

Лабораторная работа № 3. Исследование работы триггера.

Цель работы: изучение принципов функционирования простейшего конечного автомата.

Постановка задачи. На примере простейшей ячейки асинхронного RS-триггера и двух синхронных триггеров, D-триггера и JK-триггера, изучить назначение входов и принципы функционирования устройства. Для этого в пакете **Quartus15** записать программы функционирования устройств на языке VHDL, а затем построить временные диаграммы. **Рекомендуемые интервалы времени:**

RS-триггер: s – 80 нс с начальным уровнем 1, r – 50нс с начальным уровнем 0.

D – триггер: для сигналов синхронизации – 20нс, информационного D-входа – 90нс с начальным уровнем 1, для s – входа при базовом уровне 1 короткий 0 от 110 до 130 нс, для r – входа при базовом уровне 1 короткий 0 от 270 до 290 нс.

JK – триггер: для сигналов синхронизации – 20нс, информационных JK-входов J– 110нс с начальным уровнем 1 и K - 70 нс с начальным уровнем 0, , для s – входа при базовом уровне 1 короткий 0 от 150 до 170 нс, для r – входа при базовом уровне 1 короткий 0 от 30 до 50 нс.

.

Порядок выполнения работы.

1. Загрузить **Quartus15**, создать проект.
 2. Открыть VHDL файл, записать программу 3.1, отражающую работу асинхронного RS- триггера.
 3. Сохранить файл, установить его старшим в иерархии и откомпилировать.
 4. Открыть окно симулятора. Построить диаграммы работы. Интервалы выбрать из списка рекомендованных. Зарисовать диаграммы в отчет.
 5. Открыть новый VHDL файл, записать программу 3.2, отражающую работу **D – триггера**.
 6. Сохранить файл под другим именем, установить его старшим в иерархии и откомпилировать.
 7. Открыть окно симулятора. Построить диаграммы работы. Интервалы выбрать из списка рекомендованных. Зарисовать диаграммы в отчет.
 8. Открыть новый VHDL файл, записать программу 3.3, отражающую работу **JK – триггера**.
 9. Дать имя файлу и сохранить его, установить старшим в иерархии и откомпилировать.
 10. Открыть окно симулятора. Построить диаграммы работы . Интервалы выбрать из списка рекомендованных. Зарисовать диаграммы в отчет.
- При получении временных диаграмм для каждого типа триггера результаты демонстрировать преподавателю.**

Отчет должен содержать программы работы устройств , подкрепленные проверкой на временной диаграмме.

Программа 3.1

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

entity trs is
generic (delay:time:=2 ns);
port(
r,s: in std_logic;
q,nq: out std_logic
);
end trs;
architecture arch of trs is
signal p,np: std_logic;
begin
process (r,s)
begin
if r='0' and s='0' then p<='1' after delay;
                        np<='1' after delay;
elseif r='0' and s='1' then p<='0' after 2*delay;
                        np<='1' after delay;
elseif r='1' and s='0' then p<='1' after delay;
                        np<='0' after 2*delay;
elseif r='1' and s='1' then p<='X'; np<='X';
end if;
end process;
q<=p;
nq<=np;
end arch;
```

Программа 3.2

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

entity tr is
port(
data_i: in std_logic;
clk_i: in std_logic;
s : in std_logic;
r : in std_logic;
```

```

data_out: out std_logic
);
end tr;
architecture arch of tr is
begin
  process (clk_i,s,r)
  begin
    if r='0' then
      data_out<='0';
    elsif s='0' then
      data_out<='1';
    elsif(rising_edge(clk_i)) then
      data_out <= data_i;
    end if;
  end process;
end arch;

```

Программа 3.3

```

library ieee;
use ieee.std_logic_1164.all;

entity trjk2 is
port(
  clk,s,r,j,k: in std_logic;
  data_out: out std_logic
);
end trjk2;
architecture arch of trjk2 is
  signal q: std_logic;
begin
  process (clk,r,s)
  begin
    if r='0' then
      q<='0';
    elsif s='0' then
      q<='1';
    elsif(rising_edge(clk)) then
      if(j='1') then
        if(k='0') then q<='1';
        elsif(k='1')then q<=(not(q));
        end if;
      end if;
    end if;
  end process;
end arch;

```

```
    elsif (j='0') then
        if(k='0') then q<=(q or '0');
        elsif (k='1') then q<='0';
        end if;
    end if;
end if;
end process;
data_out<=q;
end arch;
```