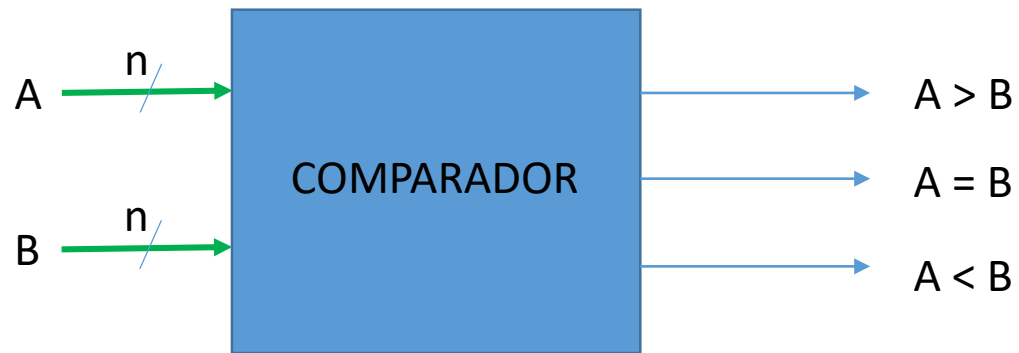


## 7.-DISEÑO DE COMPARADORES

a)Concepto

-SON CIRCUITOS LOGICOS COMBINACIONALES (CLC) QUE COMPARAN INFORMACION BINARIA Y SUS SALIDAS ESTAN EN FUNCION DE LA RELACION DE LAS ENTRADAS



¿RECUERDAN QUE SIGNIFICA  ?

**¡ UN BUS DE n bits !**



EJEMPLO :    DISEÑAR UN COMPARADOR DE DOS PALABRAS A Y B DE 4 BITS CADA UNA Y QUE DETECTE CUANDO A ES MAYOR IGUAL QUE B.

PRIMER PASO (Núm de var E/S)



SEGUNDO PASO (Tab. De verdad)

A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	f $A \geq B$
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	1	0
1	1	1	1	1	1	0	0	1
1	1	1	1	1	1	0	1	1
1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1

256 renglones  
(Combinaciones  
o minterminos)

### TERCER PASO (minimización)

# ¡NO!

ESTA FUNCION BOOLEANA TIENE 256 MINTERMINOS, DE TAL FORMA QUE LA MINIMIZACION CON MAPAS DE KARNAUGH SERIA CON UN MAPA DE 16X16 CUADRITOS, Y SERIA MUY DIFICIL OBSERVAR LA ADIACENCIA, ADEMAS, LA IMPLEMENTACION CON COMPUERTAS (CIRCUITOS SSI) SERIA MUY LABORIOSA.

# ¿ENTONCES?

# ¡TAMPOCO!

LA IMPLEMENTACION CON CIRCUITOS MSI (DEC's Y MUX's), SERIA TAMBIEN UN PROBLEMA, NECESITARIAMOS UN DEC 8X256 O UN MUX 256X1

# ¡OH QUE LA .....!, ¿Y AHORA?...

¡LA SOLUCION ES USAR CIRCUITOS LSI,  
COMO SON LA ROM Y EL PLA, QUE ME  
SIRVEN PARA IMPLEMENTAR VARIAS  
FUNCIONES BOOLEANAS CON UN GRAN  
NUMERO DE VARIABLES!

**PERO ANTES  
DE SEGUIR:**

**TAREA !!!!!!!!!!!!!!!**

**DISEÑAR UN COMPARADOR DE 2  
PALABRAS A Y B DE CUATRO BITS QUE ME  
INDIQUE CUANDO SON IGUALES**

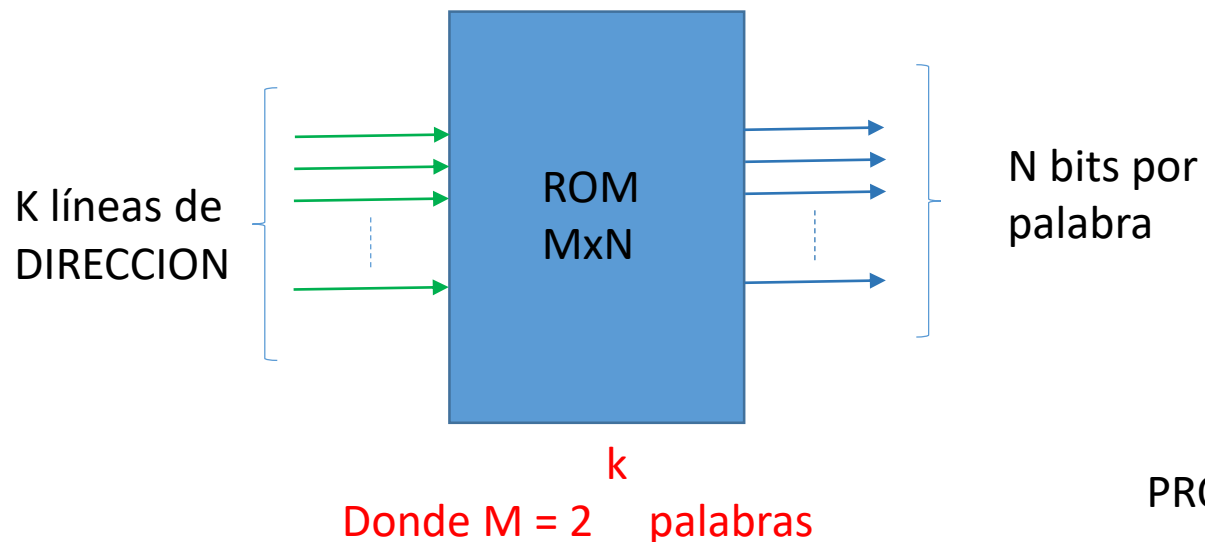
## 8.-DISEÑO CON ROM (READ ONLY MEMORY)

### a) INTRODUCCION

Aunque la ROM es un elemento de memoria (memoria de solo lectura), nos sirve para diseñar circuitos lógicos y para ello es necesario entender a la ROM no solo como un arreglo de celdas binarias (0's y 1's), sino también con su representación como **CIRCUITO LOGICO**, es decir la ROM es un circuito lógico que se puede **PROGRAMAR**.

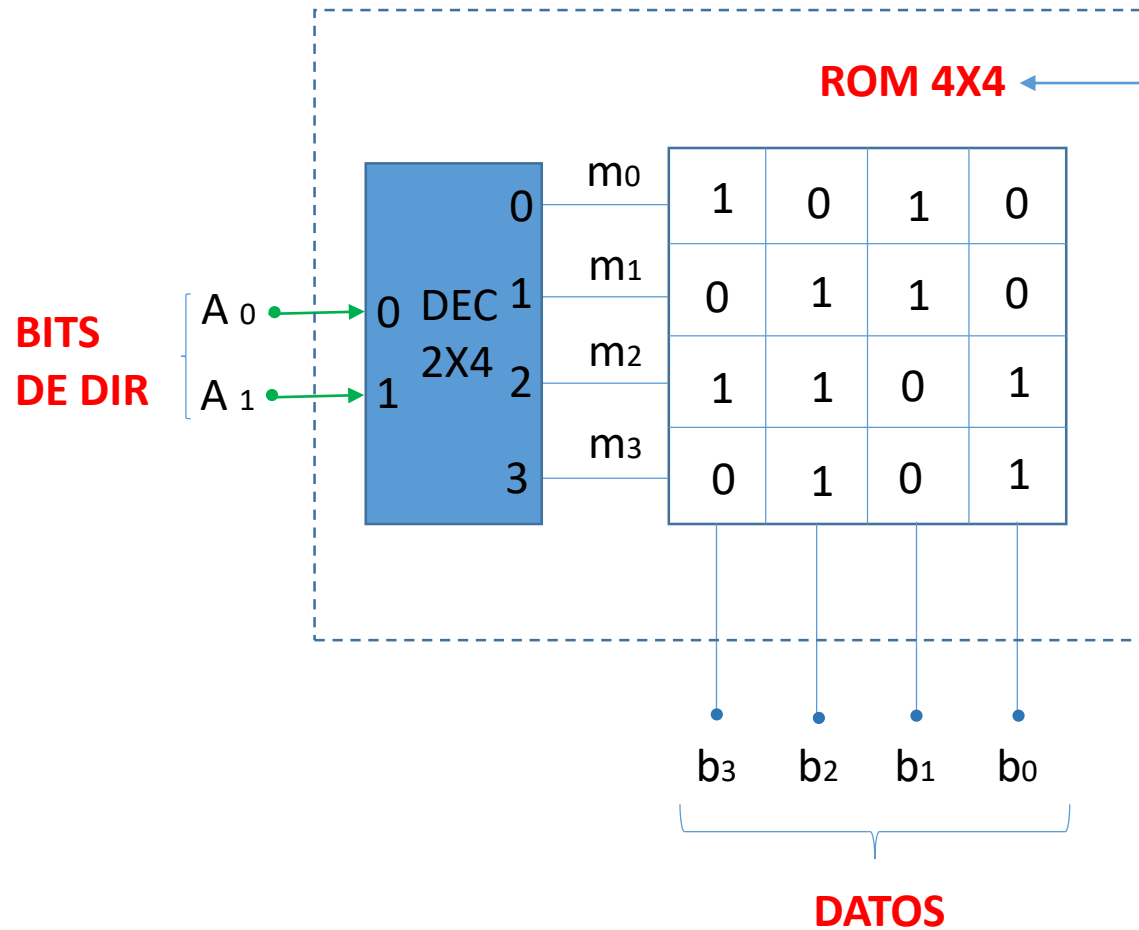
### b) CONCEPTOS

La ROM es un arreglo de celdas binarias organizadas en M palabras de N bits cada una y se requieren K líneas de DIRECCION para seleccionar la M palabras. El tamaño de la ROM estará dado por M x N.



