CHEBBI Imen

BOUGHARRIOU Emna

INDP1 F

Unité Arithmétique Et Logique

(UAL 4bits)

I. Introduction :

L'UAL (Unité d'Arithmétique et de Logique) est l'élément de base d'un microprocesseur. Comme son nom l'indique, son rôle est la réalisation d'opérations arithmétiques (additions, soustractions...), logiques (OR, AND, NOR...), mais aussi de décalage, et de transfert.

II. Principe de fonctionnement :

Sous sa forme la plus simple, l'UAL possède une entrée reliée au bus de données du microprocesseur, une autre reliée à un registre interne, et une sortie reliée au même registre et au bus de données. Elle possède également un bus destinée à la sélection des opérations à réaliser.

Le cycle pour la réalisation d'une opération est le suivant :
-Présentation d'une donnée sur la première entrée.
-Sélection de l'opération de chargement dans le registre interne. La donnée est ainsi véhiculée vers la sortie de l'UAL, chargée dans le registre, et présentée sur la seconde entrée.
-Présentation d'une nouvelle donnée sur la première entrée.
-Sélection de l'opération à réaliser.
-Récupération du résultat en sortie de l'UAL et sur le bus de données du microprocesseur.

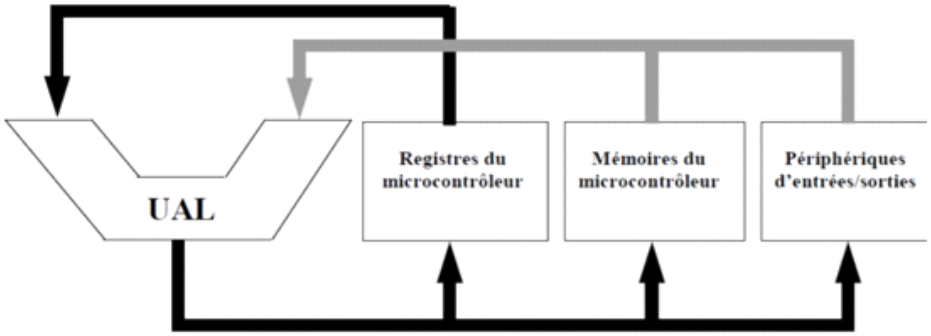


Figure 1 : principe de fonctionnement d'UAL

III. Objectifs

Réaliser une unité arithmétique et logique (UAL) qui permet de réaliser un ensemble d'opérations

Arithmétiques et logiques sur des opérandes de 4 bits.

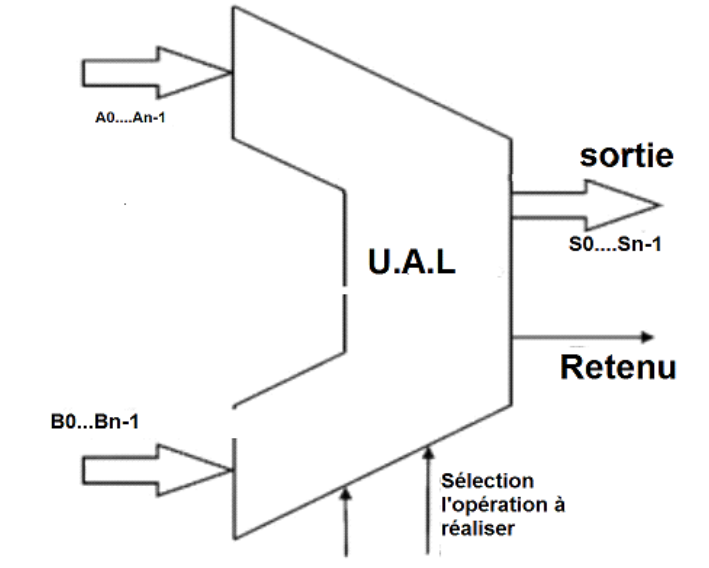


Figure 1: Architecture d'UAL

IV. Réalisation du travail

A. Additionneur

a) Présentation

On a basé sur le principe d'un additionneur 4 bits du fait que la conception d'un additionneur 4 bits peut être réalisé en cascadant 4 additionneurs 1 bits et en propageant la retenue d'un étage à l'autre.

b) Input et output

*input:

A=(A1, A2,A3,A4)

B=(B1,B2,B3,B4)

*output:

S=(S1,S2,S3,S4),C0

c) Table de vérité :

Il est composée par 2^8=256 cases. Il est difficile de la dessiner.

d) Logigramme

i) 1ère méthode :

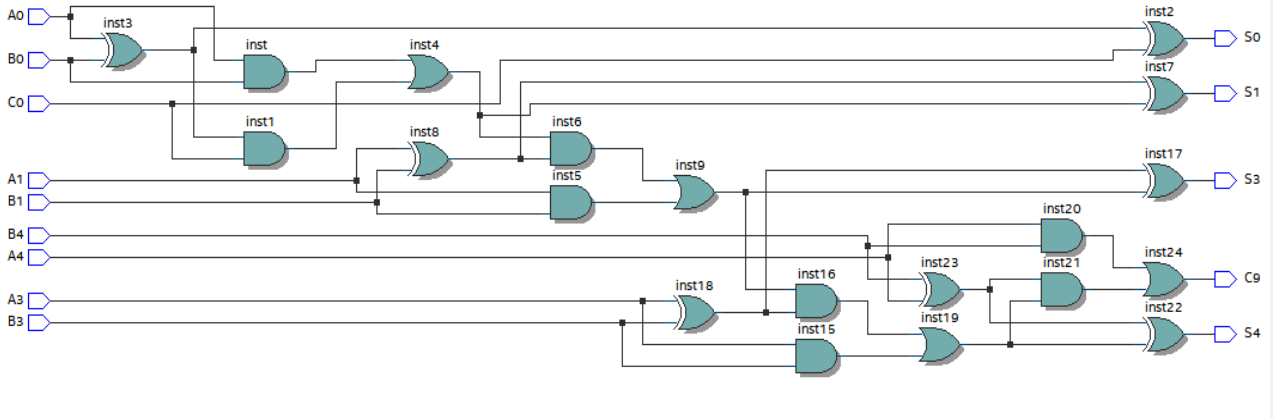


Figure 2: logigramme 1 Addition

2éme méthode : on utilise directement la plaque 7483

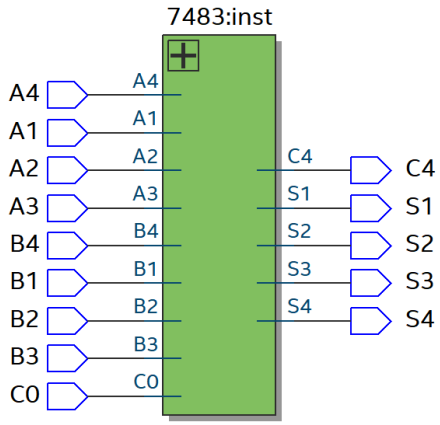


Figure 3: logigramme 2 Addition

e) Simulation

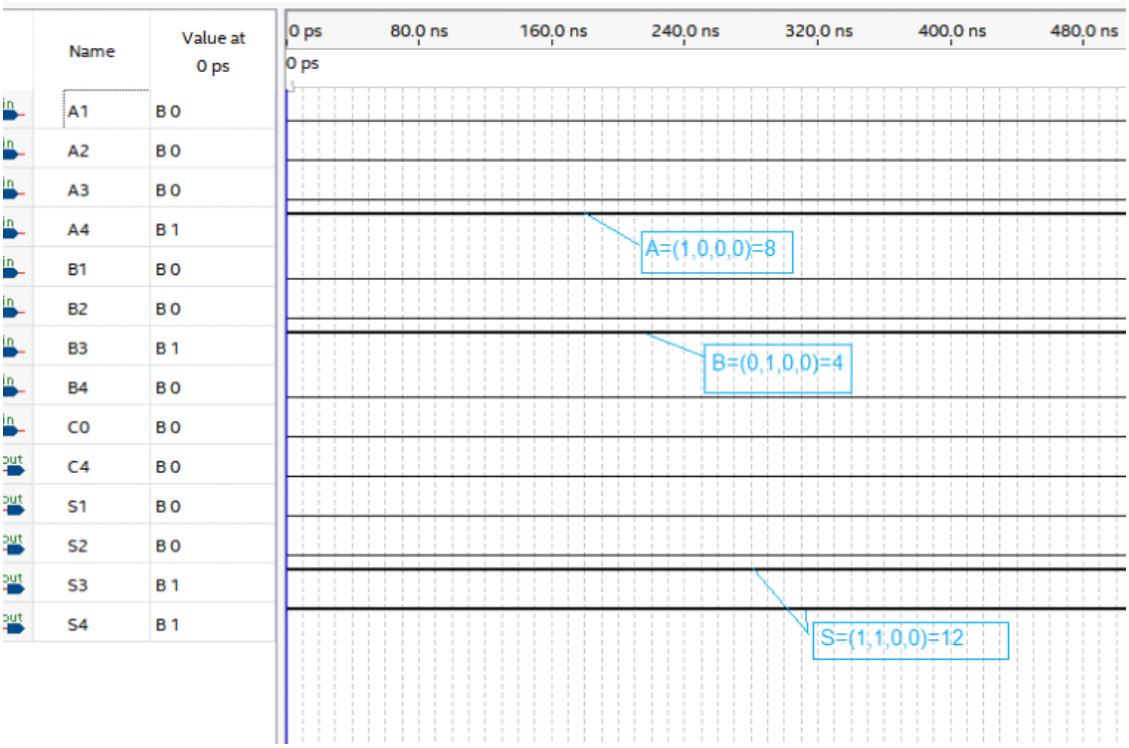


Figure 4: Simulation Addition

B. Soustraction

a) Présentation

Pour effectuer des soustractions on utilise les propriétés du complément à 2. Soit Y le nombre binaire obtenu en remplaçant les 1 de Y par un 0 et les 0 par un 1. Effectuer D = X − Y est équivalent à D = X + Y + 1 en complément à 2.

Le bit de poids faible de l'additionneur n'a pas de retenue d'entrée. Pour ajouter le 1 à D, il suffit de mettre un 1 comme retenue dans l'additionneur de poids faible.

Soit C un signal de contrôle valant 0 si on veut faire une addition, et 1 si on veut faire une soustraction.

On utilise ce signal C comme retenue du bit de poids faible de l'additionneur.

Pour avoir Y il suffit d'ajouter une porte XOR réalisant (Y XOR C) en entrée de chacun des additionneurs complets : Si C vaut 0, la valeur d'entrée de l'additionneur n est Yn (suiveur) et si C vaut 1, la valeur d'entrée est Yn (inverseur).

Donc, si C vaut 0, l'opération effectuée par le circuit est l'addition X + Y, et si C vaut 1, l'opération effectuée est X − Y.

