

Certamen 1 de Arquitectura de Computadores Semestre 2019/2

Nombre: Soluciones Propuestas Puntaje: /100 pts

Realice las operaciones indicadas en cada caso:

- Convertir a octal y hexadecimal los siguientes números decimales 63 y 29. (10 pts)
- Calcule en binario los resultados de las siguientes operaciones $+29 + (-63)$ y $-29 + (+63)$. Recuerde que los números deben ser representados en complemento 2 y asume que los números se representan con 8 bits incluido el de signo. (15 pts)
- Explique el propósito de un multiplexor 4:1. Indique cuantos bits de entrada posee, número de bits de selección y número de bits de salida. (15 pts)

a) Convertimos primero los números a binarios Pase 1pt

$$63 = 64 - 1 = 1000000 - 1 = 0111111 \Rightarrow$$

$$29 = 32 - 3 = 100000 - 011 = 11101$$

octal

$$111111 \Rightarrow 77_8 \Rightarrow 63_{10}$$

$$011101 \Rightarrow 35_8 \Rightarrow 29_{10}$$

hexadecimal

$$00111111 \Rightarrow 3F_H \Rightarrow 63_{10}$$

$$00011101 \Rightarrow 1D_H \Rightarrow 29_{10}$$

b) $+63 = 0111111$ ambos expresados en 7 bits

$$+29 = 0011101$$

Calculamos la representación de -63 y -29 en complemento de 2. (5pts)

$+63 = 0111111$

inversión 1000000

$$+1$$

$$-63 = 1000001$$

$+29 = 0011101$

inversión 1100010

$$+1$$

$$-29 = 1100011$$

$-29 + 63$	$+29 - 63$
1100011 0111111 $\hline 0100010$ $\Rightarrow 34$	0011101 1000001 $\hline 1011110$ $(-)$ 1011110 0100001 $\hline 100010$ $\Rightarrow -34$

El multiplexor 4:1 permite seleccionar 1 entrada de 4 posibles (de 1 bit) y permitir su paso a la salida (de 1 bit). Para la selección se requieren de 2 bits de selección (5pts).

2. La empresa Sofanor Rodríguez Inc le solicita que diseñe un circuito digital de 4 bits que permita identificar si un número entre 0 y 9 es un número primo (solo divisible por 1 y por si mismo). Si el número es primo, la única salida del circuito enciende un led, en caso contrario el led se apaga.

Para ello realice:

a. Determine y etiquete las entradas y salidas del circuito y luego encuentre tabla de verdad del circuito solicitado. (10 pts)

- Identificar números entre 0 y 9 requiere de 4 bits \Rightarrow 4 entradas (A, B, C, D)
 - La salida requiere señalar si número es primo o no \Rightarrow 1 salida (1 bit)
 \Rightarrow Tabla de verdad

	A	B	C	D	Z
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0

	A	B	C	D	Z
10	1	0	1	0	X
11	1	0	1	1	X
12	1	1	0	0	X
13	1	1	0	1	X
14	1	1	1	0	X
15	1	1	1	1	X

salida = 1 \Rightarrow primo
 0 \Rightarrow no-primo

b. Determine mapa de Karnaugh del circuito, con ella determine implicants (10 pts)

AB \ CD	00	01	11	10
00	0	0	1	1
01	0	1	1	0
11	X	X	X	X
10	0	0	X	X

buscaremos implicants que encierran "1"s.

"A"

"B"

los implicants

aprovechamos

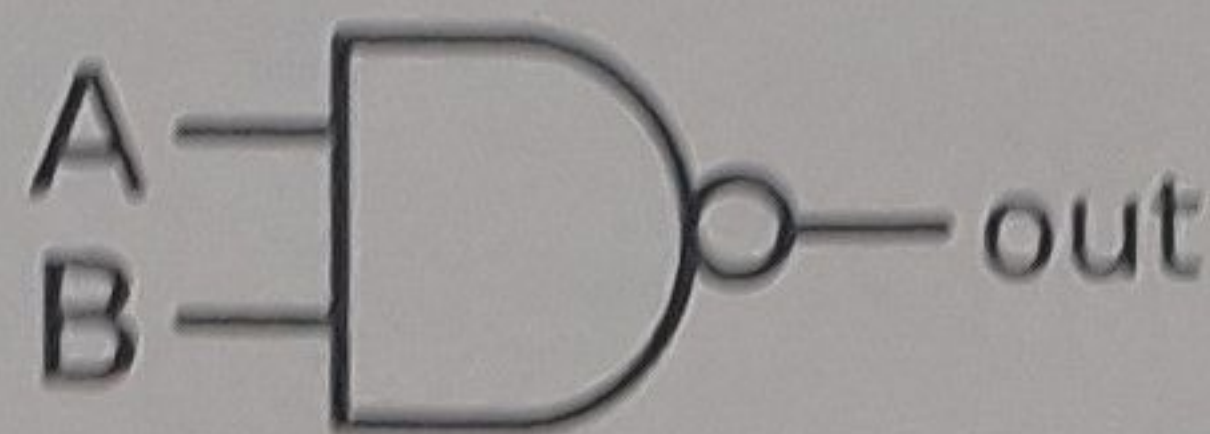
implicants

c. Proporcione expresión algebraica que se obtiene al aplicar la técnica de minimización del mapa de Karnaugh. (15 pts)

