



INF1500

Logique des systèmes numériques

Laboratoire 3

Soumis par:

Berbiche, Yacine - 2217947

Anis, Menouar - 2247873

Date : 14 mars 2023

1- Description du système (2 points)

a) Description fonctionnelle

Décrivez succinctement le système conçu durant le laboratoire. Décrire les différents modules qui le composent et comment sont-ils reliés entre eux si pertinent. Vous donnerez aussi une explication brève du fonctionnement du système.

Le système à réaliser comporte une entrée **A** à 4 bits, le module **GEST_AFFICHAGE** déjà présent dans le fichier logisim, le module **BIN_GRAY**, le module **CRYPTO**, le module **COMPL2** et finalement le module **BIN_7SEG**. En effet, l'entrée **A** sera connectée à la première entrée du **GEST_AFFICHAGE**. Après cela, l'entrée **A** sera aussi connectée au module **CRYPTO**, au module **BIN_GRAY** ainsi qu'au module **COMPL2**. Ces derniers seront respectivement connectés aux entrées 2, 3 et 4 du **GEST_AFFICHAGE**. Puis, une entrée nommée **CLK** sur un bit qui jouera le rôle d'un Clock sera connectée à la dernière entrée du module fourni. Finalement, la sortie **AN** ainsi que la sortie **S** qui sont sur 8 bits seront le résultat. En ce qui concerne la sortie **S**, cette dernière sera par ailleurs connectée au module **BIN_7SEG** qui finit par la sortie **SEVEN_SEG** qui sera sur 8 bits.

b) Circuit Module BIN_Gray :

Tableau 01 : table de vérité pour le module BIN_Gray

Code Décimal	Code Hexadécimal	Code Binaire				Code Gray			
		E ₃	E ₂	E ₁	E ₀	S ₃	S ₂	S ₁	S ₀
0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0	1
2	2	0	0	1	0	0	0	1	1
3	3	0	0	1	1	0	0	1	0
4	4	0	1	0	0	0	1	1	0
5	5	0	1	0	1	0	1	1	1
6	6	0	1	1	0	0	1	0	1
7	7	0	1	1	1	0	1	0	0
8	8	1	0	0	0	1	1	0	0
9	9	1	0	0	1	1	1	0	1
10	A	1	0	1	0	1	1	1	1
11	B	1	0	1	1	1	1	1	0
12	C	1	1	0	0	1	0	1	0
13	D	1	1	0	1	1	0	1	1
14	E	1	1	1	0	1	0	0	1
15	F	1	1	1	1	1	0	0	0

Tableau 02 : table de Karnaugh pour le bit 0 de la sortie S (S₀)

	E ₁ E ₀			
	00	01	11	10

E_3E_2	00	0	1	0	1
	01	0	1	0	1
	11	0	1	0	1
	10	0	1	0	1

➤ Équation du bit 0 de la sortie S :

$$S_0 = E_1'E_0 + E_1E_0'$$

$$S_0 = E_1 \oplus E_0$$

Tableau 03 : table de Karnaugh pour le bit 1 de la sortie S (S_1)

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

➤ Équation du bit 1 de la sortie S :

$$S_1 = E_2'E_1 + E_2E_1'$$

$$S_1 = E_2 \oplus E_1$$

Tableau 04 : table de Karnaugh pour le bit 2 de la sortie S (S_2)

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

➤ Équation du bit 2 de la sortie S :

$$S_2 = E_3'E_2 + E_3E_2'$$

$$S_2 = E_3 \oplus E_2$$

Tableau 05 : table de Karnaugh pour le bit 3 de la sortie S (S_3)

	E_1E_0
--	----------

		E ₁ E ₀			
		00	01	11	10
E ₃ E ₂	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	1	1	1	1

➤ Équation du bit 3 de la sortie S :

$$S_3 = E_3$$

c) Circuit Module CRYPTO

Tableau 06 : table de vérité pour le module CRYPTO

Entrée (E)				Sortie (S)			
E ₃	E ₂	E ₁	E ₀	S ₃	S ₂	S ₁	S ₀
0	0	0	0	0	1	0	0
0	0	0	1	X	0	0	1
0	0	1	0	X	X	X	0
0	0	1	1	X	0	0	1
0	1	0	0	X	0	X	1
0	1	0	1	X	X	0	0
0	1	1	0	X	0	1	X
0	1	1	1	X	1	0	0
1	0	0	0	X	1	0	X
1	0	0	1	X	0	1	X
1	0	1	0	X	1	0	X
1	0	1	1	X	0	0	X
1	1	0	0	X	0	1	X
1	1	0	1	X	1	1	X
1	1	1	0	X	X	1	X
1	1	1	1	X	1	X	X

Tableau 08 : table de Karnaugh pour le bit 0 de la sortie S (S₀)

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

➤ Équation du bit 0 de la sortie S :

$$S_0 = E_2'E_0 + E_2E_0'$$

$$S_0 = E_2 \oplus E_0$$

Tableau 09 : table de Karnaugh pour le bit 1 de la sortie S (S_1)

