

Белорусский государственный университет
информатики и радиоэлектроники

4-ый курс специальности ПОИТ

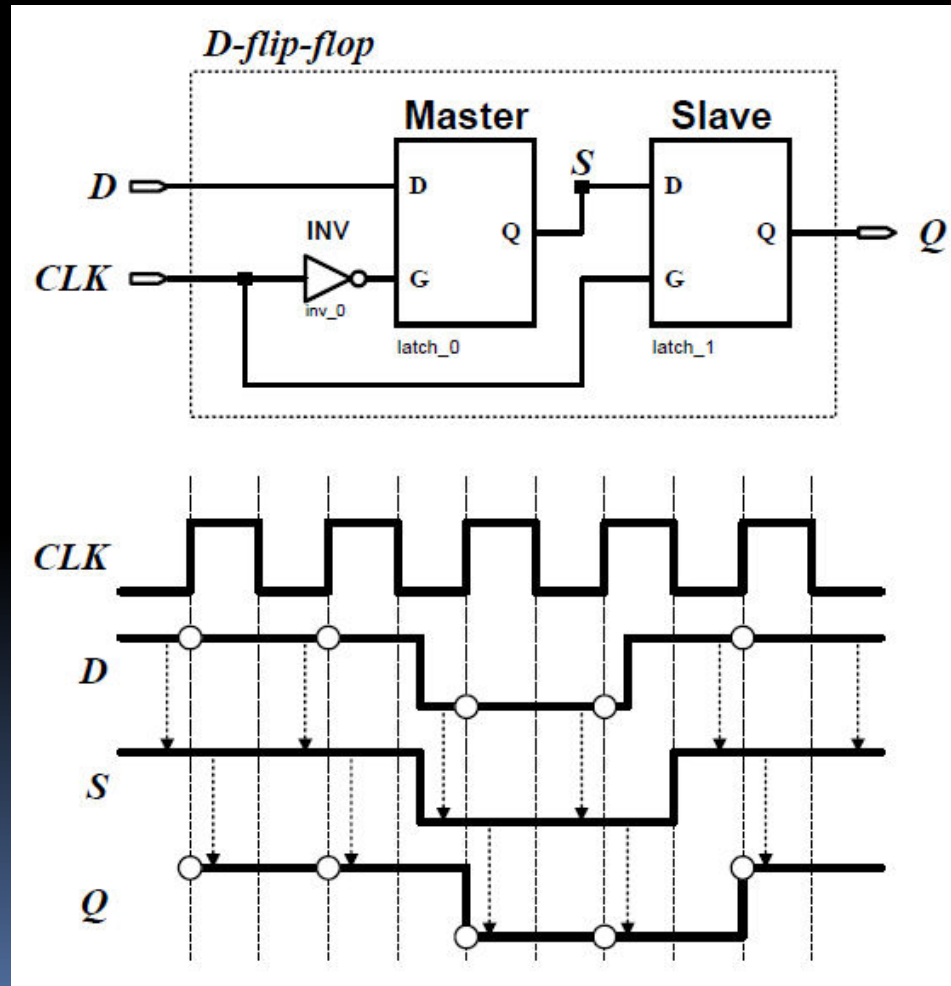
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЦИФРОВОГО ПРОЕКТИРОВАНИЯ



2020

Introducing to VHDL

Sequential Logic: flip-flops (FF)



Introducing to VHDL

Sequential Logic: flip-flops (FF)

Rising edge
condition

```
1 Entity DFF is
2   Port ( — входной порт данных
3         D   : in  STD_LOGIC;
4         — входной порт сигнала синхронизации
5         CLK : in  STD_LOGIC;
6         — выходной порт данных
7         Q   : out STD_LOGIC );
8 End DFF;
9
10 Architecture Beh of DFF is
11   — объект хранения значения сигнала входного порта данных
12   signal s : std_logic;
13   Begin
14
15   Main: process( D, CLK )
16   begin
17     — условие наступления переднего фронта сигнала CLK
18     if ( CLK'event and CLK = '1' ) then
19       — захват значения сигнала порта D
20       s <= D;
21       — при всех прочих условиях s осуществляет хранение
22       — захваченного значения сигнала порта D
23     end if;
24   end process;
25
26   — передача значения сигнала s на выходной порт Q
27   Q <= s;
28
29 End Beh;
```

