

Técnicas digitales 1

Resolución T.P.3º

1. Demostrar igualdades	3
a)	3
b)	3
c)	3
2. 3. Minimización y Demorgan	3
a)	3
b)	4
c)	5
d)	6
e)	7
4. Comparador de magnitud de 2 bits	10
a) Tabla de la verdad	10
b) Minimización	10
c) Gráfico a nivel de compuertas	11
5. Multiplicador de 2 bits signado	12
a) Cantidad de bits de la salida	12
b) Tabla de la verdad	13
c) Minimización	13
d) Gráfico a nivel de compuertas	14
6. Comparador de 2 bits signado	15
a) Circuito a nivel de bloques	15
b) Justificación	15
7. Display 7 segmentos cátodo común	16
a) Tabla de la verdad	16
b) Minimización	16
c) Gráfico a nivel de compuertas	18
d) Implementación con decodificador 4 a 8.	19
e) Implementación con multiplexores de 4 entradas de control	19
f) Implementación con multiplexores de 3 entradas de control	19
8. Conversor de unsigned a signed de 4 bits	21
9. Dado	21
a) Cantidad de salidas	21
b) Funciones lógicas	21
c) Compuertas	22

d) Diagrama temporal	22
e) Entradas rezagadas	23
10. Unsigned left shifter (8 bits)	23
a) Tabla de la verdad	23
b) Circuito con multiplexores	23
11. Unsigned right shifter (8 bits)	24
12. Unsigned left shifter, hasta MSB = 0	25
a) Tabla de la verdad	25
b) Gráfico con multiplexores	26
13. Unsigned left shifter until MSB = 1	28
a) Tabla de la verdad	28
14. Signed and unsigned right shifter	30
a) Tabla de la verdad	30
b) Circuito con multiplexores	30
15. Saturador de 16 a 8 bits	31
a) Tabla de la verdad	31
b) Circuito a nivel de compuertas	31
16. Restador del valor “2” a un número de N bits.	33
17. Sumador del valor “2” a un número de N bits.	35
18. Contador de “1” en número de 5 bits	37
a) Cantidad de salidas	37
b) Tabla de la verdad	37
c) Minimización	38
d) Implementación con sumadores	39
19. Conversor °Celsius (-273 ; 1000) a Kelvin	40
a) Cantidad de salidas y entradas	40
b) Tabla de la verdad (incompleto)	40
c) Circuito a nivel de compuertas (incompleto)	41
d) Circuito con full adder	41
20. Promedio de 4 números de 4 bits CA2	42
a) Cantidad de salidas	42
b) Tabla de la verdad	42
c) Implementación con full adder	42
21. Multiplicador por 3/2	43
a) Circuito a nivel de bloques	43
Extra 1. Sumador	44

1. Demostrar igualdades

a)

$$x + y.z = (x + y)(x + z)$$

$$x + y.z = x.x + x.z + y.x + y.z \Rightarrow \text{Sea } x.x = x$$

$$x + y.z = x + x.z + y.x + y.z$$

$$x + y.z = x.(1 + z + y) + y.z \Rightarrow \text{Sea } 1 + z + y = 1 \forall z, y$$

$$x + y.z = x + y.z$$

b)

$$(x + y).(x + \bar{y}) = x$$

$$x.x + x.\bar{y} + x.y + y.\bar{y} = x \Rightarrow \text{Sea } x.x = x, y.\bar{y} = 0$$

$$x + x.\bar{y} + x.y = x$$

$$x.(\bar{y} + y) = x \Rightarrow \text{Sea } \bar{y} + y = 1 \forall y, \Rightarrow x.1 = x$$

$$x = x$$

c)

$$x.y + x.\bar{y} = x$$

$$x.(y + \bar{y}) = x \Rightarrow \text{Sea } y + \bar{y} = 1$$

$$x = x$$

2. 3. Minimización y Demorgan

Nota: cuando se construyen los diagramas con sólo compuertas NAND o NOR, las variables negadas pueden obtenerse poniendo en ambas entradas de la compuerta NAND o NOR la variable en cuestión.

a)

C	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

		BA			
		00	01	11	10
C	0	1	1	0	0
	1	1	0	1	0

$$f(c, b, a) = \bar{C}.\bar{B} + \bar{B}.\bar{A} + C.B.A$$

$$f(c, b, a) = (C + \bar{B}).(\bar{B} + A).(\bar{C} + B + \bar{A})$$

$$f(c, b, a) = \overline{\overline{\bar{C}.\bar{B} + \bar{B}.\bar{A} + C.B.A}}$$

$$f(c, b, a) = \overline{(\bar{C}.\bar{B}).(\bar{B}.\bar{A}).(C.B.A)}$$

$$f(c, b, a) = \overline{(C + \bar{B}).(\bar{B} + A).(\bar{C} + B + \bar{A})}$$

$$f(c, b, a) = \overline{(C + \bar{B}) + (\bar{B} + A) + (\bar{C} + B + \bar{A})}$$

b)

C	B	A	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

		BA			
		00	01	11	10
C	0	1	0	0	1
	1	0	1	1	0

$$f(c, b, a) = \bar{C} \cdot \bar{A} + C \cdot A$$

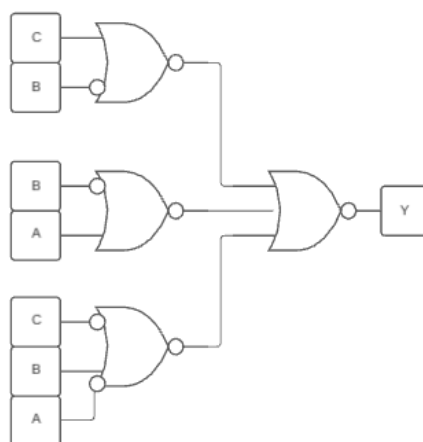
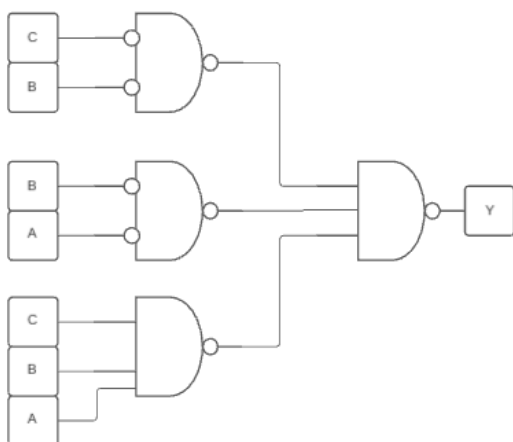
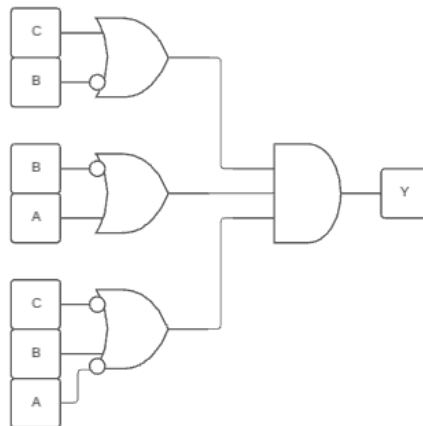
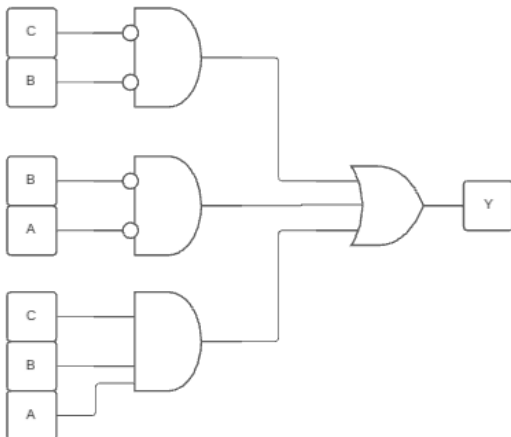
$$f(c, b, a) = (C + \bar{A}) \cdot (\bar{C} + A)$$

$$f(c, b, a) = \overline{\overline{\bar{C} \cdot \bar{A} + C \cdot A}}$$

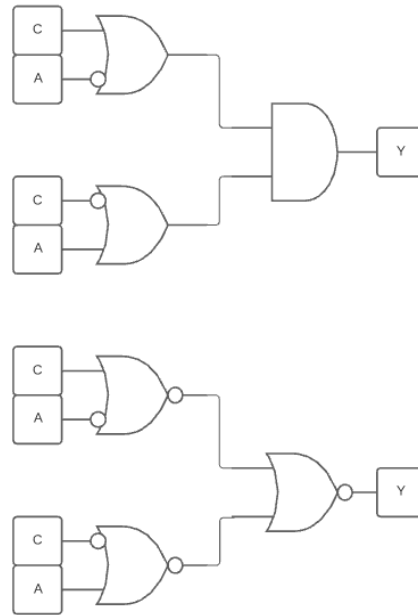
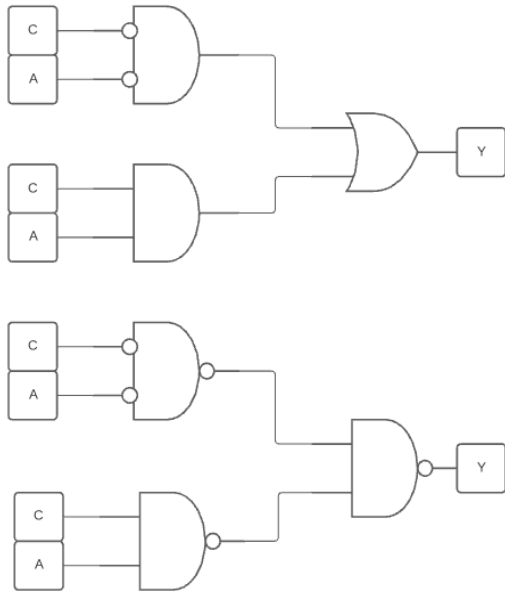
$$f(c, b, a) = \overline{(\bar{C} \cdot \bar{A}) \cdot (C \cdot A)}$$

$$f(c, b, a) = \overline{(C + \bar{A}) \cdot (\bar{C} + A)}$$

Gráficos parte A:



Gráficos parte B:



c)

C	B	A	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$$f(c, b, a) = \overline{\overline{B} \cdot \overline{A}} + \overline{C \cdot \overline{A}} + \overline{\overline{C} \cdot B \cdot A}$$

$$f(c, b, a) = (\overline{B \cdot A}) \cdot (\overline{C \cdot A}) \cdot (\overline{\overline{C} \cdot B \cdot A})$$

$$f(c, b, a) = \overline{(B + \overline{A}) \cdot (\overline{C} + \overline{A}) \cdot (C + \overline{B} + A)}$$

$$f(c, b, a) = \overline{(B + \overline{A}) + (\overline{C} + \overline{A}) + (C + \overline{B} + A)}$$

		BA			
		00	01	11	10
C	0	1	0	1	0
	1	1	0	0	1

$$f(c, b, a) = \overline{B} \cdot \overline{A} + C \cdot \overline{A} + \overline{C} \cdot B \cdot A$$

$$f(c, b, a) = (B + \overline{A}) \cdot (\overline{C} + \overline{A}) \cdot (C + \overline{B} + A)$$

d)

D	C	B	A	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0

1	1	1	1	1
---	---	---	---	---

		BA			
		00	01	11	10
DC	00	1	1	1	1
	01	0	0	1	0
	11	1	1	1	0
	10	0	0	0	0

$$f(d, c, b, a) = \overline{D}.\overline{C} + D.C.\overline{B} + C.B.A$$

$$f(d, c, b, a) = (D + \overline{C} + B).(\overline{C} + \overline{B} + A).(\overline{D} + C)$$

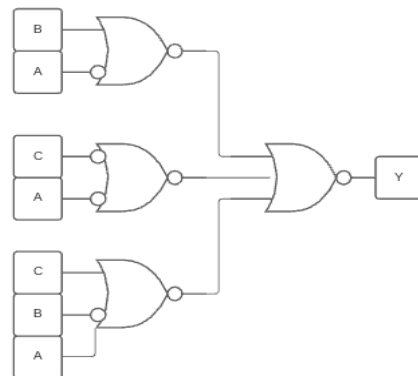
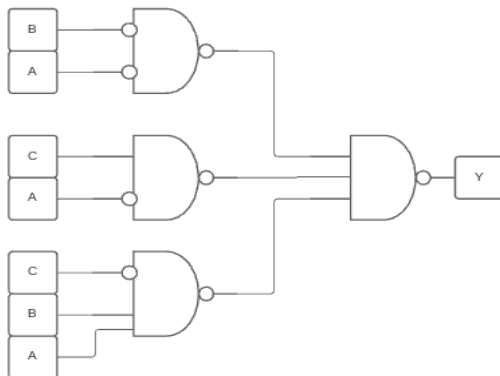
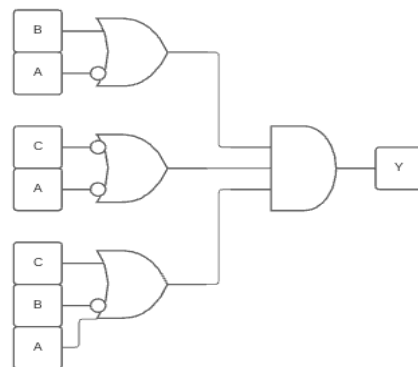
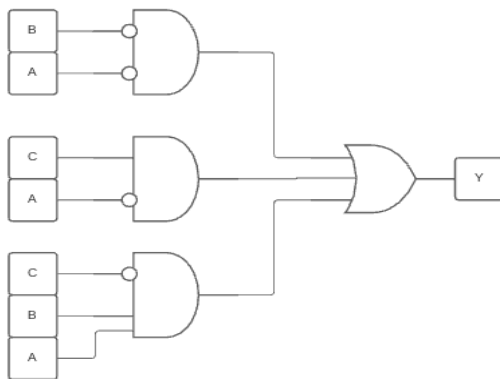
$$f(d, c, b, a) = \overline{\overline{D}.\overline{C} + D.C.\overline{B} + C.B.A}$$

$$f(d, c, b, a) = \overline{(\overline{D}.\overline{C}). (D.C.\overline{B}). (C.B.A)}$$