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

➤ Équation du bit 1 de la sortie S :

$$S_1 = E_3E_1'E_0 + E_2E_0'$$

Tableau 10 : table de Karnaugh pour le bit 2 de la sortie S (S_2)

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

➤ Équation du bit 2 de la sortie S :

$$S_2 = E_2'E_0' + E_2E_0$$

$$S_2 = E_3 \otimes E_2$$

Tableau 11 : table de Karnaugh pour le bit 3 de la sortie S (S_3)

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

➤ Équation du bit 3 de la sortie S :

$$S_3 = 0$$

d) Circuit le module COMPL2Tableau 12 : table de vérité pour le module COMPL2

Entrée (E)				Sortie (S)			
E ₃	E ₂	E ₁	E ₀	S ₃	S ₂	S ₁	S ₀
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	1	0	1	1	1	0
0	0	1	1	1	1	0	1
0	1	0	0	1	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

Tableau 13 : table de Karnaugh pour le bit 0 de la sortie S (S₀)

		E ₁ E ₀			
		00	01	11	10
E ₃ E ₂	00	0	1	1	0
	01	0	1	1	0
	11	0	1	1	0
	10	0	1	1	0

➤ Équation du bit 0 de la sortie S :

$$S_0 = E_0$$

Tableau 14 : table de Karnaugh pour le bit 1 de la sortie S (S₁)

		E ₁ E ₀			
		00	01	11	10
E ₃ E ₂	00	0	1	0	1
	01	0	1	0	1
	11	0	1	0	1
	10	0	1	0	1

➤ Équation du bit 1 de la sortie S :

$$S_1 = E_1'E_0 + E_1E_0'$$

Tableau 15 : table de Karnaugh pour le bit 2 de la sortie S (S_2)

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

➤ Équation du bit 2 de la sortie S :

$$S_2 = E_2E_1'E_0' + E_2'E_0 + E_2'E_1$$

Tableau 16 : table de Karnaugh pour le bit 3 de la sortie S (S_3)

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

➤ Équation du bit 3 de la sortie S :

$$S_3 = E_3'E_2 + E_3'E_0 + E_3'E_1 + E_3E_2'E_1'E_0'$$

e) Circuit du module BIN_7SEG :

Tableau 17: Table de vérité pour le module AFFICHEUR_7SEG

E3	E2	E1	E0	S7	S6	S5	S4	S3	S2	S1	S0
0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	0
0	0	1	1	1	0	1	1	0	0	0	0
0	1	0	0	1	0	0	1	1	0	0	1
0	1	0	1	1	0	0	1	0	0	1	0
0	1	1	0	1	0	0	0	0	0	1	0

0	1	1	1	1	0	1	1	1	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	1	0	0	0	0

Tables de Karnaugh et équations pour chaque bit de la sortie :

➤ Bit 7 :

$$S_7 = 1$$

		E ₁ E ₀			
		00	01	11	10
E ₃ E ₂	00	1	1	1	1
	01	1	1	1	1
	11	1	1	1	1
	10	1	1	1	1

➤ Bit 6 :

$$S_6 = E_3'E_2'E_1' + E_3E_2E_1'E_0' + E_3'E_2E_1E_0$$

		E ₁ E ₀			
		00	01	11	10
E ₃ E ₂	00	1	1	0	0
	01	0	0	1	0
	11	1	0	0	0
	10	0	0	0	0

➤ Bit 5 :

$$S_5 = E_3'E_2'E_0 + E_3'E_2'E_1 + E_3'E_1E_0 + E_3E_2E_1'E_0$$

		E ₁ E ₀			
		00	01	11	10
E ₃ E ₂	00	0	1	1	1
	01	0	0	1	0
	11	0	1	0	0
	10	0	0	0	0

➤ Bit 4 :

$$S_4 = E_3'E_0 + E_3'E_2E_1' + E_2'E_1'E_0$$

		E ₁ E ₀			
		00	01	11	10
E ₃ E ₂	00	0	1	1	0

	01	1	1	1	0
	11	0	0	0	0
	10	0	1	0	0

➤ Bit 3 :

$$S_3 = E_3'E_2E_1'E_0' + E_3'E_2'E_1'E_0 + E_2E_1E_0 + E_3E_2'E_1E_0'$$

		E ₁ E ₀			
		00	01	11	10
E ₃ E ₂	00	0	1	0	0
	01	1	0	1	0
	11	0	0	1	0
	10	0	0	0	1

➤ Bit 2 :

$$S_2 = E_3E_2E_1 + E_3'E_2'E_1E_0' + E_3E_2E_0'$$

		E ₁ E ₀			
		00	01	11	10
E ₃ E ₂	00	0	0	0	1
	01	0	0	0	0
	11	1	0	1	1
	10	0	0	0	0

➤ Bit 1 :

$$S_1 = E_3'E_2E_1'E_0 + E_3E_2E_0' + E_2E_1E_0' + E_3E_1E_0$$

		E ₁ E ₀			
		00	01	11	10
E ₃ E ₂	00	0	0	0	0
	01	0	1	0	1
	11	1	0	1	1
	10	0	0	1	0

➤ Bit 0 :

$$S_0 = E_3'E_2E_1'E_0' + E_3'E_2'E_1'E_0 + E_3E_2E_1'E_0 + E_3E_2'E_1E_0$$

		E ₁ E ₀			
		00	01	11	10
E ₃ E ₂	00	0	1	0	0
	01	1	0	0	0
	11	0	1	0	0

	10	0	0	1	0
--	----	---	---	---	---

f) Schémas du circuit :