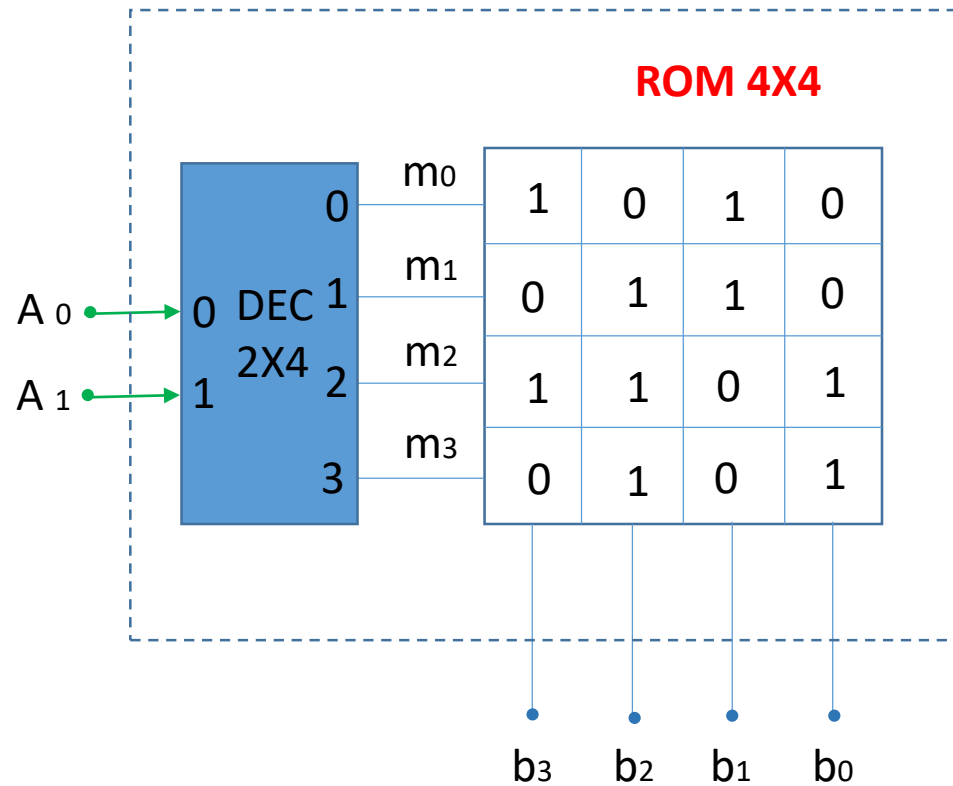
Además, podemos observar con el diagrama de bloques de la ROM, las salidas solo son función de las entradas, por lo tanto la ROM es un circuito combinacional **PROGRAMABLE**.

