

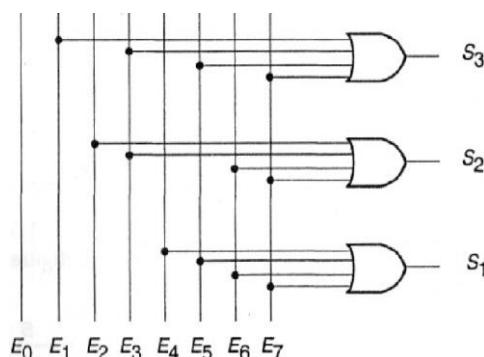
I. Circuits de codage**Exercice1**

	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	S_1	S_2	S_3
0	1								0	0	0
1		1							0	0	1
2			1						0	1	0
3				1					0	1	1
4					1				1	0	0
5						1			1	0	1
6							1		1	1	0
7								1	1	1	1

$$S_1 = E_4 + E_5 + E_6 + E_7$$

$$S_2 = E_2 + E_3 + E_6 + E_7$$

$$S_3 = E_1 + E_3 + E_5 + E_7$$



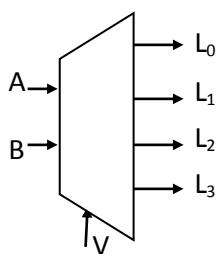
Remarque : On peut travailler en logique négative et utiliser des portes NAND

Exercice2

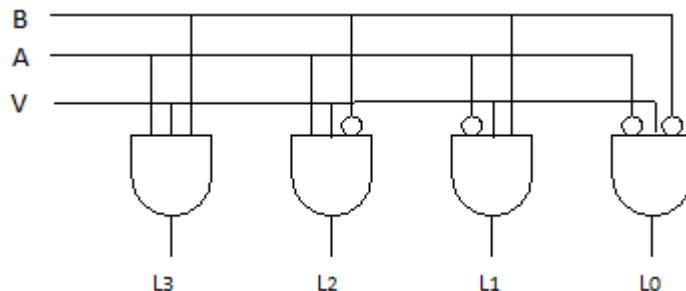
- Faire la synthèse d'un décodeur à 2 entrées, pourvu d'une entrée de validation V autorisant le fonctionnement.

$V = 0$						
A	B		L_0	L_1	L_2	L_3
0	0		0	0	0	0
0	1		0	0	0	0
1	0		0	0	0	0
1	1		0	0	0	0

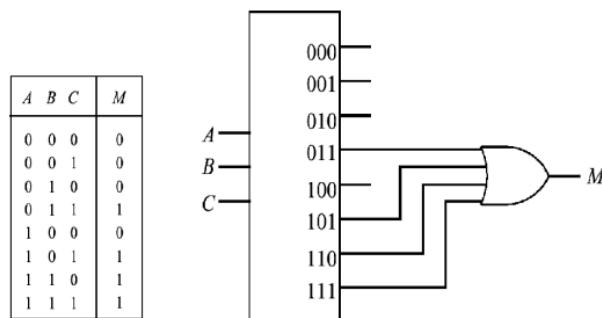
$V = 1$						
A	B		L_0	L_1	L_2	L_3
0	0		1	0	0	0
0	1		0	1	0	0
1	0		0	0	1	0
1	1		0	0	0	1



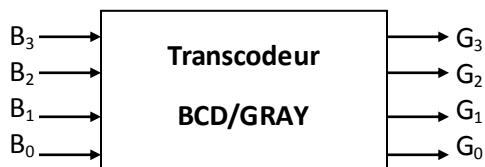
$$L_0 = \bar{A} \cdot \bar{B} \cdot V \quad L_1 = \bar{A} \cdot B \cdot V \quad L_2 = A \cdot \bar{B} \cdot V \quad L_3 = A \cdot B \cdot V$$



- Réaliser la fonction majorité à 3 variables à l'aide d'un décodeur à 3 entrées.