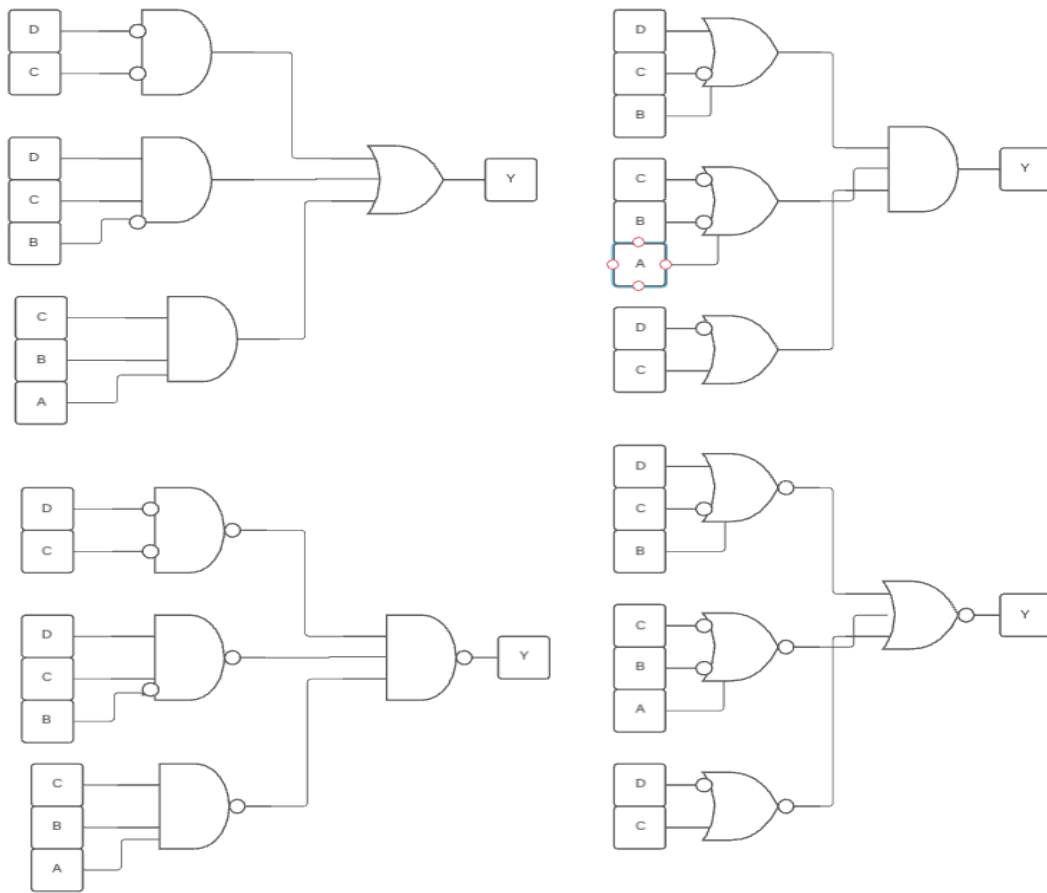
$$f(d, c, b, a) = \overline{(D + \overline{C} + B).(\overline{C} + \overline{B} + A).(\overline{D} + C)}$$

$$f(d, c, b, a) = \overline{(D + \overline{C} + B) + (\overline{C} + \overline{B} + A) + (\overline{D} + C)}$$

Gráficos parte C:



Gráficos parte D:



e)

D	C	B	A	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1

1	1	1	0	0
1	1	1	1	0

		BA			
		00	01	11	10
DC	00	0	0	1	1
	01	1	0	0	1
	11	1	1	0	0
	10	0	1	1	0

$$f(d, c, b, a) = \overline{D}.\overline{C}.B + \overline{D}.C.\overline{A} + D.C.\overline{B} + D.\overline{C}.A$$

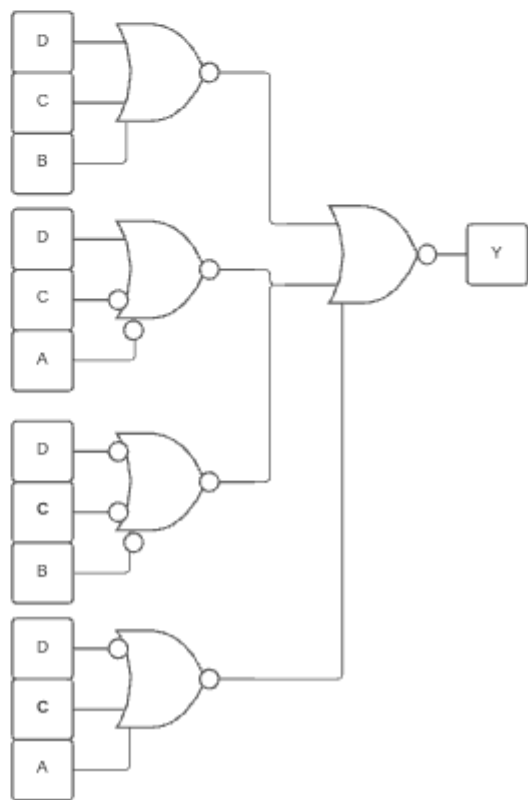
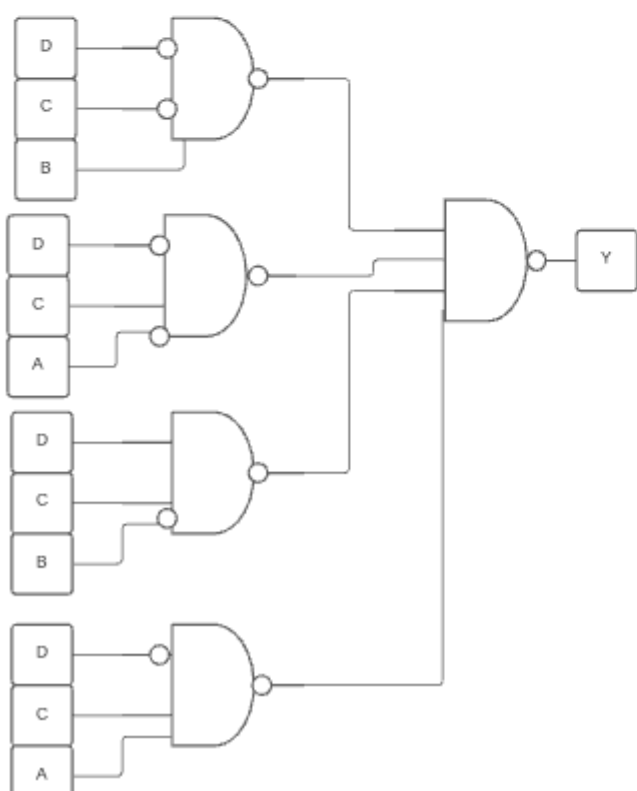
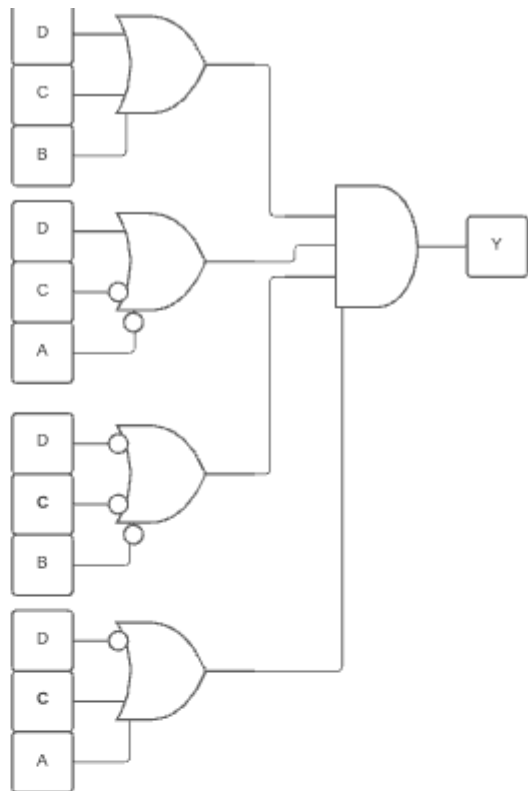
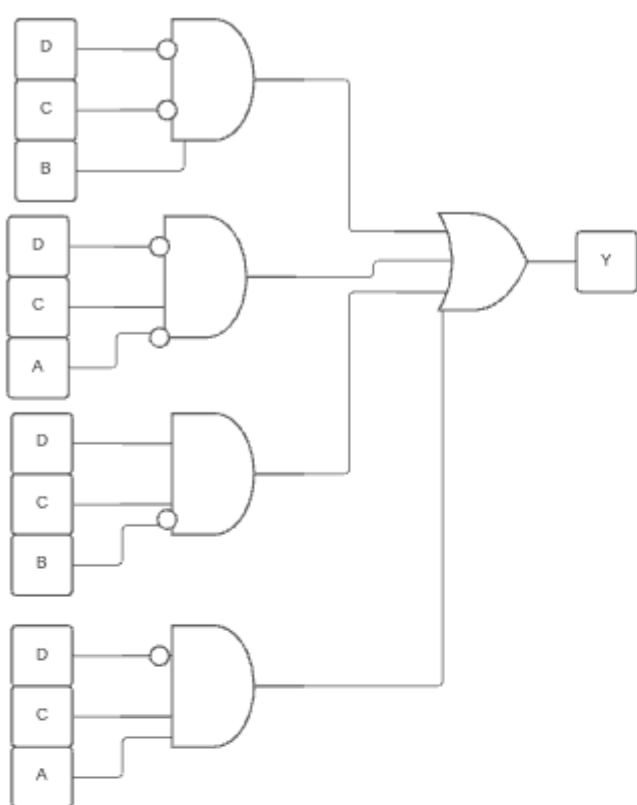
$$f(d, c, b, a) = (D + C + B). (D + \overline{C} + \overline{A}). (\overline{D} + \overline{C} + \overline{B}). (\overline{D} + C + A)$$

$$f(d, c, b, a) = \overline{\overline{\overline{D}.\overline{C}.B + \overline{D}.C.\overline{A} + D.C.\overline{B} + D.\overline{C}.A}}$$

$$f(d, c, b, a) = \overline{(\overline{D}.\overline{C}.B). (\overline{D}.C.\overline{A}). (D.C.\overline{B}). (D.\overline{C}.A)}$$

$$f(d, c, b, a) = \overline{(D + C + B). (D + \overline{C} + \overline{A}). (\overline{D} + \overline{C} + \overline{B}). (\overline{D} + C + A)}$$

$$f(d, c, b, a) = \overline{(\overline{D} + C + B) + (\overline{D} + \overline{C} + \overline{A}) + (\overline{D} + \overline{C} + \overline{B}) + (\overline{D} + C + A)}$$



4. Comparador de magnitud de 2 bits

a) Tabla de la verdad

A1	A0	B1	B0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

b) Minimización

Uso tres mapas de Karnaugh, para cada salida distinta:

		A>B			
		B1 B0			
		00	01	11	10
A1 A0	00	0	0	0	0
	01	1	0	0	0
	11	1	1	0	1
	10	1	1	0	0

$$A > B = A_1 \cdot \overline{B_1} + A_0 \cdot \overline{B_1} \cdot \overline{B_0} + A_1 \cdot A_0 \cdot \overline{B_0}$$

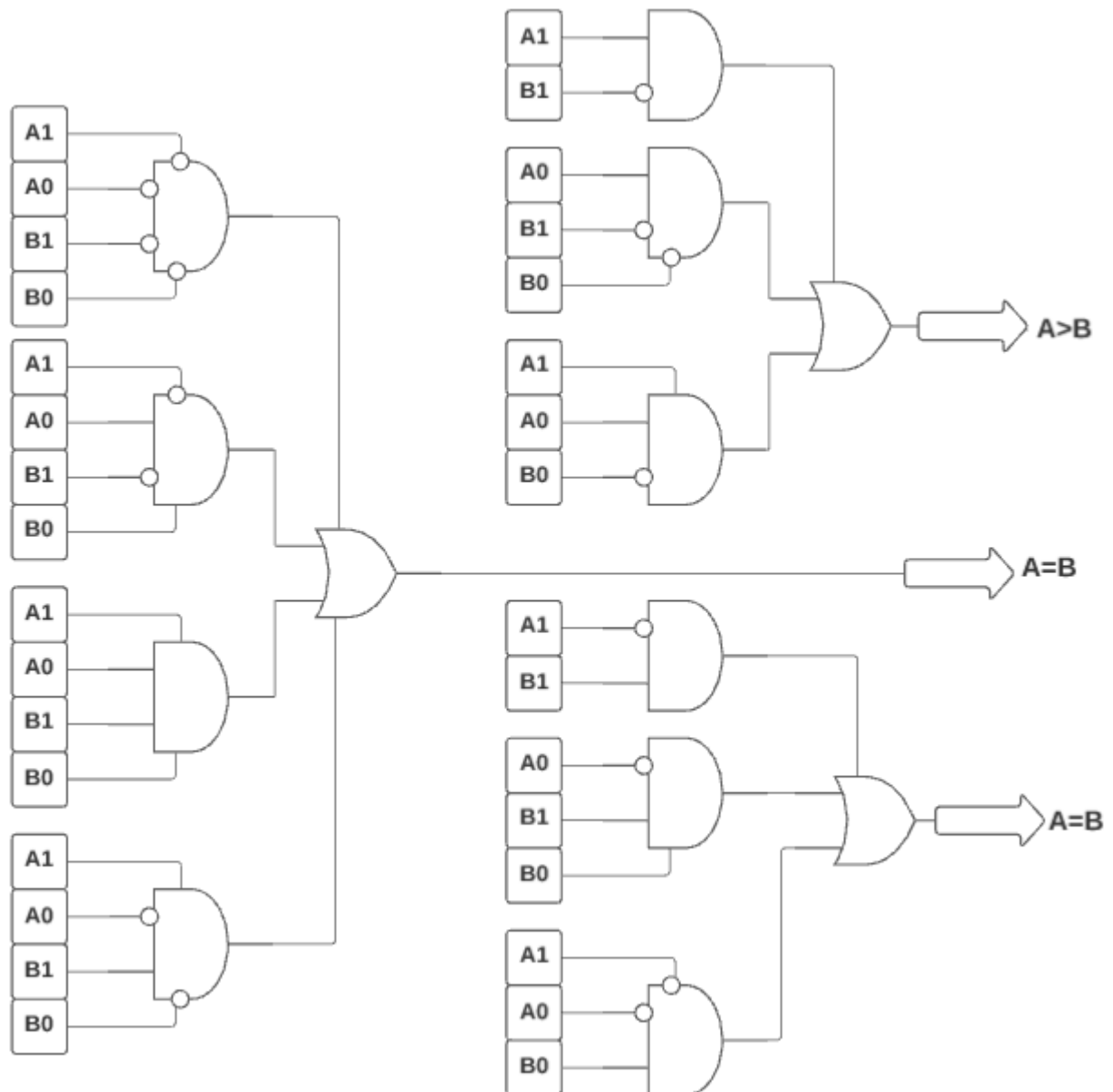
		A=B			
		B1 B0			
		00	01	11	10
A1 A0	00	1	0	0	0
	01	0	1	0	0
	11	0	0	1	0
	10	0	0	0	1

$$(A = B) = \overline{A_1} \cdot \overline{A_0} \cdot \overline{B_1} \cdot \overline{B_0} + \overline{A_1} \cdot A_0 \cdot \overline{B_1} \cdot B_0 + A_1 \cdot A_0 \cdot B_1 \cdot B_0 + A_1 \cdot \overline{A_0} \cdot B_1 \cdot \overline{B_0}$$

