

# Cahier de laboratoire

Cătălin Bozan, Liviu Arsenescu

Le 22 Février 2024

## Résumé

Section for Abstract

## Contenu

<b>1</b>	<b>Rendu 1 (Laboratoire 1 : 22.02.2024)</b>	<b>1</b>
1.1	Tâches à effectuer . . . . .	1
1.2	Parcours . . . . .	1
1.3	Checklist . . . . .	1
<b>2</b>	<b>Rendu 2 (Laboratoire 2 : date)</b>	<b>2</b>
2.1	Tâches à effectuer . . . . .	2
2.2	Checklist . . . . .	2
<b>3</b>	<b>Rendu num (Laboratoire num : date)</b>	<b>3</b>
3.1	Tâches à effectuer . . . . .	3

# 1 Rendu 1 (Laboratoire 1 : 22.02.2024)

## 1.1 Tâches à effectuer

Analyser le projet Vivado fourni et créer les documents suivants:

1. Schéma bloc du processeur (nanoProcesseur)
2. Schéma bloc du contrôleur (nanoContrôleur)
3. Graph des états du séquenceur
4. Table de vérité des sorties du séquenceur
5. Plan mémoire

## 1.2 Parcours

Voici notre parcours, pour résoudre les tâches:

- **22.02.2023:**

1. Nous avons configuré un repository GitHub partagé, pour gérer facilement le projet, et avoir une gestion des versions si quelque chose ne va pas.
2. Nous avons regardé le projet initial, analysé et discuté chaque composant du processeur.
3. Nous avons choisi les outils que nous utiliserons pour le projet (LaTeX pour la documentation, draw.io pour dessiner les schémas).
4. Nous avons réparti les tâches pour le prochain laboratoire.

Ces tâches ont duré environ 2 heures et ont été réalisées en équipe.

- **29.02.2023:**

1. Ensemble: Nous avons discuté en détail les fichiers .vhd, pour pouvoir faire les schémas.
2. Liviu: J'ai commencé à faire le schéma du nanoProcesseur.
3. Cătălin: J'ai commencé à faire le schéma pour le nanoContrôleur.
4. Ensemble: Nous avons écrit la documentation.

## 1.3 Checklist

Numéro de la tâche	Effectuée	Cătălin	Liviu
1			
2			
3			
4			
5			

## 2 Rendu 2 (Laboratoire 2 : date)

### 2.1 Tâches à effectuer

Modifier le programme qui se trouve dans la ROM en utilisant uniquement le jeu d'instructions fourni. Un testbench doit être créé afin de valider et vérifier le bon fonctionnement de programme.

L'application en assembleur doit tourner en boucle et a les fonctionnalités suivantes:

1. Lire l'état des 8 dilswitch 1 et les sauvegarder en RAM
2. Lire l'état des 8 dilswitch 2
3. Si l'état des 8 dilswitchs 2 sont 0 alors:
  - (a) Afficher la valeur binaire lue sur les dilswitch 1 sur les 8 leds du bargraphe 1
  - (b) Afficher sur l'affichage 7 segment la valeur hexadécimale correspondant à l'état des 4 bits de poids forts des dilswitch 1
4. Sinon:
  - (a) Faire un ou exclusif entre les 8 dilswitch 1 et les 8 dilswitch 2, afficher le résultat sur le bargraphe 1 et afficher un C(pour calcul) sur l'affichage 7 segments

### 2.2 Checklist

Numéro de la tâche	Effectuée	Cătălin	Liviu
1			
2			
3.a			
3.b			
4.a			

### 3 Rendu num (Laboratoire num : date)

#### 3.1 Tâches à effectuer

- Task1