**Exercice3**

Faire la synthèse d'un transcodeur BCD / Gray



Chiffre Converti	Entrée BCD				Sorties Gray			
	B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1

B ₃ B ₂ B ₁ B ₀	00	01	11	10	B ₃ B ₂ B ₁ B ₀	00	01	11	10
00	0	0	X	1	00	0	1	X	1
01	0	0	X	1	01	0	1	X	1
11	0	0	X	X	11	0	1	X	X
10	0	0	X	X	10	0	1	X	X

B ₃ B ₂ B ₁ B ₀	00	01	11	10
00	0	1	X	0
01	0	1	X	0
11	1	0	X	X
10	1	0	X	X

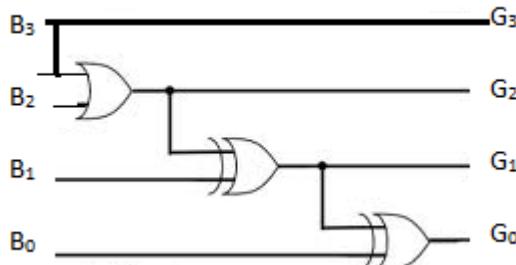
B ₃ B ₂ B ₁ B ₀	00	01	11	10
00	0	0	X	0
01	1	1	X	1
11	0	0	X	X
10	1	1	X	X

$$G_3 = B_3$$

$$G_2 = B_3 + B_2$$

$$G_1 = \overline{B_2}B_1 + B_2\overline{B_1} = B_2 \oplus B_1$$

$$G_0 = \overline{B_1}B_0 + B_1\overline{B_0} = B_1 \oplus B_0$$



Remarque : Pour le transcodeur Binaire /gray : $G_2 = B_3 \oplus B_2$

Exercice4

- a. table de vérité du transcodeur **BCD/ 7 segments**.

E_3	E_3	E_3	E_3		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		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

E_3E_2	00	01	11	10	
E_1E_0	00	1	0	X	1
	01	0	1	X	1
	11	1	1	X	X
	10	1	1	X	X

sorties **a**

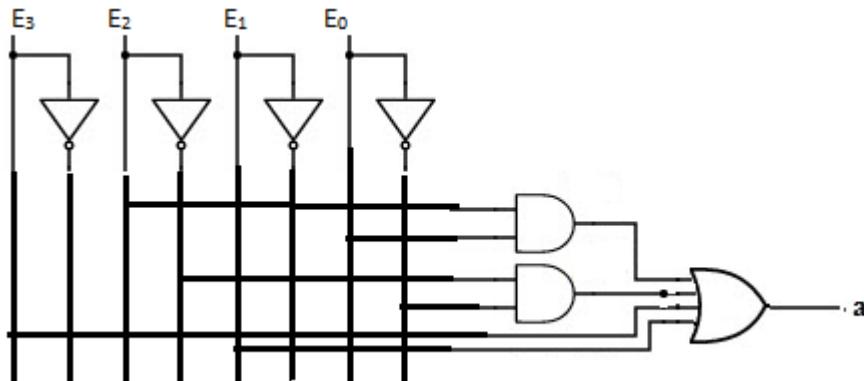
E_3E_2	00	01	11	10	
E_1E_0	00	1	0	X	1
	01	0	1	X	1
	11	1	0	X	X
	10	1	1	X	X

b. expressions simplifiées des et **d**.

