



Le génie pour l'industrie

**Conception des systèmes numériques (ELE140-01)**

**Laboratoire 2**

**Remis à :** Myiah Catwell

**Remis par :**

Ourania Voyatzis (VOYO78260401)  
Jhermain Louis-Jean (LOUJ67360401)

**École de technologie supérieure**

**Date :** 12 novembre 2025

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Cellule-type</b>	<b>2</b>
2.1	La définition générale d'une cellule-type . . . . .	2
2.2	La définition spécifique de la cellule-type pour comparaison . . . . .	2
<b>3</b>	<b>Schéma fonctionnel du comparateur</b>	<b>3</b>
3.1	La définition des signaux inter cellules . . . . .	3
3.2	La définition des signaux inter cellules $X_{out}$ et $Y_{out}$ . . . . .	3
3.3	Comparateur . . . . .	3
<b>4</b>	<b>Table de vérité pour la cellule-type</b>	<b>3</b>
4.1	La définition générale d'une table de vérité . . . . .	3
4.2	Table de vérité . . . . .	4
<b>5</b>	<b>Tables de Karnaugh : <math>X_{out}</math> et <math>Y_{out}</math></b>	<b>4</b>
5.1	Définition . . . . .	4
5.2	K-Maps : $X_{out}$ et $Y_{out}$ . . . . .	5
<b>6</b>	<b>Résultat et discussion</b>	<b>5</b>
6.1	Chronogramme de cellule-type . . . . .	5
6.2	Chronogramme de Test Bench . . . . .	6
6.3	Fichier data.out . . . . .	6
6.4	Résultat implémentation matérielle . . . . .	6
<b>7</b>	<b>Conclusion</b>	<b>7</b>

## 1 Introduction

Dans le cadre de cette laboratoire, la tâche consiste à créer un circuit itératif capable d'effectuer une comparaison bit à bit entre deux entiers signés de 4 bits ( $A[3..0]$  et  $B[3..0]$ ) codés en complément à deux. Le cahier des charges nous autorise à utiliser et à définir deux signaux de retenue et à créer (si nécessaire) une fonction de sortie finale qui ne doit comporter que trois signaux de sortie : PP ( $A < B$ ), PG ( $A > B$ ), EG ( $A = B$ ).

## 2 Cellule-type

### 2.1 La définition générale d'une cellule-type

Une cellule-type est un ensemble de fonctions booléennes combinatoires, ou une seule fonction, destinée à être itérée et instanciée plusieurs fois, puis inter-connectée à l'aide de signaux inter cellule (copy-paste). Ces fonctions logiques identiques et chaînées permettent de réaliser des opérations multibits plus complexes, capables de modifier la sortie finale en fonction de l'état/l'index précédent des entrées. Ces signaux inter cellules agissent comme des sorties lors d'une itération et comme des entrées lors de l'itération suivante; il s'agit, en quelque sorte, d'une manière très rudimentaire et inefficace d'implémenter une « mémoire » d'entrée primaires.

Un exemple de mise en œuvre de ce type de logique est l'additionneur « carry-lookahead », dont les entrées principales sont deux chiffres binaires de même indice et dont les signaux de sortie sont  $P_i$  et  $G_i$ . Ces signaux indiquent si la cellule génère et /ou propage une retenue en fonction des entrées principales des indices précédents et actuels.

### 2.2 La définition spécifique de la cellule-type pour comparaison

La définition de la cellule générique capable d'itérer sur un nombre quelconque de bits et de comparer deux nombres codés en complément à deux, à condition que la séquence commence par le bit de poids fort (MSB). Cette définition de cellule est soumise à certaines conditions :

- Nous supposons que  $A$  et  $B$  étaient égaux juste avant le démarrage de la machine ( $A=B=Z$ ).
- Si  $A$  et  $B$  sont égaux et l'ont toujours été, alors  $X_{out} \neq Y_{out}$ .
- Si  $A$  vaut 1 et  $B$  vaut 0 sur le premier bit (bit de signe), pour cette itération et toutes les itérations suivantes,  $X_{out} = Y_{out} = 0$  ( $A < B \Rightarrow 0 = X_{out} = Y_{out}$ ).
- Si  $A$  vaut 1 et  $B$  vaut 0 sur un autre bit que le premier, pour cette itération et toutes les itérations suivantes,  $X_{out} = Y_{out} = 1$  ( $A > B \Rightarrow 1 = X_{out} = Y_{out}$ ).
- Si, pour les deux conditions ci-dessus,  $B$  est interverti avec  $A$  et vice versa, alors la sortie est inversée.

