Белорусский государственный университет
информатики и радиоэлектроники

4-ый курс специальности ПОИТ

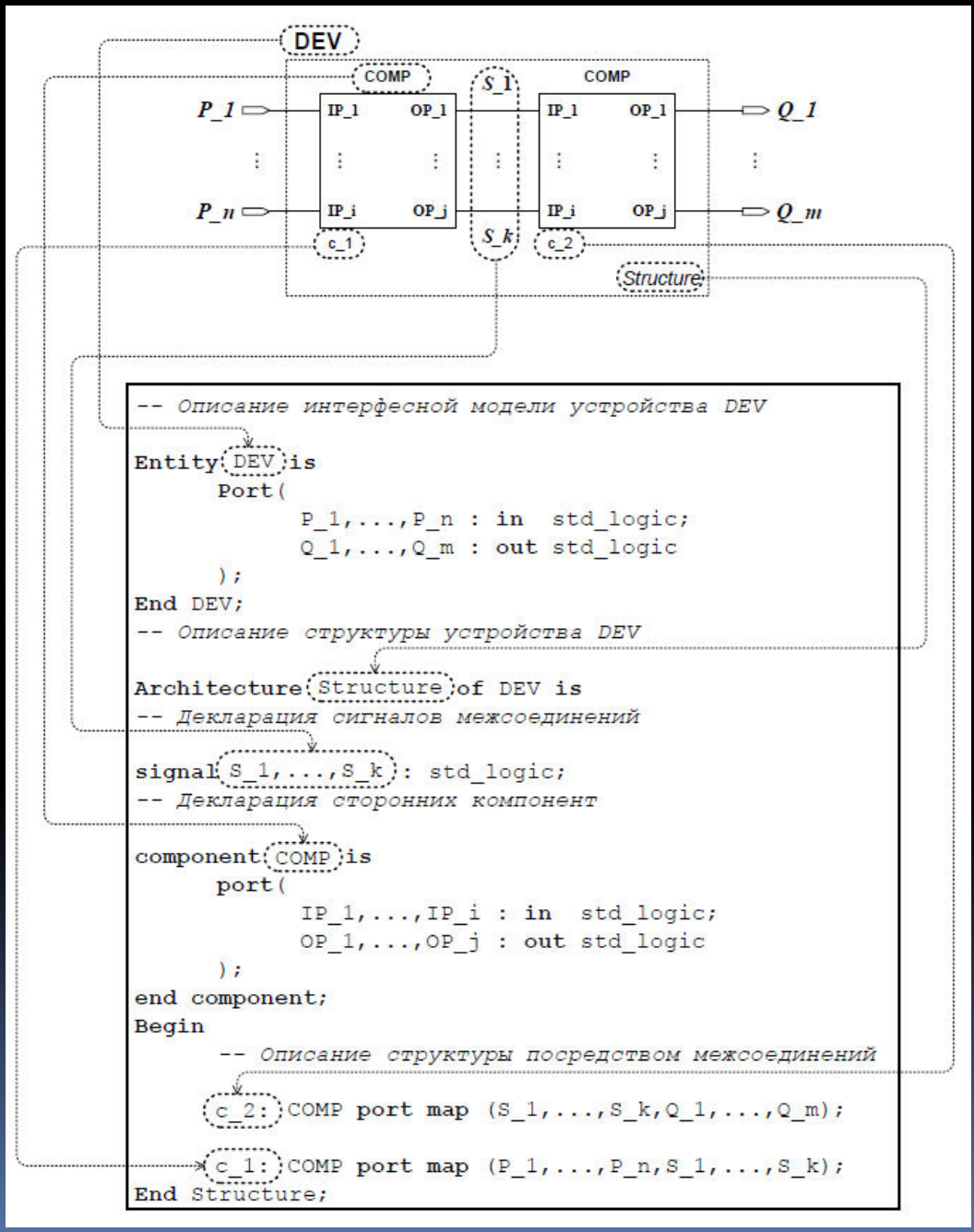
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЦИФРОВОГО ПРОЕКТИРОВАНИЯ

2020

Introducing to VHDL

Description Styles:

Structural

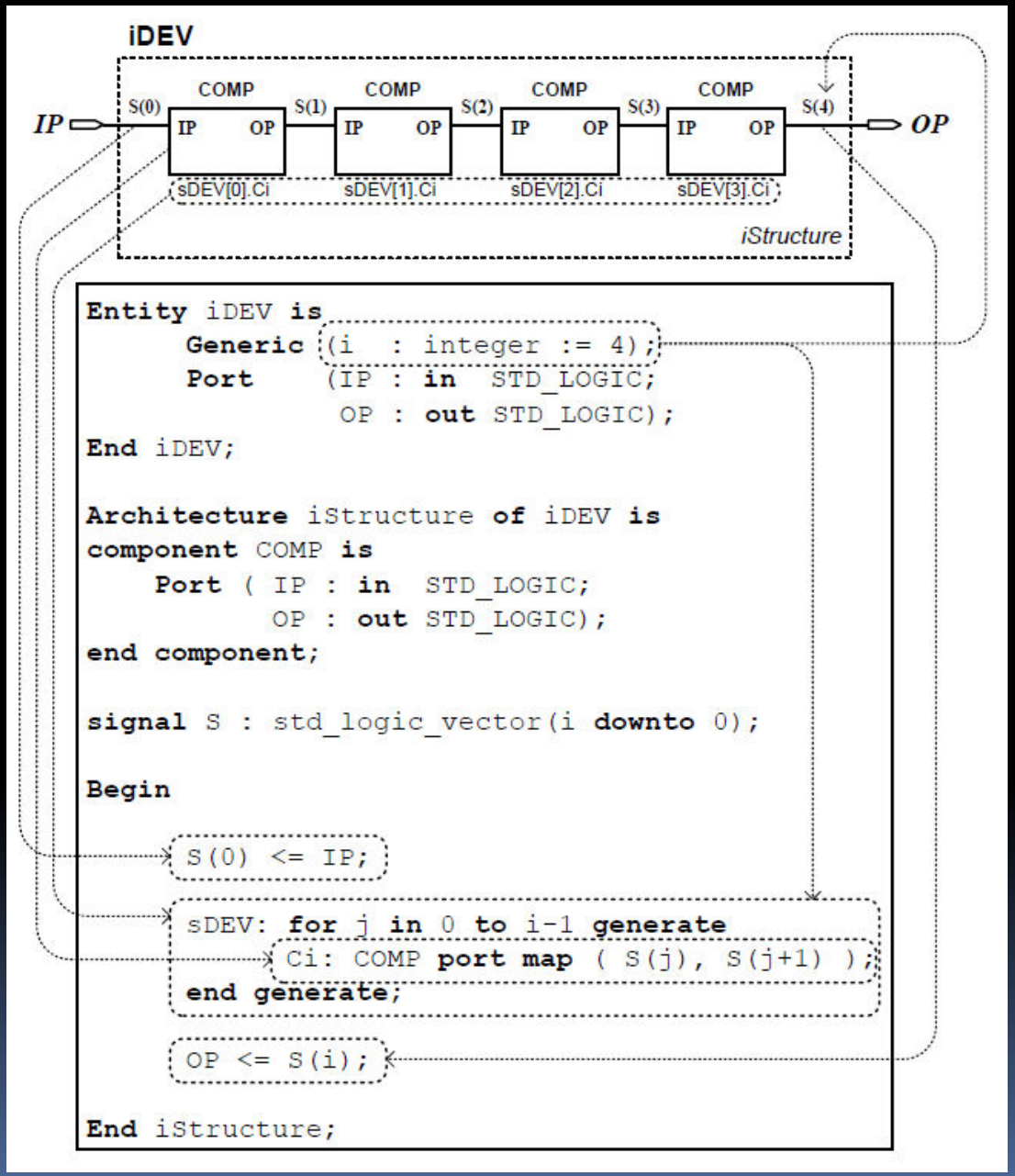



```
-- Описание интерфесной модели устройства DEV

Entity DEV is
      Port(
            P_1,...,P_n : in  std_logic;
            Q_1,...,Q_m : out std_logic
      );
End DEV;
-- Описание структуры устройства DEV

Architecture Structure of DEV is
-- Декларация сигналов межсоединений

signal S_1,...,S_k : std_logic;
-- Декларация сторонних компонент

component COMP is
      port(
            IP_1,...,IP_i : in  std_logic;
            OP_1,...,OP_j : out std_logic
      );
end component;
Begin
      -- Описание структуры посредством межсоединений

      c_2: COMP port map (S_1,...,S_k,Q_1,...,Q_m);

      c_1: COMP port map (P_1,...,P_n,S_1,...,S_k);
End Structure;
```

Introducing to VHDL

Description Styles:

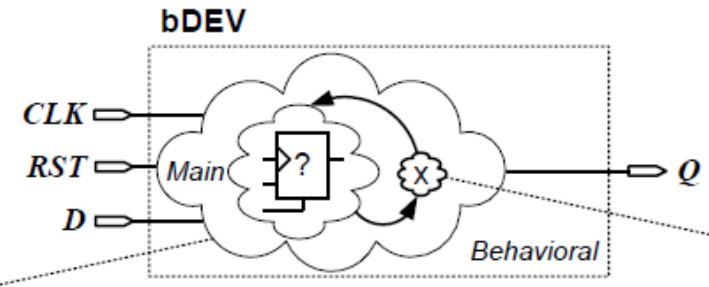
Iterative Structural



Introducing to VHDL

Description Styles:

Behavioral

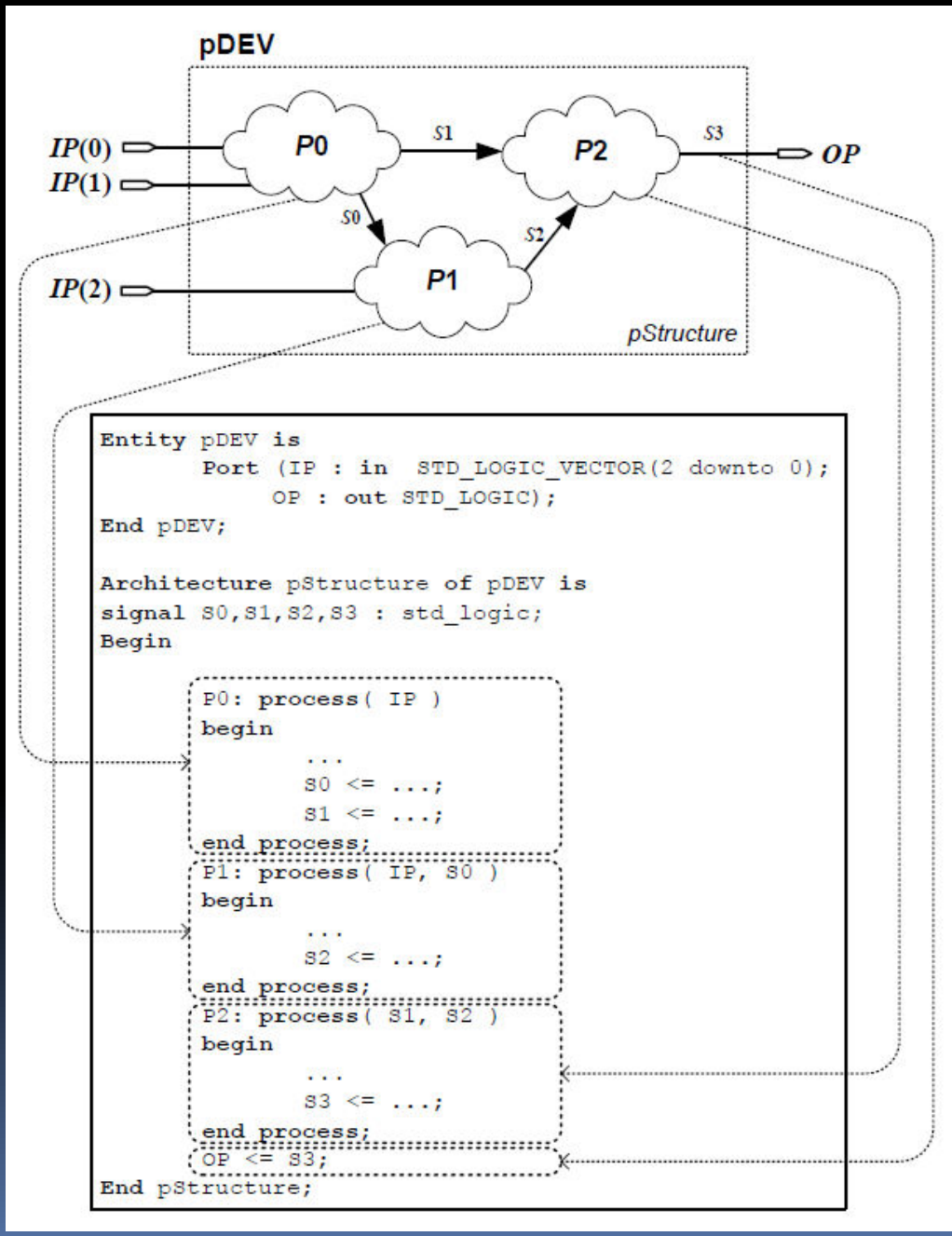



```
Entity bDEV is
      Port (CLK,RST,D : in  STD_LOGIC;
                    Q : out STD_LOGIC);
End bDEV;

Architecture Behavioral of bDEV is
Begin

      MAIN: process( CLK, RST, D )
      variable x: std_logic;
      begin
            if RST='1' then
                  x := '0';
            elsif CLK'event and CLK='1' then
                  x := D;
            end if;
            Q <= x;
      end process;

End Behavioral;
```

