

Université de Cergy-Pontoise

RAPPORT

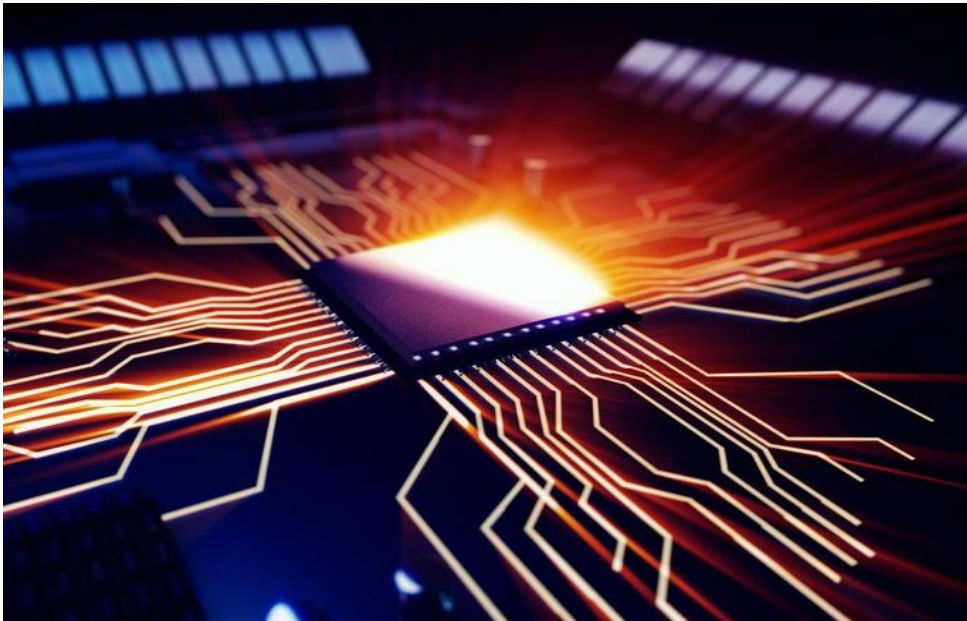
pour le projet d'architecture des ordinateurs
Licence d'Informatique deuxième année

sur le sujet

Conception d'un processeur 4 bits

rédigé par

GERARD Quentin et **PETITEVILLE** Valentin



Mai 2017

Table des matières

1	Introduction	2
2	Spécification du processeur	2
2.1	L'ALU	2
2.2	Le banc de registre	2
2.3	L'unité d'adressage	2
2.4	L'unité de contrôle	2
3	Réalisation	3
3.1	L'ALU	3
3.2	Le banc de registre	5
3.3	L'Unité d'adressage	7
3.4	L'unité de contrôle	8
3.5	Le CPU	9
4	Extension	10
4.1	Schéma de la soustraction	10
4.2	La soustraction	10
5	Déroulement du projet	11
5.1	Synchronisation du travail	11
5.2	Répartition du travail	11
5.3	Problèmes rencontrés	11

Table des figures

1	Schéma complet de l'ALU	3
2	Schéma d'une ALU 1bit	4
3	Schéma du FullAdder	4
4	Schéma du Multiplexeur	5
5	Schéma du banc de registres	6
6	Schéma d'un registre 4bits	6
7	Schéma du sélecteur de registre	7
8	Schéma complet de l'Unité d'adressage	7
9	Schéma du registre d'adresse d'instruction (PC)	8
10	Schéma complet de l'Unité de contrôle	8
11	Schéma complet du sélecteur de lecture	9
12	Schéma complet du processeur	10
13	Schéma de la soustraction	10

Liste des tableaux

1	Code opération de l'ALU	3
2	Code opération du CPU	9
3	Répartition des tâches	11

Remerciements

Les auteurs du projet voudraient remercier E.Ansermin, M.Belkaid et J.Lorandel.

1 Introduction

Dans le cadre du module de d'architecture des ordinateurs du second Semestre de L2, les étudiants doivent réaliser en binôme un projet avec le logiciel Logisim en réutilisant les éléments appris en cours. Le projet consiste en la réalisation d'un processeur 4 bits. Notre binôme est composé de Valentin PETITEVILLE, étudiant en L2-I dans le groupe A, et de Quentin GERARD étudiant en L2 CMI SIC.

