Конечные автоматы
Диаграммы состояний
Рекомендации по стилю кодирования

Дмитрий Смехов

2023

Лекция по конечным автоматам

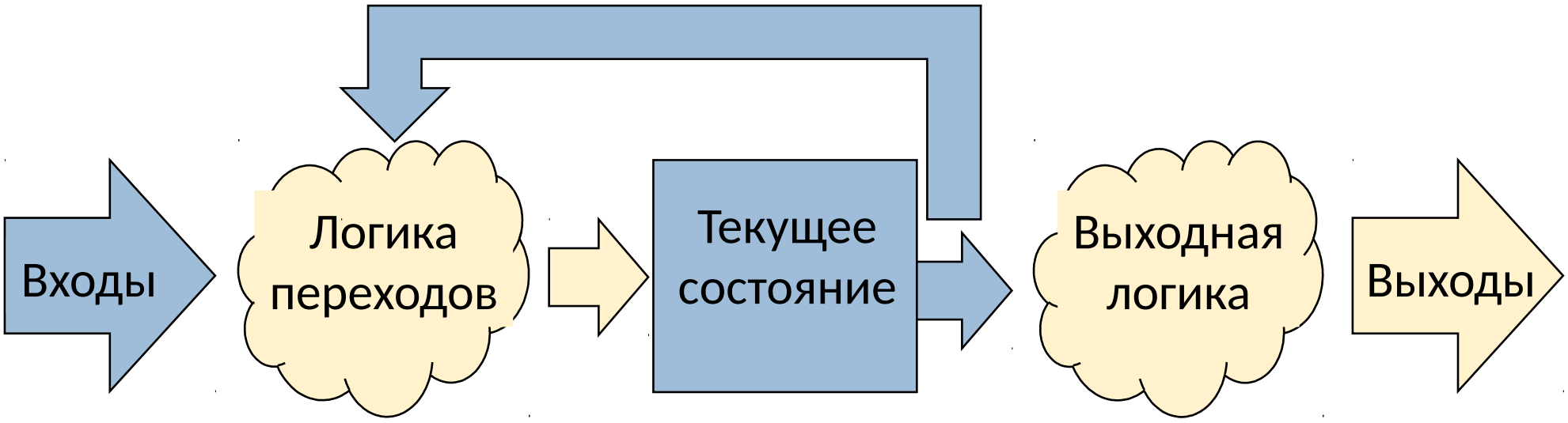
Концепция конечного автомата на ПЛИС

Представление алгоритмов в виде конечных автоматов

Александр Силаньтев **https://youtu.be/MDkaXKM5XYI**

2023

Автомат Мура



Автомат Мили

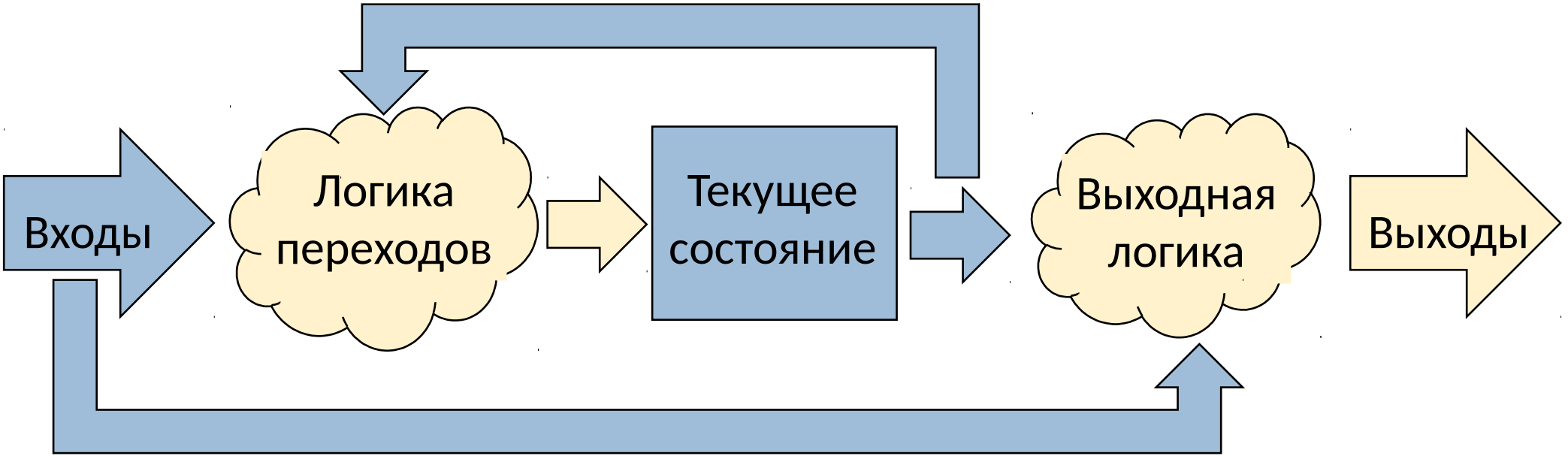
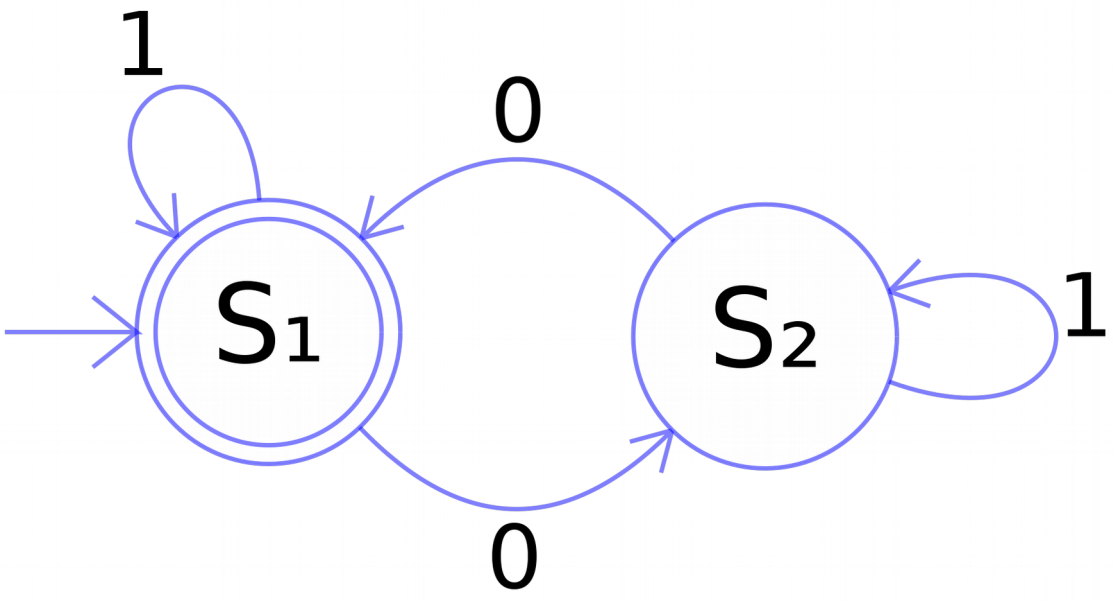
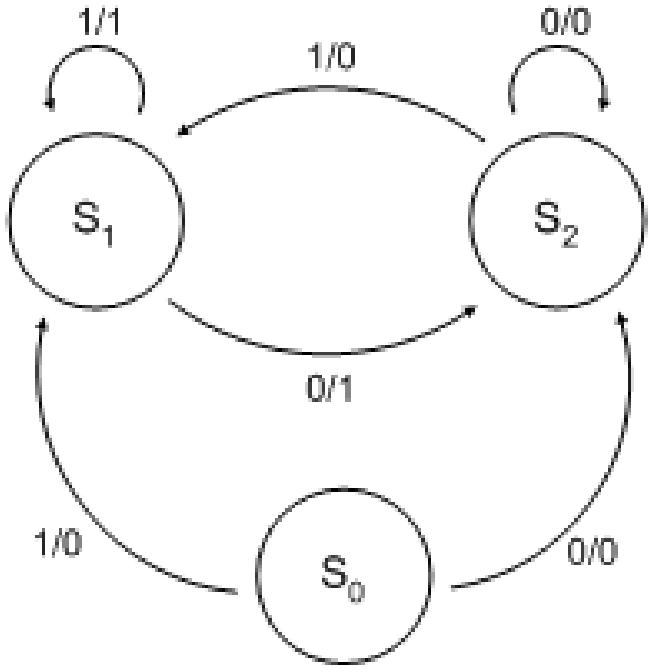


