

9. Глава 9. О необходимости тактирования схем ЦУ.

Последовательные и параллельные регистры

Рассматриваются риски при работе RS-триггера, а также анализируется во времени работа регистров сдвига, регистра памяти и универсального регистра.

9.1.Переключение RS-триггера на "ИЛИ-НЕ" во времени

Рассмотрим переключение RS-триггера на "ИЛИ-НЕ" (рис. 9.1) из состояния "0" в состояние "1", приняв во внимание, что любой ЛЭ (в том числе и "ИЛИ-НЕ") имеет время задержки распространения сигнала (T_p)

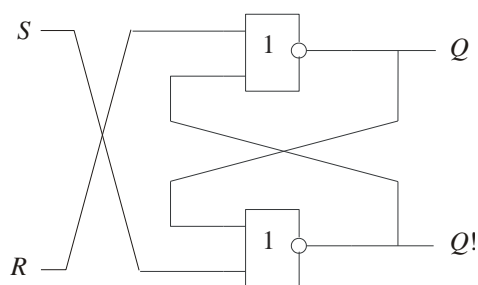


Рис. 9.1. Схема RS-триггера на элементах "ИЛИ-НЕ"

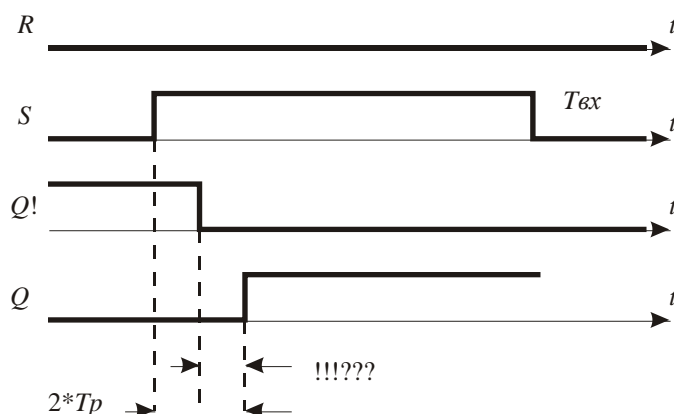


Рис. 9.2. Временная диаграмма RS-триггера

Из временного анализа (рис. 9.2) видно, что в ответ на входной сигнал триггер переключается с задержкой $2 \cdot T_p$, поэтому $T_{вх}$ должно быть больше $(3...4)T_p$.

На отрезке "!!!???": $Q = Q!$ — такого быть не должно, и если в этот момент прервать "1"-сигнал на Set, то триггер "растеряется". Ситуация, когда $Q = Q!$, называется *риском*.

Из временного анализа вытекает, что необходимо добиться такой работы триггера, при которой на отрезке времени "!!!???" информация последующими элементами не воспринималась бы, так как она *ложная*.

Для устранения возникающих трудностей подачу информации осуществляют от элемента к элементу с помощью синхронизирующих, тактирующих или стробирующих импульсов. В другие отрезки времени считывание информации запрещено.

Таким образом, продвижение информации по регистрам сдвига и все действия в вычислительных устройствах строго тактированы или синхронизированы.

9.2. Системы синхронизации на примере работы регистров сдвига (последовательных регистров)

Каждая из систем синхронизации соответствует различному воздействию импульсов синхронизации на логическое устройство (ЛУ). Различают одно-, двух- и более тактные (фазные) системы синхронизации.

Рассмотрим одно- и двухтактные системы синхронизации на примере соответствующих регистров сдвига. *Регистр сдвига* — это устройство, состоящее из нескольких разрядов, каждый из которых может находиться в одном из двух устойчивых состояний, отвечающих двум значениям логической переменной. Разряд регистра может хранить один бит информации.

Регистры сдвига строятся на одно- и двухтактных D-триггерах.

9.2.1. Схема однотоктного последовательного регистра

При передаче чего-либо, не обязательно информации в регистрах, на какое-то короткое время передающее звено (разряд регистра) должно хранить одновременно то, что у него было, и уже вновь поступившее. Поэтому разряд регистра должен содержать минимум два запоминающих элемента (ЗЭ). На рис. 9.3 и 9.4 представлены схемы однотоктных последовательных регистров сдвига на однотоктных и двухтактных D-триггерах, соответственно.