Schéma 01 : schéma du module BIN_GRAY

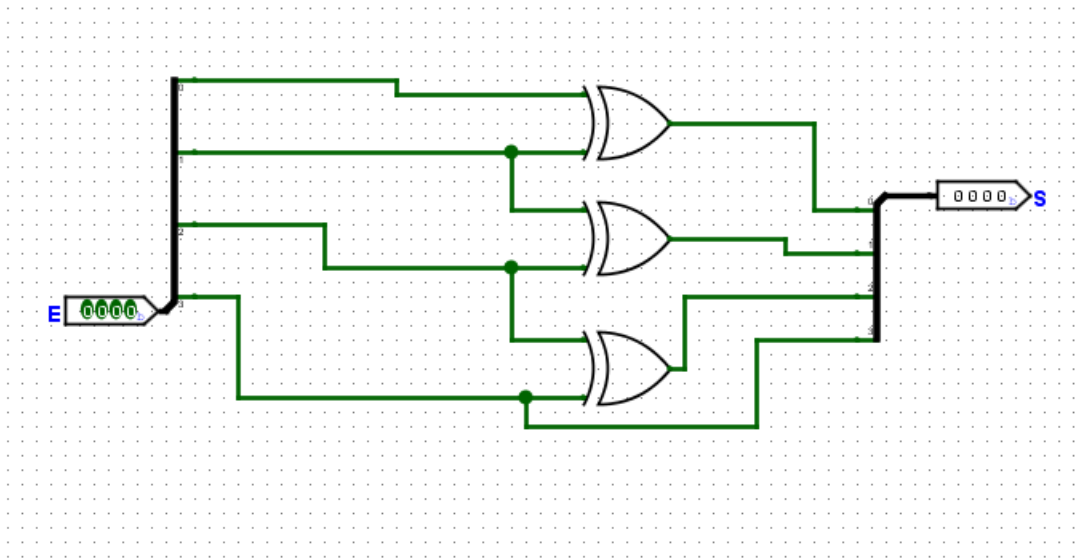


Schéma 02 : schéma du module CRYPTO

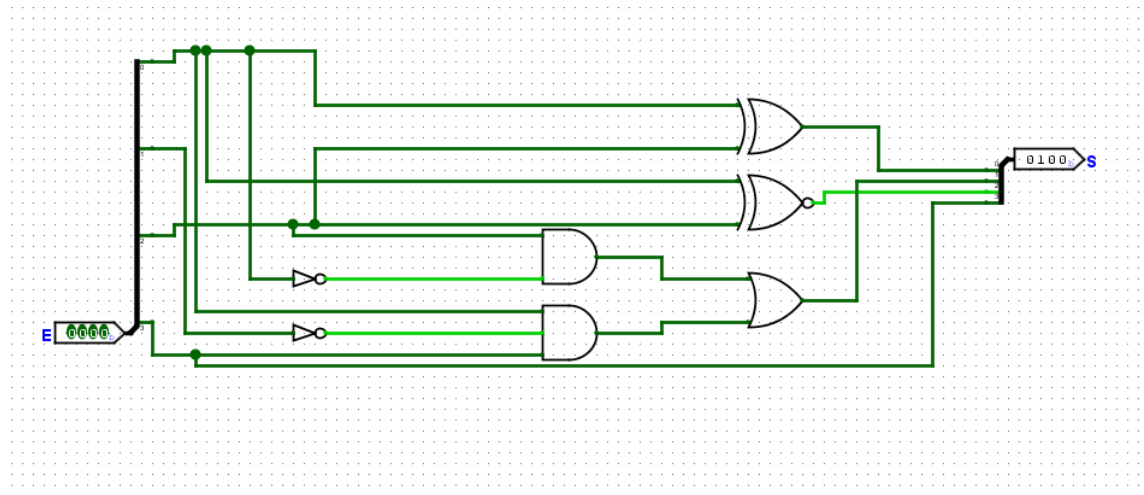


Schéma 03 : schéma du module COMPL2

