

Projet d'architecture des ordinateurs

Conception d'un processeur

Table des matières

I Introduction.....	3
II Cahier des charges.....	4
III Rôle et Réalisation.....	5
1 ALU.....	5
2 Banc de registres.....	7
3 Unité d'adressage.....	7
4 Unité de contrôle.....	9
IV Processeur.....	12
V Conclusion.....	13
VI Annexes.....	14
VII Annexe.....	15

I Introduction

Ce projet est réalisé dans le cadre de la deuxième année de licence en informatique à l'Université de Cergy-Pontoise en cours d'architecture des ordinateurs. Il a pour but de réaliser un processeur 4bits en utilisant les différentes notions vues en cours. J'ai réaliser ce projet sans binôme.

Ce projet est réalisé sous la tutelle de J. J.Lorandel, professeur de l'Université de Cergy-Pontoise.

Pour ce projet nous utilisons le logiciel Logisim pour la conception et le test du CPU. Étant donné que je suis tous seul, je n'ai pas eu besoin d'utiliser de logiciel de partage de fichier tel que GitHub ou Dropbox. Enfin, les sources que j'ai utilisé pour réaliser sont le cours magistraux du professeur J.Lorandel ainsi que quelques vidéo sur youtube pour la compréhension de l'utilisation de certains composant.

II Cahier des charges

Objectif : Réalisation d'un processeur 4Bits sur l'outil Logisim© Éléments du processeur :

- Une unité arithmétique et logique (ALU)
- Un banc de 4 registres généraux
- Une unité d'adressage
- Une unité de contrôle
- Deux mémoires d'instructions et de données (ROM – RAM) (pas de conception)

Chacun de ces éléments doivent posséder les spécificités suivantes :

- ALU : possibilité d'effectuer 8 opérations différentes.
- Banc de registres : 4 registre de 4 bits ayant chacun :
 - 2 sorties de 4bits
 - 1 entrée de 4bits
 - 2 signaux de contrôle de lecture d'un bit chacun
 - 1 signal de contrôle d'écriture d'un bit
 - 2 ports d'extension et une horloge
- Unité d'adressage :
 - 2 registres AD et PC et pour chaque registre :
 - 2 ports d'extension pour les ajouts d'extension.
- Unité de contrôle :
 - Réaction attendu à la réception d'un signal RESET.

III Rôle et Réalisation

1 ALU

L'ALU (Arithmetical and Logical Unit) est l'unité en charge de faire les calculs. Elle prend un mot de 4bits en entrée A et un autre mot de 4bits en entrée B pour ensuite effectuer avec ces mots une des 8 opérations disponibles. Le choix de l'opération se fait par 3 entrée de sélection sur 1bits selon le tableau suivant :