Le bit de la retenue finale devient un bit de signe. (bit à 1 résultat positif, bit à 0 résultat négatif).

b) input_output

*input:

A=(A1, A2,A3,A4)

B=(B1,B2,B3,B4)

*output:

S=(S1,S2,S3,S4),C4

c) Logigramme

la soustraction d'opérandes sur 4 bits (Addition et Complément Vrai)

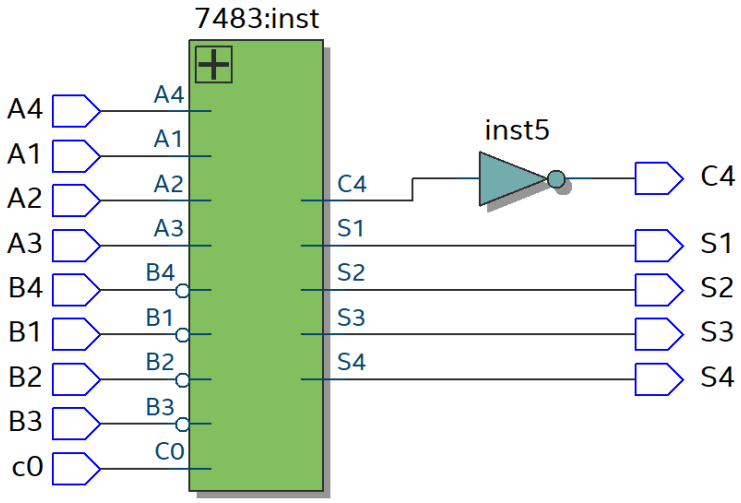


Figure 5: Logigramme Soustraction

d) Simulation

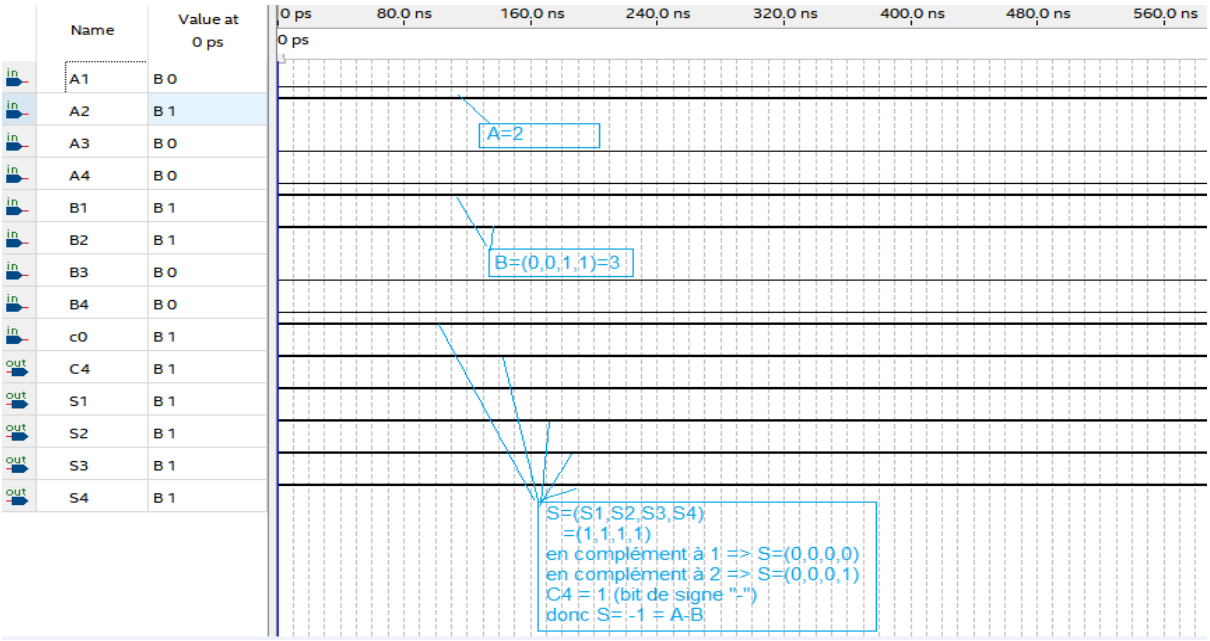


Figure 6: Simulation Soustraction

C. Multiplieur 4 bits

a) Présentation

Le multiplieur 4 bits est réalisé avec 16 portes AND et 3 additionneurs de 4 bits, les portes AND permettent de faire les multiplications logiques entre les entrées, après ces opérations de multiplication les additionneurs vont réaliser l'addition logique verticalement, puis on obtient les résultats de sortie. Ce multiplieur 4 bits a 8 entrées et 8 sorties et il est capable de faire n'importe qu'elle multiplication arithmétique de 4 bits.

b) input_output

*input:

A= (A1, A2, A3, A4)

B= (B1, B2, B3, B4)

*output:

S=(S1,S2,S3,S4,S5,S6,S7,S8)

c) Logigramme

Opération multiplication d'opérandes sur 4 bits avec un résultat sur 8 bits

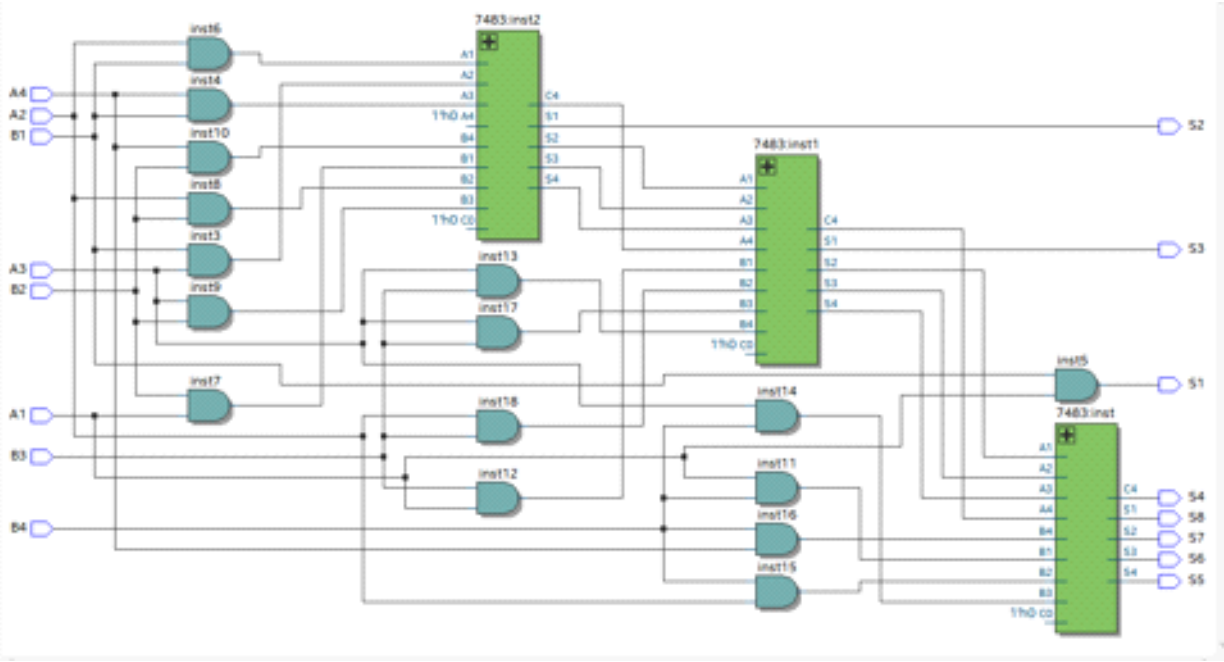


Figure 7: Logigramme Multiplieur 4 Bits

d) Simulation

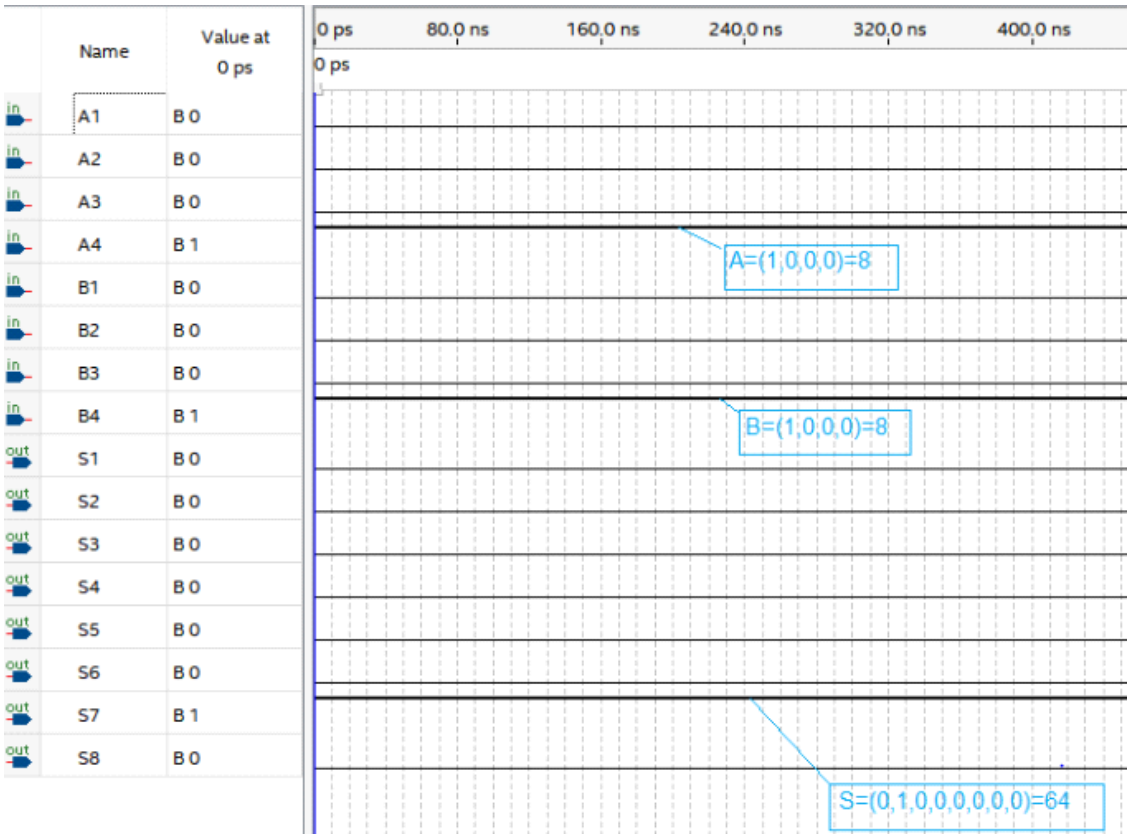


Figure 8: Simulation multiplueur 4 1bits

D. Comparateur

a) Présentation

Un comparateur binaire est un circuit logique qui effectue la comparaison entre 2 nombres binaires généralement notés A et B.

Il possède 3 sorties notées A = B, A > B et A < B qui indiquent le résultat de la comparaison comme suit :

- Si le nombre A est égal au nombre B (A = B), la sortie A = B passe à l'état 1 tandis que les sorties A > B et A < B passent à l'état 0.
- Si le nombre A est strictement supérieur au nombre B, seule la sortie A > B passe à l'état 1.
- Si le nombre A est strictement inférieur au nombre B, seule la sortie A < B passe à l'état 1.

Nous allons voir comment réaliser à l'aide de portes logiques un comparateur de 2 chiffres binaires.

Le circuit intégré 7485 est un comparateur 4 bits, c'est-à-dire qu'il effectue la comparaison de

deux nombres de 4 bits.

