Proyecto de Arquitectura de Computadoras

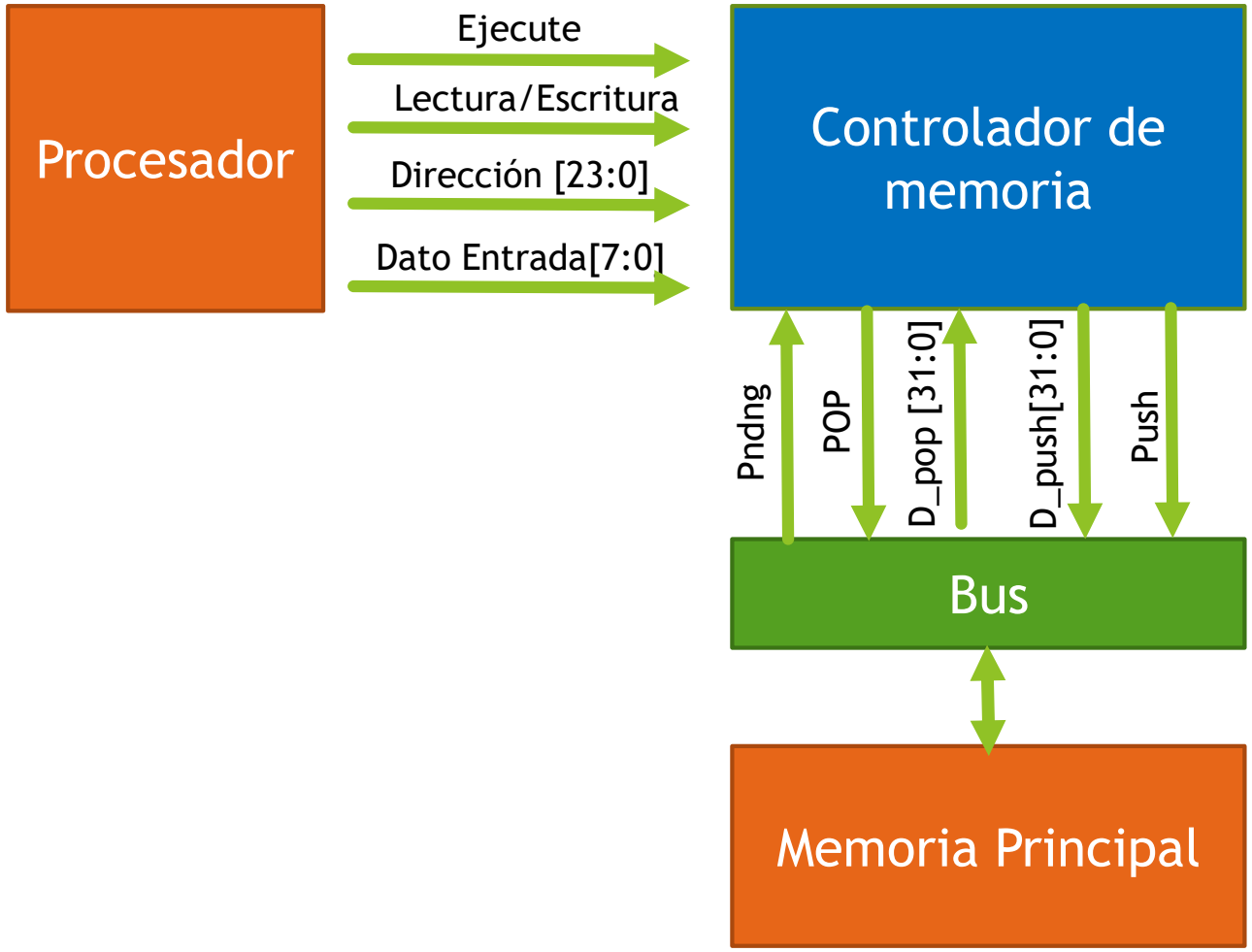
Jacobo De Bruyn Monge

2014079654

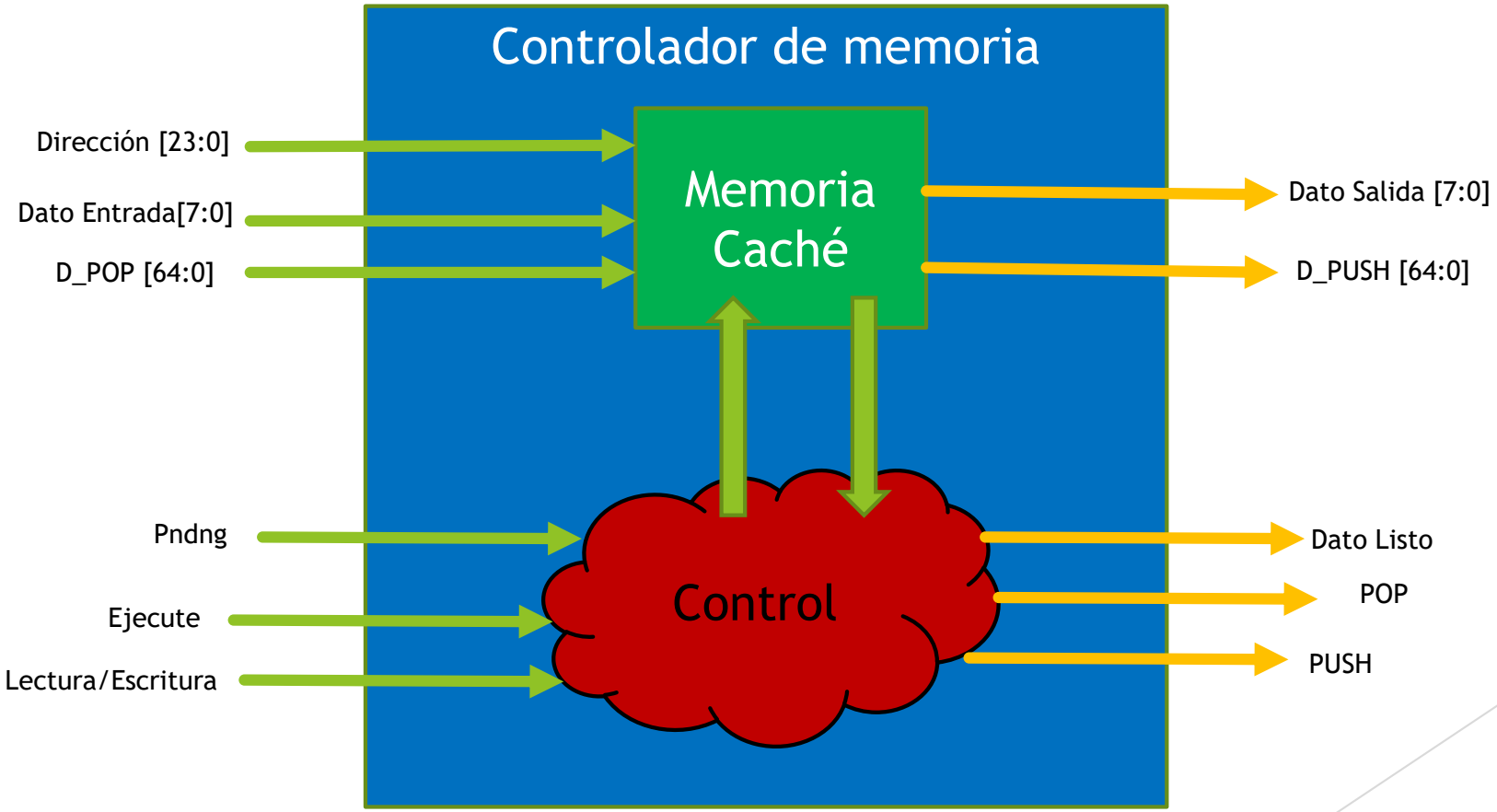
Características del Sistema

- Asociatividad 4
- Líneas de caché de 64 bits=8B
- 32KB de caché total
- Cada asociatividad tiene una tamaño de 8KB