Opération	Code
ADD	000
OR	001
AND	010
NOT B	011
SUB	100
XOR	101
NAND	110
NOR	111

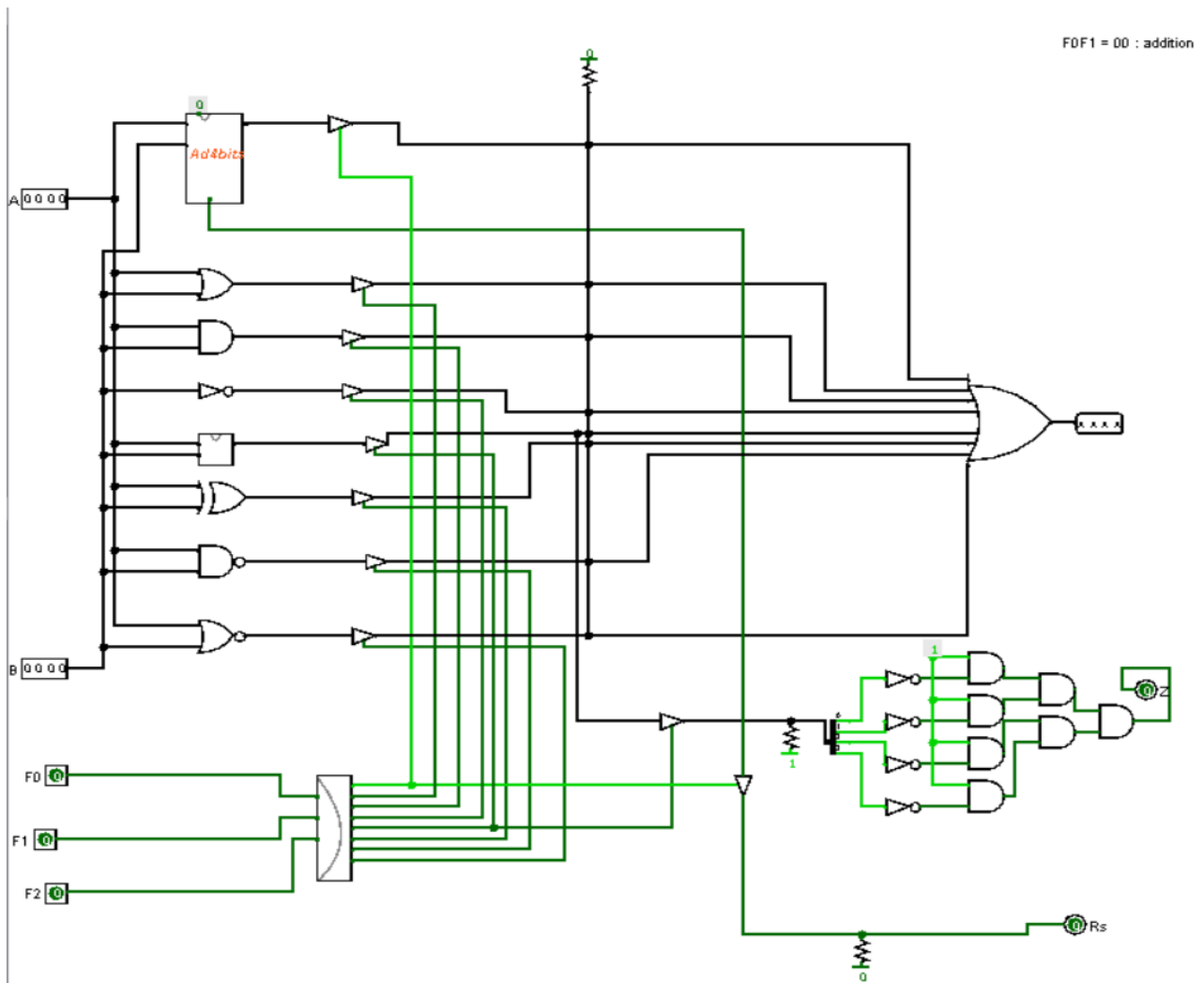


Illustration 1: ALU 4bits

2 Banc de registres

Le banc de registres 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 les diriger soit vers la branche A soit vers la branche B de l'ALU pour qu'un calcul soit effectué sur celles-ci. Pour cela il prend un mot de 4bits en entrée qu'il va stocker dans l'un de ses 4 registres en fonction de la sélection qui se fait sur une entrée Sel sur 2bits. Chaque registre peut alors envoyer la donnée qu'il stocke soit vers la sortie X (vers la branche A de l'ALU) soit vers la sortie Y (vers la branche B de l'ALU), on a donc 2 bits de sélection par registre.