De plus, il dispose de 3 entrées notées A = B, A > B et A < B qui autorisent la mise en cascade de plusieurs circuits comparateurs du même type.

b) input_output

*input :

A= (A0, A1, A2, A3)

B= (B0, B1, B2, B3)

*output :

S= (S1, S2, S3)

c) Logigramme

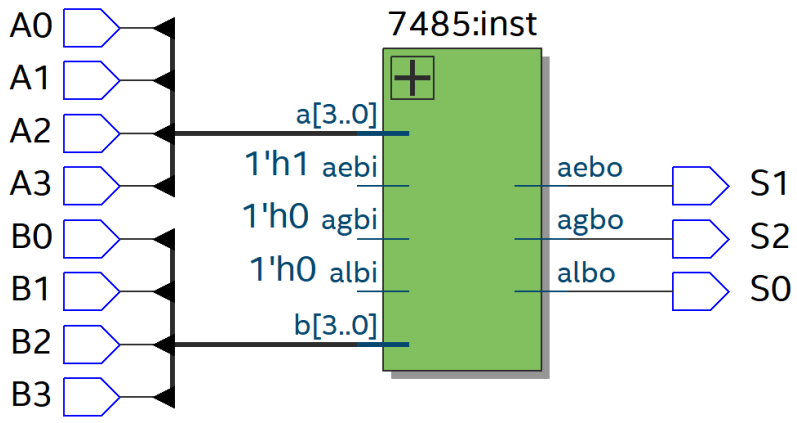


Figure 9: Logigramme comparateur 4 bits

d) Table de vérité

Entrées des nombres				Entrées cascadables			Sorties		
A3, B3	**A2, B2**	**A1, B1**	**A0, B0**	**A > B**	**A < B**	**A = B**	**A > B**	**A < B**	**A = B**
A3 > B3	X	X	X	X	X	X	1	0	0
A3 < B3	X	X	X	X	X	X	0	1	0
A3 = B3	A2 > B2	X	X	X	X	X	1	0	0
A3 = B3	A2 < B2	X	X	X	X	X	0	1	0
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	1	0	0
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	0	1	0
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	1	0	0
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	0	1	0
A3 = B3	A2 = B2	A1 = B1	A0 = B0	1	0	0	1	0	0
A3 = B3	A2 = B2	A1 = B1	A0 = B0	0	1	0	0	1	0
A3 = B3	A2 = B2	A1 = B1	A0 = B0	0	0	1	0	0	1
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	1	0	0	1
A3 = B3	A2 = B2	A1 = B1	A0 = B0	1	1	0	0	0	0
A3 = B3	A2 = B2	A1 = B1	A0 = B0	0	0	0	1	1	0

Figure 10: Table de vérité d'un comparateur

S_i=1	Signification
S0	A<B
S1	A=B
S2	A>B

e) Simulation

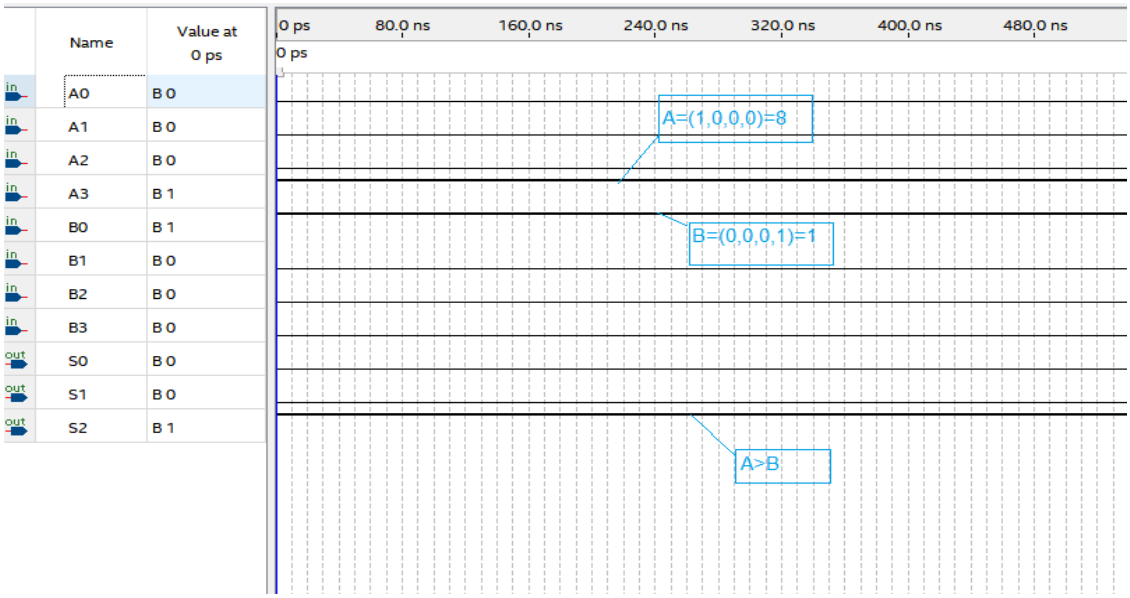


Figure 11: Simulation Comparateur 4 Bits

E. Opération logique de 4 bits

1. Porte And

a) input_output

*input :

A= (A0, A1, A2, A3)

B= (B0, B1, B2, B3)

*output :

S= (S0, S1, S2, S3)

b) Logigramme

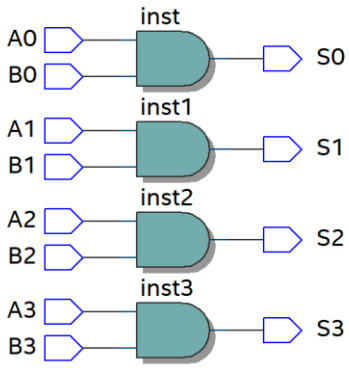


Figure 12: Logigramme And 4 Bits

c) Simulation

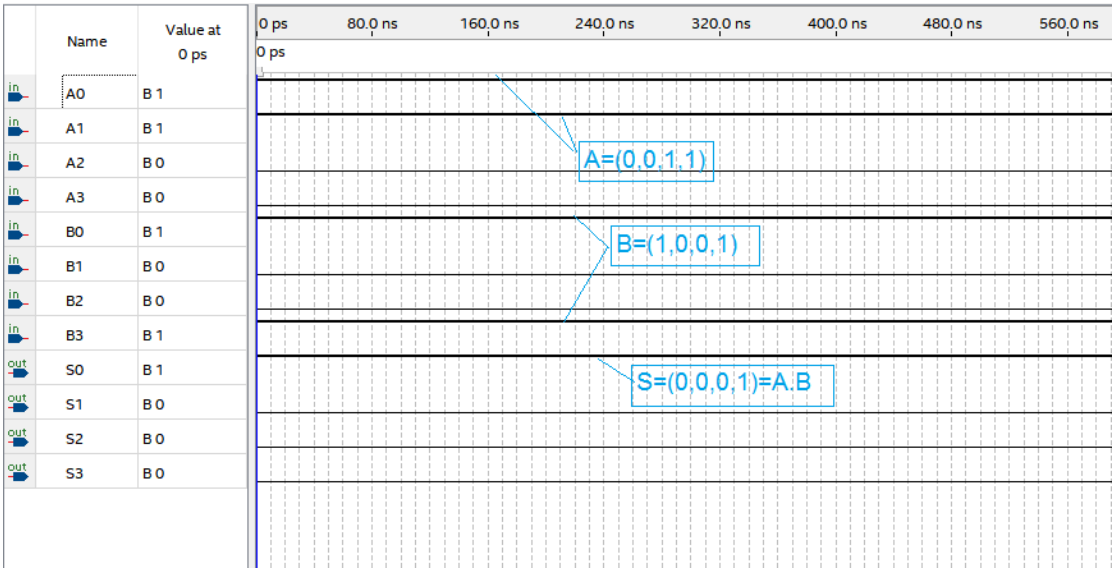


Figure 13: Simulation And 4 Bits

2. Porte Nor

a) input_output

*input :

A= (A0, A1, A2, A3)

B= (B0, B1, B2, B3)

*output:

S= (S0, S1, S2, S3)

b) Logigramme

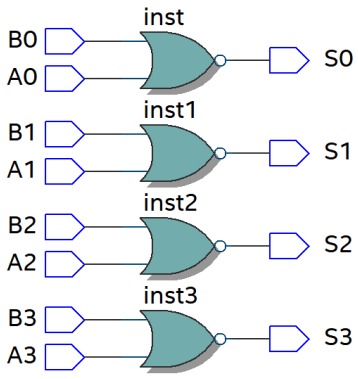


Figure 14: Logigramme Nor 4 Bits

c) Simulation

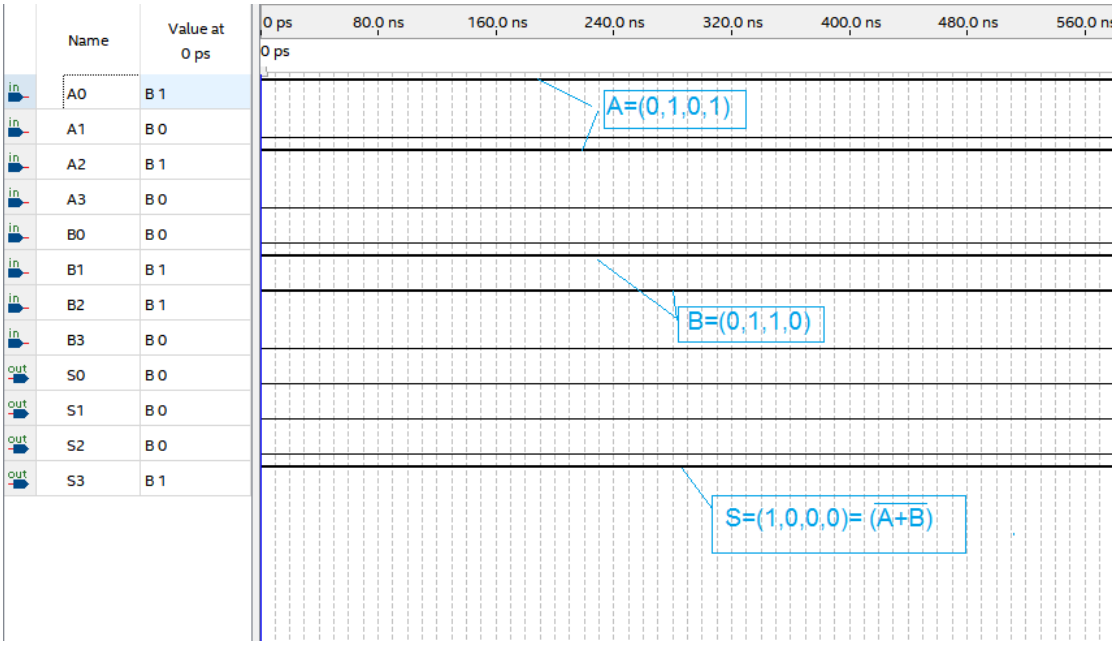


Figure 15: Simulation Nor 4 Bits

3. Porte XOR

a) input_output

*input :

A= (A0, A1, A2, A3)

B= (B0, B1, B2, B3)

*output :