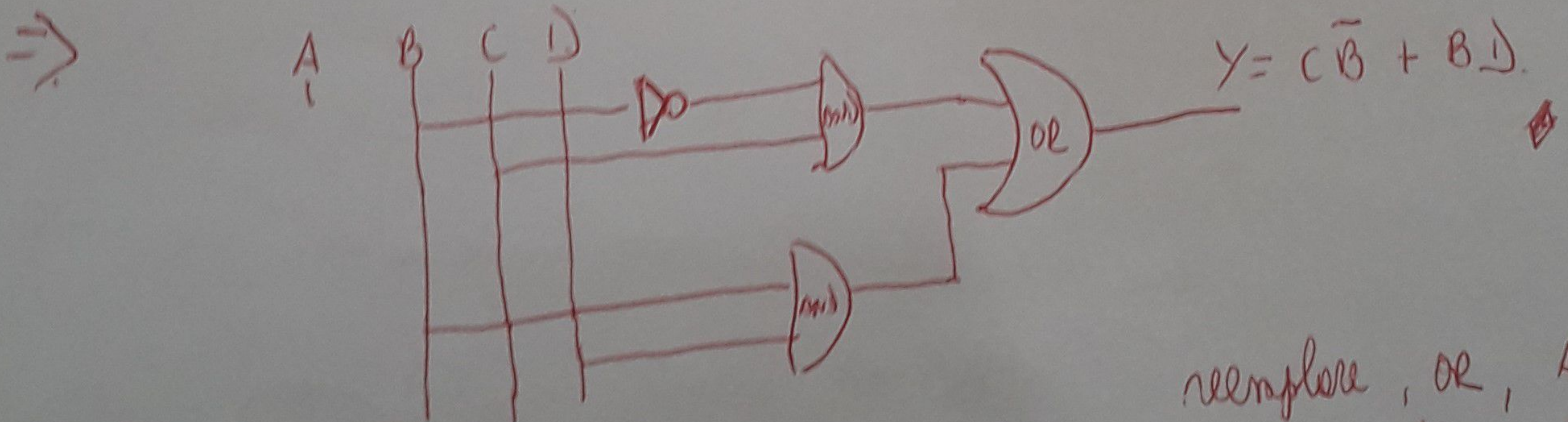
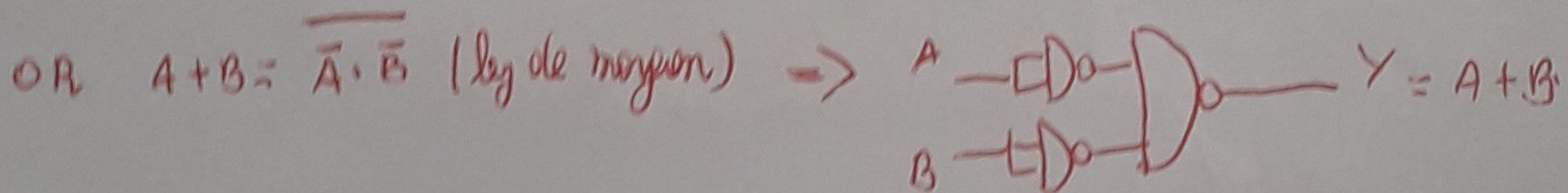
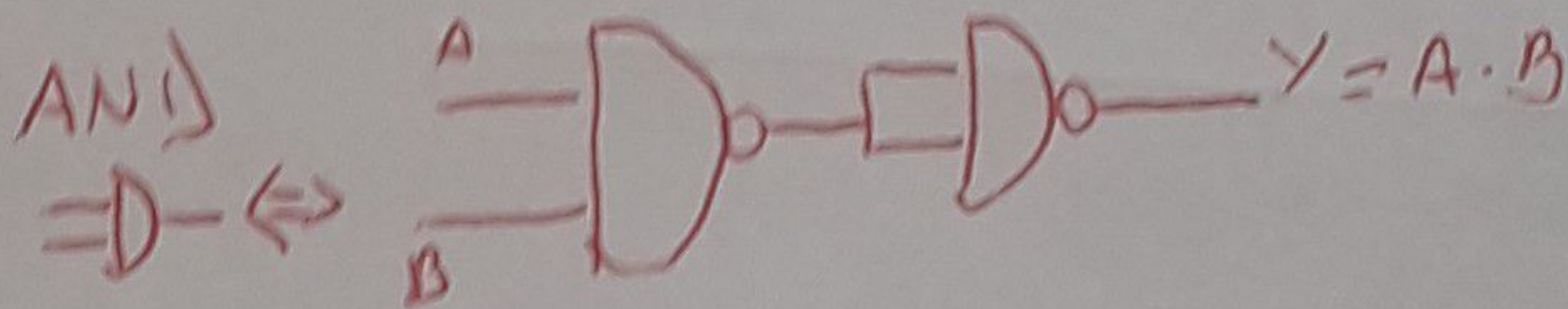
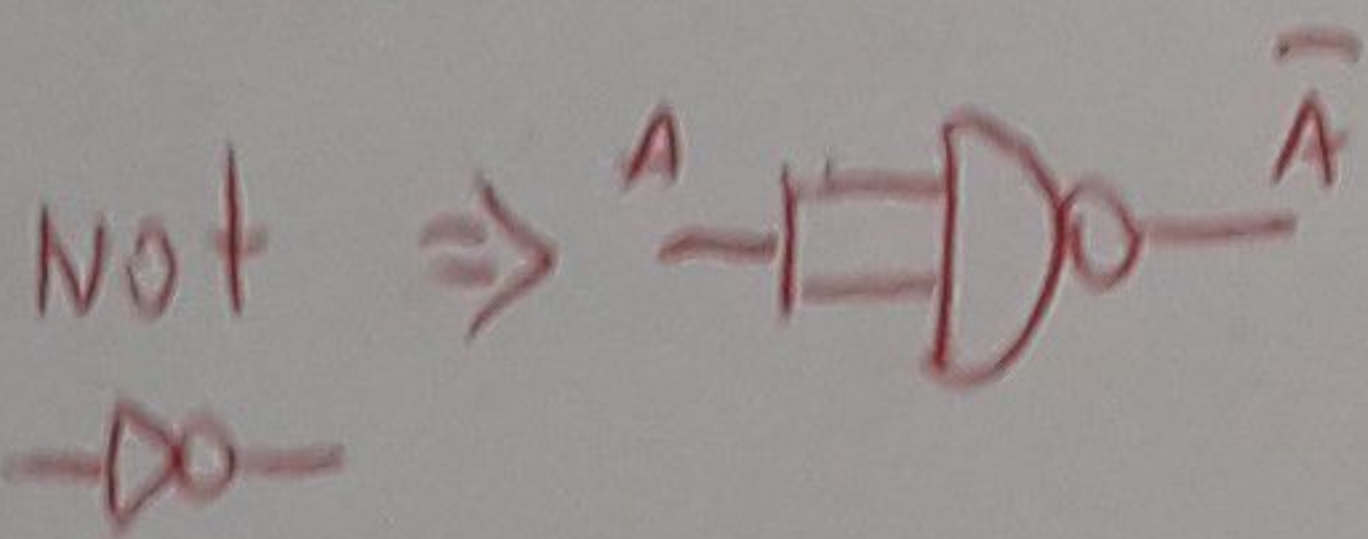
Del mapa de Karnaugh obtenemos suma de productos

