

**INGENIERÍA INFORMÁTICA-TECNOLOGÍAS INFORMÁTICAS**  
**CIRCUITOS ELECTRÓNICOS DIGITALES**  
**Grupo1. Primera prueba de evaluación continua. Noviembre 2011**

Nombre: .....

1. Indicar el valor de las intensidades en cada rama y las tensiones en cada nudo para el circuito de la figura 1. Suponer que se trabaja en régimen estacionario y  $V_i=5\text{ V}$ .

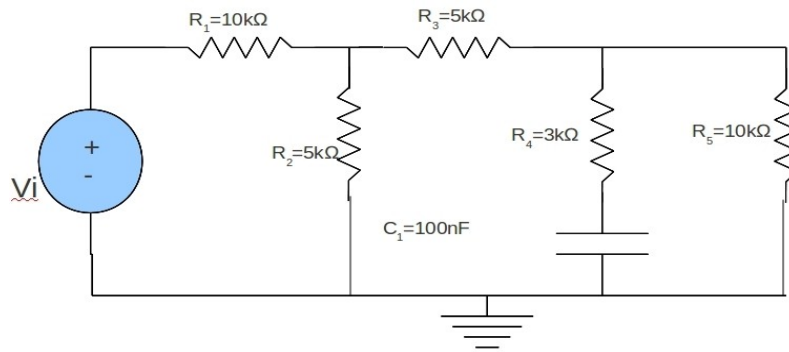


figura 1

2. Analice lógicamente el circuito de la figura 2 representando el valor de la función F mediante un k-mapa.

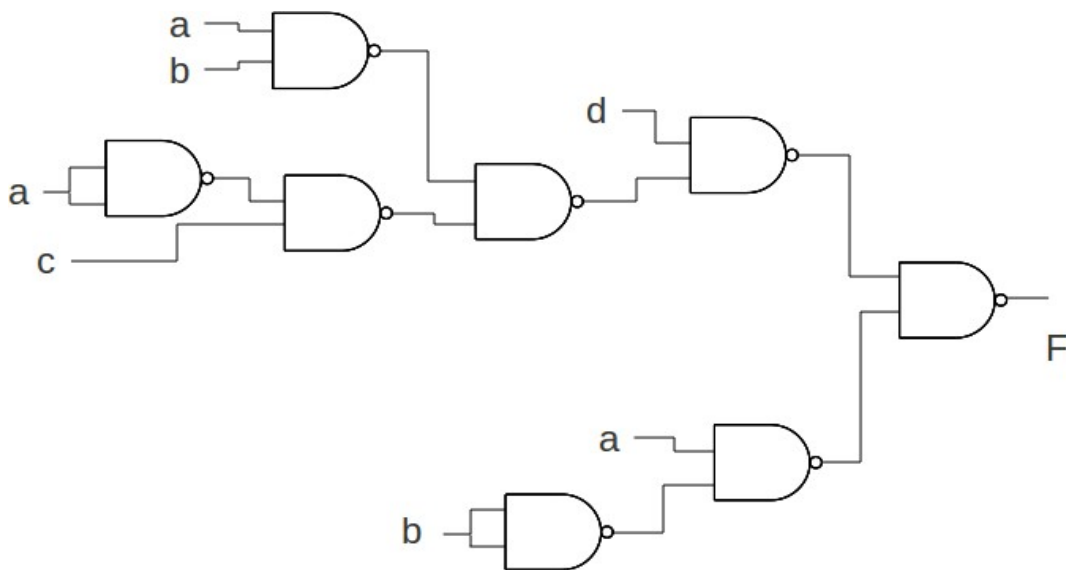


Figura 2.

3. Analice temporalmente el circuito de la figura 2, para el caso  $b=c=d=1$  y  $a$  una señal variable de periodo 10MHz. Considere que las puertas tienen un retardo de valor 10ns. Utilice la plantilla de la figura 3 para dibujar los cronogramas. Indique si el circuito presenta algún comportamiento anómalo.

4. Es obvio que el circuito de la figura 2 no es un circuito óptimo. Se pide en este ejercicio realizar un circuito que realice la misma función que el de la figura 2 pero que sea óptimo y en dos niveles (o tres) utilizando únicamente puertas NAND.
5. Realice un Sumador Completo (FA) utilizando dos MUX 4:1 y un inversor.
6. ¿Qué tipo de subsistema se describe en el código verilog que se muestra a continuación?. Dibuje su diagrama de bloques.

```

module circuitoexamen(
    input signed [WIDTH-1:0] a,
    input signed [WIDTH-1:0] b,
    input [2:0] op,
    input cin,
    output signed [WIDTH-1:0] f,
    output ov,
    output cout
);
parameter WIDTH = 8;
reg f, ov;
always @*
begin
    ov = 0;
    cout = 0;
    if (op[2] == 0)
    begin :
        reg signed [WIDTH:0] s;
        case (op[1:0])
            2'b00: s = a + cin;
            2'b01: s = a + b + cin;
            2'b10: s = a + (~b) + cin;
            2'b11: s = a - 1 + cin;
        endcase
// Cálculo del desbordamiento
        ov = (s[WIDTH] == s[WIDTH-1])? 0: 1;
// Salidas
        f = s[WIDTH-1:0];
        cout = s[WIDTH];
    end
    else
    begin
        case (op[1:0])
            2'b00: f = a & b; // AND
            2'b01: f = a | b; // OR
            2'b10: f = ~a; // NOT
            2'b11: f = a ^ b; // XOR
        endcase
    end
end // always
endmodule