$$a = E_3 + E_1 + E_2E_0 + \overline{E_2} \cdot \overline{E_0}$$

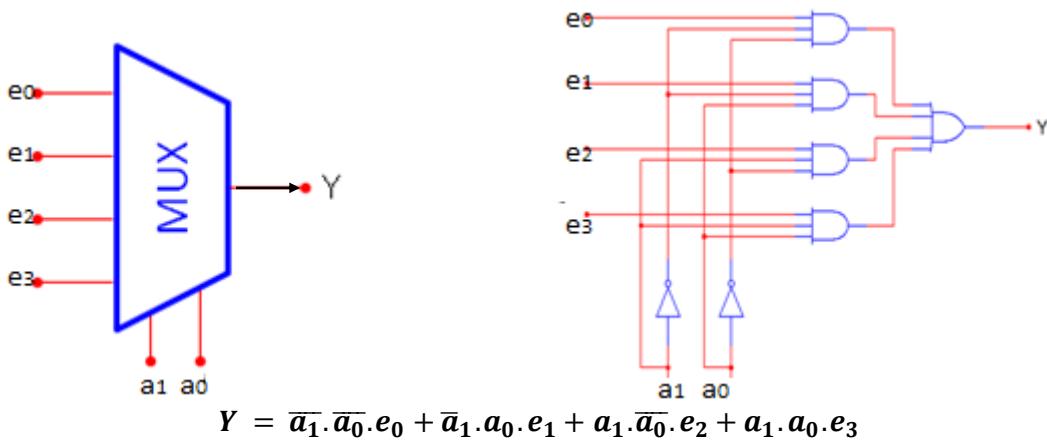
$$d = E_3 + \overline{E_2}E_1 + E_1\overline{E_0} + \overline{E_2} \cdot \overline{E_0} + E_2\overline{E_1}E_0$$

- c. Proposer un logigramme pour la sortie a avec un minimum de portes logiques.

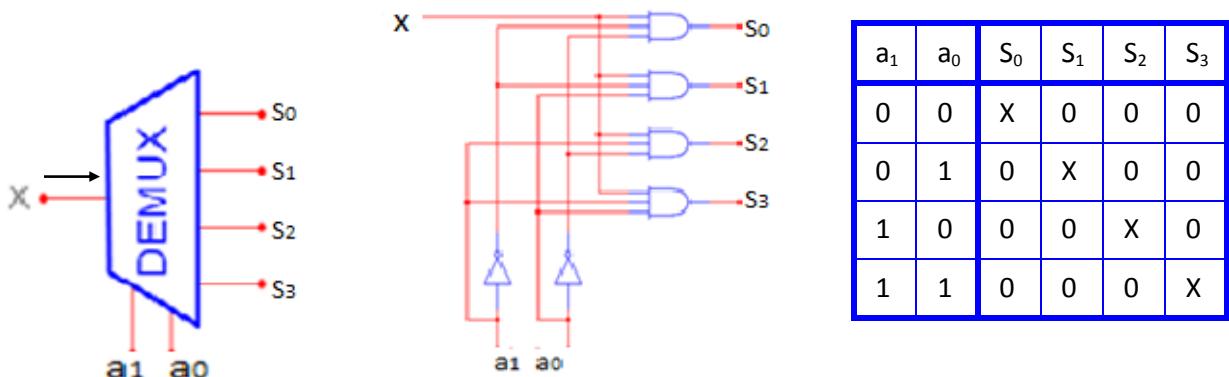


II. Multiplexeurs – Démultiplexeurs

- a. logigramme d'un multiplexeur à 2 entrées de commande (adresses).



- b. Réaliser le logigramme d'un démultiplexeur à 2 entrées de commande.

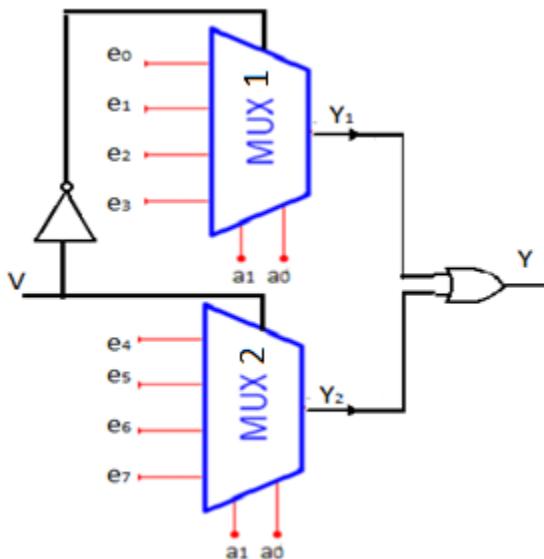


- c. Réaliser un multiplexeur à 3 entrées de commande à l'aide de 2 multiplexeurs à 2 entrées d'adresse.

