

ПОСЛЕДОВАТЕЛЬНОСТНАЯ ЛОГИКА

Лекционное занятие



Регистры

Регистр – это элемент последовательной логики, который «запоминает» предыдущее состояние на своем входе.

Сигналы, значения которым присвоены в операторах **always** сохраняют свое состояние, пока не случится событие из списка чувствительности оператора, приводящее к изменению их значения.

```
module flop(  
    input  logic  clk,  
    input  logic [3:0] d,  
    output logic [3:0] q);  
    always_ff @(posedge clk)  
        q <= d;  
endmodule
```

У триггера в списке чувствительности есть только сигнал **clk**, и потому триггер хранит старое значение **q** до следующего переднего фронта **clk**, даже если входной сигнал **d** изменился раньше.

Регистры

Оператор `q <= d` выполняется, когда случается событие, заданное в списке чувствительности. Триггер копирует `d` в `q` по переднему фронту тактового сигнала, а в остальное время значение `q` остается неизменным.

```
module flop(  
    input  logic  clk,  
    input  logic [3:0] d,  
    output logic [3:0] q);  
    always_ff @(posedge clk)  
        q <= d;  
endmodule
```

Для создания триггеров, защелок используется операторы `always` или `always_ff`.

Регистры со сбросом

Использование регистров со сбросом, даёт возможность привести систему в predetermined состояние.

Сброс бывает **синхронным** (сбрасывает выходной сигнал только по следующему переднему фронту такта) или **асинхронным** (происходит немедленно).

Сигналы в списке чувствительности оператора **always** разделяются **запятой** или словом **or**.

```

module flopr(
    input    logic    clk,
    input    logic    reset,
    input    logic [3:0] d,
    output   logic [3:0] q);

// асинхронный сброс
    always_ff @(posedge clk, posedge reset)
        if (reset)  q <= 4' b0;
        else       q <= d;

endmodule

module flopr(
    input    logic    clk,
    input    logic    reset,
    input    logic [3:0] d,
    output   logic [3:0] q);

// синхронный сброс
    always_ff @(posedge clk)
        if (reset)  q <= 4' b0;
        else       q <= d;

endmodule

```

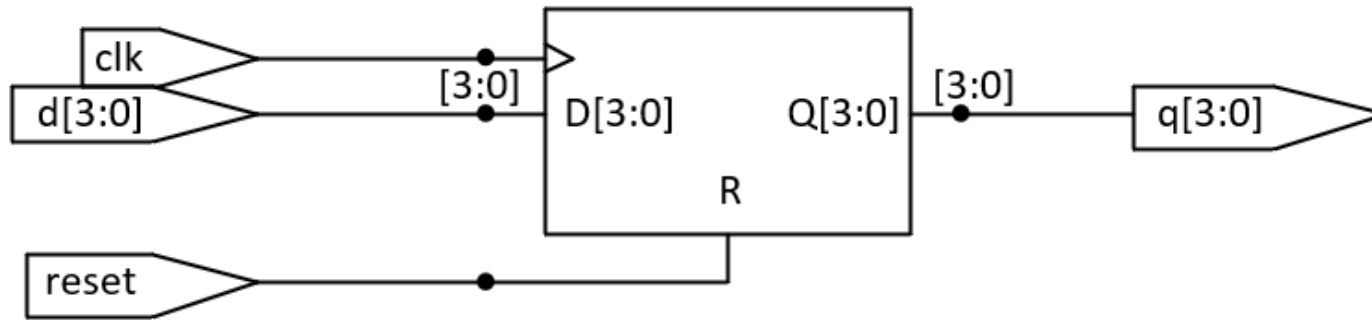
Регистры со сбросом

У триггера с асинхронным сбросом в списке чувствительности присутствует `posedge reset`.