i) Diagrama Conceptual de la ROM

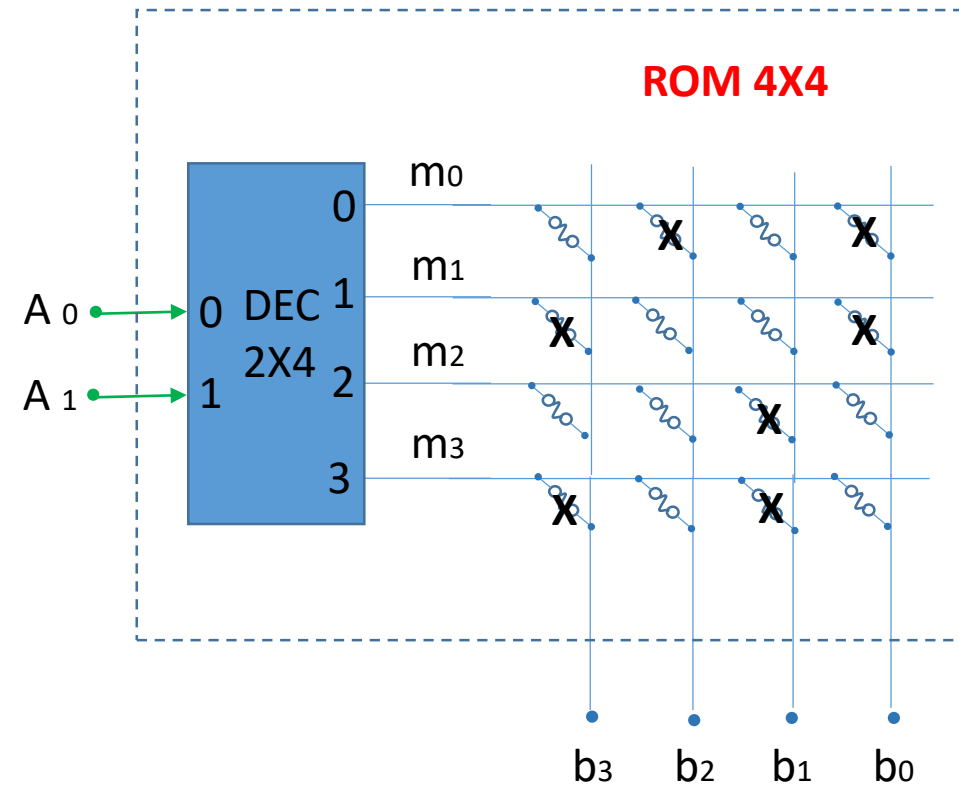


**IMPORTANTE: UNA ROM ESTA DADA POR DIRECCION Y DATOS**

i) Diagrama Conceptual de la ROM



i) Diagrama "Físico" de la ROM

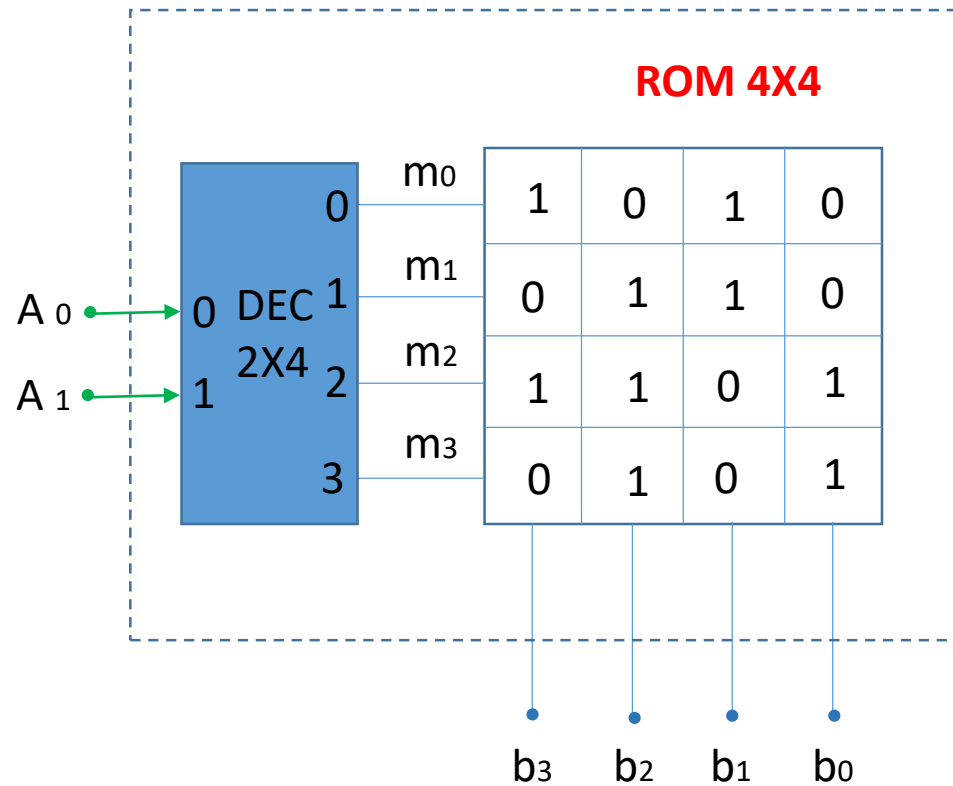


**¡LISTO!**

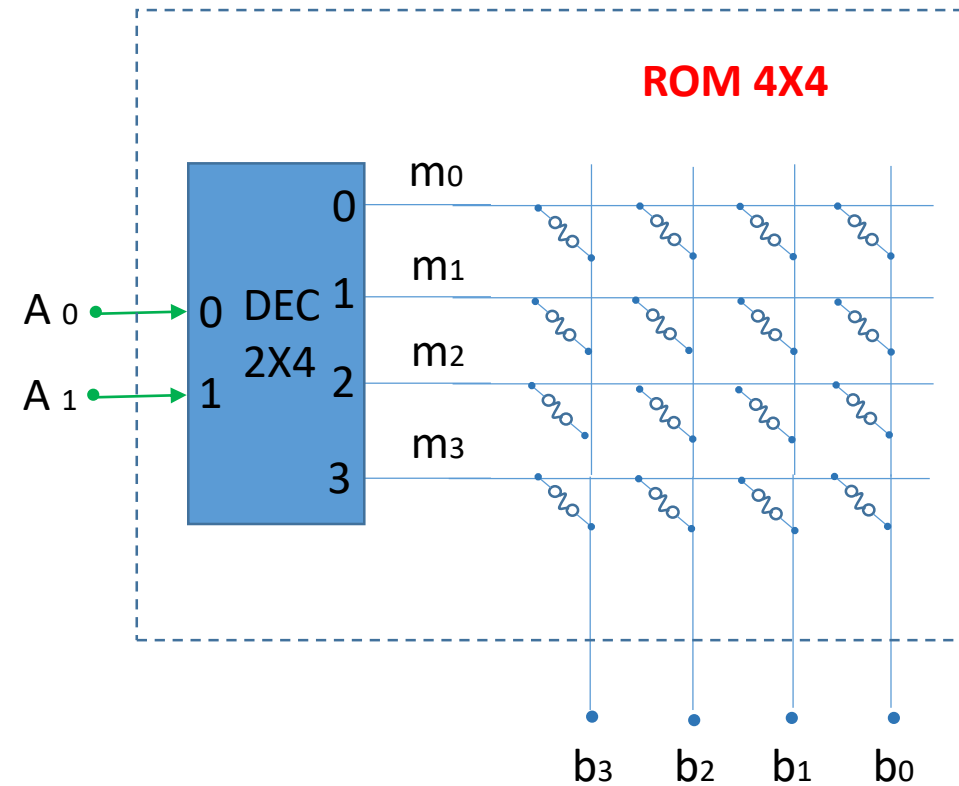
**FUSIBLES  
¡SE QUEMAN!**



i) Diagrama Conceptual de la ROM



i) Diagrama “Físico” de la ROM



**¿PERO COMO SE PROGRAMA?**

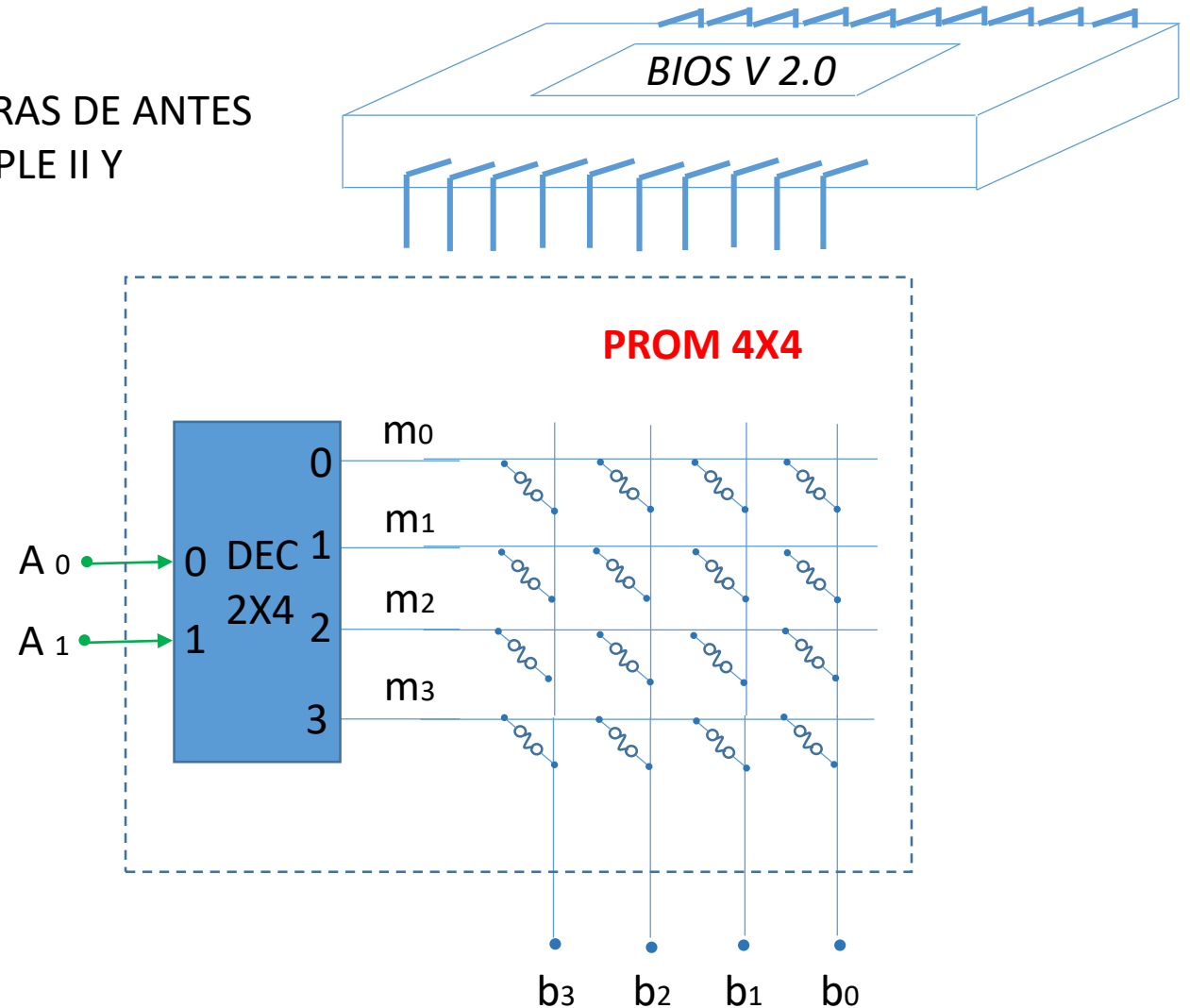
# PARENTESIS

**ROM** (READ ONLY MEMORY)

EJEMPLO: BIOS DE LAS COMPUTADORAS DE ANTES  
(¿SE ACUERDAN DEL EJEMPLO DE APPLE II Y  
PINEAPPLE)

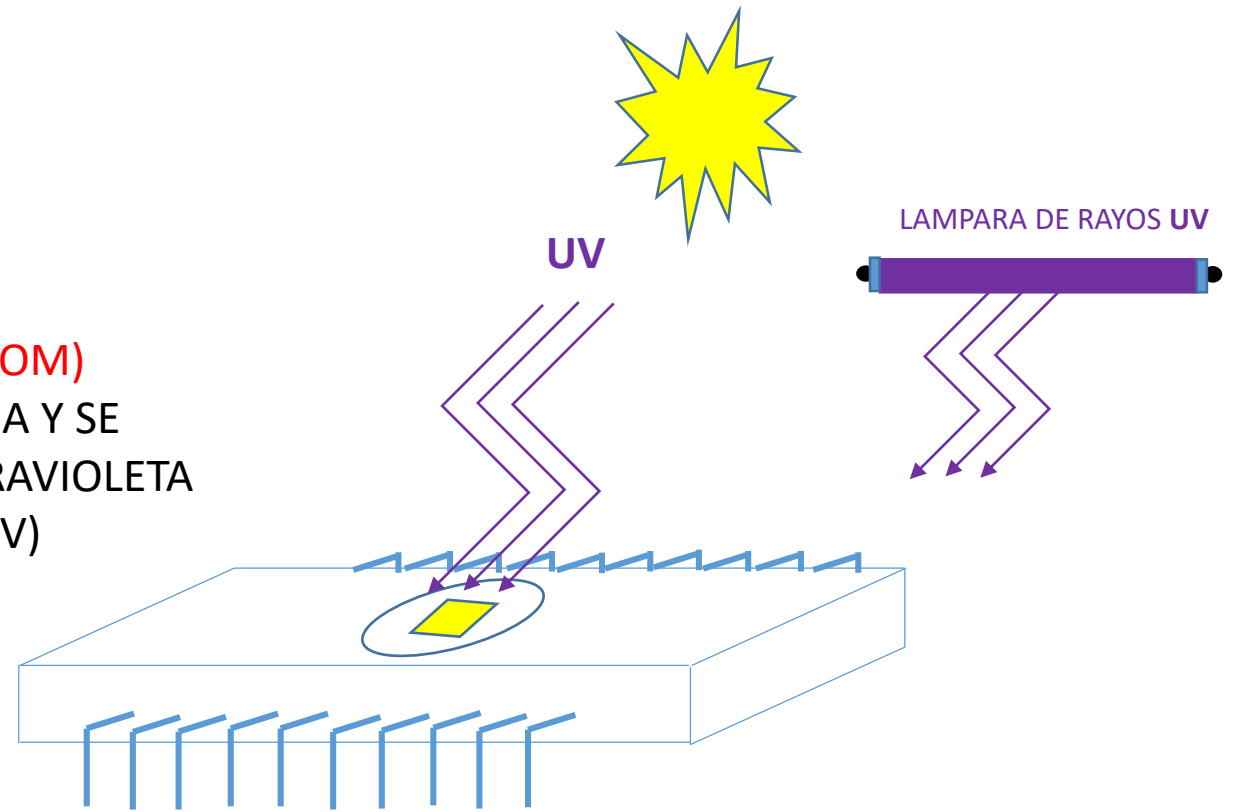
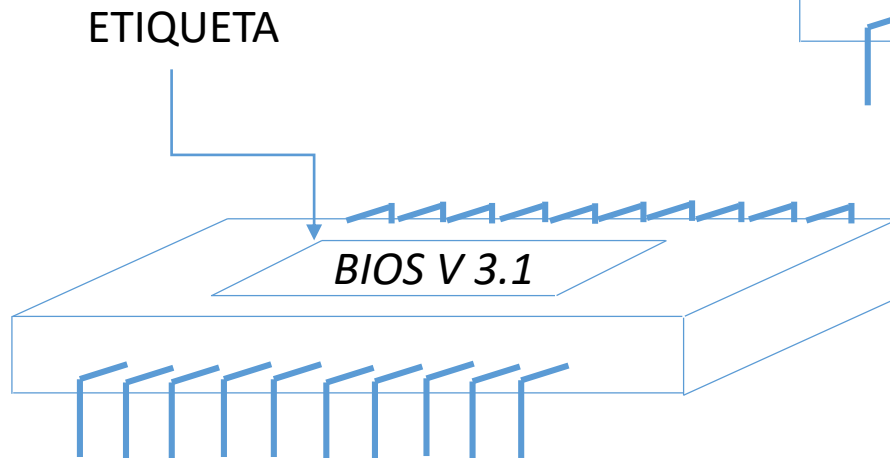
**PROM** (PROGRAMMABLE ROM)

ES UNA ROM QUE SE PUEDE  
PROGRAMAR, NO TIENE  
INFORMACION HAY QUE  
“QUEMARLA”, CONTIENE PUROS “1”



# PARENTESIS

**EPROM (ERASABLE PROGRAMMABLE ROM)**  
ES UNA ROM QUE SE PUEDE PROGRAMAR Y SE  
PUEDE BORRAR MEDIANTE RAYOS ULTRAVIOLETA  
(RAYOS DEL SOL O UN BORRADOR DE UV)



LAS PROM's COMO LAS EPROM's SON COMERCIALES  
Y LAS PODEMOS ENCONTRAR DE 256 X 8, 512 X 8,  
1K X 8, 2K X 8, ..., 32K X 8.

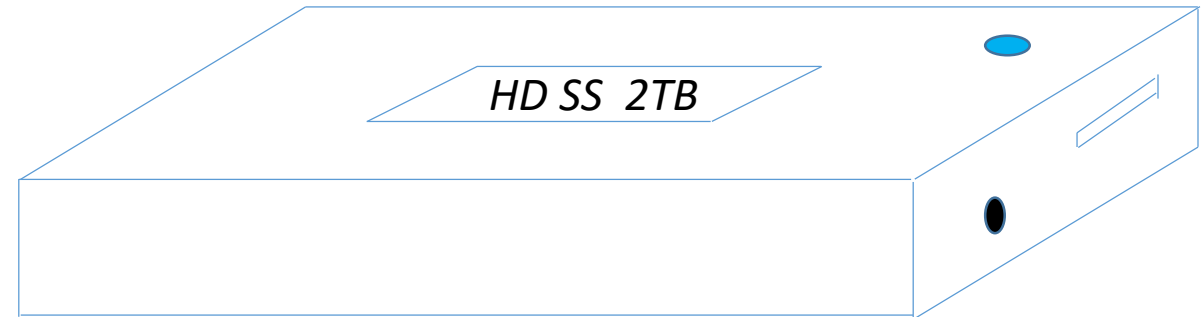
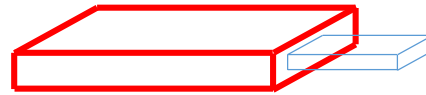
# PARENTESIS

## EEPROM (ELECTRICAL ERASABLE PROGRAMMABLE ROM)

ES UNA ROM QUE SE PUEDE PROGRAMAR, QUE SE PUEDE BORRARSE POR MEDIOS ELECTRICOS (ELECTRONICA)

EJEMPLOS: LA MEMORIA USB Y SUS DISCOS DUROS DE ESTADO SOLIDO.

