

### **Representación en Punto fijo (Parte 1): Números sin signo. Lógica y compuertas (Parte 1): Funciones lógicas elementales. Puertas lógicas.**

**Objetivos de la práctica:** que el alumno sea capaz de:

- Representar e interpretar números sin signo.
- Realizar operaciones aritméticas e interpretar los flags de acarreo y cero.
- Realizar operaciones lógicas
- Usar máscaras y realizar equivalencias entre operaciones sucesivas.
- Establecer la salida de circuitos combinatorios simples.
- Confeccionar tablas de verdad.
- Describir la relación entre entradas y salidas por ecuaciones.

**Bibliografía:**

- “Organización y Arquitectura de Computadoras” de W. Stallings, capítulo 8.
- Apunte 1 de la cátedra, “Sistemas de Numeración: Sistemas Enteros y Punto Fijo”.
- “Principios de Arquitectura de Computadoras” de Miles J. Murdocca, apéndice A, pág. 441.
- Apunte 3 de la cátedra, “Sistemas de Numeración: Operaciones Lógicas”.

### **Representación de números en Punto Fijo. Operaciones Aritméticas**

1. Represente los siguientes números en el sistema BSS restringido a 8 bits. En los casos que no se pueda representar, aclarar por qué.  
**0; 1; 127; 128; 255; 256; -1; -8; -127; -128; -199; -256; 137; 35; 100; -100; 0,5; 1,25.**
2. Interprete las siguientes cadenas de 8 bits en el sistema BSS.  
**00000000; 01010101; 10000000; 11111110; 11111111; 10101010; 01111111; 01100110**
3. Calcule el rango y resolución de un sistema de punto fijo en BSS con 6 bits de parte entera y 4 de fraccionaria.
4. Represente los siguientes números en el sistema del ejercicio 3. Si no es posible obtener una representación exacta, indique cuál es la más próxima y calcule en ese caso el error cometido. Si el número a representar está fuera del rango del sistema, señale que ese número “NO SE PUEDE REPRESENTAR”.  
**3; 5,25; 1,2; 2,001; 23,125; 62,0625; 1,0625; 35**
5. Interprete las siguientes cadenas en el sistema del ejercicio 3.  
**0000000000; 0101010101; 1000000000; 1111111110; 1111111111; 1010101010; 0111111111; 0110110110**
6. Calcule el resultado de realizar las sumas (ADD) y restas (SUB) indicadas a continuación. Calcule el valor en el que quedarán los flags luego de realizada cada operación, de acuerdo a que haya habido acarreo (flag C, de Carry) o se haya producido borrow (flag B, es el mismo que C pero en la resta), o que el resultado sea cero en todos sus bits (flag Z, de Zero).  
**ADD 00011101 00011011; ADD 01110000 11110001; SUB 00011101 00011011; SUB 01110000 11110001**  
**ADD 10011101 01110010; ADD 01001100 01110000; SUB 10011101 01110010; SUB 01001100 01110000**  
**ADD 01110110 01110001; ADD 11001100 11110000; SUB 01110110 01110001; SUB 11001100 11110000**  
**ADD 10111001 11100011; ADD 10000000 10000000; SUB 10111001 11100011; SUB 10000000 10000000**  
**ADD 00111010 00001111; ADD 00000000 10000000; SUB 00111010 00001111; SUB 00000000 10000000**  
Recuerde que:  

0+0=0 con C=0	1+0=1 con C=0	0-0=0 con B=0	1-1=0 con B=0
0+1=1 con C=0	1+1=0 con C=1	1-0=1 con B=0	0-1=1 con B=1
7. Suponga que los operandos del ejercicio anterior (ej. 6) eran números representados en BSS. Determine si el resultado obtenido es correcto. Para eso, interprete en BSS tanto los operandos como el resultado de cada operación y luego compare con el resultado esperado trabajando en decimal. En caso de que la operación haya dado resultado incorrecto, indicar la/s posible/s cadena/s de bits que representa/n el resultado correcto.

