

Регистры.

Регистром называется функциональный узел, предназначенный для приема, хранения, преобразования и выдачи многоразрядных двоичных чисел.

Регистры, выпускаемые промышленностью, разделяются на:

- параллельные (статические) регистры. Запись и считывание информации происходит в параллельном коде, слова в таких регистрах могут быть подвергнуты поразрядным логическим операциям;

- последовательные (сдвигающие) регистры. Запись и считывание информации происходит в последовательном коде, поразрядный сдвиг осуществляется или влево, или вправо. В реверсивных регистрах возможен сдвиг в обоих направлениях;

- последовательно-параллельные (универсальные) регистры. Запись и считывание информации проводится как в параллельном, так и в последовательном коде, возможен сдвиг в обоих направлениях;

4.3.1. Параллельные регистры.

Условное графическое отображение параллельного регистра на D-триггерах представлено на рис.4.16.

Обозначение входов и выходов:

- информационные входы D_i ;
- вход сигнала управления C ;
- вход сброса R ;
- информационные выходы Q_i ;.

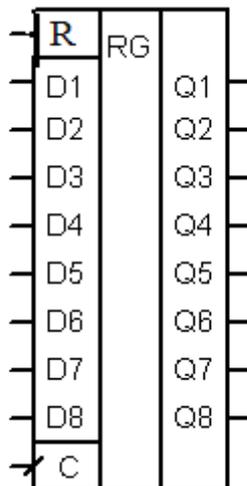


Рис. 4.16. Параллельный регистр на D-триггерах с синхронным сбросом.

VHDL-описание такого регистра, предполагающее, что сброс синхронный, т.е. происходит при приходе фронта синхросигнала C , дано ниже.

library IEEE; [2][1]

```

use IEEE.std_logic_1164.all;
entity Reg_1 is
    port ( R : in std_logic;
          D : in std_logic_vector(7 downto 0);
          C : in std_logic;
          Y : out std_logic_vector(7 downto 0) );
end entity;

architecture BEH of Reg_1 is
    begin
    process (C)
    begin
        if (C'event and C = '1') then
            if R = '1' then
                Y <= (others => '0');
            else
                Y <= D;
            end if;
        end if;
    end process;
end;

```

На выходах триггеров регистра могут быть установлены выходные каскады с тремя состояниями. Схема и УГО 8-ми разрядного параллельного регистра с входами сброса R, синхронизации C и разрешения выхода EN, собранного на D-триггерах с динамическим управлением(аналог микросхемы ИР23), приведена на рис. 4.17.

Входное 8-ми разрядное число подается на входы D1 – D8, запись числа в регистр происходит по фронту синхроимпульса C. Активный сигнал на входе разрешения выхода EZ открывает выходные каскады с тремя состояниями, в результате чего записанное число появляется на выходы Q1 – Q8.

Практически по такой же схеме собран регистр типа ИР22, в котором используются D-триггера защелки с потенциальным управлением. Запись информации в этом регистре проводится по уровню синхросигнала C.

Динамические характеристики регистров по сути повторяют динамические характеристики входящих в них триггеров.

На входе регистра могут быть установлены дополнительные логические элементы, позволяющие реализовать различные логические функции, в частности ввод информации от нескольких источников.