S= (S0, S1, S2, S3)

b) Logigramme

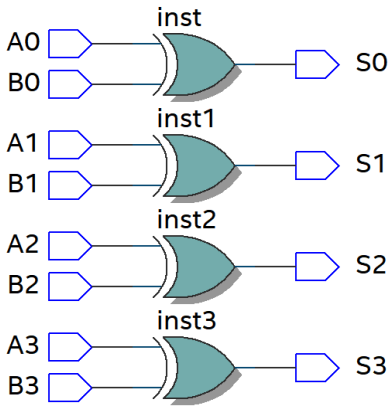


Figure 16: Logigramme Xor 4 Bits

c) Simulation

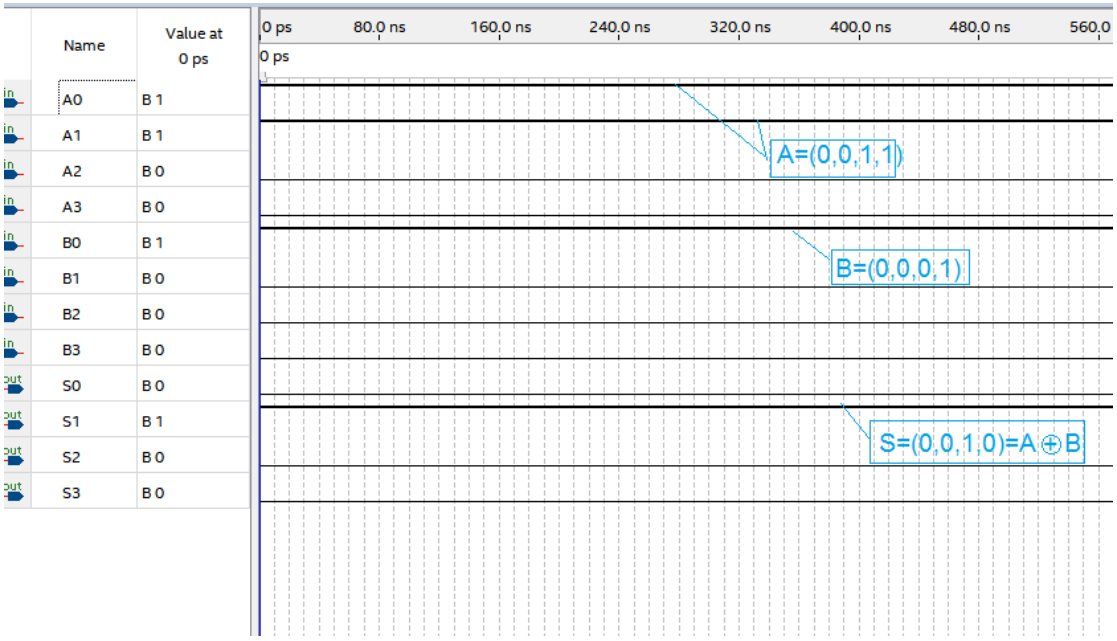


Figure 17: Simulation XOR 4 Bits

4. Porte OR

a) input_output :

*input :

A= (A0, A1, A2, A3)

B= (B0, B1, B2, B3)

*output:

S=(S0, S1, S2, S3)

b) Logigramme

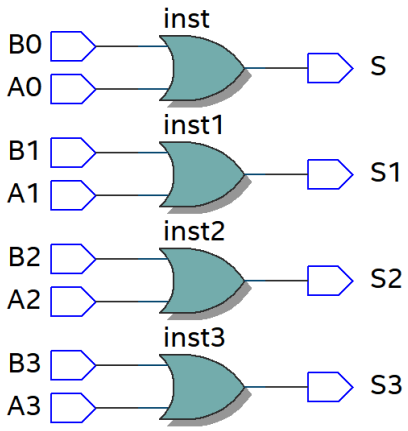


Figure 18: Logigramme OR 4 bits

c) Simulation

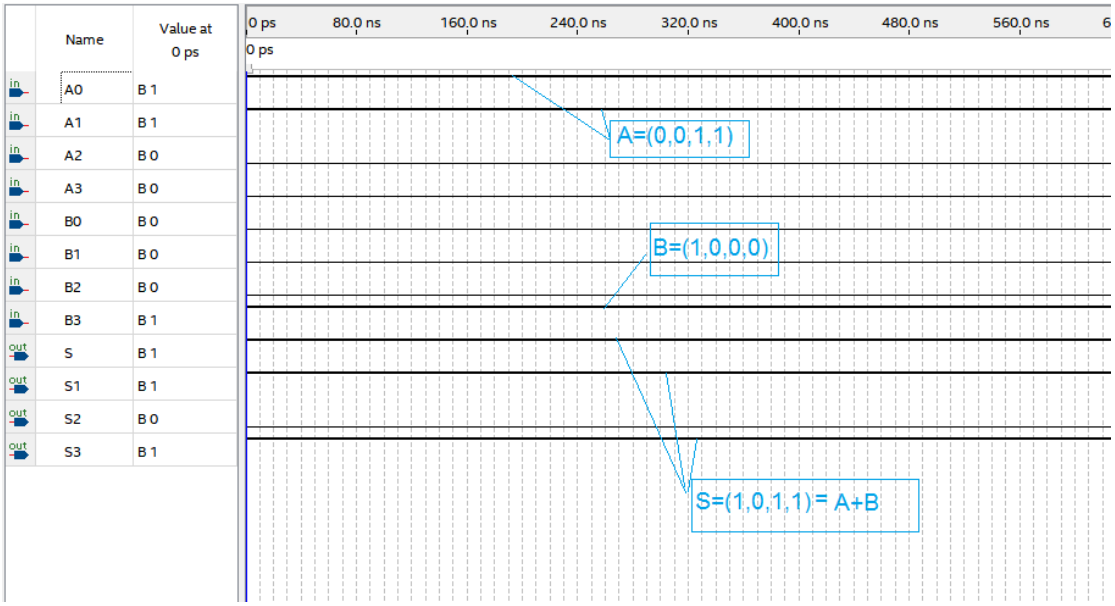


Figure 19: Logigramme OR 4 Bits

F. Décalage à gauche à droite

a) Présentation

Le circuit intégré 74 194 est un registre à décalage bidirectionnel 4 bits ayant deux entrées de commande (S0 et S1), une entrée d'horloge (CK), une entrée de données série pour le décalage à gauche (ESG), une entrée de données série pour le décalage à droite (ESD), quatre entrées parallèles (E1 à E4), une entrée asynchrone de remise à zéro générale prioritaire (CLR) et quatre sorties parallèles (Q1 à Q4).

