

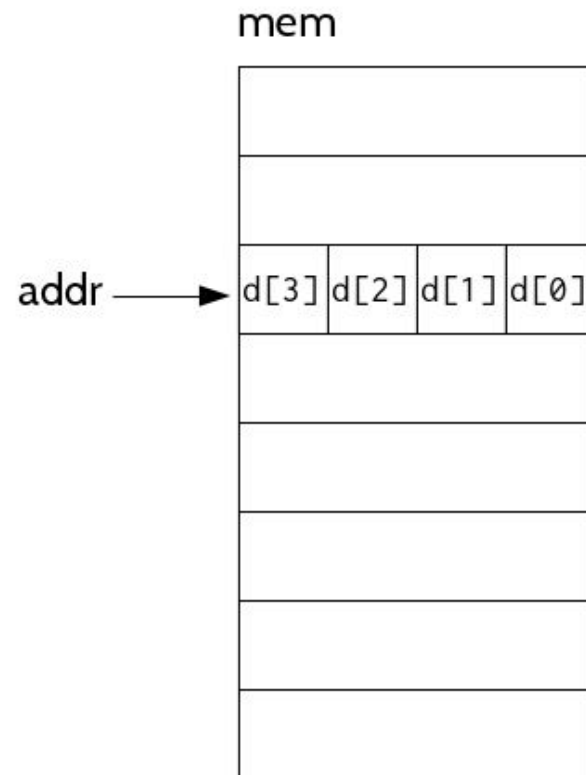
Разработка под FPGA

Занятие 6.
RAM, FIFO.

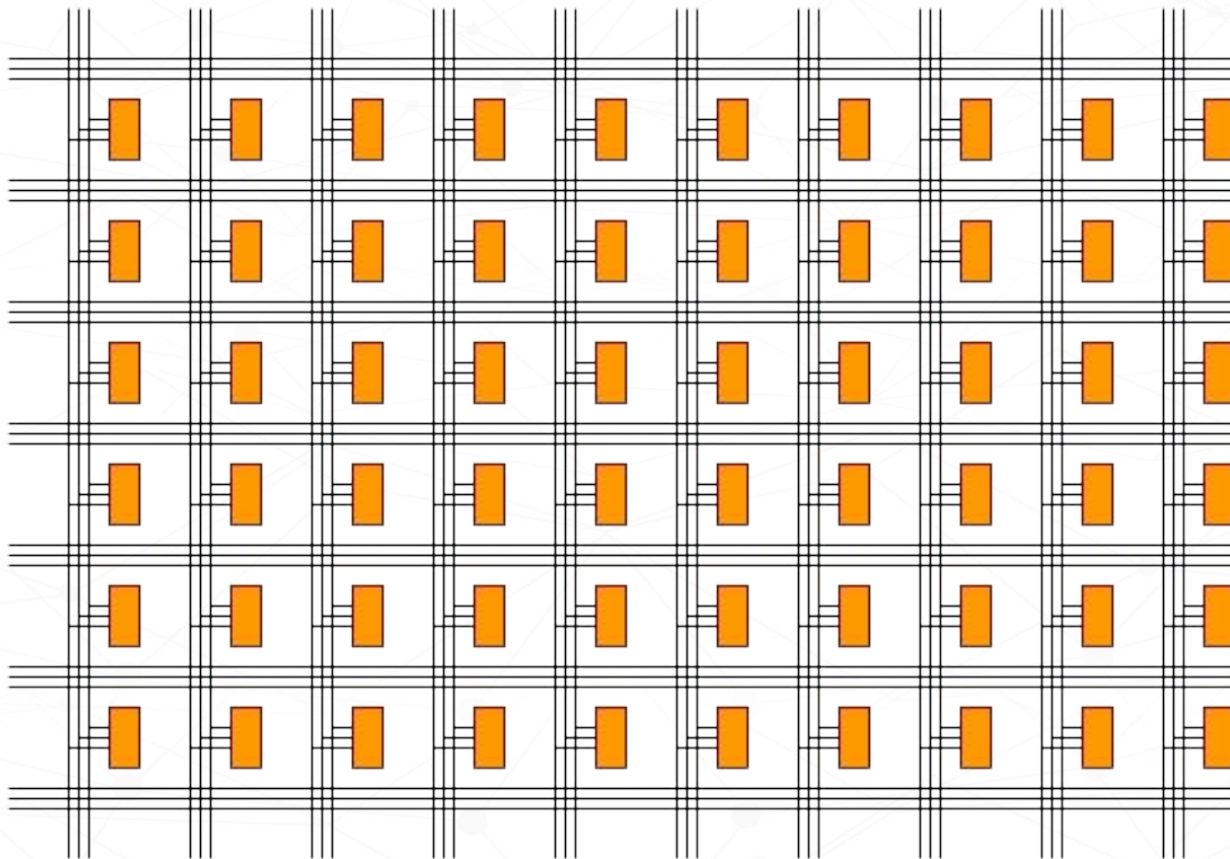
- Оперативная память и её реализация в ПЛИС
- FIFO
- Многодоменные схемы и FIFO.

```
reg [3:0]    mem[7:0];
wire [3:0]   d;
reg [2:0]    addr;
```