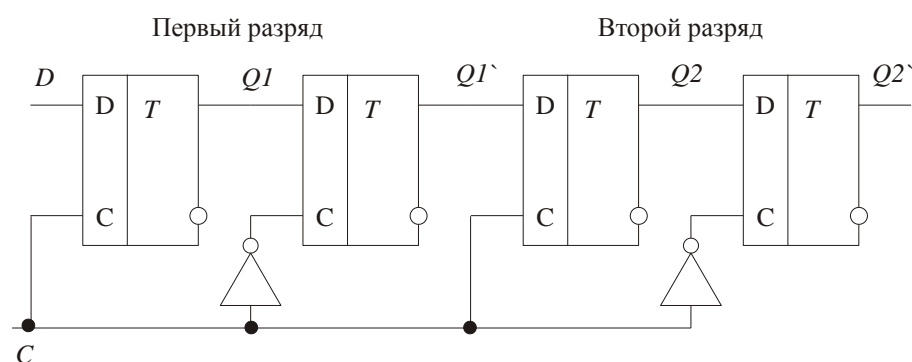


Рис. 9.3. Схема однотоктного последовательного регистра на однотоктных D-триггерах

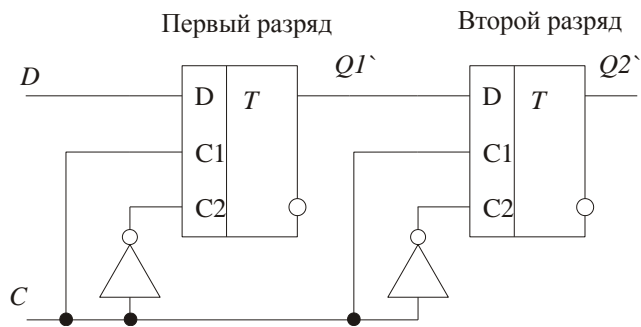


Рис. 9.4. Схема одноклапного последовательного регистра на двухтактных D-триггерах

Рассмотрим работу этого регистра во времени (рис. 9.5).

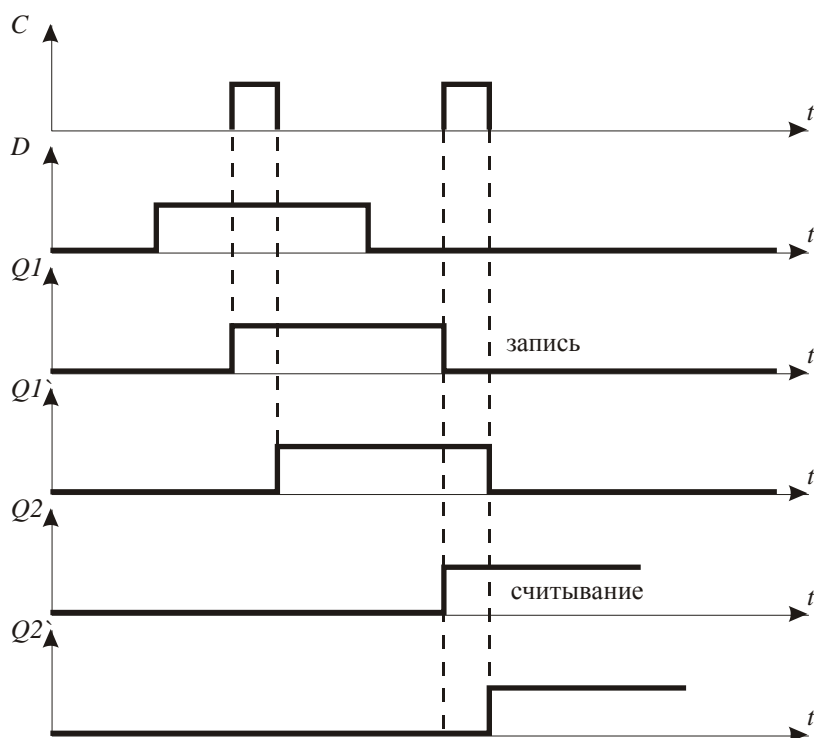


Рис. 9.5. Временная диаграмма работы одноклапного последовательного регистра

Главное в регистрах сдвига: обеспечить *одновременность* операций записи и считывания информации в каждый разряд регистра. В одноклаптных

последовательных регистрах сдвига эта *одновременность* обеспечивается за время тактового импульса.