2 Spécification du processeur

Dans cette section, nous présentons la spécification du processeur réalisé. Pour cela, nous allons décrire chaque élément qui le compose.

2.1 L'ALU

Une ALU est une Unité Arithmétique et Logique. Elle permet de réaliser des opérations sur des opérandes présentes à ses entrées. Pour ce projet, le but est de réaliser une ALU capable d'effectuer 8 opérations différentes.

2.2 Le banc de registre

Un banc de registres est une mémoire interne au processeur, dans laquelle sont stockées des valeurs envoyées ou sortant de l'ALU. Ce banc de registre contient :

- 1 entrée sur 4 bits
- 2 sorties sur 4 bits
- 3 signaux de controle (lecture sur X, lecture sur Y et écriture sur un registre)
- une horloge

2.3 L'unité d'adressage

L'unité d'adressage est en charge de communiquer avec les mémoires externes au processeur, elle a un rôle d'interface entre l'Unité de Contrôle et les mémoire externes. Ainsi c'est elle qui va s'occuper des données d'instruction envoyées à l'UC, mais aussi des données à envoyer aux registres ou à enregistrer dans la mémoire RAM.

2.4 L'unité de contrôle

L'UC (pour Unité de Contrôle) a pour but de décoder l'instruction contenu dans le registre d'instruction propre à elle même, et d'en déduire le positionnement des différents signaux de l'architecture (ALU, sources, destinations, accès mémoire).

3 Réalisation

3.1 L'ALU

Composition Notre ALU est donc composée de :

- 2 Entrées sur 4 Bits
- 1 Sortie sur 4 Bits
- 8 Opérations (Dont un Full Adder)
- 1 Multiplexeur 4Bits, 8 vers 1

Voici le tableau des instructions de l'ALU :

Opération	Code d'Opération
ADD	000
OR	001
AND	010
NOT	011
SUBSTRACT	100
NOR	101
NAND	110
NXOR	111