		A<B			
		B1 B0			
		00	01	11	10
A1 A0	00	0	1	1	1
	01	0	0	1	1
	11	0	0	0	0
	10	0	0	1	0

$$A < B = \overline{A_1} \cdot B_1 + \overline{A_0} \cdot B_1 \cdot B_0 + \overline{A_1} \cdot \overline{A_0} \cdot B_0$$

c) Gráfico a nivel de compuertas



5. Multiplicador de 2 bits signado

a) Cantidad de bits de la salida

Los posibles números a multiplicar, con dos bits, son: $\{-2, -1, 0, 1\}$, el valor más grande lo obtengo que $-2 \cdot (-2) = 4 = 0100$. Por lo que la salida debe ser de 4 bits.

b) Tabla de la verdad

A1	A0	B1	B0	Y
0	0	0	0	0000
0	0	0	1	0000
0	0	1	0	0000
0	0	1	1	0000
0	1	0	0	0000
0	1	0	1	0001
0	1	1	0	1110
0	1	1	1	1111
1	0	0	0	0000
1	0	0	1	1110
1	0	1	0	0100
1	0	1	1	0010
1	1	0	0	0000
1	1	0	1	1111
1	1	1	0	0010
1	1	1	1	0001

c) Minimización

Tengo que hacer 4 mapas de Karnaugh, uno para cada bit de salida.

		B1 B0			
		00	01	11	10
A1 A0	00	0	0	0	0
	01	0	0	1	1
	11	0	1	0	0
	10	0	1	0	0

$$Y(3) = A_1 \cdot \overline{B_1} \cdot B_0 + \overline{A_1} \cdot A_0 \cdot B_1$$

		B1 B0			
		00	01	11	10
A1 A0	00	0	0	0	0
	01	0	0	1	1
	11	0	1	0	0
	10	0	1	0	1

$$Y(2) = A_1 \cdot \overline{B_1} \cdot B_0 + \overline{A_1} \cdot A_0 \cdot B_1 + A_1 \cdot \overline{A_0} \cdot B_1 \cdot \overline{B_0}$$

		B1 B0			
		00	01	11	10
A1 A0	00	0	0	0	0
	01	0	0	1	1
	11	0	1	0	1
	10	0	1	1	0

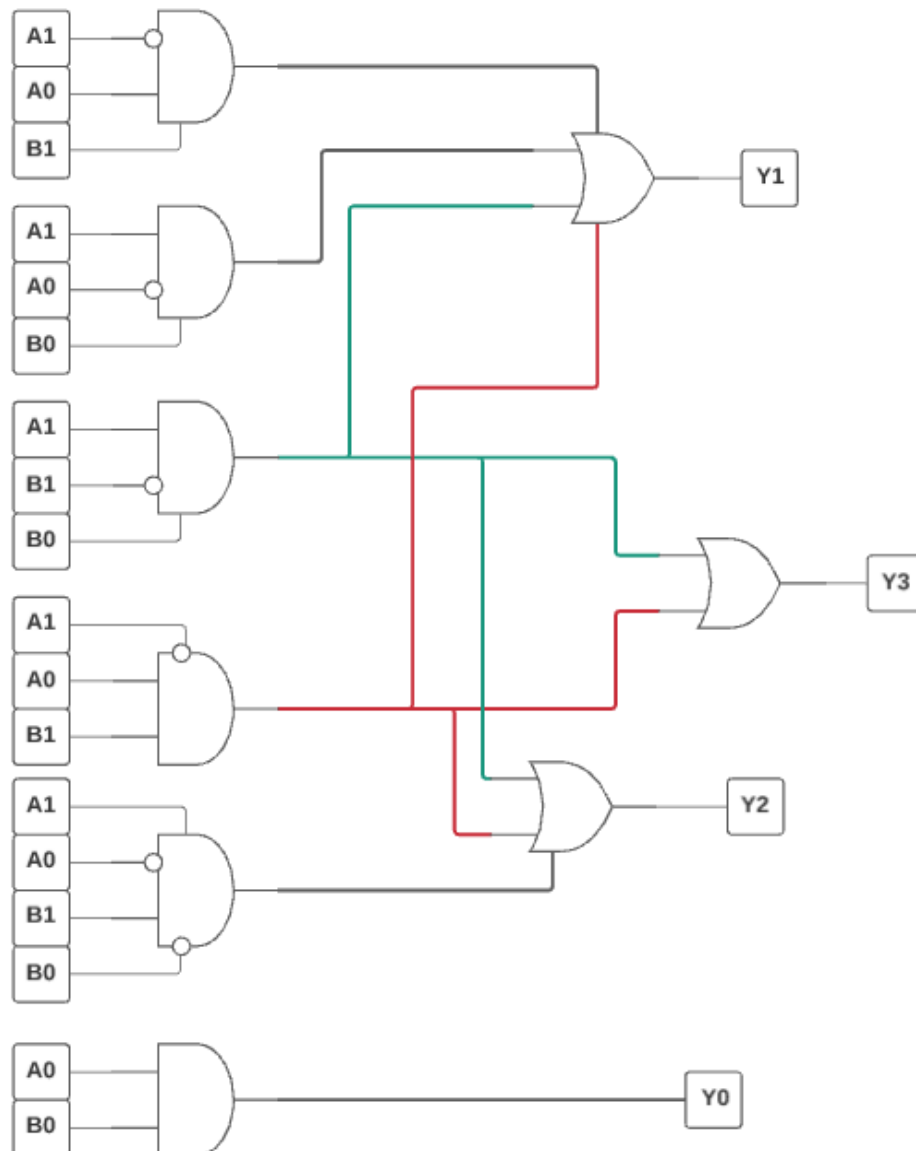
		B1 B0			
		00	01	11	10
A1 A0	00	0	0	0	0
	01	0	1	1	0
	11	0	1	1	0
	10	0	0	0	0

$$Y(1) = A_1 \cdot \overline{B_1} \cdot B_0 + \overline{A_1} \cdot A_0 \cdot B_1 + A_1 \cdot \overline{A_0} \cdot B_0 + \overline{A_1}$$

$$Y(0) = A_0 \cdot B_0$$

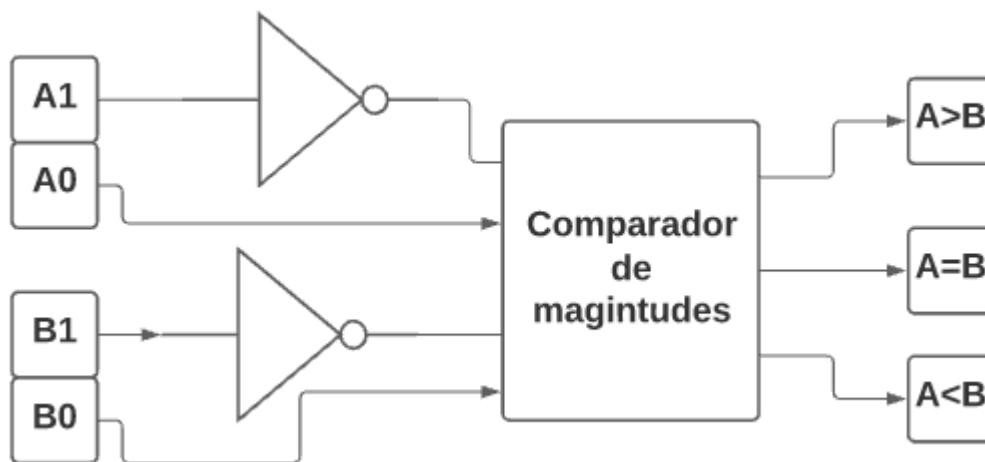
d) Gráfico a nivel de compuertas

Viendo los tres bits más significativos, todos comparten el naranja y el verde en el mapa de Karnaugh, así que puedo usar la salida de esas compuertas tres veces.



6. Comparador de 2 bits signado

a) Circuito a nivel de bloques



b) Justificación

Para dos bits, sea el número a comparar:

A1	A0	A	Niego el MSB	$\sim A1$	A0	A
0	0	0		0	0	2
0	1	1		0	1	3
1	0	-2		1	0	1
1	1	-1		1	1	0

Puede verse que al negar el MSB, la magnitud del número está ordenada de la misma forma que los números signados en cuanto a "valor".

7. Display 7 segmentos cátodo común

a) Tabla de la verdad

Para los números entre 10 y 16, los considero como "x", osea, que pueden tomar cualquier valor. Esto va a simplificar el circuito final.

I3	I2	I1	I0	A	B	C	D	E	F	G
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x

b) Minimización

		A			
		I1 I0			
		00	01	11	10
I3 I2	00	1	0	1	1
	01	0	1	1	1
	11	x	x	x	x
	10	1	1	x	x

$$A = I_3 + I_1 + I_2 \cdot \overline{I_1} \cdot I_0 + \overline{I_2} \cdot \overline{I_1} \cdot \overline{I_0}$$

		B			
		I1 I0			
		00	01	11	10
I3 I2	00	1	1	1	1
	01	1	0	1	0
	11	x	x	x	x
	10	1	1	x	x

$$B = I_3 + \overline{I_1} \cdot \overline{I_0} + I_1 \cdot I_0 + \overline{I_3} \cdot \overline{I_2}$$

		C			
		I1 I0			
		00	01	11	10
I3 I2	00	1	1	1	0
	01	1	1	1	1
	11	x	x	x	x
	10	1	1	x	x

$$C = I_3 + I_2 + \bar{I}_1 + I_0$$

		D			
		I1 I0			
		00	01	11	10
I3 I2	00	1	0	1	1
	01	0	1	0	1
	11	x	x	x	x
	10	1	1	x	x

$$D = I_3 + I_1 \cdot \bar{I}_0 + \bar{I}_2 \cdot \bar{I}_1 \cdot \bar{I}_0 + I_2 \cdot \bar{I}_1 \cdot I_0 + \bar{I}_3 \cdot \bar{I}_2 \cdot I_1$$

		E			
		I1 I0			
		00	01	11	10
I3 I2	00	1	0	0	1
	01	0	0	0	1
	11	x	x	x	x
	10	1	0	x	x

$$E = I_1 \cdot \bar{I}_0 + \bar{I}_2 \cdot \bar{I}_1 \cdot \bar{I}_0$$

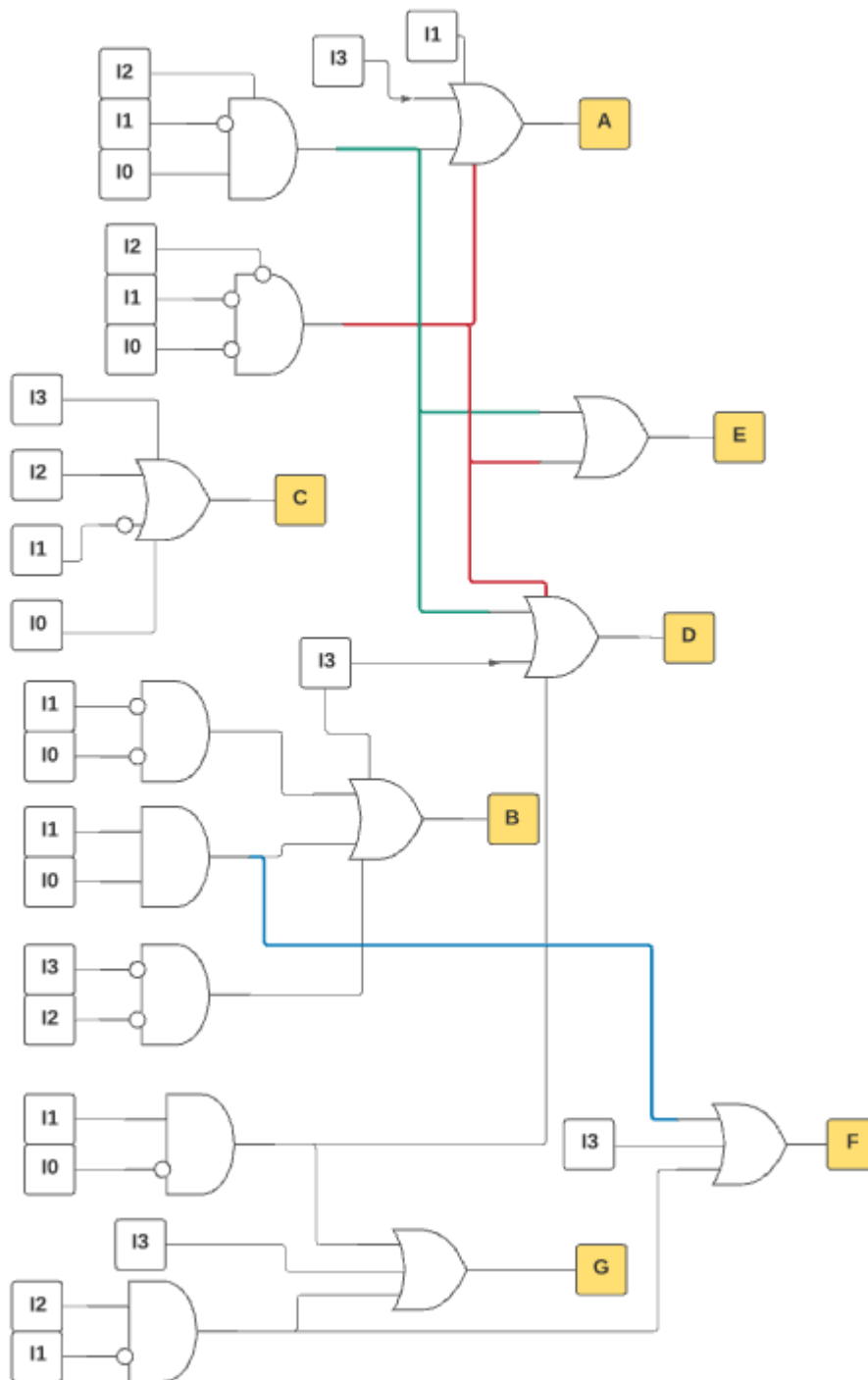
		F			
		I1 I0			
		00	01	11	10
I3 I2	00	1	0	0	0
	01	1	1	0	1
	11	x	x	x	x
	10	1	1	x	x

$$F = I_3 + \bar{I}_1 \cdot \bar{I}_0 + I_2 \cdot \bar{I}_1 + I_2 \cdot I_1 \cdot \bar{I}_0$$