Introducing to VHDL

Description Styles:

Pseudo Structural

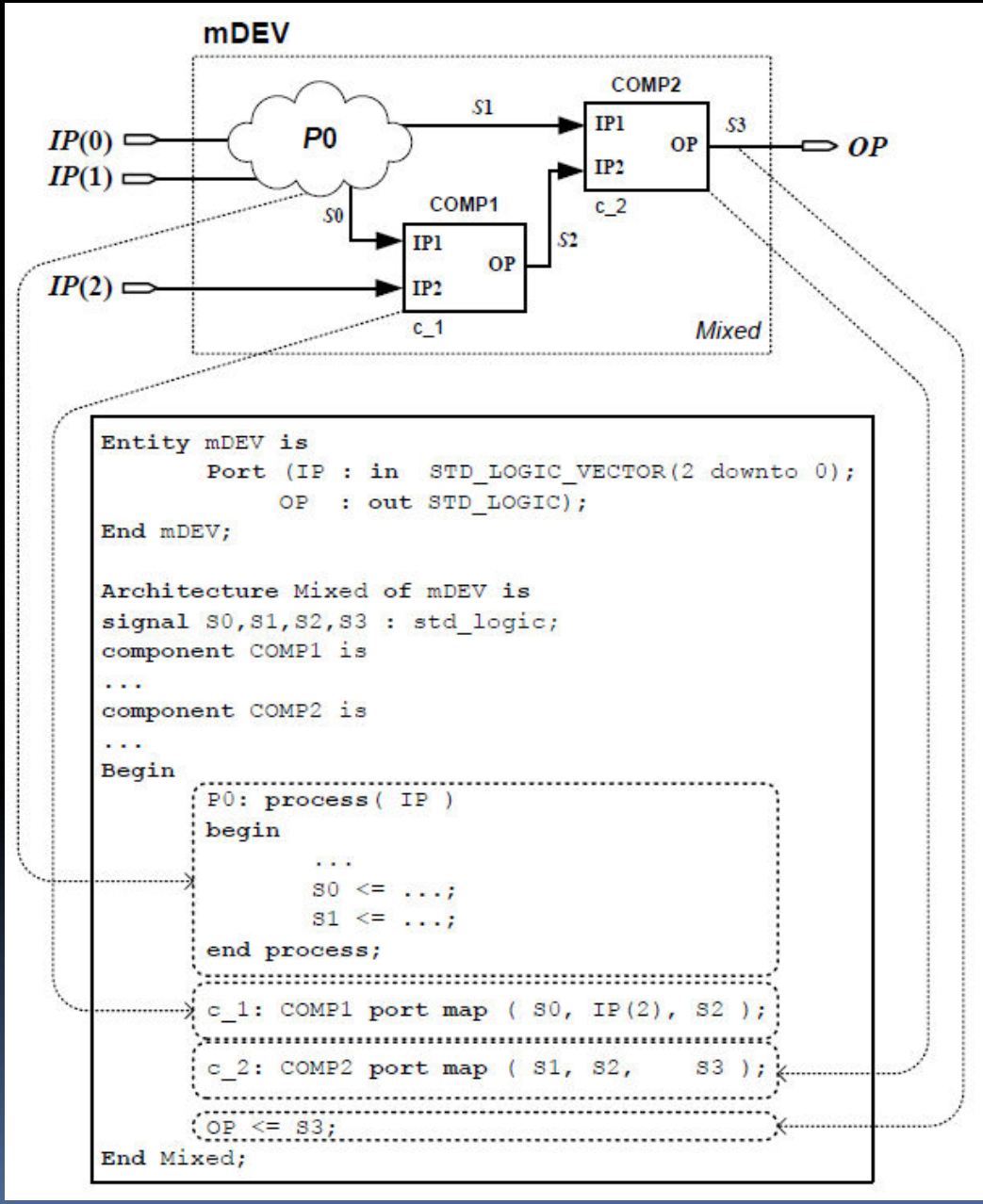



```
Entity pDEV is
        Port (IP : in  STD_LOGIC_VECTOR(2 downto 0);
              OP : out STD_LOGIC);
End pDEV;

Architecture pStructure of pDEV is
signal S0,S1,S2,S3 : std_logic;
Begin

        P0: process( IP )
        begin
                ...
                S0 <= ...;
                S1 <= ...;
        end process;
        P1: process( IP, S0 )
        begin
                ...
                S2 <= ...;
        end process;
        P2: process( S1, S2 )
        begin
                ...
                S3 <= ...;
        end process;
        OP <= S3;
End pStructure;
```

Introducing to VHDL

Description Styles:

Mixed




```
Entity mDEV is
        Port (IP : in  STD_LOGIC_VECTOR(2 downto 0);
              OP  : out STD_LOGIC);
End mDEV;

Architecture Mixed of mDEV is
signal S0,S1,S2,S3 : std_logic;
component COMP1 is
...
component COMP2 is
...
Begin
        P0: process( IP )
        begin
                ...
                S0 <= ...;
                S1 <= ...;
        end process;
        c_1: COMP1 port map ( S0, IP(2), S2 );
        c_2: COMP2 port map ( S1, S2,    S3 );
        OP <= S3;
End Mixed;
```

Introducing to VHDL

Combinational Logic: Universal Table of Truth

```
1  Library IEEE;
2  Use IEEE.STD_LOGIC_1164.ALL;
3  Use IEEE.STD_LOGIC_ARITH.ALL;
4  Use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  Entity CLOGIC_4x1 is
7
8      Generic ( -- Число входных портов компоненты
9                NI    : Integer range 1 to 3          := 1;
10               -- Значения функции в векторном шестнадцатиричном формате
11               INIT : std_logic_vector( 15 downto 0 ) := x"0001");
12
13     Port    ( -- Входная Ni-разрядная шина
14               IP    : in  std_logic_vector( NI-1 downto 0 );
15               -- Выходной порт
16               F     : out std_logic );
17 End CLOGIC_4x1;
18
19 Architecture Behavioral of CLOGIC_4x1 is
20 Begin
21
22   -- Из младшей части таблицы INIT,
23   --  размерность которой ограничивается параметром Ni,
24   --  выбирается двоичное значение, хранящееся по адресу IP.
```

```
25
26   -- Выбранное значение есть значение функции, транслируемое
27   --  на выходной порт F.
28   F <= INIT( 2**NI-1 downto 0 )( CONV_INTEGER( IP ) );
29
30 End Behavioral;
```

Introducing to VHDL

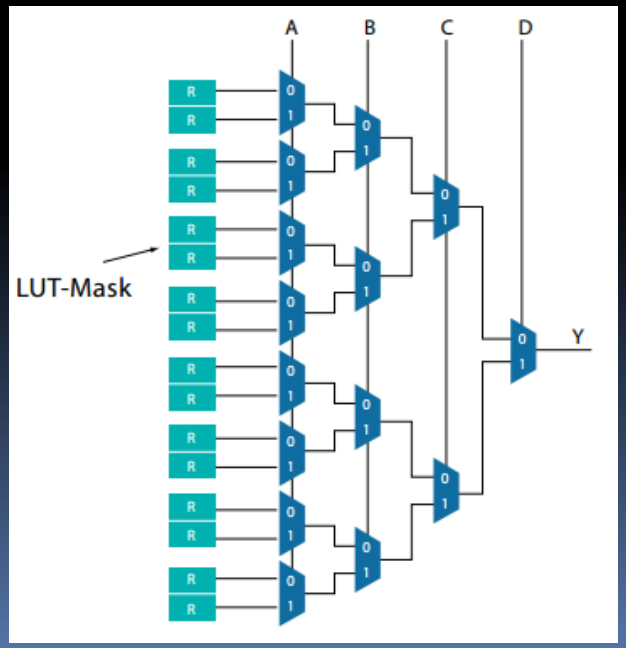
Combinational Logic: Universal Table of Truth

```
-- ...
-- F <= not A;
U0: CLOGIC_4    generic map ( 1, x"0001" )
                port map    ( A, F );

-- F <= A and B;
U1: CLOGIC_4    generic map ( 2, x"0004" )
                port map    ( A & B, F );

-- F <= A xor B;
U1: CLOGIC_4    generic map ( 2, x"0006" )
                port map    ( A & B, F );

-- F <= A xor B xor C;
U1: CLOGIC_4    generic map ( 3, x"0096" )
                port map    ( A & B & C, F );

-- ...
```