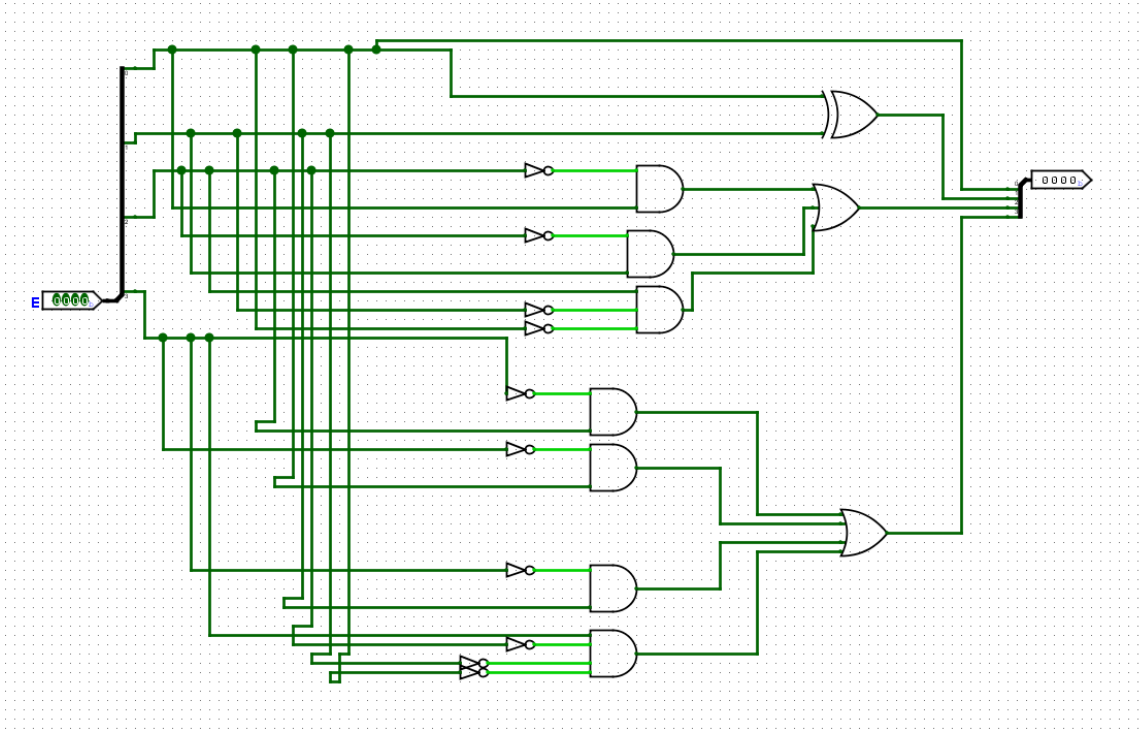


Schéma 04 : schéma du module afficheur_7seg

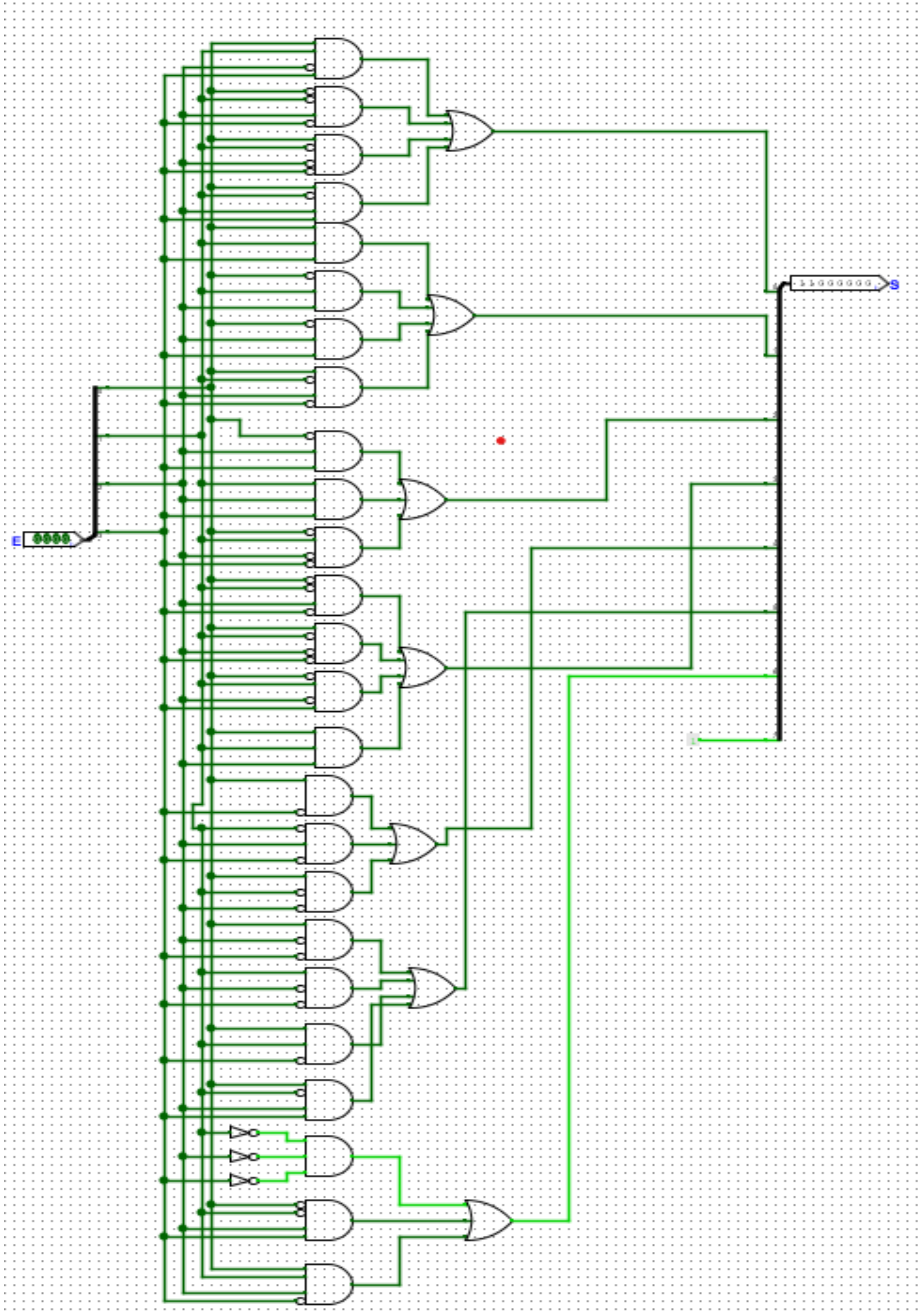
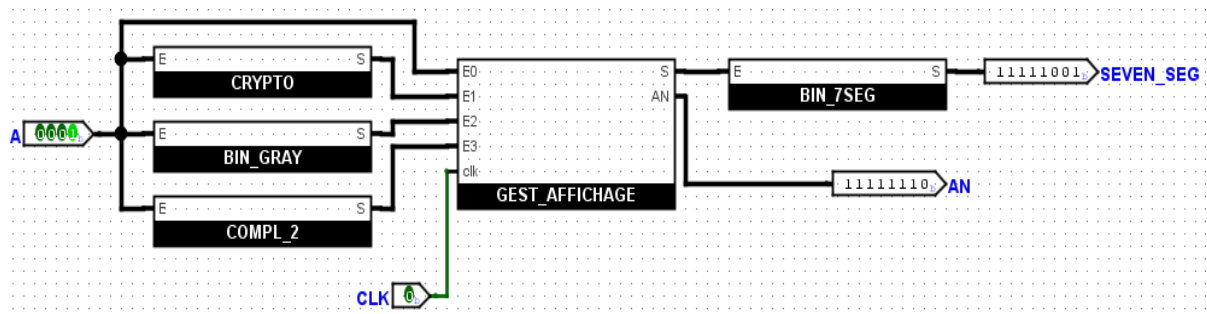