MEMORIA USB



LAS EEPROM's SON COMERCIALES  
Y LAS PODEMOS ENCONTRAR DE 8GB,...64GB,  
O EN DISCOS DUROS DE HASTA 2TB

# PARENTESIS

## ANALOGIAS

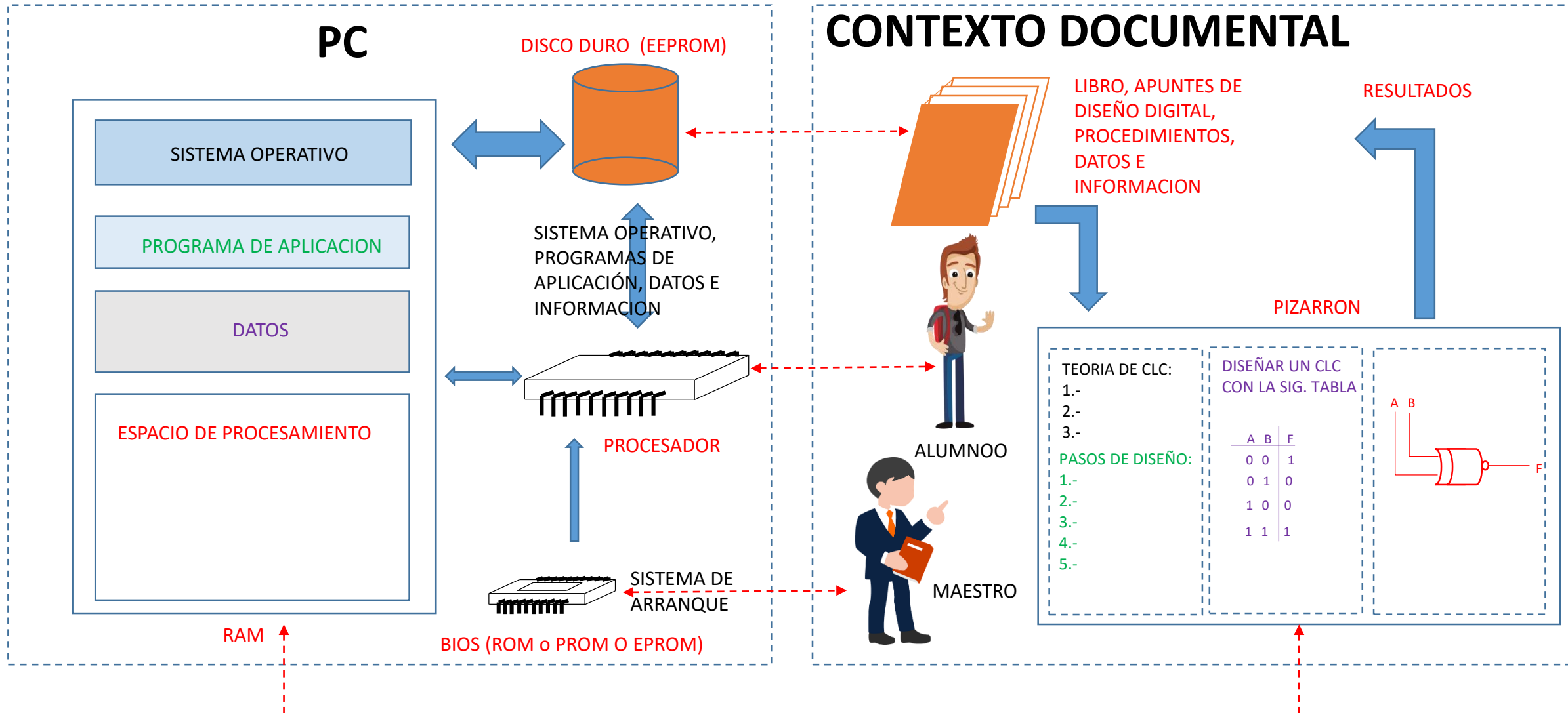
COMPONENTE	CD ROM	DOCUMENTO
<b>ROM</b>	CD DE JOSE JOSE, DVD DE TITANIC	“LADRON DE ESPERANZAS”, “LOS MITOS QUE NOS DIERON TRAUMAS”
<b>PROM</b>	CD R 700MB, DVD R 4.7GB	APUNTES ESCRITOS CON PLUMA
<b>EPROM</b>	CD R/W 700MB, DVD R/W 4.7GB,	APUNTES ESCRITOS CON LAPIZ
<b>EEPROM</b>	MEMORIA USB, DISCOS DUROS DE ESTADO SOLIDO, (FLOPPY DISK, DISCOS DUROS MAGNETICOS)	UNA TABLET KINDLE

# PARENTESIS

COMPONENTE	DOCUMENTO
<b>ROM</b>	“LADRON DE ESPERANZAS”, “LOS MITOS QUE NOS DIERON TRAUMAS”
<b>PROM</b>	APUNTES ESCRITOS CON PLUMA
<b>EPROM</b>	APUNTES ESCRITOS CON LAPIZ
<b>EEPROM</b>	UNA TABLET, KINDLE
<b>RAM (RANDOM ACCESS MEMORY)</b>	<b>EL PIZARRON</b>

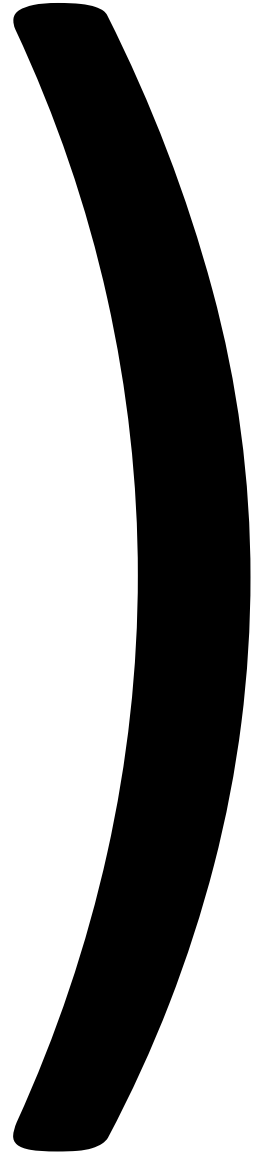
# PARENTESIS

## MAS ANALOGIAS



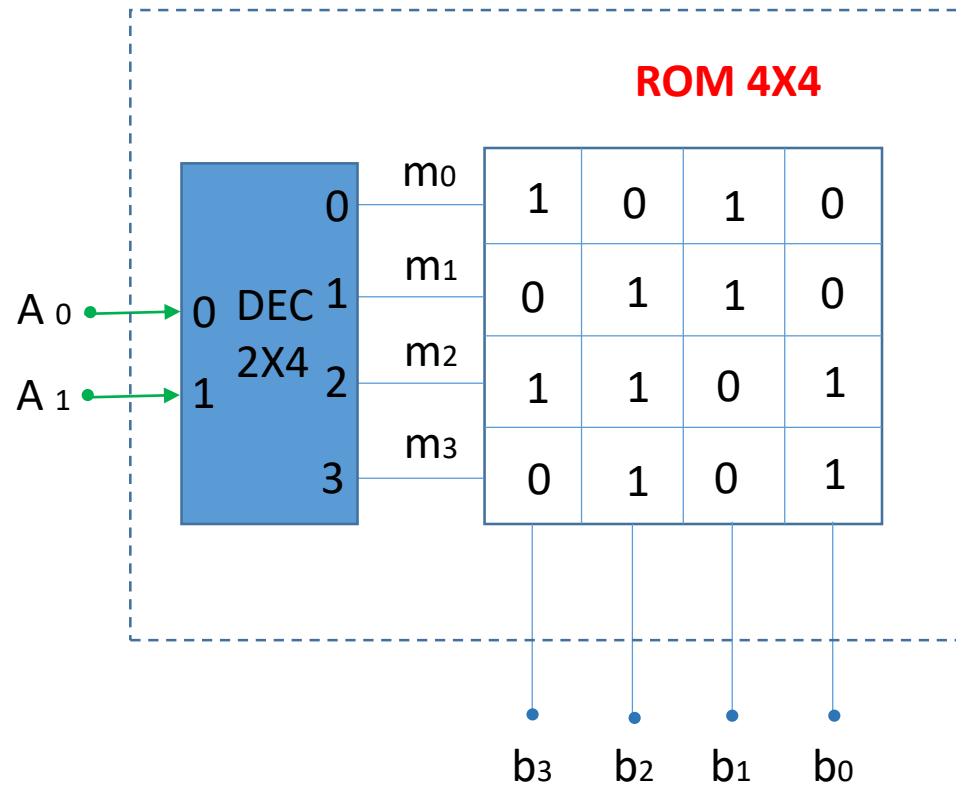
PARENTESIS

FIN DEL  
PARENTESIS!!!!!!!!!!