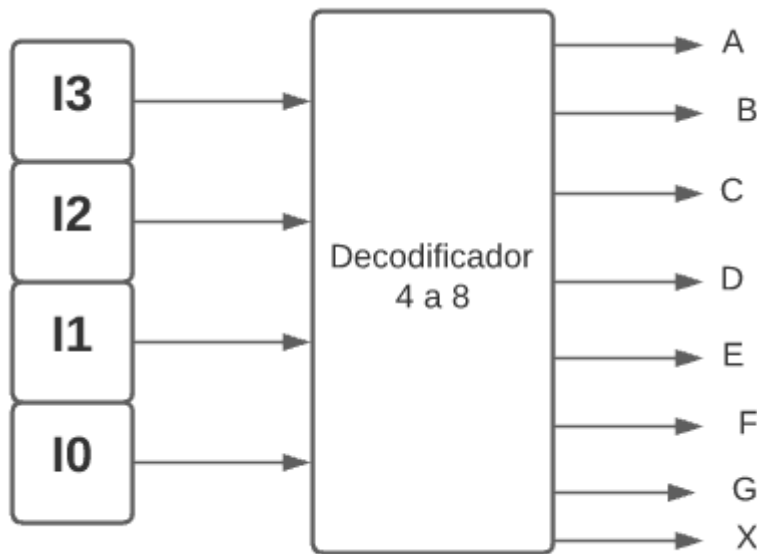
		G			
		I1 I0			
		00	01	11	10
I3 I2	00	0	0	1	1
	01	1	1	0	1
	11	x	x	x	x
	10	1	1	x	x

$$G = I_3 + I_2 \cdot \bar{I}_1 + I_1 \cdot \bar{I}_0$$

c) Gráfico a nivel de compuertas

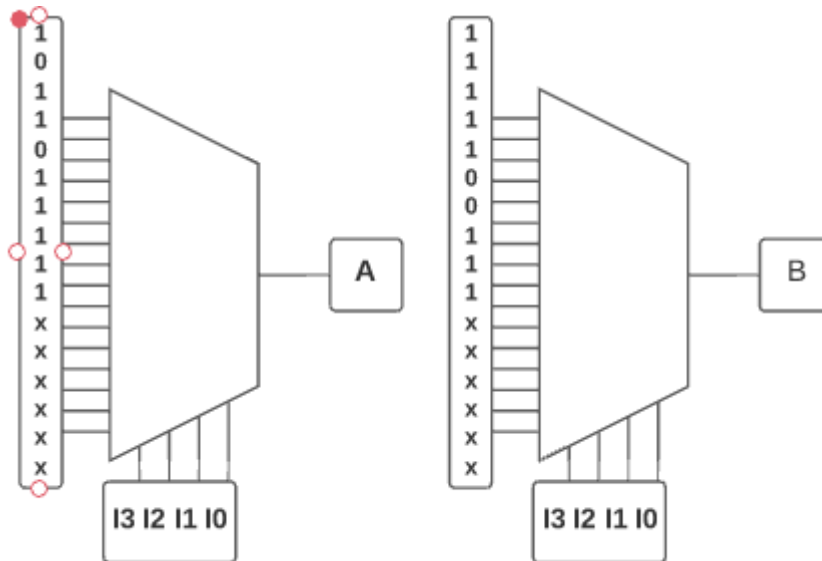


d) Implementación con decodificador 4 a 8.



e) Implementación con multiplexores de 4 entradas de control

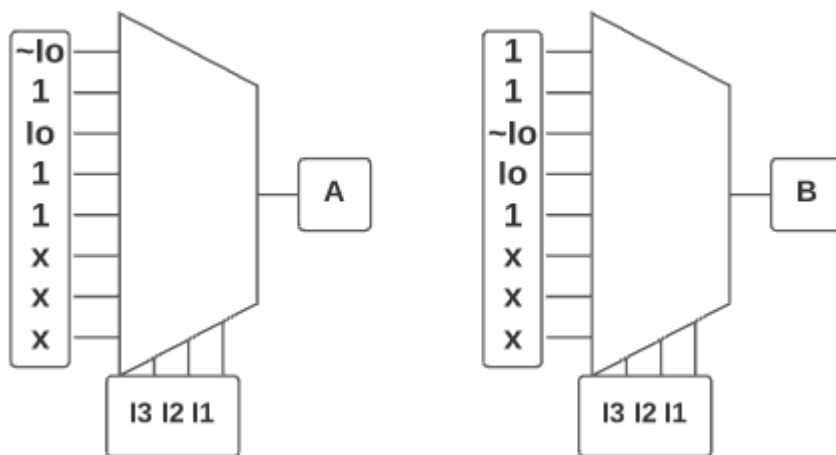
Para implementarlo con multiplexor, ponemos en las entradas de control las entradas reales, y en las entradas del multiplexor las salidas tal cual aparecen en la tabla. Solo se hacen los gráficos para A y B, para el resto es lo mismo.



f) Implementación con multiplexores de 3 entradas de control

Voy agarrando de la tabla original de a dos elementos, y elijo qué poner: "0", "1", I_0 , $\overline{I_0}$. El gráfico lo hago solo para las primeras dos salidas, para el resto es lo mismo.

I3	I2	I1	A	B	C	D	E	F	G
0	0	0	~I0	1	1	~I0	~I0	~I0	0
0	0	1	1	1	I0	1	~I0	0	1
0	1	0	I0	~I0	1	I0	0	1	1
0	1	1	1	I0	1	~I0	~I0	~I0	~I0
1	0	0	1	1	1	1	~I0	1	1
1	0	1	x	x	x	x	x	x	x
1	1	0	x	x	x	x	x	x	x
1	1	1	x	x	x	x	x	x	x



8. Conversor de unsigned a signed de 4 bits

Se indicó no hacerlo.

9. Dado

a) Cantidad de salidas

Los LEDS los nombro así:

C		G
B	D	F
A		E

Al ver la tabla de la verdad, notamos que se cumple que: $A = G$, $B = F$, $C = E$

I2	I1	I0	A	B	C	D	E	F	G
0	0	0	x	x	x	x	x	x	x
0	0	1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	1	0	0
0	1	1	0	0	1	1	1	0	0
1	0	0	1	0	1	0	1	0	1
1	0	1	1	0	1	1	1	0	1
1	1	0	1	1	1	0	1	1	1
1	1	1	x	x	x	x	x	x	x

Por lo que en realidad solo tenemos 4 salidas distintas: A, B, C y D.

b) Funciones lógicas

A = G					
I1 I0					
	00	01	11	10	
I2	0	x	0	0	0
	1	1	1	x	1

$$A = G = I_2$$

B = F					
I1 I0					
	00	01	11	10	
I2	0	x	0	0	0
	1	0	0	x	1

$$B = F = I_2 \cdot I_1$$

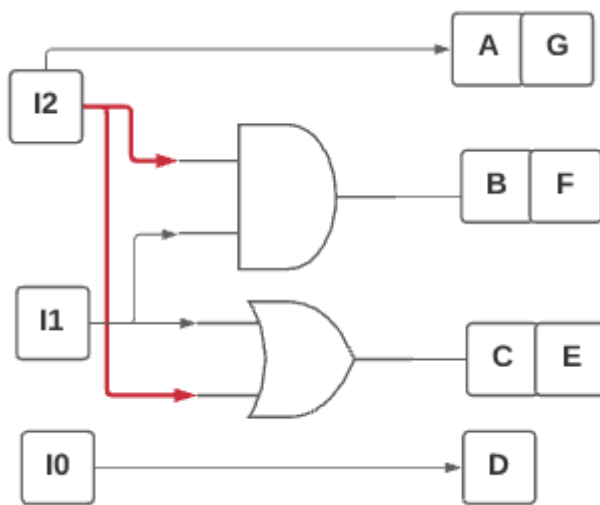
C = E					
		I1 I0			
		00	01	11	10
I2	0	x	0	1	1
	1	1	1	x	1

$$C = E = I_2 + I_1$$

D					
		I1 I0			
		00	01	11	10
I2	0	x	1	1	0
	1	0	1	x	0

$$D = I_0$$

c) Compuertas



d) Diagrama temporal

I2	0	0	0	1	1	1
I1	0	1	1	0	0	1
I0	1	0	1	0	1	0
A	0	0	0	1	1	1
B	0	0	0	0	0	1
C	0	1	1	1	1	1
D	1	0	1	0	1	0
E	0	1	1	1	1	1
F	0	0	0	0	0	1
G	0	0	0	1	1	1

e) Entradas rezagadas

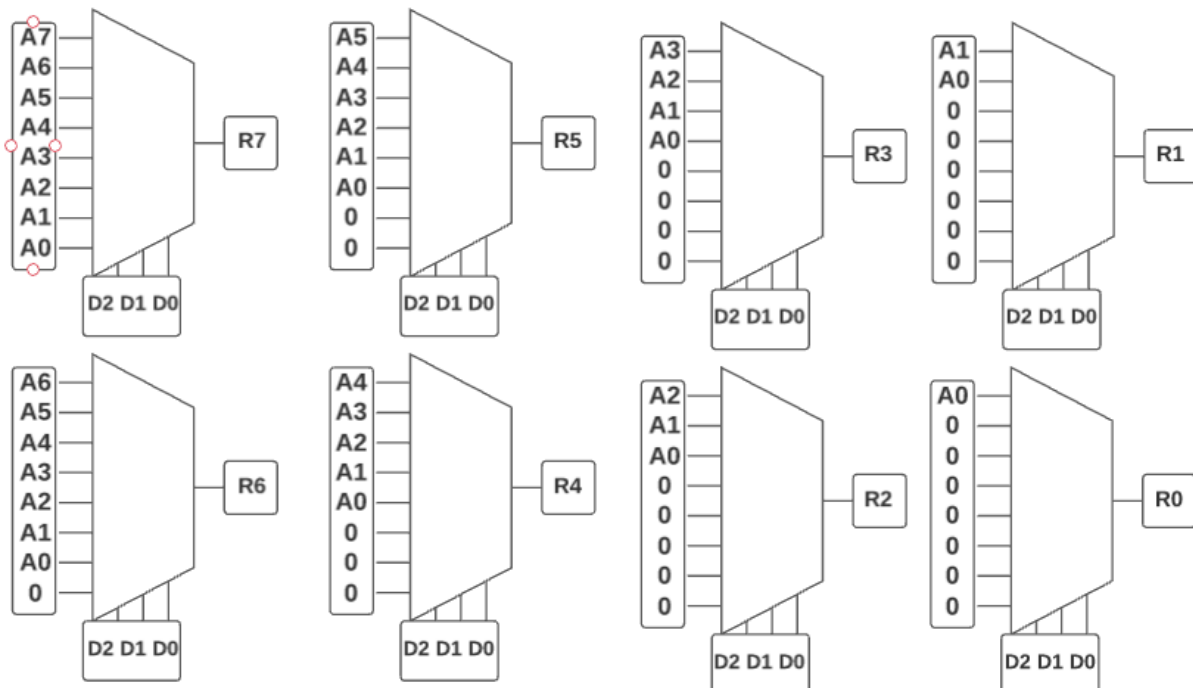
A las entradas "000" y "111" las puse como "don't care", pero viendo el diagrama que armé, deduzco que para "000" todos los Leds quedan apagados, y para "111" todos los Leds quedan encendidos.