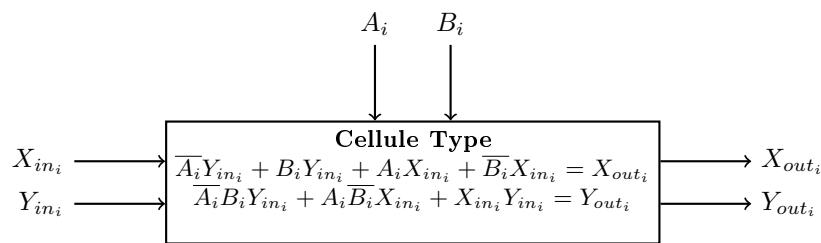


FIGURE 1 – La  $i$ -ième cellule-type

### 3 Schéma fonctionnel du comparateur

#### 3.1 La définition des signaux inter cellules

Les signaux intercellulaires sont les signaux qui circulent entre les cellules répétées ; ils transportent des informations codées sur les entrées primaires des cellules précédentes. La signification de chaque signal varie en fonction de l'architecture de la cellule de base répétée.

#### 3.2 La définition des signaux inter cellules $X_{out}$ et $Y_{out}$

$X_{in}$  et  $Y_{in}$  doivent être initialisés respectivement à 0 et 1. En effet, nous devons supposer que  $A_{N+1}$  et  $B_{N+1}$  seront égaux et à haute impédance au démarrage de la machine.(INIT /SIGN BIT)

Les signaux intercellulaires du circuit conçu représentent cumulativement si A est supérieur, inférieur ou égal à B. Comme spécifié dans la table de vérité ci-dessus, lorsque les signaux intercellulaires  $X_{out}$  et  $Y_{out}$  sont égaux à 11, cela signifie  $A > B$ . Inversement, lorsqu'ils sont égaux à 00,  $A < B$ . Lorsque  $X_{out}$  et  $Y_{out}$  sont différents ( $X_{out} \neq Y_{out}$ ), cela signifie que jusqu'à présent, tous les bits précédents de A et B ont été égaux, en commençant par MSB. Si cet état est maintenu jusqu'au dernier bit (LSB), alors les deux nombres A et B sont exactement identiques.

#### 3.3 Comparateur

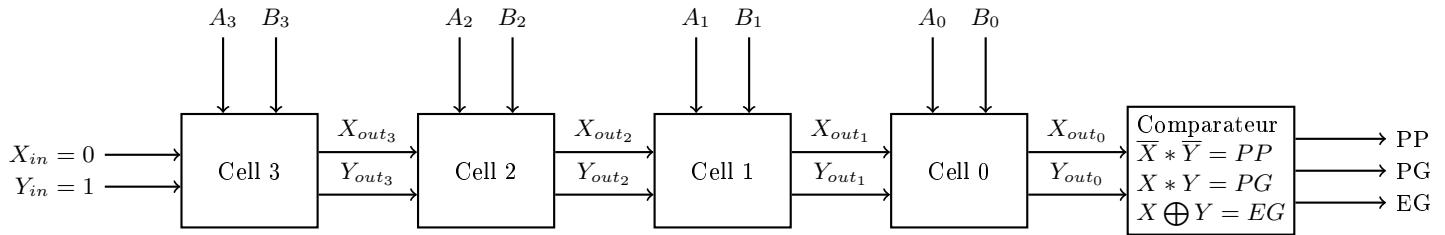


FIGURE 2 – Schéma fonctionnel du comparateur

### 4 Table de vérité pour la cellule-type

#### 4.1 La définition générale d'une table de vérité

Une table de vérité est une méthode permettant de coder et de visualiser des fonctions logiques. Elle consiste à modifier progressivement les entrées et à afficher la ou les sorties correspondantes.

## 4.2 Table de vérité

FIGURE 3 – Table de vérité pour la cellule-type :  $X_{out}$  et  $Y_{out}$

	No.	$X_{in}$	$Y_{in}$	$A_i$	$B_i$	$X_{out}$	$Y_{out}$
Set $A < B$	0	0	0	0	0	0	0
	1	0	0	0	1	0	0
	2	0	0	1	0	0	0
	3	0	0	1	1	0	0
Bit de signe/init	4	0	1	0	0	1	0
	5	0	1	0	1	1	1
	6	0	1	1	0	0	0
	7	0	1	1	1	1	0
Bits suivants	8	1	0	0	0	1	0
	9	1	0	0	1	0	0
	10	1	0	1	0	1	1
	11	1	0	1	1	1	0
Set $A > B$	12	1	1	0	0	1	1
	13	1	1	0	1	1	1
	14	1	1	1	0	1	1
	15	1	1	1	1	1	1