TABLE 1 – Code opération de l'ALU

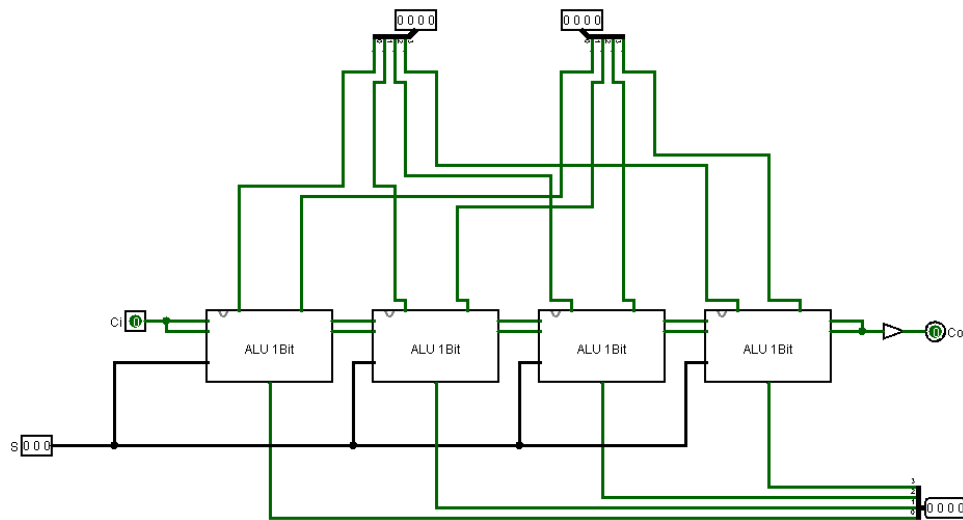


FIGURE 1 – Schéma complet de l'ALU

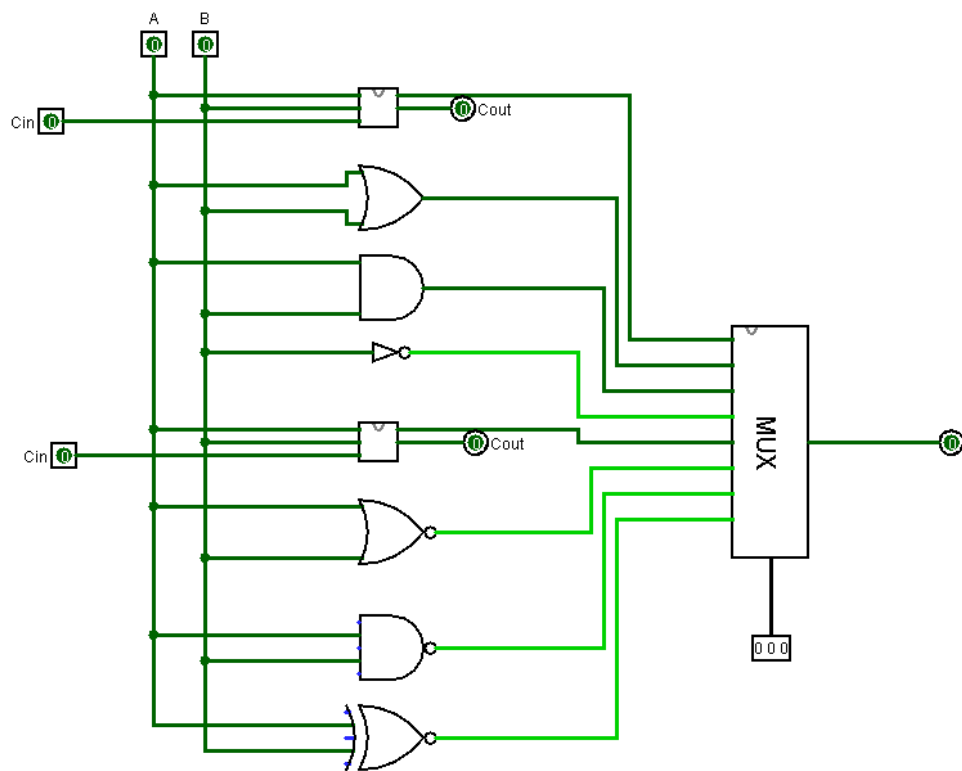


FIGURE 2 – Schéma d'une ALU 1bit

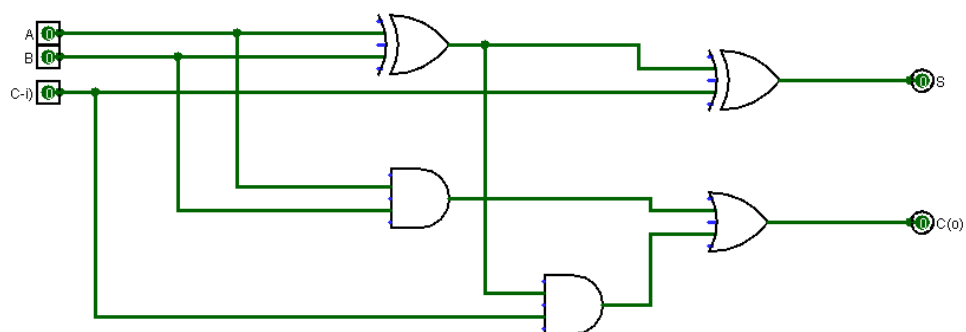


FIGURE 3 – Schéma du FullAdder

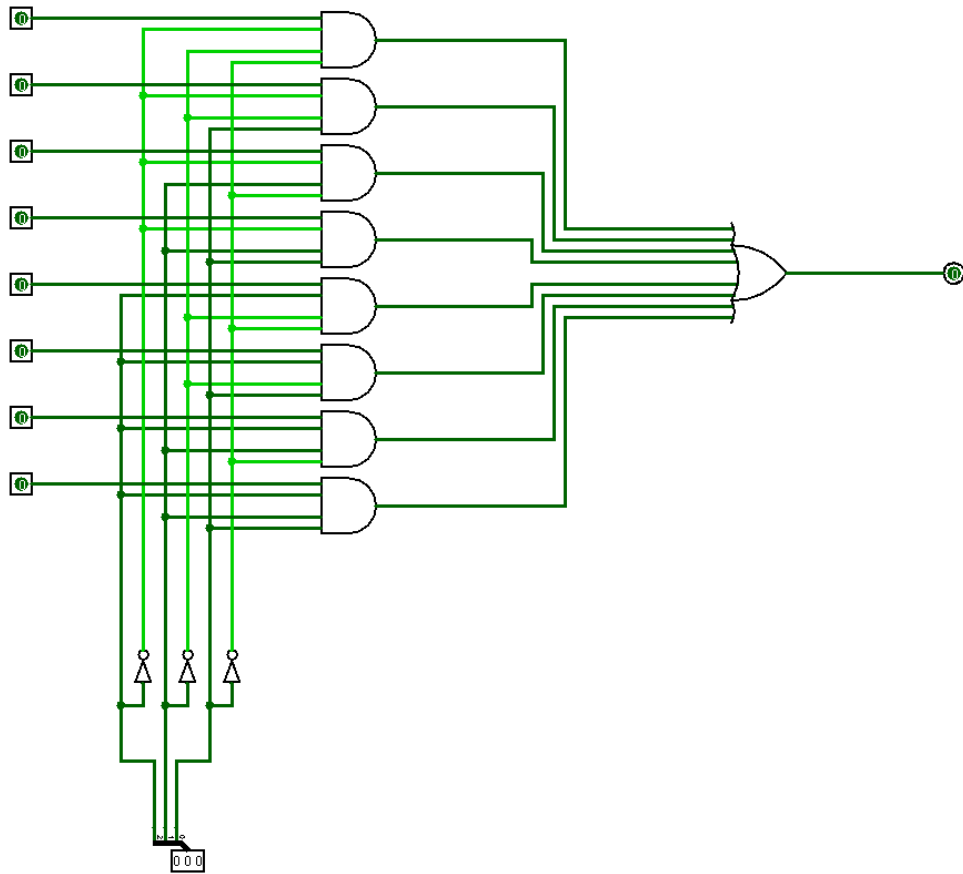


FIGURE 4 – Schéma du Multiplexeur

3.2 Le banc de registre

Notre banc de registre est composé de :

- Une entrée 4 bits
- 2 sorties 4 bits
- Un décodeur sur 2 bits en entrée
- Un bit de validation pour l'écriture
- 4 registres 4 bits
- 2 sélecteurs de registre (un pour le BUS X et l'autre pour le BUS Y)

Problème rencontré La bascule D que nous avons conçu (grâce aux schémas du cours), créait des problèmes d'oscillations lors du fonctionnement du registre, nous avons donc utilisé les registres implementés dans Logisim pour notre registre 4 bit. Nous avons préféré utiliser les registres de Logisim plutôt que de remplacer les Bascules D par des D Latch, malgré que ces derniers fonctionnaient très bien, car nous voulions profiter des avantages que nous offrait l'utilisation de l'horloge native à Logisim, notamment l'insensibilité au changement du signal entre deux fronts d'horloge, et ainsi anticiper un éventuel futur dysfonctionnement du banc de registre.