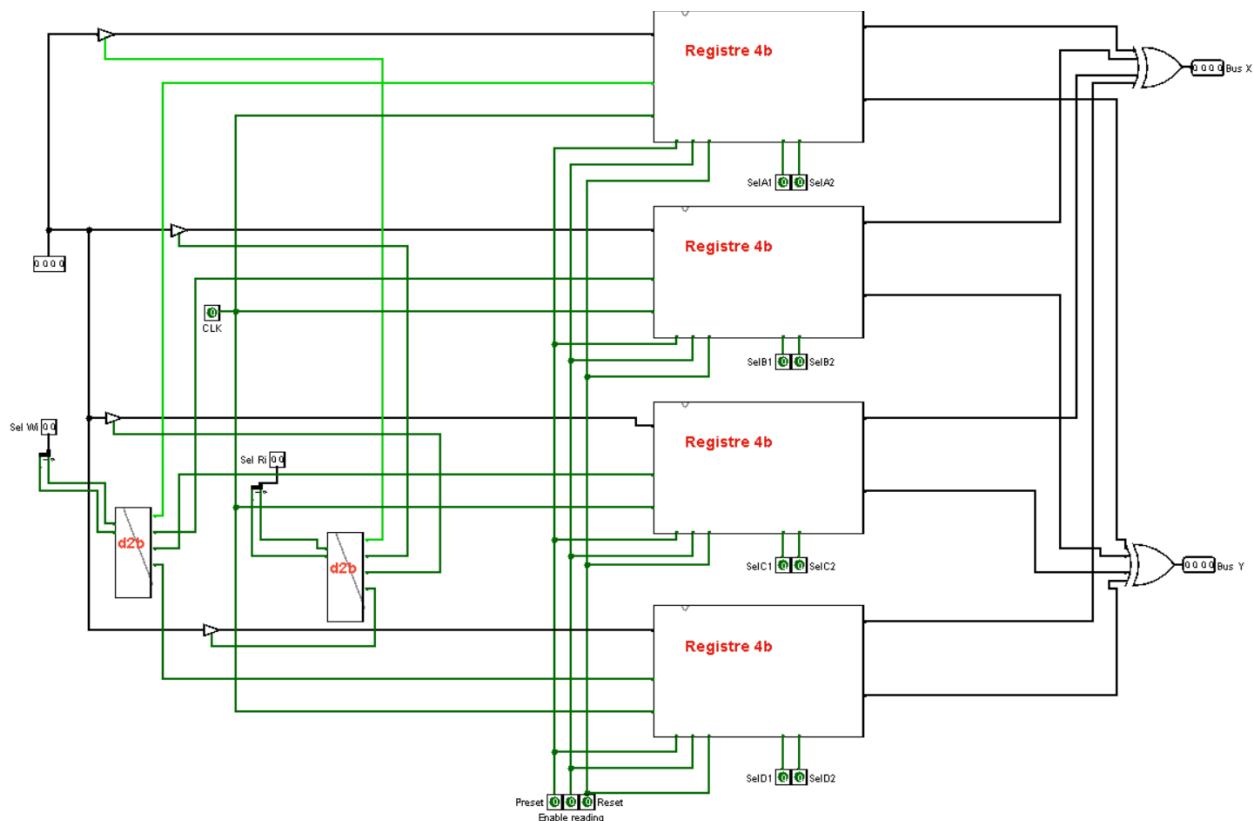


Illustration 2: Banc de registre 4 bits

3 Unité d'adressage

L'unité d'adressage contient 2 registres 4bits : le PC (Program Counter) et le AD (Adress Data). Le PC est le registre contenant l'adresse de la mémoire d'instruction qui est active, c'est-à-dire l'adresse contenant l'instruction traitée par le CPU. Le AD, lui, 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.