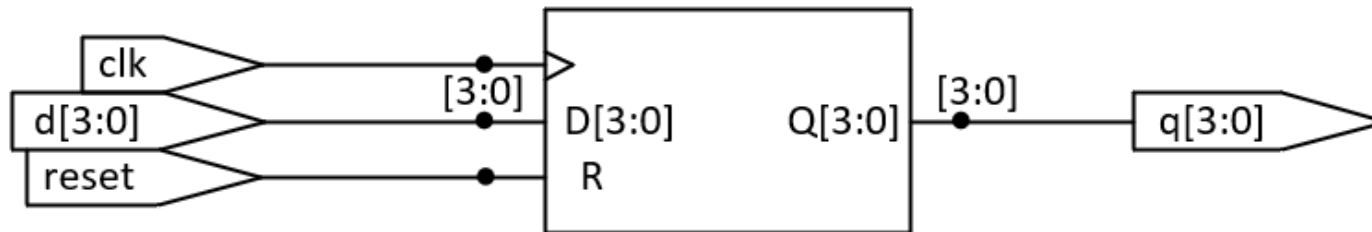
У триггера с синхронным сбросом этого сигнала нет. Триггер с асинхронным сбросом реагирует на передний фронт `reset` немедленно, а с синхронным – только по переднему фронту такта.

```
module flopr(  
    input    logic    clk,  
    input    logic    reset,  
    input    logic [3:0] d,  
    output   logic [3:0] q);  
  
// асинхронный сброс  
    always_ff @(posedge clk, posedge reset)  
        if (reset)  q <= 4' b0;  
        else       q <= d;  
  
endmodule  
  
module flopr(  
    input    logic    clk,  
    input    logic    reset,  
    input    logic [3:0] d,  
    output   logic [3:0] q);  
  
// синхронный сброс  
    always_ff @(posedge clk)  
        if (reset)  q <= 4' b0;  
        else       q <= d;  
  
endmodule
```

Регистры со сбросом



a)



b)

Синтезированная схема модуля flopr
(a) с асинхронным сбросом,
(b) с синхронным сбросом

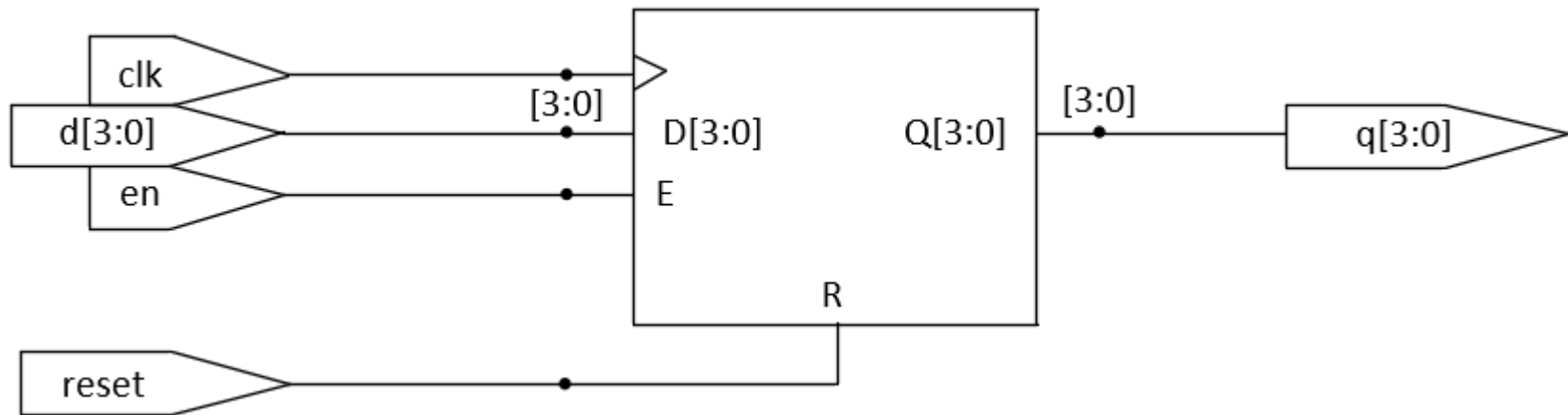
Регистры с сигналом разрешения

```
module flopenr(  
    input  logic  clk,  
    input  logic  reset,  
    input  logic  en,  
    input  logic [3:0] d,  
    output logic [3:0] q);  
  
// асинхронный сброс  
always_ff @(posedge clk, posedge reset)  
    if (reset)    q <= 4' b0;  
    else if (en)  q <= d;  
  
endmodule
```

РЕГИСТР С УСЛОВИЕМ И СБРОСОМ

Регистры с сигналом разрешения реагируют на тактовый импульс только при условии подачи активного уровня на вход разрешения **en**.