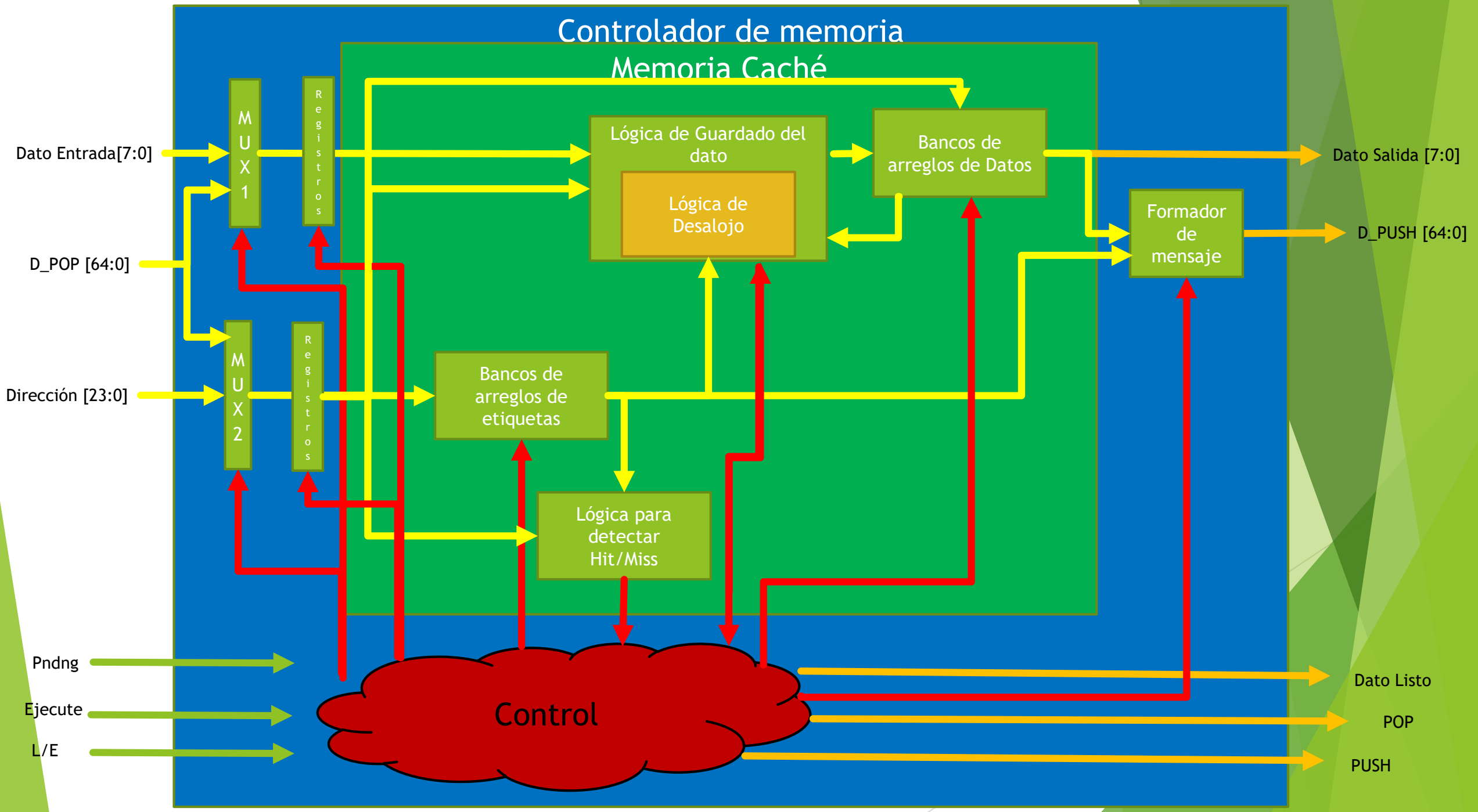
Diagrama de General



Controlador de Memoria



Controlador de Memoria Diagrama de Segundo Nivel


Controlador de memoria
Memoria Caché
MUX 1
Registros
Lógica de Guardado del dato
Lógica de Desalojo
Bancos de arreglos de Datos
Formador de mensaje
Dato Entrada[7:0]
D_POP [64:0]
Dirección [23:0]
MUX 2
Registros
Bancos de arreglos de etiquetas
Lógica para detectar Hit/Miss
Dato Salida [7:0]
D_PUSH [64:0]
Pndng
Ejecute
L/E
Control
Dato Listo
POP
PUSH

Controlador de Memoria Diagrama de Tercer Nivel

Controlador de Memoria Diagramas de Cuarto Nivel

Bancos de arreglos de Datos

- Diagrama general de los bancos de datos de acuerdo a las especificaciones
- Tamaño de la línea 8B
- Tamaño Total de cada Banco 8KB
- Cantidad de líneas 1024

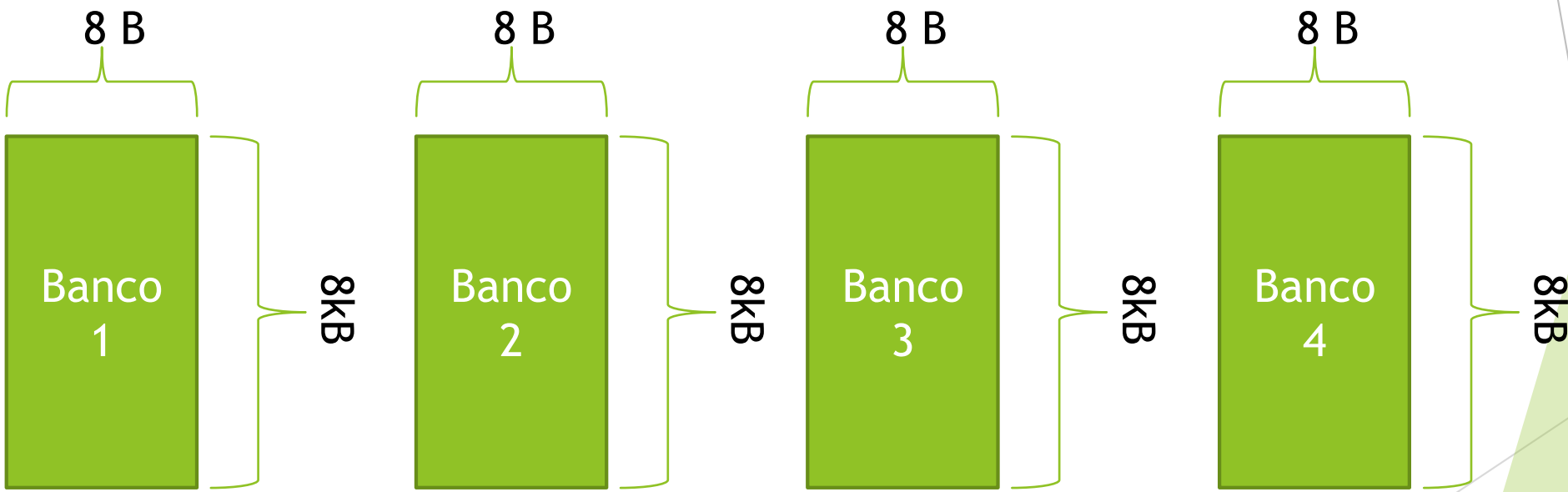
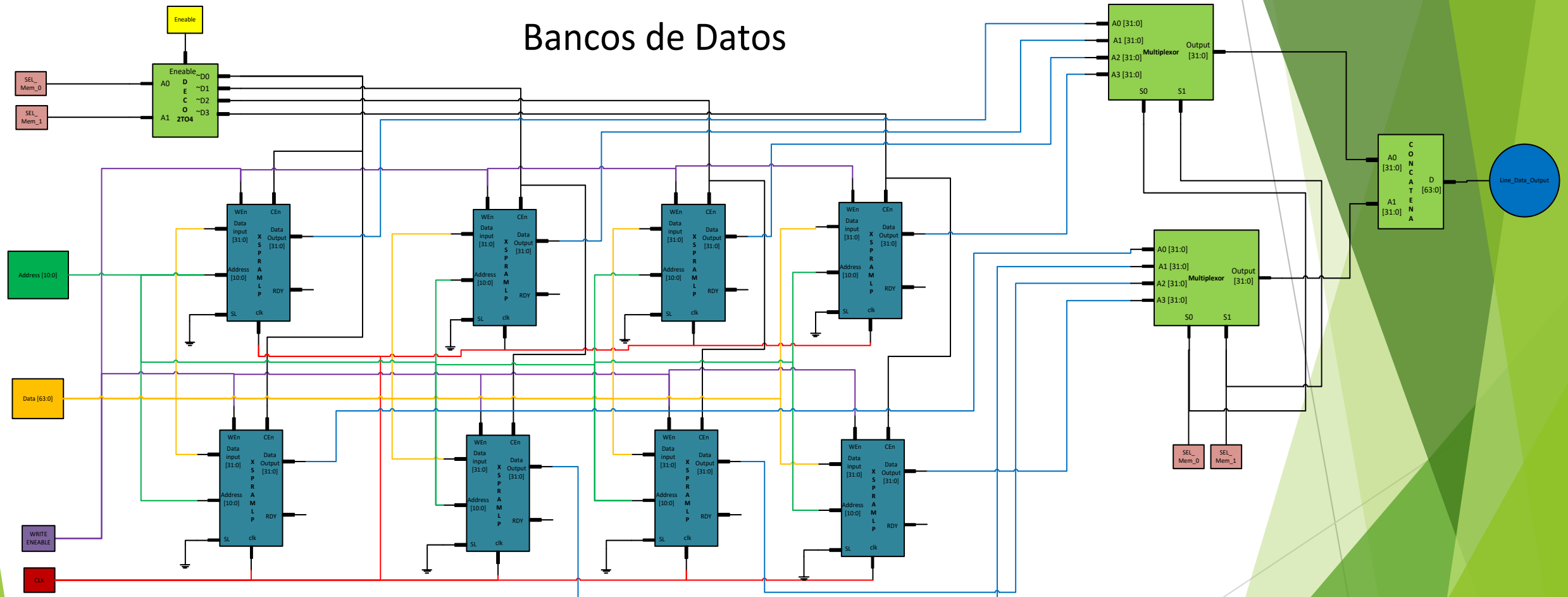