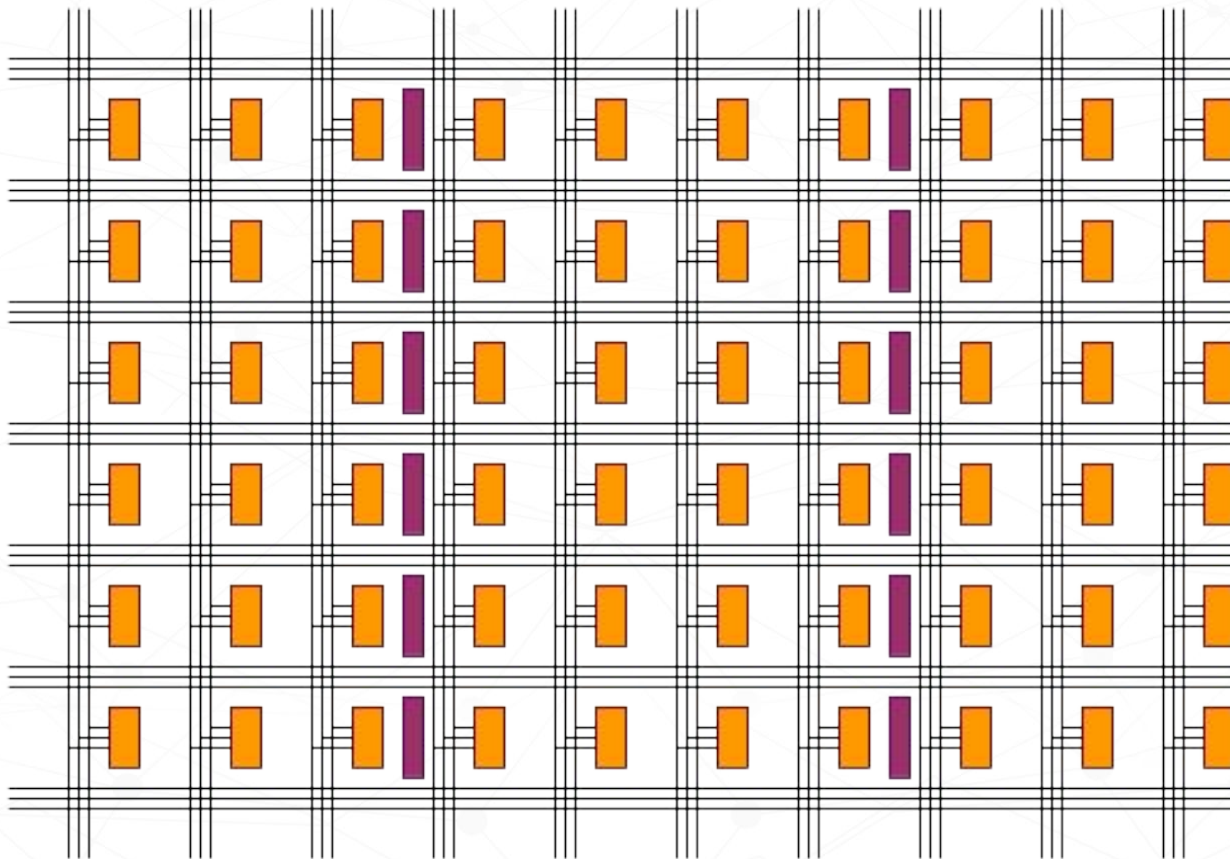
```
assign d = mem[addr];
```



- При помощи обычной логики
- Специализированные блоки памяти в кристалле



Block RAM



- Описание на языке Verilog/SystemVerilog
 - Quartus: Edit -> Insert Template -> ...
- Используя “мегафункцию”
 - MegaWizard
 - IP catalog

