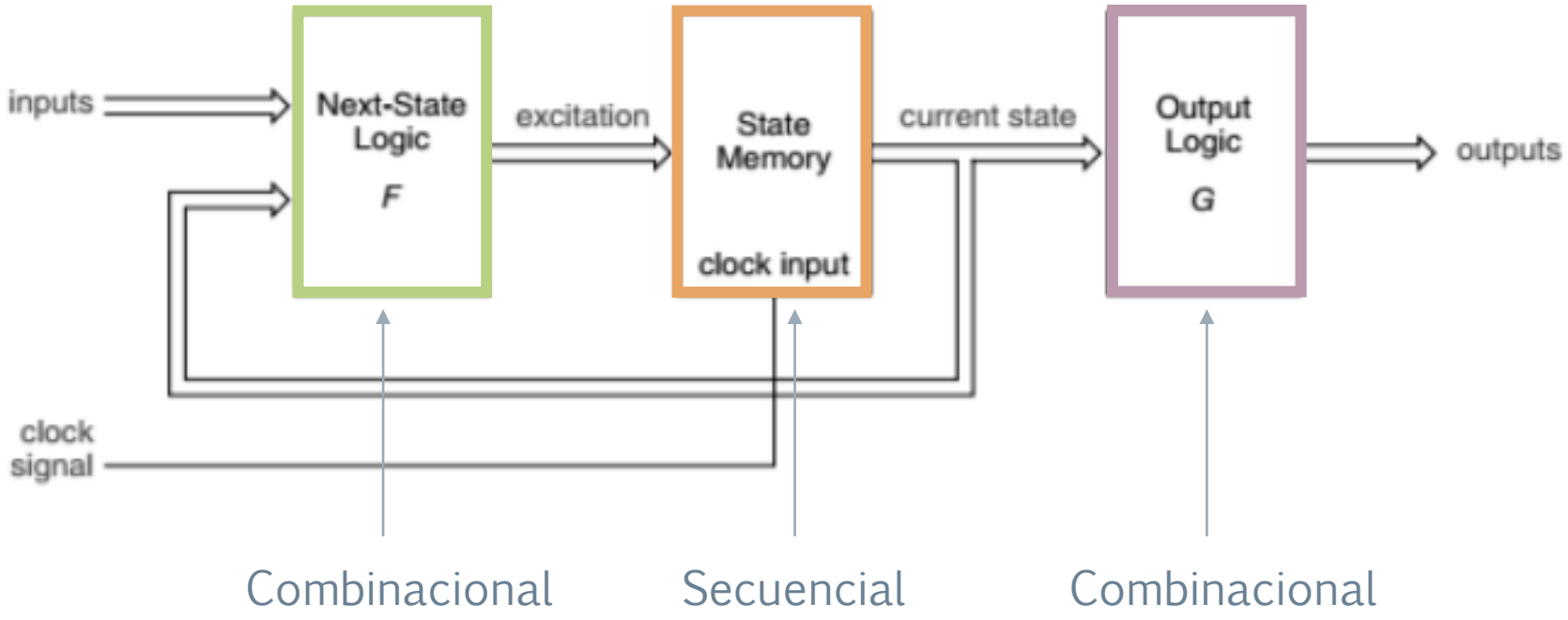
Electrónica III

Curso 2021

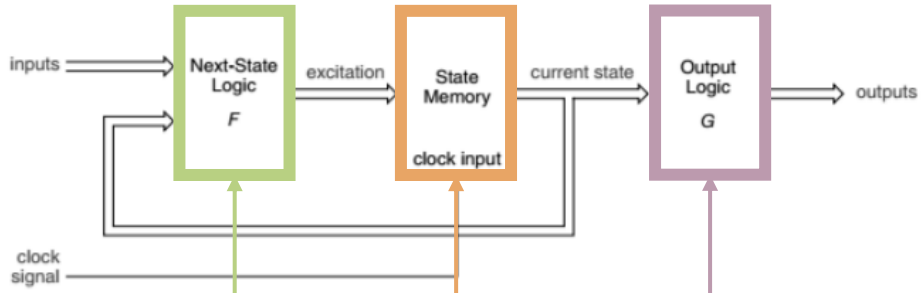
Máquinas de Estado en Verilog

Maquina de estados de Moore

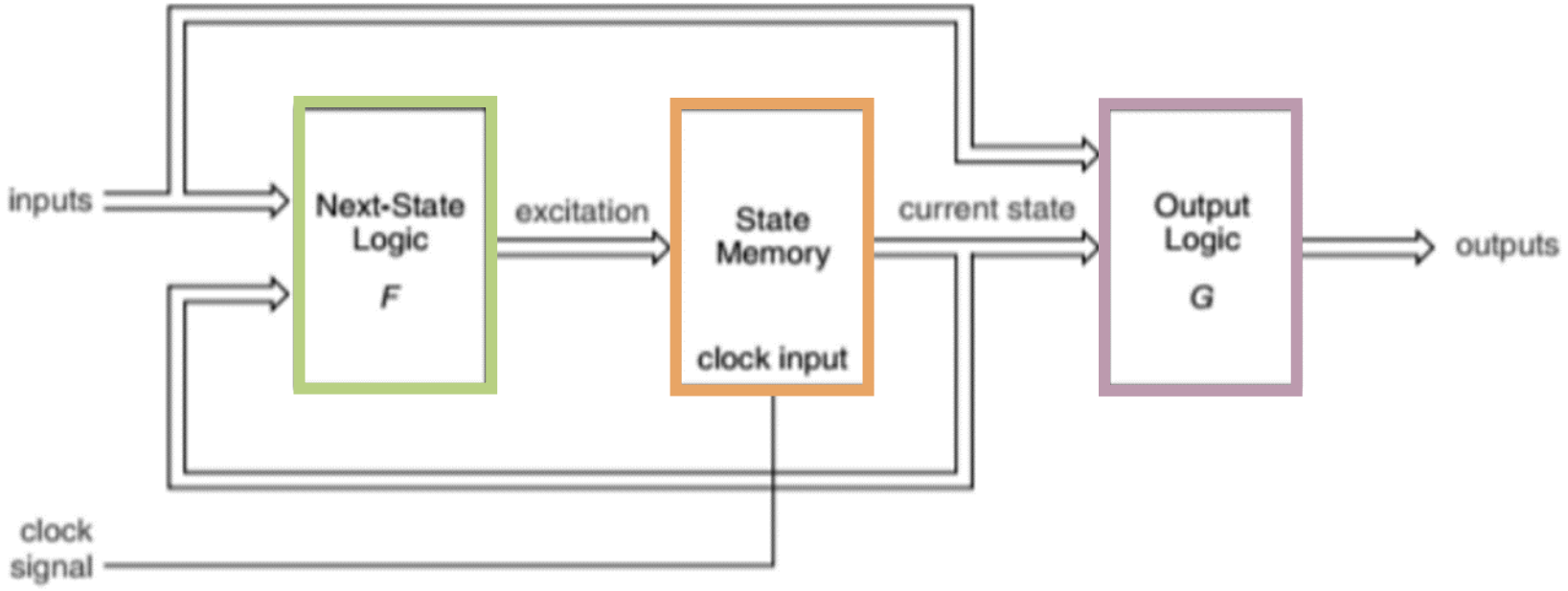


Maquina de estados de Moore

```verilog
//Implementacion de una maquina de estados de Moore en Verilog
module fsm_moore (clk, resetn, w, y, z);
    input clk, resetn, w;   // Clock, reset, sensor inputs (async)
    output z;               // Control output
    output [2:1] y;         // State output (para debug)

    reg [2:1] y, Y;

    // Asignacion de estados
    parameter [2:1] A = 2'b00;
    parameter [2:1] B = 2'b01;
    parameter [2:1] C = 2'b10;

    // Logica de proximo estado (combinacional)
    always @(w, y)
        case (y)
            A:  if (w) Y = B;
                else Y = A;
            B:  if (w) Y = C;
                else Y = A;
            C:  if (w) Y = C;
                else Y = A;
            default: Y = 2'bxx;
        endcase

    // Transicion al proximo estado (secuencial)
    always @(negedge resetn, posedge clk)
        if (resetn == 0) y <= A;
        else y <= Y;

    // Salida (combinacional)
    assign z = (y == C);
endmodule
```