b) Logigramme

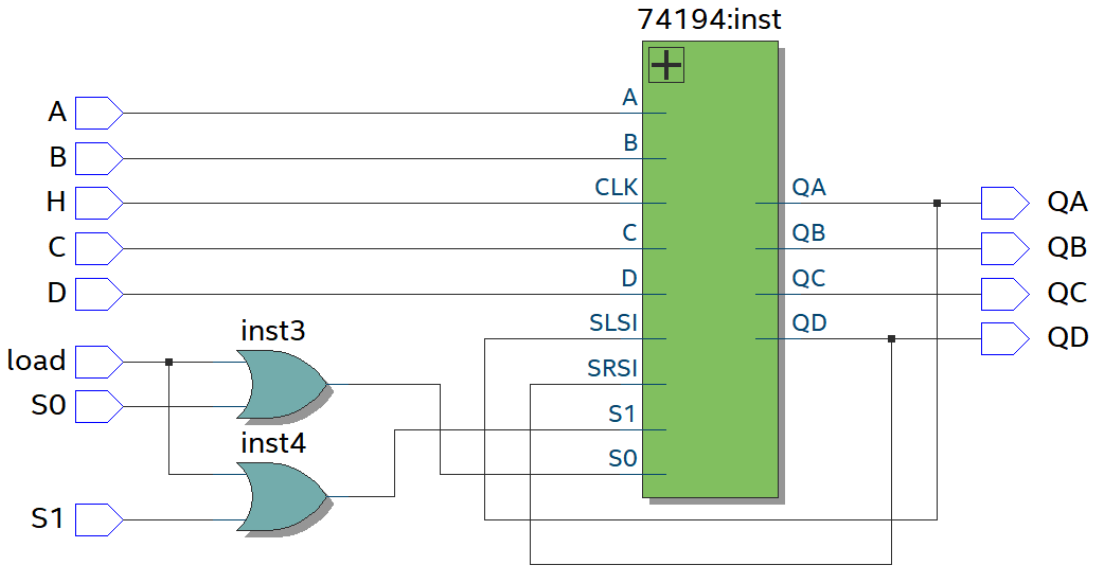


Figure 20: Logigramme Décalage 4 Bits

S0 et S1 sont deux bits de sélection

Mode 0 : rien ne se passe

Mode1 : Décalage à droite

Mode2 : Décalage à gauche

Mode3 : Chargement en parallèle de A,B,C et D

c) Simulation

1er cas : Décalage à gauche

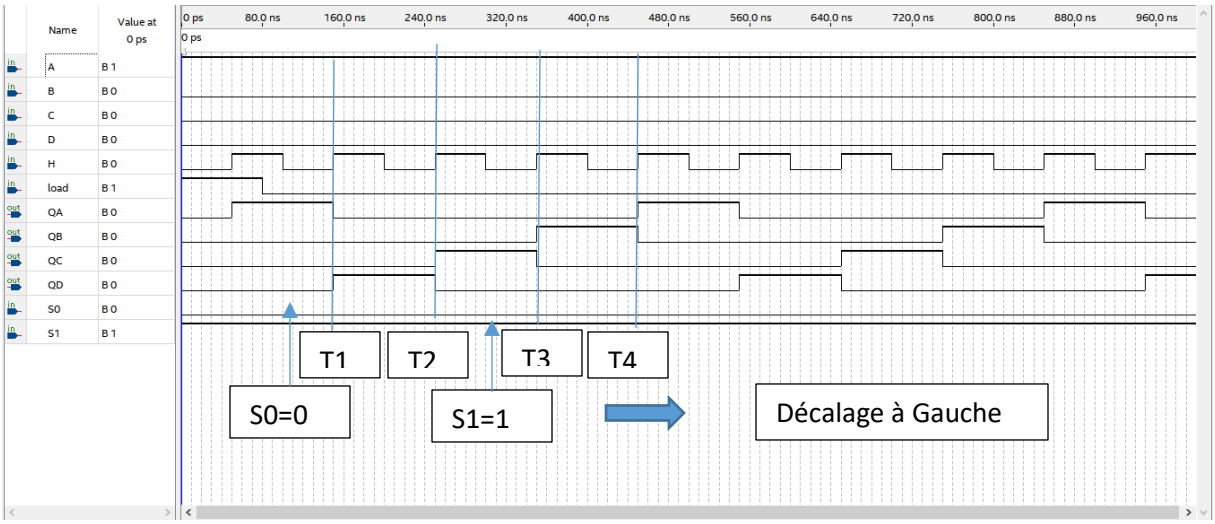


Figure 21: Simulation Décalage à gauche

2ème cas : Décalage à droite

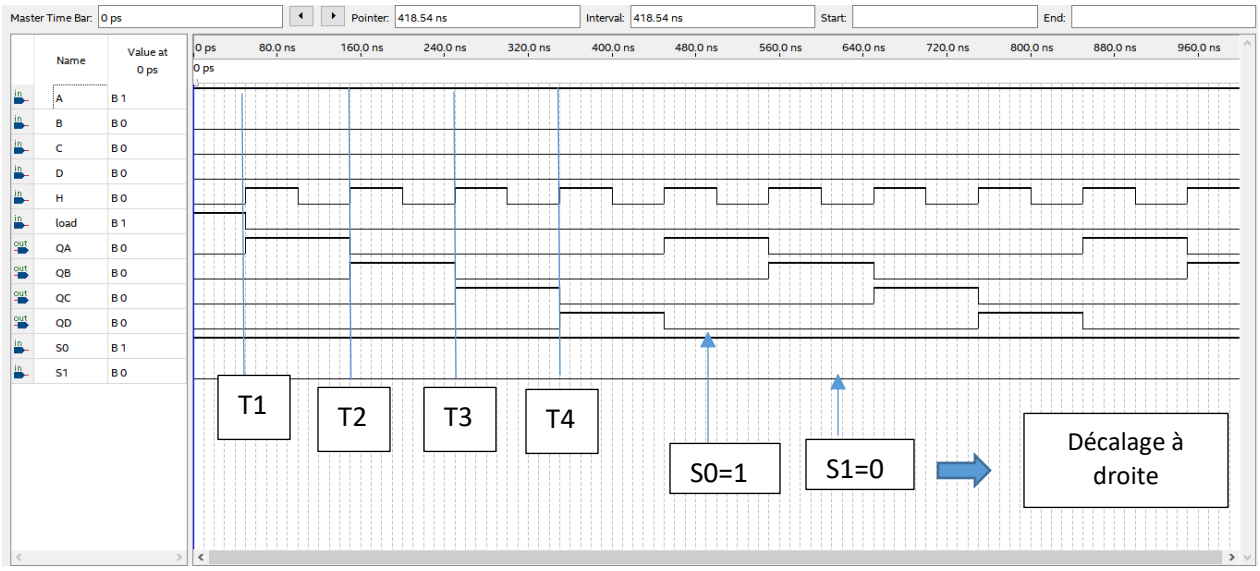


Figure 22: Simulation Décalage à droite

V. 1ère Unité arithmétique obtenue

1. Présentation :

On a utilisé 8 multiplexeurs 8-1 afin d'obtenir l'unité arithmétique composé par 8 Opérations qu'on a déjà définies ci-dessus.

81mux comporte :

- 8 entrées de données D0 à D7 et GN
- 3 entrées de sélection A, B et C
- 2sorties

Dans cette unité on a choisi 4 Opérations :

i. Soustraction
ii. Addition
iii. Multiplication
iv. Comparaison

Opération	Réference
Soustraction	100
Addition	101
Multiplication	110
Comparaison	111

2. Logigramme

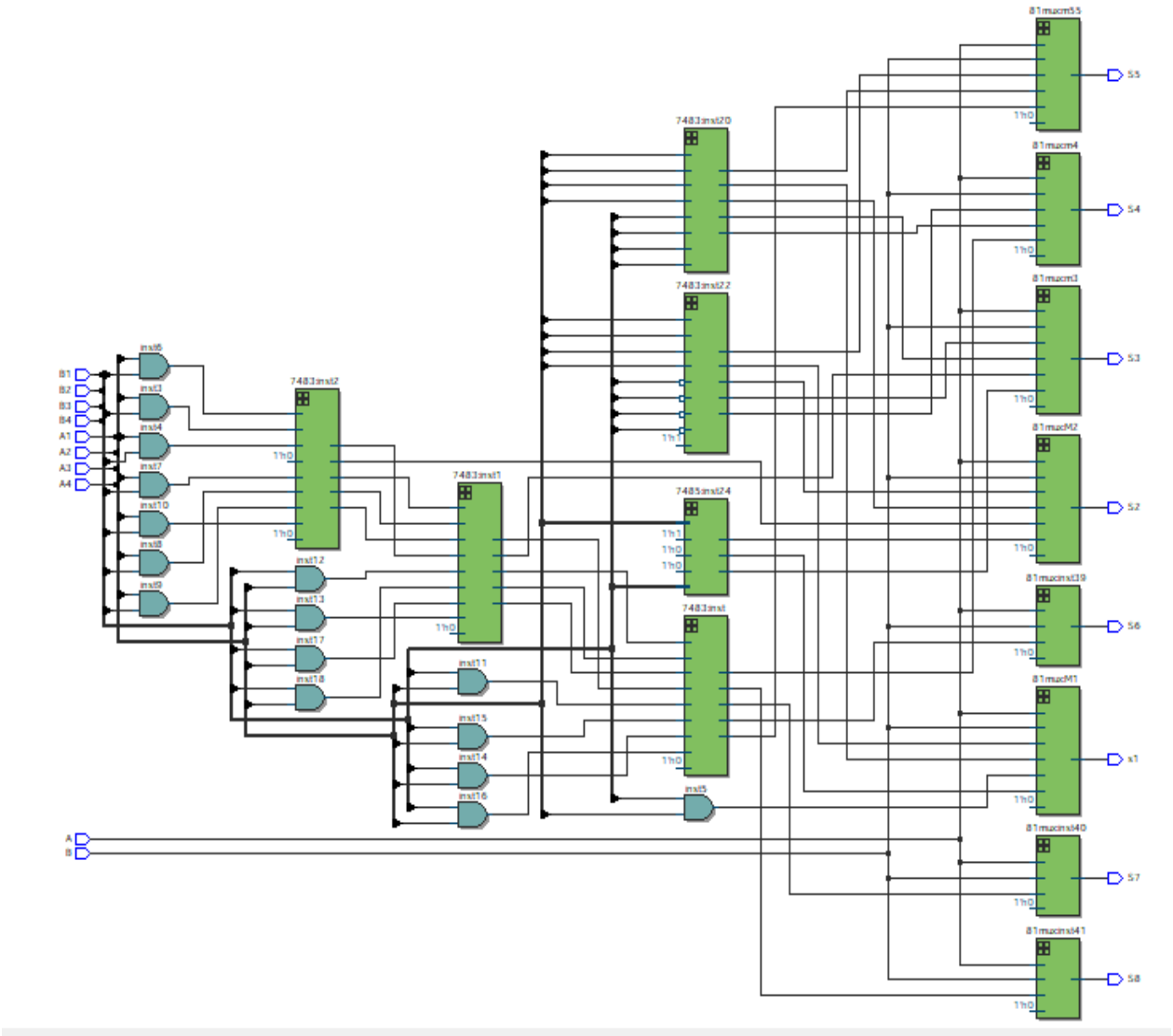


Figure 23: Logigramme Unité arithmétique n°1

3. Simulation

A chaque fois on fixe les valeurs de bits de sélection pour choisir l'opération à effectuer et cela en respectant le tableau situé au début de cette partie.

