

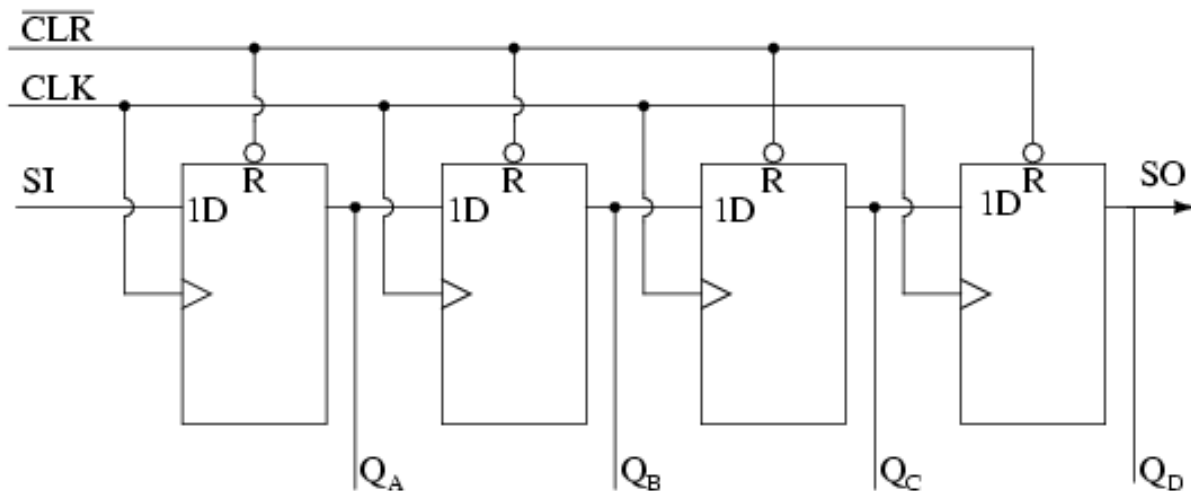


УНИВЕРСИТЕТ ИТМО

«Ликбез по описанию логики»

Весна, 2023

- ✓ **Always-блок;**
- ✓ **Тактовый сигнал** в списке чувствительности;
- ✓ Ключевые слова **posedge/negedge;**



Serial-in/ Parallel out shift register details

- ✓ Простейший тип регистров — активен каждый такт;
- ✓ Активность = энергопотребление;

```
always @ (posedge clk)
    q1 <= d;
```

```
always @ (negedge clk)
    q1 <= d;
```

- ✓ Наличие сигнала сброса;
- ✓ Сигнал сброса может быть как синхронным, так и асинхронным

```
always @ (posedge clk or posedge rst) begin
    if (rst)
        q3 <= 0;
    else
        q3 <= d;
end
```

```
always @ (negedge clk or negedge rst) begin
    if (~rst)
        q4 <= 0;
    else
        q4 <= d;
end
```

```
always @ (posedge clk) begin
    if (rst)
        q3 <= 0;
    else
        q3 <= d;
end
```

- ✓ Наличие **сигнала разрешения**;
- ✓ Запись данных в «**определённые**» моменты времени снижает энергопотребление;

```
.. always @ (posedge clk) begin
..   |.. if (enable)
..   |..   |.. q5 <= d;
.. end
```

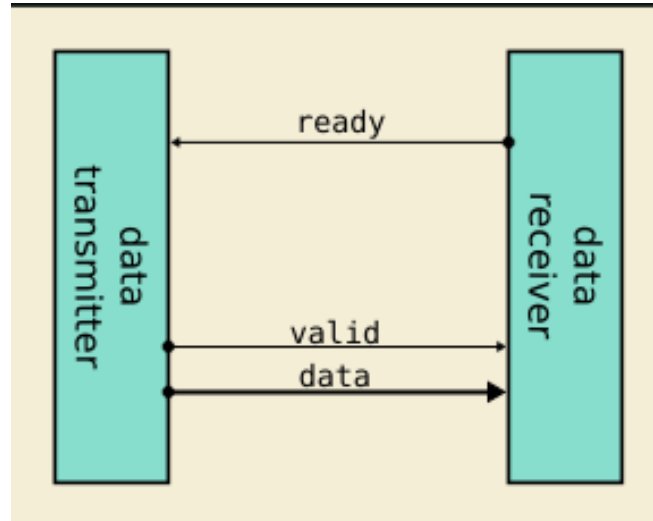
- ✓ Комбинационная логика должна учитывать все возможные комбинации входных данных

```
reg [2:0] data_in;  
reg [15:0] data_out;  
  
always @*  
case (data_in)  
    3'b000 : data_out = 16'h1 << dmem_address;  
    3'b001 : data_out = 16'h3 << dmem_address;  
    3'b010 : data_out = 16'hF << dmem_address;  
    3'b011 : data_out = 16'hFF << dmem_address;  
    3'b100 : data_out = 16'hFFFF;  
endcase
```

- ✓ Не мешайте описание регистров с остальной логикой.