Образец кода

```

module single_port_ram #(
parameter DATA_WIDTH    =    8,
parameter ADDR_WIDTH     =    6    )(
    input    [DATA_WIDTH-1:0]    data,
    input    [ADDR_WIDTH-1:0]    addr,
    input    we, clk,
    output   [DATA_WIDTH-1:0]    q
);
    reg [DATA_WIDTH-1:0]    ram[2**ADDR_WIDTH-1:0];
    reg [ADDR_WIDTH-1:0]    addr_reg;
    always @ (posedge clk)
        begin
            if (we)
                ram[addr] <= data;
                addr_reg <= addr;
            end
    assign q = ram[addr_reg];
endmodule

```


MegaWizard

MegaWizard Plug-In Manager [page 1 of 6]

RAM: 1-PORT

About Documentation

1 Parameter Settings 2 EDA 3 Summary

Widths/Blk Type/Clks > Regs/Cken/Byte Enable/Aclrs > Read During Write Option > Mem Init >

Currently selected device family: Cyclone IV E

☒ Match project/default

How wide should the 'q' output bus be? 8 bits

How many 8-bit words of memory? 256 words

Note: You could enter arbitrary values for width and depth

What should the memory block type be?

☐ Auto ☐ MLAB ☒ M9K

☐ M144K ☐ LCs Options...

Set the maximum block depth to Auto words

What clocking method would you like to use?

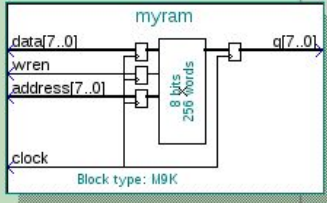
☒ Single clock

☐ Dual clock: use separate 'input' and 'output' clocks

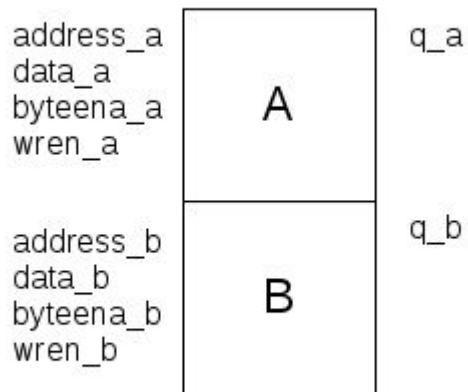
Resource Usage

1 M9K

Cancel < Back Next > Finish



Режимы работы блочной памяти



- Большой объем хранимых данных -> блоки
- Блочная память не сбрасывается
- Данные возвращаются минимум на следующий такт
- В кристалле их сотни
- Объем одного блока: 9К, 16К, 144К

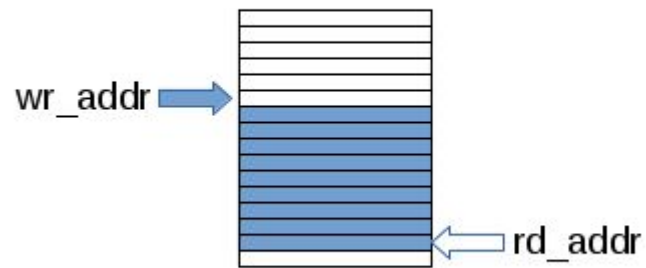
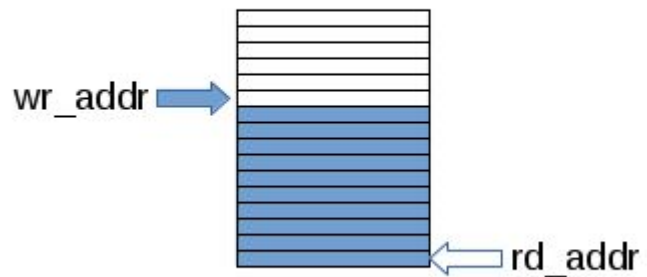
FIFO