# Organización de Computadoras 2015

8. Referido al ejercicio 7 sobre la operación ADD: Observando cuáles resultados fueron correctos y cuáles fueron incorrectos y relacionándolos con los flags, describa una regla para determinar la correctitud de la operación ADD en el sistema BSS con la mera observación de los flags (sin verificar la operación pasando por el sistema decimal).
9. Trabaje de forma similar al ejercicio 8 pero con la operación SUB.
10. Considere en el ejercicio 6, que el punto o coma fraccionaria se encuentra entre el bit 2 y el 3. Interprete el valor que tendrán las cadenas de bits que representan los operandos y los resultados como BSS. Observe los flags. ¿Qué concluye?
11. Escriba las cadenas de los sistemas BSS restringido a 4 bits. Considere el punto (o coma fraccionaria) fijo en cada una de todas las posibles posiciones (son 5 posibilidades en total, considerando que el punto fijo puede estar colocado a la izquierda del MSB y a la derecha del LSB) y obtenga el rango y resolución de cada uno de los sistemas de punto fijo resultantes. ¿Cuántas cadenas se pueden escribir en cada caso? ¿Cuántos números se pueden representar en los distintos sistemas?
12. Represente los números **0, 1, 2, 9, 10, 11, 20, 34, 99, 100 y 1220** en los sistemas BCD y BCD empaquetado. Describa, con el mayor nivel de detalle posible, un procedimiento para calcular sumas en BCD. Sin considerar representación de signo, realice las siguientes operaciones en BCD: **20 + 34; 34 + 99; 1220 + 880**
13. Escriba los números **21398, 2183, 972 y 89737** en los sistemas BCD, BCD empaquetado y BSS. Observe la cantidad de bits necesarios. ¿Qué conclusiones saca respecto de las ventajas y desventajas del sistema BCD sobre BSS?
14. Haga el pasaje de binario a hexadecimal y de hexadecimal a BCH en forma directa (sin utilizar sistema decimal). ¿Por qué cree que el sistema hexadecimal es muy utilizado?

Binario a Hexadecimal	
<b>1001010010000</b>	
<b>11010010101011</b>	
<b>101011011001101</b>	
<b>1001111000100011</b>	
<b>1100101011101010</b>	
<b>11100101011011</b>	

Hexadecimal a BCH	
<b>1290</b>	
<b>34AB</b>	
<b>56CD</b>	
<b>E7F8</b>	
<b>8D71</b>	
<b>123B</b>	

## Operaciones Lógicas

15. Realizar las siguientes operaciones lógicas:
 

a. <b>10101100 AND 11000101</b>	f. <b>00100010 XOR 11111101</b>	k. <b>10101110 NOR 11010101</b>
b. <b>00100010 AND 11111101</b>	g. <b>NOT 00101100</b>	l. <b>00101010 NOR 11100101</b>
c. <b>10101100 OR 11000101</b>	h. <b>NOT 11000101</b>	m. <b>10101100 XNOR 11000101</b>
d. <b>00100010 OR 11111101</b>	i. <b>10101011 NAND 11000101</b>	n. <b>10101100 XNOR 11111101</b>
e. <b>10101100 XOR 11000101</b>	j. <b>00100010 NAND 11111101</b>	
16. Dado un byte  $X=[X_7, X_6, X_5, X_4, X_3, X_2, X_1, X_0]$  (indeterminado), ¿qué resultado obtendré al aplicarle una operación lógica junto a un valor predeterminado (máscara)? Analice para cada operación cómo los bits de la ‘máscara’ condicionan el resultado que se obtendrá.
 

a. <b>X OR 10101010</b> b. <b>X OR 11111000</b> c. <b>X AND 10101010</b> d. <b>X AND 10001111</b> e. <b>X XOR 10101010</b> f. <b>X XOR 00001111</b> g. <b>X OR 10000000</b> , al resultado <b>AND 11110000</b> , y al resultado <b>XOR 00011110</b>	h. <b>X AND 10101111</b> , al resultado <b>OR 11110000</b> , y al resultado <b>XOR 00011110</b> i. <b>X XOR 10101010</b> , al resultado <b>AND 11110000</b> , y al resultado <b>OR 00011110</b> j. <b>X XNOR 10101010</b> , al resultado <b>NAND 11110000</b> , y al resultado <b>NOR 00011110</b> k. <b>X XOR 10101010</b> , al resultado <b>NAND 11110000</b> , y al resultado <b>NOR 00011110</b>
---	---

En los casos de más de una operación, obtenga el resultado y a ése resultado aplíquelo la operación siguiente.

