

# Compte rendu du TP de simulation fonctionnelle avec ModelSim

**Created By:** IMHAMED BOUJEMAA

February 21, 2023

## 1 Introduction

Dans ce compte rendu, je présente les résultats de mon travail pratique sur la simulation fonctionnelle avec ModelSim. Le TP a été divisé en plusieurs étapes, allant de l'étude d'un demi-additionneur à l'étude d'un registre à décalage sur 4 bits. J'ai également utilisé LaTeX pour produire ce compte rendu.

## 2 Étude d'un demi-additionneur

### 2.1 Description du circuit

Un demi-additionneur est un circuit logique combinatoire qui permet d'additionner deux bits de données et de produire une somme et une retenue. Le circuit est constitué de deux entrées (A et B) et de deux sorties (S et C).

### 2.2 Simulation avec ModelSim

J'ai utilisé ModelSim pour simuler le circuit d'un demi-additionneur. J'ai créé une nouvelle simulation en utilisant le fichier de test fourni par l'enseignant et j'ai exécuté la simulation. Les résultats de la simulation sont présentés dans la Figure 1.

### 2.3 Analyse des résultats

Les résultats de la simulation montrent que le demi-additionneur fonctionne comme prévu. La somme S est égale à 0 pour les entrées A=0 et B=0, à 1 pour les entrées A=0 et B=1, à 1 pour les entrées A=1 et B=0, et à 0 pour les entrées A=1 et B=1. La retenue C est égale à 0 pour les entrées A=0 et B=0, à 0 pour les entrées A=0 et B=1, à 0 pour les entrées A=1 et B=0, et à 1 pour les entrées A=1 et B=1.