Регистры с сигналом разрешения



Синтезированная схема модуля flopenr

Группы регистров

Синхронизатор – это устройство, на вход которого поступает асинхронный сигнал **D** и тактовый сигнал **CLK**.

```
module sync(  
    input  logic  clk,  
    input  logic  d,  
    output logic  q);  
    logic          n1;  
    always_ff @(posedge clk) begin  
        n1 <= d; // неблокирующий  
        q <= n1; // неблокирующий  
    end  
endmodule
```

За ограниченное время он формирует выходной сигнал, который с очень высокой вероятностью имеет корректный логический уровень.

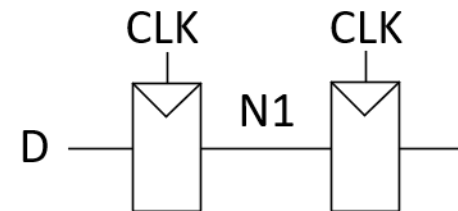


Схема синхронизатора

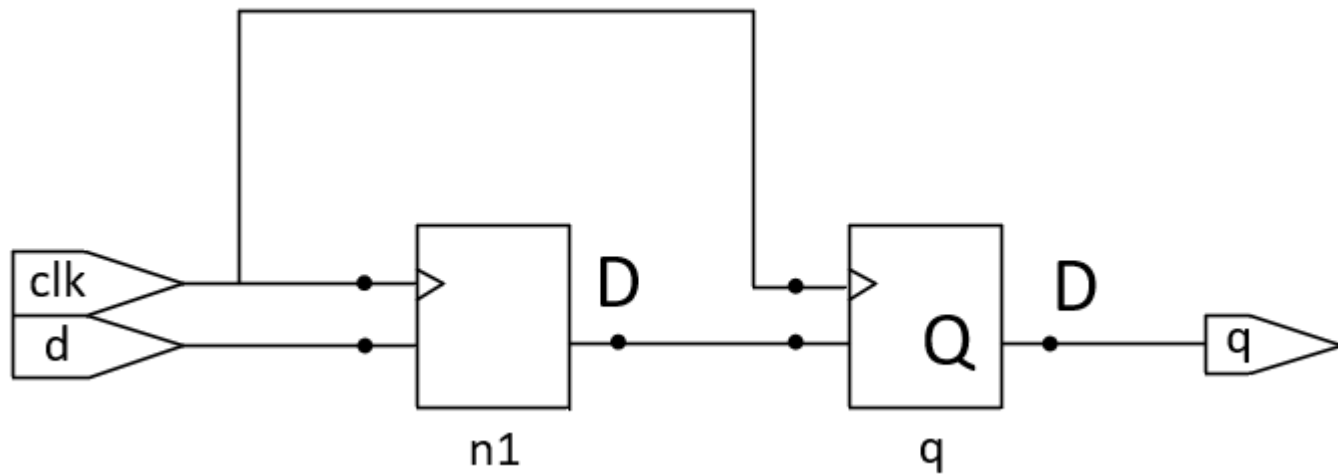
Группы регистров

```
module sync(  
    input  logic  clk,  
    input  logic  d,  
    output logic  q);  
    logic          n1;  
    always_ff @(posedge clk) begin  
        n1 <= d; // неблокирующий  
        q <= n1; // неблокирующий  
    end  
endmodule
```

По переднему фронту `clk`, `d` копируется в `n1`, и в то же время `n1` копируется в `q`.

Конструкция `begin/end` обрамляет группу из нескольких операторов, находящихся внутри оператора `always`.

Группы регистров



Синтезированная схема модуля sync

Литература

1. Дэвид М. Хэррис, Сара Л. Хэррис Цифровая схемотехника и архитектура компьютера. Второе издание. Morgan Kaufman, 2013 г. – 1648 стр.
2. FPGA Tutorial. / [Электронный ресурс] / URL: <https://fpgatutorial.com/>

СПАСИБО ЗА ВНИМАНИЕ