On utilise l'entrée de validation V (Strobe) comme troisième adresse a_2 :

Quand $V = 0$ le MUX1 est activé (MUX2 est désactivé) et $Y = Y_1$

Quand $V = 1$ le MUX2 est activé (MUX1 est désactivé) et $Y = Y_2$



$a_2=V$	a_1	a_0	Y
0	0	0	e_0
0	0	1	e_1
0	1	0	e_2
0	1	1	e_3
1	0	0	e_4
1	0	1	e_5
1	1	0	e_6
1	1	1	e_7

- d. Soit le schéma ci-contre : Donnez l'équation de S en fonction de A, B, C, D, et E .

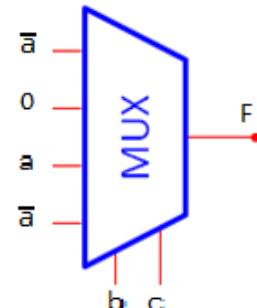
$$S_1 = \bar{A} \cdot B + A \bar{B} = A \oplus B$$

$$S_2 = C \bar{D}$$

$$S = \overline{S_2} S_1 \cdot E + S_2 \bar{S}_1 \bar{E} = S_1 (S_2 \oplus E) = A \oplus B (C \bar{D} \oplus E)$$

- e. A l'aide d'un multiplexeur à 2 entrées d'adresses, réaliser la fonction : $\bar{a}\bar{b}\bar{c} + ab\bar{c} + \bar{a}bc$

On peut prendre b et c comme entrées d'adresse et on obtient

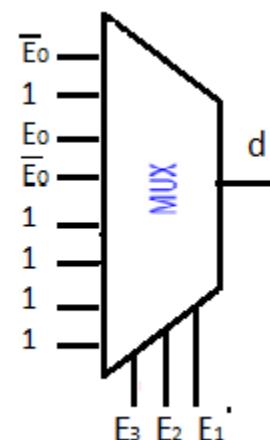


- f. A l'aide d'un multiplexeur à 3 entrées d'adresses, réaliser la fonction d de l'exercice précédent (transcodeur $BCD/7$ segments)

E_3E_2	00	01	11	10	
E_1E_0	00	1	0	X	1
01	0	1	X	1	
11	1	0	X	X	
10	1	1	X	X	

$$d = E_3 + \bar{E}_2 E_1 + E_2 \bar{E}_1 E_0 + E_1 \bar{E}_0 + \bar{E}_2 \cdot \bar{E}_0$$

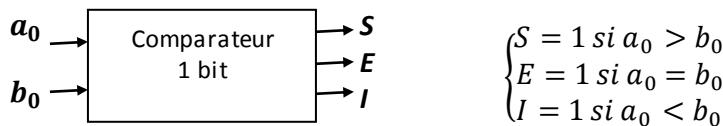
On peut prendre $E_3E_2E_1$ comme entrées d'adresse, et on obtient



III. Comparateurs - additionneurs

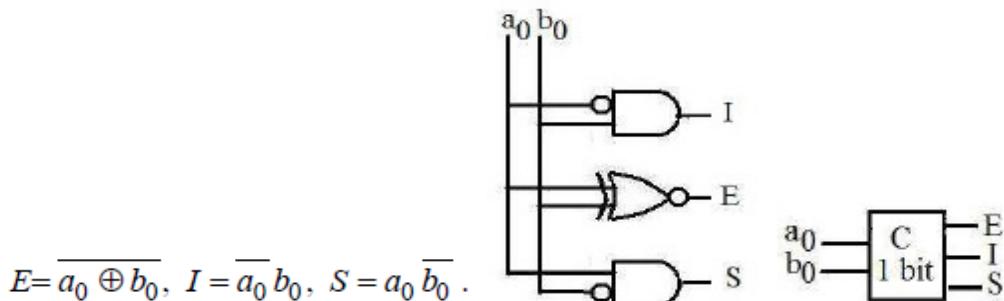
Exercice1

La figure suivante représente un comparateur de deux nombres binaires a_i et b_i à 1 bit