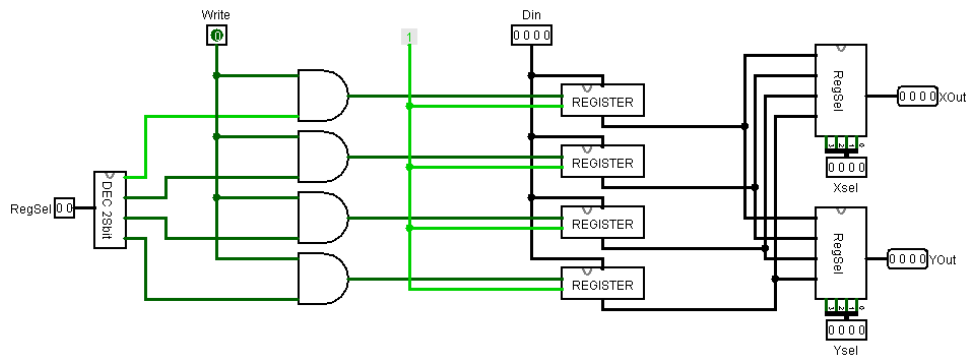


FIGURE 5 – Schéma du banc de registres

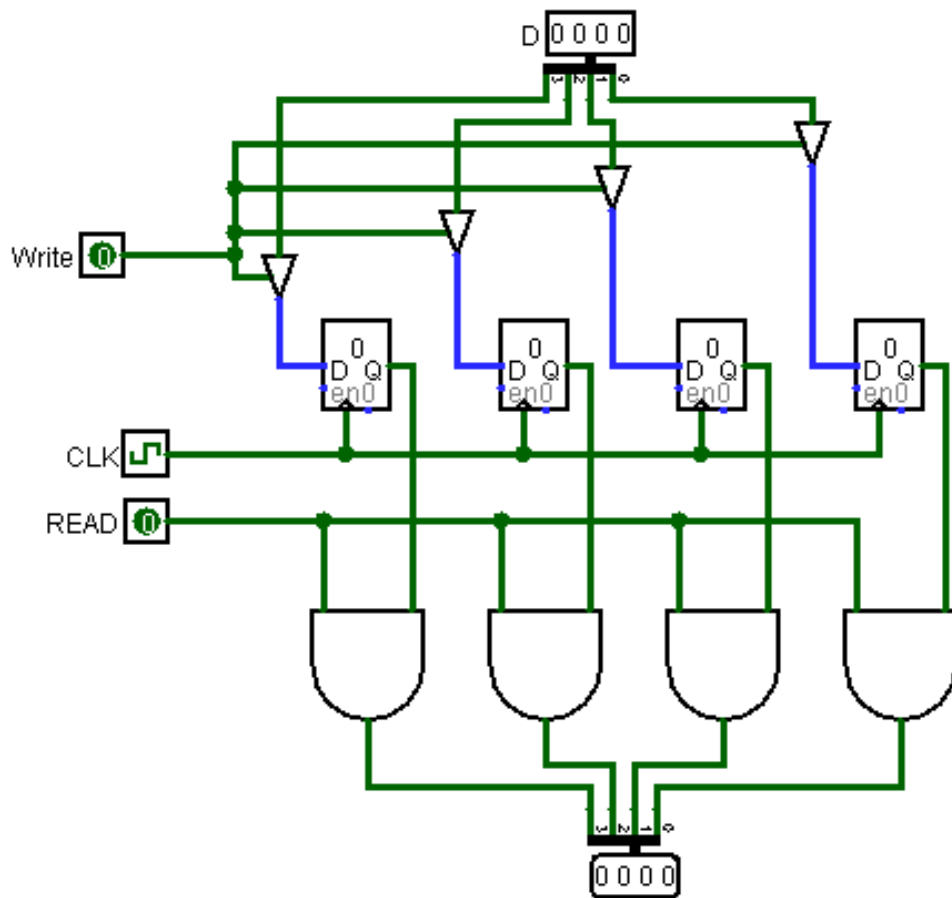


FIGURE 6 – Schéma d'un registre 4bits

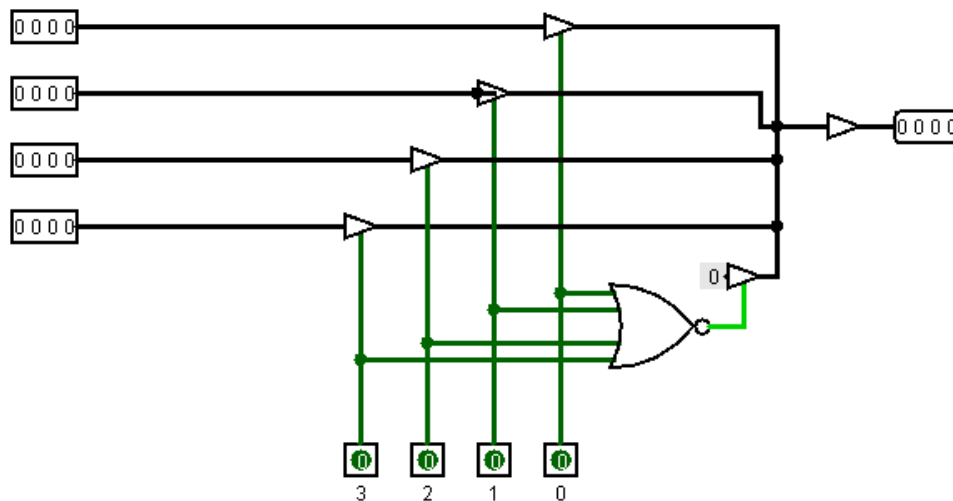


FIGURE 7 – Schéma du sélecteur de registre

3.3 L'Unité d'adressage

Notre Unité d'adressage est composée de 2 parties :