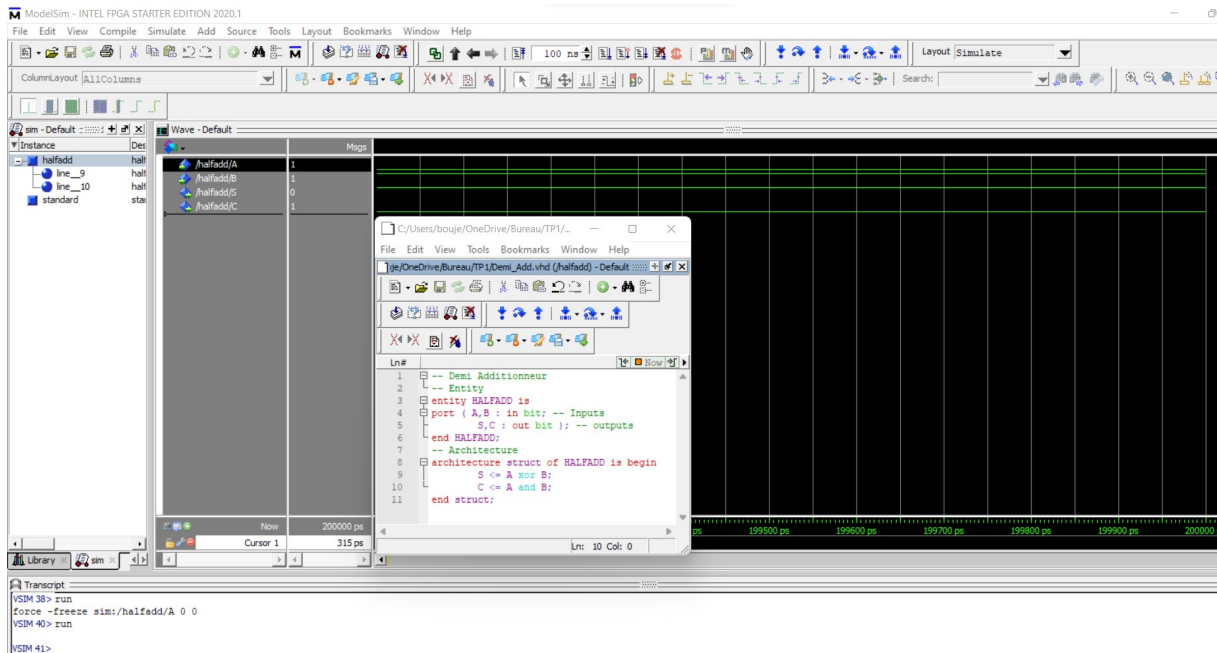


Figure 1: Résultats de la simulation d'un demi-additionneur avec ModelSim

### 3 Étude d'un additionneur complet

#### 3.1 Description du circuit

Un additionneur complet est un circuit logique combinatoire qui permet d'additionner deux nombres binaires de plusieurs bits et de produire une somme et une retenue. Le circuit est constitué de plusieurs blocs de demi-additionneurs et d'un bloc de logique de retenue.

#### 3.2 Simulation avec ModelSim

J'ai utilisé ModelSim pour simuler le circuit d'un additionneur complet. J'ai créé une nouvelle simulation en utilisant le fichier de test fourni par l'enseignant et j'ai exécuté la simulation. Les résultats de la simulation sont présentés dans la Figure 2.

#### 3.3 Analyse des résultats

Les résultats de la simulation montrent que l'additionneur complet fonctionne comme prévu. La somme est égale à 5 pour les entrées A=2 et B=3, ce qui correspond à l'addition binaire  $0010 + 0011 = 0101$ . La retenue est égale à 0 pour cette opération.

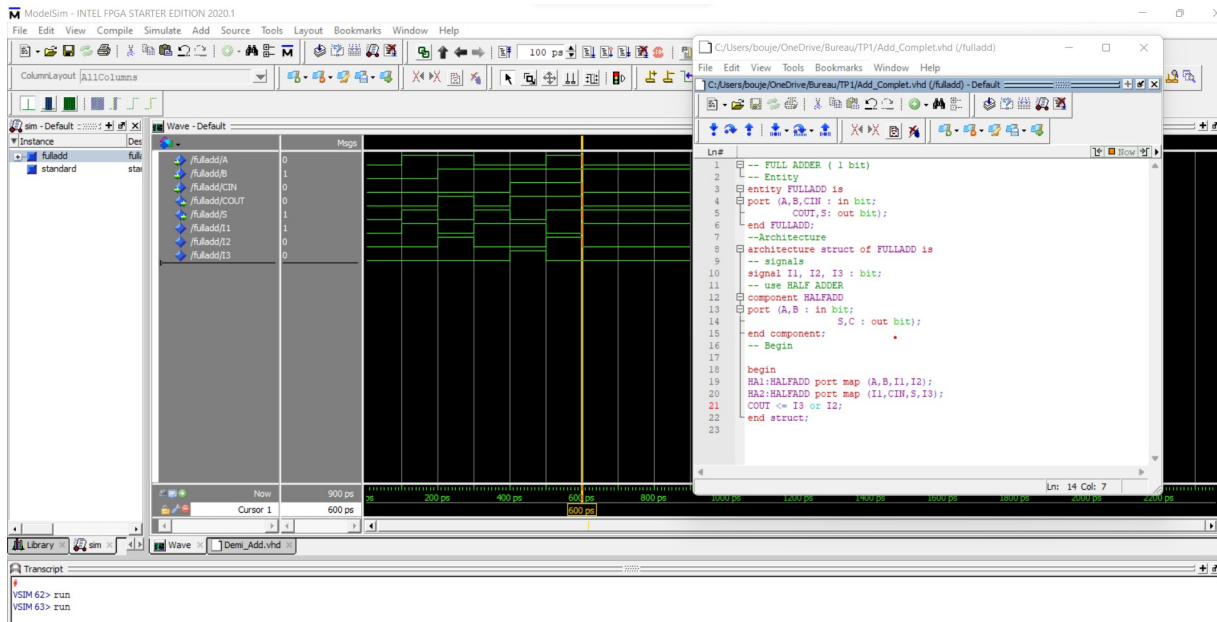


Figure 2: Résultats de la simulation d'un additionneur complet avec ModelSim

## 4 Étude d'un additionneur complet sur 4 bits

### 4.1 Description du circuit

Un additionneur complet sur 4 bits est un circuit logique combinatoire qui permet d'additionner deux nombres binaires de 4 bits et de produire une somme sur 4 bits et une retenue. Le circuit est constitué de 4 blocs de demi-additionneurs et d'un bloc de logique de retenue.