9.2.2. Схема двухтактного последовательного регистра

В двухтактном последовательном регистре сдвига каждый разряд также содержит два ЗЭ, а передача информации от разряда к разряду происходит за два такта. На рис. 9.6 и 9.7 представлены схемы двухтактных последовательных регистров сдвига на одноктактных и двухтактных D-триггерах, соответственно.

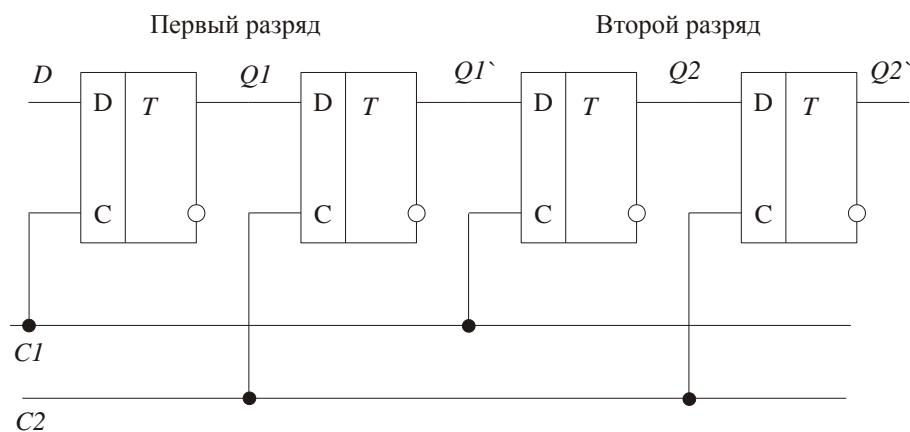


Рис. 9.6. Схема двухтактного последовательного регистра на одноктактных D-триггерах

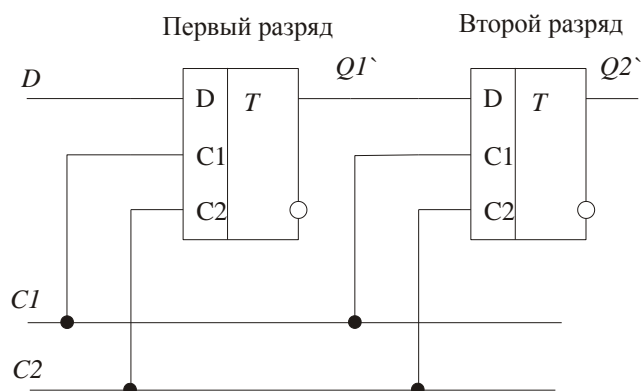


Рис. 9.7. Схема двухтактного последовательного регистра на двухтактных D-триггерах

Рассмотрим работу этого регистра во времени (рис. 9.8).