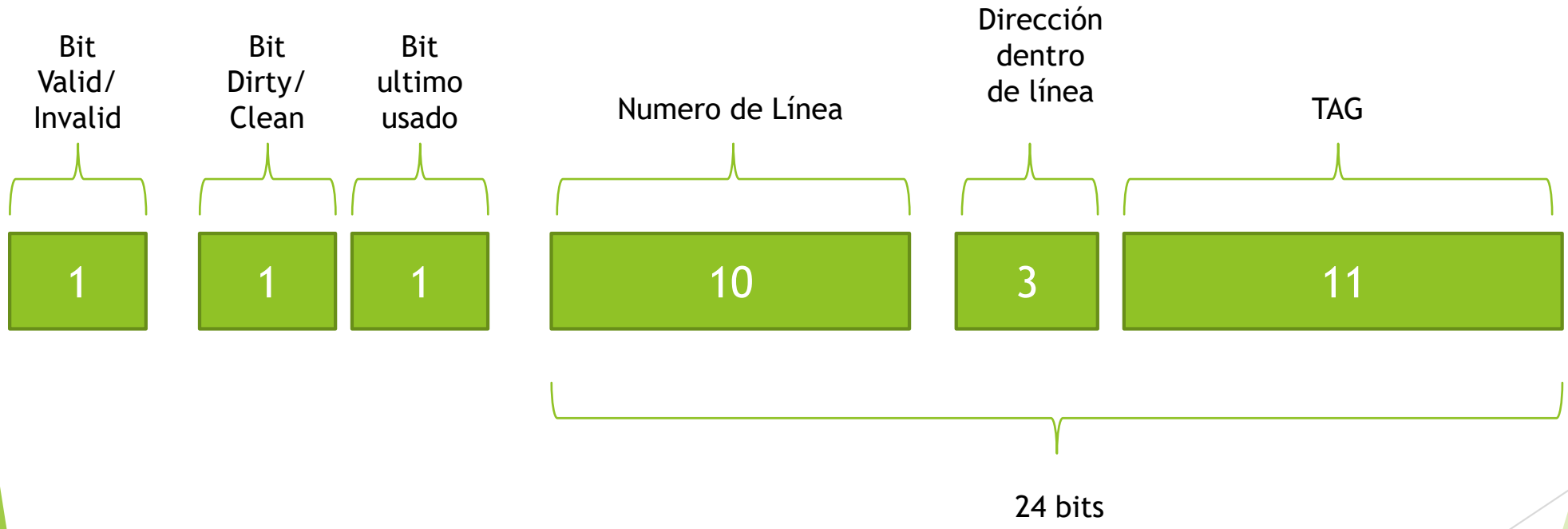
Diagrama de cuarto nivel de un banco de datos

Bancos de Datos



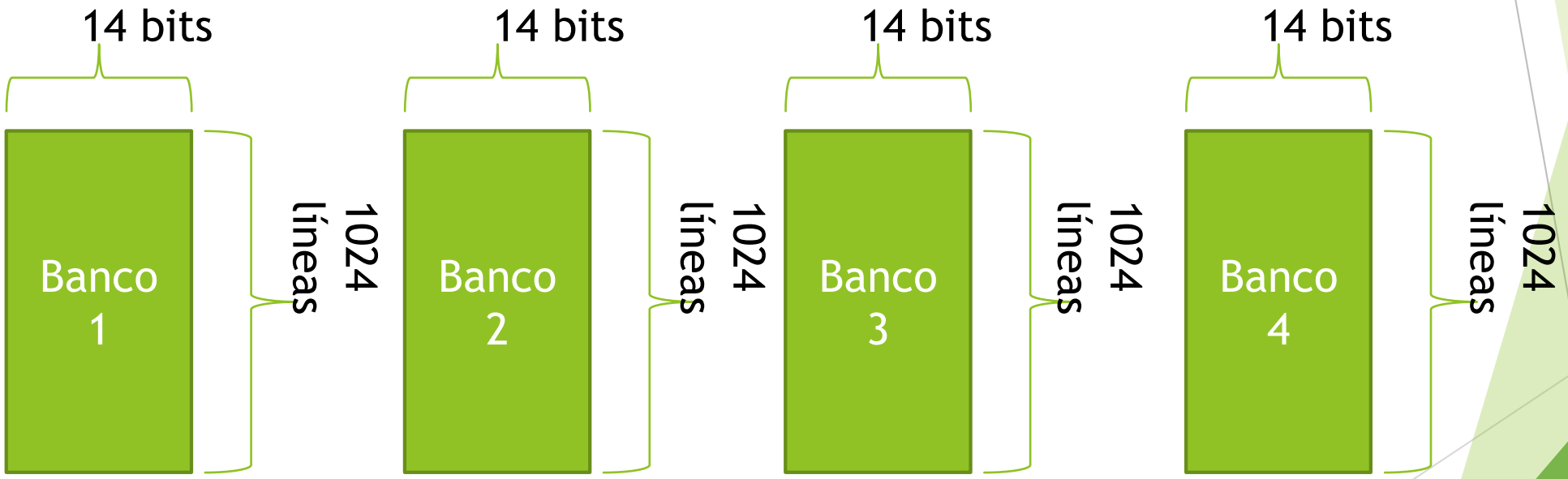
Bancos de arreglos de etiquetas

- Estructura del Tag



Bancos de arreglos de etiquetas

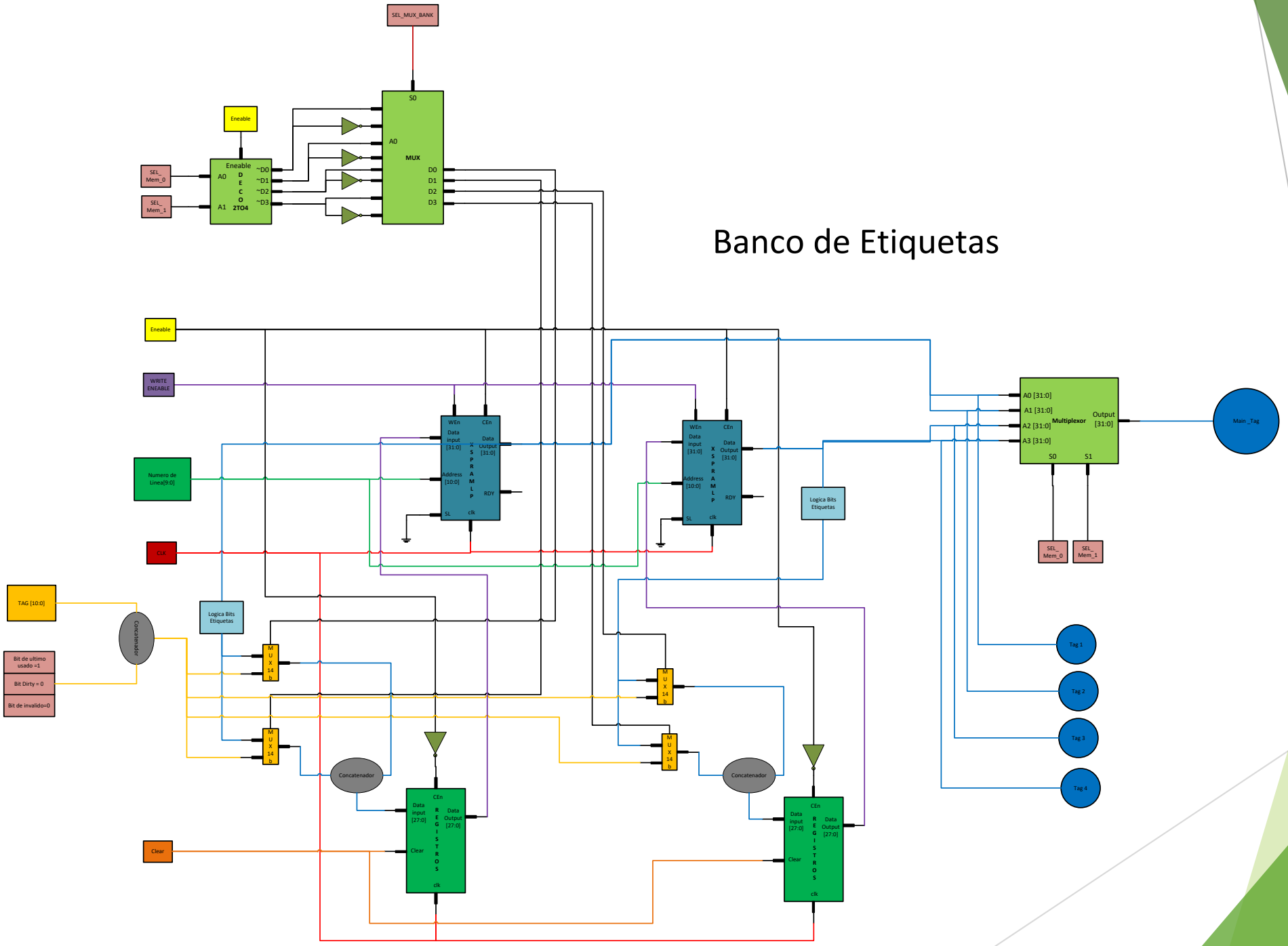
- Diagrama general de los bancos de Tags de acuerdo a las especificaciones
- Largo del Tag 14 Bits
- Cantidad de líneas 1024



Puedo reducir el numero de memorias a la mitad en la implementacion

Diagrama de cuarto nivel de un banco de Etiquetas

Banco de Etiquetas



Lógica para determinar bits de la etiqueta

- Esta lógica se encarga de actualizar los bits de dirty, invalid y ultimo en ser usado
- Cada cuadro de estos en el diagrama equivale a dos unidades de lógica para determinar bits de la etiqueta, debido que cada memoria almacena dos bancos de Tags, por lo cual su salida debe ser analizada para cada banco.
- La señales con nombres Ai, Bi corresponde a entradas de control de multiplexores. Donde i es un numero.