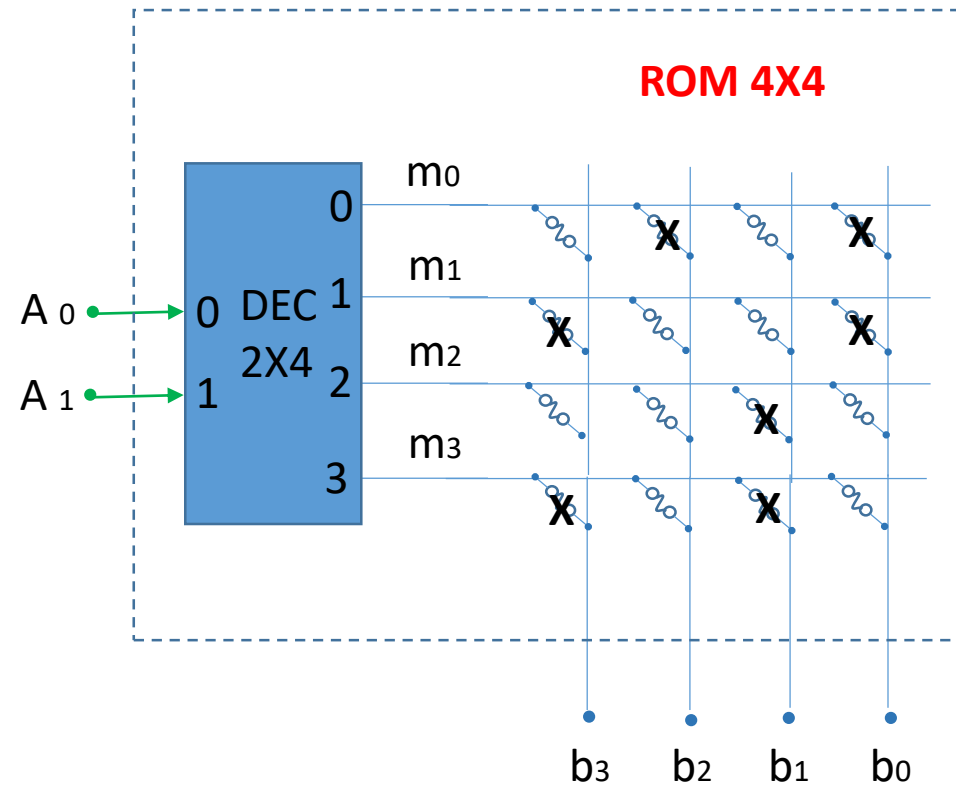




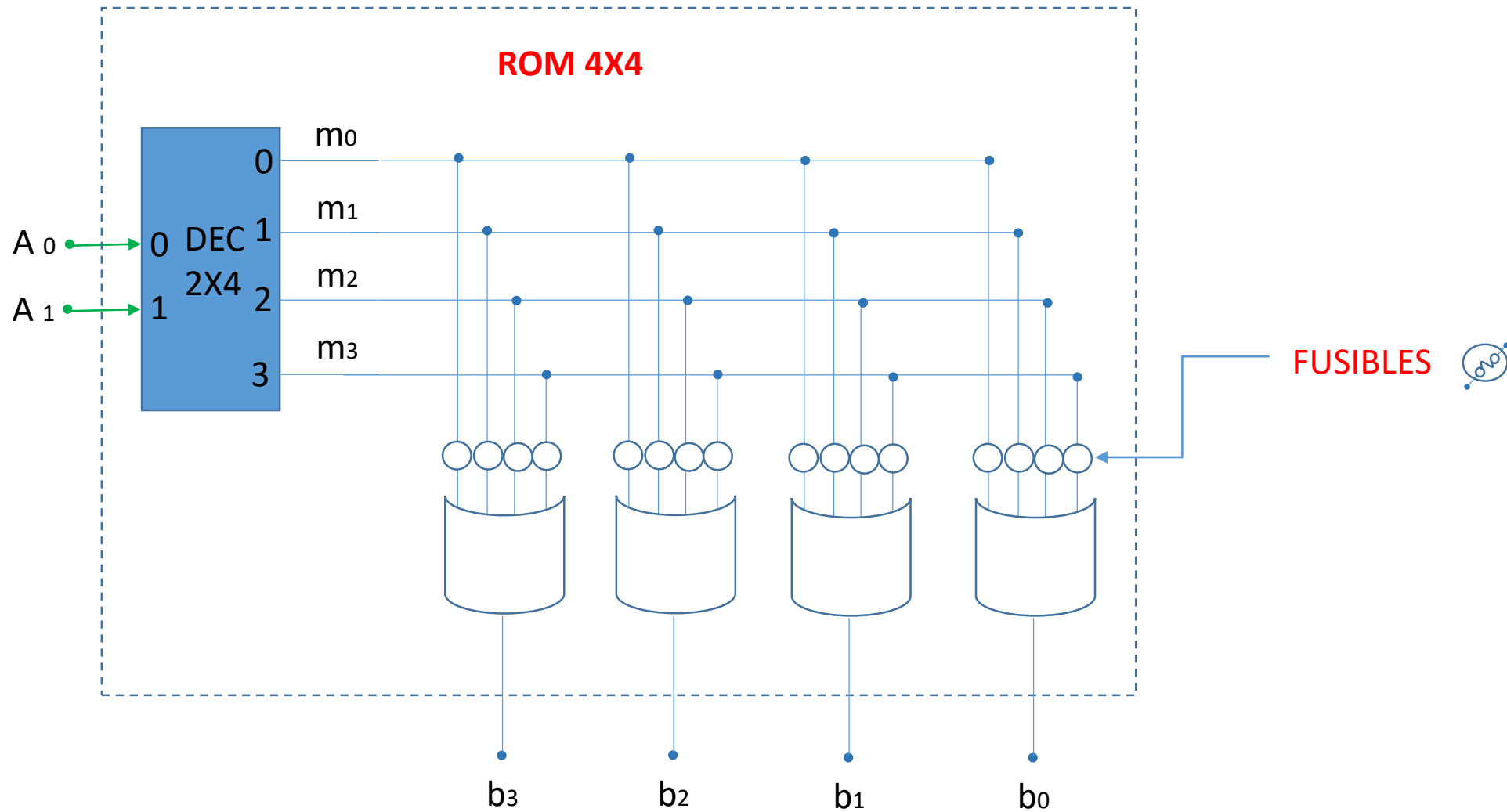
i) Diagrama Conceptual de la ROM



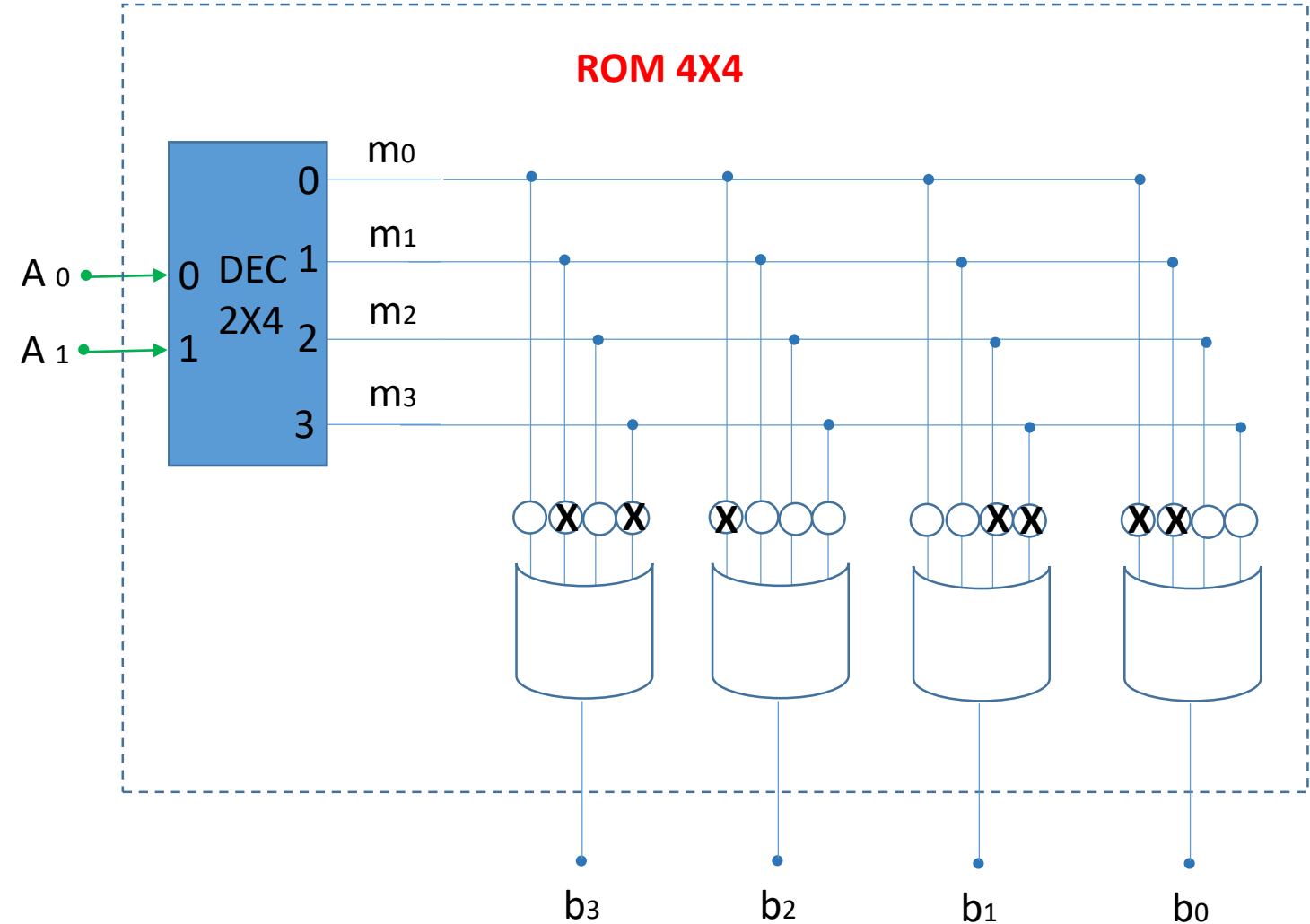
i) Diagrama "Físico" de la ROM



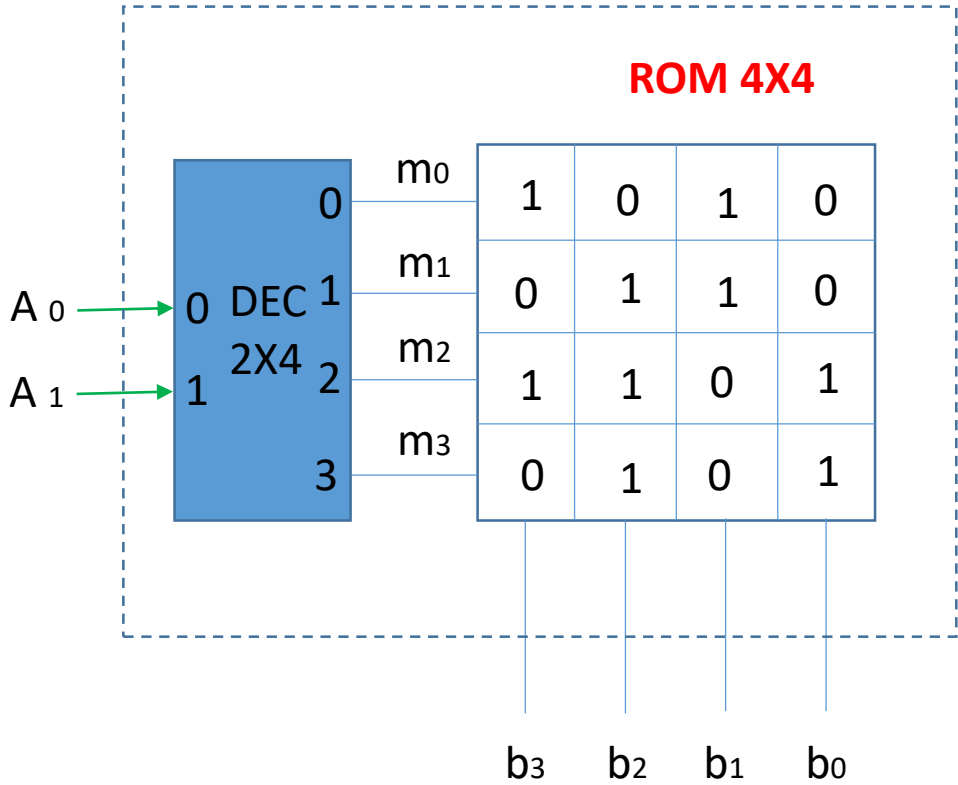
### iii) Diagrama Lógico de la ROM



iii) Diagrama Lógico de la ROM

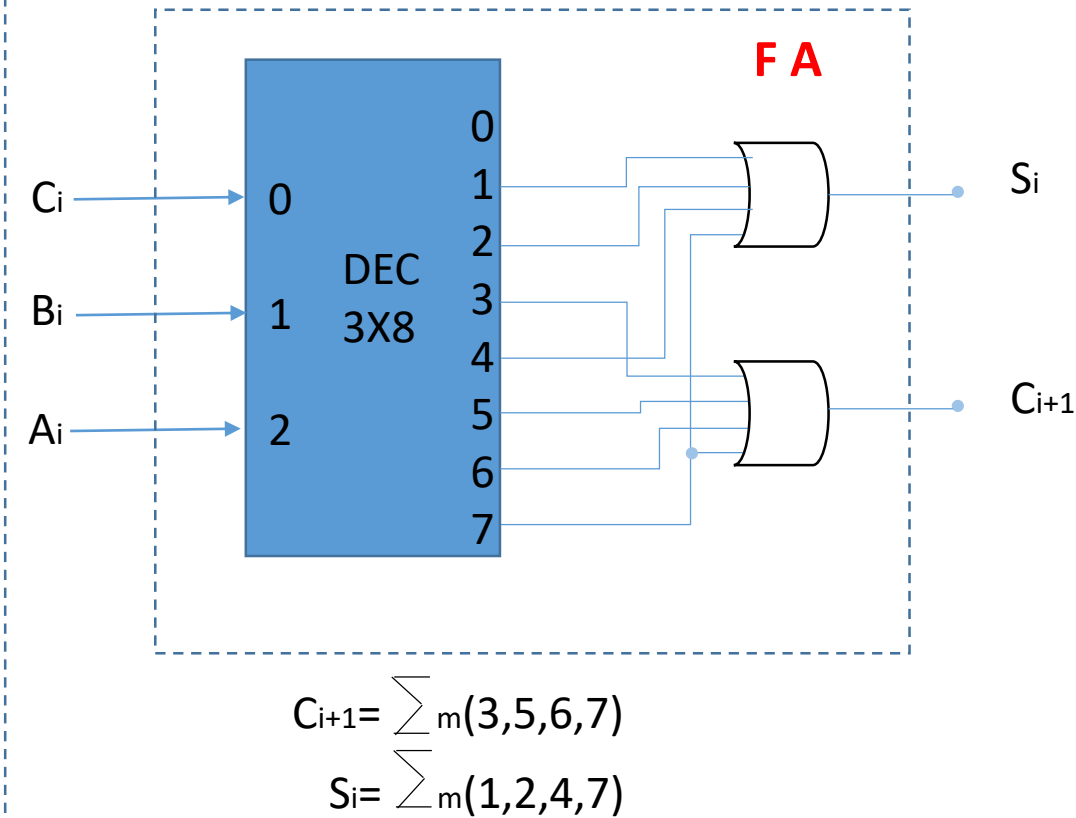
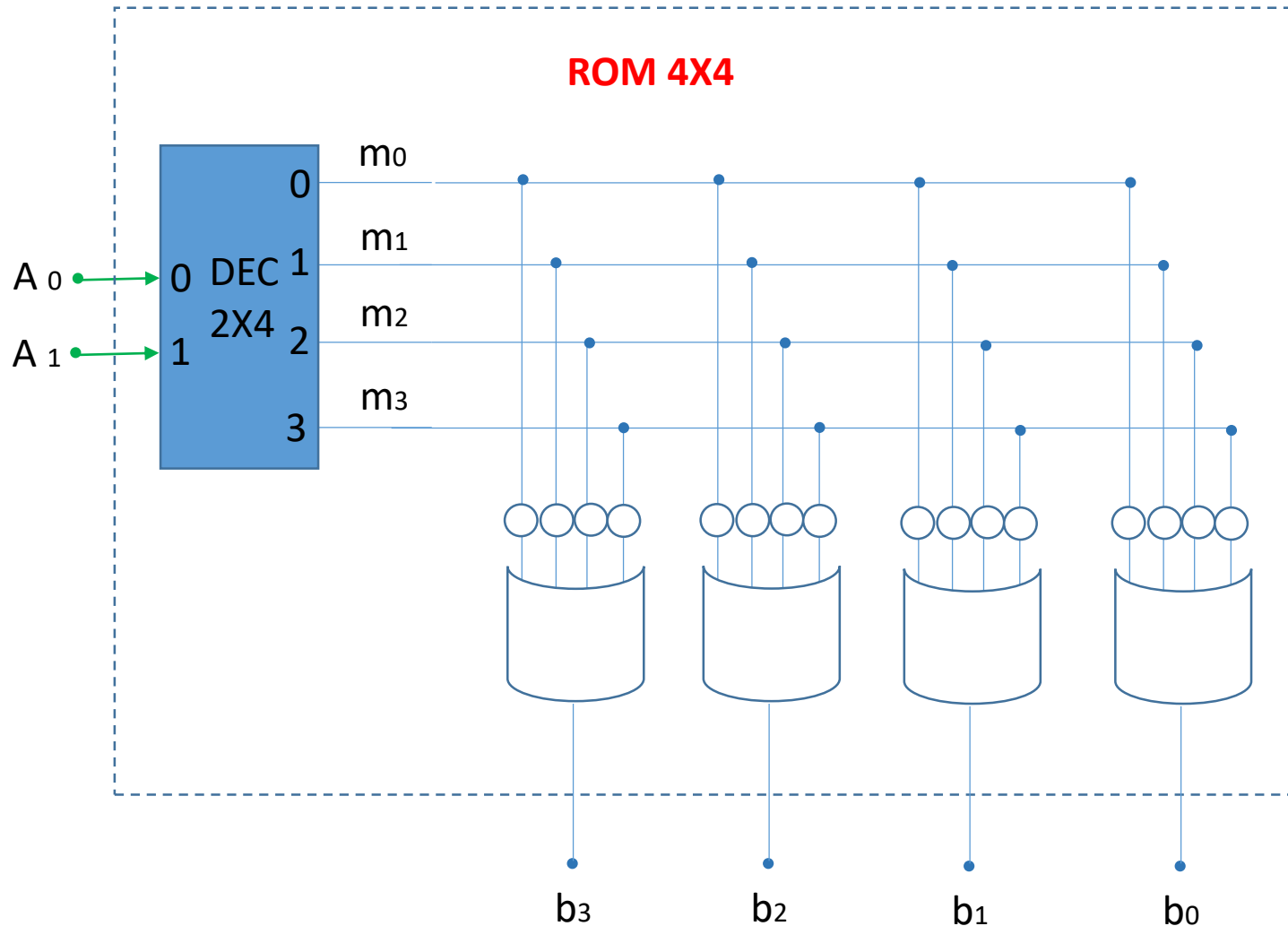


i) Diagrama Conceptual de la ROM



### c) EJEMPLOS DE DISEÑO.....

.....ANTES DEL PRIMER EJEMPLO, ¿A QUE SE PARECE EL DIAGRAMA LOGICO DE LA ROM?



**...A LA IMPLEMENTACION DE  
FUNCIONES BOOLEANAS CON  
DECODIFICADORES!!!**

EJEMPLO : IMPLEMENTAR LAS SIGUIENTES FUNCIONES BOOLEANAS UTILIZANDO UNA ROM:

$$f_1(A,B) = \sum m(0,2,3)$$

$$f_2(A,B) = \sum m(0,1,2)$$

$$f_3(A,B) = \sum m(1,2,3)$$

PRIMER PASO (TABLA DE VERDAD)

A	B	$f_1$	$f_2$	$f_3$
0	0	1	1	0
0	1	0	1	1
1	0	1	1	1
1	1	1	0	1

PARA IMPLEMENTAR FUNCIONES BOOLEANAS USANDO UNA ROM, BASTA CON CONSIDERAR A LAS VARIABLES DE ENTRADA COMO BITS DE DIRECCION Y LAS FUNCIONES BOOLEANAS A IMPLEMENTAR LOS DATOS

SEGUNDO PASO (TABLA DE PROGRAMACION DE LA ROM)

DIR		DATOS		
A <sub>1</sub>	A <sub>0</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
A	B	$f_1$	$f_2$	$f_3$
0	0	1	1	0
0	1	0	1	1
1	0	1	1	1
1	1	1	0	1

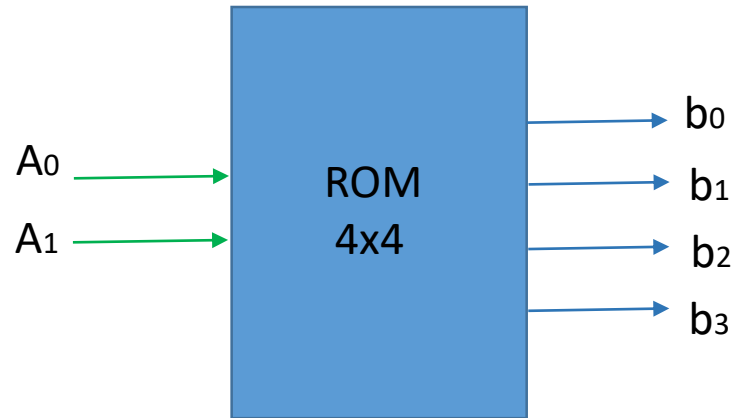
TERCER PASO (TAMAÑO DE LA ROM)

(NUM DE VARIABLES DE ENTRADA)