Schéma 05 : schéma du circuit intégral



2- Vérification du système (3 points)

a) Choix de validation

Expliquez comment vous avez validé votre système (test exhaustif ou non exhaustif). Dans le cas d'un test non exhaustif, vous devez justifier vos choix de tests. Si votre système comporte plusieurs modules, vous devez montrer comment vous avez validé chaque module avant de valider le système complet.

Pour valider les quatre modules conçus, on a utilisé la fonction Force Clock présente dans le logiciel VIVADO.

BIN GRAY :

Le test que nous avons réalisé est un test exhaustif effectué par la simulation dont les résultats figurent dans la section **b** de « Vérification du système ». Ce test prend en compte toutes les possibilités des entrées. Toutes ces possibilités figurent dans la table de vérité et les tables de Karnaugh présentées plus haut.

CRYPTO :

Le test que nous avons réalisé est un test exhaustif effectué par la simulation dont les résultats figurent dans la section **b** de « Vérification du système ». Ce test prend en compte toutes les possibilités des entrées. Toutes ces possibilités figurent dans la table de vérité et les tables de Karnaugh présentées plus haut.

COMPL2 :

Le test que nous avons réalisé est un test exhaustif effectué par la simulation dont les résultats figurent dans la section **b** de « Vérification du système ». Ce test prend en compte toutes les possibilités des entrées. Toutes ces possibilités figurent dans la table de vérité et les tables de Karnaugh présentées plus haut.

BIN 7SEG :

Le test que nous avons réalisé est un test exhaustif effectué par la simulation dont les résultats figurent dans la section **b** de « Vérification du système ». Ce test prend en compte toutes les possibilités des entrées. Toutes ces possibilités figurent dans la table de vérité et les tables de Karnaugh présentées plus haut.