Lógica para determinar bits de la etiqueta

Logica para determinar bits en etiquetas					
Sel_Mux_Mem_1	Sel_Mux_Mem_0	Bit linea en uso	A1	A2	A3
0	0	0	1	0	0
0	0	1	x	x	1
0	1	0	0	0	0
0	1	1	x	x	x
1	0	0	1	0	0
1	0	1	x	0	0
1	1	0	x	x	x
1	1	1	x	x	x

A1=(~Mem0)

A2=0

A3=(~Mem1*Uso)

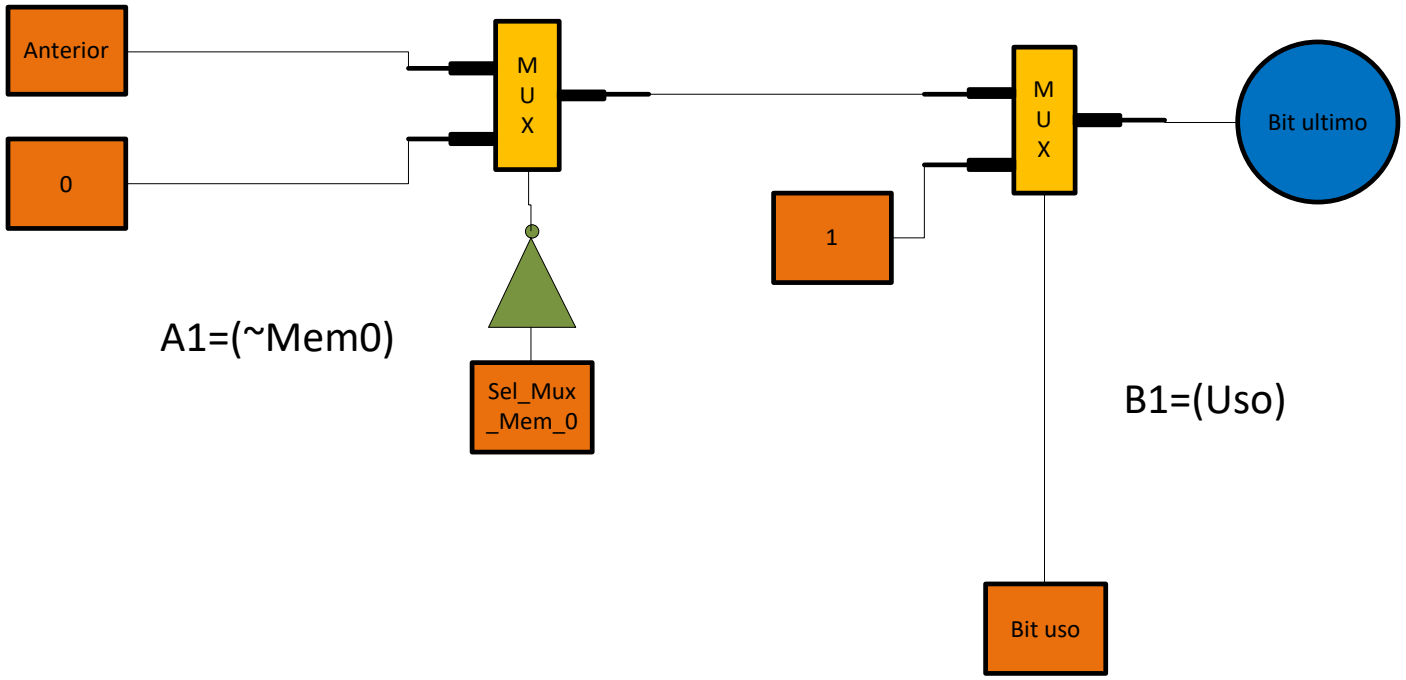
Logica para determinar bits en etiquetas					
Sel_Mux_Mem_1	Sel_Mux_Mem_0	Bit linea en uso	B1	B2	B3
0	0	0	0	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	1	x	x	1
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	x	x	x
1	1	1	x	x	x

B1=(Uso)

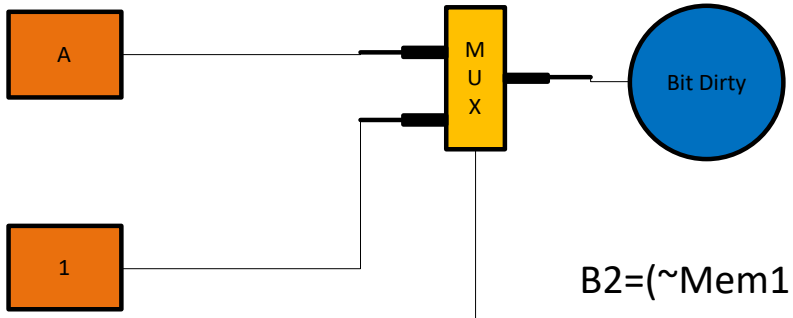
B2=(~Mem1*Uso)

B3=(Mem0*Uso)

Bit Ultimo


Anterior
0
M
U
X
A1=(~Mem0)
Sel_Mux
_Mem_0
1
M
U
X
Bit ultimo
B1=(Uso)
Bit uso

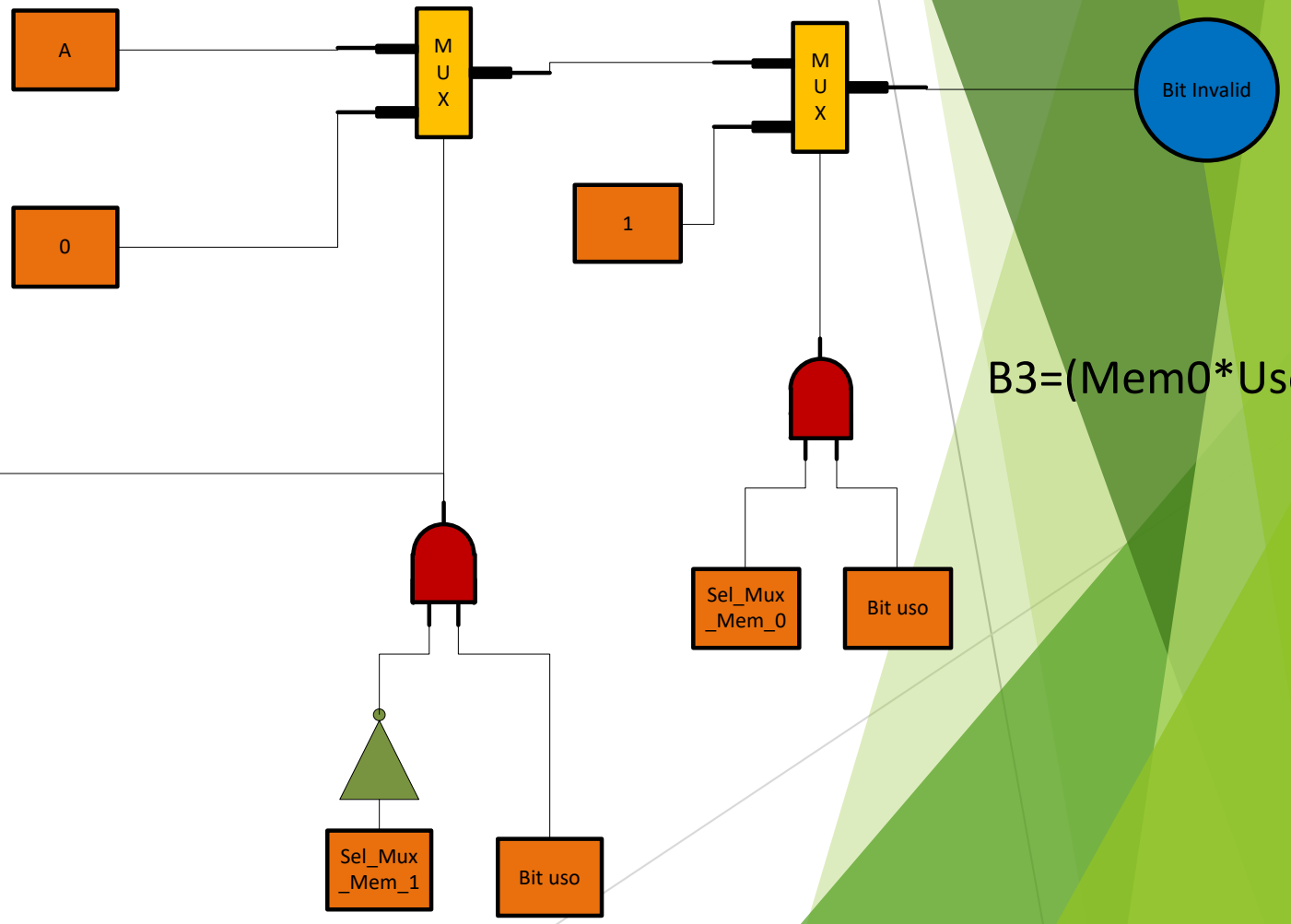
Bit Dirty



B2=(~Mem1*Uso)