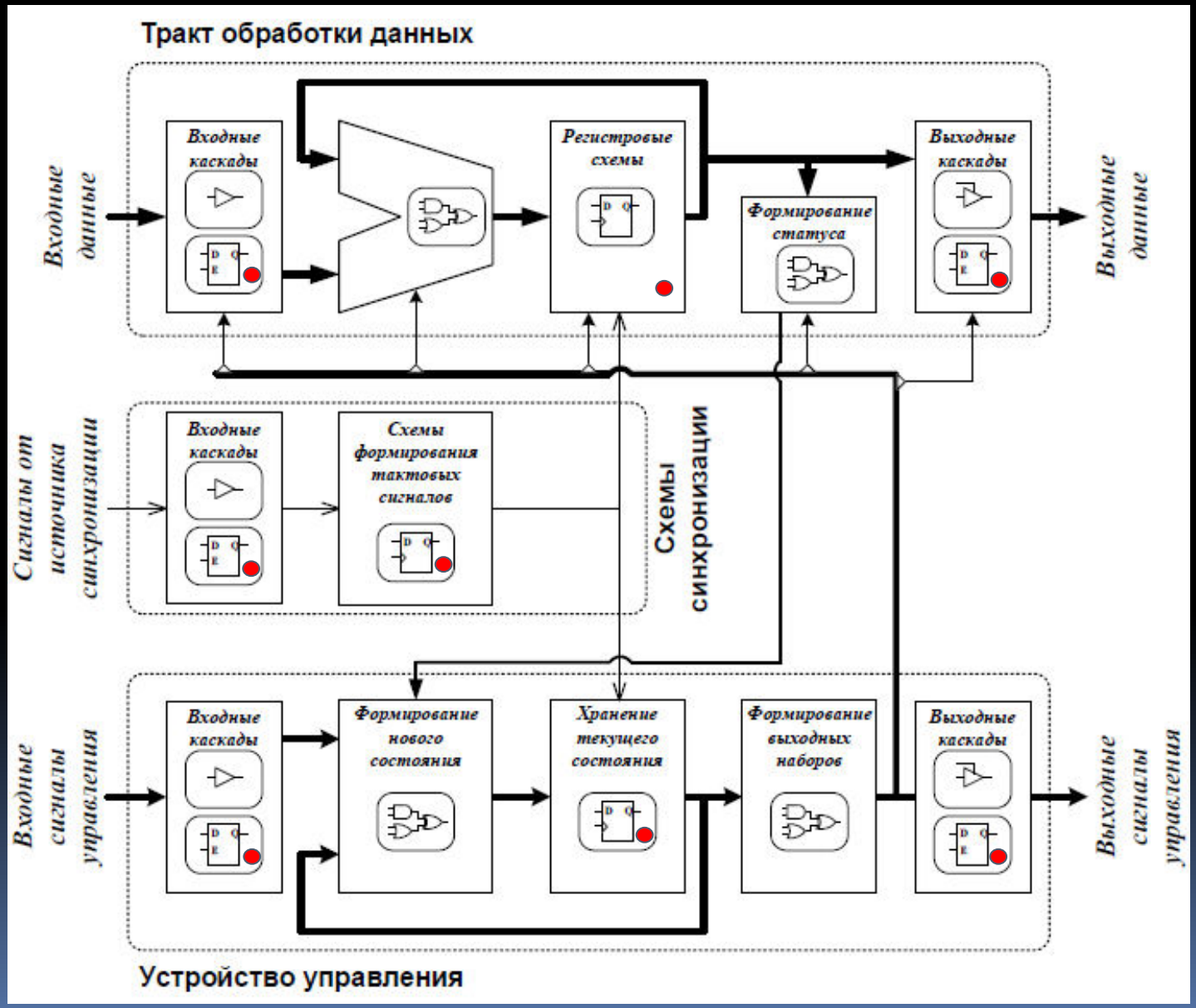
Introducing to VHDL

Combinational Logic: FPGA Look-Up Table usage

```
Library UNISIM;
use UNISIM.vcomponents.all;
-- ...
   -- LUT4: 4-input Look-Up Table with general output
   --       Spartan-3E
   -- Xilinx HDL Language Template, version 12.4

   LUT4_inst : LUT4
   generic map (
    INIT => X"0000")
port map (
   O => O,     -- LUT general output
   I0 => I0,   -- LUT input
   I1 => I1,   -- LUT input
   I2 => I2,   -- LUT input
   I3 => I3    -- LUT input
);

-- End of LUT4_inst instantiation
```

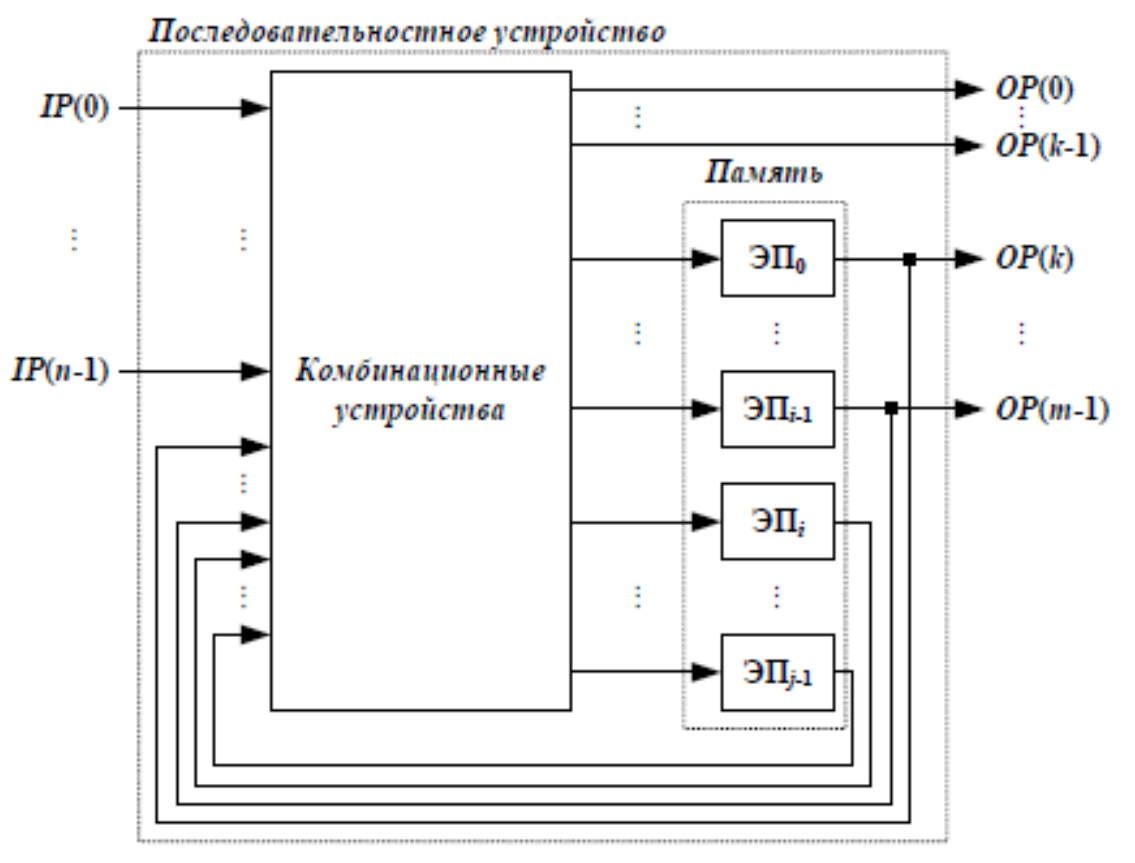


Introducing to VHDL Sequential Logic:



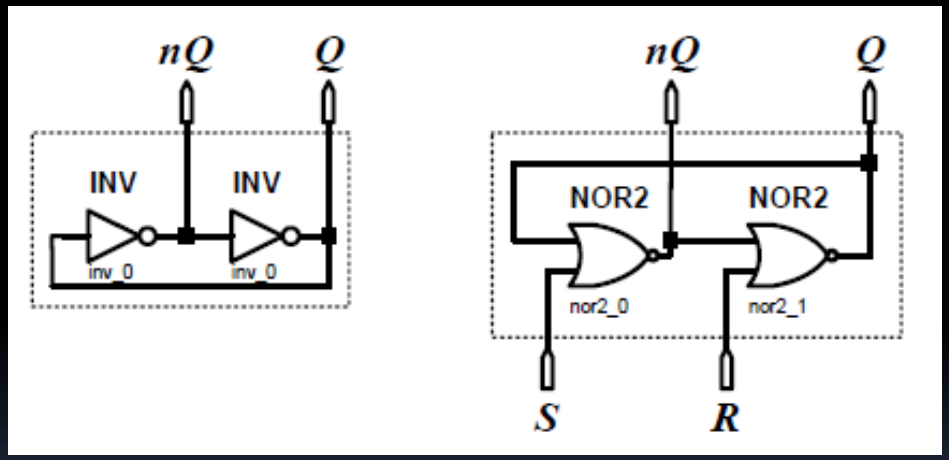
Introducing to VHDL

Sequential Logic:



Introducing to VHDL

Sequential Logic: bi-stable elements



Introducing to VHDL

Sequential Logic: bi-stable elements

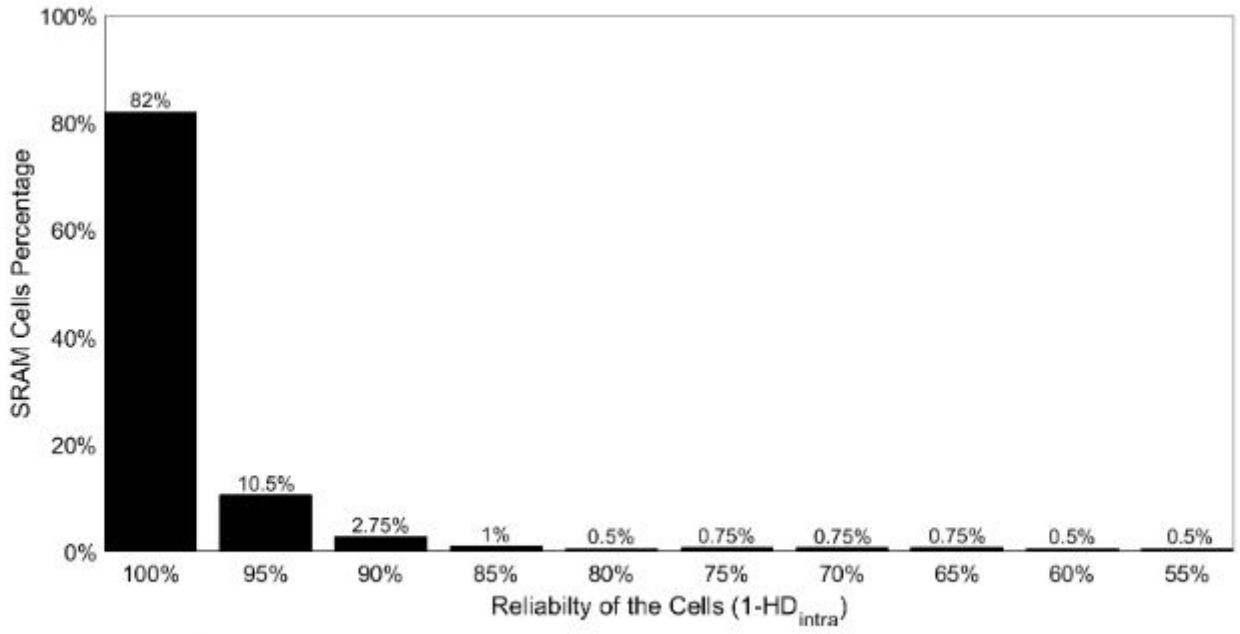


Figure 1: The Reliability of the SRAM Cells Represented by the Number of Cells Which Belong to the Same Reliability Category

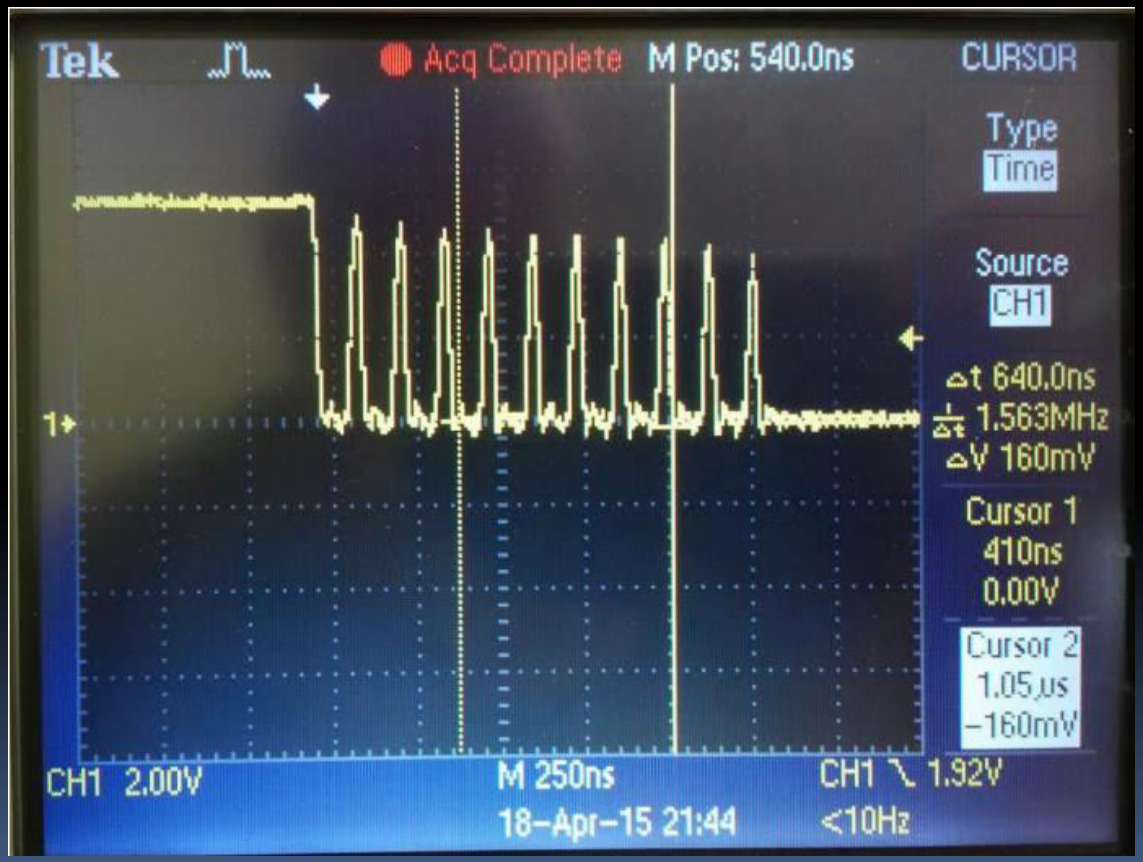
Physical Unclonable Functions Based Secret Keys Scheme for Securing Big Data Infrastructure Communication