## 5 Tables de Karnaugh : $X_{out}$ et $Y_{out}$

### 5.1 Définition

Les tableaux de Karnaugh, ou cartes de Karnaugh, constituent une autre méthode de visualisation et de codage des fonctions logiques. Au lieu de modifier progressivement les entrées selon la méthode binaire standard, chaque ligne et chaque colonne incrémentent un groupement d'entrées à l'aide du code de Gray. Les entrées sont divisées en deux groupes, généralement et plus facilement simplifiables si ces groupes sont de taille égale. Ces deux groupes sont répartis le long d'un axe vertical et d'un axe horizontal, l'incrémentation étant effectuée à l'aide du code de Gray, créant ainsi un cadre carré doté d'un système de coordonnées représentant les valeurs d'entrée. Chaque cellule est ensuite remplie avec sa valeur de sortie correspondante, basée sur la valeur des groupements d'entrées codés en code de Gray aux coordonnées spécifiques.

Cette méthode, bien qu'intuitivement déroutante au premier abord, permet de simplifier facilement les fonctions sous leurs formes somme de produits/produit de sommes en regroupant respectivement tous les 1 ou tous les 0 sur le tableau. Les cartes de Karnaugh permettent même de corriger très rapidement les aléas statiques dans les circuits, avant même leur conception ou leur schématisation.

## 5.2 K-Maps : $X_{out}$ et $Y_{out}$

FIGURE 4 –  $X_{out} = \overline{A}Y + BY + AX + \overline{B}X$

$\overline{AB}$	$\overline{A}B$	$AB$	$A\overline{B}$
$X\overline{B}$	XY	$\overline{XY}$	$XY$
0	0	0	0
1	1	1	0
1	1	1	1
1	0	1	1

FIGURE 5 –  $Y_{out} = \overline{A}BY + A\overline{B}X + XY$

$\overline{AB}$	$\overline{A}B$	$AB$	$A\overline{B}$
$X\overline{Y}$	XY	$\overline{XY}$	$XY$
0	0	0	0
0	1	0	0
1	1	1	1
0	0	0	1

## 6 Résultat et discussion

Cette section sera sujet de nos résultats ainsi que de nos conclusions faisant référence à ceux-ci. Pour visualiser nos résultats, nous devions utiliser des chronogrammes afin de pouvoir illustrer l'évolution temporelle de nos entrées et sorties.

### 6.1 Chronogramme de cellule-type

Dans celui de la cellule type, généré avec Waveform, il faut qu'il soit représentatif de la table de vérité de la cellule-type. Étant donné le chronogramme mis ci-dessous, nous avons donc eu une bonne représentation graphique de notre table de vérité montrant les sorties Xout et Yout étant à 1.

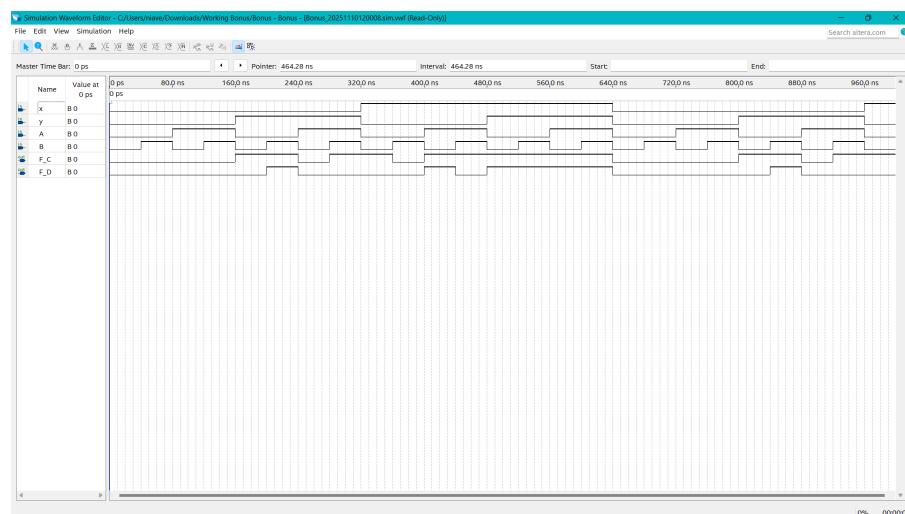


FIGURE 6  
Chronogramme de la cellule-type

## 6.2 Chronogramme de Test Bench

Dans le chronogramme du test Bench, il faut qu'il soit représentatif du circuit dans tout son entièreté. Il faut donc se fier aux équations de sorties trouver dans les tables de Karnaugh des signaux d'intercellules.