A2=0

Bit Invalid

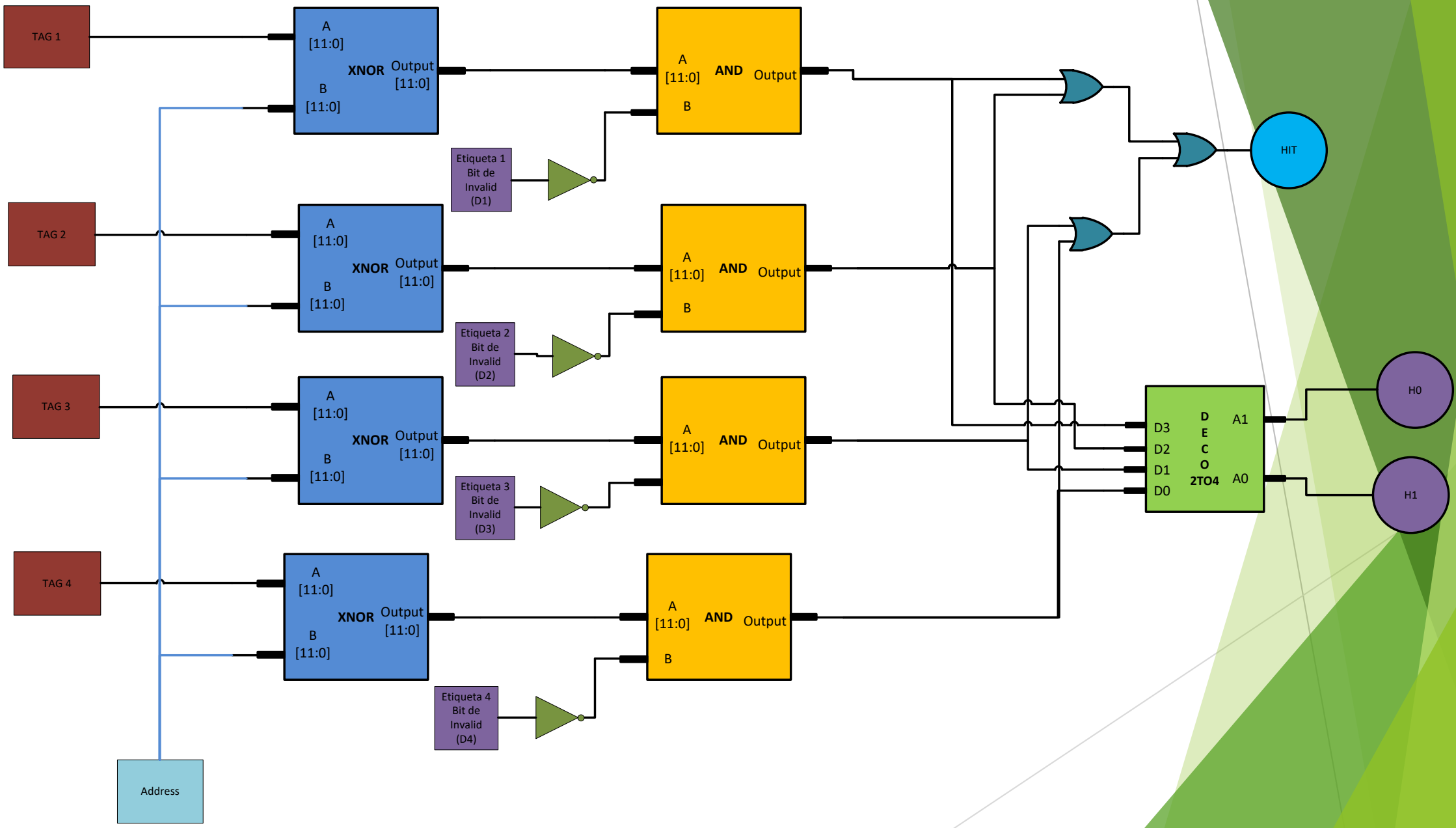


B3=(Mem0*Uso)

A3=(~Mem1*Uso)

Lógica para detectar Hit/Miss

Se compara cada uno de los Tags correspondientes a la posición donde la línea puede ser guardada
y si alguno de estos es un Tag valido, y además es igual al Tag dado por la dirección entonces ocurre un Hit.
Además se identifica en cual Tag ocurrió el hit mediante un decodificador el cual determina en binario
el numero del tag donde ocurrió el hit


TAG 1
TAG 2
TAG 3
TAG 4
Address
A [11:0]
B [11:0]
XNOR
Output [11:0]
AND
Etiqueta 1 Bit de Invalid (D1)
Etiqueta 2 Bit de Invalid (D2)
Etiqueta 3 Bit de Invalid (D3)
Etiqueta 4 Bit de Invalid (D4)
HIT
D3
D2
D1
D0
DECO 2TO4
A1
A0
H0
H1

Lógica de Guardado del Dato

Nomenclatura de las tablas de verdad

- La letra A hace mención al bit de Dirty de cada línea, el cual es 0 si la línea esta limpia, y es 1 si la línea esta sucia
- La letra B se refiere al bit para identificar si fue la ultima línea en ser usada, donde B es 0 si la línea no fue la ultima en ser usada y B es 1 si la línea fue la ultima en ser usada
- La letra D es para identificar si la línea es valida o invalida, para lo cual D es 0 si la línea esta valida y D es 1 si la línea ha sido invalidada

Dirty	Usado	Valid
A	B	D

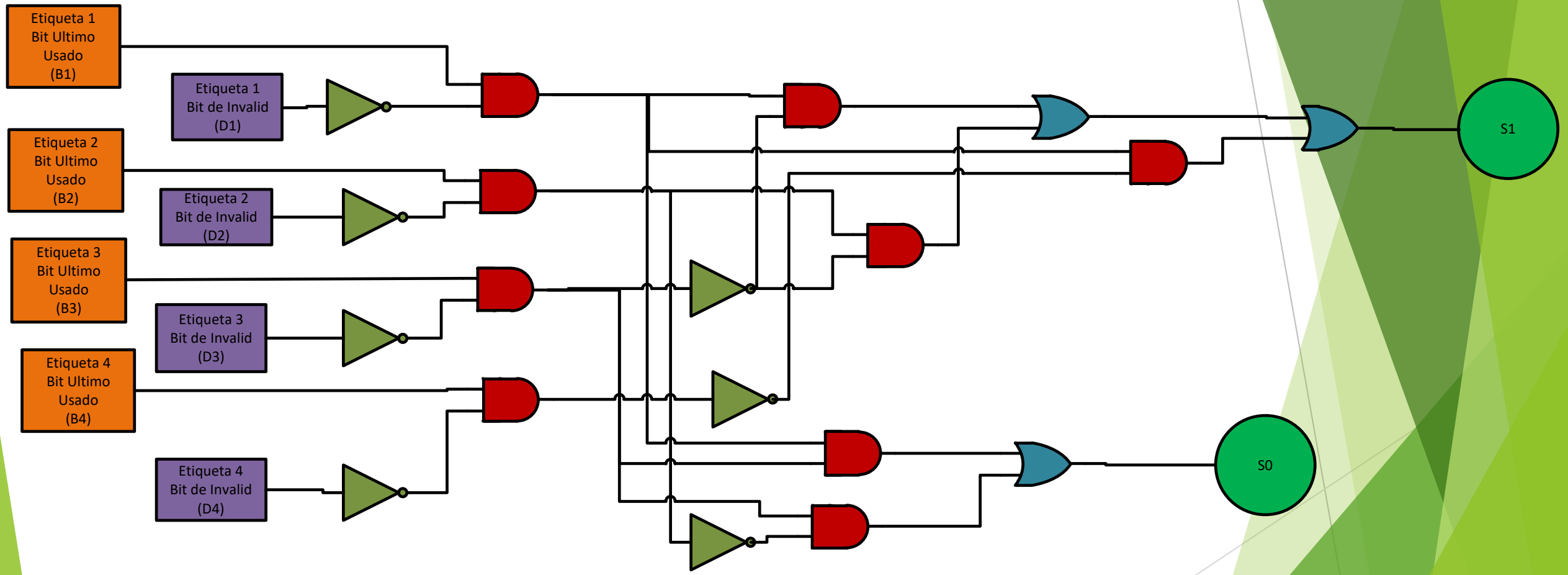
Política De Desalojo (Cualquiera menos el ultimo utilizado)