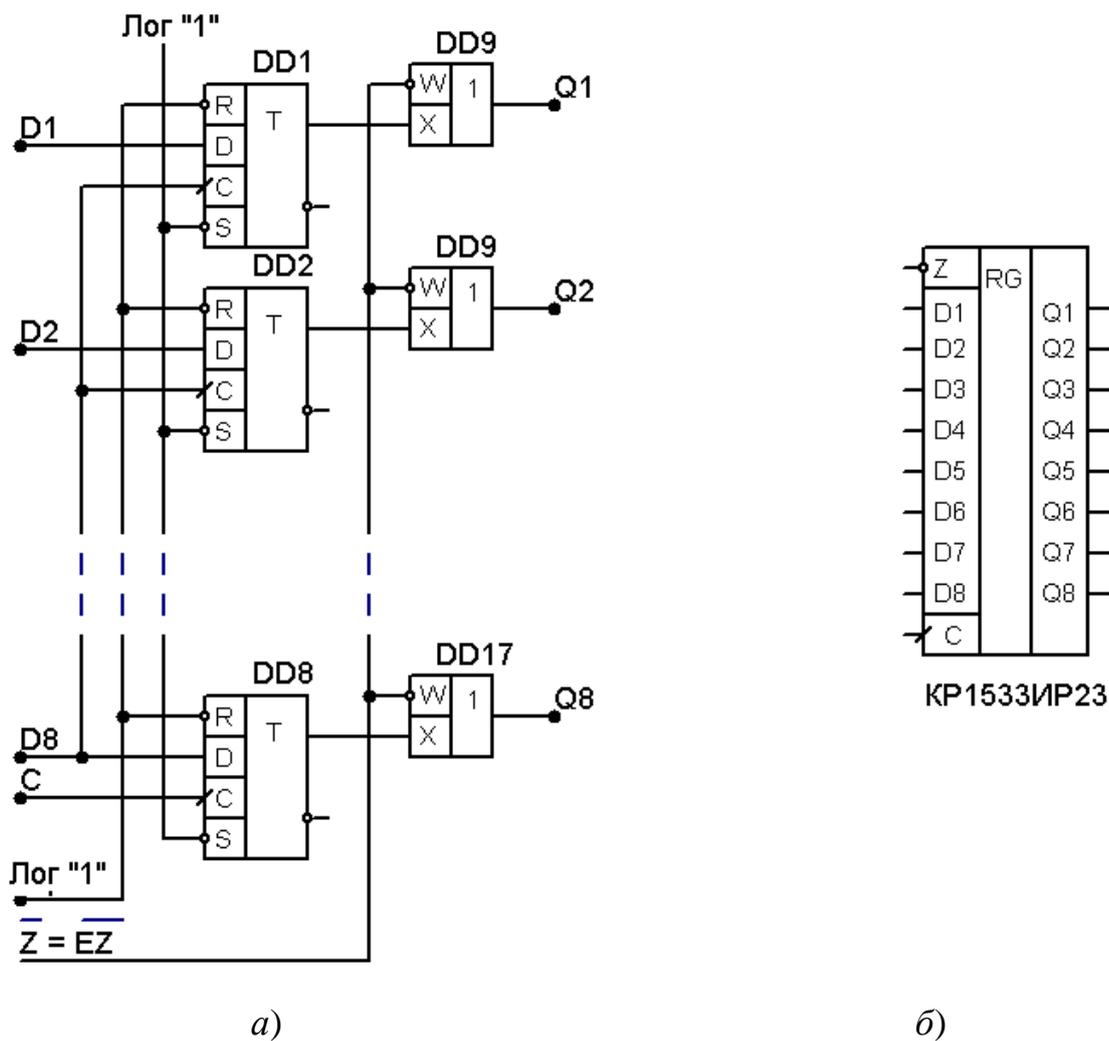


Рис. 4.17. Параллельный регистр с выходными каскадами с тремя состояниями.

4.3.2. Описание параллельного регистра с выходами с тремя состояниями на VHDL.

Описание восьмиразрядного регистра (nbgf ВН23) с синхронизацией по фронту синхросигнала С и выходами с тремя состояниями и управляющим асинхронным сигналом EZ.

```

library IEEE; use IEEE.std_logic_1164.all; [1][2]
entity Reg_2 is
    port ( EZ,C : in std_logic;
          D : in std_logic_vector(7 downto 0);
          Y : out std_logic_vector(7 downto 0) );
end entity;

```

```

architecture Reg_1_arc of Reg_2 is
    signal Y1: std_logic_vector(7 downto 0);
begin
    process (C)
    begin
        if (C'event and C = '1') then
            Y1 <= D;
        end if;
    end if;
    end process;

    Y <= Y1 when EZ='0' else 'Z';
end Reg_2_arc;

```

4.3.3. Регистры сдвига.

Последовательные или сдвигающие регистры содержат цепочку триггеров, связанных цепями переноса. В сдвигающих регистрах применяются или двухступенчатые триггеры, или триггеры с синхронизацией по фронту. D-триггеры с синхронизацией по уровню (защелки) в регистрах сдвига не применяются, потому что в таких регистрах за время действия синхроимпульса возможен проскок импульса сразу через несколько разрядов. Промышленность выпускает несколько типов регистров сдвига в интегральном исполнении, например, ИР9.