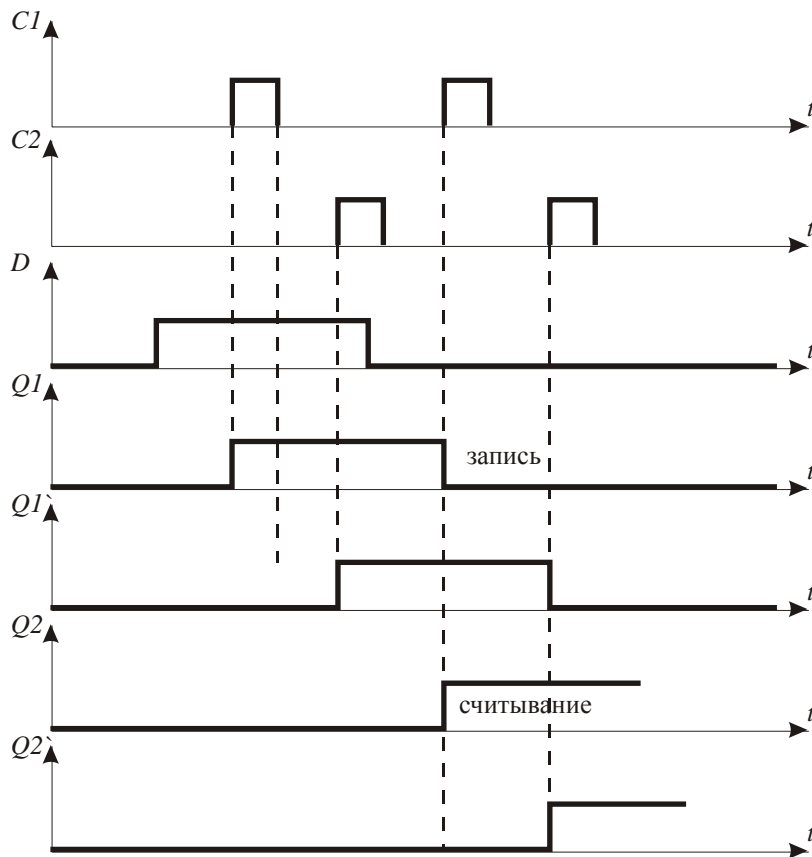


Рис. 9.8. Временная диаграмма работы двухтактного последовательного регистра

Здесь одновременные запись в первый D-триггер и считывание из второго D-триггера для одного и того же разряда разнесены между импульсами $C1$ и $C2$.

В одноклапном регистре сдвига, в отличие от двухклапного, одновременные запись и считывание в отдельный разряд регистра происходят за время тактового импульса, поэтому одноклапный регистр

менее устойчив к помехам (в нем больше вероятности потерять информацию из-за уменьшения длительности тактового импульса под действием помех).

Каждый разряд *многотактного* регистра содержит *m* последовательно включенных D-триггеров. Управление, здесь, производится пачками, каждая содержит *m* импульсов.

Последовательный регистр позволяет осуществлять преобразование последовательного кода в параллельный.

9.3. Параллельные регистры (регистры памяти)

Упрощенная (приблизительная) схема регистра памяти приведена на рис. 9.9.

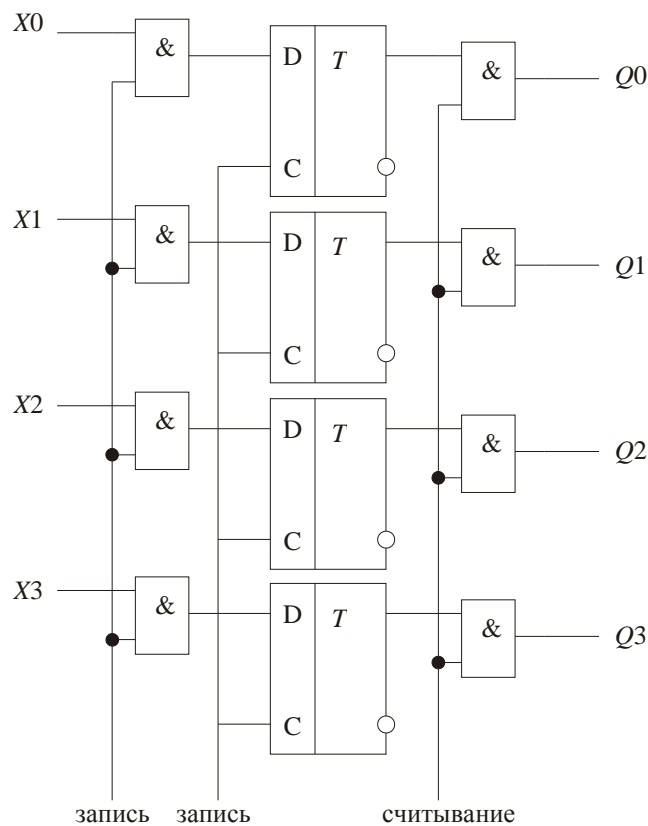


Рис. 9.9. Упрощенная схема регистра памяти

Функциональные возможности: прием числа в параллельном коде (через защелку), хранение и передача числа через защелку на выход.

9.4. Универсальные (параллельные, последовательные и реверсивные) регистры

Универсальные регистры (рис. 9.10) объединяют функциональные возможности последовательных и параллельных регистров. Они также могут получать информацию в параллельном коде и выдавать ее на выход в последовательном коде (преобразовывать параллельный код в последовательный и наоборот).

В этих регистрах возможен реверс (обратный ход) информации с выхода на вход. Реверсивные регистры сдвигают код числа в сторону старшего или младшего разрядов.