- Partie Instructions :
 - 1 entrée 4bits (Adresse de la prochaine instruction)
 - 2 sorties 4bits (Adresse de l'instruction en cours ; Une sortie pour la mémoire d'instruction ; Une sortie pour l'Unité de contrôle destinée au calcul de l'adresse de la prochaine instruction)
 - Un registre d'adresse 4bits
 - Un bit de validation d'écriture
- Partie Données :
 - 1 entrée 4bits
 - 1 sortie 4bits
 - 1 bit de validation d'écriture dans la mémoire de données

Problème rencontré Nous pouvons remarquer que la partie gérant la mémoire de données ne contient pas de registre d'adresse, en effet nous n'avons pas compris le rôle de ce registre, et son implémentation provoque un dysfonctionnement dans notre processeur dû au retard provoqué sur l'arrivée de l'information à la mémoire RAM.

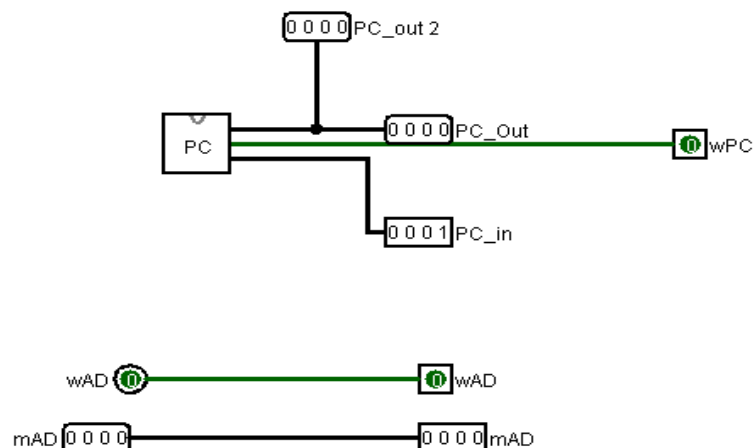


FIGURE 8 – Schéma complet de l'Unité d'adressage

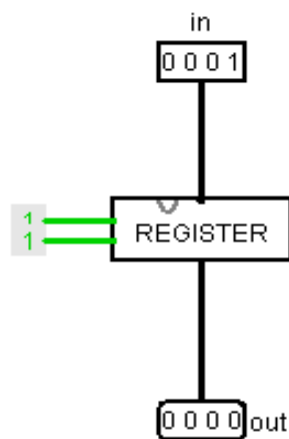


FIGURE 9 – Schéma du registre d'adresse d'instruction (PC)

3.4 L'unité de contrôle

Notre Unité de contrôle est composée d'un registre d'instruction ainsi que de multiple décodeurs d'instruction, s'occupant chacun d'une partie de l'instruction, et de plusieurs sélecteurs de sorties.

Problème rencontré Bien que nous l'ayons implémenté, la sortie Fetch nous a posé problème, en effet nous ne comprenons pas son rôle dans le processeur, c'est pourquoi la sortie est tout de même branchée à une constante (elle doit visiblement être constamment à 1) mais pas utilisée.