Схема регистра сдвига вправо (DSR) на D-триггерах с синхронизацией по фронту сигнала C приведена на рис. 4.18.

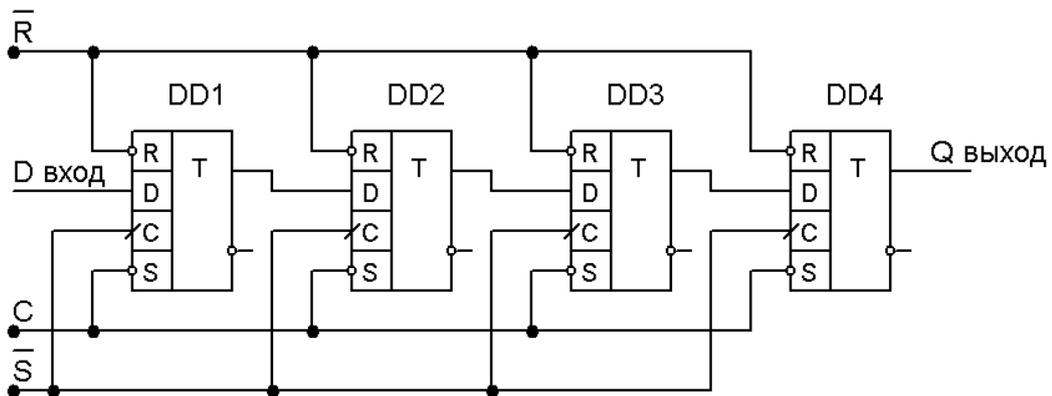


Рис. 4.18. Схема регистра сдвига вправо (DSR).

Информационный сигнал поступает на D вход левого триггера DD1 и по фронту импульса синхронизации записывается в этот триггер, оказываясь через время задержки триггера DD1 на D входе второго триггера DD2. Следующий импульс синхронизации сдвигает информацию еще на один разряд вправо.

Временная диаграмма работы регистра сдвига вправо приведена на рис. 4.19.

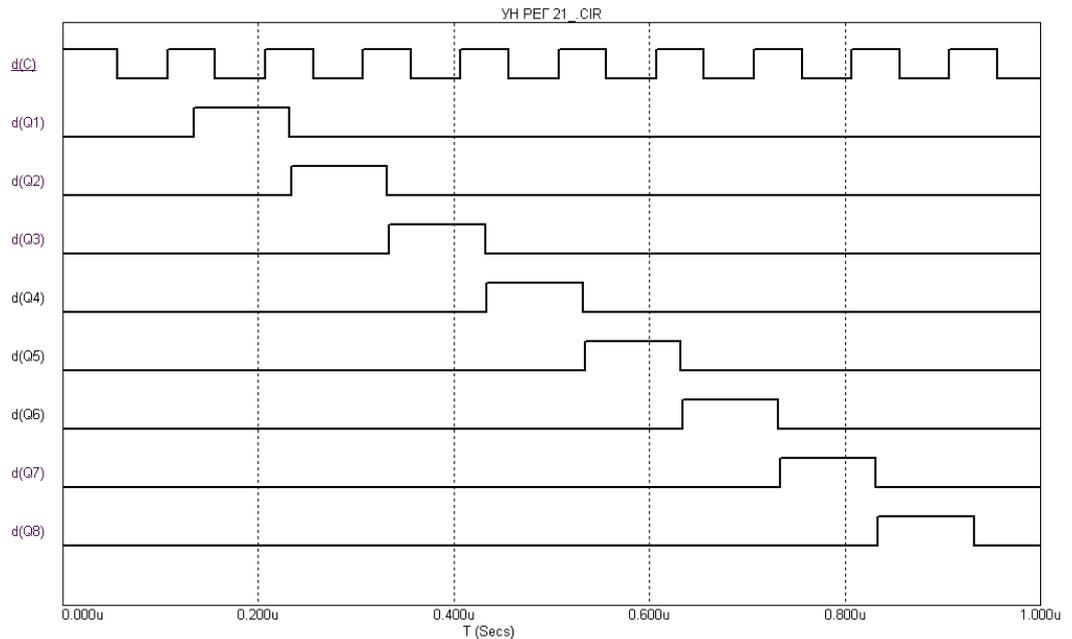


Рис. 4.19. Временная диаграмма работы регистра сдвига вправо.

