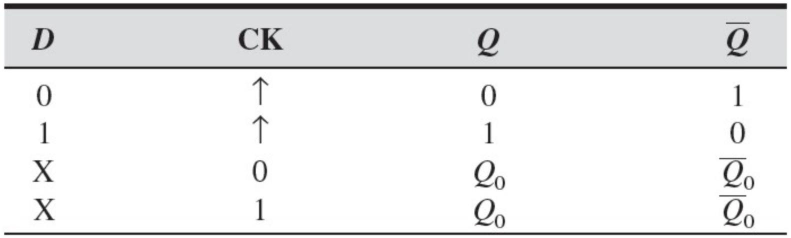
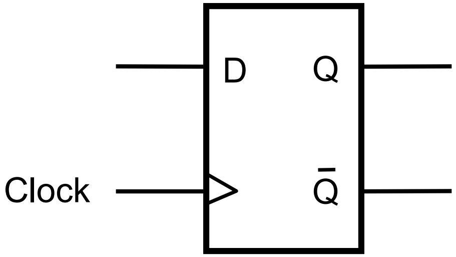
Lab 09: Flip-flops and counters

- Design an edge–triggered D flip-flop with Quartus II library and VHDL



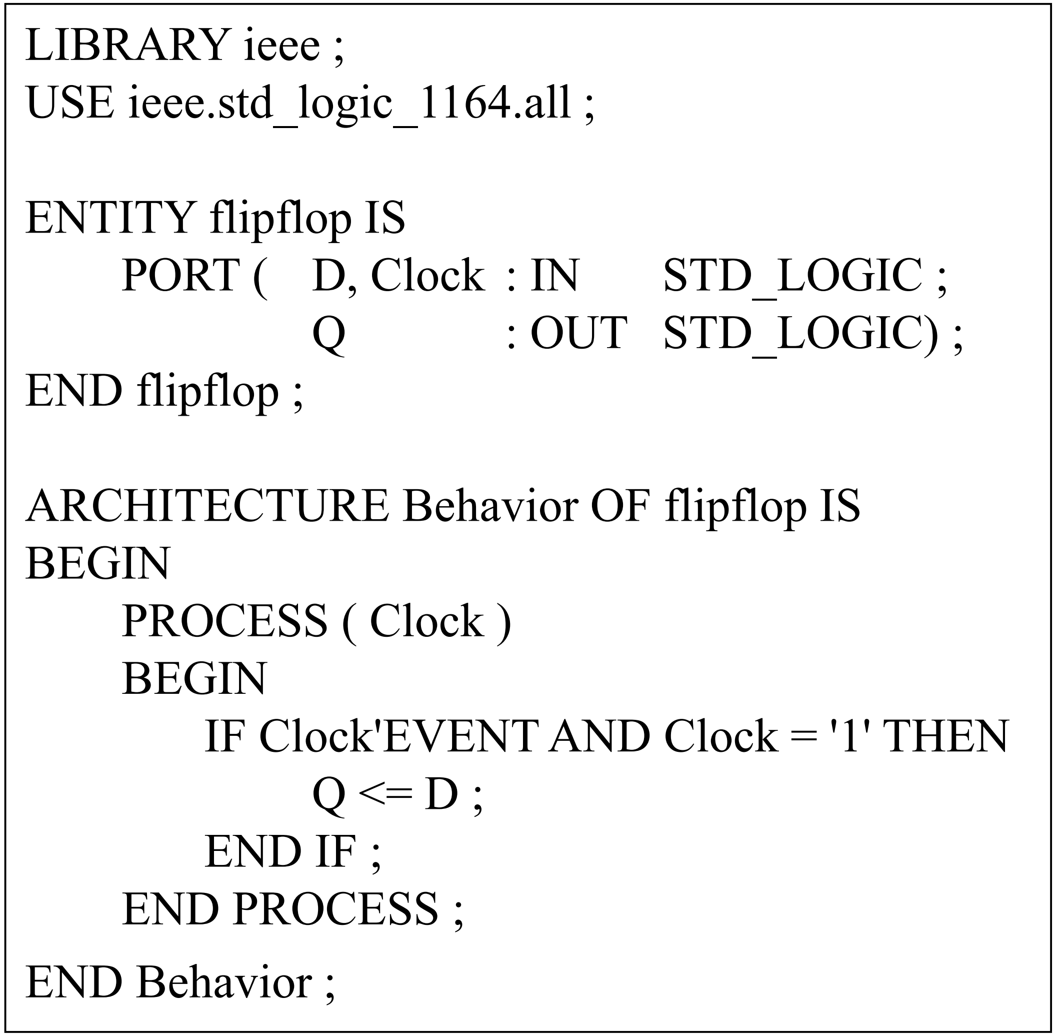
D	CK	Q	$\overline{Q}$
0	↑	0	1
1	↑	1	0
X	0	Q_0	$\overline{Q}_0$
X	1	Q_0	$\overline{Q}_0$



Graphical symbol

D flip-flop

- Process module
- Attribute: Clock’EVENT
- Postive edge: Clock'EVENT AND Clock = '1'