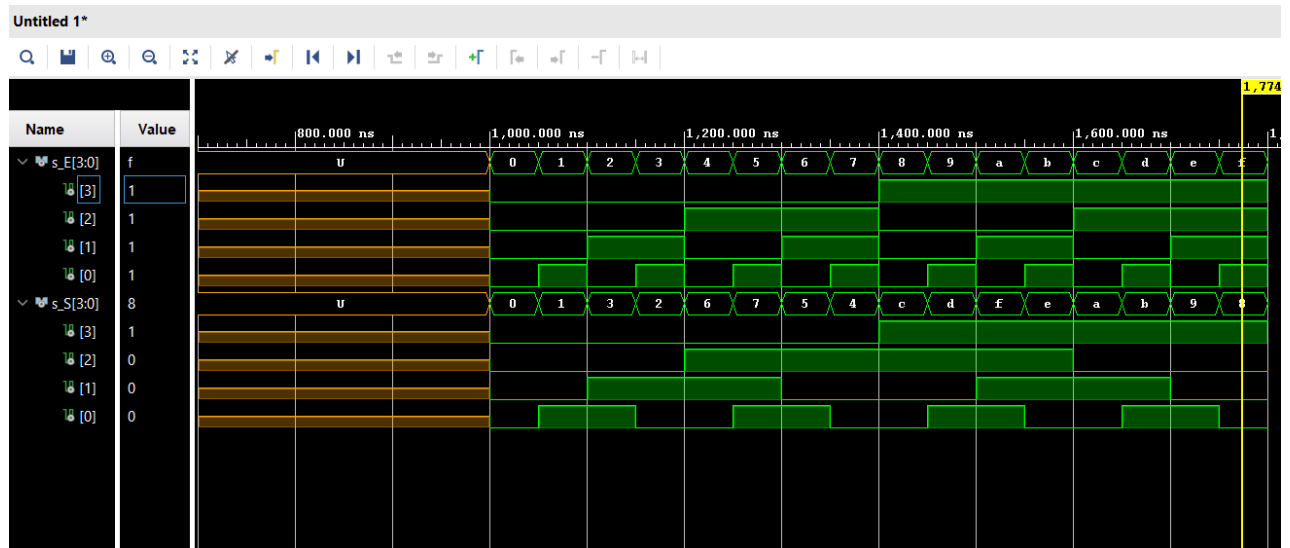
CIRCUIT INTÉGRAL :

Le test que nous avons réalisé est un test exhaustif effectué par la simulation dont les résultats figurent dans la section **b** de « Vérification du système ». Ce test prend en compte toutes les possibilités des entrées de l'entrée **A**.

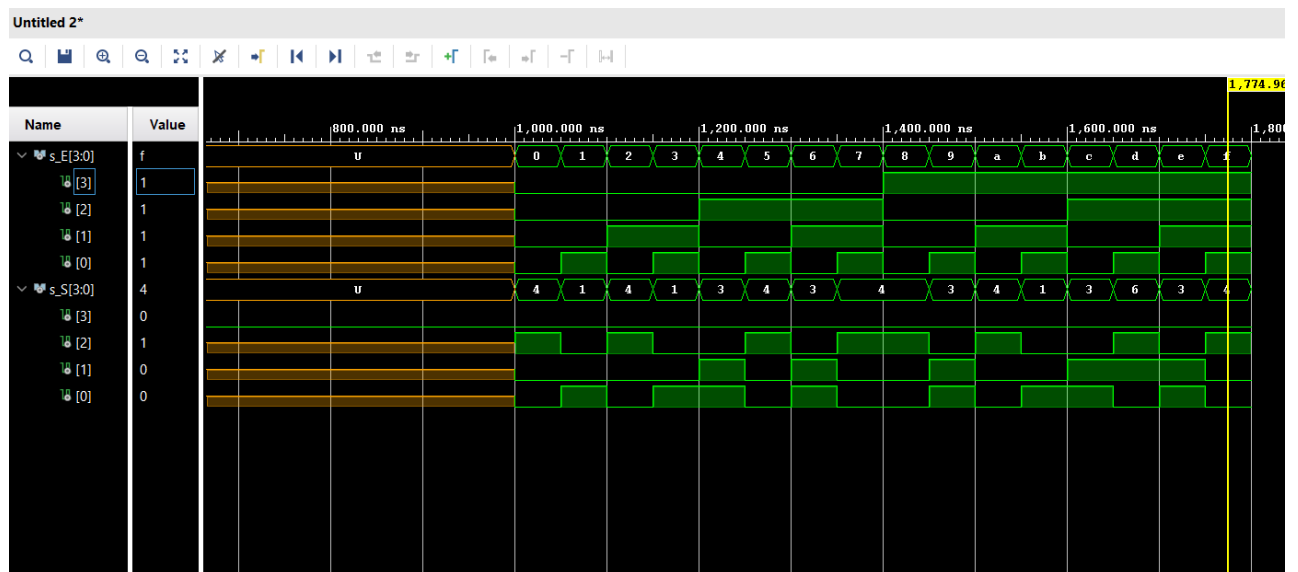
b) Simulations

Vous pourrez mettre des captures d'écran de vos simulations ici.