### 4.2 Simulation avec ModelSim

J'ai utilisé ModelSim pour simuler le circuit d'un additionneur complet sur 4 bits. J'ai créé une nouvelle simulation en utilisant le fichier de test fourni par l'enseignant et j'ai exécuté la simulation. Les résultats de la simulation sont présentés dans la Figure 3.

### 4.3 Analyse des résultats

Les résultats de la simulation montrent que l'additionneur complet sur 4 bits fonctionne comme prévu. La somme est égale à 9 pour les entrées  $A=5$  et  $B=4$ , ce qui correspond à l'addition binaire  $0101 + 0100 = 1001$ . La retenue est égale à 0 pour cette opération.

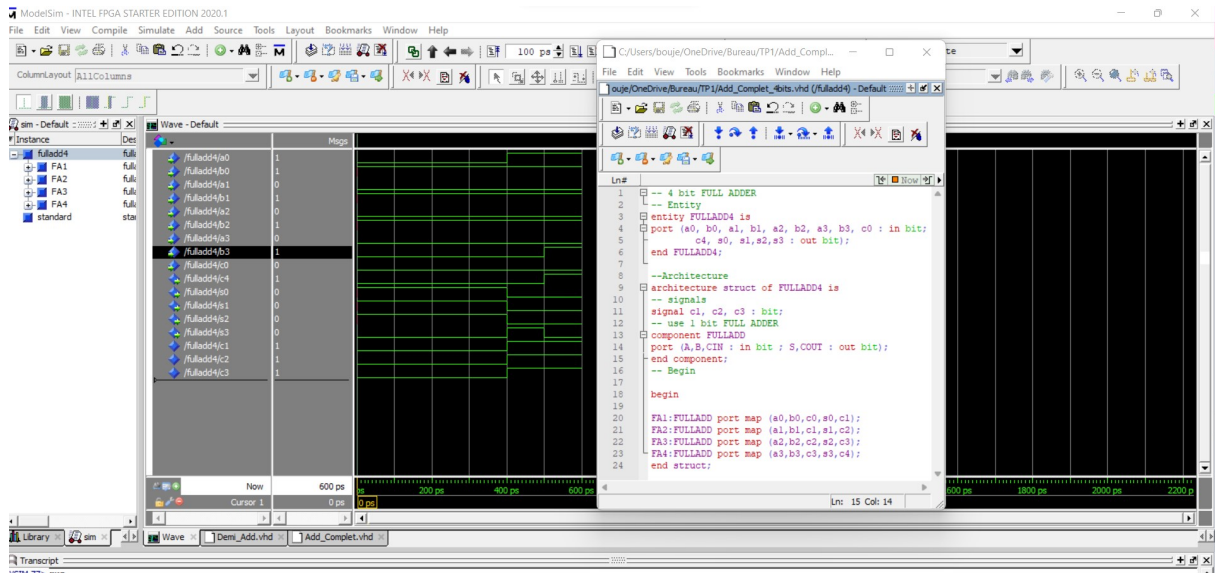


Figure 3: Résultats de la simulation d'un additionneur complet sur 4 bits avec ModelSim

## 5 Étude d'une bascule D

### 5.1 Description du circuit

Une bascule D est un circuit logique séquentiel qui permet de stocker un bit de données et de le conserver jusqu'à ce qu'un signal de déclenchement soit reçu. Le circuit est constitué d'une entrée de données (D), d'une entrée de déclenchement (CLK) et de deux sorties (Q et  $\overline{Q}$ ).