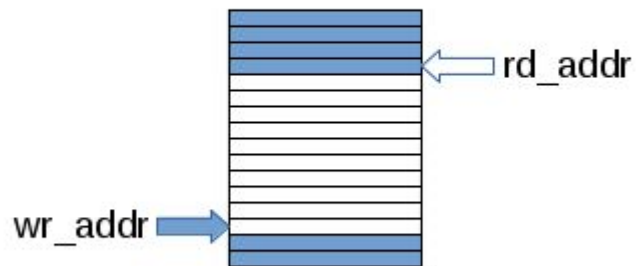
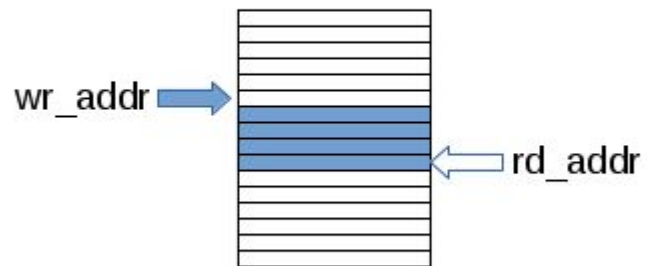
Как работает



Как работает



Как работает



Описание FIFO на Verilog

```

module fifo(
    input          push,
    input          pull,
    input          [7:0] data_in,
    output         [7:0] data_out,
    input         reg  [7:0] reset,