Fadi Farha[a], Huansheng Ning[a,*], Hong Liu[b], Laurence T. Yang[c] and Liming Chen[d]

[a]*The School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China. 100083*
[b]*The School of Computer Science and Software Engineering, East China Normal University, China*
[c]*The Department of Computer Science, St. Francis Xavier University, Antigonish, NS, Canada.*
[d]*The School of Computer Science and Informatics, De Montfort University, Leicester, UK.*

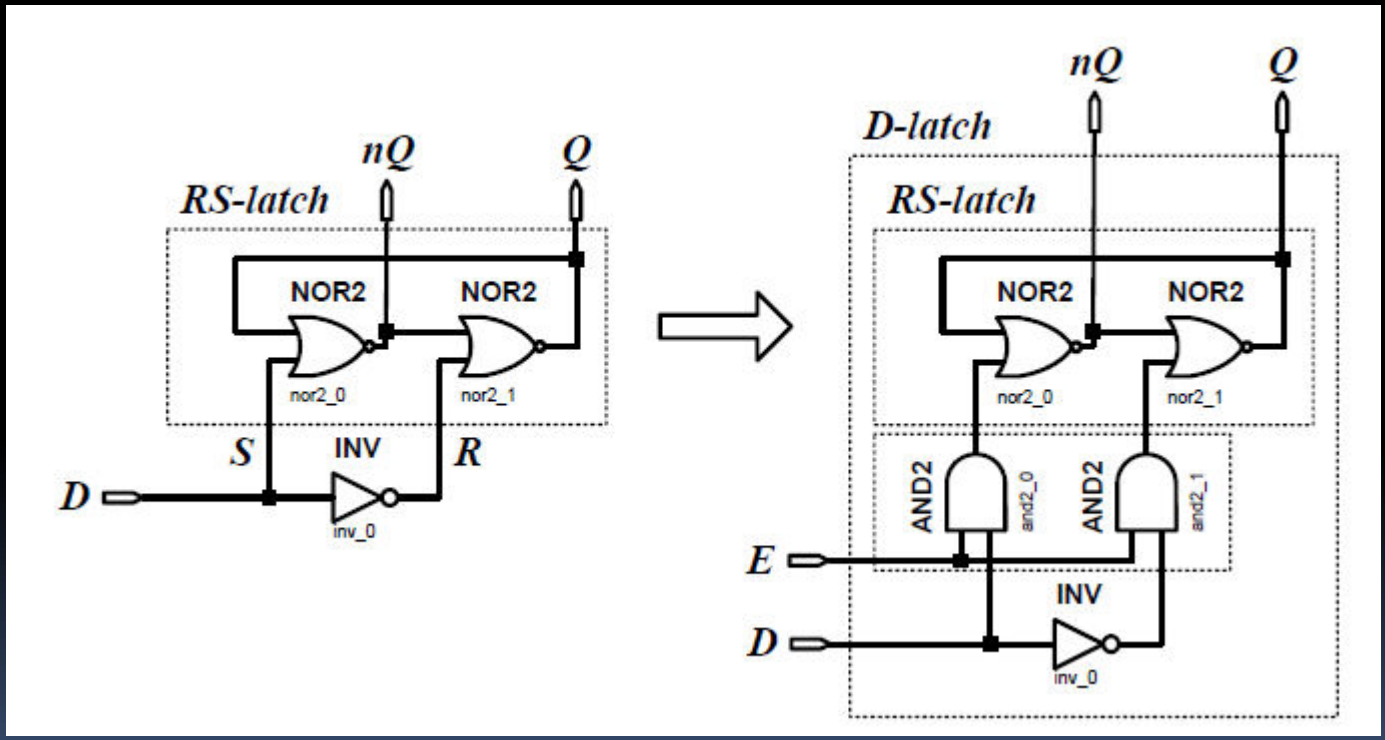
Introducing to VHDL

Sequential Logic: bi-stable elements



Introducing to VHDL

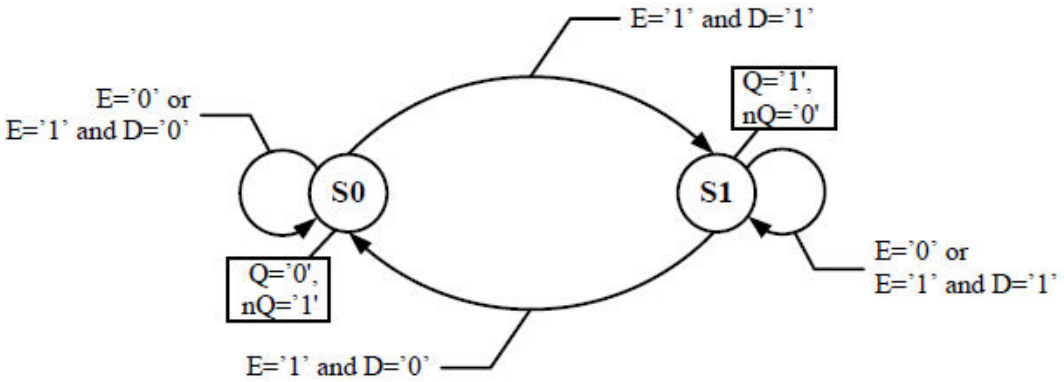
Sequential Logic: latches



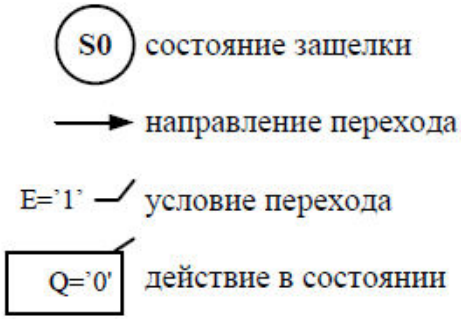
Introducing to VHDL

Sequential Logic: latches

E	*D*	*Q*(*t*)	*Q*(*t*+1)	*nQ*(*t*+1)
0	X	q	q	$\overline{q}$
1	0	q	0	1
1	1	q	1	0



X – безразличное значение
q={0,1} – значение состояния
***Q*(*t*)** – текущее значение на выходе
***Q*(*t*+1)** – следующее значение на выходе