Диаграмма состояния автомата Мура



S1 и S2 — состояния. Дуги помечены входными сигналами

Диаграмма состояния автома Мили



S0, S1, S2 — состояния.
Дуги помечены как j / k
j — входные данные
k — выходные данные

Реализация на SystemVerilog

```
logic [0:0]     state;

always_ff @(posedge clk) begin
    if( rstp ) begin
        state <= # 1 0;
        c <= #1 0;
    end else begin
         case( state )
            0: begin
                 c <= #1 0;
                 if( a )
                     state <= #1 1;
            end

            1: begin
                 c <= #1 1;
                 if( ~a )
                     state <= #1 0;
            end
        endcase
    end
end
```

```
logic [0:0]     state;

always_ff @(posedge clk) begin
        case( state )
            0: begin
                 c <= #1 0;
                 if( a )
                     state <= #1 1;
            end

            1: begin
                 c <= #1 1;
                 if( ~a )
                     state <= #1 0;
            end
        endcase

        if( rstp )
             state <= # 1 0;
end
```

Два always блока

```
logic [0:0]     state;
logic [0:0]     next_state;

always_ff @(posedge clk) begin
    if( rstp )
        state <= # 1 0;
    else
        state <= # 1 next_state;
end
```

```
always_comb begin
        case( state )
        0: begin
           c = 0;
            if( a ) begin
                next_state = 1;
        end

        1: begin
           c = 1;
            if( ~a ) begin
                next_state = 0;
        end

end
```

Это автомат Мура. Выходы не регистровые

Два always блока — автомат Мили

```
logic [0:0]     state;
logic [0:0]     next_state;

always_ff @(posedge clk) begin
    if( rstp )
        state <= # 1 0;
    else
        state <= # 1 next_state;
end
```

```
always_comb begin
        case( state )
        0: begin
           c = 0;
            if( a ) begin
                c=1;
                next_state = 1;
            end
        end

        1: begin
           c = 1;
            if( ~a ) begin
                next_state = 0;
        end

end
```

Это автомат Мили. Выходы не регистровые
В данном примере во время действия rstp выходной сигнал c может быть равен 1

Два always блока — сложная логика

```
logic [0:0]     state;
logic [0:0]     n_state;

always_ff @(posedge clk) begin

    if( rstp ) begin
        cnt  <= #1 0;
        state <= # 1 0;
    end else begin
        cnt  <= #1 n_cnt;
        state <= # 1 n_state;
    end

end
```

```
always_comb begin

        n_state = state;
        n_cnt = cnt;

        if( a )
            n_state = 1;

        if( 1==n_state ) begin
           n_cnt++;
        end

        if( b )
           n_cnt=0;

        if( 10==n_cnt )
            n_state = 0;



end
```