Politica de Desalojo					
Entradas				Salidas	
B4*~D4	B3*~D3	B2*~D2	B1*~D1	S1	S0
0	0	0	0	x	x
0	0	0	1	1	x
0	0	1	0	1	x
0	0	1	1	1	x
0	1	0	0	x	1
0	1	0	1	x	1
0	1	1	0	0	0
0	1	1	1	1	1
1	0	0	0	0	x
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	x
1	1	0	1	0	1
1	1	1	0	0	0
1	1	1	1	x	x

Si la líneas son validas, se implementa la siguiente tabla de verdad donde se desaloja cualquier línea valida, excepto la ultima línea en ser utilizada, S1 y S0 determinan en binario la línea que va a ser desalojada. Para simplificar la expresión de la ecuación se utiliza la nomenclatura Fi para indicar Bi*~Di, donde i es un numero

S1= (~F3*F1)+(~F3*F2)+(~F4*F1)

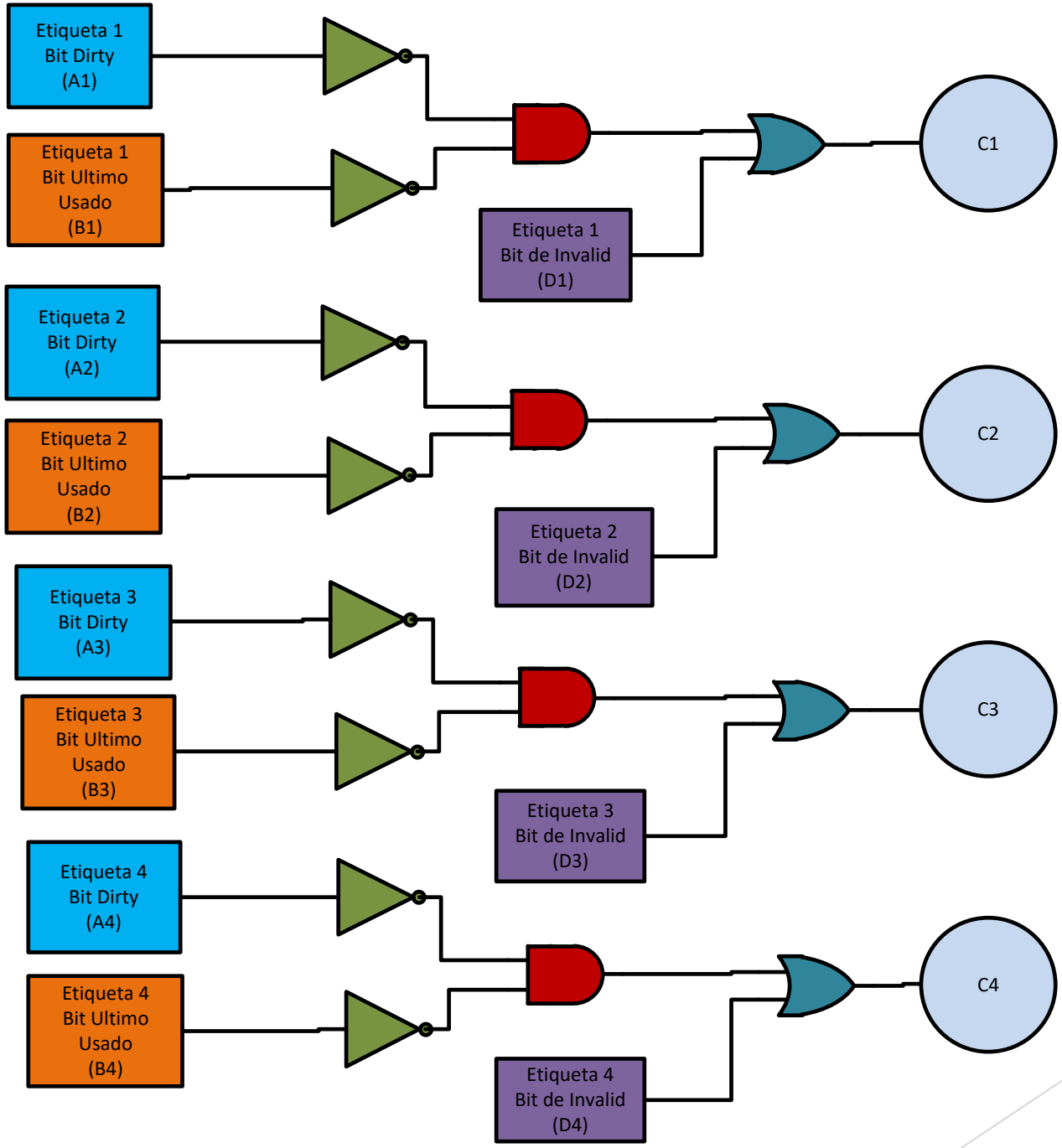
S0 = (F3*F1)+(F3*~F2)


Etiqueta 1
Bit Ultimo
Usado
(B1)
Etiqueta 1
Bit de Invalid
(D1)
Etiqueta 2
Bit Ultimo
Usado
(B2)
Etiqueta 2
Bit de Invalid
(D2)
Etiqueta 3
Bit Ultimo
Usado
(B3)
Etiqueta 3
Bit de Invalid
(D3)
Etiqueta 4
Bit Ultimo
Usado
(B4)
Etiqueta 4
Bit de Invalid
(D4)
S1
S0

Determinar las posibles lineas donde se puede guardar el dato

Donde puede ser escrito			
Entradas			Salidas
D_i	A_i	B_i	C_i
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

En esta parte se determina en cuales líneas puede ser escrito.
Se puede escribir en cualquier línea que se encuentre invalida y además se puede escribir en una línea que se encuentre valida pero que no este sucia y además no sea la ultima en ser usada


Etiqueta 1
Bit Dirty
(A1)
Etiqueta 1
Bit Ultimo
Usado
(B1)
Etiqueta 1
Bit de Invalid
(D1)
C1
Etiqueta 2
Bit Dirty
(A2)
Etiqueta 2
Bit Ultimo
Usado
(B2)
Etiqueta 2
Bit de Invalid
(D2)
C2
Etiqueta 3
Bit Dirty
(A3)
Etiqueta 3
Bit Ultimo
Usado
(B3)
Etiqueta 3
Bit de Invalid
(D3)
C3
Etiqueta 4
Bit Dirty
(A4)
Etiqueta 4
Bit Ultimo
Usado
(B4)
Etiqueta 4
Bit de Invalid
(D4)
C4

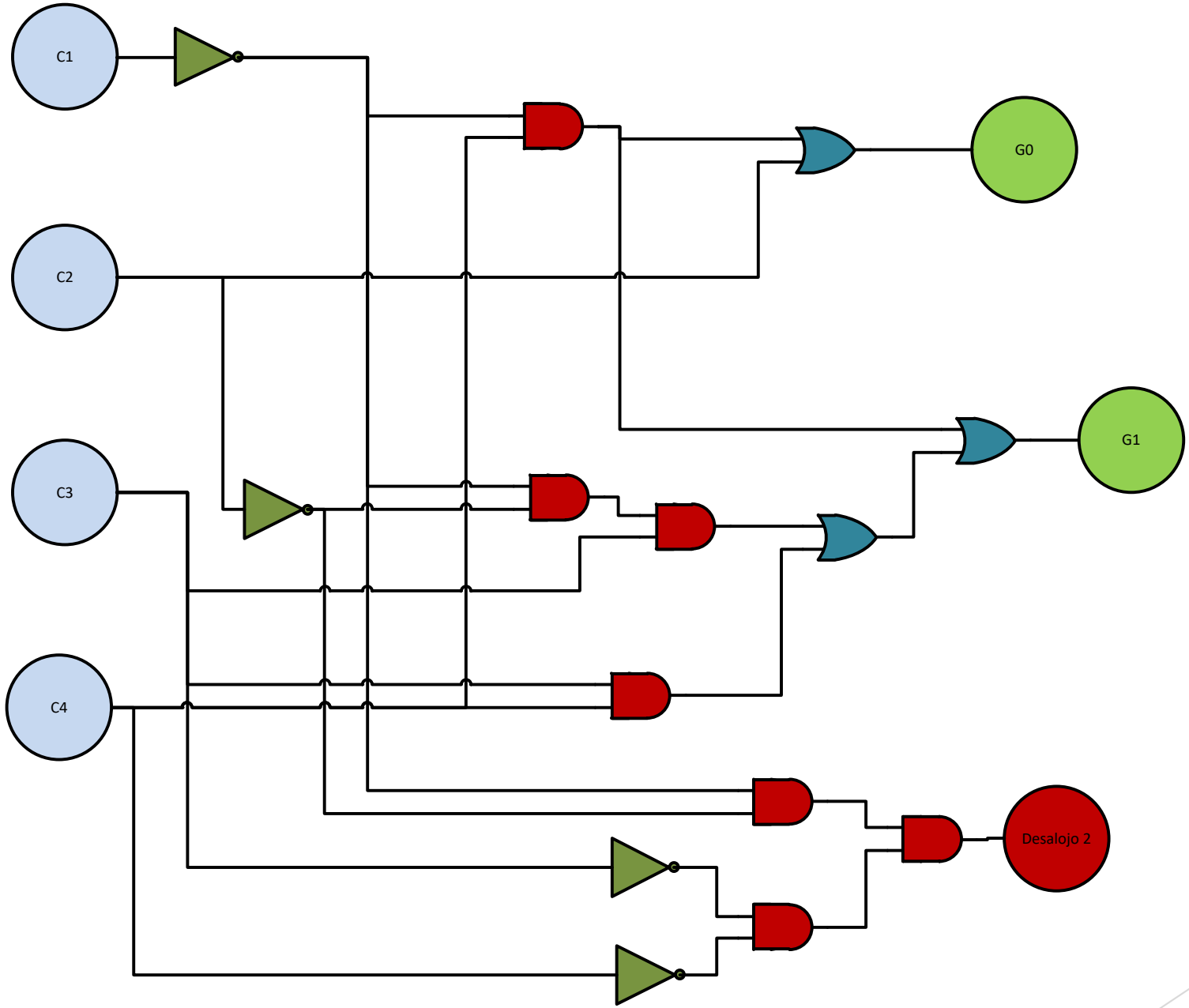
Determinar en cual línea se va a almacenar el dato