Nous voyons aussi dans le chronogramme les fonctionnements de comparaison :

- Le fonctionnement d'une comparaison  $+/+$  : Veux dire que l'on compare directement les bits A et B
- Le fonctionnement d'une comparaison  $-/-$  : Veux dire que l'on inverse la valeur de A et B et ensuite on les compare
- Le fonctionnement d'une comparaison  $+/-$  : Veux dire que la Valeur A ou B est inversé et puis on la compare à la valeur de l'autre bit

Ainsi, étant donné le chronogramme mis ci-dessous, nous avons donc eu une bonne représentation graphique de notre table de vérité montrant les sorties PGout, PPout et EGout.

Nous voyons le fonctionnement d'une comparaison  $+/+$ , qui veux dire que les valeurs de A et B sont égales et positives. Le fonctionnement d'une comparaison  $-/-$ , qui veux dire que les

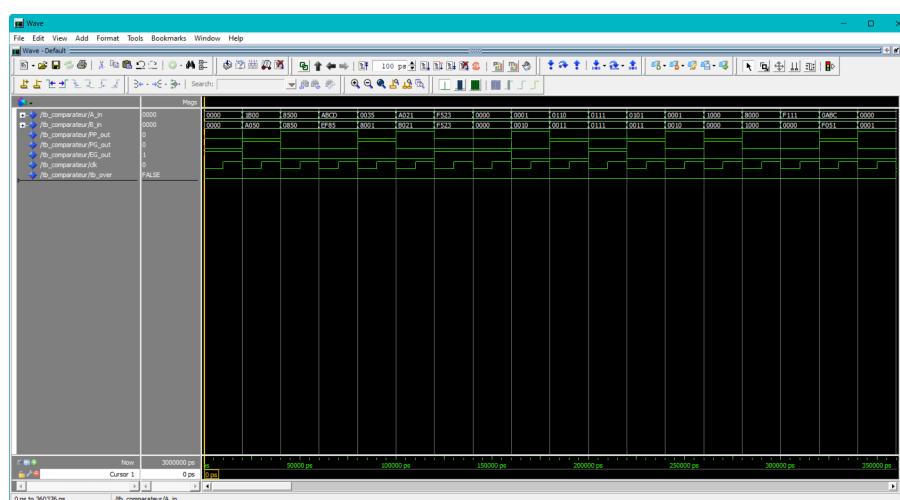


FIGURE 7  
Chronogramme du textbench

## 6.3 Fichier data.out

Le fichier data.out a été généré par un banc d'essai VHDL. Cependant, pour générer ce fichier, il faut tout d'abord compiler les fichiers VHDL de notre design en utilisant des scripts que nous avons complétés. Ensuite il faut lancer une simulation dans la fenêtre de commande dans le logiciel Modelsim. En comparant les entrées A et B avec les signaux intercellules, nous avons un résultat à la fin de chaque ligne qui nous indique si l'on a réussi. Dans notre cas, nous avons juste eu des réussites. Ceci veut dire que la procédure a fonctionné.

## 6.4 Résultat implémentation matérielle

Pour réaliser au niveau matérielle notre design, nous avons dû l'implémenter dans un circuit programmable FPGA. Pour l'implémenter il fallait :

1. Reproduire le schéma bloc donné dans le laboratoire
2. Assigner les broches d'entrées, d'interrupteurs et de DEL dans un fichier QSF
3. Configurer le FPGA dans Quartus selon la procédure
4. Télécharger le design dans le FPGA

## 7 Conclusion

Ce laboratoire a permis de concevoir un comparateur 4 bits en complément à deux. Nous avons d'abord effectué une préparation avant chaque séance de laboratoire afin que nous puissions comprendre le fonctionnement général des circuits itératif. Après cette préparation, nous avons pu écrire en VHDL nos circuits planifiés. La simulation de notre design nous a permis de visualiser les entrées et les sorties que nous avions déclarées afin valider la fonctionnalité de notre design. Enfin, l'implémentation de notre circuit en FPGA nous a permis de valider en conditions réelles la logique conçue en VHDL. Malgré le bon déroulement du laboratoire, nous éprouvions des difficultés à ouvrir les fichiers du laboratoire sur des postes autre que l'ordinateur source. Ceci est probablement dû à l'utilisation des outils infonuagiques.