10. Unsigned left shifter (8 bits)

a) Tabla de la verdad

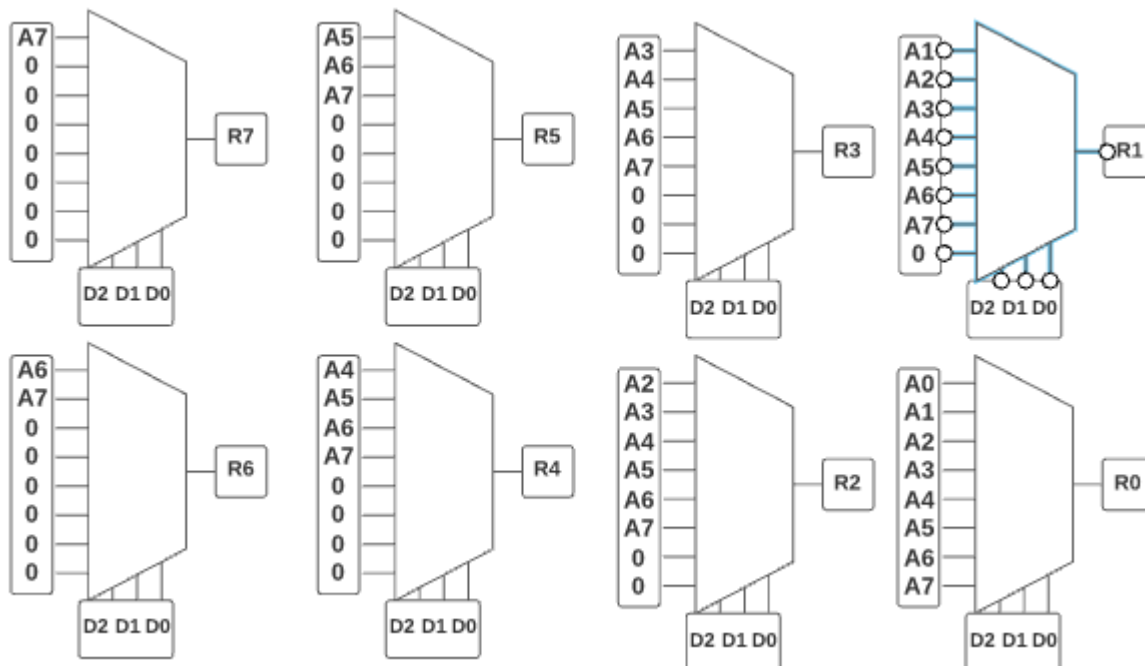
D2	D1	D0	R7	R6	R5	R4	R3	R2	R1	R0
0	0	0	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	A6	A5	A4	A3	A2	A1	A0	0
0	1	0	A5	A4	A3	A2	A1	A0	0	0
0	1	1	A4	A3	A2	A1	A0	0	0	0
1	0	0	A3	A2	A1	A0	0	0	0	0
1	0	1	A2	A1	A0	0	0	0	0	0
1	1	0	A1	A0	0	0	0	0	0	0
1	1	1	A0	0	0	0	0	0	0	0

b) Circuito con multiplexores



11. Unsigned right shifter (8 bits)

D2	D1	D0	R7	R6	R5	R4	R3	R2	R1	R0
0	0	0	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	0	A7	A6	A5	A4	A3	A2	A1
0	1	0	0	0	A7	A6	A5	A4	A3	A2
0	1	1	0	0	0	A7	A6	A5	A4	A3
1	0	0	0	0	0	0	A7	A6	A5	A4
1	0	1	0	0	0	0	0	A7	A6	A5
1	1	0	0	0	0	0	0	0	A7	A6
1	1	1	0	0	0	0	0	0	0	A7



12. Unsigned left shifter, hasta MSB = 0

a) Tabla de la verdad

A7	A6	A5	A4	A3	A2	A1	A0	R7	R6	R5	R4	R3	R2	R1	R0
0	x	x	x	x	x	x	x	A7	A6	A5	A4	A3	A2	A1	A0
1	0	x	x	x	x	x	x	A6	A5	A4	A3	A2	A1	A0	1
1	1	0	x	x	x	x	x	A5	A4	A3	A2	A1	A0	1	1
1	1	1	0	x	x	x	x	A4	A3	A2	A1	A0	1	1	1
1	1	1	1	0	x	x	x	A3	A2	A1	A0	1	1	1	1
1	1	1	1	1	0	x	x	A2	A1	A0	1	1	1	1	1
1	1	1	1	1	1	0	x	A1	A0	1	1	1	1	1	1
1	1	1	1	1	1	1	x	A0	1	1	1	1	1	1	1

Para implementarlo con multiplexores, se me ocurre hacer unas signals auxiliares que me devuelvan un número del 0 al 7 (el desplazamiento), en función de dónde esté colocado el "0".

A7	A6	A5	A4	A3	A2	A1	A0	D2	D1	D0
0	x	x	x	x	x	x	x	0	0	0
1	0	x	x	x	x	x	x	0	0	1
1	1	0	x	x	x	x	x	0	1	0
1	1	1	0	x	x	x	x	0	1	1
1	1	1	1	0	x	x	x	1	0	0
1	1	1	1	1	0	x	x	1	0	1
1	1	1	1	1	1	0	x	1	1	0
1	1	1	1	1	1	1	x	1	1	1

Escribiendo los minitérminos para C2, C1 y C0 y simplificando...

$$D_2 = A_7 \cdot A_6 \cdot A_5 \cdot A_4 \cdot \overline{A_3} + A_7 \cdot A_6 \cdot A_5 \cdot A_4 \cdot A_3 \cdot \overline{A_2} + A_7 \cdot A_6 \cdot A_5 \cdot A_4 \cdot A_3 \cdot A_2 \cdot \overline{A_1} + A_7 \cdot A_6 \cdot A_5 \cdot A_4 \cdot A_3 \cdot A_2 \cdot A_1$$

$$D_2 = A_7 \cdot A_6 \cdot A_5 \cdot A_4 \cdot (\overline{A_3} + A_3 \cdot \overline{A_2} + A_3 \cdot A_2 \cdot \overline{A_1} + A_3 \cdot A_2 \cdot A_1) \Rightarrow \text{Sea } \overline{A_3} + A_3 \cdot \overline{A_2} = \overline{A_3 \cdot A_2} \text{ (tabla abajo)}$$

$$D_2 = A_7 \cdot A_6 \cdot A_5 \cdot A_4 \cdot (\overline{A_3 \cdot A_2} + A_3 \cdot A_2 \cdot (\overline{A_1} + A_1)) \Rightarrow \text{Sea } \overline{A_1} + A_1 = 1$$

$$D_2 = A_7 \cdot A_6 \cdot A_5 \cdot A_4 \cdot (\overline{A_3 \cdot A_2} + A_3 \cdot A_2) \Rightarrow Sea \overline{A_3 \cdot A_2} + A_3 \cdot A_2 = 1$$

$$D_2 = A_7 \cdot A_6 \cdot A_5 \cdot A_4$$

A3	A2	$\sim A3 + A3 \cdot \sim A2$	$\sim (A3 \cdot A2)$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

$$D_1 = A_7 \cdot A_6 \cdot \overline{A_5} + A_7 \cdot A_6 \cdot A_5 \cdot \overline{A_4} + A_7 \cdot A_6 \cdot A_5 \cdot A_4 \cdot A_3 \cdot A_2 \cdot \overline{A_1} + A_7 \cdot A_6 \cdot A_5 \cdot A_4 \cdot A_3 \cdot A_2 \cdot A_1$$

$$D_1 = A_7 \cdot A_6 (\overline{A_5} + A_5 \cdot \overline{A_4} + A_5 \cdot A_4 \cdot A_3 \cdot A_2 \cdot (\overline{A_1} + A_1)) \Rightarrow Sea \overline{A_1} + A_1 = 1 \text{ y } \overline{A_5} + A_5 \cdot \overline{A_4} = \overline{A_5 \cdot A_4}$$

$$D_1 = A_7 \cdot A_6 \cdot (\overline{A_5 \cdot A_4} + A_5 \cdot A_4 \cdot A_3 \cdot A_2)$$

$$D_0 = A_7 \cdot \overline{A_6} + A_7 \cdot A_6 \cdot A_5 \cdot \overline{A_4} + A_7 \cdot A_6 \cdot A_5 \cdot A_4 \cdot A_3 \cdot \overline{A_2} + A_7 \cdot A_6 \cdot A_5 \cdot A_4 \cdot A_3 \cdot A_2 \cdot A_1$$

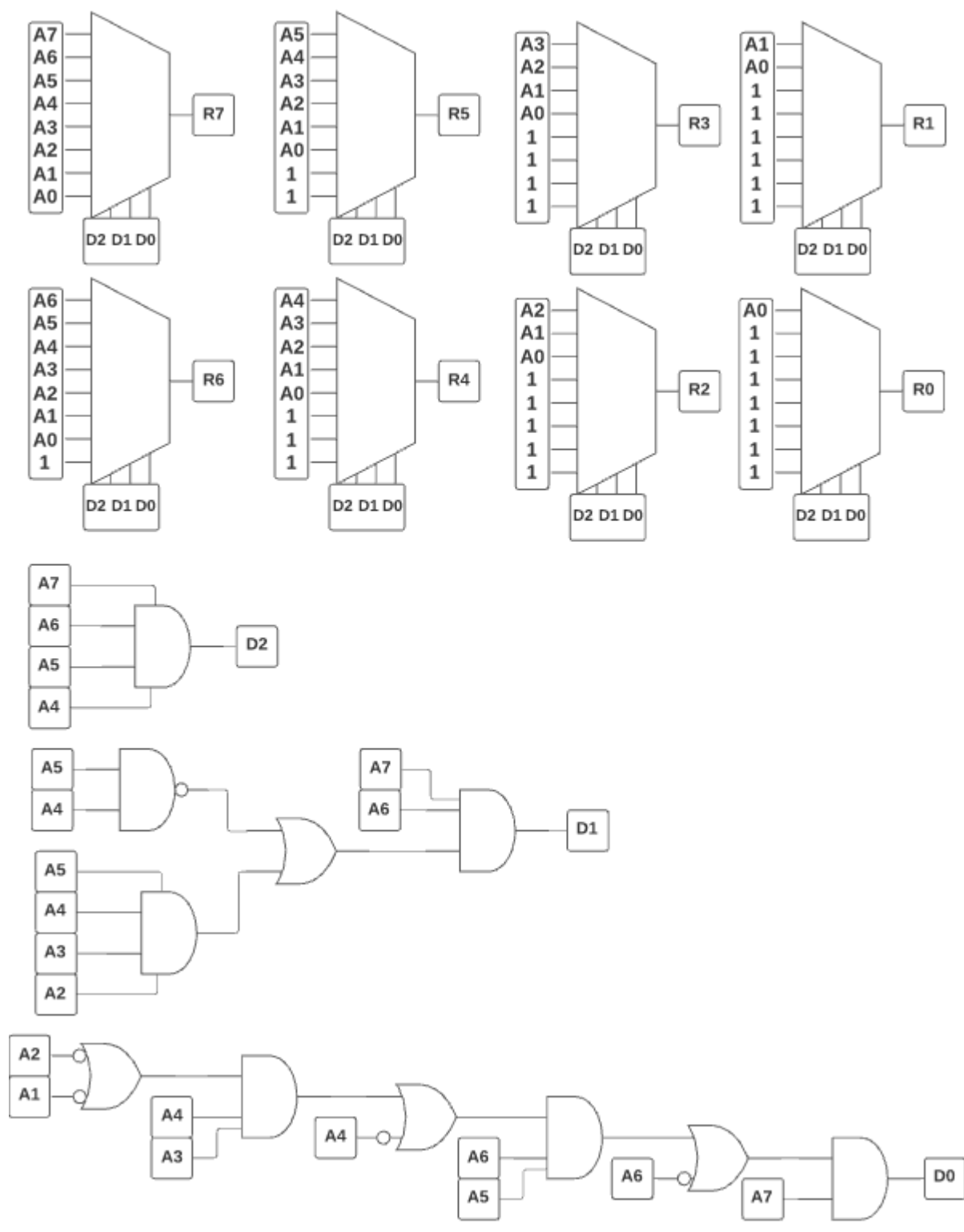
$$D_0 = A_7 \cdot (\overline{A_6} + A_6 \cdot A_5 \cdot (\overline{A_4} + A_4 \cdot A_3 \cdot (\overline{A_2} + A_2 \cdot A_1))) \Rightarrow Sea \overline{A_2} + A_2 \cdot A_1 = \overline{A_2} + \overline{A_1} \text{ (tabla abajo)}$$

$$D_0 = A_7 \cdot (\overline{A_6} + A_6 \cdot A_5 \cdot (\overline{A_4} + A_4 \cdot A_3 \cdot (\overline{A_2} + \overline{A_1})))$$

A2	A1	$\sim A2 + A2 \cdot A1$	$\sim A2 + \sim A1$
0	0	1	1
0	1	1	1
1	0	0	1
1	1	1	0

b) Gráfico con multiplexores

Entonces, para obtener las salidas de control, aplico la lógica combinacional de abajo, que me va a indicar cuántos lugares desplazarme, y luego reuso el bloque de multiplexores del "unsigned left shifter".



13. Unsigned left shifter until MSB = 1

a) Tabla de la verdad

A7	A6	A5	A4	A3	A2	A1	A0	R7	R6	R5	R4	R3	R2	R1	R0
1	x	x	x	x	x	x	x	A7	A6	A5	A4	A3	A2	A1	A0
0	1	x	x	x	x	x	x	A6	A5	A4	A3	A2	A1	A0	0
0	0	1	x	x	x	x	x	A5	A4	A3	A2	A1	A0	0	0
0	0	0	1	x	x	x	x	A4	A3	A2	A1	A0	0	0	0
0	0	0	0	1	x	x	x	A3	A2	A1	A0	0	0	0	0
0	0	0	0	0	1	x	x	A2	A1	A0	0	0	0	0	0
0	0	0	0	0	0	1	x	A1	A0	0	0	0	0	0	0
0	0	0	0	0	0	0	x	A0	0	0	0	0	0	0	0

A7	A6	A5	A4	A3	A2	A1	A0	D2	D1	D0
1	x	x	x	x	x	x	x	0	0	0
0	1	x	x	x	x	x	x	0	0	1
0	0	1	x	x	x	x	x	0	1	0
0	0	0	1	x	x	x	x	0	1	1
0	0	0	0	1	x	x	x	1	0	0
0	0	0	0	0	1	x	x	1	0	1
0	0	0	0	0	0	1	x	1	1	0
0	0	0	0	0	0	0	x	1	1	1

Construyo los minitérminos para las entradas de control y minimizo:

$$D_2 = \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot (A_3 + \overline{A_3} \cdot A_2 + \overline{A_3} \cdot \overline{A_2} \cdot A_1 + \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1})$$

$$D_2 = \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot (A_3 + \overline{A_3} \cdot A_2 + \overline{A_3} \cdot \overline{A_2} \cdot (A_1 + \overline{A_1})) \Rightarrow \text{Sea } A_1 + \overline{A_1} = 1$$

$$D_2 = \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot (A_3 + \overline{A_3} \cdot (A_2 + \overline{A_2}))$$

$$D_2 = \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot (A_3 + \overline{A_3})$$

$$D_2 = \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4}$$

$$D_1 = \overline{A_7} \cdot \overline{A_6} \cdot (A_5 + \overline{A_5} \cdot A_4) + \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot (A_1 + \overline{A_1})$$