Selección de Lugar para Guardar						
Entradas				Salidas		
C4	C3	C2	C1	G1	G0	Desalojo 2
0	0	0	0	0	0	1
0	0	0	1	0	0	0
0	0	1	0	0	1	0
0	0	1	1	0	x	0
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	1	0
0	1	1	1	0	x	0
1	0	0	0	1	1	0
1	0	0	1	0	0	0
1	0	1	0	x	1	0
1	0	1	1	0	x	0
1	1	0	0	1	x	0
1	1	0	1	1	x	0
1	1	1	0	x	1	0
1	1	1	1	x	x	0

Los Bits G1 y G0 dan en binario el numero de línea en el cual se va a almacenar el dato.
Además, si no existe ninguna línea donde se pueda escribir se establece una señal de desalojo

$$G1 = (C3*\sim C2*\sim C1)+(C4*\sim C1)+(C4*C3)$$

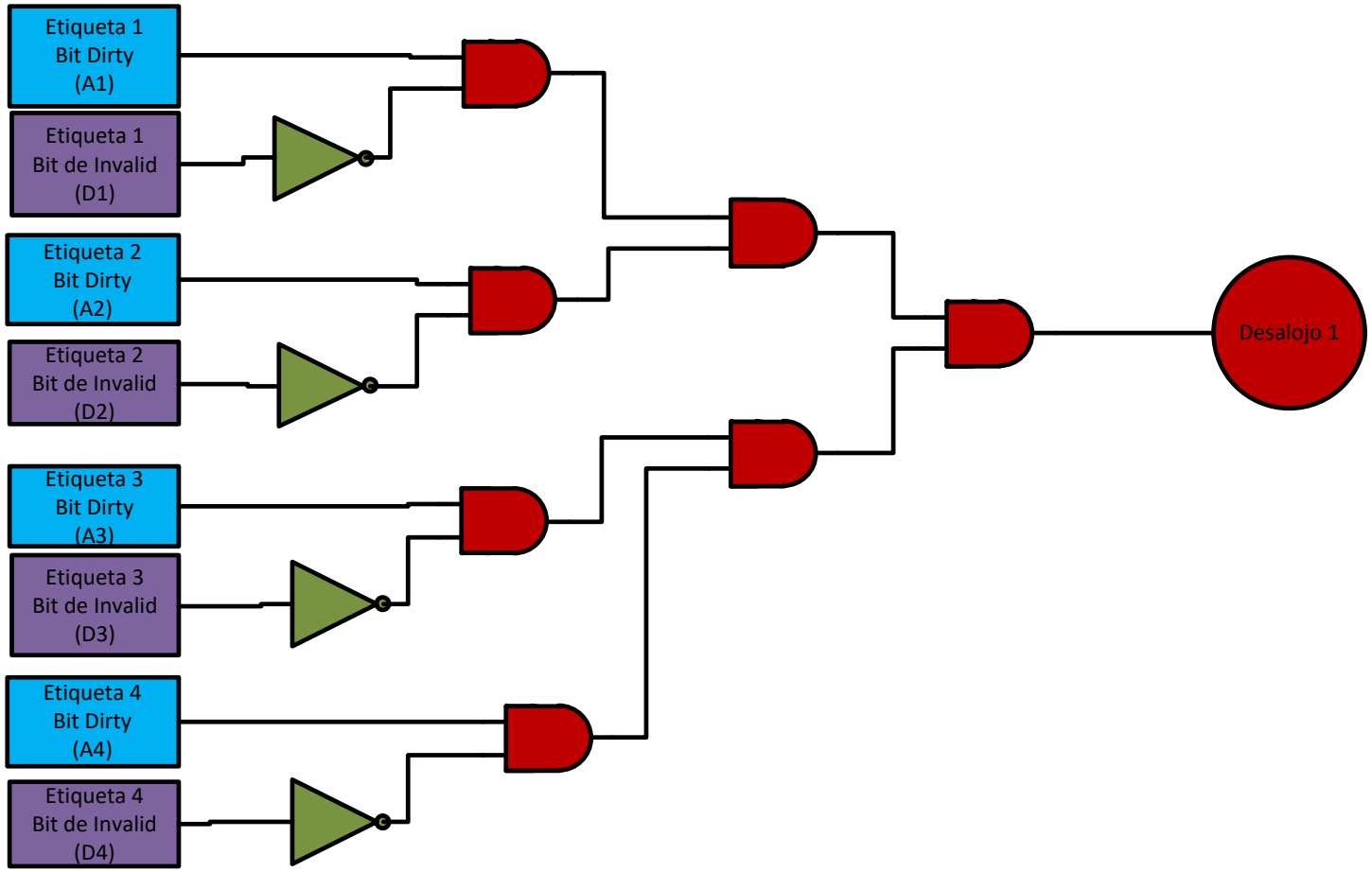
$$G0= (C2)+(C4*\sim C1)$$


C1
C2
C3
C4
G0
G1
Desalojo 2

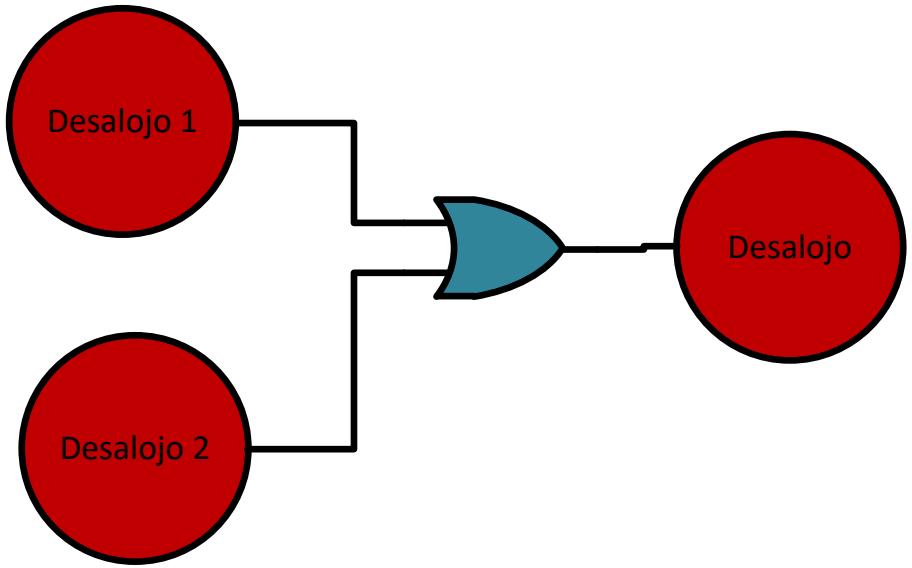
Determina el primer caso cuando se debe hacer desalojo

Desalojo 1				
Entradas				Salida
A1*~D1	A2*~D2	A3*~D3	A4*~D4	Desalojo 1
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