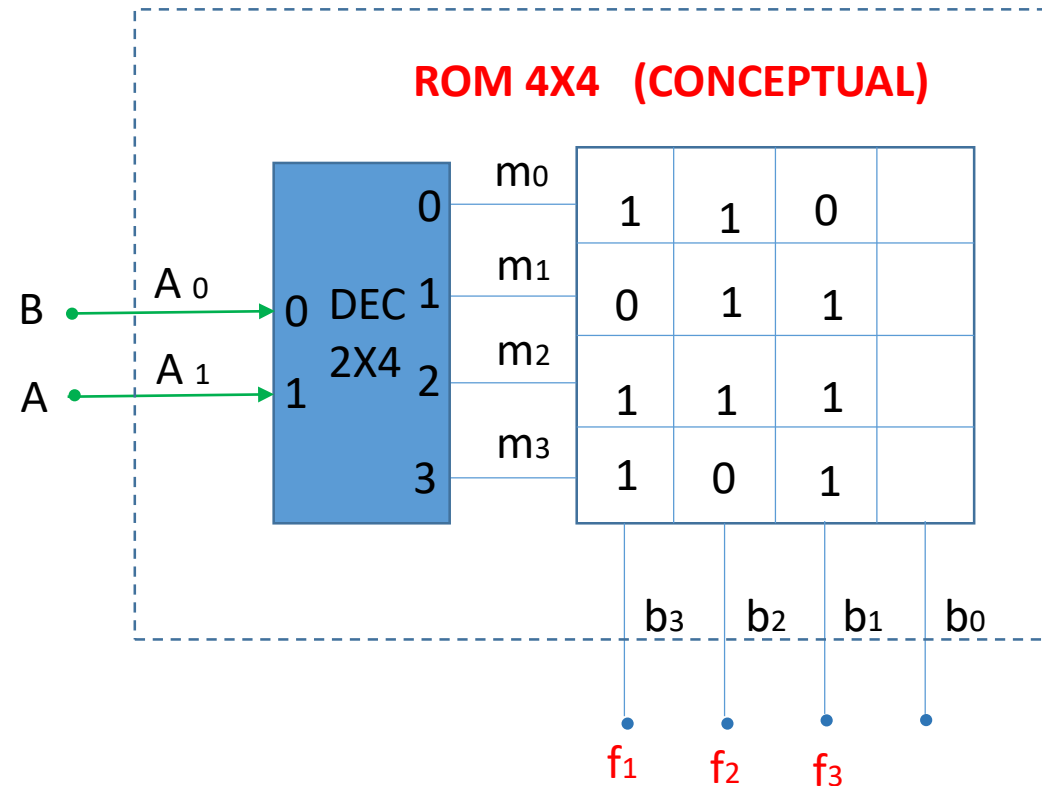
ROM 2

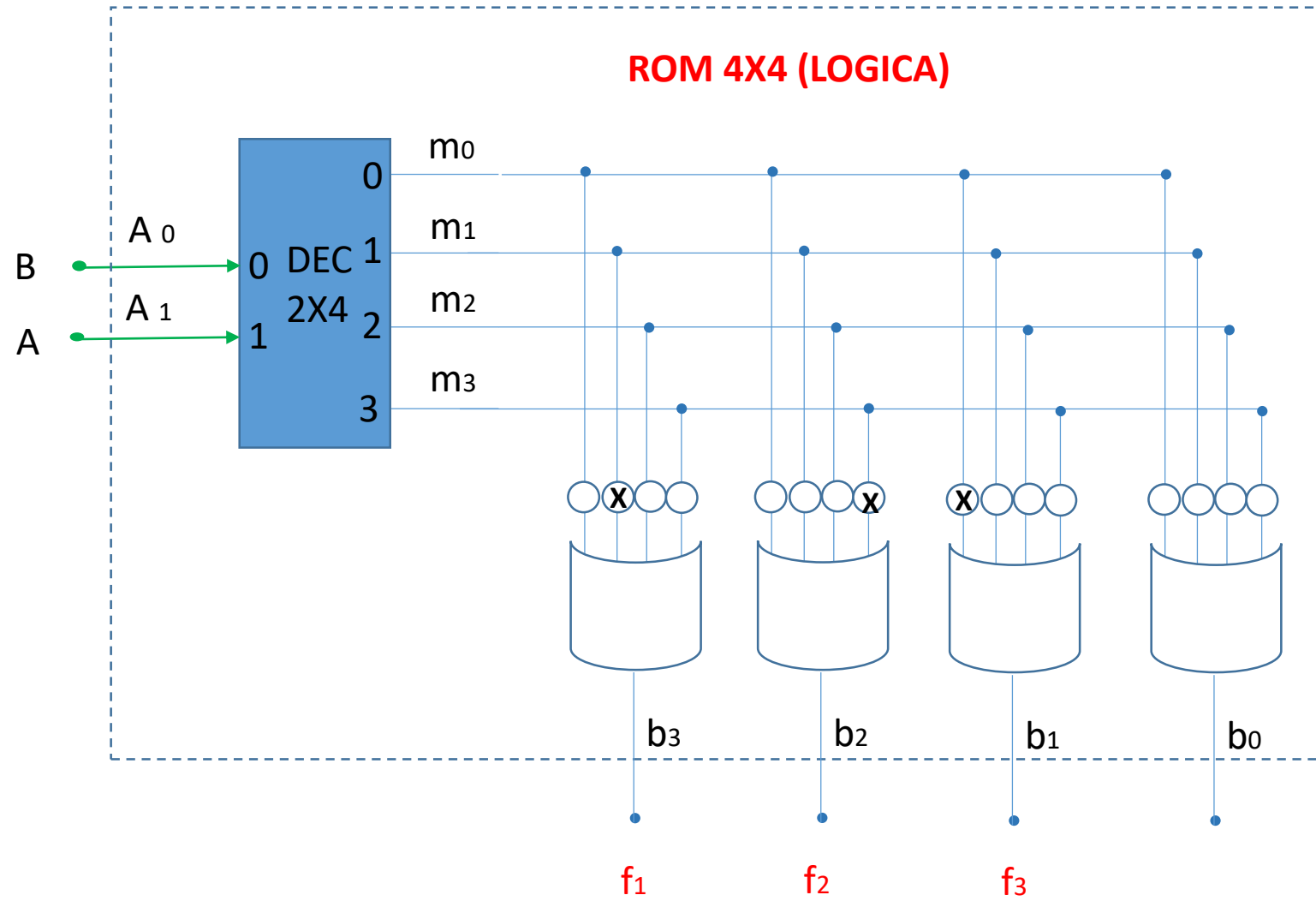
X NUMERO DE FUNCIONES BOOLEANAS

PARA EL EJEMPLO: ROM 2<sup>2</sup> X 3 = ROM 4 X 3



CUARTO PASO (PROGRAMACION DE LA ROM)



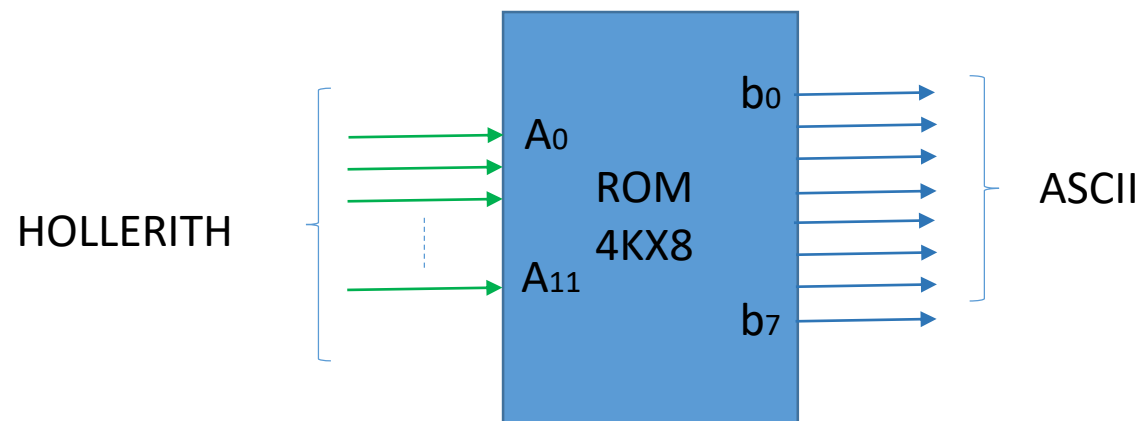


EJEMPLO : DISEÑAR UN CONVERTIDOR DE CODIGO HOLLERITH A CODIGO ASCII

¿CUANTAS VARIABLES DE ENTRADA TIENE ESTE CONVERTIDOR? **12**

¿CUANTAS VARIABLES DE SALIDA? **7**

¿DE QUE TAMAÑO ES LA ROM? **ROM 4K X 8**



EXISTEN MUCHAS CONDICIONES DON'T CARE Y SE DESPERDICIA EL 97% DE LA ROM

PARA ELLO PODEMOS UTILIZAR UN PLA QUE ME SIRVE PARA IMPLEMENTAR VARIAS FUNCIONES BOOLEANAS CON UN GRAN NUMERO DE VARIABLES, PERO CON MUCHAS CONDICIONES DON'T CARE

SOLO HAY UN DETALLE: DE LAS 4096 PALABRAS DE LA ROM (CODIGO HOLLERITH) SOLO SON VALIDAS.....128 (CODIGO ASCII)



## 9.-DISEÑO CON PLA (PROGRAMMABLE LOGIC ARRAY)

### a) INTRODUCCION

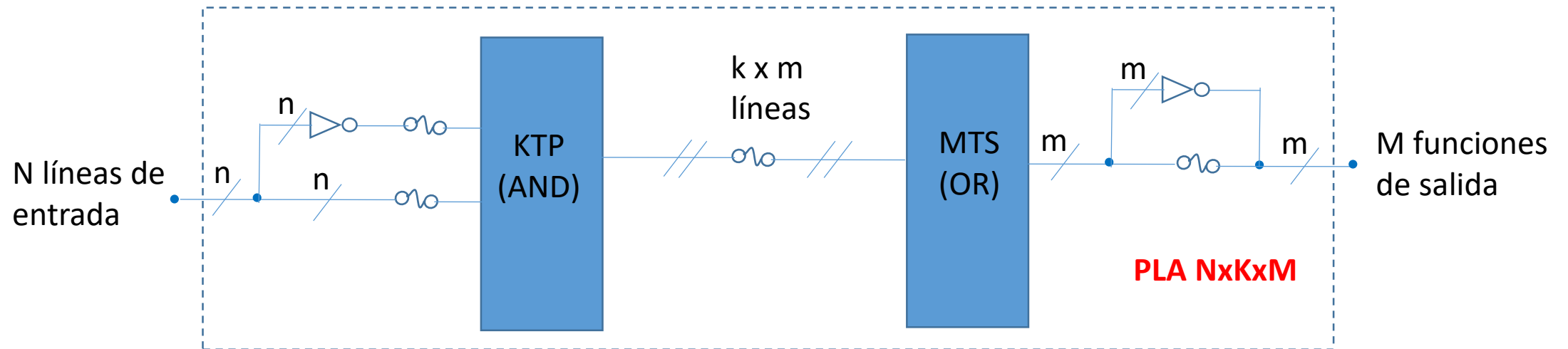
El **PLA** es similar en concepto a la ROM, pero a diferencia de la ROM, no genera todos los minterminos posibles, es decir, no tiene un Decodificador a la entrada, o al menos un Decodificador completo.

El **PLA** solo genera los Términos Producto necesarios, y como mencionamos el **PLA** nos sirve para implementar varias funciones booleanas, con un gran número de variables, pero con muchas condiciones **DON'T CARE**.

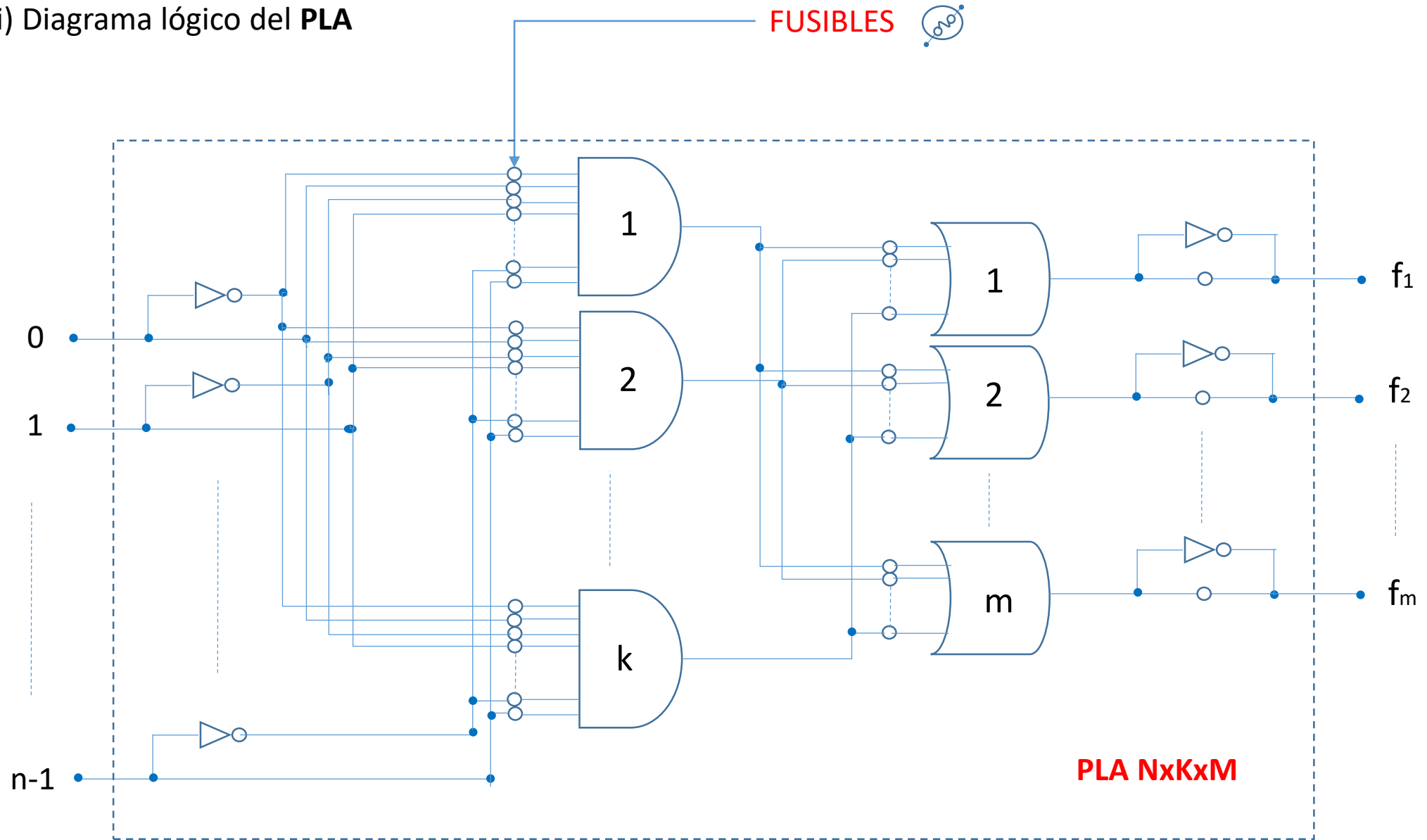
### b) CONCEPTO

El **PLA** consiste en un conjunto de compuertas **AND** que se pueden programar para generar Términos Producto y un conjunto de compuertas **OR** que se pueden programar para generar Términos Suma. Debemos recordar que **CUALQUIER FUNCION BOOLEANA SE PUEDE REPRESENTAR COMO UNA SUMA DE TERMINOS PRODUCTO**