$$Y = \hat{A} + \hat{B} = C\bar{B} + BD$$

d. Dibuje circuito obtenido empleando solo compuertas NAND. (10 Pts)



Mostremos como se implementan los NOT, OR y AND con compuertas NAND



reemplazare, OR, AND, y NOT como se indicio en las conversión con los NAND

3. Sofanor, estudiante autodidacta que no ha tomado Arquitectura de Computadores, ejecuta en su laptop el código C que se muestra en columna adyacente. Sofanor observa en pantalla una serie de valores y el programa TERMINA sin mostrar ningún error. Sofanor no entiende el por que de este hecho ya que el considera que el código C esta correcto y por tanto le pide ayuda.

```
#include <stdio.h>
#include <stdint.h>
int main()
{
    int4_t value=0;
    while(value>=0){
        printf("%d\n",value);
        value=value+1;
    }
    printf("%d\n",value);
    return 0;
}
```

Muestre que valores son los observados por Sofanor en pantalla y cual es el valor que muestra el último printf(). De acuerdo a estos valores presente una explicación del porque del comportamiento del código C (15 pts)

De la definición de la variable `value`, sabemos que es un entero con signo de 4 bits (`int4_t`) por tanto los valores que observo Sofanor fueron:

0, 1, 2, 3, 4, 5, 6, 7, ...

signo \uparrow 0111
+1

negativo \rightarrow 1000

\hookrightarrow el while (condición) se hace falso.

ya que 1000 es la representación en base complemento 2 de -8
(ya que 1000 \Rightarrow ~~menos~~ 0011 $\xrightarrow{+1}$ 1000)

\Rightarrow el último printf imprime
"-8"