17. Complete con el operador lógico adecuado (AND, OR, XOR, NOT) las siguientes expresiones de modo tal que se cumpla la igualdad propuesta:
 

a. <b>1000 ...¿op?... 1101 = 1101</b>	b. <b>1111 ...¿op?... 0101 = 0101</b>
---------------------------------------	---------------------------------------

# Organización de Computadoras 2015

c.  $1101 \dots \text{¿op?} \dots 1001 = 0100$

d.  $\dots \text{¿op?} \dots (1111 \dots \text{¿op?} \dots 0011) = 1100$

e.  $X_3 X_2 X_1 X_0 \dots \text{¿op?} \dots 1110 \dots \text{¿op?} \dots 0101 \dots \text{¿op?} \dots 0101 = X_3 0 X_1 0$

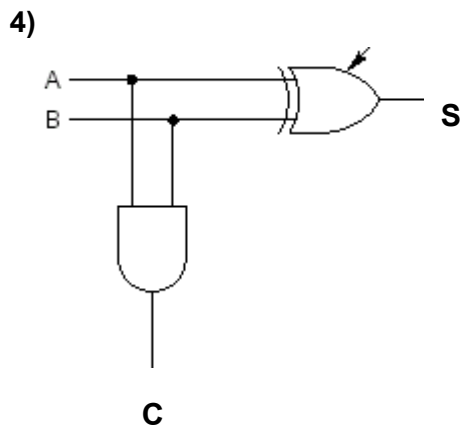
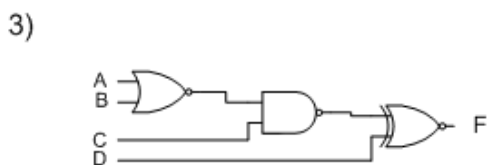
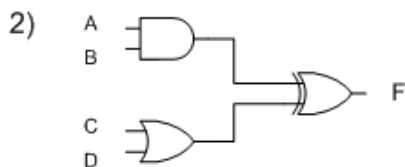
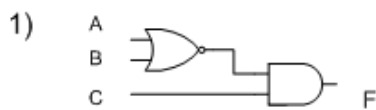
Se entiende que cada X es un bit que puede ser 1 o 0, debiendo obtenerse el resultado final combinando diferentes operaciones lógicas en un orden correcto.

18. Dado un byte  $X=[X_7, X_6, X_5, X_4, X_3, X_2, X_1, X_0]$  (indeterminado), aplíquese operaciones lógicas (1 o más) con un byte MASK, que deberá también determinar, para lograr los siguientes efectos:

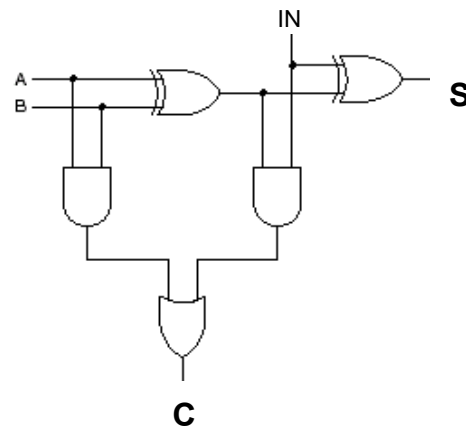
- Poner a 1 los bits 0, 2 y 6 dejando los demás inalterados.
- Poner a 1 los bits 4 y 7 dejando los demás inalterados.
- Poner a 0 los bits 0, 2 y 6 dejando los demás inalterados.
- Poner a 0 los bits 4 y 7 dejando los demás inalterados.
- Cambiar los bits 0, 2 y 6 dejando los demás inalterados.
- Cambiar los bits 4 y 7 dejando los demás inalterados.
- Poner el bit 3 en 1, el bit 6 en 0, cambiar el bit 2 y dejar los demás inalterados.
- Poner a 0 los bits 0, 3 y 7, cambiar el bit 2 y dejar los demás inalterados.

## Circuitos Combinatorios

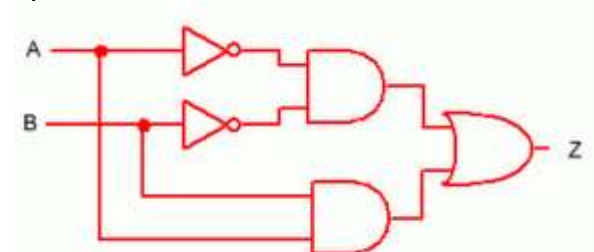
19. Construir la tabla de verdad de los siguientes circuitos. Especifique además la ecuación que describe la relación entre entradas-salidas.



5)



6)



7)