Регистр сдвига влево (DSL) отличается только другой организацией связей между разрядами. Схема регистра приведена на рис. 4.20. На ней вход D находится справа, а выход Q – слева.

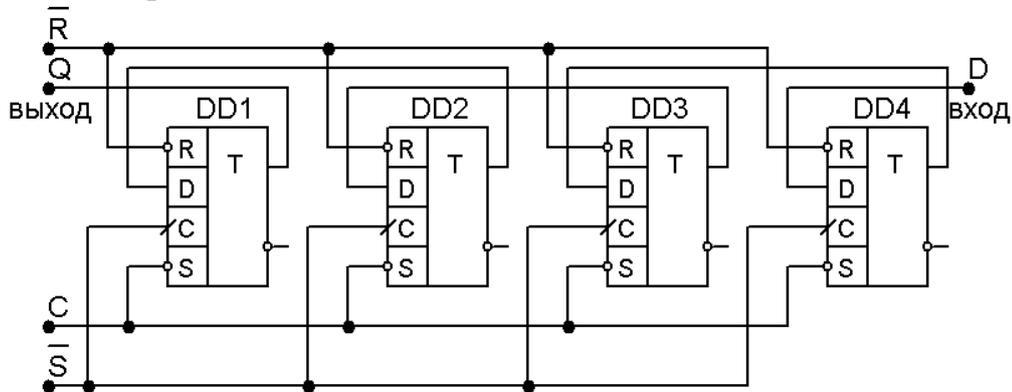


Рис. 4.20. Схема регистра сдвига влево (DSL).

Разработано большое количество схем универсальных регистров, в которых объединены функции параллельного регистра и регистра сдвига. Универсальный регистр выполняет следующие функции:

- Параллельную загрузку двоичного числа;
- Параллельную выгрузку двоичного числа, и
- Сдвиг числа влево или вправо.

Схема универсального регистра сдвига достаточно сложна. Поэтому приведем описание универсального регистра на VHDL, позволяющее достаточно просто понять алгоритм его работы.

4.3.4. Описание универсального регистра на VHDL.

УГО регистра сдвига с параллельной загрузкой по сигналу DL, последовательной загрузкой по сигналу S1 и асинхронным сбросом по сигналу R приведено на рис.4.23.

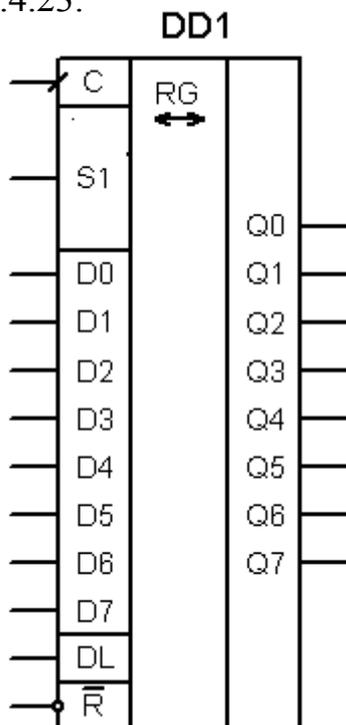


Рис.4.21. УГО регистра сдвига с параллельной загрузкой по команде DL, последовательной загрузкой по команде S1 и сбросом по R = 0.

```

library IEEE;
use IEEE.std_logic_1164.all;

entity shft_reg is
    port ( C : in std_logic; -- синхроимпульсы
          S1 : in std_logic; -- вход данных последовательный
          D : in std_logic_vector(7 downto 0); -- вход данных параллельный
          DL : in std_logic; -- параллельная загрузка данных
          R_n : in std_logic; -- асинхр. сброс
          Q : out std_logic_vector(7 downto 0)); --
end entity;

architecture shft_reg_arch of shft_reg is
    signal TEMP_Q : std_logic_vector(7 downto 0);
begin

    process(C, R_n)
    begin
        if R_n = '0' then
            TEMP_Q <= (others => '0');
        elsif (C'event and C = '1') then
            if DL = '1' then
                TEMP_Q <= D;
            else

```

```
                TEMP_Q <= SI & TEMP_Q(7 downto 1);
            end if;
        end if;
    end process;

    Q <= TEMP_Q;

end architecture;
```