj for petlji?

```
FOR i IN 0 TO 5 LOOP
```

a) 6

b) 4

c) 5

d) 7

33. Koja je od donjih tacna upotreba signala?

a) To set default value

b) To declare a variable

c) To represent local information

d) To pass value between circuits

34. Koja je od donjih promenljivih lokalna za blok u kojem se deklarise?

a) Signal

b) Variable

c) Constant
d) Float

35. Konstanta koja je definisana u ARCHITECTURE, bice dostupna

a) In the process within the architecture
b) Whole code
c) Within the same architecture
d) In the entity associated and corresponding architecture

36. U procesu se dodjeljuje variabla, nova vrijednost se dodjeljuje
a) After one delta cycle
b) Immediately
c) At the end of a process
d) At the end of architecture

37. Ne postoji delay u slucaju varijabli
a) True
b) False

38. Sta je tacno u vezi package?
a) Package is collection of libraries
b) Library is collection of packages
c) Package is collection of entities
d) Entity is collection of packages

e) Package is collection of the components

39. Ono sto je deklarisano u package je vidljivo za:
a) Every design unit
b) Package body only
c) Library containing that package
d) Design unit that USE the package

40. Koji je standardni package ukljucen u VHDL kod i ne mora se deklarisati?

a) STD_LOGIC_1164

b) STANDARD
c) TEXTIO
d) STD_LOGIC_ARITH

41. Funkcija se poziva iz:
a) Function itself
b) Library

c) Main code

d) Package

42. Koliko vracenih vrijednosti posjeduje funkcija?

a) 1

b) 2

c) 3

d) 4

43. Da li funkcija u VHDLu moze imati imati vise parametara

a) False

b) True

44. Sta od sledeceg moze biti parameter funkcije?

```
SIGNAL a, b : IN STD_LOGIC
VARIABLE c : INTEGER
CONSTANT d : INTEGER
```

a) a

b) a,b

c) a,b,c

d)

d

45. Kakve direkcije je signal kao argument donje funkcije?

```
FUNCTION my_func (SIGNAL a : STD_LOGIC_VECTOR) RETURN INTEGER IS
…..;
```

a) IN

b) OUT

c) INOUT

d) BUFFER

46. Procedura u VHDL-u vraca vrijednos?

a) False

b) True

47. Sta ce da bude vrijednost sekvence my_array'LENGTH, ako je definisana na sledeci nacin?

```
TYPE my_array IS ARRAY (15 DOWNTO 0) OF STD_LOGIC;
```

a) 15

b) 8

c) 0

d) 16

48. Sekvenca atributa definisana dolje opisuje stanje kloka:

```
IF (clk'EVENT and clk = '1')
```

a) Rising edge of the clock signal
b) Falling edge of the clock signal
c) Clock signal frequency
d) Time period of clock signal

49. Koje je logicko kolo opisano donjim kodom?

```
ARCHITECTURE gate OF my_gate IS
BEGIN
WITH ab SELECT
y<= 0 WHEN "01" OR "10";
      1 WHEN OTHERS;
END gate;
```

a) NAND
b) NOR
c) EXOR
d) EXNOR

50. Koji se od donjih atributa primenjuju u sekvencijalnim kolima?

a) 'STABLE
b) 'LENGTH
c) 'LAST_EVENT
d) 'EVENT

51. Koliko imamo tipova reseta u sekvencijalnim kolima?

a) One
b) Two
c) Three
d) Four

52. Sta je rezultat sledeceg izraza bez obzira na B A + AB?

a) A
b) B
c) AB
d) A + B

53. Sta je funkcija donjeg kola ako SRL operator shift desno?

```
ARCHITECTURE my_func OF my_logic IS
BEGIN
y <= x SRL 3;
END my_func;
```

a) Divide by 8
b) Divide by 4
c) Multiply by 2
d) Multiply by 4

54. Kod dat dolje predstavlja VHDL implementaciju?

```
ARCHITECTURE my_circuit OF my_logic IS
BEGIN
WITH ab SELECT
y <= x0 WHEN “00”;
       x1 WHEN “01”;
       x2 WHEN “10”;
       x3 WHEN “11”;
END my_circuit;
```

a) 4 to 1 MUX
b) 1 to 4 DEMUX
c) 8 to 1 MUX
d) 1 to 8 DEMUX

55. Sta je funkcija donjeg koda?

```
ENTITY my_logic IS
PORT (din : STD_LOGIC_VECTOR(7 DOWNTO 0);
          Count : STD_LOGIC_VECTOR(3 DOWNTO 0));
END my_logic;
ARCHITECTURE behavior OF my_logic IS
BEGIN
Count <= “0000”
PROCESS(din)
BEGIN
L1: FOR i IN 0 TO 7 LOOP
IF(din(i) = ‘0’) THEN
Count = count+1;
ELSE
NEXT L1;
END LOOP;
END PROCESS;
END behavior;
```

a) To count number of ones in the given data
b) To count number of zeroes in the given data
c) To reverse the order of given data
d) To perform binary multiplication of two data inputs

56. Generator bita parnosti je sekvencijalno kolo?
a) True
b) False

57. Shift registar se koristi za kasnjenje signala?
a) True
b) False

58. Koji se od donjih flip-flop-a koristi kao ring counter?
a) T flip-flops
b) SR flip-flops
c) JK flip-flops
d) D flip-flops

59. Sta je greska u donjem kodu?

```
signal A, B, C, X, Y : integer;
begin
  process (A, B, C)
    variable M, N : integer;
  begin
    M <= A;
    N := B;
    X <= M + N;
    M := C;
    Y <= M + N;
  end process;
```

60. Data je kod binarnog komparatora realizovanog preko procedure i prekou funkcije.

a) Dopuniti kod gdje se nalaze XXXX

b) Ako je X=0 i Y=1 koja je vrijednost za Z1 i Z2?

```
libraryieee;

use ieee.std_logic_1164.all;

Entity comp_by_func_by_proc is

port

(

X,Y : in bit;

Z1,Z2 : out bit
```

```
);
end entity;
architecturebehav of comp_by_func_by_proc is
-- comparator by function
function COMP_BITS(A, B: in bit) return bit is
begin
if A=B then return '1'; else return '0';
end if;
end COMP_BITS;
-- comparator by procedure
procedure COMP_BITS(signal A, B: bit; signal C: out bit) is
begin
if A=B then C<='1'; else C<='0'; end if;
end procedure;
-- begin of architecture
begin
Z1<=XXXXX -- call architecture
XXXXX --  call procedure and store in Z2
end architecture;
```

61. Nacrtati semu osnovnog logickog elementa FPGA i objasniti princip rada

62. Pomocu LUT (look up table) implementirati funkciju z= a and b or(c xor d)

63. Element prekidacke matrice kod FPGA se satoji od:

a) tranzistora

b) MOSFET-ova

c) operacionih pojacavaca

64. FPGA cip se konfigurise preko

a) 1 pina

b) vise pinova, poznatih kao konfiguracioni pinovi

c) opticki

64. Nastavlja se dodavanje

D dio, Praktični rad

E dio, Laboratorija

F dio, Praktični problemi

G-dio, Seminarski