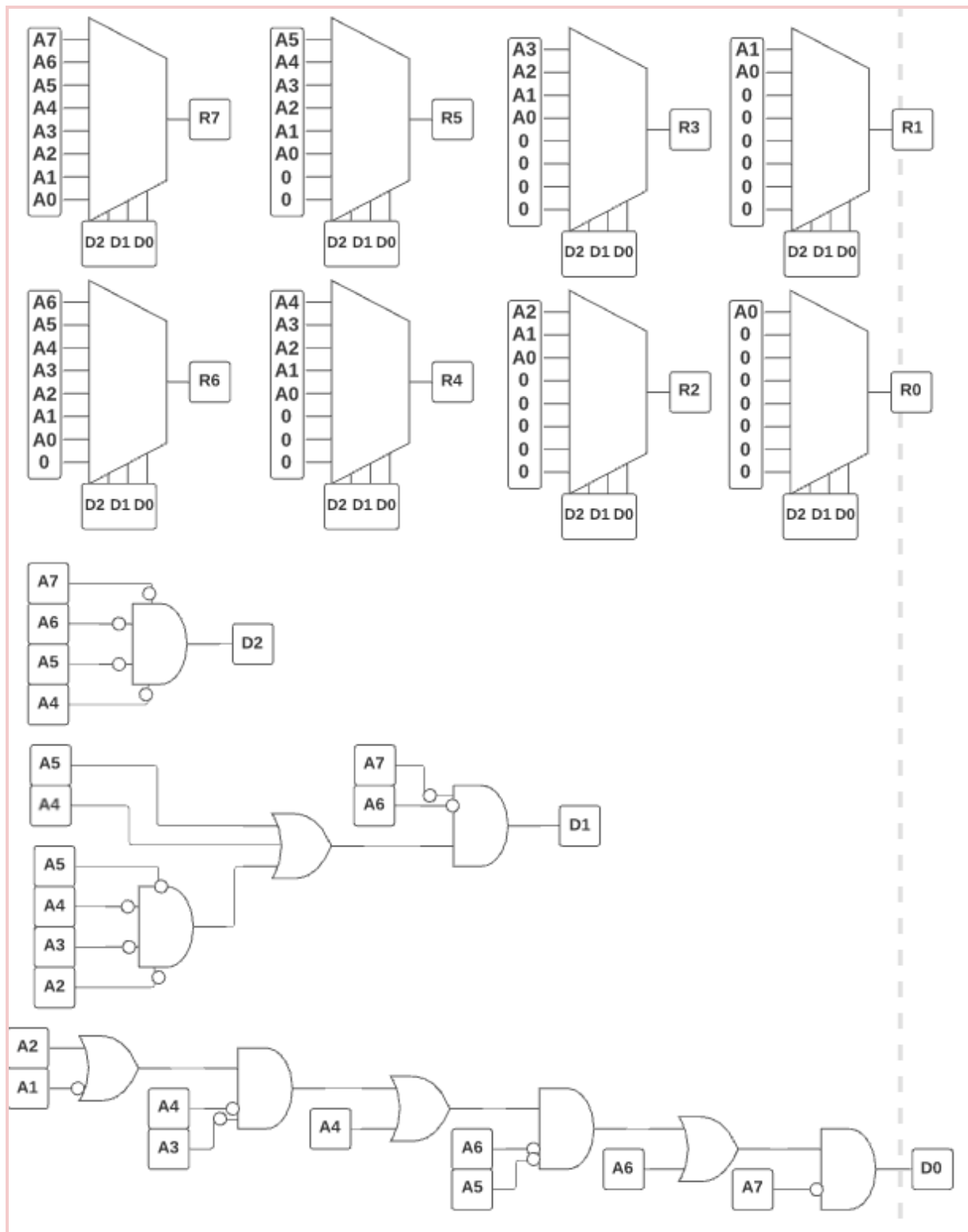
$$D_1 = \overline{A_7} \cdot \overline{A_6} \cdot (A_5 + \overline{A_5} \cdot A_4) + \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \Rightarrow Sea A_5 + \overline{A_5} \cdot A_4 = A_5 + A_4$$

$$D_1 = \overline{A_7} \cdot \overline{A_6} \cdot (A_5 + A_4 + \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2})$$

$$D_0 = \overline{A_7} \cdot A_6 + \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot A_4 + \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot A_2 + \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1}$$

$$D_0 = \overline{A_7} \cdot (A_6 + \overline{A_6} \cdot \overline{A_5} \cdot (A_4 + \overline{A_4} \cdot \overline{A_3} \cdot (A_2 + \overline{A_2} \cdot \overline{A_1}))) \Rightarrow Sea A_2 + \overline{A_2} \cdot \overline{A_1} = A_2 + \overline{A_1}$$

$$D_0 = \overline{A_7} \cdot (A_6 + \overline{A_6} \cdot \overline{A_5} \cdot (A_4 + \overline{A_4} \cdot \overline{A_3} \cdot (A_2 + \overline{A_1})))$$



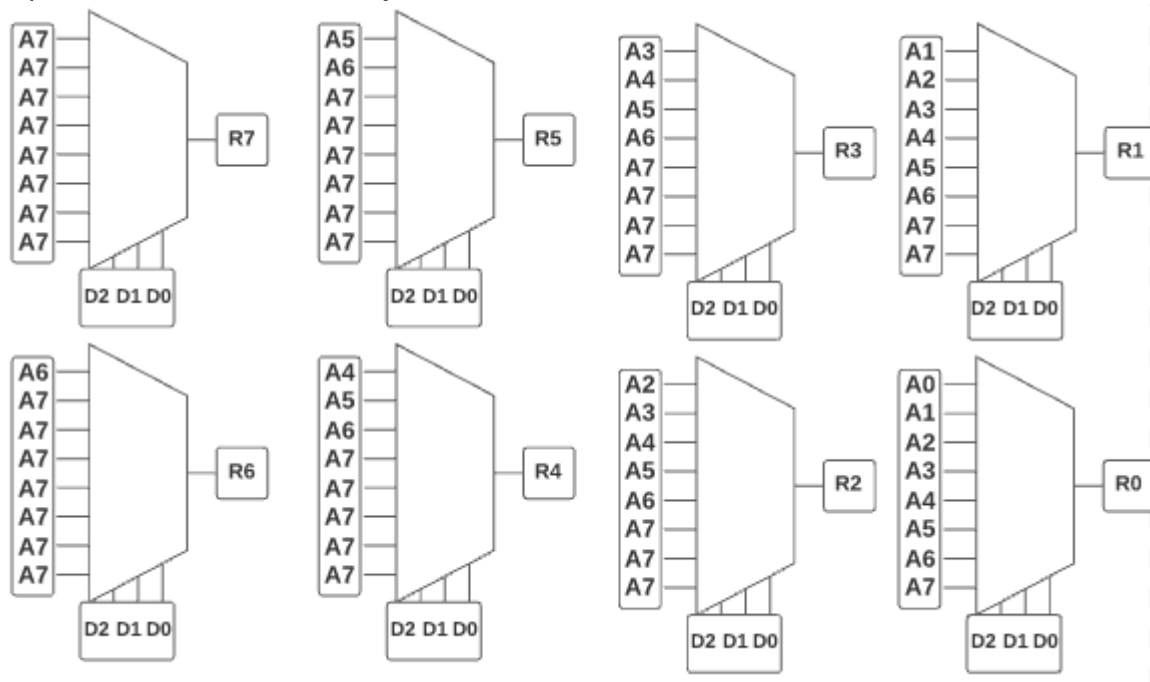
14. Signed and unsigned right shifter

a) Tabla de la verdad

En vez de completar con "1" o con "0", simplemente complemento con el MSB que guarda el signo y listo.

D2	D1	D0	R7	R6	R5	R4	R3	R2	R1	R0
0	0	0	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	A7	A7	A6	A5	A4	A3	A2	A1
0	1	0	A7	A7	A7	A6	A5	A4	A3	A2
0	1	1	A7	A7	A7	A7	A6	A5	A4	A3
1	0	0	A7	A7	A7	A7	A7	A6	A5	A4
1	0	1	A7	A7	A7	A7	A7	A7	A6	A5
1	1	0	A7	A7	A7	A7	A7	A7	A7	A6
1	1	1	A7	A7	A7	A7	A7	A7	A7	A7

b) Circuito con multiplexores



15. Saturador de 16 a 8 bits

a) Tabla de la verdad

A15	A14-A7	A6-A0	R7	R6-R0
0	0	x	A15	A6-A0
1	1	x	A15	A6-A0
0	“al menos un 1”	x	A15	“111111”
1	“al menos un 0”	x	A15	“000000”

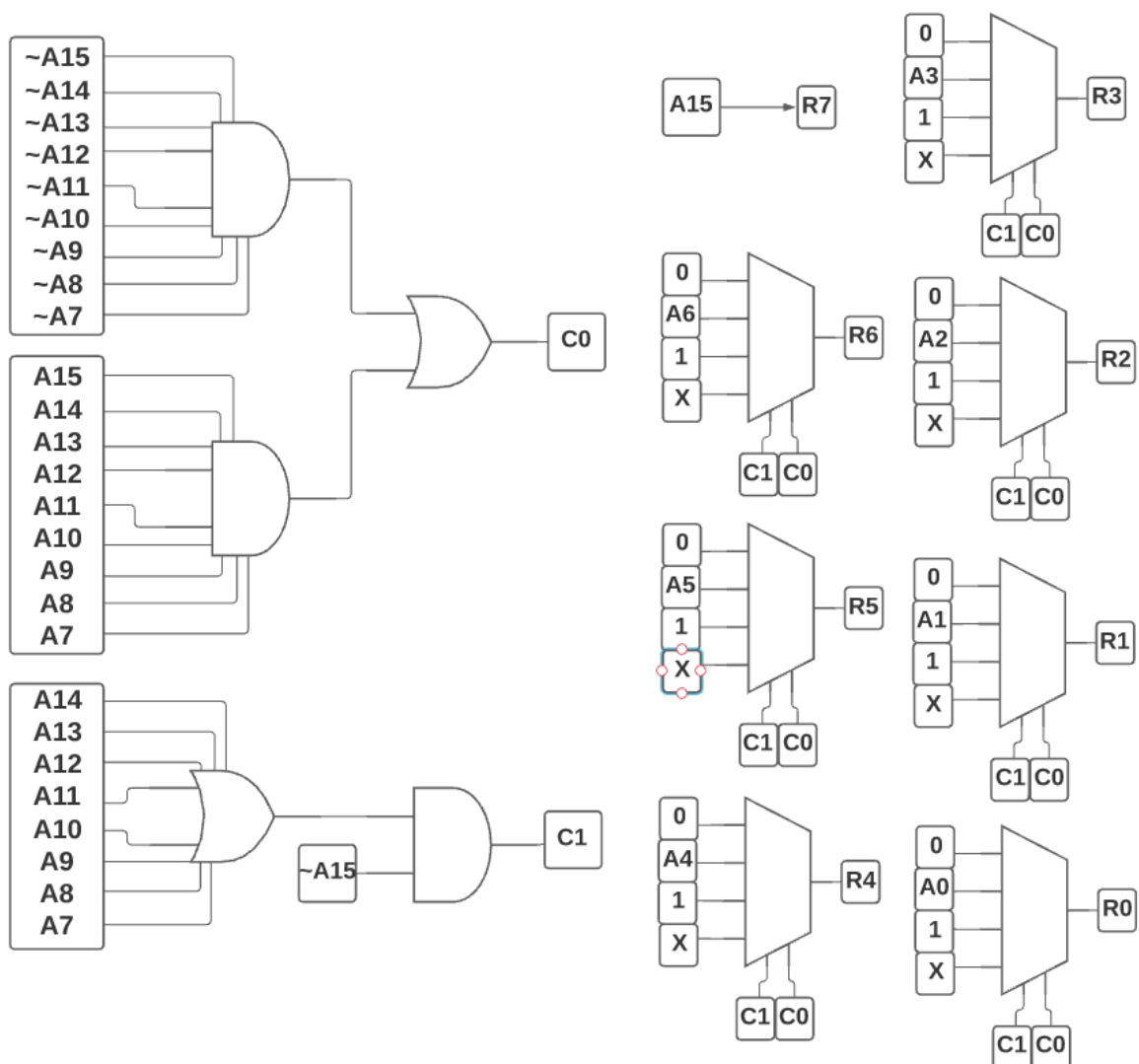
b) Circuito a nivel de compuertas

Explico un poco el gráfico. Las cuatro condiciones de la tabla de la verdad de arriba son mutuamente excluyentes y exhaustivas (ocupan todas las posibilidades de entrada). La entrada de control del multiplexor “C0” valdrá “1” cuando la primera OR la segunda condición sean verdaderas. Y la entrada de control “C1” valdrá “1” cuando se cumpla la tercera.

Al cumplirse alguna de las dos primeras, “C0 = 1”, y “C1 = 0”, porque si ya se cumplió una, no puede cumplirse ni la tercera o cuarta.

Luego, si no se cumplen las primeras dos “C0=0”, y hay dos casos: “C1=1” si se cumple la tercera, o “C1=0” si no se cumple. Si “C0=0” y “C1=0”, por descarte, se cumple la cuarta condición.

C1 (se cumple la tercera)	C0 (se cumple la primera o la segunda)	R6-R0	Se cumple la condición:
0	0	0	Cuarta
0	1	A6-A0	Primera o segunda
1	0	1	Tercera
1	1	X	Caso imposible



16. Restador del valor “2” a un número de N bits.

Analizando para números positivos y negativos qué pasa cuando les resto 2, veo que ocurre el siguiente comportamiento:

A6	A5	A4	A3	A2	A1	A0	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Overflow
0	x	x	x	x	1	x	0	x	x	x	x	0	x	0
0	x	x	x	1	0	x	0	x	x	x	0	1	x	0
0	x	x	1	0	0	x	0	x	x	0	1	1	x	0
0	x	1	0	0	0	x	0	x	0	1	1	1	x	0
0	1	0	0	0	0	x	0	0	1	1	1	1	x	0
1	x	x	x	x	1	x	1	x	x	x	x	0	x	0
1	x	x	x	1	0	x	1	x	x	x	0	1	x	0
1	x	x	1	0	0	x	1	x	x	0	1	1	x	0
1	x	1	0	0	0	x	1	x	0	1	1	1	x	0
1	1	0	0	0	0	x	1	0	1	1	1	1	x	0
1	0	0	0	0	0	x	1	0	0	0	0	0	0	1

Como se ve, para números negativos y positivos se cumple la misma “forma” de cambiar los bits, así que puedo juntarlos en (incluyendo los casos límites del “1” y el “0”.