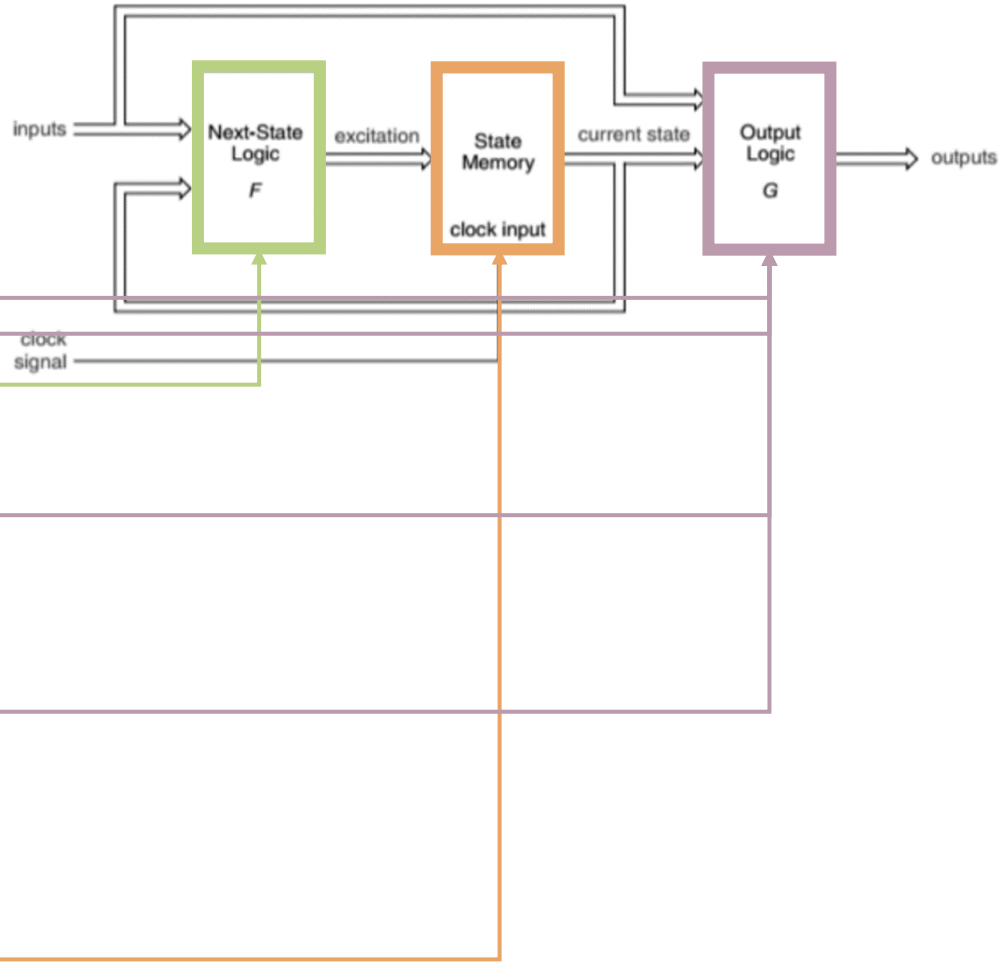


Maquina de estados de Mealy



Maquina de estados de Mealy

```
module fsm_mealy (clk, resetn, w, y, z);
    input clk, resetn, w;
    output reg z;
    output [2:1] y;
    reg [2:1] y, Y;
    // Asignacion de estados
    parameter [2:1] A = 2'b00;
    parameter [2:1] B = 2'b01;

    // Logica de proximo estado y salida (combinacional)
    always @(w, y)
        case (y)
            A: if (w)
                begin
                z = 0;
                Y = B;
                end
               else
                begin
                z = 0;
                Y = A;
                end
            B: if (w)
                begin
                z = 1;
                Y = B;
                end
               else
                begin
                z = 0;
                Y = A;
                end
        endcase

    // Transicion de estado
    always @(negedge resetn, posedge clk)
        if (resetn == 0) y <= A;
        else y <= Y;

endmodule
```



Maquina de estados de Moore (alternativa)

```
//Implementacion de una maquina de estados de Moore en Verilog
//Implementacion alternativa con un solo bloque always
module fsm_moore2 (clk, resetn, w, y, z);
    input clk, resetn, w;
    output z;
    output [2:1] y;

    reg [2:1] y;

    // Asignacion de estados
    parameter [2:1] A = 2'b00;
    parameter [2:1] B = 2'b01;
    parameter [2:1] C = 2'b11;

    // Logica de proximo estado y avance de estado combinadas
    always @(negedge resetn, posedge clk)
        if (resetn == 0) y <= A;
        else
            case (y)
                A:  if (w)  y <= B;
                    else    y <= A;
                B:  if (w)  y <= C;
                    else    y <= A;
                C:  if (w)  y <= C;
                    else    y <= A;
                default:    y <= 2'bxx;
            endcase

    assign z <= (y == C);

endmodule
```