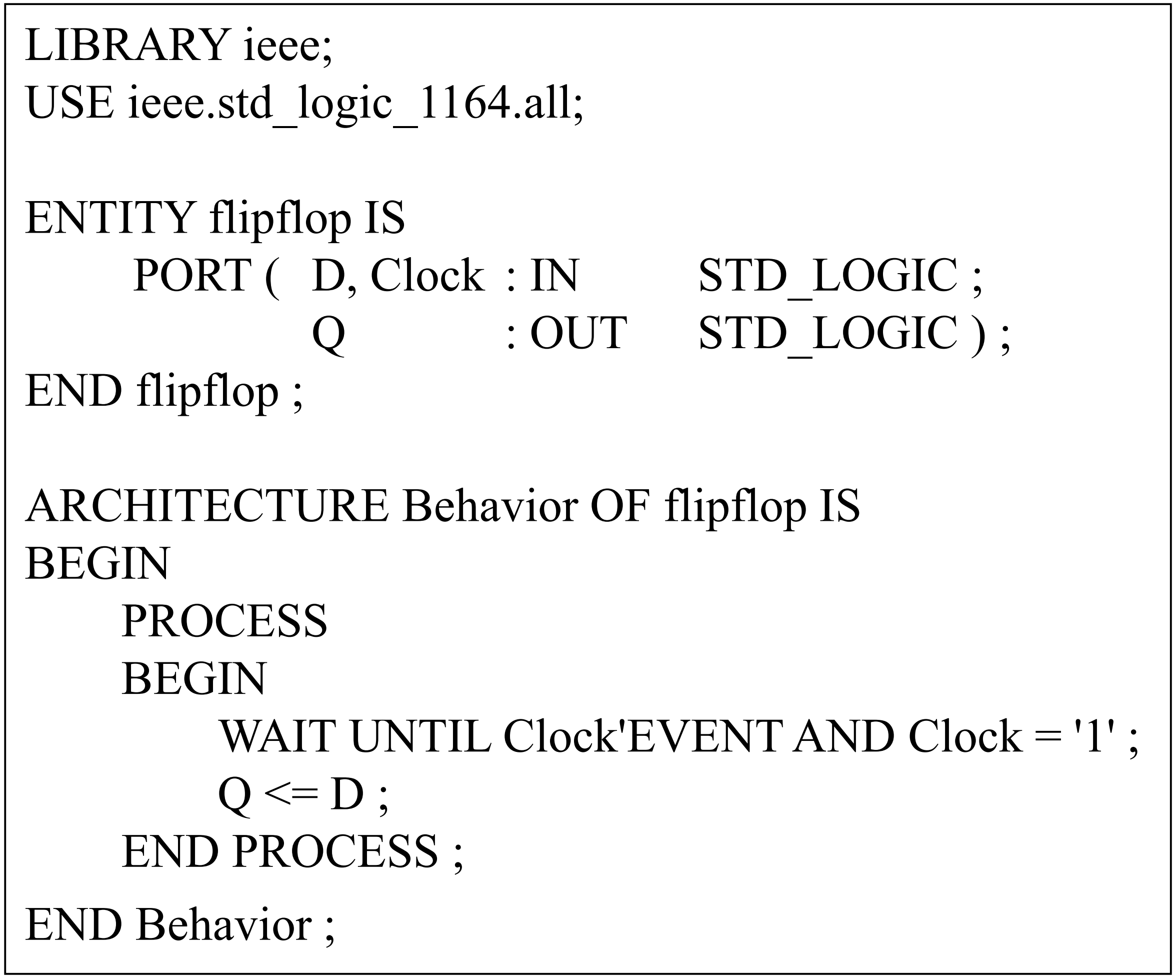


```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY flipflop IS
	PORT (	D, Clock	: IN	STD_LOGIC ;
		Q		: OUT	STD_LOGIC) ;
END flipflop ;

ARCHITECTURE Behavior OF flipflop IS
BEGIN
	PROCESS ( Clock )
	BEGIN
		IF Clock'EVENT AND Clock = '1' THEN
			Q <= D ;
		END IF ;
	END PROCESS ;
END Behavior ;
```

Figure 7.37. D flip-flop.

D flip-flop: WAIT-UNTIL

- Process module with no sensitivity list
- WAIT-UNITL



```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY flipflop IS
        PORT (  D, Clock  : IN        STD_LOGIC ;
                Q         : OUT       STD_LOGIC ) ;
END flipflop ;

ARCHITECTURE Behavior OF flipflop IS
BEGIN
        PROCESS
        BEGIN
                WAIT UNTIL Clock'EVENT AND Clock = '1' ;
                Q <= D ;
        END PROCESS ;
END Behavior ;
```

Figure 7.38. D flip-flop using WAIT-UNTIL statement.

Design C

- Modulo-10 counter with CLEAR, CLK, and En
- Synchronous counter with asynchronous CLEAR
- Sample VHDL code: next page

Modulo-10 up-counter

- Asynchronous reset
- “Enable” input

```
library    ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity counter10 is
port(Clk: in std_logic; Clear: in std_logic;Enable   : in std_logic;
    Q : out std_logic_vector(3 downto 0));
end counter10;

architecture RTL of counter10 is
     signal Count: std_logic_vector(3 downto 0);
begin
counter10: process(Clk,Clear)
begin
  if (Clear = '0')then
    Count <= (others => '0');
  elsif(Clk'event and Clk = '1')then
     if(Enable = '1')then
      if (Count = "1001")then
        Count <= (others => '0');
      else
        Count <= Count + 1;
      end if;
    end if;
  end if;
end process counter10;
Q <= Count;
end RTL;
```