Simulation 01 : module BIN_GRAY

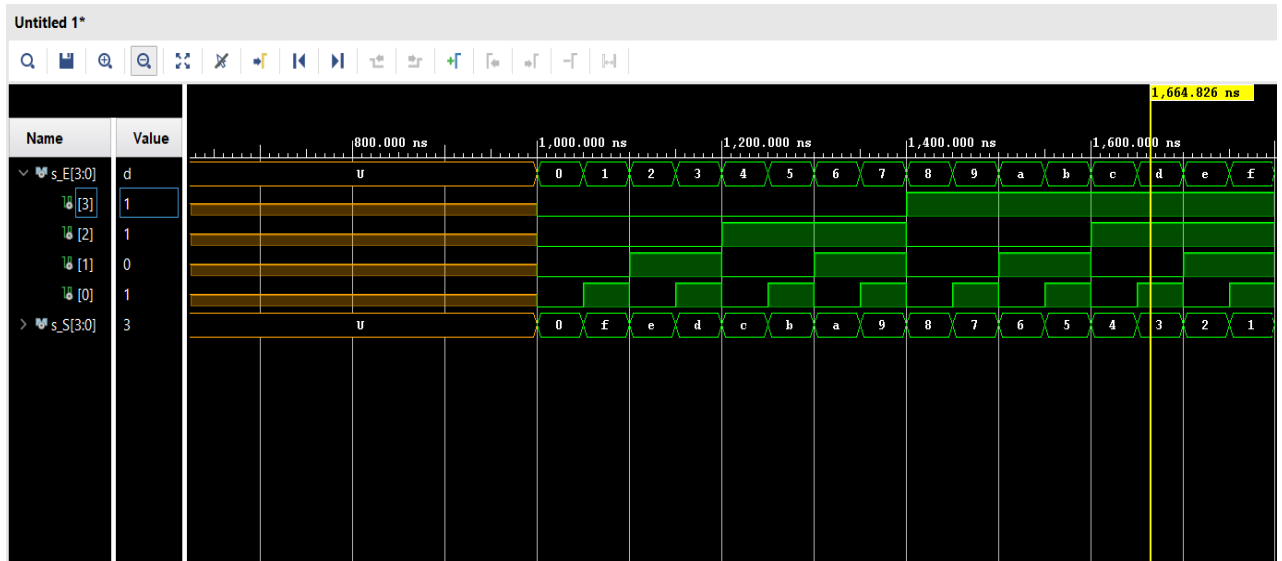


Simulation 02 : module CRYPTO

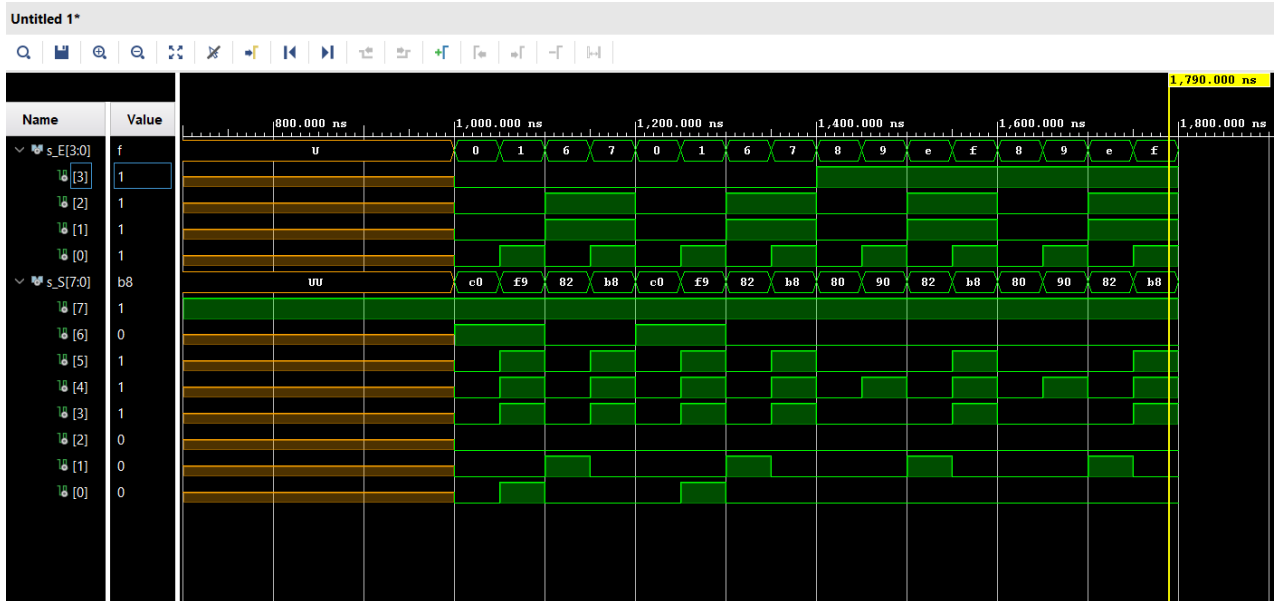


Simulation 03 : module COMPL2

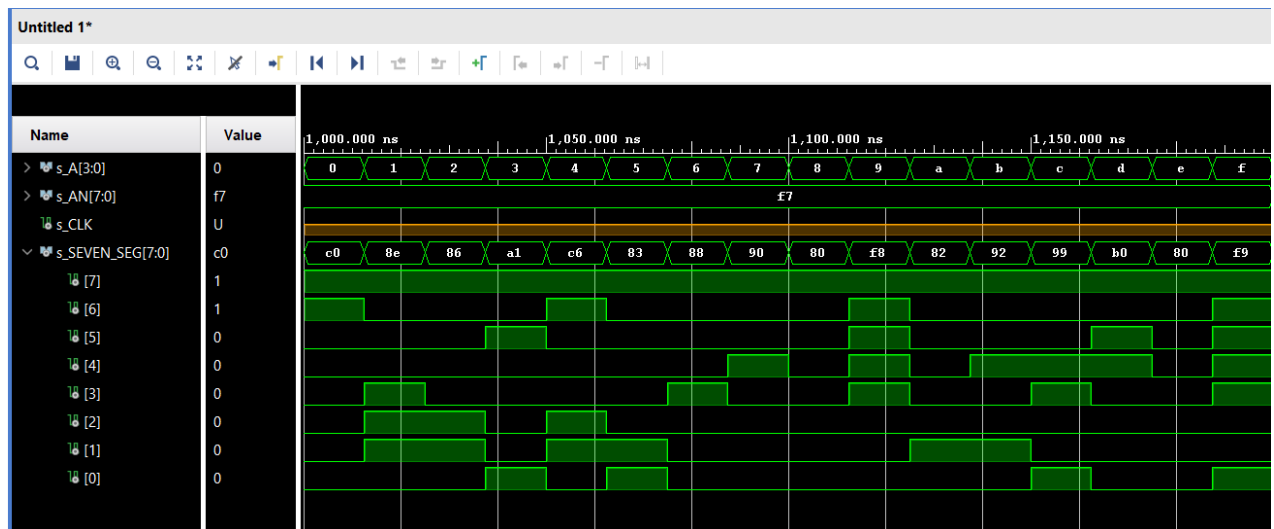
INF1500 - Logique des systèmes numériques



Simulation 04 : module BIN_7SEG



Simulation 05 : Circuit intégral



c) Observations et validation

Expliquez pourquoi vous pouvez affirmer que votre système fonctionne.

Nous pouvons confirmer que les résultats obtenus grâce aux différentes simulations déjà faites pour chaque module, notamment le module **BIN_GRAY**, **CRYPTO**, **COMPL2**, **BIN_7SEG** et celui du **CIRCUIT_INTÉGRAL**, correspondent bel et bien à ceux qui sont attendus et ceux présents dans les tables des valeurs ainsi que les résultats des tables de Karnaugh correspondants pour chaque circuit. Nous concluons donc la validité de notre circuit.

3- Réponses aux questions (1.5 points)

Si des questions spécifiques vous sont posées dans l'énoncé du laboratoire, veuillez y répondre ici.

Question 1:

Est-ce que la conversion Gray vers binaire serait la même fonction que la conversion de binaire vers Gray ? Justifiez.

Non, ces deux conversions n'ont pas la même fonction. Pour la conversion du code Gray vers le code binaire, on commence par garder le bit le plus significatif et on additionne chaque bit avec le bit situé à sa droite dans le code Gray de départ. Prenons comme exemple, le code gray **1111**. On commence par garder le bit le plus significatif et on l'additionne au bit qui se situe à sa droite dans le code gray ($1+1$). Cette addition qui est sur 1 bit nous donne **0** (correspond au 2e bit). Ensuite, j'additionne ce bit au bit situé à sa droite dans le code Gray ($1+0 = 1$), on obtient **1** (correspond au 3e bit). Pour finir, on additionne ce bit avec le bit situé à sa droite ($1+1 = 0$). On obtient finalement **1010**. Concernant la conversion de binaire vers Gray, on garde