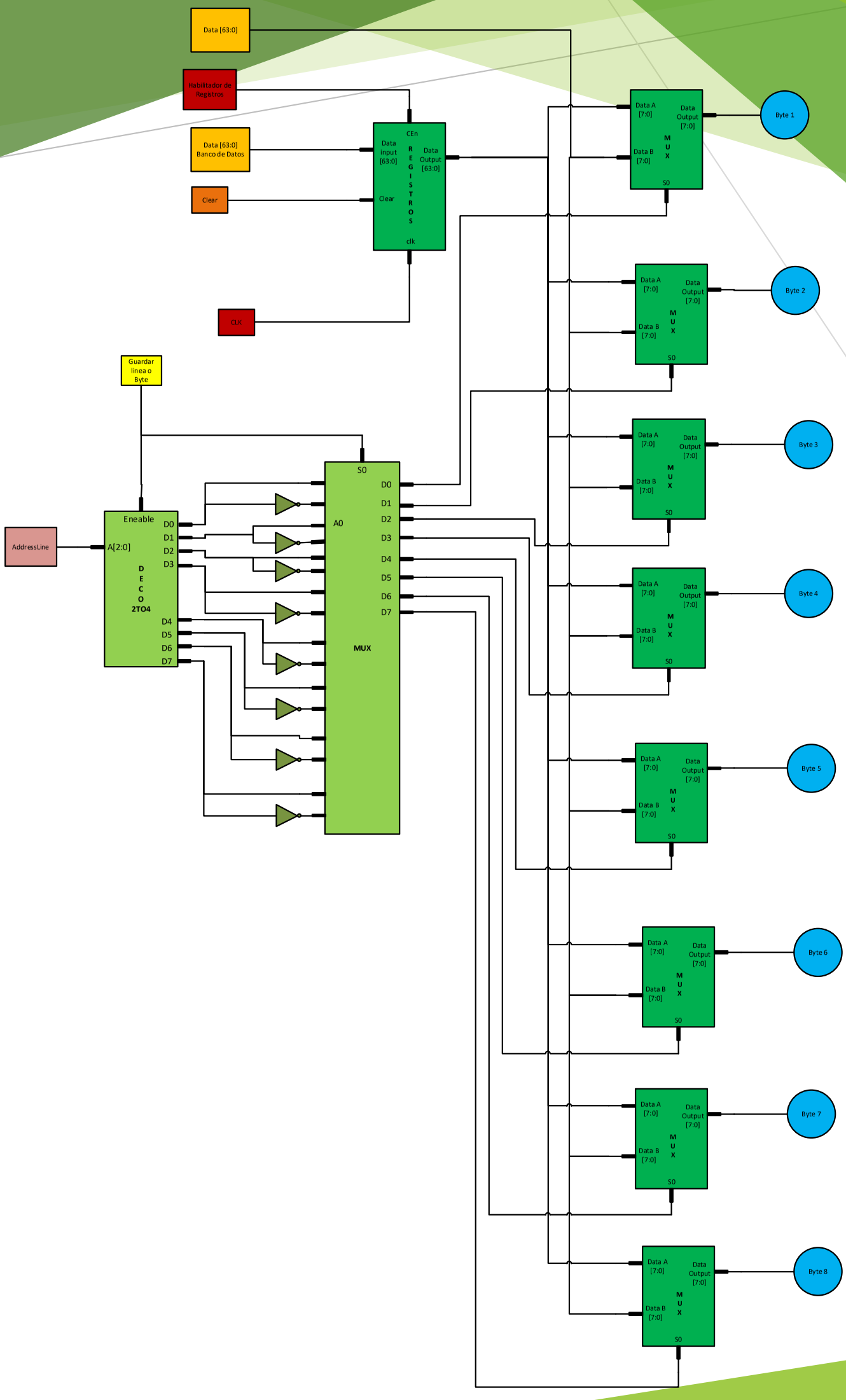
Si Todas las líneas son validas y además todas se encuentran sucias, significa que hay que hacer un desalojo


Etiqueta 1
Bit Dirty
(A1)
Etiqueta 1
Bit de Invalid
(D1)
Etiqueta 2
Bit Dirty
(A2)
Etiqueta 2
Bit de Invalid
(D2)
Etiqueta 3
Bit Dirty
(A3)
Etiqueta 3
Bit de Invalid
(D3)
Etiqueta 4
Bit Dirty
(A4)
Etiqueta 4
Bit de Invalid
(D4)
Desalojo 1

Unificación de las señales de desalojo



Lógica que permite diferenciar entre Guardar una línea o Guardar un byte


Data [63:0]
Habilitador de Registros
Data [63:0] Banco de Datos
Clear
CLK
CEn
Data input [63:0]
REGISTROS
Data Output [63:0]
clk
Guardar linea o Byte
AddressLine
Eneable
A[2:0]
DECO 2TO4
D0
D1
D2
D3
D4
D5
D6
D7
S0
A0
MUX
Data A [7:0]
Data B [7:0]
MUX
Data Output [7:0]
S0
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
Byte 6
Byte 7
Byte 8

Control

Señales de Entrada			Señales de Salida		
	Señal	Caracteristica		Señal	Caracteristica
1	PNDNG	Indica que la memoria principal tiene un dato listo en el bus	1	CLEAR MAIN REG	Inicializa (1) los registros principales donde entran los datos
2	Ejecute	Desencadena el proceso	2	ENEABLE MAIN REG	Desabilita (0) o Habilita (1) los registros principales donde entran datos
3	Lectura/Escritura	Indica Escritura(0), lectura (1)	3	SEL MUX BANK	Selecciona entre actualizar etiquetas (1) o usar la memoria de etiquetas (0)
4	Hit	Indica si ocurrió un hit	4	CLEAR TAG BANK REG	Inicializa (1) los registros auxiliares utilizados dentro del banco de etiquetas para actualizar bits de dirty y ultimo en ser usado
5	Desalojo	Indica si se debe hacer un desalojo	5	SEL_MUX_MEM_0	Selecciona dentro de cual banco de arregles se va a escribir o de cual se va a dejar pasar el dato A:Guardar B: Desalojo C:Lectura
6			6	SEL_MUX_MEM_1	
7			7	ENEABLE REG	Habilita (1) registro que permite guardar byte o linea dentro de la memoria de datos
8			8	CLEAR LDG REG	Limpia (1) el registro registro que permite guardar byte o linea dentro de la memoria de datos
9			9	WRITE ENEABLE	Escribe (0) o lee(1) en los bancos
10			10	BANKS ENEABLE	Habilita (1) o deshabilita (0) los bancos de registros
11			11	R/W	Indica a el tipo de solicitud que se realizará a la memoria Read(1) o Write(0)
12			12	CLEAR FORMADOR	Limpiar (1) el registro que almacena el mensaje a ser enviado a la memoria principal
13			13	ENEABLE FORMADOR	Habilita (1) o deshabilita (0) el registro que almacena el mensaje a ser enviado a la memoria principal
14			14		

Maquina de Estados

Considerar que si la línea esta invalida entonces no hace falta enviar a memoria

Eliminar el bit de RDY

Estado	Entradas						Estado	Salidas												
Actual	PNDNG	Ejecute	Lectura/Escritura	Hit	Desalojo	RDY	Siguiente	1	2	3	4	5	6	7	8	9	10	11	12	13
Inicio	x	0	x	x	x	x	Inicio	1	1	x	1	x	x	x	1	x	0	x	1	x
	x	1	0	x	x	x	Escribir	0	0	0	0	1	0	0	0	1	1	x	x	0
	x	1	1	x	x	x	Leer	0	0	0	0	1	0	0	0	1	1	x	x	0
Leer	x	x	x	x	x	0	Leer	0	0	0	0	1	0	0	0	1	1	x	x	0
	x	x	x	0	x	1	Traer Dato de Memoria	0	1	0	0	0	0	x	1	1	1	1	0	1
	x	x	x	1	x	1	Dato Listo	0	0	0	0	1	0	0	x	1	1	x	x	x
Traer Dato de Memoria	0	x	x	x	x	x	Traer Dato de Memoria	0	1	0	0	0	0	x	1	1	1	1	0	1
	1	x	x	x	x	x	Escribir	0	0	0	0	x	x	0	0	1	1	x	x	0
Escribir	x	x	x	x	x	0	Escribir	0	0	0	0	x	x	0	0	1	1	x	x	0
	x	x	x	x	0	1	Dato Escrito	0	0	0	0	0	0	1	0	0	1	x	x	x
	x	x	x	x	1	1	Desalojo	0	0	0	0	0	1	1	0	0	1	0	0	1
Desalojo	x	x	x	x	x	x	Dato Escrito	0	0	0	0	0	0	1	0	0	1	x	x	x
Dato Escrito	x	x	0	x	x	x	Actualizar etiquetas	0	0	1	0	1	0	1	0	0	1	x	0	0
	x	x	1	x	x	x	Dato Listo	0	1	0	0	1	0	0	x	1	1	x	x	x
Dato Listo	x	x	x	x	x	x	Actualizar etiquetas	0	0	1	0	1	0	1	0	0	1	x	0	0
Actualizar Etiquetas	x	x	x	x	x	x	Inicio	1	1	x	1	x	x	x	1	x	0	x	1	x