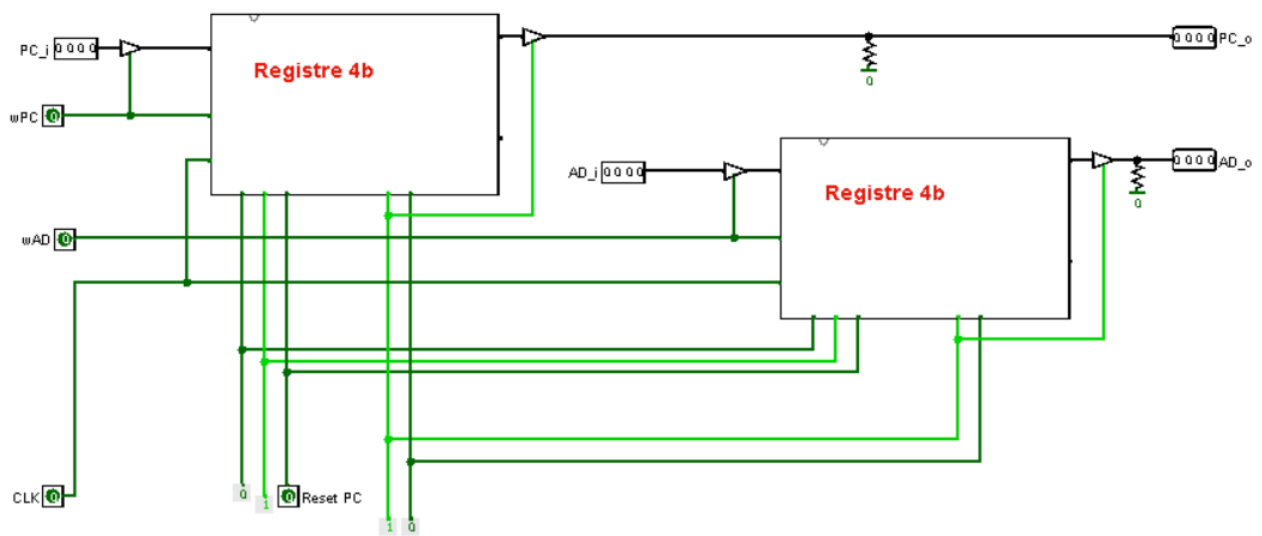


Illustration 3: Unité d'adressage

4 Unité de contrôle

L'unité de contrôle est l'unité principale du CPU qui reçoit les instructions de la mémoire d'instruction et pilote le chemin des données dans le reste du CPU grâce à l'activation de signaux de contrôle : la sélection de l'opération réalisée par l'ALU, la sélection des registres dont les contenus sont lus et écrits, et les indicateurs d'accès mémoire Read, Write et Fetch. Elle 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 initiale Boot. Les instructions peuvent avoir 2 formats : registre ou immédiat. Elles sont décomposées comme suit :
Format registre :

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

Où :

- OPCODE: code de l'opération à réaliser
- RES : sélection du registre où sera stocké le résultat de l'opération
- X : sélection du registre contenant l'opérande X (envoyée vers l'entrée A de l'ALU)
- Y : sélection du registre contenant l'opérande Y (envoyée vers l'entrée B de l'ALU)
- EXT : extension possible des instructions

Format immédiat :

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

Où :

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

Le jeu d'instructions du 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	Immédiat
STORE	1000	Immédiat
SUB	0101	Registre
XOR	0110	Registre
NAND	0111	Registre
NOR	1010	Registre

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 (f2 f1 f0)	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	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
XOR	0110	101	0	0	0	1	1
NAND	0111	110	0	0	0	1	1
NOR	1010	111	0	0	0	1	1

Cette activation dépend d'un décodeur d'instruction décodant le code opération et activant les sorties correspondantes. (voir annexes pour le circuit du décodeur d'instruction) L'activation des signaux de lecture (et d'écriture) dans le banc de registres est pilotée comme suit :

Source ou destination	Code	A	B	C	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

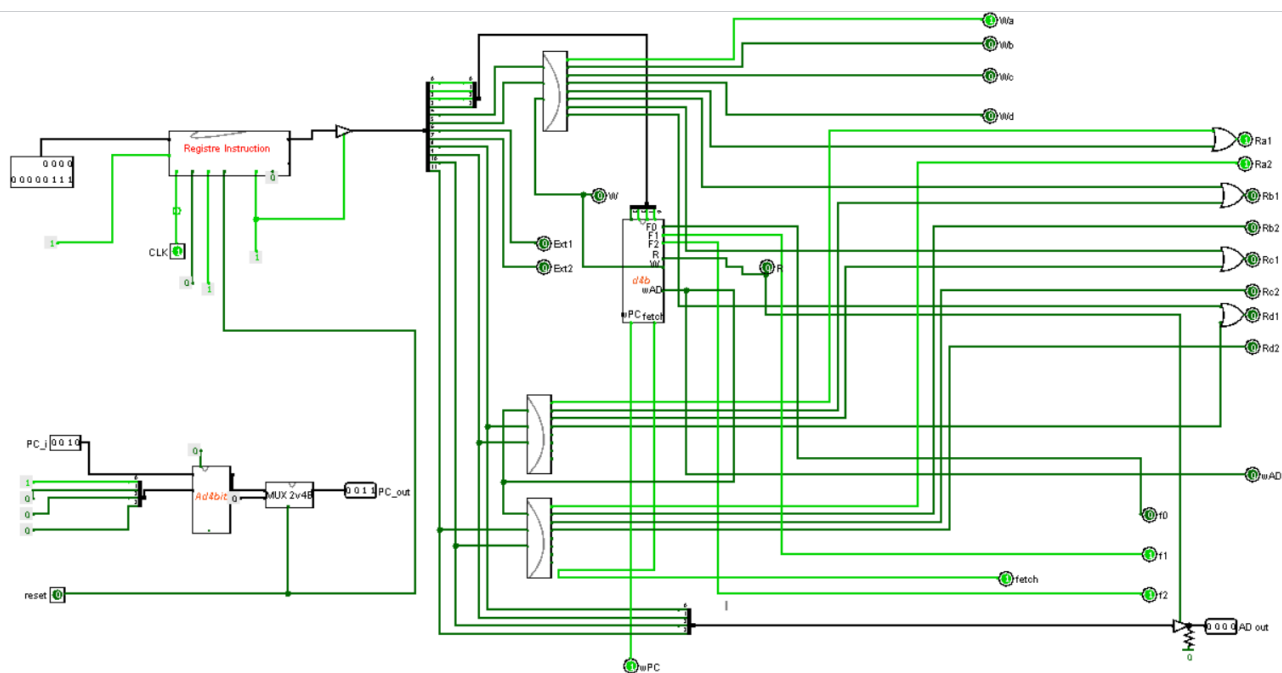


Illustration 4: Unité de contrôle

IV Processeur

Le CPU correspond l'assemblage de l'ALU, du banc de registre, de l'unité d'adressage et de l'unité de contrôle selon le schéma suivant :

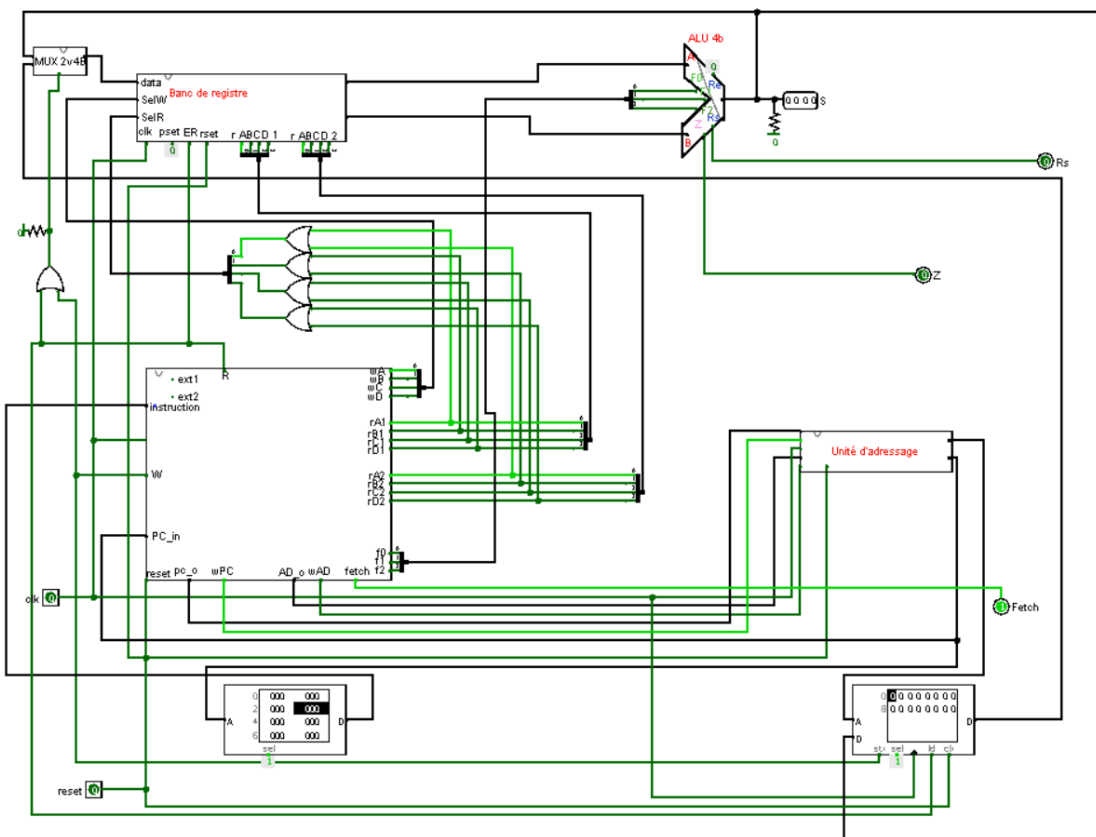
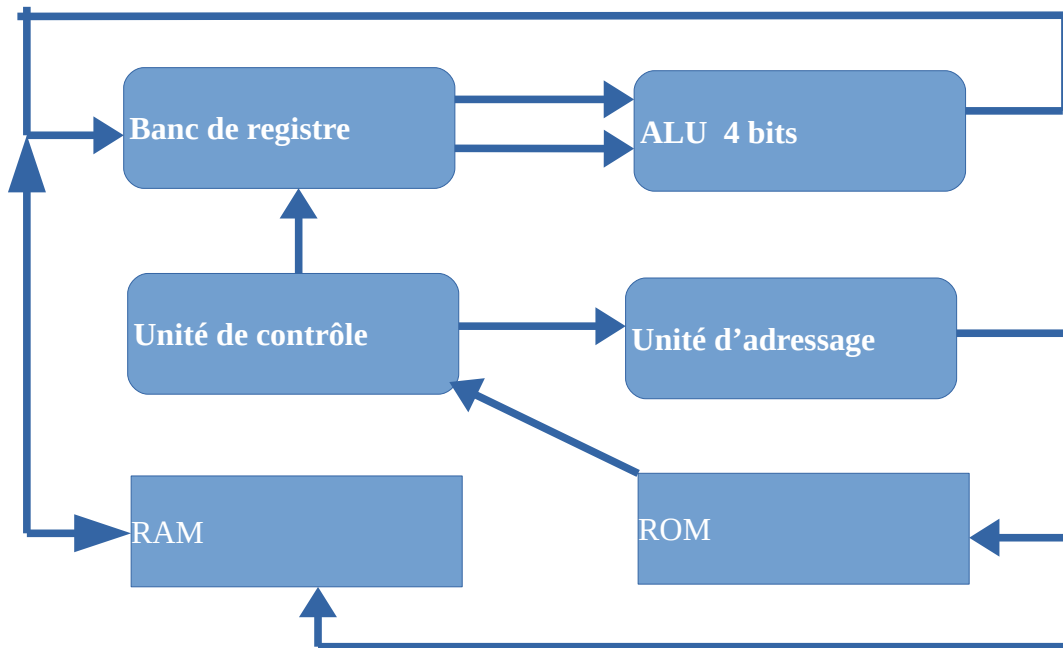