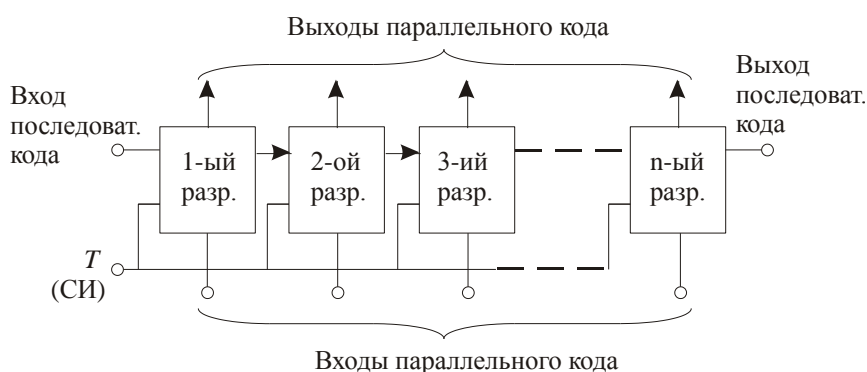


Рис. 9.10. Функциональная схема универсального регистра

Лабораторная работа. Исследование схем сдвигающего регистра, регистра памяти и универсального регистра

Цель работы

В данной лабораторной работе исследуется функционирование различных схем сдвигающих регистров и регистров памяти. Сдвигающие регистры осуществляют прием, хранение и передачу информации, а также сдвиг информации на требуемое число разрядов влево или вправо, преобразование последовательного кода в параллельный (и наоборот). Регистры памяти предназначены для хранения информации. Основу реализации сдвигающих и параллельных регистров составляют D-триггеры (элемент DFF из библиотеки примитивов).

Программа работы

1. Исследовать работу элемента библиотеки примитивов DFF.
2. Создать в графическом редакторе проект схемы двухразрядного сдвигающего регистра на D-триггерах (элемент DFF) с одним тактовым входом, откомпилировать и промоделировать его работу. Зарисовать временные диаграммы. Схема сдвигающего регистра приведена на рис. 9.11.

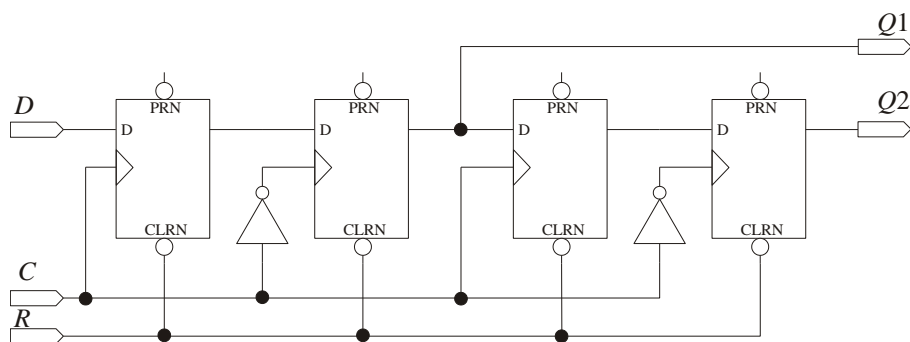


Рис. 9.11. Схема двухразрядного одноктакного сдвигающего регистра

3. Создать в графическом редакторе проект схемы D-триггера с двумя тактовыми входами на элементах DFF из библиотеки примитивов, откомпилировать и промоделировать его работу. Зарисовать временные диаграммы. Схема D-триггера приведена на рис. 9.12. Сохранить схему в виде символа.

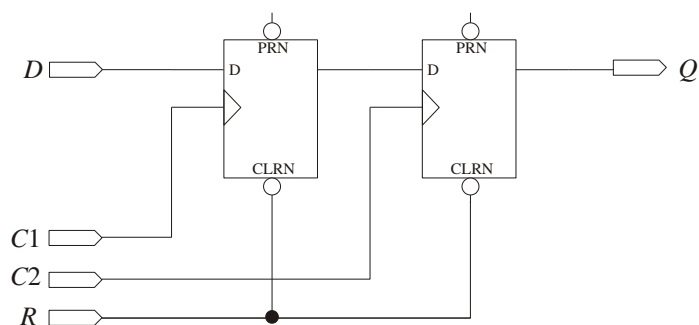


Рис. 9.12. Схема двухтактного D-триггера

4. Создать в графическом редакторе проекты схем двухразрядного сдвигающего регистра на D-триггерах с двумя тактовыми входами, откомпилировать и промоделировать работу этих схем. Зарисовать временные диаграммы. Схема сдвигающего регистра приведена на рис. 9.13.