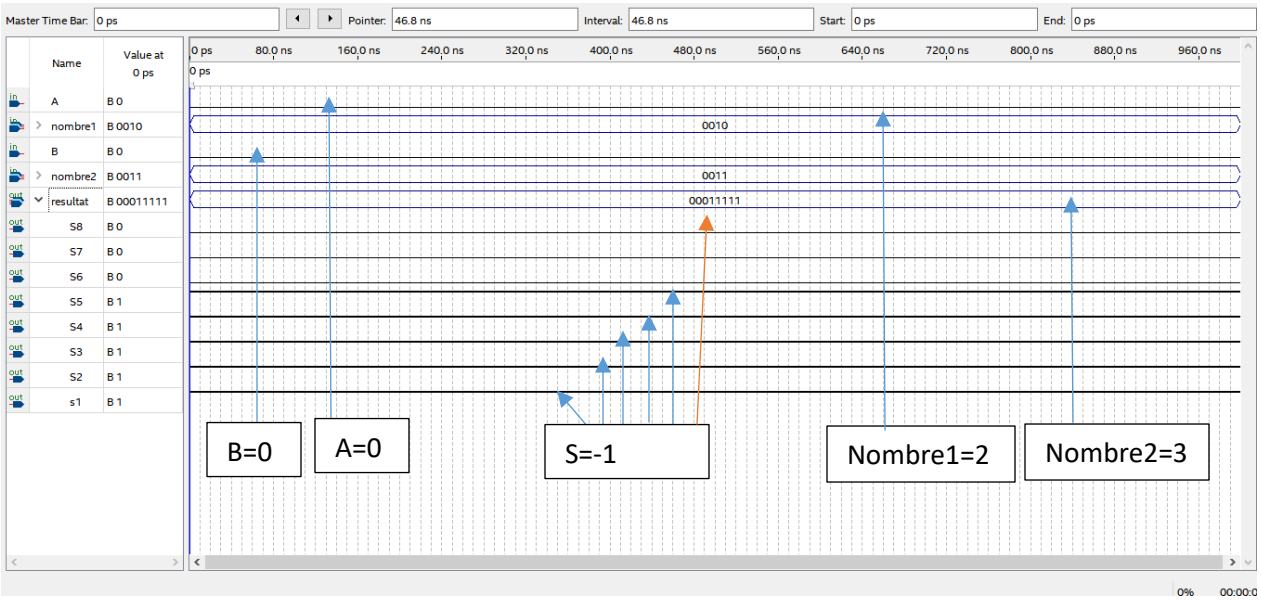


Figure 24: Simulation soustraction

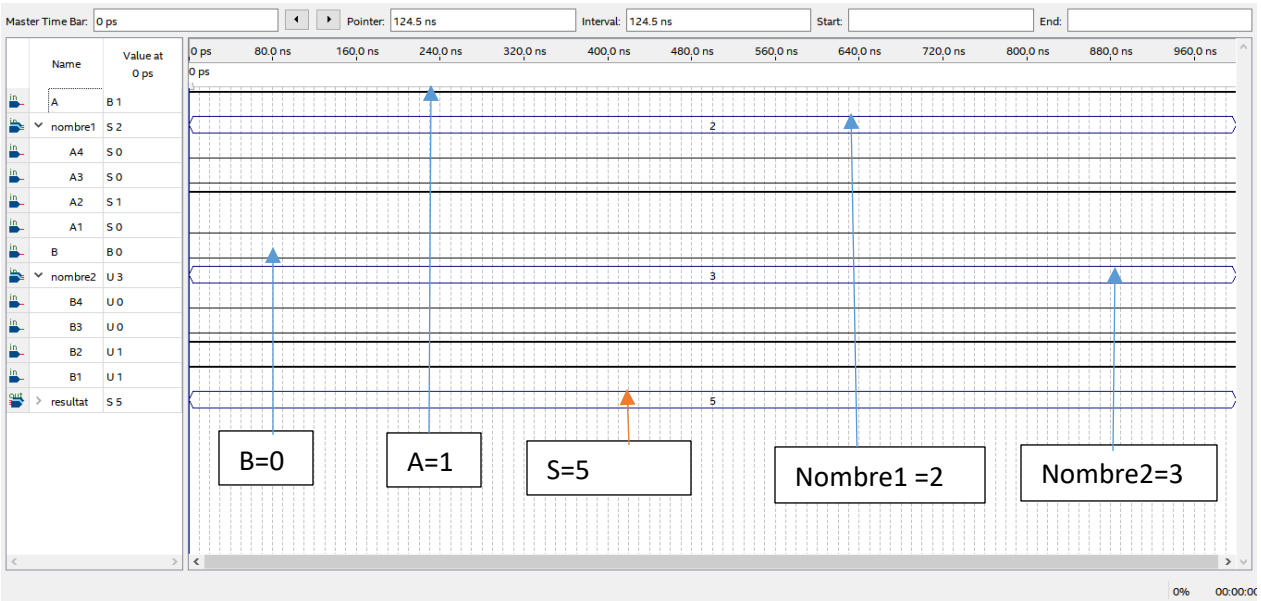


Figure 25: Simulation Addition

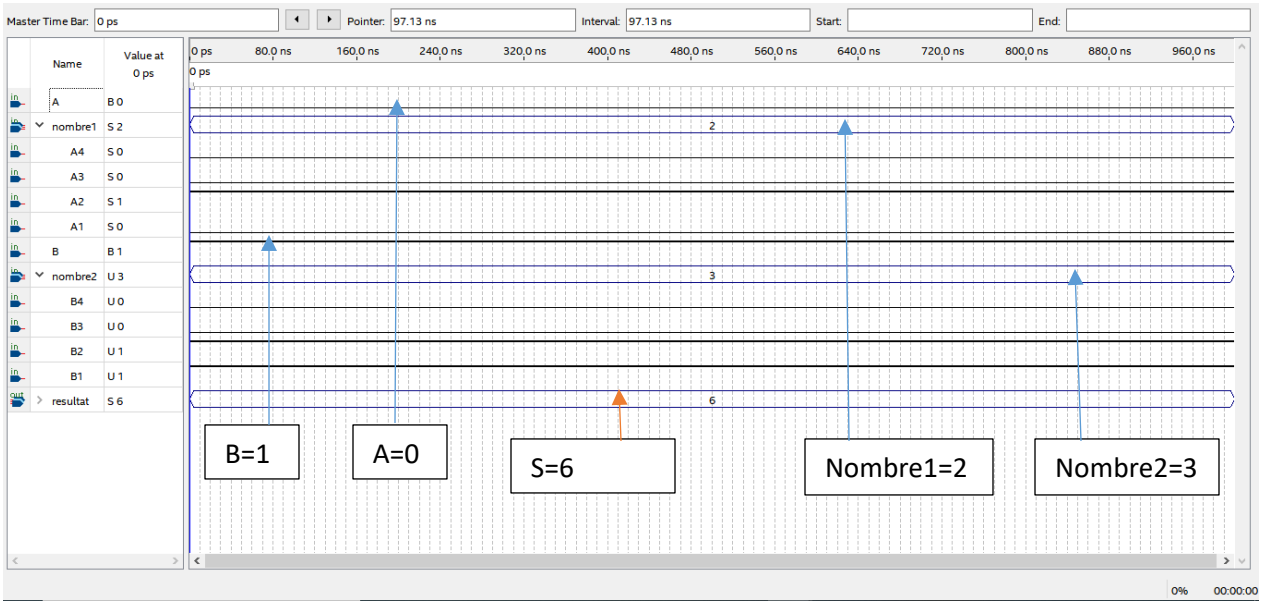


Figure 26: Simulation Multiplication

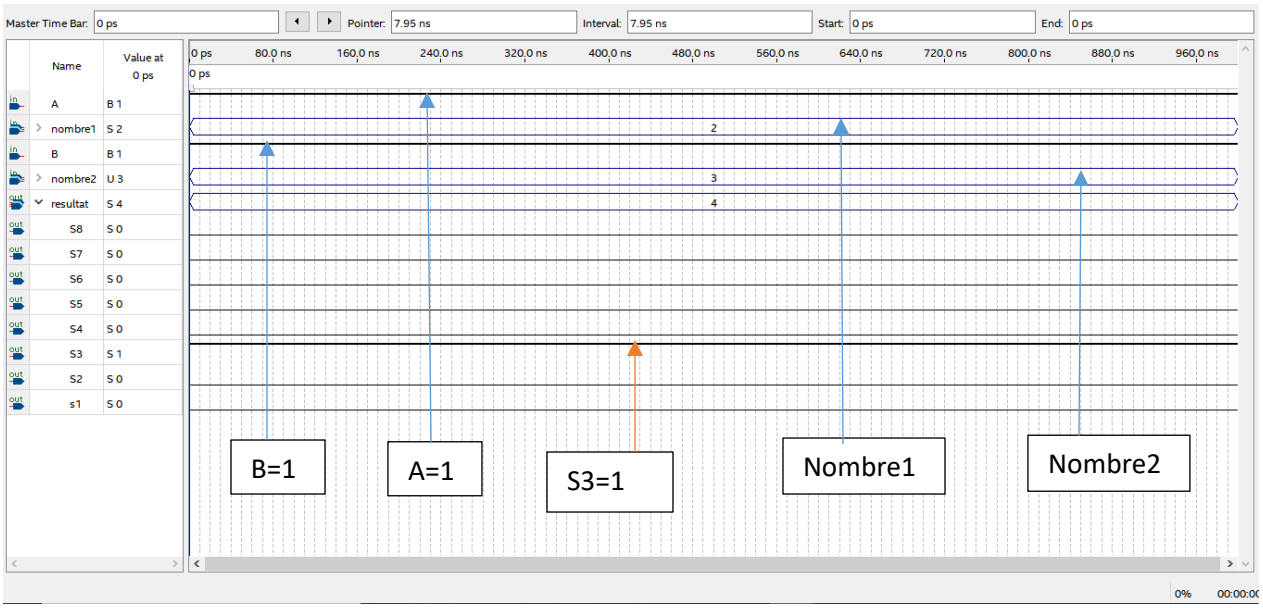
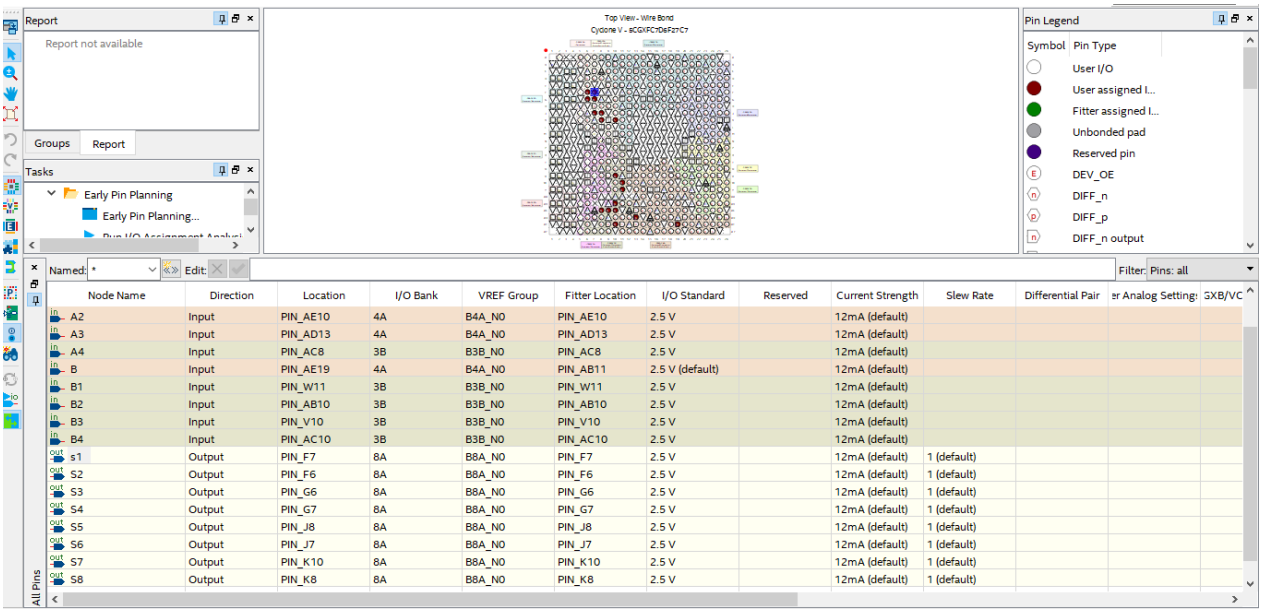


Figure 27: Simulation Comparaison

4. Implémentation



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	er Analog Setting:	GXB/VC
A2	Input	PIN_AE10	4A	B4A_N0	PIN_AE10	2.5 V		12mA (default)				
A3	Input	PIN_AD13	4A	B4A_N0	PIN_AD13	2.5 V		12mA (default)				
A4	Input	PIN_AC8	3B	B3B_N0	PIN_AC8	2.5 V		12mA (default)				
B	Input	PIN_AE19	4A	B4A_N0	PIN_AB11	2.5 V (default)		12mA (default)				
B1	Input	PIN_W11	3B	B3B_N0	PIN_W11	2.5 V		12mA (default)				
B2	Input	PIN_AB10	3B	B3B_N0	PIN_AB10	2.5 V		12mA (default)				
B3	Input	PIN_V10	3B	B3B_N0	PIN_V10	2.5 V		12mA (default)				
B4	Input	PIN_AC10	3B	B3B_N0	PIN_AC10	2.5 V		12mA (default)				
s1	Output	PIN_F7	8A	B8A_N0	PIN_F7	2.5 V		12mA (default)	1 (default)			
S2	Output	PIN_F6	8A	B8A_N0	PIN_F6	2.5 V		12mA (default)	1 (default)			
S3	Output	PIN_G6	8A	B8A_N0	PIN_G6	2.5 V		12mA (default)	1 (default)			
S4	Output	PIN_G7	8A	B8A_N0	PIN_G7	2.5 V		12mA (default)	1 (default)			
S5	Output	PIN_J8	8A	B8A_N0	PIN_J8	2.5 V		12mA (default)	1 (default)			
S6	Output	PIN_J7	8A	B8A_N0	PIN_J7	2.5 V		12mA (default)	1 (default)			
S7	Output	PIN_K10	8A	B8A_N0	PIN_K10	2.5 V		12mA (default)	1 (default)			
S8	Output	PIN_K8	8A	B8A_N0	PIN_K8	2.5 V		12mA (default)	1 (default)			

Figure 28: Implémentation sur la carte

VI. 2ème unité arithmétique :

1. Présentation

Dans cette unité on a choisi 4 Opérations :

i. XOR
ii. AND
iii. NOR
iv. DECALAGE :
   - droite
   - gauche

Remarque : Concernant le Décalage On a utilisé B0 et B1 comme les deux bits de sélection du circuit 74194.