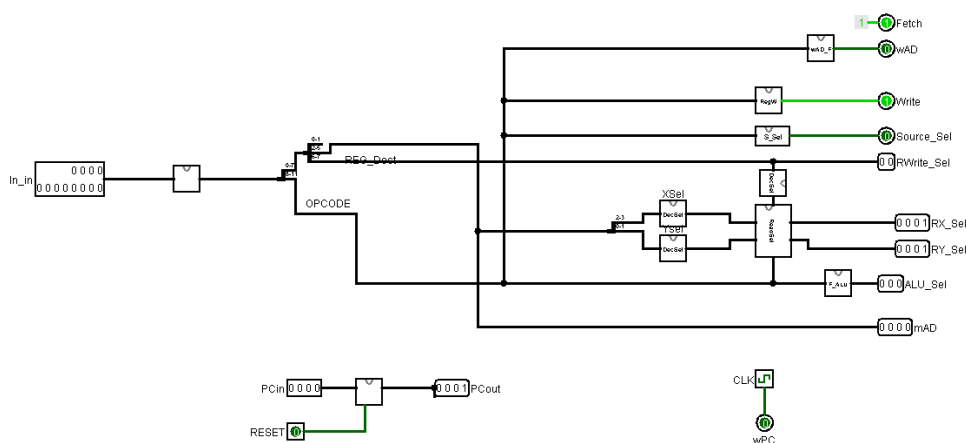


FIGURE 10 – Schéma complet de l'Unité de contrôle

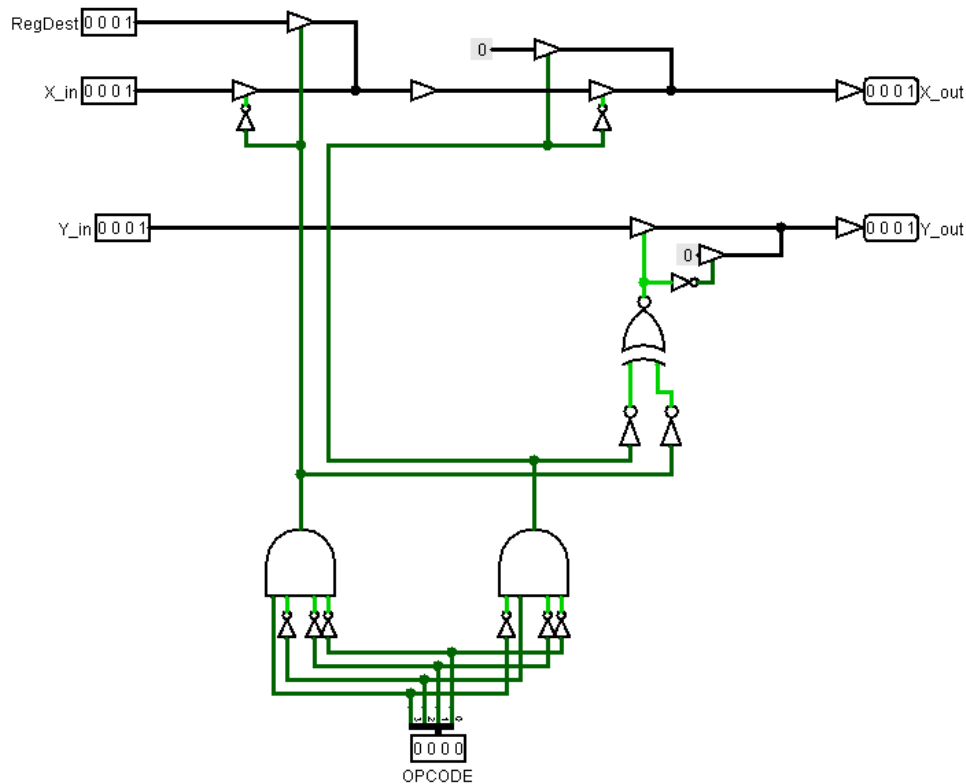


FIGURE 11 – Schéma complet du sélecteur de lecture

3.5 Le CPU

Enfin notre CPU relie tous les composants vu ci-dessus, ainsi que la mémoire RAM servant de mémoire de données ainsi qu'une mémoire ROM contenant le programme du processeur (la mémoire d'instruction).

Voici le tableau des instructions du processeur :

Opération	Code d'Opération
ADD	0000
OR	0001
AND	0010
NOT	0011
SUBSTRACT	0101
LOAD	0100
STORE	1000