toujours le bit le plus significatif. Ensuite, si on veut connaître le bit n dans le code gray, il suffit d'appliquer un **XOR** sur le $n-1$ bit et le n ème bit du code binaire. Par exemple, pour trouver le 2e bit du code binaire, il suffit d'appliquer un XOR sur le premier et le deuxième bit du code binaire. Lorsqu'on

applique XOR sur deux bits, s'ils sont différents, le bit dans le code gray sera 1, autrement le bit sera 0.

Question 2:

Que se passe-t-il si nous appliquons deux fois de suite l'opération de complément à 2 à une valeur ? Donnez un exemple.

On obtient la même valeur de départ. Prenons comme valeur de départ 1001. Si on applique une fois le complément à 1, on obtient 0110 et en ajoutant 1 pour obtenir le complément à 2, on trouve 0111. Si l'on applique une deuxième fois le complément à 2, on commence par faire le complément à 1, on obtient 1000 et en ajoutant 1 à cette valeur pour trouver le complément à 2, on arrive à 1001, c'est-à-dire la valeur de départ.

Question 3:

Pourquoi utiliser des X (Don't care) dans les circuits ? Quel est l'impact sur l'équation d'un circuit ? (Utiliser comme exemple une des sorties du module CRYPTO)

Utiliser les X ou les Don't care pour toutes les entrées dont les valeurs n'affectent pas le résultat de la sortie. Autrement dit, si l'entrée a un signal haut (**1**) ou un signal bas (**0**), cela n'entraîne aucune conséquence sur la valeur de la sortie. Par exemple, dans le module **CRYPTO** on a forcé la quatrième sortie à une **constante 0** car cela n'affecte pas le résultat de la sortie. Par ailleurs on pourrait attribuer $E_0 + E_1 + E_2 + E_3$ pour le dernier bit de la sortie **S**, donc **S**₃ reçoit un signal bas quand les quatre bits de l'entrée ont **0** comme valeur, autre que ça, reçoit un signal haut (**1**) et cela n'affecte pas la sortie **S**.

Clarté et forme du rapport (0.5 points)

(1 point sera retiré si le rapport n'est pas rendu sous le format PDF)