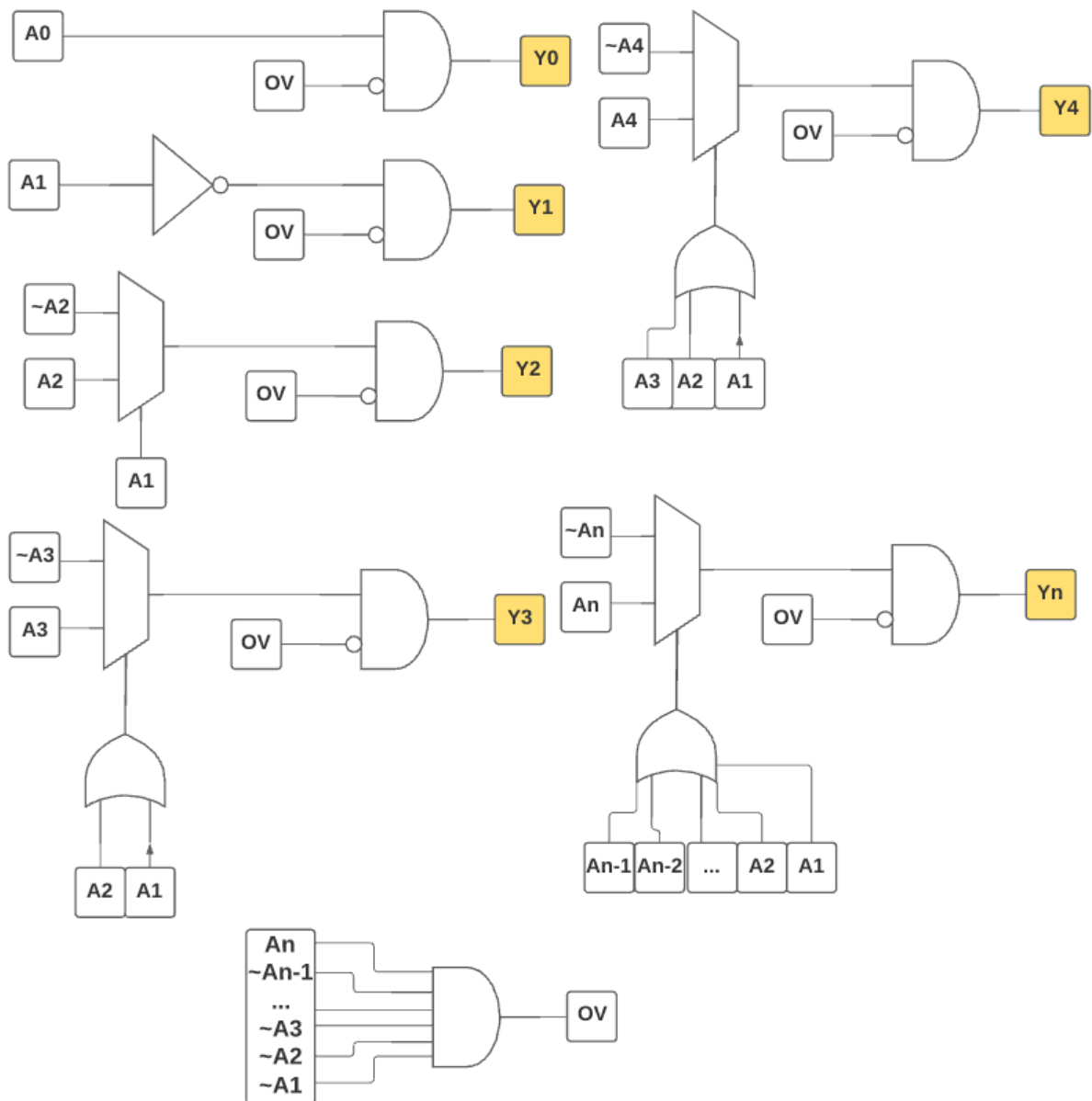
A6	A5	A4	A3	A2	A1	A0	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Overflow
x	x	x	x	x	1	x	A6	A5	A4	A3	A2	0	A0	0
x	x	x	x	1	0	x	A6	A4	A4	A3	0	1	A0	0
x	x	x	1	0	0	x	A6	A5	A4	0	1	1	A0	0
x	x	1	0	0	0	x	A6	A5	0	1	1	1	A0	0
x	1	0	0	0	0	x	A6	0	1	1	1	1	A0	0
1	0	0	0	0	0	x	1	0	0	0	0	0	0	1
0	0	0	0	0	0	x	1	1	1	1	1	1	A0	0

Entonces este circuito es como un inversor, que invierte todos los bits desde A1 hasta el próximo “1” inclusive.

Hacer el circuito no es tan evidente. En estos casos, conviene escribirlo en lenguaje “llano”, resaltando las palabras claves que resultan en compuertas:

- $Y_0 = A_0$ **SI** \sim Overflow, "0" **SI** Overflow
- $Y_1 = \sim A_1$ **SI** \sim Overflow, "0" **SI** Overflow.
- $Y_2 = A_2$ **SI** A_1 , $\sim A_2$ **si** $\sim A_1$, "0" **si** Overflow
- $Y_3 = A_3$ **SI** $(A_2 \text{ o } A_1)$, $\sim A_3$ **si** $(\sim A_2 \text{ y } \sim A_1)$, "0" **si** Overflow
- $Y_4 = A_4$ **si** $(A_3 \text{ o } A_2 \text{ o } A_1)$, $\sim A_4$ **si** $(\sim A_3 \text{ y } \sim A_2 \text{ y } \sim A_1)$, "0" **si** Overflow
- $Y_n = A_n$ **si** $(A_{n-1} \text{ o } A_{n-2} \text{ o } \dots \text{ o } A_2 \text{ o } A_1)$, $\sim A_n$ **si** $(\sim A_{n-1} \text{ y } \sim A_{n-2} \text{ y } \dots \text{ y } \sim A_2 \text{ y } \sim A_1)$, "1" **si** Overflow

La utilización de condicionales (la palabra "**si**"), indica la presencia de un multiplexor. La utilización de "**o**" o "**y**" de compuertas **OR** o **AND**.



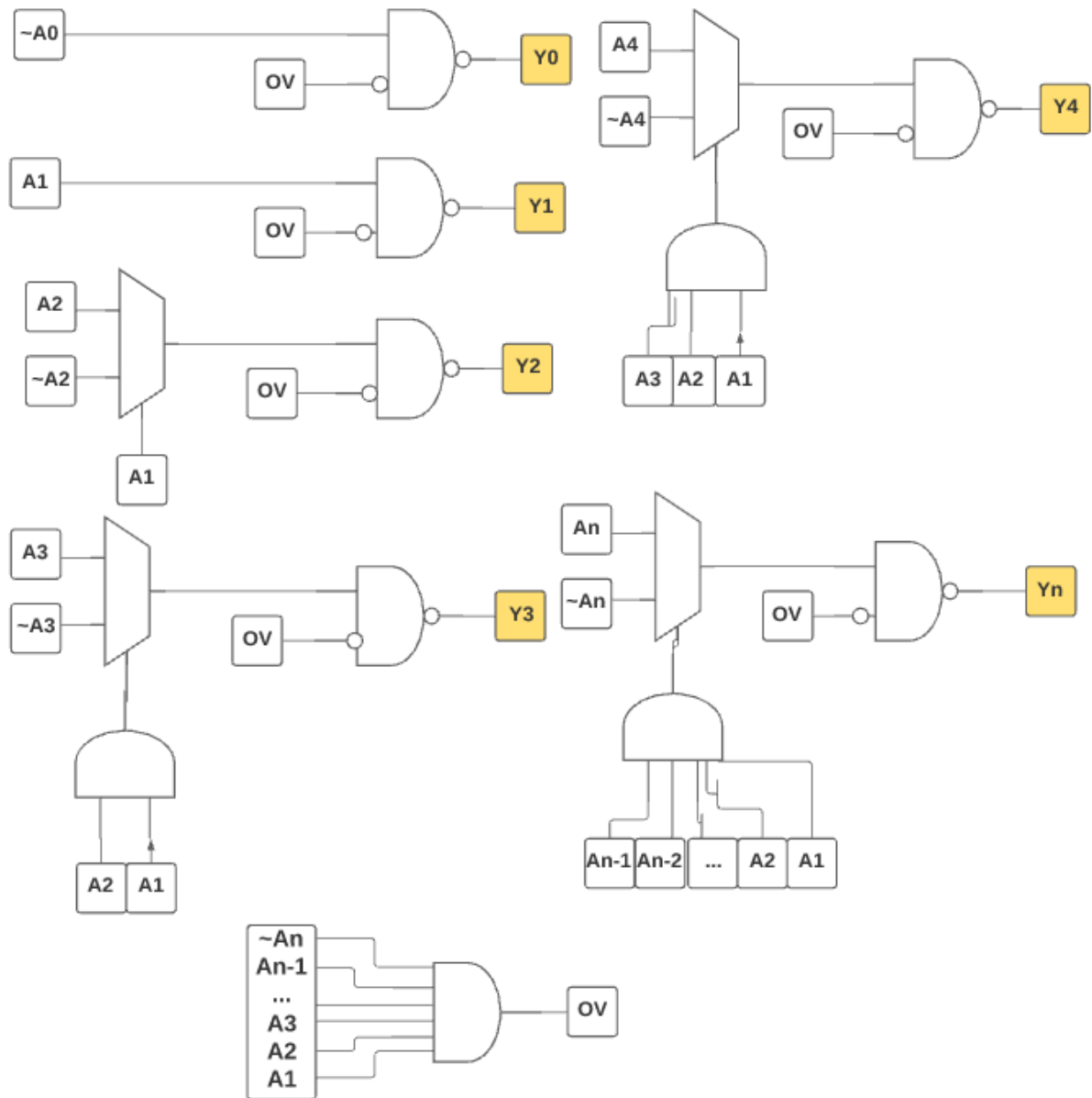
17. Sumador del valor “2” a un número de N bits.

Al sumarle dos, la lógica es parecida al ej. anterior, solo que cambian de lugar los “1” y los “0”.

A6	A5	A4	A3	A2	A1	A0	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Overflow
x	x	x	x	x	0	x	A6	A5	A4	A3	A2	1	A0	0
x	x	x	x	0	1	x	A6	A5	A4	A3	1	0	A0	0
x	x	x	0	1	1	x	A6	A5	A4	1	0	0	A0	0
x	x	0	1	1	1	x	A6	A5	1	0	0	0	A0	0
x	0	1	1	1	1	x	A6	1	0	0	0	0	A0	0
0	1	1	1	1	1	x	0	1	1	1	1	1	1	1
1	1	1	1	1	1	x	0	0	0	0	0	0	A0	0

En lenguaje llano es:

- $Y0 = A0$ si $\sim Ov$, else 1.
- $Y1 = \sim A1$ si $\sim Ov$, else 1
- $Y2 = A2$ si $\sim A1$, $\sim A2$ si $A1$, 1 si Ov .
- $Y3 = A3$ si $(\sim A1 \text{ o } \sim A2)$, $\sim A3$ si $(A2 \text{ y } A1)$, 1 si Ov
- $Y4 = A4$ si $(\sim A1 \text{ o } \sim A2 \text{ o } \sim A3)$, $\sim A4$ si $(A3 \text{ y } A2 \text{ y } A1)$, 1 si Ov .



18. Contador de “1” en número de 5 bits

a) Cantidad de salidas

El número más alto posible es 5 = “101”, por lo tanto la salida es de 3 bits.

b) Tabla de la verdad

E	D	C	B	A	Y2	Y1	Y0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	1	0	1	0
0	0	1	0	0	0	0	1
0	0	1	0	1	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
0	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	0	1	1	0	1	1
0	1	1	0	0	0	1	0
0	1	1	0	1	0	1	1
0	1	1	1	0	0	1	1
0	1	1	1	1	1	0	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	0	1	0	0	1	0
1	0	0	1	1	0	1	1
1	0	1	0	0	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	0	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	0	0	1	0
1	1	0	0	1	0	1	1
1	1	0	1	0	0	1	1
1	1	0	1	1	1	0	0
1	1	1	0	0	0	1	1
1	1	1	0	1	1	0	0
1	1	1	1	0	1	0	0
1	1	1	1	1	1	0	1

c) Minimización

		Y2								
		A=0				CB	A=1			
		00	01	11	10		00	01	11	10
ED	00	0	0	0	0		0	0	0	0
	01	0	0	0	0		0	0	1	0
	11	0	0	1	0		0	1	1	1
	10	0	0	0	0		0	0	1	0

Y2 = rojo + azul + celeste + rosa + verde

$$Y_2 = E.D.C.B + E.D.B.A + D.C.B.A + E.D.C.A + E.C.B.A$$

		Y1								
		A=0				CB	A=1			
		00	01	11	10		00	01	11	10
ED	00	0	0	1	0		0	1	1	1
	01	0	1	1	1		1	1	0	1
	11	1	1	0	1		1	0	0	0
	10	0	1	1	1		1	1	0	1

Y1 = Naranja + Amarillo + Rojo + Verde brillante + celeste + azul +
+ verde oscuro + naranja claro + magenta + rosa

$$Y_1 = E.\bar{D}.\bar{C}.B + E.\bar{D}.C.\bar{B} + E.\bar{C}.\bar{B}.A + \bar{E}.D.\bar{C}.A + \bar{E}.\bar{D}.B.A + \bar{E}.C.B.A +$$

$$+ E.D.\bar{B}.\bar{A} + D.\bar{C}.B.\bar{A} + \bar{E}.D.C.\bar{A} + \bar{D}.C.B.\bar{A}$$

		Y0									
		A=0				CB	A=1				
		00	01	11	10		00	01	11	10	
ED	00	0	1	0	1		1	0	1	0	
	01	1	0	1	0		0	1	0	1	
	11	0	1	0	1		1	0	1	0	
	10	1	0	1	0		0	1	0	1	