### 5.2 Simulation avec ModelSim

J'ai utilisé ModelSim pour simuler le circuit d'une bascule D. J'ai créé une nouvelle simulation en utilisant le fichier de test fourni par l'enseignant et j'ai exécuté la simulation. Les résultats de la simulation sont présentés dans la Figure 4.

### 5.3 Analyse des résultats

Les résultats de la simulation montrent que la bascule D fonctionne comme prévu. Lorsque le signal d'horloge (CLK) est à l'état haut, la sortie Q prend la valeur de l'entrée de données (D). Lorsque le signal d'horloge passe à l'état bas, la sortie Q conserve la valeur précédente. Cela permet de stocker la valeur de D jusqu'à ce que le signal d'horloge change à nouveau d'état.

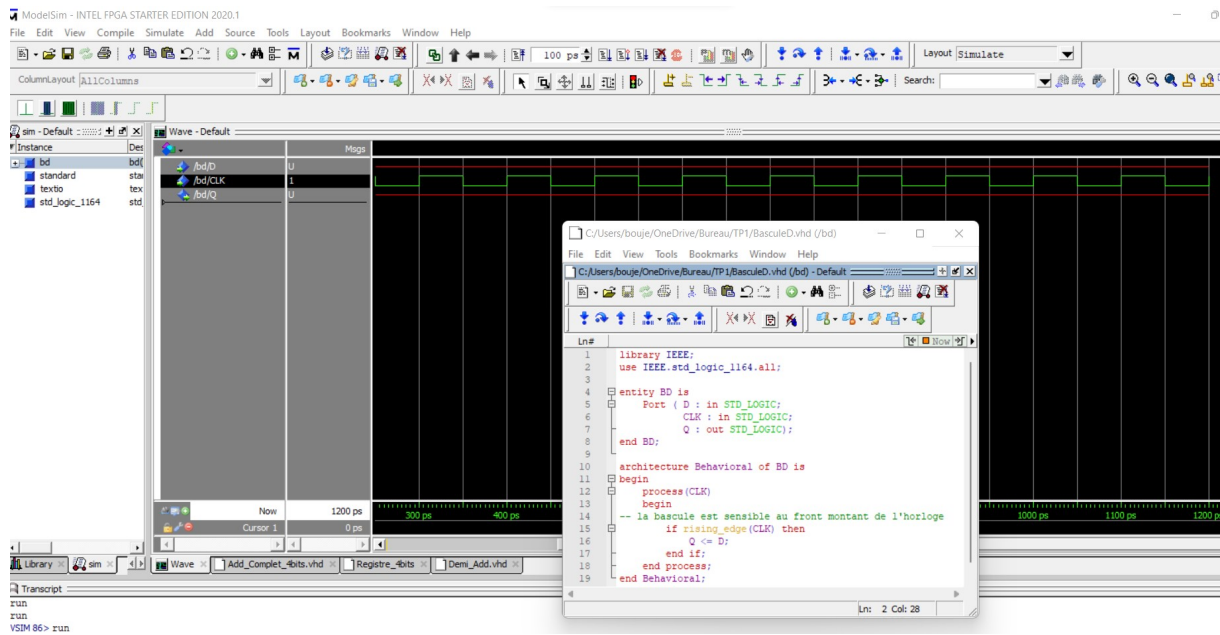


Figure 4: Résultats de la simulation d'une bascule D avec ModelSim

## 6 Étude d'un registre à décalage sur 4 bits

### 6.1 Description du circuit

Un registre à décalage sur 4 bits est un circuit logique séquentiel qui permet de stocker une séquence de 4 bits et de les décaler vers la gauche ou vers la droite. Le circuit est constitué de quatre bascules D et d'un multiplexeur qui permet de sélectionner la direction de décalage.