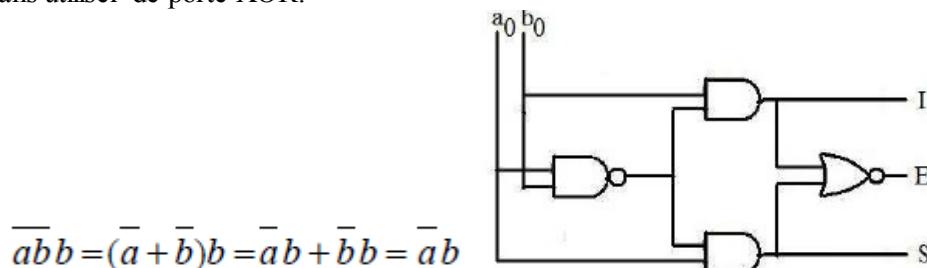


- a. Effectuer la synthèse de ce circuit logique.

On trouve facilement les sorties en fonction des deux bits d'entrée, soit :



- b. Vérifier que $a_i \oplus b_i = \bar{a}_i b_i + a_i \bar{b}_i$ et $\bar{a}_i \bar{b}_i b_i = \bar{a}_i b_i$. En déduire le schéma du circuit sans utiliser de porte XOR.



- c. On prend deux nombres de deux bits chacun en entrée, soit $a_1 a_0$ et $b_1 b_0$. Donner les équations des trois sorties de ce comparateur 2 bits.

$a_1 a_0 = b_1 b_0$ signifie : $a_1 = b_1$ et : $a_0 = b_0$, d'où $E = (a_1 \oplus b_1)(a_0 \oplus b_0)$

$a_1 a_0 < b_1 b_0$ signifie : $a_1 < b_1$ ou $(a_1 = b_1 \text{ et } a_0 < b_0)$, d'où $I = \overline{a_1} b_1 + (a_1 \oplus b_1) \overline{a_0} b_0$.

De même $S = a_1 \overline{b_1} + (a_1 \oplus b_1) a_0 \overline{b_0}$

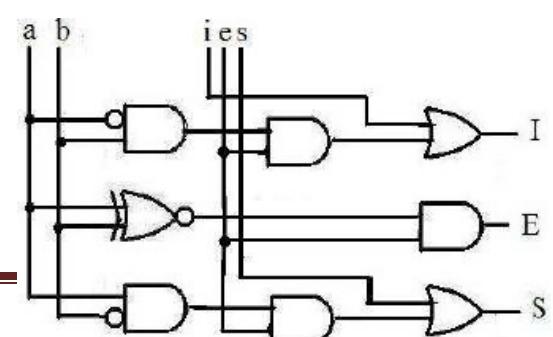
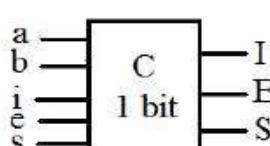
- d. L'objectif est d'obtenir le circuit du comparateur deux bits du c. en utilisant deux comparateurs 1 bit en cascade. On utilise le schéma suivant : Donner les équations I, E, S

On trouve :

$$I = e \bar{a} \bar{b} + i$$

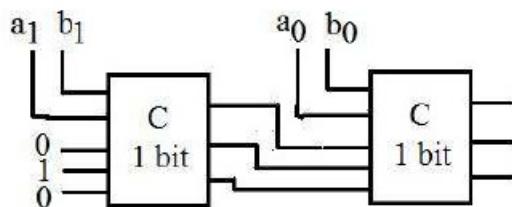
$$E = \overline{a \oplus b} e$$

$$S = e a \bar{b} + s$$



- e. Construire un comparateur 2 bits grâce à deux comparateurs 1 bit. En s'inspirant du point d