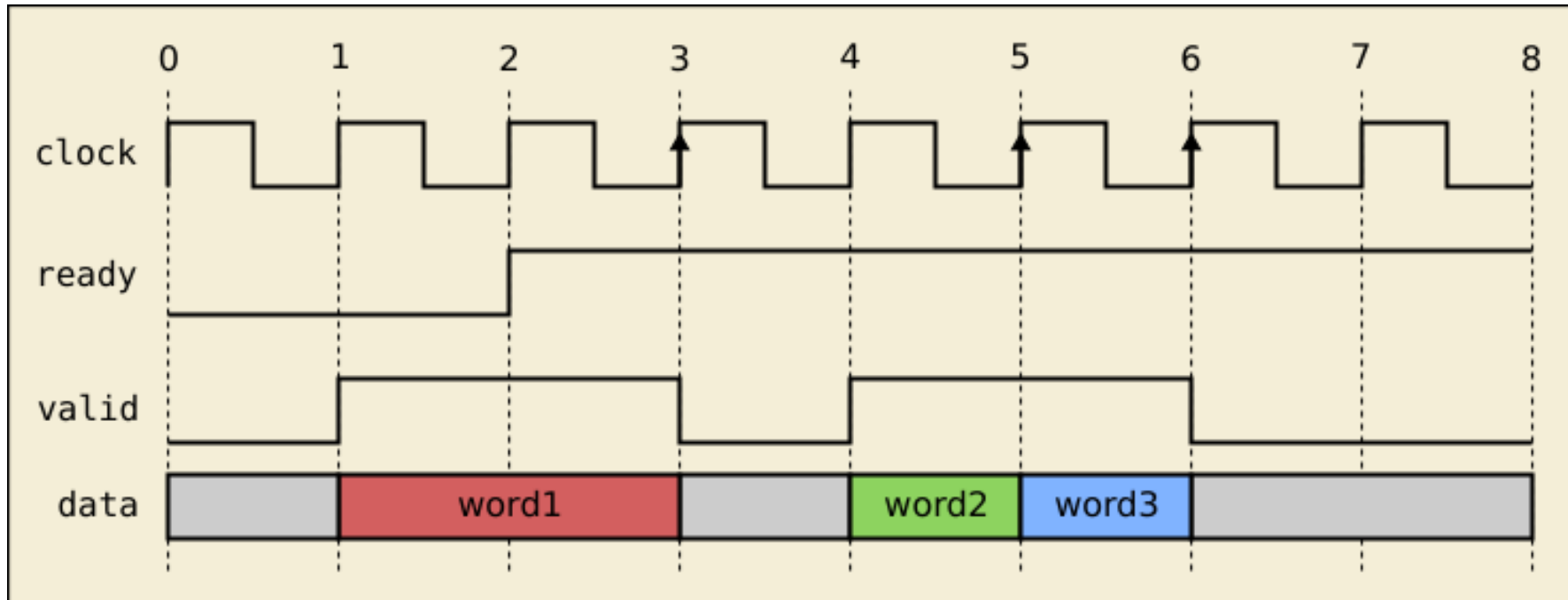
```
reg [2:0] data_in;  
reg [15:0] data_out;  
reg [15:0] data_ff;  
  
always @ (posedge clk) begin  
    case (data_in)  
        3'b000 : data_out = 16'h1 << dmem_address;  
        3'b001 : data_out = 16'h3 << dmem_address;  
        3'b010 : data_out = 16'hF << dmem_address;  
        3'b011 : data_out = 16'hFF << dmem_address;  
        3'b100 : data_out = 16'hFFFF;  
    endcase  
  
    if (enable) begin  
        data_ff <= data_out;  
    end  
end
```

- ✓ Один из простейших протоколов для обмена данными;
- ✓ Обмен данными происходит между двумя участниками;
- ✓ «Ready» - сигнал готовности принять транзакцию;
- ✓ «Valid» - сигнал корректности данных на шине данных.



- ✓ Перед тем, как отправить новую транзакцию на «свободную» шину, сигнал **VALID** не дожидается установки **READY**;
- ✓ Сигнал **VALID** удерживается и шина будет «занята» до того момента, пока не установится сигнал **READY**;

State Name	ready	valid	Description
Idle	0	0	Transmitter does not have valid data.
Wait for Ready	0	1	Transmitter has valid data, but Receiver is not ready for it. Data will NOT be transferred.
Wait for Valid	1	0	Receiver is ready for data, but transmitter has none.
Transfer	1	1	Transmitter has valid data, and Receiver is ready for it. Data will be transferred.



- ✓ Сигнал должен быть стабилен в момент его записи в регистр.

