

Los siguientes problemas son solo ejemplos y no deben tomarse como indicativos del examen final. Intentan proveer practica adicional a los problemas del libro. La mejor forma de prepararse para el examen es estudiando el material de los capítulos 1 al 7, los ejemplos del libro y los problemas de practica que corresponden a cada capitulo.

1. Determine el equivalente en decimal del numero binario 10111110100
 - a) si es un numero sin signo,
 - b) si esta escrito en complemento de 2,
 - c) si esta escrito en BCD, y
 - d) representa la parte fraccional de un numero. Puede expresar el resultado como una suma de potencias de dos.
2. Sume los números binarios 101110100 y 111010011. Efectúe todas las operaciones en el sistema binario.
3. Exprese el numero decimal 456 en
 - a) binario
 - b) octal
 - c) hexadecimal
 - d) BCD

4. Demuestre que

$$A'C' + ABC + AC' = AB + C'$$

usando los postulados y teoremas del álgebra booleana.

5. Utilice un mapa-K para simplificar la siguiente función de 4 variables

$$A'B'C'D' + AC'D' + B'CD' + A'BCD + BC'D$$

usando

- a) los unos del mapa para producir una suma de productos,
 - b) los ceros del mapa para producir un producto de sumas,
 - c) los ceros del mapa para producir una suma de productos, y
 - d) los unos del mapa para producir un producto de sumas.
6. Exprese la expresión del problema anterior como una suma de *minterms*.
 7. Muestre como implementar la función del problema 5 usando un multiplexer.
 8. Muestre como implementar la función del problema 5 usando un decoder y compuertas OR.
 9. Diseñe un circuito con 4 bits de entrada y uno de salida. La entrada esta compuesta de dos números de dos bits $A = a_1a_0$ y $B = b_1b_0$. La salida Y debe ser un 1-logico cuando A es mayor que B. Use mapas-K para minimizar el circuito y expresar la salida como una función estándar de las entradas.
 10. Diseñe un circuito que detecte la secuencia 010 en su única entrada serial de 1 bit. Use
 - a) una maquina de estados finitos de Moore y flip-flops tipo JK, y
 - b) una maquina de estados finitos de Mealy y flip-flops tipo T.
 11. Diseñe un contador que repita la siguiente secuencia binaria: 0, 1, 2, 3, 4, 5, 6. Use flip-flops tipo T y compuertas AND, OR y NOT según necesario.
 12. Repita el problema anterior pero utilice un ROM en lugar de compuertas.

13. Una palabra que utiliza el código de Hamming es leída de la memoria. Determine la data original almacenada si la data leída es:

101111110100

y se asume que no ocurrió mas de un error.

14. Escriba la tabla de programación de un PLA para un circuito que convierta de BCD al código *excess-3* descrito en el capítulo 1 del libro de texto.
15. Repita el problema anterior usando un PAL.
16. Solucione el problema 6.21 del libro de texto.
17. Solucione el problema 6.22 del libro de texto, utilizando el contador de la figura 6.14 como un modulo similar al que se muestra en la figura 6.15.
18. El siguiente código de verilog es usado para simular un contador binario. Analice el programa y determine cual será la salida del contador *A* y el *carry CO* desde t=0 hasta t=155ns.

```
//Binary counter with parallel load
module counter (Count,Load,IN,CLK,Clr,A,CO);
    input Count,Load,CLK,Clr;
    input [3:0] IN;           //Data input
    output CO;               //Output carry
    output [3:0] A;         //Data output
    reg [3:0] A;
    assign CO = Count & ~Load & (A == 4'b1111);
    always @ (posedge CLK or negedge Clr)
        if (~Clr) A = 4'b0000;
        else if (Load) A = IN;
        else if (Count) A = A + 1'b1;
        else A = A;          // no change, default condition
endmodule

// Stimulus for testing counter
module testcounter;

    reg Count, Load, CLK, Clr;
    reg [3:0] IN ;
    wire CO ;
    wire [3:0] A ;
    counter cnt (Count, Load, IN, CLK, Clr, A, CO)
    always
#5 CLK = ~CLK ;
        initial
            begin
Clr = 0;
CLK = 1;
Load = 0; Count = 1;
        #5 Clr = 1;
        #50 Load = 1; IN = 4'b1100;
            #10 Load = 0;

        #70 Count = 0;
        #20 $finish;
            end
endmodule
```