Opération	Référence
XOR	100
And	101
NOR	110
Décalage	111

2. Logigramme

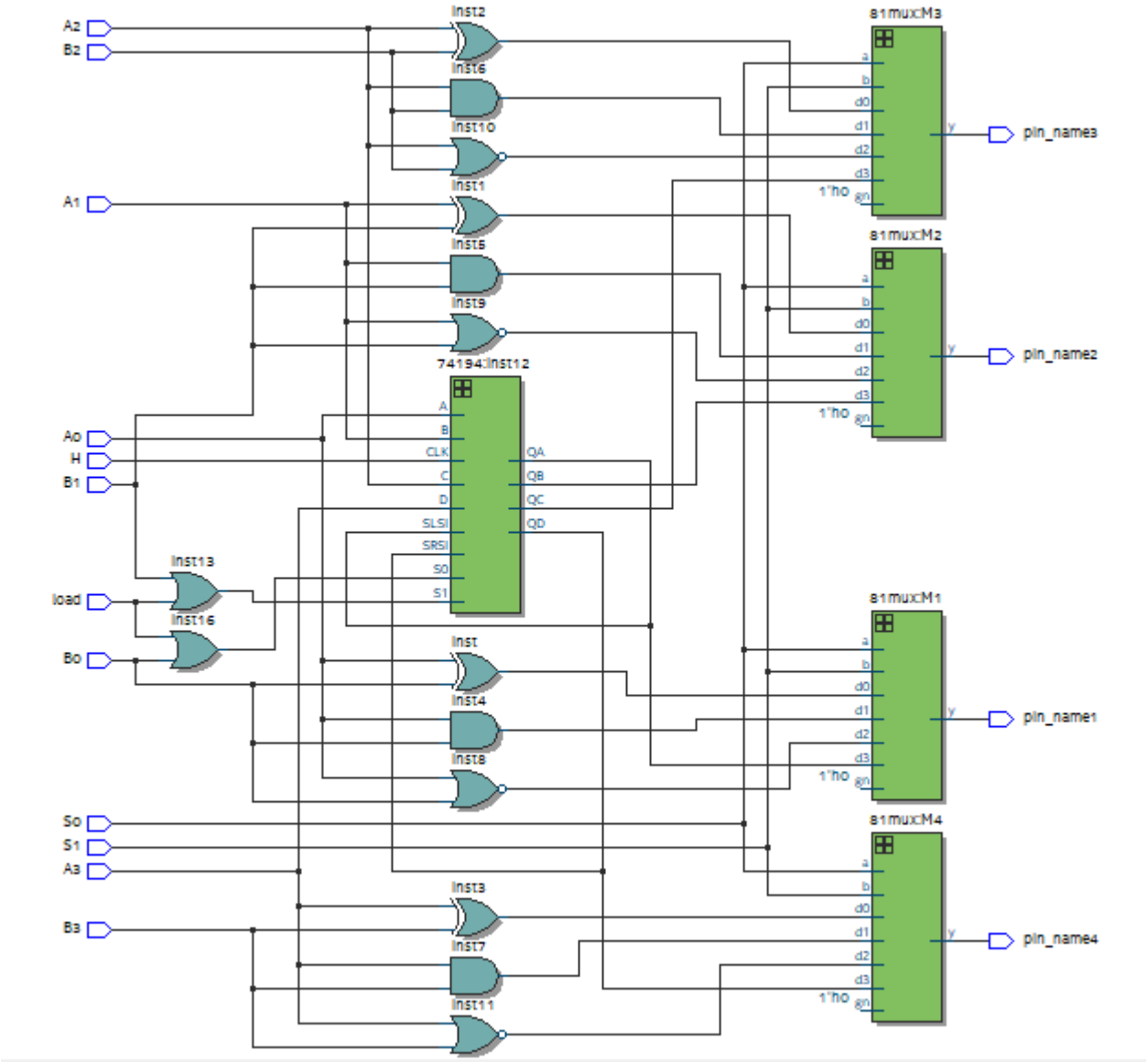


Figure 29: Logigramme de l'unité n°2

3. Simulation

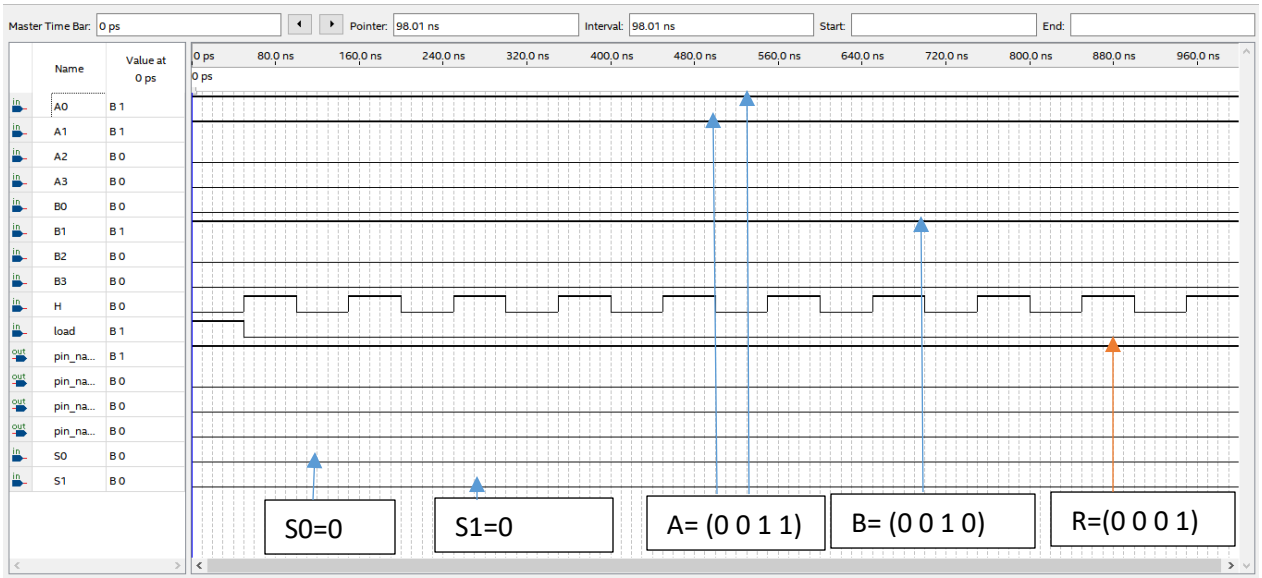


Figure 30: Simulation de la porte XOR

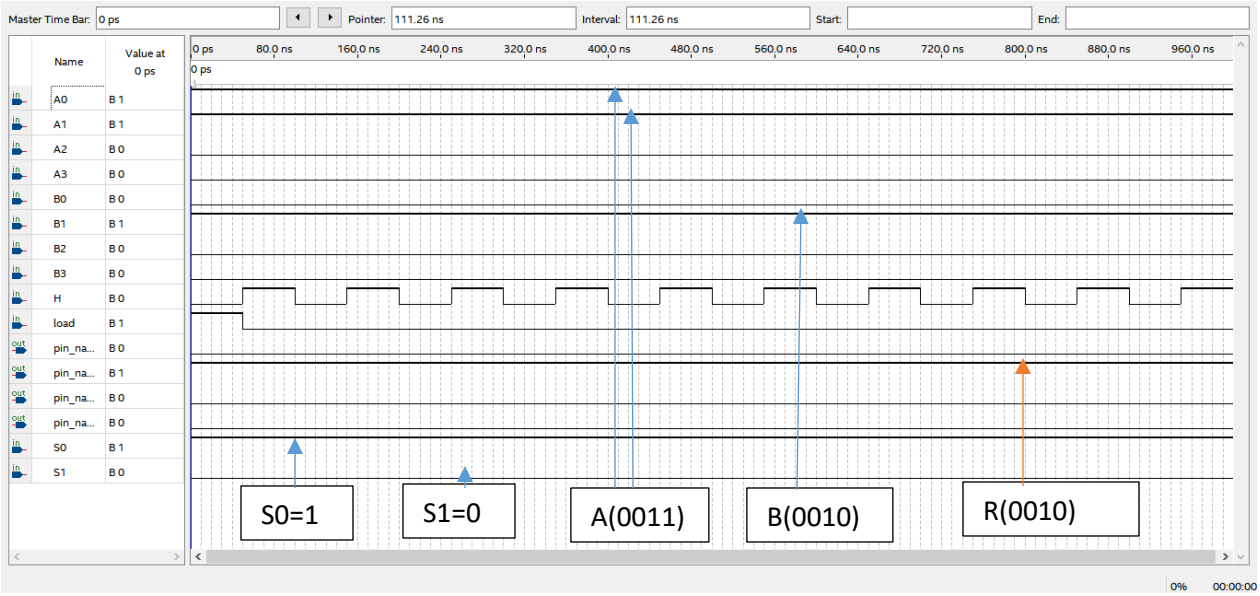


Figure 31: Simulation de la porte AND

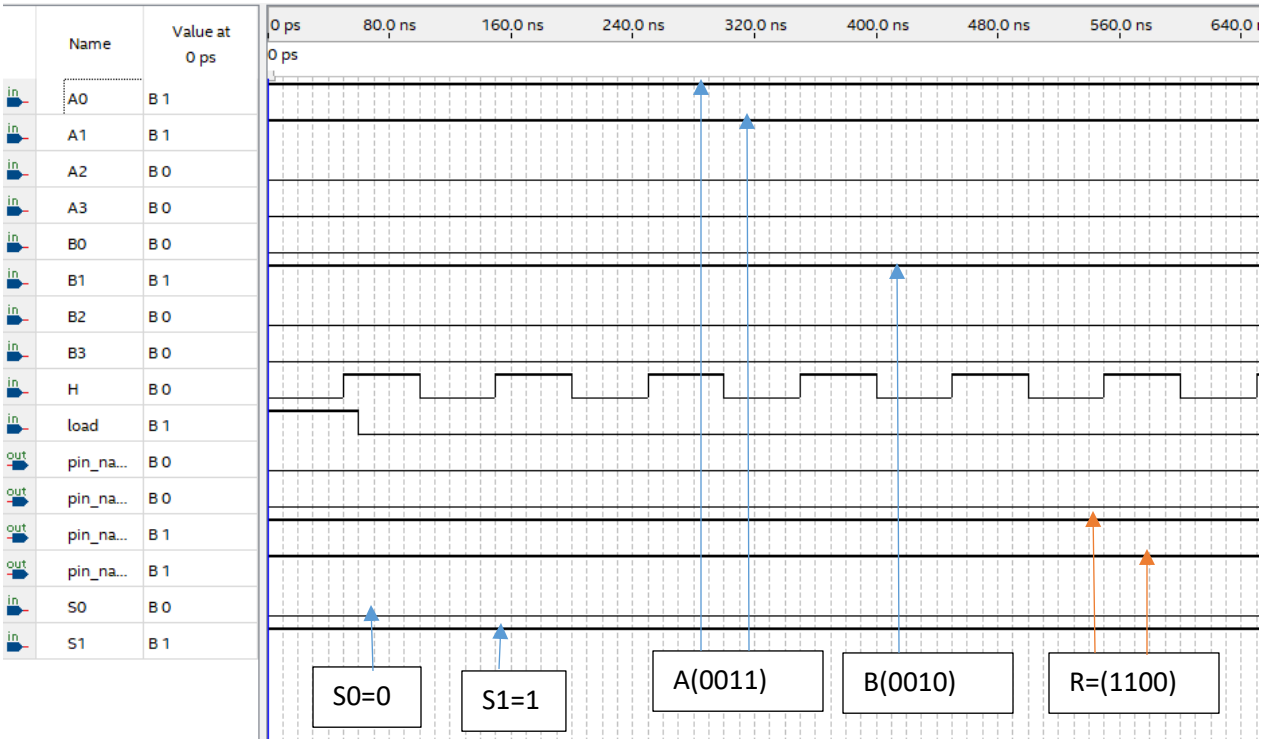


Figure 32: Simulation de la porte NOR

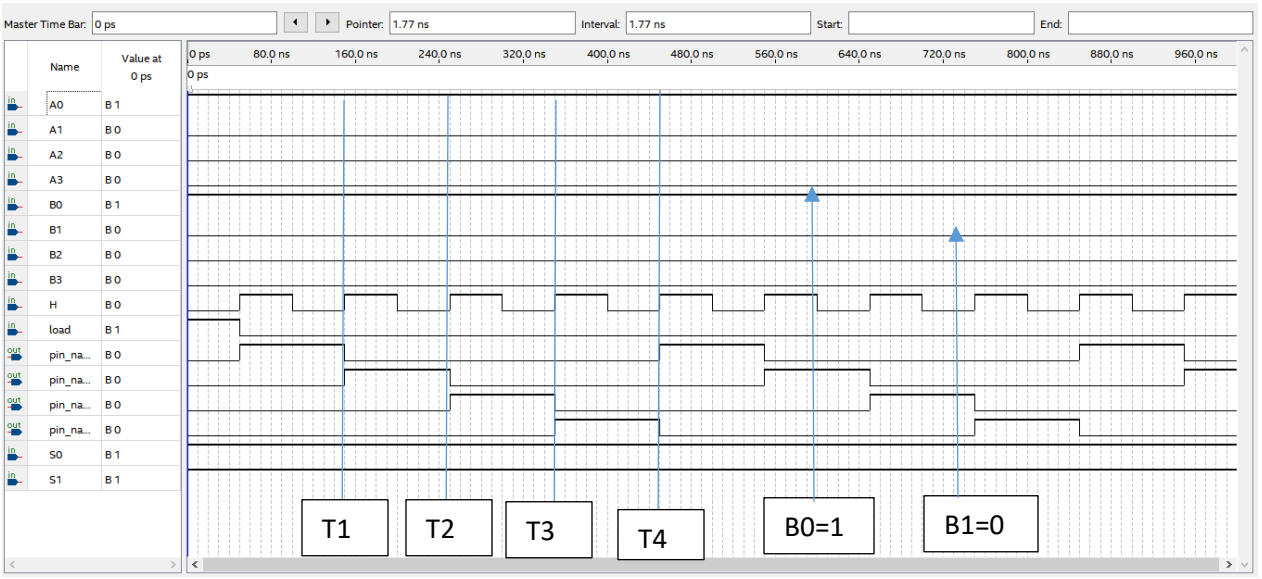


Figure 33: Simulation Décalage à Droite

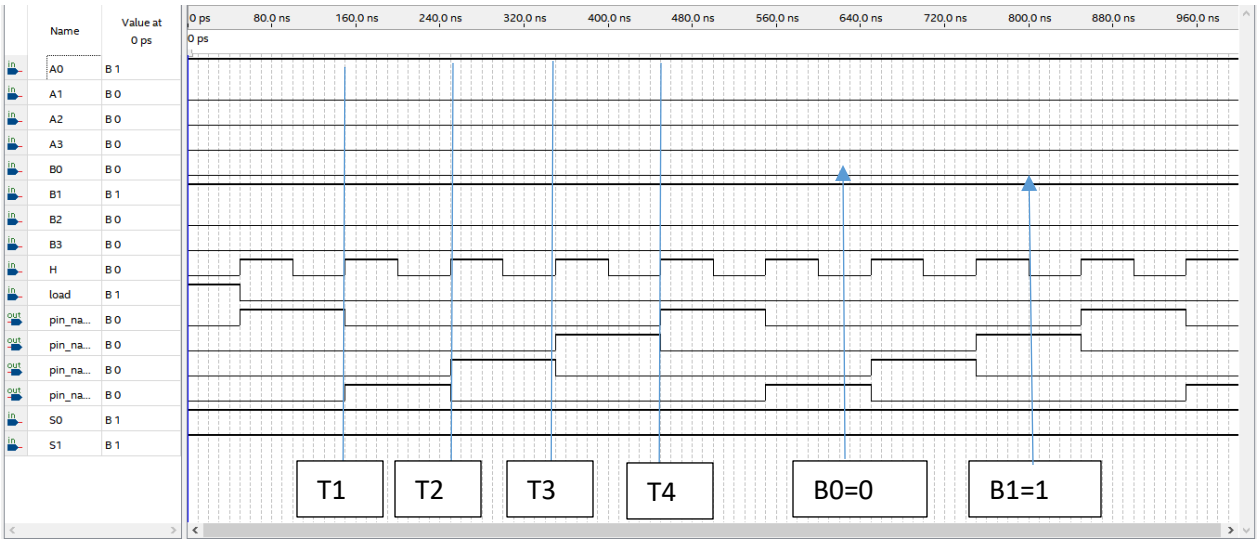
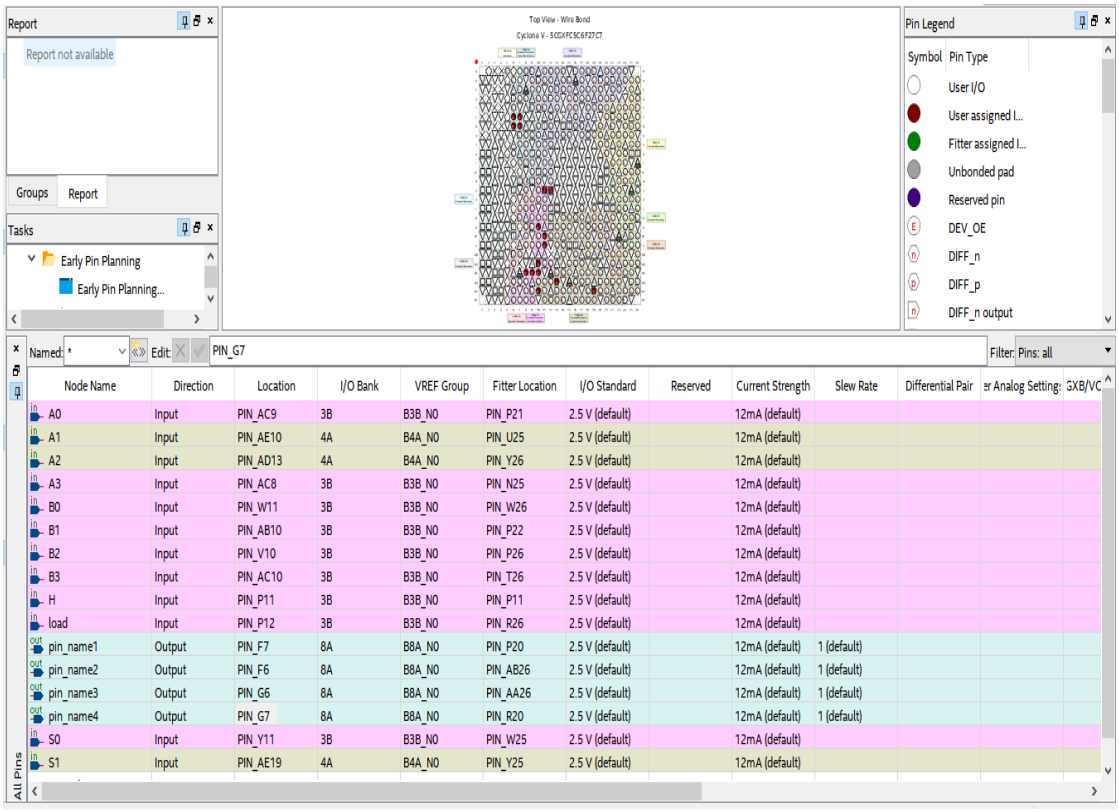


Figure 34: Simulation Décalage à gauche

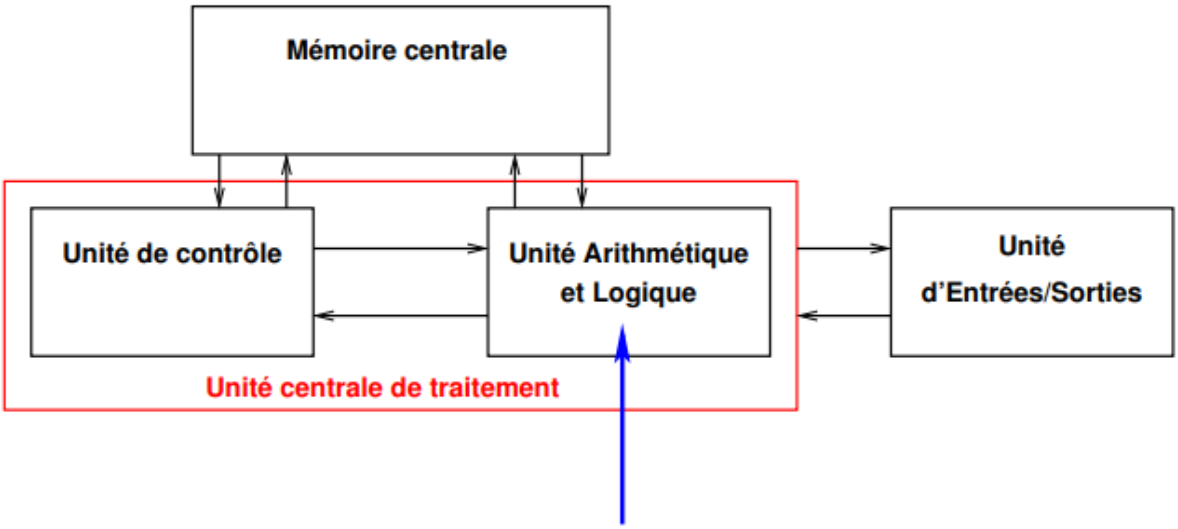
4. Implémentation



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair
A0	Input	PIN_AC9	3B	B3B_N0	PIN_P21	2.5 V (default)		12mA (default)		
A1	Input	PIN_AE10	4A	B4A_N0	PIN_U25	2.5 V (default)		12mA (default)		
A2	Input	PIN_AD13	4A	B4A_N0	PIN_Y26	2.5 V (default)		12mA (default)		
A3	Input	PIN_AC8	3B	B3B_N0	PIN_N25	2.5 V (default)		12mA (default)		