### 6.2 Simulation avec ModelSim

J'ai utilisé ModelSim pour simuler le circuit d'un registre à décalage sur 4 bits. J'ai créé une nouvelle simulation en utilisant le fichier de test fourni par l'enseignant et j'ai exécuté la simulation. Les résultats de la simulation sont présentés dans la Figure 5.

### 6.3 Analyse des résultats

Les résultats de la simulation montrent que le registre à décalage sur 4 bits fonctionne comme prévu. Lorsque le signal de direction est à l'état haut, la séquence de bits est décalée vers la gauche à chaque front montant de l'horloge. Lorsque le signal de direction est à l'état bas, la séquence de bits est décalée vers la droite à chaque front montant de l'horloge. Les sorties du registre correspondent à la séquence de bits stockée dans le registre après chaque décalage.

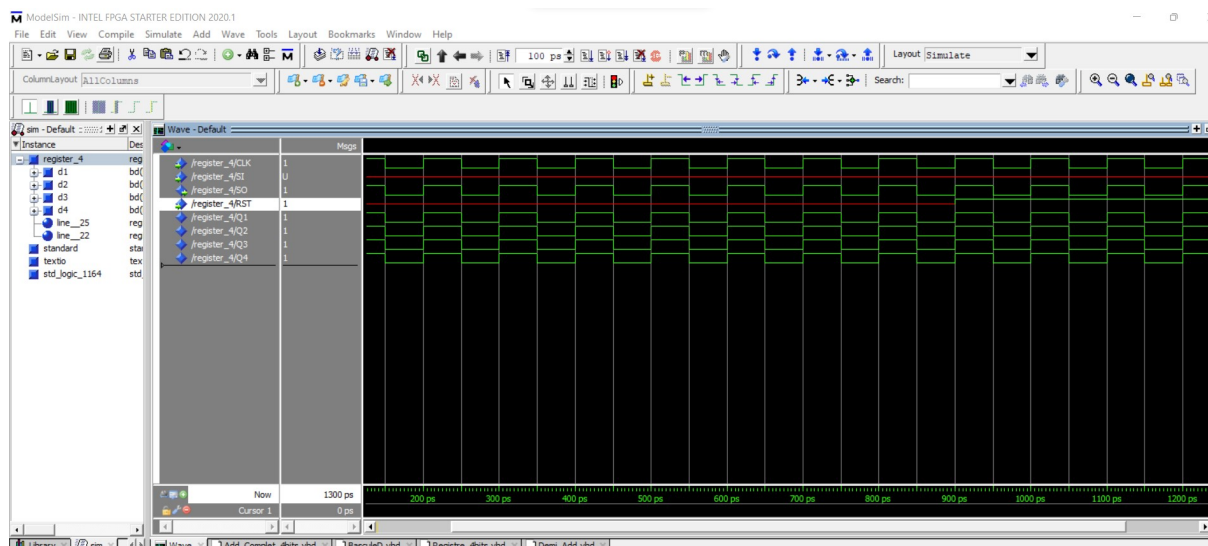


Figure 5: Résultats de la simulation d'un registre à décalage sur 4 bits avec ModelSim

## 7 Conclusion

Au cours de ce TP, j'ai étudié plusieurs circuits logiques combinatoires et séquentiels. J'ai simulé ces circuits en utilisant ModelSim et j'ai analysé les résultats des simulations. J'ai ainsi pu comprendre le fonctionnement de ces circuits et l'impact de différentes entrées sur leurs sorties. Ce TP m'a également permis de me familiariser avec le langage VHDL et l'environnement de simulation ModelSim.

The source code and associated files for this LAB can be found on Github at the following address: [https://github.com/IMHAMEDBOUJEMAA/Master\\_Course\\_1\\_2/tree/main/4\\_FPGA\\_VHDL/TP\\_WORK](https://github.com/IMHAMEDBOUJEMAA/Master_Course_1_2/tree/main/4_FPGA_VHDL/TP_WORK).