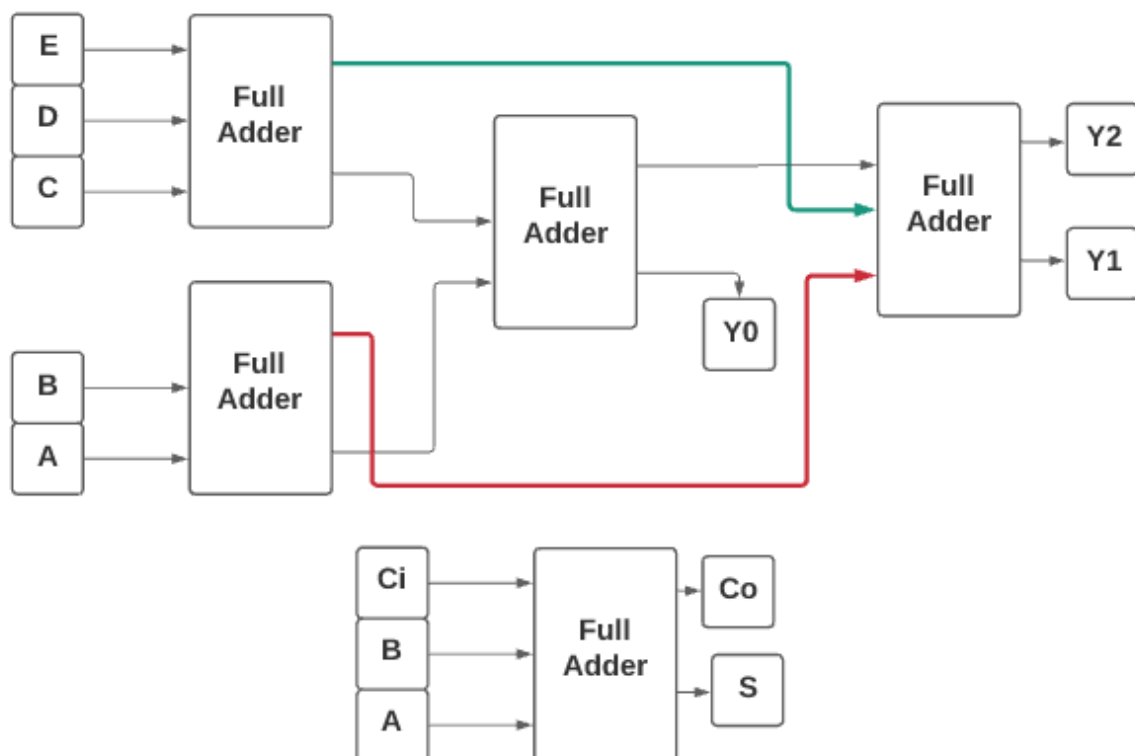
$$Y_0 = (\bar{E}.\bar{D}.\bar{C}.B + \bar{E}.\bar{D}.C.\bar{B} + \bar{E}.D.\bar{C}.\bar{B} + \bar{E}.D.C.B +$$

$$\begin{aligned}
& + E.D.\bar{C}.B + E.D.C.\bar{B} + E.\bar{D}.\bar{C}.\bar{B} + E.\bar{D}.C.B). \bar{A} + \\
& + (\bar{E}.\bar{D}.\bar{C}.\bar{B} + \bar{E}.\bar{D}.C.B + \bar{E}.D.\bar{C}.B + \bar{E}.D.C.\bar{B} + \\
& + E.D.\bar{C}.\bar{B} + E.D.C.B + E.\bar{D}.\bar{C}.B + E.\bar{D}.C.\bar{B}). A
\end{aligned}$$

d) Implementación con sumadores

Abajo se muestra un template del sumador, no forma parte del circuito.

Primero sumo los 5 números de un bit. Aprovecho y pongo la entrada “E” como “carriage input” en uno de los sumadores, para poder sumar 3 bits de una. Esto me devuelve 2 números de 2 bits, donde el valor de la suma son los LSB, y el “carriage out” los MSB. Luego, estos dos números los sumo entre sí, obteniendo el número de 3 bits buscado.



19. Conversor °Celsius (-273 ; 1000) a Kelvin

a) Cantidad de salidas y entradas

El número más alto de entrada es 1000 = "011_1110_1000", contando el bit de signo son 11 bits.

Luego, el rango de salida es: (0; 1273). A pesar de ser solo números positivos, nos piden que esté en CA2. El número más alto es: 1273="0100_1111_1001" que tiene 12 bits.

b) Tabla de la verdad (incompleto)

El número 273="01_0001_0001", podemos usar una lógica parecida al ejercicio 17 para hacer la tabla de la verdad, donde si en el lugar a sumar hay un "0", ese valor se invierte, y si hay un "1", ese valor se convierte en "0"; repitiendo el proceso hasta que se encuentre con el próximo "0" del número.

Podemos plantear primero una versión reducida de la tabla:

A3	A2	A1	A0	Y4	Y3	Y2	Y1	Y0
x	x	x	0	x	A3	A2	A1	1
x	x	0	1	x	A3	A2	1	0
x	0	1	1	x	A3	1	0	0
0	1	1	1	x	1	0	0	0
1	1	1	1	x+1	0	0	0	0

Y ahora seguir con A4, teniendo la posibilidad que tenga que sumarle "1" a Y4 o no:

A7	A6	A5	A4	Y8	Y7	Y6	Y5	Y4
x	x	x	0	x	A7	A6	A5	1
x	x	0	1	x	A7	A6	1	0
x	0	1	1	x	A7	1	0	0
0	1	1	1	x	1	0	0	0
1	1	1	1	x+1	0	0	0	0

A7	A6	A5	A4 +1	Y8 Y7	Y6	Y5	Y4	
x	x	x	0	A4	0	x	x	1
x	x	0	1	A4	0	x	x	1
x	0	1	1	A4	0	x	x	1
0	1	1	1	A4	0	x	x	1
1	1	1	1	A4+	x	x	0	x

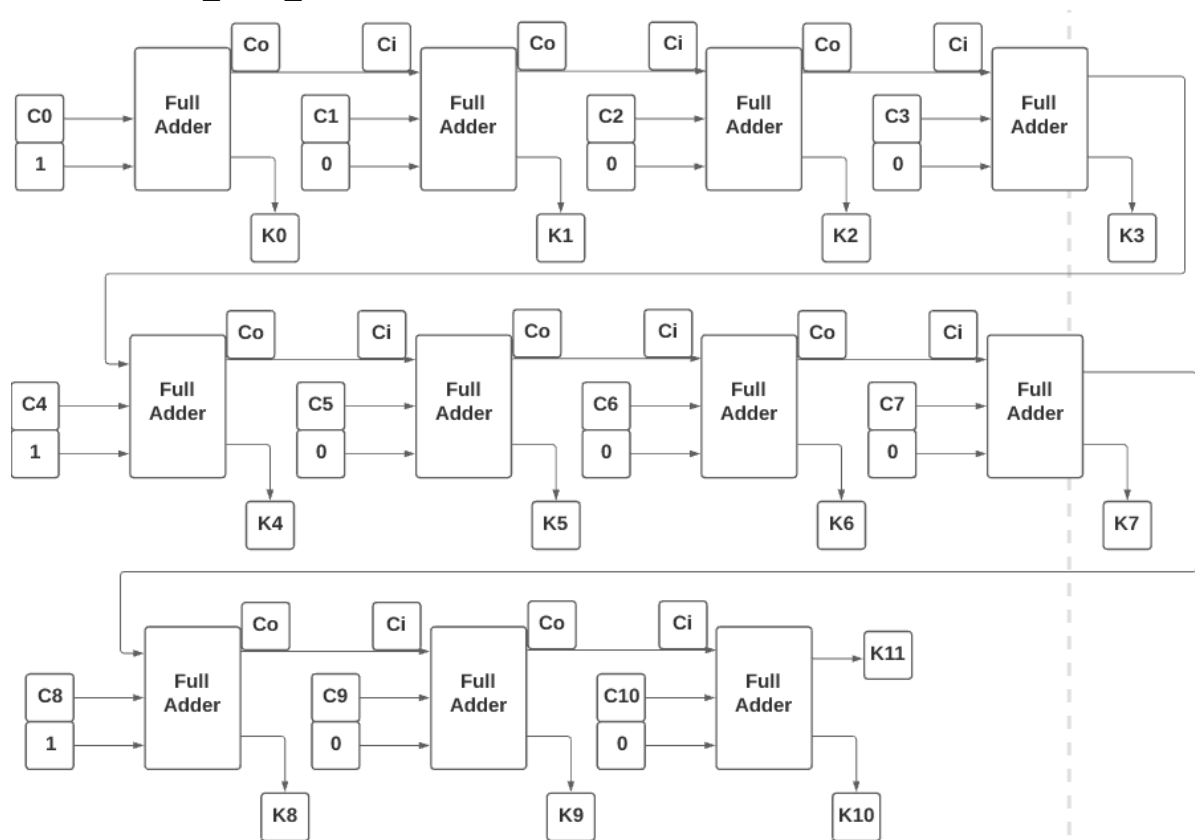
				1				
--	--	--	--	---	--	--	--	--

A	A	A	A	A	A	A	A	A	A	A	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
1	9	8	7	6	5	4	3	2	1	0	1	1	9	8	7	6	5	4	3	2	1	0
0											1	0										
x	x	0	x	x	x	0	x	x	x	0	0	x	x	1	x	x	x	1	x	x	x	1
x	x	0	x	x	x	0	x	x	0	1	0	x	x	1	x	x	x	1	x	x	1	0
x	x	0	x	x	x	0	x	0	1	1	0	x	x	1	x	x	x	1	x	1	0	0
x	x	0	x	x	x	0	0	1	1	1	0	x	x	1	x	x	x	1	1	0	0	0
x	x	0	x	x		0	1	1	1	1	x	x	0	x	x	x	x	1	0	0	0	0
				0	1	1	1	1	1	x	0	1	1	1	1	1	1					
				1	1	1	1	1	1	x	0	0	0	0	0	0	x					

c) Circuito a nivel de compuertas (incompleto)

d) Circuito con full adder

Para hacer el circuito con sumadores, solo tengo que sumar la entrada C (11 bits) con el valor 273="001_0001_0001"



20. Promedio de 4 números de 4 bits CA2

a) Cantidad de salidas

El peor caso sería que los 4 números sean $-8 = "1000"$, en tal caso el resultado sería $-8 * 4 / 4 = -8 = "1000"$, por lo que la cantidad de salidas es 4 bits.

b) Tabla de la verdad

Implementar una tabla de la verdad completa para las 16 entradas no lo veo viable. Busqué alguna manera de simplificarla pero tampoco. Lo que puedo plantear es un shift derecho de 2 lugares a la suma de los 4 números. (La suma ocupa 6 bits como máximo, el rango es $(-32, 28)$, y el peor caso es $-32 = "10_0000"$).

S5	S4	S3	S2	S1	S0	Y3	Y2	Y1	Y0
x	x	x	x	x	x	S5	S4	S3	S2

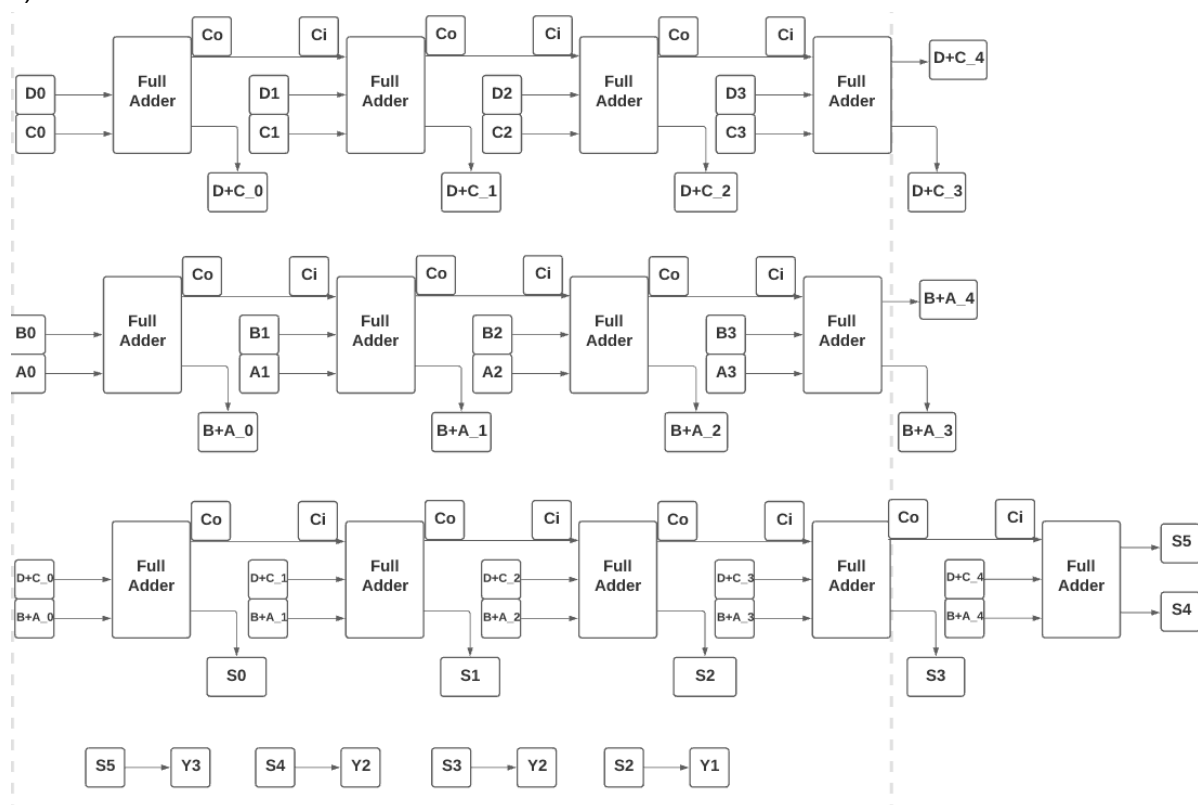
c) Implementación con full adder

En la primera rama sumo los números "D" y "C" de 4 bits. El resultado puede ser de hasta 5 bits.

En la segunda rama sumo los números "B" y "A" de 4 bits. El resultado puede ser de hasta 5 bits.

En la tercera rama, sumo las sumas anteriores, ambas de hasta 5 bits, obteniendo un número de hasta 6 bits.

Finalmente, la salida será el resultado de esa suma desplazado dos a la derecha (dividido 4).



Extra 1. Sumador

Ci	B	A	Co	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

		Co			
		BA			
		00	01	11	10
Ci	0	0	0	0	1
	1	0	1	1	1

$$C_0 = C_i \cdot A + C_i \cdot B + B \cdot A$$

		S			
		BA			
		00	01	11	10
Ci	0	0	0	1	0
	1	1	0	1	0

$$S = C_i \cdot \bar{B} \cdot \bar{A} + \bar{C}_i \cdot \bar{B} \cdot A + C_i \cdot B \cdot A + \bar{C}_i \cdot B \cdot A$$

Así, el circuito queda:

