

**Université de Cergy-Pontoise**

**RAPPORT**

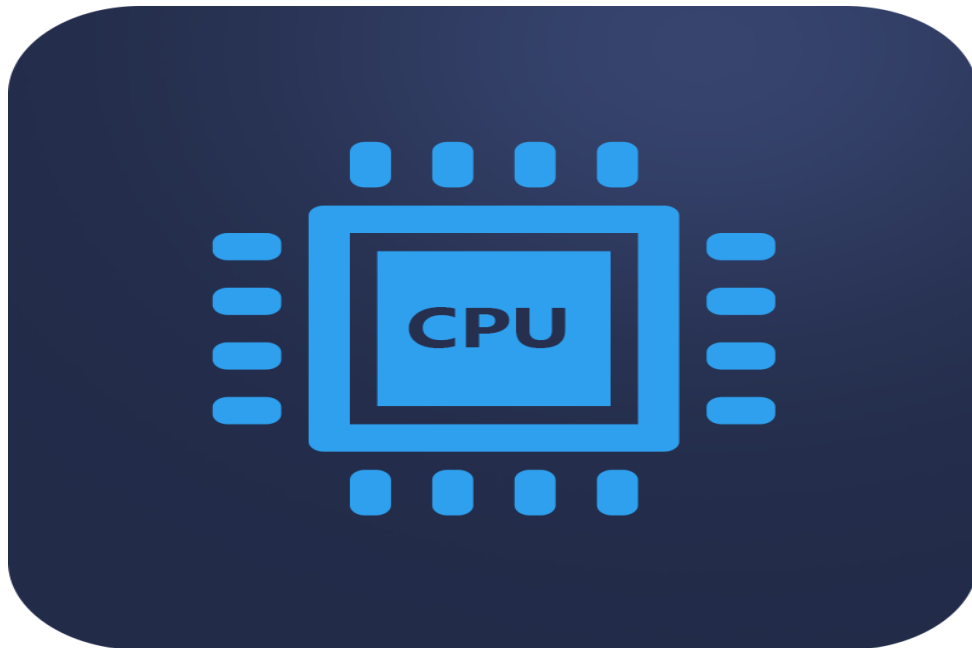
pour le projet d'architecture des Ordinateurs  
**Licence d'Informatique deuxième année**

sur le sujet

Conception d'un processeur 4 bits

rédigé par

**AYAD Ishak, METIDJI Fares**



Mai 2018

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Spécification</b>	<b>4</b>
2.1	L'ALU . . . . .	4
2.2	Le banc de registre . . . . .	4
2.3	L'unité d'adressage . . . . .	5
2.4	L'unité de contrôle . . . . .	5
<b>3</b>	<b>Réalisation</b>	<b>6</b>
3.1	L'ALU . . . . .	6
3.2	Le banc de registre . . . . .	7
3.3	L'unité d'adressage . . . . .	8
3.4	L'unité de contrôle . . . . .	8
3.5	Le CPU . . . . .	11
<b>4</b>	<b>Extension</b>	<b>12</b>
4.1	La soustraction . . . . .	12
4.1.1	La soustraction . . . . .	12
4.1.2	Schéma de la soustraction . . . . .	12
4.2	Indicateurs de l'UAL . . . . .	12
4.2.1	Résultat nul . . . . .	12
4.2.2	débordement et signaux d'entrée . . . . .	13
<b>5</b>	<b>Annexes</b>	<b>14</b>
5.1	Full adder 1bit . . . . .	14
5.2	Décodeur 3bits . . . . .	14
5.3	le décodeur d'instructions . . . . .	15
5.4	registre 12bits . . . . .	15
5.5	CPU . . . . .	16
<b>6</b>	<b>Déroulement du projet</b>	<b>17</b>
6.1	Répartition des tâches . . . . .	17
6.2	Synchronisation du travail . . . . .	17
6.3	Problèmes rencontrés . . . . .	17
6.4	Calendrier . . . . .	18
<b>7</b>	<b>Conclusion</b>	<b>19</b>

## Table des figures

1	Vu d'ensemble sur l'ALU . . . . .	4
2	Vu d'ensemble sur le banc de registre . . . . .	4
3	Vu d'ensemble sur l'unité d'adressage . . . . .	5
4	Vu d'ensemble sur l'unité de contrôle . . . . .	5
5	Circuit de l'ALU . . . . .	6
6	Circuit du banc de registres . . . . .	7
7	Circuit de l'unité d'adressage . . . . .	8
8	Circuit de l'unité de contrôle . . . . .	10
9	Schéma du CPU . . . . .	11
10	Circuit logique de la soustraction . . . . .	12
11	Circuit logique du comparateur quatre bits . . . . .	13
12	implémentation du comparateur quatre bits . . . . .	13
13	implémentation du comparateur quatre bits et des signaux d'entrées . . . . .	13

14	Circuit logique du Full adder 1bit . . . . .	14
15	Circuit logique du décodeur 3bits . . . . .	14
16	Circuit logique du décodeur d'instructions . . . . .	15
17	Circuit logique du registre 12bits . . . . .	15
18	Vu d'ensemble sur le CPU . . . . .	16
19	Calendrier . . . . .	18

## Remerciements

Avant tout développement de ce projet, il apparaît opportun d'adresser nos remerciements à tous ceux qui nous ont aidés pour la réalisation de ce projet. Nous tenons à remercier en premier lieu Monsieur N.Beausse et J.Lorandel, notre encadrant lors de ce projet, auprès duquel nous avons pu bénéficier d'un grand soutien. Nous remercions également Monsieur M.A.Khelif.

# 1 Introduction

**Contexte :** Dans le cadre du module 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 à réaliser un processeur de 4 bits. Notre binôme est composé de Ayad Ishak et de Metidji Fares, étudiants en L2-I dans le groupe D.

**Objet :** Créer un processeur 4 bits.

**Outils de développement :** Nos outils de développement sont ceux qui nous ont été conseillés par notre enseignant et que nous avons utilisé au cours de ce semestre en TP's d'architecture des ordinateurs et Projet. Nous avons utilisé le logiciel logisim. Nous avons synchronisé notre travail en utilisant le service web d'hébergement GitHub qui nous a beaucoup aidés pour travailler en groupe. Enfin, ce rapport de projet a été rédigé avec LaTeX sur le service web Overleaf.

**Structure du rapport :** Notre première partie concernera les spécifications de notre projet, elle contiendra toutes les fonctionnalités du processeur. Ensuite, nous parlerons de la partie réalisation, dans laquelle nous présenterons la conception détaillée de notre processeur et viendra après les extensions de notre processeur. Puis nous aborderons la partie déroulement où le calendrier de notre travail et la répartition des tâches seront présentées, pour arriver ensuite à la conclusion où nous allons donner le bilan de notre projet.

## 2 Spécification

**Chapeau :** Nous avons présenté l'objectif du projet dans la section 1. Dans cette section, nous présentons la spécification de notre processeur 4bits réalisé. Ceci correspond principalement au cahier des charges. 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.

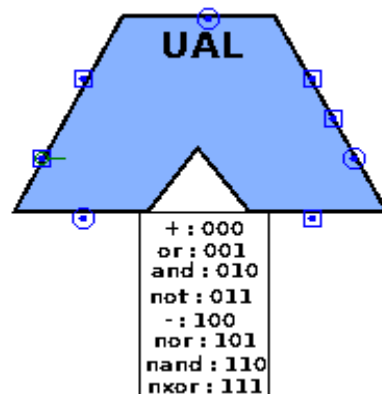


FIGURE 1 – Vu d'ensemble sur l'ALU

### 2.2 Le banc de registre

Le registre de 4 bits devra suivre le modèle suivant :

1. deux sorties de 4 bits pour chacune.
2. une entrée de 4 bits.
3. deux signaux de contrôle de lecture d'un bit chacun.
4. un signal de contrôle d'écriture d'un bit.
5. deux ports d'extension et une horloge.

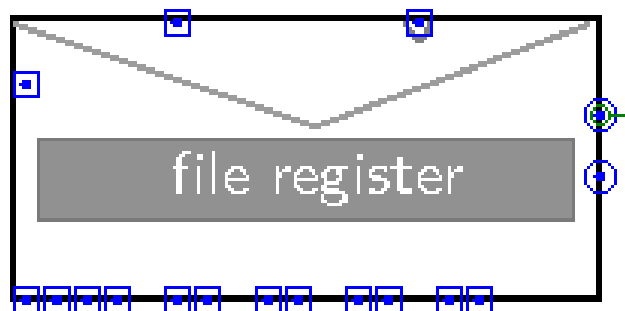


FIGURE 2 – Vu d'ensemble sur le banc de registre

## 2.3 L'unité d'adressage

L'unité d'adressage est uniquement composée de deux registre 4 bits : PC et AD.

1. PC est les registre d'adresse d'instruction.
2. AD est le registre d'adresse de données.

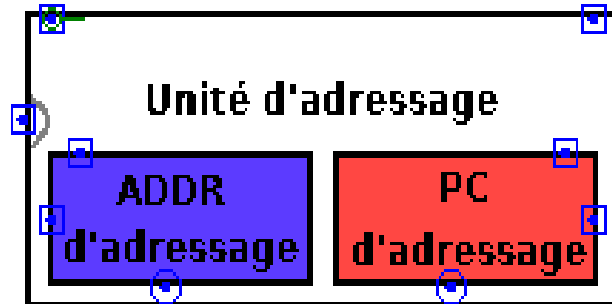


FIGURE 3 – Vu d'ensemble sur l'unité d'adressage

## 2.4 L'unité de contrôle

L'unité de contrôle (UC) c'est l'unité qui contrôle l'exécution des instructions machines par le processeur a pour rôle de placer les valeurs des différentes signaux de commande de l'architecture à chaque cycle. Le contrôle du chemin de données :

1. l'UAL par 3 signaux de commande
2. le choix de l'opérande X de l'ALU parmi 4 registres, donc 4 bits de commande
3. le choix de l'opérande Y de l'ALU parmi 4 registres, donc 4 bits de commande
4. le choix de la destination du rangement parmi 4 registres, donc 4 bits de commande
5. le choix d'un type d'accès mémoire (vers la mémoire) : Ecriture ou Lecture et le choix de faire un Fetch de l'instruction, soit 3 nouveaux signaux de commande

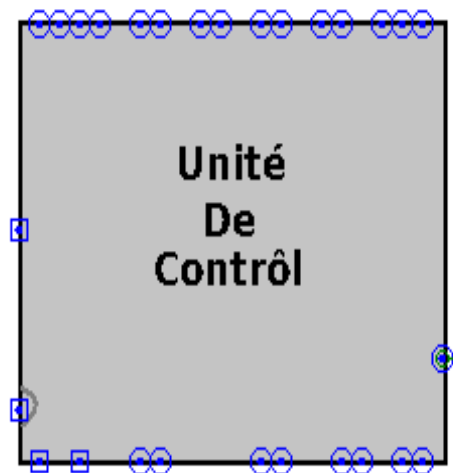


FIGURE 4 – Vu d'ensemble sur l'unité de contrôle

### 3 Réalisation

**Chapeau :** Nous avons présenté les spécifications du projet dans la section 2. Dans cette section, nous détaillons la conception de notre processeur et les techniques logique utilisées.

#### 3.1 L'ALU

**Composition :** Notre ALU est donc composée de :

1. deux entrées sur quatre bits.
2. une sortie sur quatre bits.
3. huit opération (dont un FULL-adder et FULL-subtractor).
4. un décodeur sur trois bits, trois vers huit.

Voici le tableau des instructions de l'ALU :

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

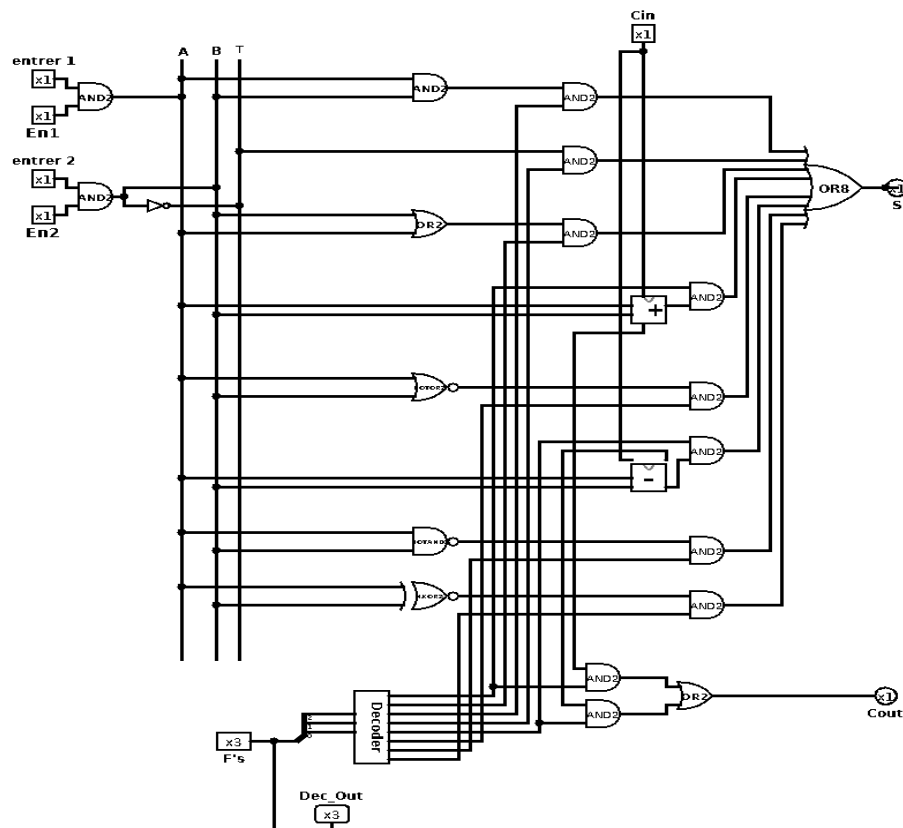


FIGURE 5 – Circuit de l'ALU

### 3.2 Le banc de registre

Le banc de registre est l'unité du CPU qui va stocker provisoirement les données provenant de la mémoire RAM ou de la sortie de l'ALU avant de diriger soit vers l'entrée A ou l'entrée B de l'ALU pour qu'un calcul soit effectué sur celle-ci.

**Composition :** Notre Banc de registre est donc composée de :

1. une entrée un mot de quatre bits
2. quatre registre de quatre bits
3. chaque registre peut envoyer la donnée qu'il stocke soit vers la sortie X (l'entrée A de l'ALU) soit vers la sortie Y (l'entrée B de l'ALU) , on a donc deux bits de sélection par registre , soit huit bits de sélection en sortie de registre.

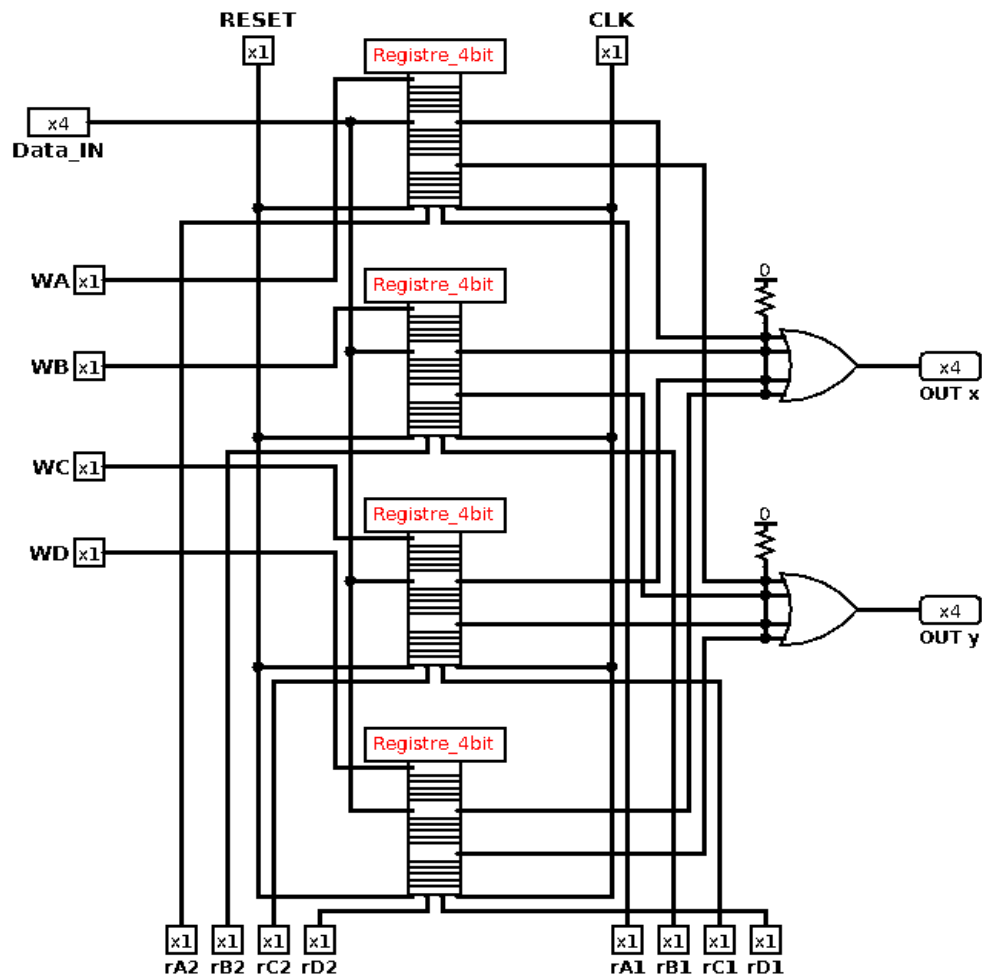


FIGURE 6 – Circuit du banc de registres



### 3.3 L'unité d'adressage

L'unité d'adressage contient deux registre de quatre bits : le PC(Programm counter) et le AD(Address)

1. Le PC est le registre qui contient l'adresse de la mémoire d'instruction a exécuté c'est-à-dire l'adresse contenant l'instruction a traité pas le CPU.
2. Le AD est le registre contenant l'adresse de la mémoire de données qui est active c'est-à-dire l'adresse contenant les données traitées en lecture ou en écriture par le CPU.

**Composition :** Notre unité d'adressage est donc composée de :

1. deux entrées de quatre bits vers les deux registres.
2. deux signaux de lecture.
3. deux sortie des deux registres.

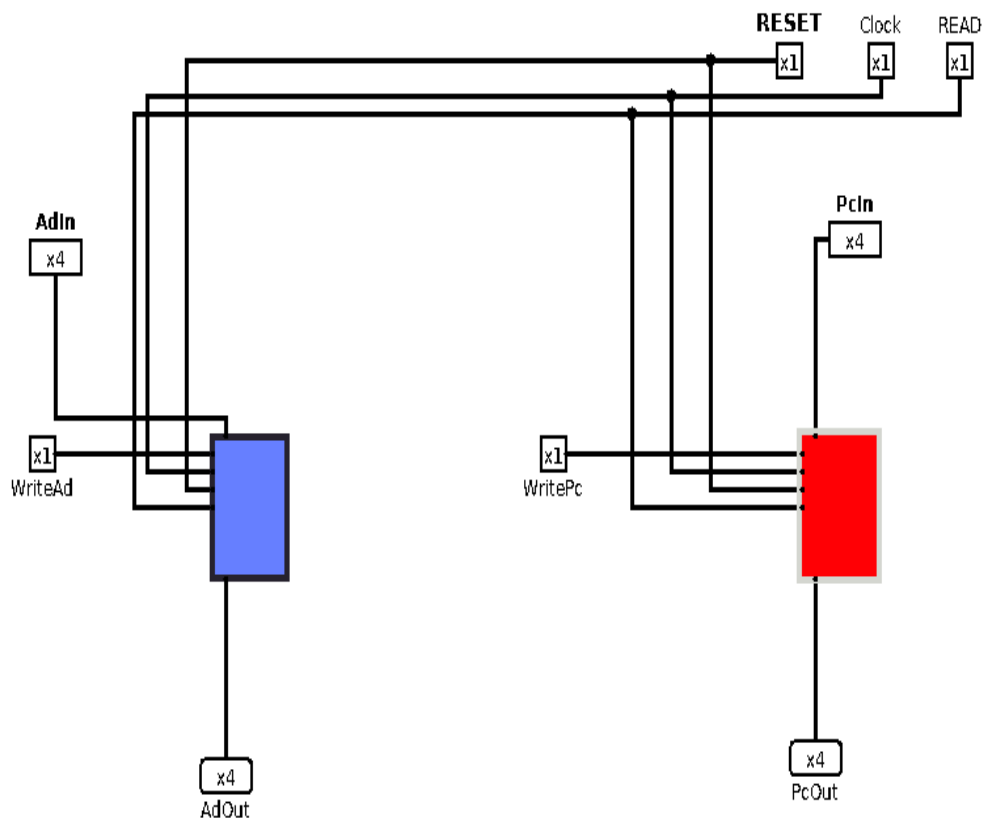


FIGURE 7 – Circuit de l'unité d'adressage

### 3.4 L'unité de contrôle

L'unité de contrôle est l'unité principale du CPU elle vient activer tous les signaux de commande du processeur en fonction de l'instruction qui a été chargée depuis la mémoire. Cette unité contient donc le registre d'instruction qui contient les différents champs correspondant aux catégories de commandes du processeur.

**les différents signaux :**

1. la sélection de l'opération réaliser par l'ALU.
2. la sélection des registre dont le contenus sont lus et écrits.
3. les indicateurs d'accès mémoire READ,WRITE et FETCH

L'unité de contrôle incrémente également le contenu du registre PC pour que l'instruction suivante soit lue à chaque cycle d'horloge. Si l'entrée Reset est activée, le registre PC est remplacé à sa position BOOT. Les instructions peuvent avoir deux formats :

**Format registre :**

OPCODE (4bits)	RES (2bits)	X (2bits)	Y (2bits)	EXT (2bits)
----------------	-------------	-----------	-----------	-------------

1. OPCODE : code de l'opération à réaliser.
2. RES : code de sélection du registre où sera stocké le résultat de l'opération.
3. X : sélection du registre contenant l'opérande X (entrée A de l'ALU)
4. Y : sélection du registre contenant l'opérande Y (entrée B de l'ALU)
5. EXT : extension possible des instructions.

**Format immédiat :**

OPCODE (4bits)	RES (2bits)	ADRESS (4bits)	EXT (2bits)
----------------	-------------	----------------	-------------

1. OPCODE : code de l'opération à réaliser.
2. RES : sélection du registre source pour écrire dans la mémoire de données, ou destination pour lire dans la mémoire de données.
3. ADRESS : sélection de l'adresse mémoire à lire ou écrire selon l'opération effectuée.
4. EXT : extension possible des instructions.

**Le jeu d'instruction de CPU est le suivant :**

Nom de l'instruction	OPCODE	format
ADD	0000	Registre
OR	0001	Registre
AND	0010	Registre
NOT(B)	0011	Registre
LOAD	0100	Immédiate
STORE	1000	Immédiate
SUB	0101	Registre
NOR	0110	Registre
NAND	0111	Registre
NXOR	1010	Registre

**L'activation des signaux de lecture et écriture dans le banc de registre est pilotée comme suit :**

Source	Code	(w,r)A	(w,r)B	(w,r)C	(w,r)D
A	00	1	0	0	0
B	01	0	1	0	0
C	10	0	0	1	0
D	11	0	0	0	1

L'activation des signaux de commande de l'ALU,READ,WRITE,FETCH et accès à l'unité d'adressage est pilotée comme suit :

instruction	OPCODE	Sélection ALU	READ	WRITE	WRITE AD	WRITE PC	FETCH
ADD	0000	000	0	0	0	1	1
OR	0001	001	0	0	0	1	1
AND	0010	010	0	0	0	1	1
NOT(B)	0011	011	0	0	0	1	1
LOAD	0100	000	1	0	1	1	1
STORE	1000	000	0	1	1	1	1
SUB	0101	100	0	0	0	1	1
NOR	0110	101	0	0	0	1	1
NAND	0111	110	0	0	0	1	1
NXOR	1010	111	0	0	0	1	1

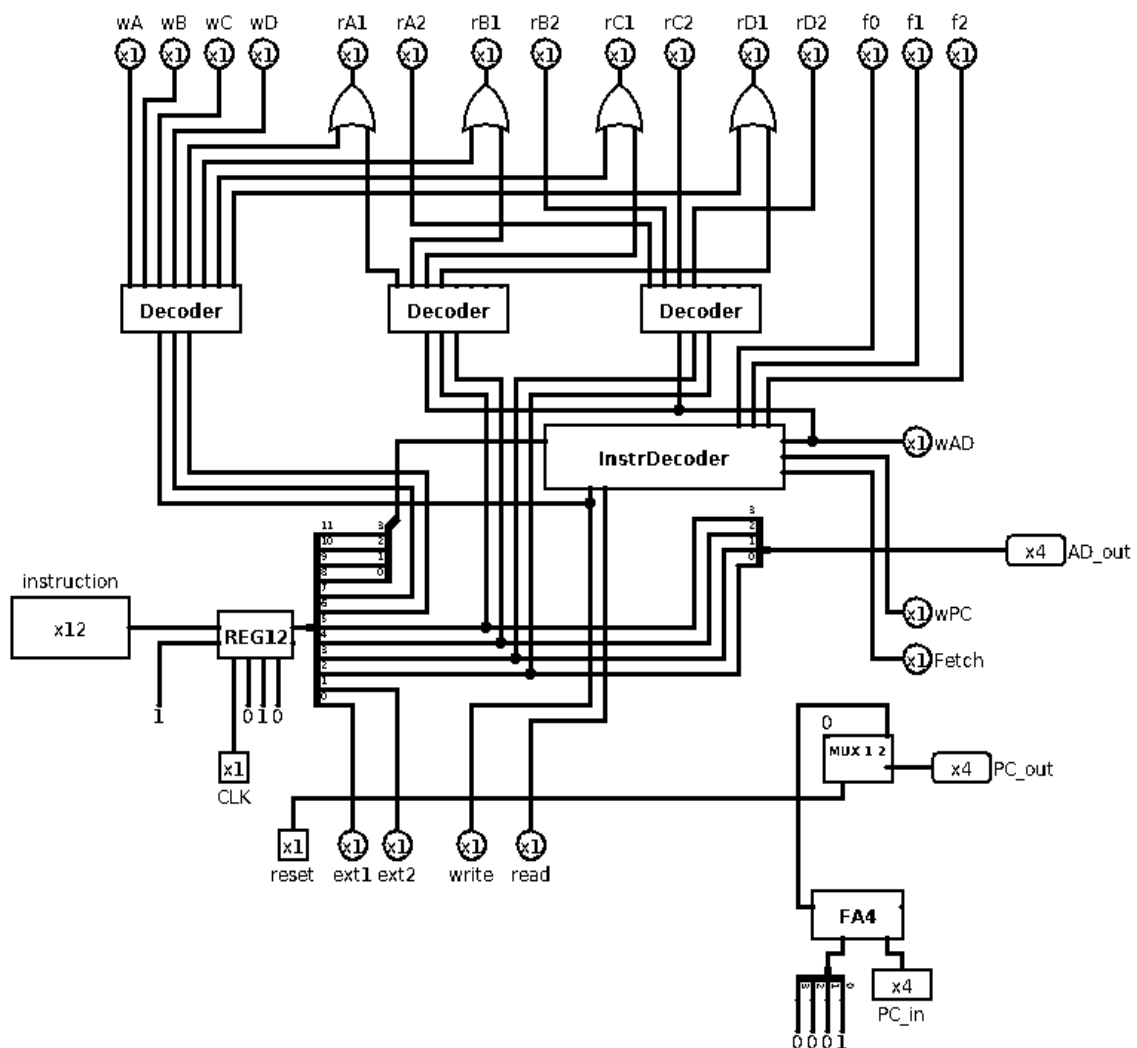


FIGURE 8 – Circuit de l'unité de contrôle

### 3.5 Le CPU

les élément du CPU indiqué dans 3.1, 3.2, 3.3, 3.4 serrant lié entre eux de la manière suivante :

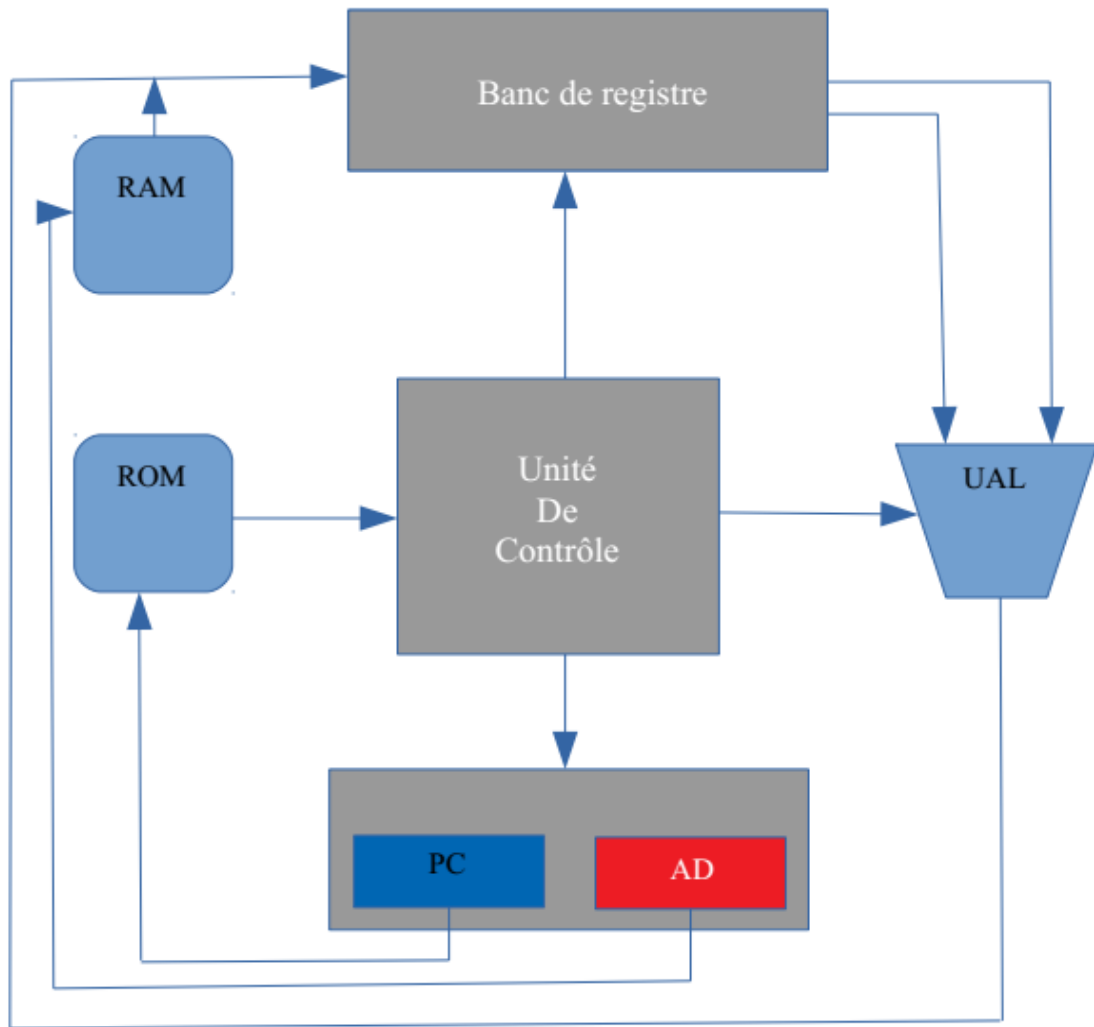


FIGURE 9 – Schéma du CPU

L'unité de contrôle va lire les instruction dans la ROM. Elle va les décoder soit au format immédiat soit au format registre en fonction du code d'opération et active ces différents sortie et elle va alimenté les autres composants.

## 4 Extension

**Chapeau :** Nous avons présenté la réalisation de notre projet dans section3. Dans cette section nous allons expliquer les extension apporté au projet.

### 4.1 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.

#### 4.1.1 La soustraction

table de vérité de la soustraction :

A	B	C	DIFF	BORROW
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

#### 4.1.2 Schéma de la soustraction

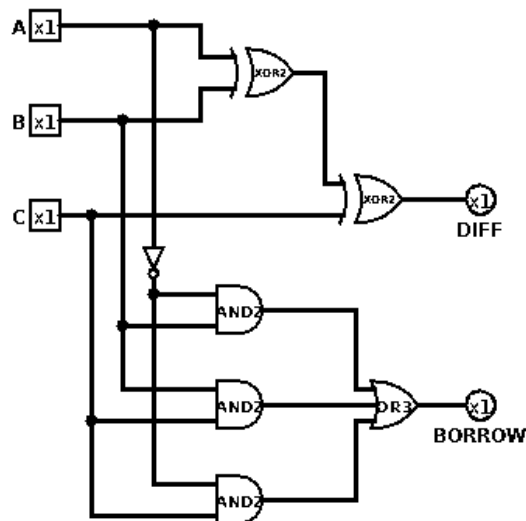


FIGURE 10 – Circuit logique de la soustraction

## 4.2 Indicateurs de l'UAL

### 4.2.1 Résultat nul

pour vérifier si le résultat est nul ou non on a mis en place un comparateur de quatre bits. En comparant entre le résultat obtenu de l'ALU et une constante nulle on a eu le résultat.

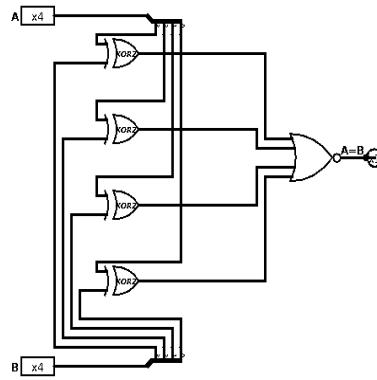


FIGURE 11 – Circuit logique du comparateur quatre bits

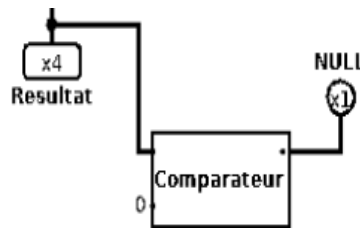


FIGURE 12 – implémentation du comparateur quatre bits

#### 4.2.2 débordement et signaux d'entrée

l'ajout d'un signal d'indication d'OVF, d'un signal de retenue sortante, d'un signal de résultat négatif dans l'ALU qui provoquent l'allumage d'une diode s'ils sont actifs.

l'ajout signaux d'entrée de l'UAL : enA et enB qui, s'ils sont actifs, autorisent la prise en compte des entrées A et /ou B. Dans le cas contraire, les entrées sont à 0.

Relier ces signaux à l'unité de contrôle de manière à ce qu'ils apparaissent dans l'instruction dans le champs EXT.

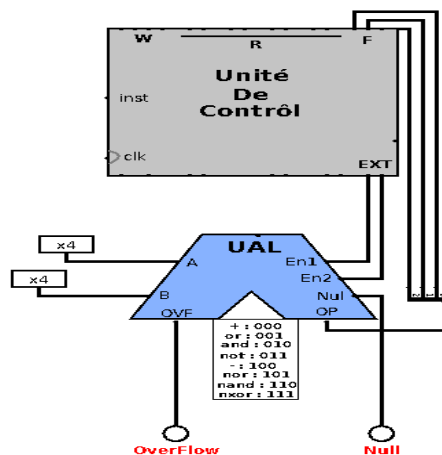


FIGURE 13 – implémentation du comparateur quatre bits et des signaux d'entrées

## 5 Annexes

**Chapeau :** Nous avons présenté les extension de notre projet dans section4. Dans cette section nous allons expliquer les différents éléments logique utilisé dans le projet.

### 5.1 Full adder 1bit

Le full adder 1bit fait une addition de 2bits A et B et d'un bit de retenu en entrée. in en ressort un bit de retenu et un bit de résultat.

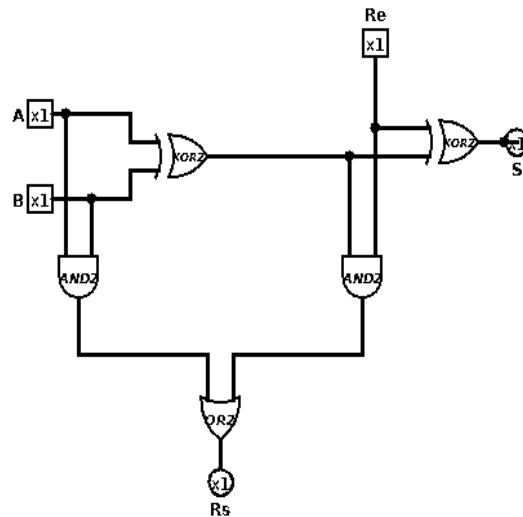


FIGURE 14 – Circuit logique du Full adder 1bit

### 5.2 Décodeur 3bits

Le décodeur 3bits prend en entrée un mot de 3bits et active l'une des 8 sorties en fonction du signal d'entrée reçu.

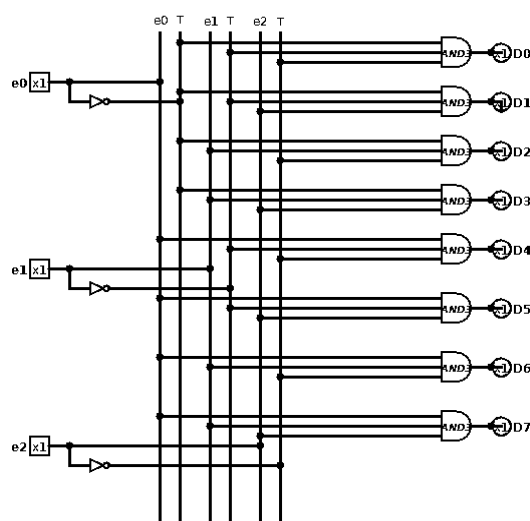


FIGURE 15 – Circuit logique du décodeur 3bits

### 5.3 le décodeur d'instructions

le décodeur d'instruction est un élément primordial de l'unité de contrôle il prend en entrée un mot de 4bits (OPCODE) et renvoie des signaux de sélection pour l'ALU et l'unité d'adressage.

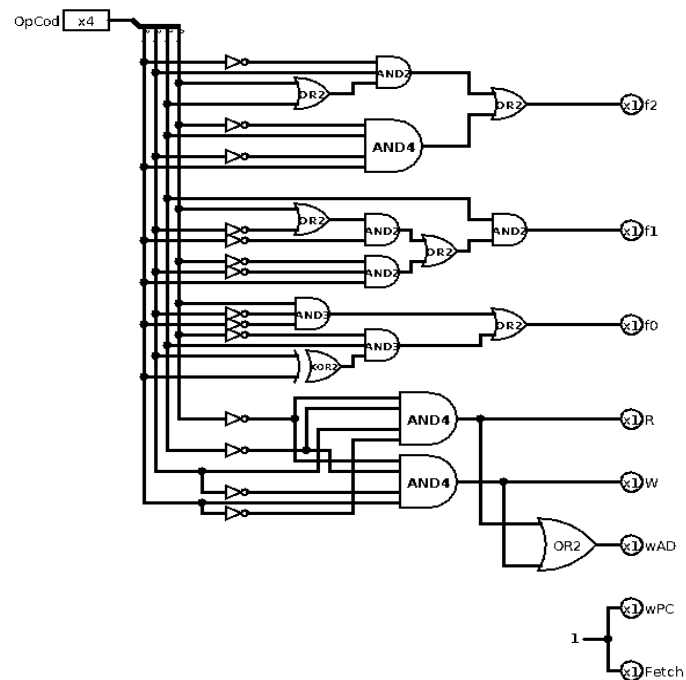


FIGURE 16 – Circuit logique du décodeur d'instructions

### 5.4 registre 12bits

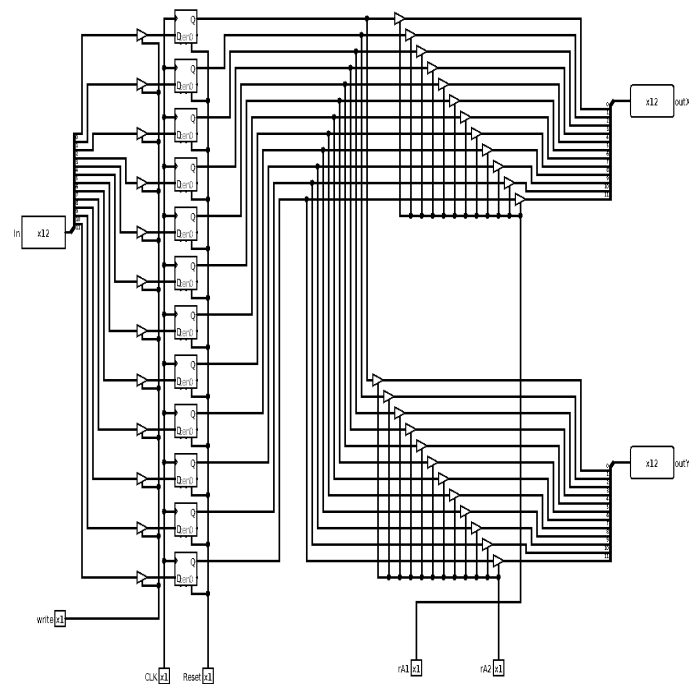


FIGURE 17 – Circuit logique du registre 12bits



## 5.5 CPU

le CPU est donc l'assemblage de l'ALU, le banc de registre, l'unité de contrôle, l'unité d'adressage.

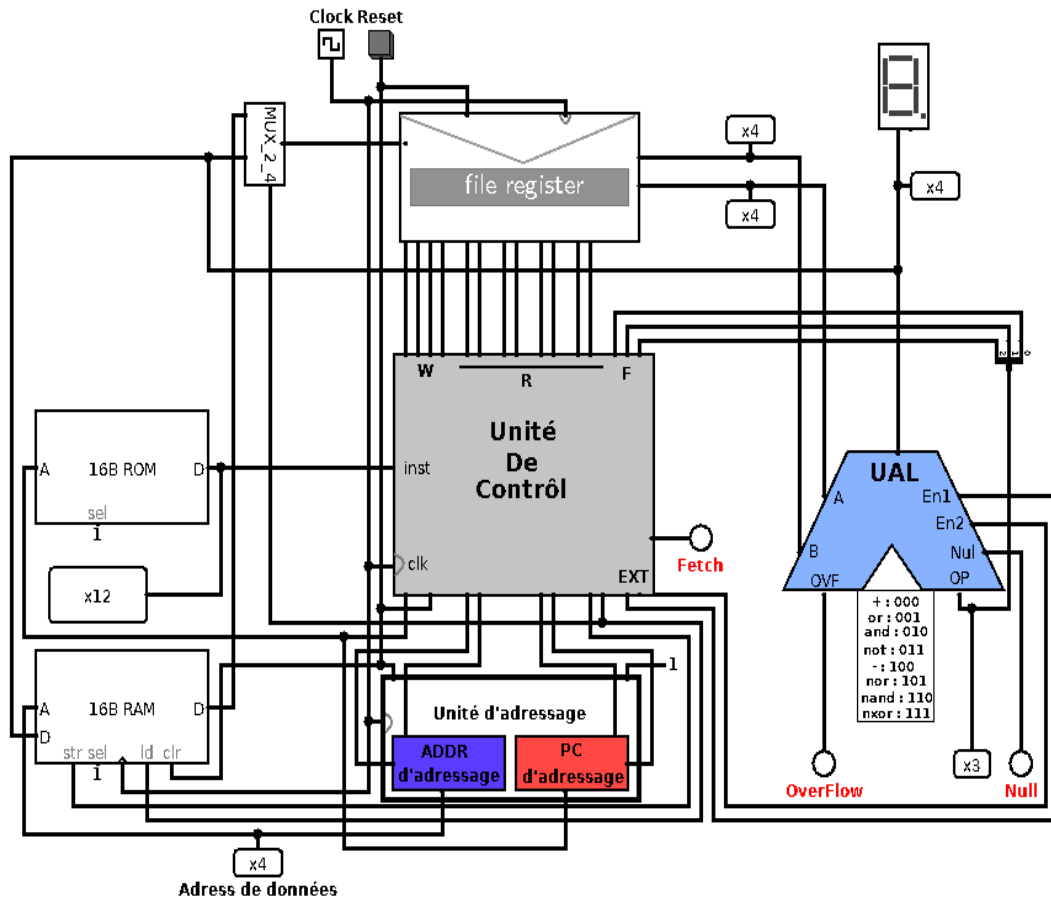


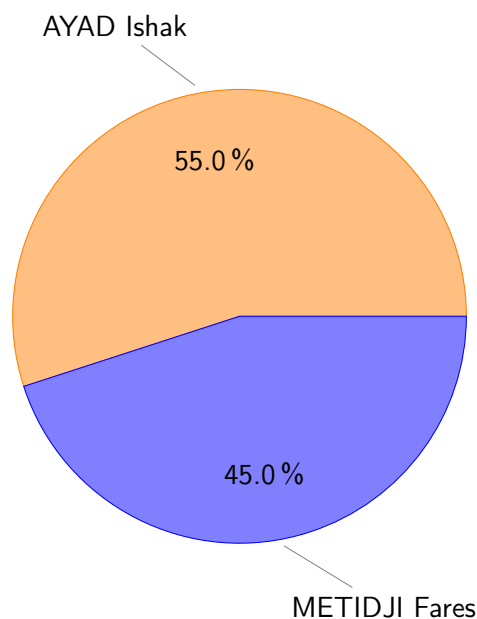
FIGURE 18 – Vu d'ensemble sur le CPU

## 6 Déroulement du projet

**Chapeau** Dans cette section, nous décrivons comment le projet a été réalisé en équipe : la répartition des tâches, la synchronisation du travail entre membres de l'équipe, etc.

### 6.1 Répartition des tâches

AYAD Ishak	METIDJI Fares
UAL	UAL
Banc de registre	Unité d'adressage
Unité de contrôle	
Modifications personnelles	



### 6.2 Synchronisation du travail

- Pour la réalisation du projet nous avons utilisé le service web d'hébergement et de gestion de développement de logiciels GitHub pour synchroniser nos travaux. Ce dernier nous a permis d'effectuer des commites à un rythme élevé au commencement du projet pour que tous les membres de l'équipe aient accès aux bases, puis après la répartition des tâches le rythme a diminué.
- Pour la rédaction de ce document nous avons utilisé la plateforme web Overleaf pour répartir les tâches, le rythme de rédaction a été très élevé à la fin du projet.

### 6.3 Problèmes rencontrés

1. L'implémentation de l'opération soustraction.
2. Réalisation de l'unité d'adressage.
3. La compréhension de l'utilité d'un registre d'adresse de données.
4. La compréhension de certains signaux de sorties de l'UC (EXT).
5. Notre bascule D avait un problème donc nous avons utilisé le registre fourni par Logisim.

## 6.4 Calendrier

Semaine	AYAD Ishak	MATIDJI Fares
09-AVRIL	-Concertation,	-Concertation.
16-AVRIL	-éléments de bases (MUX,DEC..),	-éléments de bases (MUX,DEC..),
23-AVRIL	-UAL,	-UAL,
30-AVRIL	-Banc de registre,	-Unité d'adressage,
7-MAI	-Unité de contrôle,	
14-MAI	-Modifications personnelles,	
20-MAI	-Concertation,	-Concertation,

FIGURE 19 – Calendrier

## 7 Conclusion

Au final, nous avons donc réalisé une simulation d'un processeur CPU de quatre bits, en prenant d'effectuer quelque changement d'information entre une machine et sa mémoire.

D'un point de vue de réalisation, nous avons réalisé le principe du processeur avec tous les éléments et composants avec leur exigence fonctionnelles. Nous avons pu approfondir certains points.

Grâce à ce projet, on a pu découvrir et étudier de près les processeurs, mais surtout ce projet nous a permis de manipuler les différentes instructions machines permettant de piloter le processeur.