Introducing to VHDL

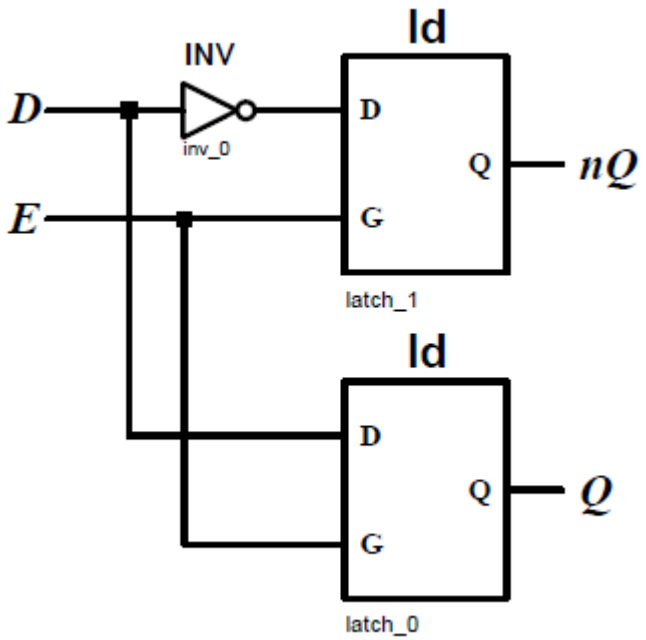
Sequential Logic: latches::D-Latch

```
1  Entity DLATCH is
2      Port ( D,E  : in  STD_LOGIC;
3             Q,nQ : out STD_LOGIC );
4  End DLATCH;
5
6  Architecture Beh of DLATCH is
7  Begin
8
9  Main: process( D, E )
10 begin
11   if E = '1' then
12     Q  <= D;
13     nQ <= not D;
14   end if;
15 end process;
16
17 End Beh;
```

Introducing to VHDL

Sequential Logic: latches::D-Latch

```
1  Entity DLATCH is
2      Port ( D,E  : in  STD_LOGIC;
3             Q,nQ : out STD_LOGIC );
4  End DLATCH;
5
6  Architecture Beh of DLATCH is
7  Begin
8
9  Main: process( D, E )
10 begin
11   if E = '1' then
12     Q  <= D;
13     nQ <= not D;
14   end if;
15 end process;
16
17 End Beh;
```



Introducing to VHDL

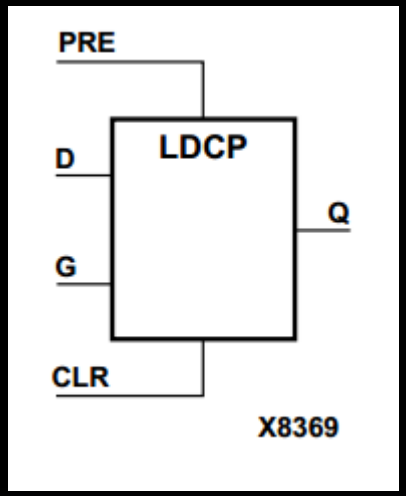
Sequential Logic: latches::D-Latch

```
1  Entity DLATCH is
2      Port ( D,E  : in  STD_LOGIC;
3             Q,nQ : out STD_LOGIC );
4  End DLATCH;
5
6  Architecture Beh of DLATCH is
7  Begin
8
9  Main: process( D, E )
10 begin
11   if E = '1' then
12     Q  <= D;
13     nQ <= not D;
14   end if;
15 end process;
16
17 End Beh;
```

```
-- ...
Architecture Beh of DLATCH is
-- Объект синтеза элемента памяти
signal q_t : std_logic;
Begin

-- Поведенческое описание элемента памяти
Main: process( D, E )
begin
        if E = '1' then
                q_t <= D;
        end if;
end process;

-- Передача хранящегося значения на выходной порт
Q  <= q_t;
-- Формирование парафазного значения
nQ <= not q_t;

End Beh;
-...
```

Introducing to VHDL

Sequential Logic: latches::real technological latches



Libraries Guide, Release 1.5

XILINX®

Таблица переходов RTL-примитива ldcp

CLR	PRE	G	D	$Q(t)$	$Q(t+1)$
1	X	X	X	X	0
0	1	X	X	X	1
0	0	1	d	X	d
0	0	0	X	q	q

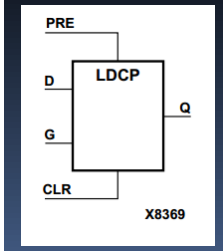
Introducing to VHDL

Sequential Logic: latches::real technological latches

Таблица переходов RTL-примитива ldcp

CLR	PRE	G	D	$Q(t)$	$Q(t+1)$
1	X	X	X	X	0
0	1	X	X	X	1
0	0	1	d	X	d
0	0	0	X	q	q

```
-- ...
Architecture Beh of LDCP is
-- Объект синтеза элемента памяти
signal q_t : std_logic;
Begin

-- Поведенческое описание элемента памяти
Main: process( CLR, PRE, D, G )
begin
        if CLR = '1' then     -- сигнал с наивысшим приоритетом
                q_t <= '0';
        elsif PRE = '1' then -- если CLR='0'
                q_t <= '1';
        elsif G = '1' then    -- если CLR='0' и PRE='0'
                q_t <= D;
        end if;

end process;

-- Передача хранящегося значения на выходной порт
Q  <= q_t;

End Beh;
--...
```



Introducing to VHDL

Sequential Logic: latches::real technological latches

Таблица переходов RTL-примитива ldcp

CLR	PRE	G	D	$Q(t)$	$Q(t+1)$
1	X	X	X	X	0
0	1	X	X	X	1
0	0	1	d	X	d
0	0	0	X	q	q

```
-- ...
        if PRE = '1' then      -- сигнал с наивысшим приоритетом
            q_t <= '1';
        elsif CLR = '1' then   -- если PRE='0'
            q_t <= '0';
        elsif G = '1' then     -- если PRE='0' и CLR='0'
            q_t <= D;
        end if;
-- ...
```