    input         clk
);

reg [3:0] wr_addr;
reg [3:0] rd_addr;
reg [7:0] mem[15:0];

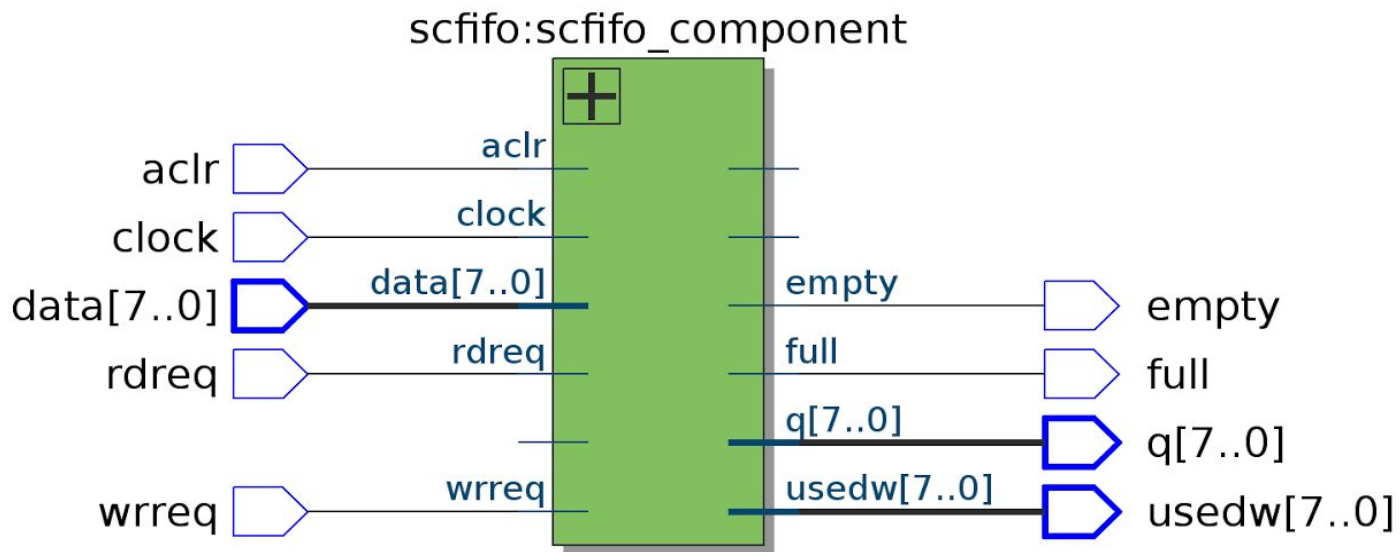
always @(posedge clk)
    if (push)
        mem[wr_addr] <= data_in;
    
```

```

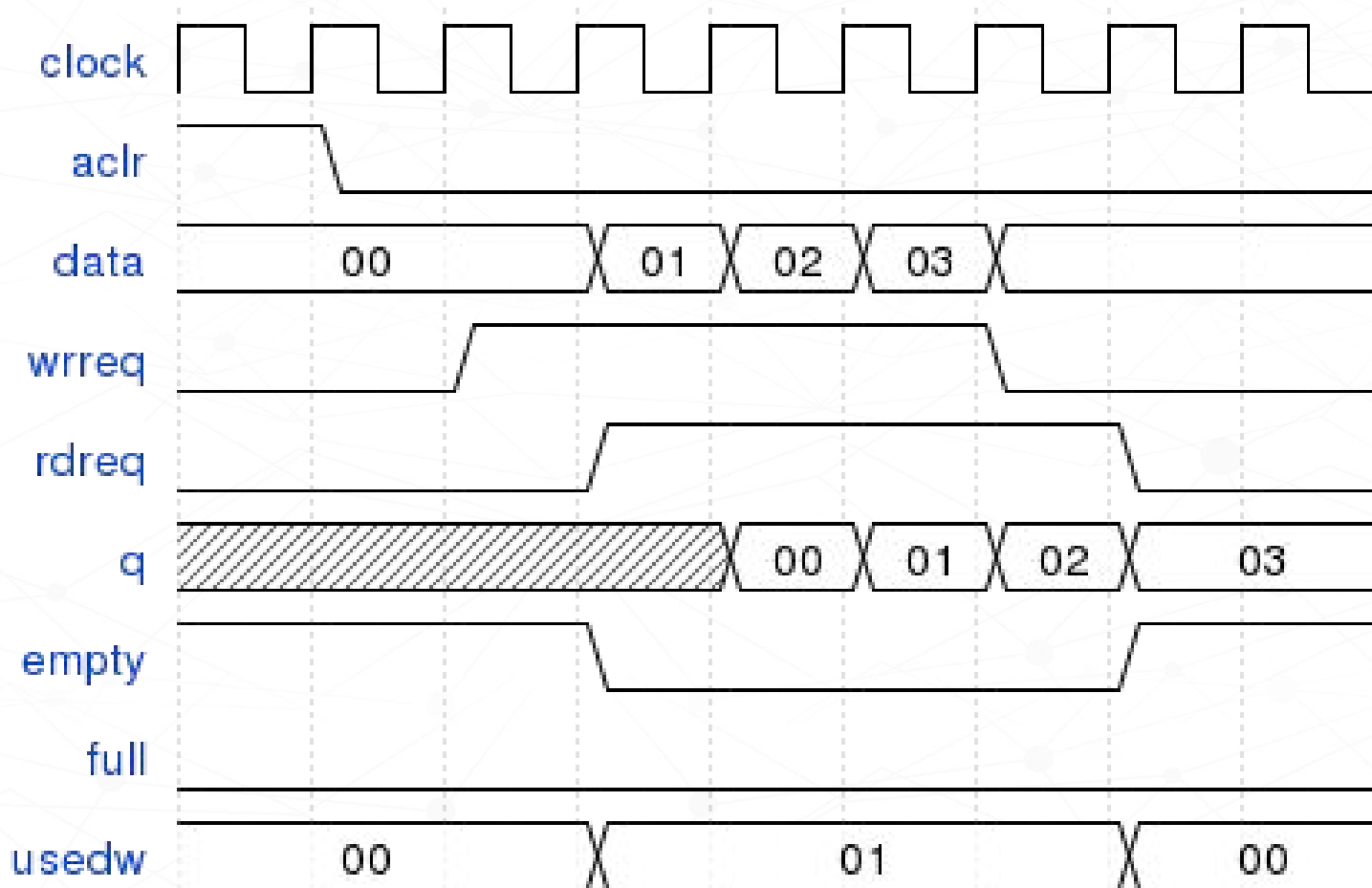
always @(posedge clk)
    if (reset)
        begin
            wr_addr <= 0;