Introducing to VHDL

Sequential Logic: flip-flops (FF)

Falling edge
condition

```
-- ...  
-- условие наступления заднего фронта сигнала CLK  
if ( CLK'event and CLK = '0' ) then  
    -- захват значения сигнала порта D  
    s <= D;  
    -- при всех прочих условиях s осуществляет хранение  
    -- захваченного значения сигнала порта D  
end if;  
-- ...
```

```
-- ...  
-- условие наступления переднего фронта сигнала CLK  
if ( CLK'event and CLK = '1' and CLK'last_value = '0' ) then  
-- ...
```

```
-- ...  
-- условие наступления переднего фронта сигнала CLK  
if rising_edge( CLK ) then  
-- ...
```

Introducing to VHDL

Sequential Logic: flip-flops (FF) :: real technological FF

fdcpe

D Flip-Flop with Clock Enable and Asynchronous Preset and Clear)

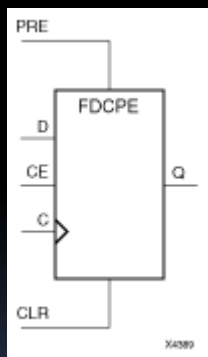


Таблица переходов триггера fdcpe

<i>CLR</i>	<i>PRE</i>	<i>CE</i>	<i>D</i>	<i>C</i>	<i>Q(t)</i>	<i>Q(t + 1)</i>
1	X	X	X	X	X	0
0	1	X	X	X	X	1
0	0	0	X	X	q	q
0	0	1	d	↑	X	d

Introducing to VHDL

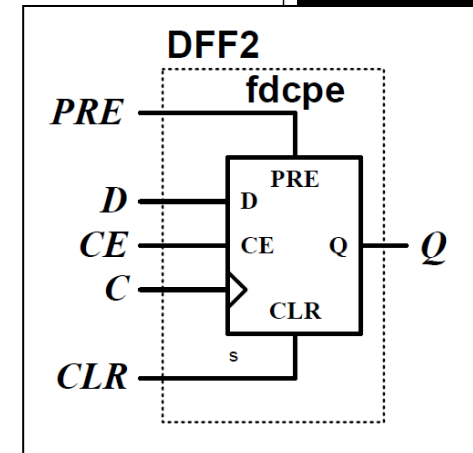
Sequential Logic: flip-flops (FF) :: real technological FF :: **fdcpe**

```
1  Entity DFF2 is  
2      Port ( — Входной порт сигнала асинхронного сброса  
3          CLR : in  STD_LOGIC;  
4          — Входной порт сигнала асинхронной установки  
5          PRE : in  STD_LOGIC;  
6          — Входной порт сигнала разрешения синхронизации  
7          CE  : in  STD_LOGIC;  
8          — Входной порт сигнала синхронизации  
9          C   : in  STD_LOGIC;  
10         — Входной порт данных  
11         D   : in  STD_LOGIC;  
12         — Выходной порт данных  
13         Q   : out STD_LOGIC );  
14 End DFF2;  
15
```