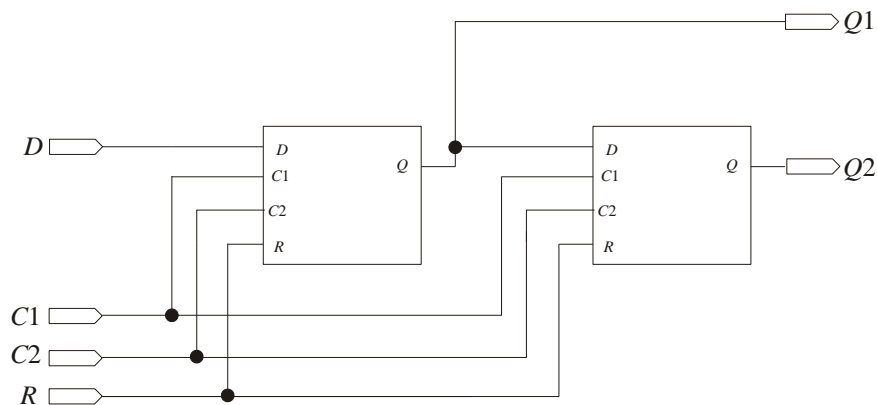


Рис. 9.13. Схема двухтактного двухразрядного сдвигающего регистра

5. Создать в графическом редакторе проект схемы четырехразрядного регистра памяти на D-триггерах (элемент DFF), откомпилировать и промоделировать его работу. Зарисовать временные диаграммы. Схема регистра памяти приведена на рис. 9.14.

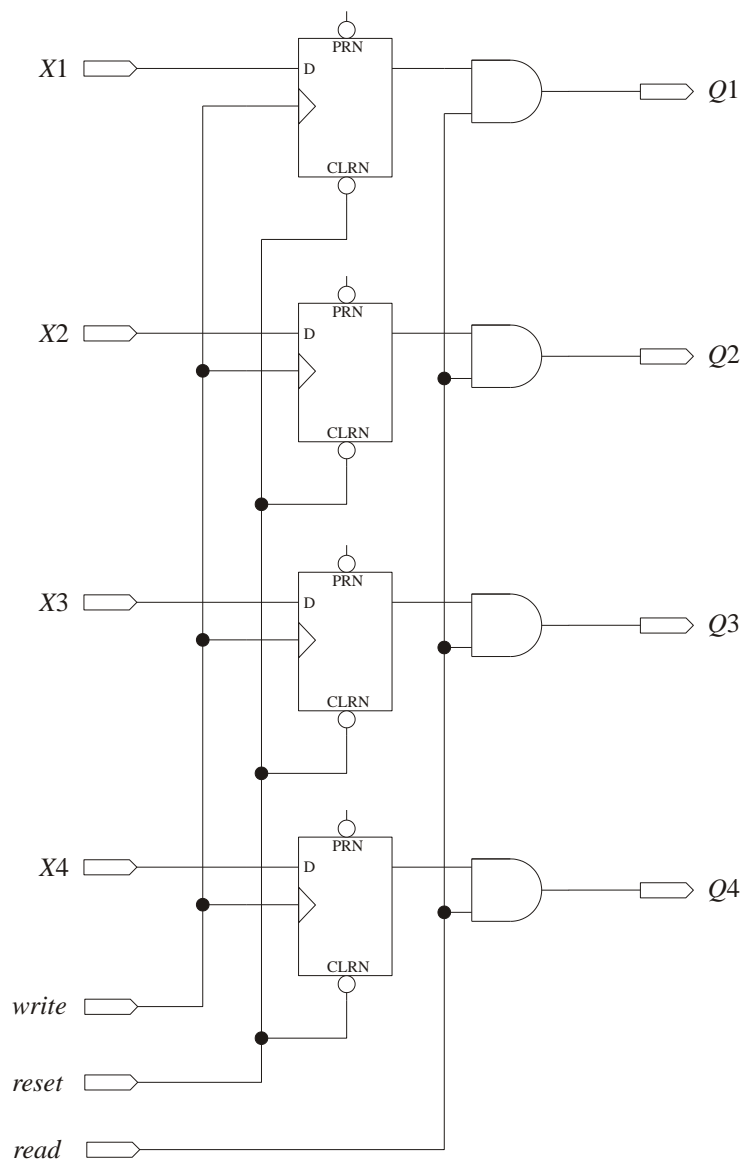


Рис. 9.14. Схема простейшего регистра памяти

6. Исследовать работу элемента 74395 библиотеки макрофункций mf

САПР фирмы Альтера Max+Plus II (рис. 9.15).