#### i) Diagrama de bloques del PLA



i) Diagrama lógico del PLA



UN PLA COMERCIAL LO PODEMOS ENCONTRAR DE **12 X 64 X 8** O **12 X 128 X 8**

**¿CUANTOS FUSIBLES SE QUEMAN EN UNA ROM DE 4K X 8 Y CUANTOS EN UN PLA 12 X 128 X 8?**

**ROM →  $4096 \times 8 = 32,768$  FUSIBLES**

**PLA →  $12 \times 2 \times 128 + 128 \times 8 + 8 = 4104$  FUSIBLES**

EJEMPLO 1: IMPLEMENTAR LAS SIGUIENTES FUNCIONES BOOLEANAS UTILIZANDO UN PLA:

$$f_1(A,B,C) = \sum m(4,5,7)$$

$$f_2(A,B,C) = \sum m(3,5,7)$$

PRIMER PASO (MINIMIZAR LAS f's Y f's')

A B		00	01	11	10
C		00	01	11	10
0		0	0	0	1
1		0	0	1	1

$$f_1(A,B,C) = AB' + AC$$

$$f_1'(A,B,C) = A' + BC'$$

A B		00	01	11	10
C		00	01	11	10
0		0	0	0	0
1		0	1	1	1

$$f_2(A,B,C) = BC + AC$$

$$f_2'(A,B,C) = C' + A'B'$$

SEGUNDO PASO (OBTENER NUMERO MINIMO DE TP's)

núm. TP = 3

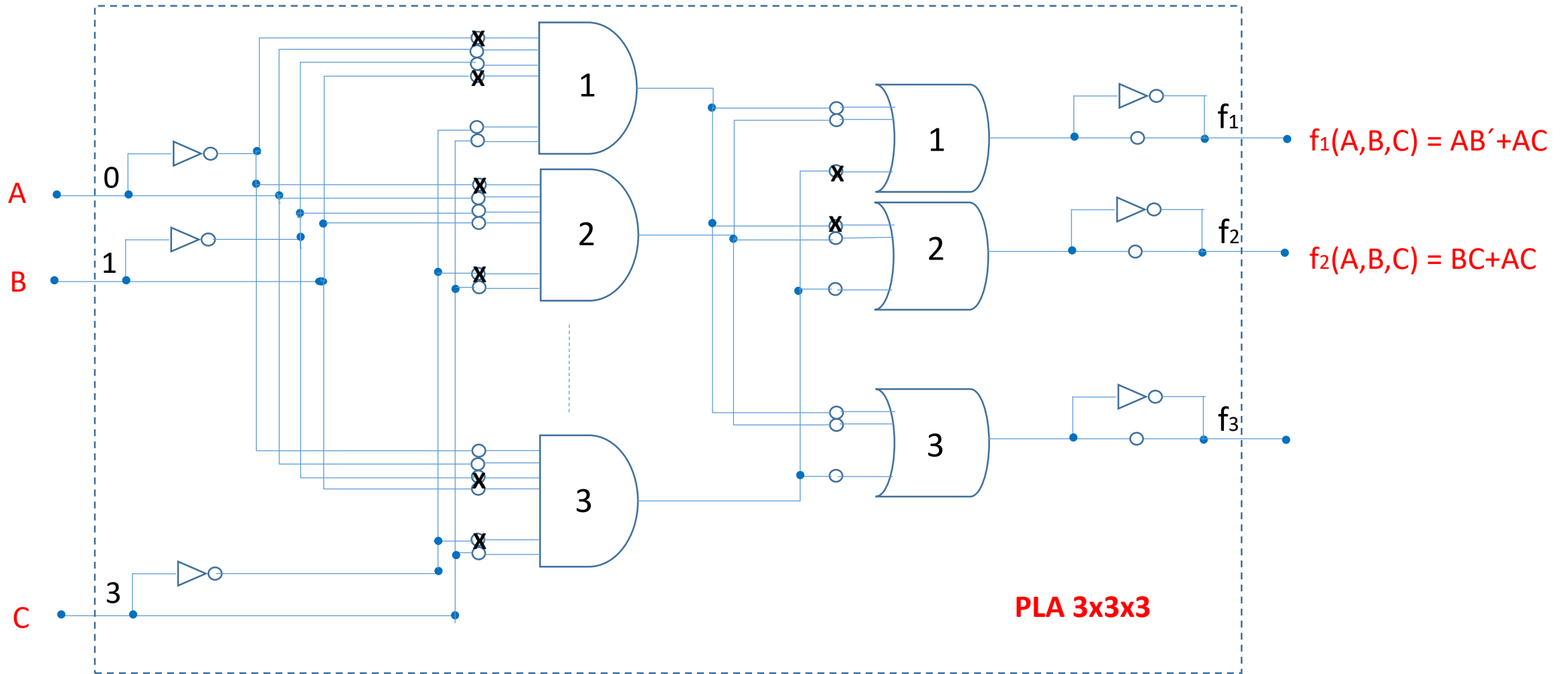
TERCER PASO (TABLA DE PROGRAMACION DEL PLA)

TP	A	B	C	f <sub>1</sub>	f <sub>2</sub>
AB'	1	0	-	1	-
AC	1	-	1	1	1
BC	-	1	1	-	1
				V	V

CUARTO PASO (TAMAÑO DEL PLA)

PLA 3 X 3 X 2

## QUINTO PASO (PROGRAMAR “QUEMAR” EL PLA)



EJEMPLO 2: IMPLEMENTAR LAS SIGUIENTES FUNCIONES BOOLEANAS UTILIZANDO UN PLA:

$$f_1(A,B,C) = \sum m(3,5,6,7)$$

$$f_2(A,B,C) = \sum m(0,2,4,7)$$

PRIMER PASO (MINIMIZAR LAS  $f$ 's Y  $f$ 's')

A B		00	01	11	10
C	0	0	0	1	0
	1	0	1	1	1

$$f_1(A,B,C) = AB+AC+BC$$

$$f_1'(A,B,C) = A'B'+A'C'+B'C'$$

A B		00	01	11	10
C	0	1	1	0	1
	1	0	0	1	0

$$f_2(A,B,C) = A'C'+B'C'+ABC$$

$$f_2'(A,B,C) = A'C+ABC'+B'C$$

SEGUNDO PASO (OBTENER NUMERO MINIMO DE TP's)

núm. TP = 4

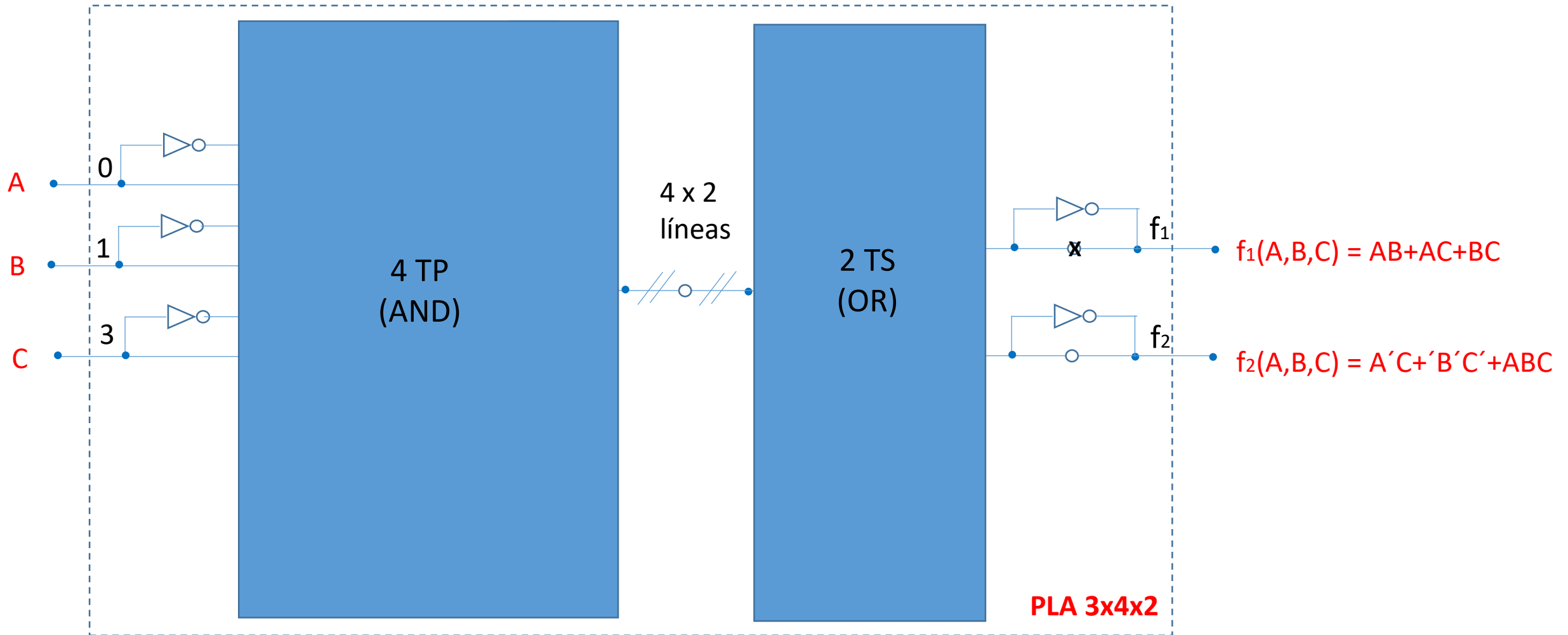
TERCER PASO (TABLA DE PROGRAMACION DEL PLA)

TP	A	B	C	$f_1$	$f_2$
$A'C'$	0	-	0	1	1
$B'C'$	-	0	0	1	1
$A'B'$	0	0	-	1	-
ABC	1	1	1	-	1
				F	V

CUARTO PASO (TAMAÑO DEL PLA)

PLA 3 X 4 X 2

## QUINTO PASO (PROGRAMAR "QUEMAR" EL PLA)



EJEMPLO 3: IMPLEMENTAR UN CIRCUITO QUE OBTENGA EL CUADRADO DE NUMEROS DE 3 BITS  
a) USANDO UNA ROM MINIMA  
b) USANDO UN PLA MINIMO

¿QUIEN DIJO YO?

**TAREA !!!!!!!!!!!!!!!**