Introducing to VHDL

Sequential Logic: flip-flops (FF) :: real technological FF :: **fdcpe**

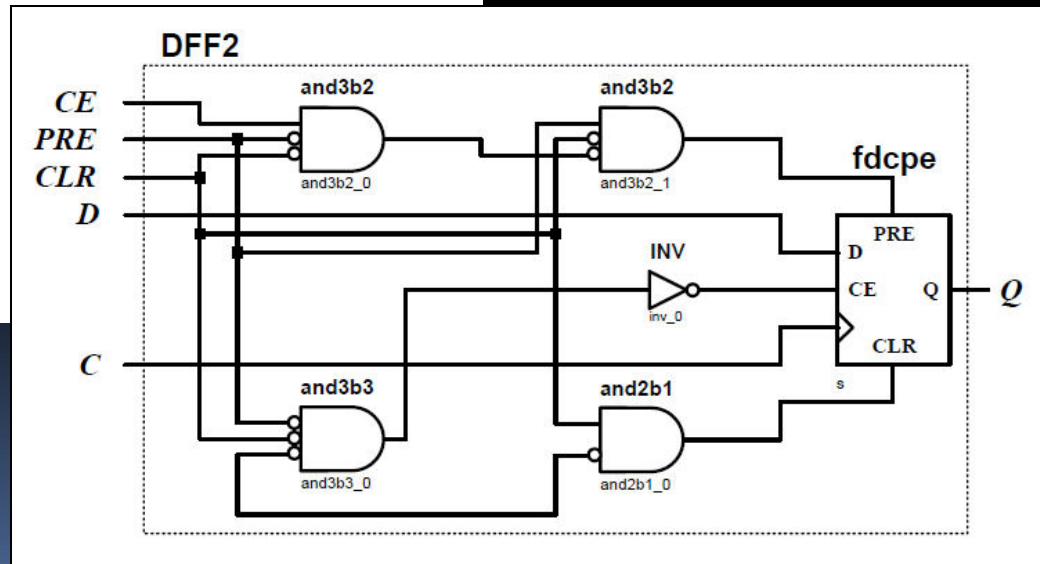
```
16 Architecture Beh of DFF2 is
17   — объект хранения значения сигнала Входного порта данных
18   signal s : std_logic;
19   Begin
20
21   Main: process( CLR, PRE, CE, C, D )
22   begin
23     — условие асинхронного сброса
24     if CLR = '1' then
25       s <= '0';
26     — условие асинхронной установки
27     elsif PRE = '1' then
28       s <= '1';
29     — условие асинхронного разрешения
30     elsif CE = '1' then
31       — условие наступления переднего фронта синхросигнала
32       if rising_edge( C ) then
33         s <= D;
34       — при всех прочих условиях s осуществляет хранение
35       — захваченного значения сигнала порта D
36       end if;
37     end if;
38   end process;
39
40   — передача значения сигнала s на выходной порт Q
41   Q <= s;
42
43 End Beh;
```



Introducing to VHDL

Sequential Logic: flip-flops (FF) :: real technological FF :: **fdcpe**

```
— ...  
Main: process( CLR, PRE, CE, C, D )  
begin  
    if ( CLR = '0' and PRE = '0' and CE = '1' ) then  
        if rising_edge( C ) then  
            s <= D;  
        end if;  
    elsif CLR = '1' then  
        s <= '0';  
    elsif PRE = '1' then  
        s <= '1';  
    end if;  
end process;  
  
Q <= s;  
— ...
```



Introducing to VHDL

Sequential Logic: flip-flops (FF) :: **ffc** (Toggle Flip-Flop with Asynchronous Clear)

There is no
RTL-primitive

```
-- ...
-- CLR -- Входной порт сигнала асинхронного сброса
-- C    -- Входной порт сигнала синхронизации
-- T    -- Входной порт управляющего сигнала
-- Q    -- Выходной порт данных
-- ...
signal s : std_logic;
Begin

Main: process( CLR, C, T, s )
begin
    if CLR = '1' then
        s <= '0';
    elsif T = '1' then
        if rising_edge(C) then
            s <= not s;
        end if;
    end if;
end process;

Q <= s;
-- ...
```