TABLE 2 – Code opération du CPU

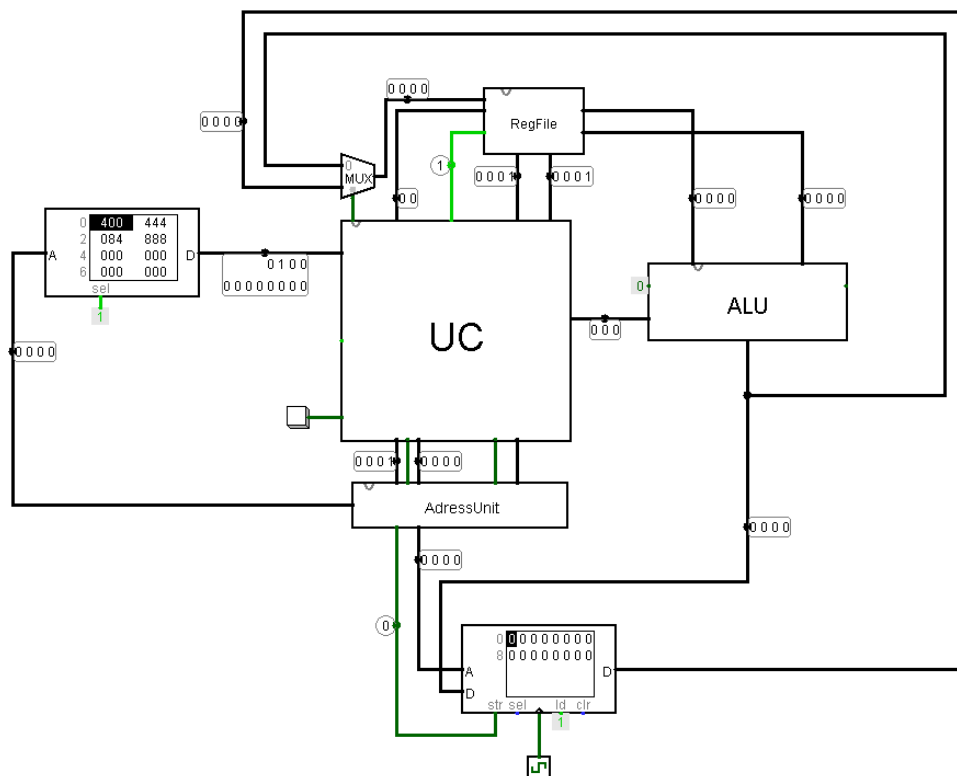


FIGURE 12 – Schéma complet du processeur

4 Extension

4.1 Schéma de la soustraction

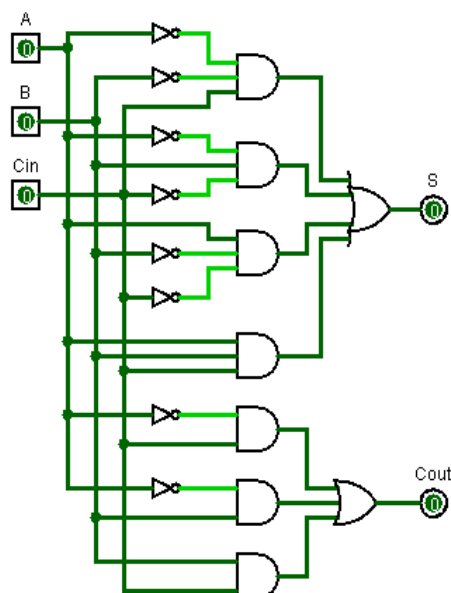


FIGURE 13 – Schéma de la soustraction

4.2 La soustraction

La soustraction a été implémenté à l'ALU, et possède son propre code d'instruction pour l'Unité de contrôle. L'Opération est complètement opérationnelle. Cependant par manque de temps et par soucis de conception, nous n'avons pu ajouter le signal sortant de l'ALU indiquant si la sortie était négative.

5 Déroulement du projet

Dans cette section, nous décrivons comment la réalisation du projet s'est déroulée au sein de l'équipe de projet.

5.1 Synchronisation du travail

Afin de pouvoir travailler sur ce projet nous avons utilisé la plateforme Github.

5.2 Répartition du travail

Valentin	Quentin
ALU	Banc de Registre
Extension	Unité de Controle
Compte rendu	Unité d'Adressage

TABLE 3 – Répartition des tâches

5.3 Problèmes rencontrés

- L'implémentation de l'opération soustraction
- Réalisation de l'unité d'adressage
- La compréhension de l'utilité d'un registre d'adresse de données
- La compréhension de certains signaux de sorties de l'UC (Fetch)
- La bascule D qui créait un problème d'oscillation (Nous avons donc utilisé les registres fourni par Logisim)