B0	Input	PIN_W11	3B	B3B_N0	PIN_W26	2.5 V (default)		12mA (default)		
B1	Input	PIN_AB10	3B	B3B_N0	PIN_P22	2.5 V (default)		12mA (default)		
B2	Input	PIN_V10	3B	B3B_N0	PIN_P26	2.5 V (default)		12mA (default)		
B3	Input	PIN_AC10	3B	B3B_N0	PIN_T26	2.5 V (default)		12mA (default)		
H	Input	PIN_P11	3B	B3B_N0	PIN_P11	2.5 V (default)		12mA (default)		
load	Input	PIN_P12	3B	B3B_N0	PIN_R26	2.5 V (default)		12mA (default)		
pin_name1	Output	PIN_F7	8A	B8A_N0	PIN_P20	2.5 V (default)		12mA (default)	1 (default)	
pin_name2	Output	PIN_F6	8A	B8A_N0	PIN_AB26	2.5 V (default)		12mA (default)	1 (default)	
pin_name3	Output	PIN_G6	8A	B8A_N0	PIN_AA26	2.5 V (default)		12mA (default)	1 (default)	
pin_name4	Output	PIN_G7	8A	B8A_N0	PIN_R20	2.5 V (default)		12mA (default)	1 (default)	
S0	Input	PIN_Y11	3B	B3B_N0	PIN_W25	2.5 V (default)		12mA (default)		
S1	Input	PIN_AE19	4A	B4A_N0	PIN_Y25	2.5 V (default)		12mA (default)		

Figure 35: Implémentation de la 2ème unité

VII. Conclusion :



On a construit l'UAL en utilisant des circuits combinatoires.

MERCI POUR VOTRE ATTENTION !