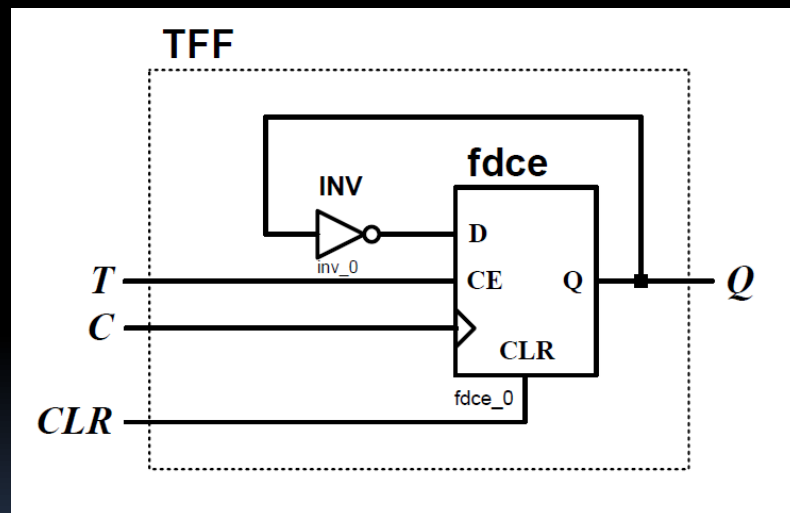
Introducing to VHDL

Sequential Logic: flip-flops (FF) :: **ftc** on the base of fdce

```
— ...
— CLR — входной порт сигнала асинхронного сброса
— C   — входной порт сигнала синхронизации
— T   — входной порт управляющего сигнала
— Q   — выходной порт данных
— ...
signal s : std_logic;
Begin

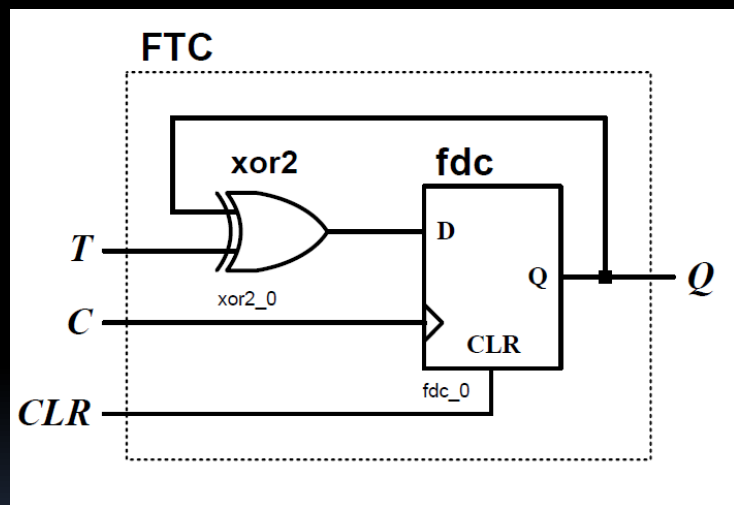
Main: process( CLR, C, T, s )
begin
    if CLR = '1' then
        s <= '0';
    elsif T = '1' then
        if rising_edge(C) then
            s <= not s;
        end if;
    end if;
end process;

Q <= s;
— ...
```



Introducing to VHDL

Sequential Logic: flip-flops (FF) :: **ftc** on the base of **fdc**



```
-- ...
signal s : std_logic;
Begin

Main: process( CLR, C, T, s )
begin
    if CLR = '1' then
        s <= '0';
    elsif rising_edge(C) then
        s <= s xor T;
    end if;
end process;

Q <= s;
-- ...
```

Introducing to VHDL

Sequential Logic: flip-flops (FF) :: **ftc** on the base of **fdc**

Combinational & Sequential Logic Description

```
-- ...  
-- сигнал, описывающий значение выходного порта триггера  
signal s : std_logic;  
-- сигнал, описывающий значение выхода комбинационной схемы  
signal tx : std_logic;
```

Begin

```
-- описание комбинационной схемы  
tx <= s xor T;  
  
-- описание элемента памяти (D-триггер)  
DFF: process( CLR, C, tx )  
begin  
    if CLR = '1' then  
        s <= '0';  
    elsif rising_edge( C ) then  
        s <= tx;  
    end if;  
  
end process;
```

```
-- Выходной комбинационный каскад  
Q <= s;
```

```
-- ...
```

```
-- ...  
-- Выходной трехстабильный буфер-усилитель  
Q <= s when OE='0' else 'Z';  
-- ...
```