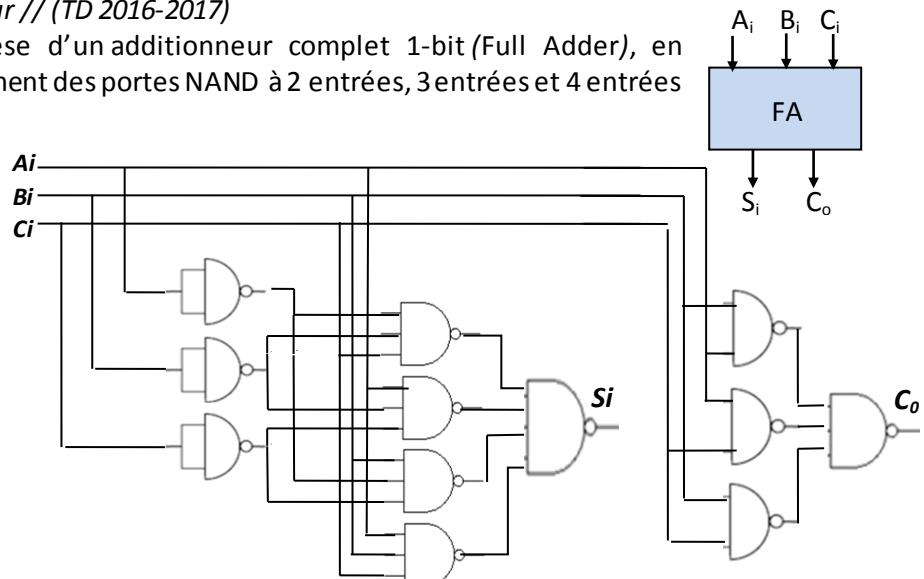
Pour retrouver les équations du c), il suffit de joindre les deux comparateurs indiqué, en mettant au départ $i = 0$, $e = 1$ et $s = 0$. En généralisant on passerai à un comparateur n bits.



Exercice2 Additionneur // (TD 2016-2017)

- a. Faites la synthèse d'un additionneur complet 1-bit (Full Adder), en utilisant uniquement des portes NAND à 2 entrées, 3 entrées et 4 entrées

C_i	A_i	B_i	S_i	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

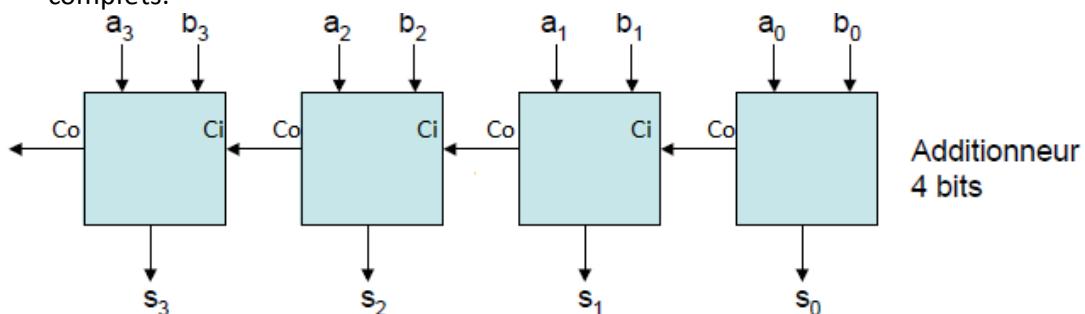


$$S_i = \bar{C}_i \bar{A}_i B_i + \bar{C}_i A_i \bar{B}_i + C_i \bar{A}_i \bar{B}_i + C_i A_i B_i$$

$$C_o = \bar{C}_i A_i B_i + C_i \bar{A}_i B_i + C_i A_i \bar{B}_i + C_i A_i B_i$$

$$C_o = A_i B_i + C_i B_i + C_i A_i$$

- b. Donner le schéma logique d'un additionneur parallèle 4 bits en utilisant les additionneurs complets.



- c. Calculer le délai de calcul de S_3 et de C_4 ? on prendra comme temps de réponse des portes NAND les valeurs suivantes : $t_{PHL} = t_{PLH} = 10\text{ns}$ (valeur typique), 15ns (valeur max)

	S ₀	S ₁	S ₂	S ₃	C ₀ [4]
Typ (ns)	30	50	70	90	80
Max (ns)	45	75	105	135	120

- d. Comparer les délais calculés, avec les valeurs données pour le circuit 74LS283 (tpHL et tpLH comprises entre 11 et 16ns)

En typique la version réalisée avec des portes élémentaires effectue une addition 4 bits en un temps de 90ns qui est 7 à 8 fois plus grand que le temps mis par le SN 74LS283 et qui est compris entre 11 et 16 ns.