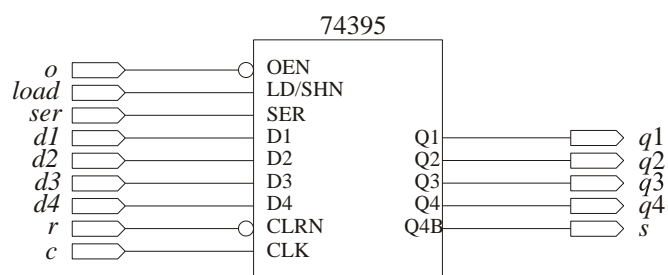


Рис. 9.15. Элемент 74395 (универсальный регистр)

Этот элемент представляет собой универсальный регистр, работающий и в режиме параллельной загрузки данных, и в режиме последовательного ввода и сдвига данных. Входы **d1**, **d2**, **d3**, **d4** используются для параллельного ввода информации, а вход **SER** — для последовательного. Вход **LD/SHN** управляет режимом работы регистра ("0" на этом входе означает, что разрешена последовательная загрузка данных, "1" — параллельная). Вход **CLRN** — установка нуля, вход **CLK** — разрешение записи (запись производится по спаду единичного сигнала).

Откомпилировать проект и получить временные диаграммы работы регистра в разных режимах.

7. Реализовать проект двухразрядного сдвигающего регистра с одним тактовым входом на языке программирования VHDL, используя поведенческую модель функционирования устройства.

Откомпилировать проект и получить временные диаграммы. Сравнить их с диаграммами, полученными в п.2.

8. Реализовать проект двухразрядного сдвигающего регистра с двумя тактовыми входами, основанного на двухтактных D-триггерах (рис. 9.13), на языке программирования VHDL, используя поведенческую модель функционирования устройства.

Откомпилировать проект и получить временные диаграммы. Сравнить их с диаграммами, полученными в п.4.

9. Реализовать проект регистра памяти на языке программирования VHDL, используя поведенческую модель функционирования устройства. Откомпилировать проект и получить временные диаграммы. Сравнить их с диаграммами, полученными в п.5.
10. Для проверки на синтезируемость написанных ранее VHDL-кодов рассмотрим синтезируемые VHDL-коды D-триггера с асинхронным сбросом и установкой (D-триггер срабатывает только при наличии соответствующего синхронизирующего импульса) и четырехразрядного регистра сдвига с асинхронным сбросом. При рассмотрении будем использовать встроенные средства САПР Quartus II фирмы Altera. Для описания триггеров, или регистров, обычно используются условия наступления события (event) прохождения во времени фронта или среза (спада) синхронизирующего импульса.

Примеры таких условий для VHDL:

- (clk'event and clk = '1') — условие фронта синхроимпульса;
- (clk'event and clk = '0') — условие среза синхроимпульса.

Синтезируемый VHDL-код D-триггера с асинхронным сбросом и установкой будет выглядеть так:

```
library IEEE;  
  
use IEEE.std_logic_1164.all;  
  
entity dff_async is
```

```

port (data, clk, reset, preset : in std_logic;
      q : out std_logic);
end dff_async;

architecture behav of dff_async is
begin
process (clk, reset, preset) begin
    if (reset = '0') then
        q <= '0';
    elsif (preset = '0') then
        q <= '1';
    elsif (clk'event and clk = '1') then
        q <= data;
    end if;
end process;
end behav;

```

Для синтеза этого VHDL-кода необходимо выполнить следующие действия.

- Запустить САПР Quartus II фирмы Altera.
- Создать новый проект, выбрав в меню **File** пункт **New Project Wizard** и нажать кнопку **Next** в открывшемся окне. Ввести имя проекта в поле **What is the name of this project?** и нажать кнопку **Finish**.

- Создать новый файл, выбрав в меню **File | New**. В открывшемся окне на закладке **Device Design Files** выберите файл типа VHDL. В результате создастся файл для написания кода на соответствующем языке.
- Набрать или скопировать вышеприведенный VHDL-код D-триггера.
- Сохранить файл с именем dff_async.
- Откомпилировать проект, выбрав в меню **Processing | Start Compilation**. В процессе компиляции, в окне **Messages** отображаются текущие сообщения компилятора, если в коде присутствуют ошибки или не синтезируемые конструкции, компилятор выдаст ошибку.
- При успешном завершении компиляции откроется закладка **Compilation Report**, где в разделе **Analysis & Synthesis** можно посмотреть отчет синтезатора, например отчет об использовании ресурсов микросхемы (раздел **Resource Usage Summary**) (рис. 9.16).