            rd_addr <= 0;
        end
    else
        begin
            wr_addr <= push ? wr_addr + 4'b1 : wr_addr;
            rd_addr <= pull ? rd_addr + 4'b1 : rd_addr;
        end

always @(posedge clk)
    if (pull)
        data_out <= mem[rd_addr];

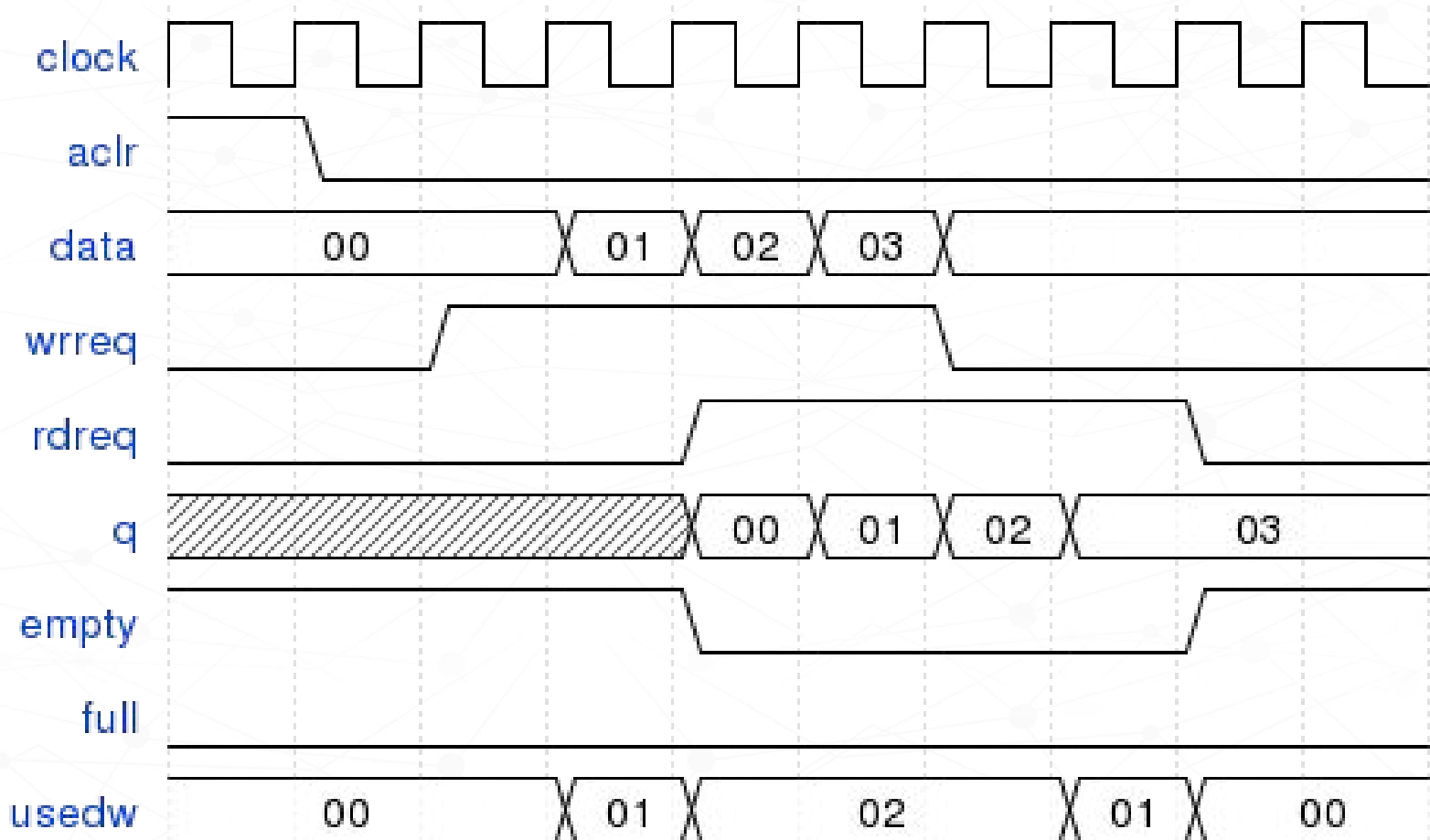
endmodule
    
```