```

NOTA: todos los problemas puntúan igual.

Nombre: .....

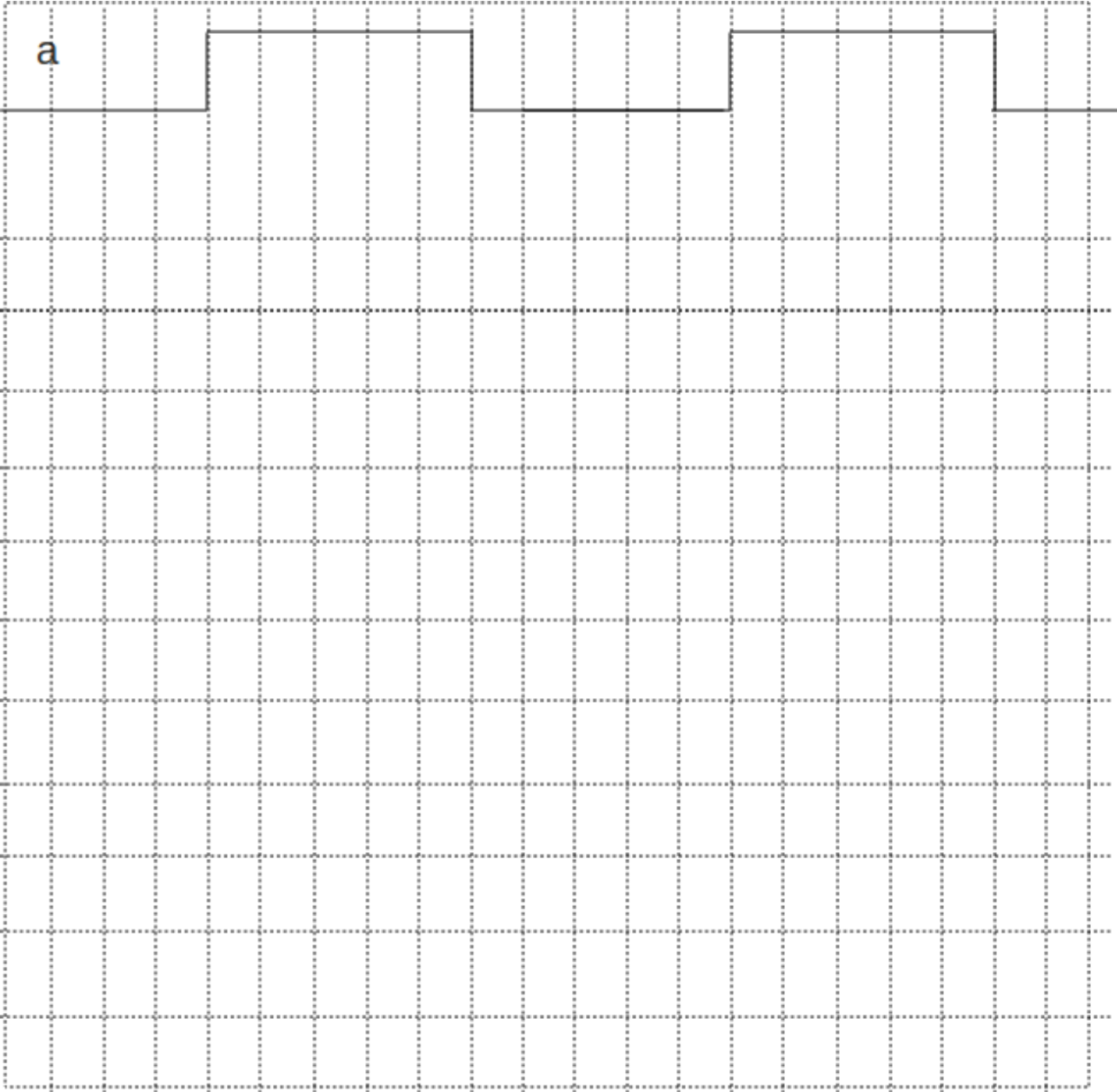


Figura 3