	Resource	Usage
1	Total logic elements	1
2	Total combinational functions	0
3	-- Total 4-input functions	0
4	-- Total 3-input functions	0
5	-- Total 2-input functions	0
6	-- Total 1-input functions	0
7	-- Total 0-input functions	0
8	Combinational cells for routing	0
9	Total registers	1
10	I/O pins	5
11	Maximum fan-out node	q~reg0
12	Maximum fan-out	1
13	Total fan-out	5
14	Average fan-out	0.83

Рис. 9.16. Использование ресурсов микросхемы

Как видно из таблицы на рис. 9.16, для построения схемы синтезатор использовал 1 логический элемент (**Total logic elements — 1**), сконфигурированный как регистр (**Total registers — 1**) и 5 выводов микросхемы (**I/O pins — 5**).

- Для того чтобы открыть RTL-представление схемы, нужно выбрать пункт меню **Tools | RTL Viewer**. Откроется искомая схема (рис. 9.17)

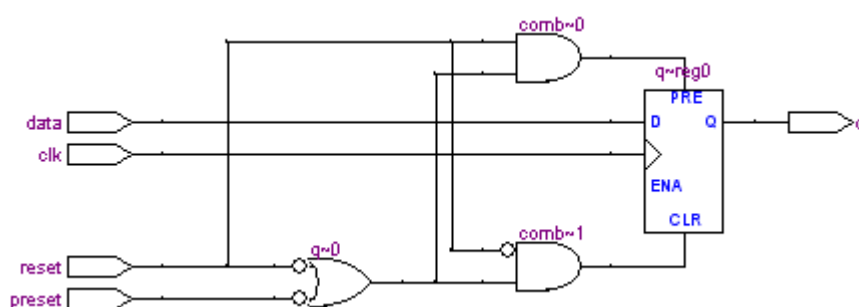


Рис. 9.167. RTL-представление

Для того чтобы открыть представление схемы в технологическом базисе, нужно выбрать пункт меню **Tools | Technology Map Viewer**. Откроется искомая схема (рис. 9.18)

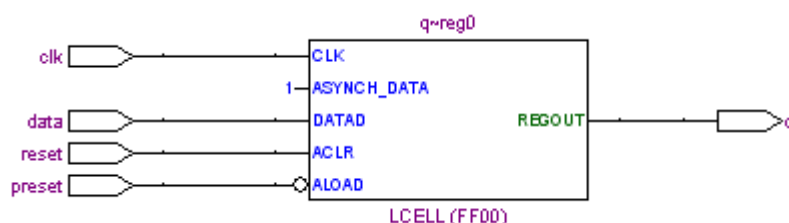


Рис. 9.178. Представление схемы в технологическом базисе

Повторите описанные действия, получите самостоятельно RTL-представление, представление в технологическом базисе и таблицу

использования ресурсов микросхемы для D-триггера с асинхронным сбросом и установкой.

Синтезируемый VHDL-код четырехразрядного регистра сдвига с асинхронным сбросом:

```
LIBRARY ieee;
```

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

```
ENTITY shift_reg IS
```

```
    PORT
```

```
        (clk   : IN STD_LOGIC ;
```

```
         data  : IN STD_LOGIC ;
```

```
         reset : IN STD_LOGIC ;
```

```
         q     : OUT STD_LOGIC
```

```
    );
```

```
END shift_reg;
```

```
ARCHITECTURE behav OF shift_reg IS
```

```
    SIGNAL rs: STD_LOGIC_VECTOR (3 downto 0) ;
```

```
BEGIN
```

```
process (clk, reset) begin
```

```
    if (reset = '0') then
```

```
        rs <= "0000";
```

```
elseif (clk'event and clk = '1') then  
    rs <= data & rs(3 downto 1);  
end if;  
end process;  
q <= rs(0);  
END behav;
```

Для четырехразрядного регистра сдвига с асинхронным сбросом повторите действия, ранее описанные для D-триггера, получите самостоятельно RTL-представление, представление в технологическом базисе и таблицу использования ресурсов микросхемы.