scfifo SHOWAHEAD = OFF

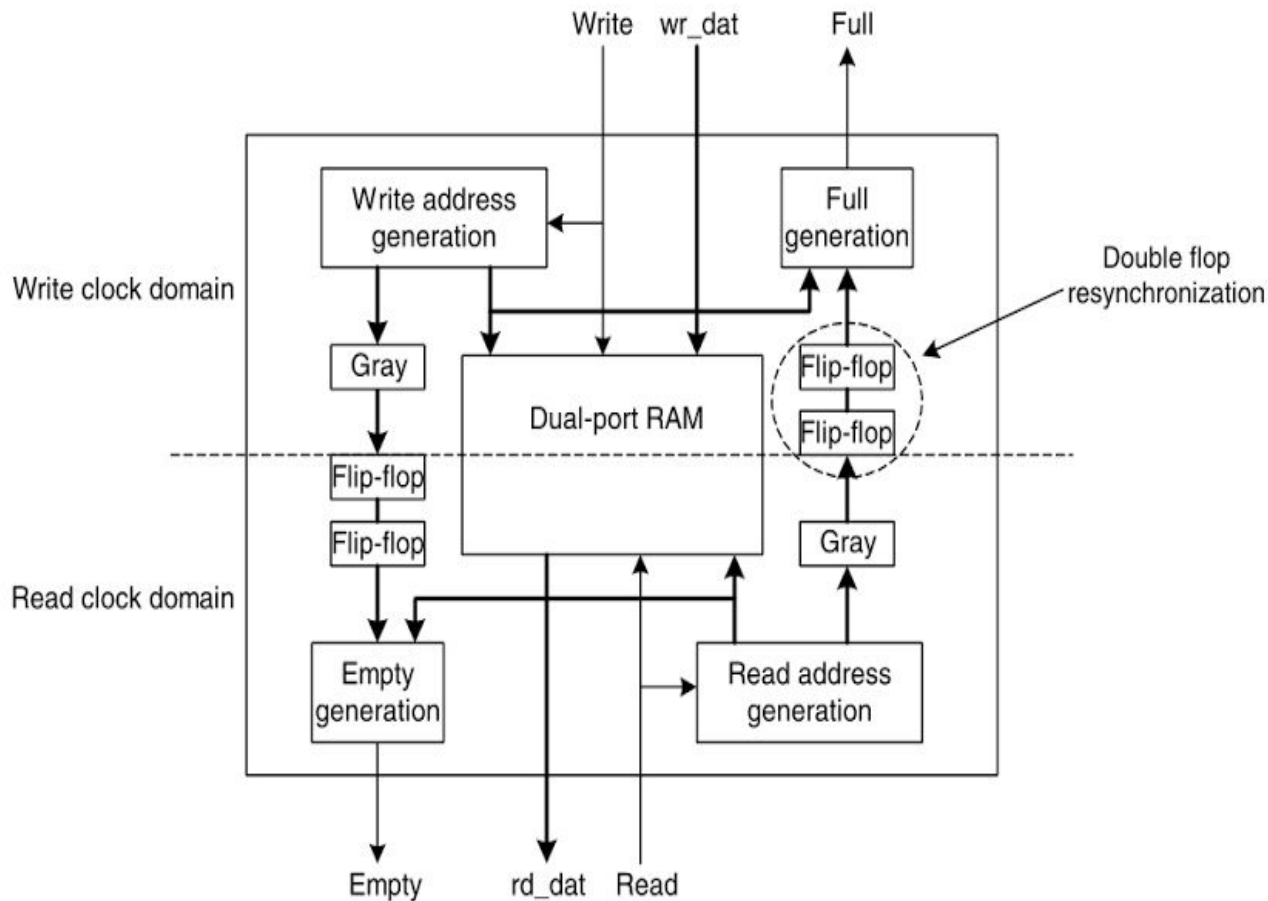


scfifo SHOWAHEAD = ON



- FIFO может переполняться
- Quartus дает возможность применять аппаратное FIFO (**scfifo**).
- Перенос данных с домена на домен (**dcfifo**).

Асинхронное FIFO



Спасибо за внимание!

Данил Бычков

d.bychkov@metrotek.spb.ru