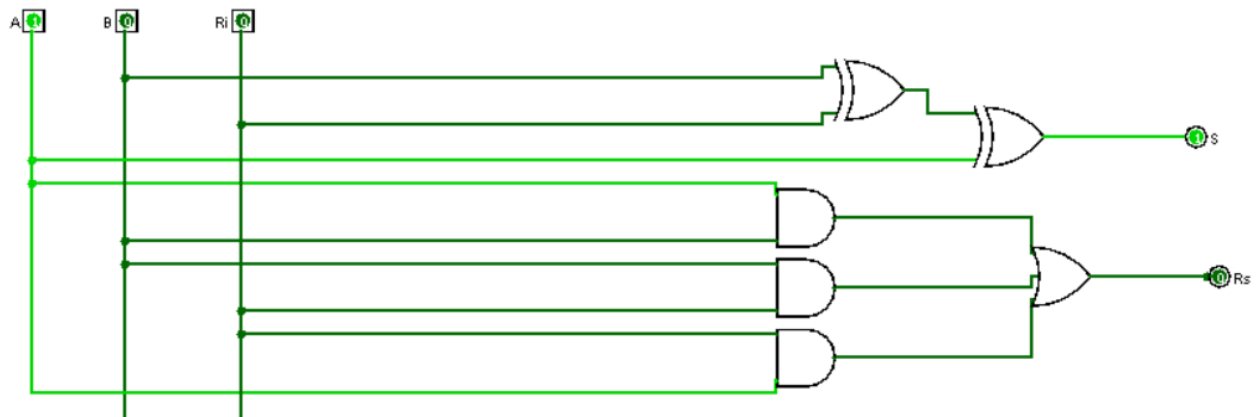
Illustration 5: CPU

V Conclusion

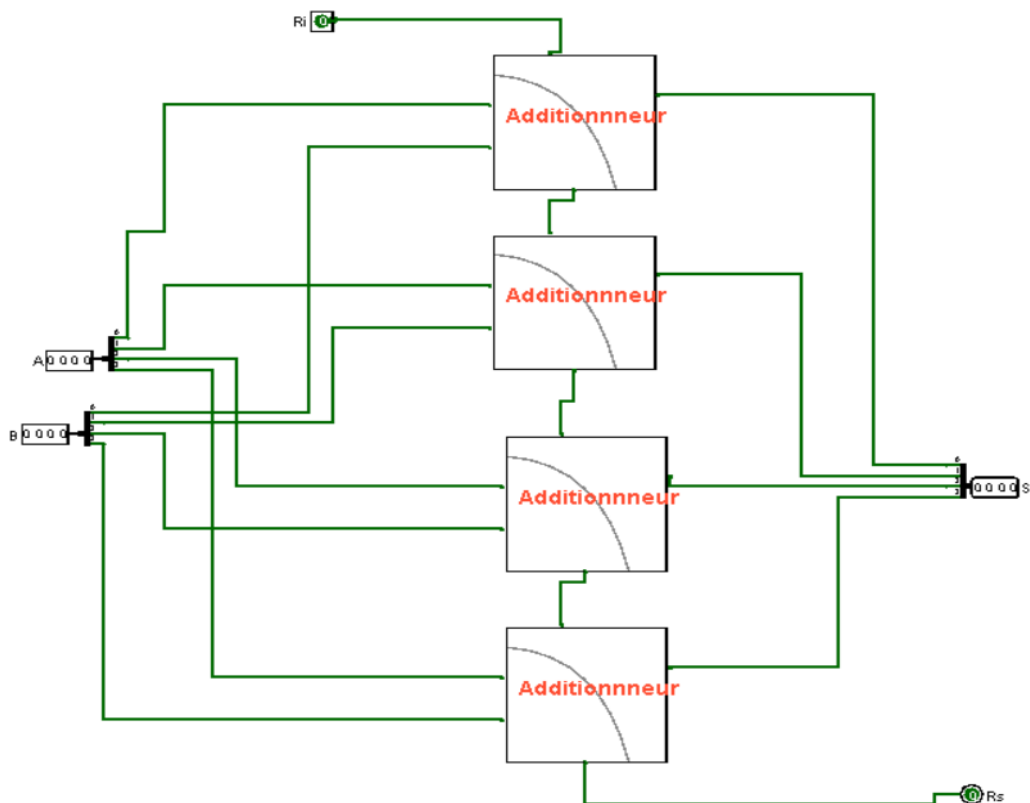
J'ai finalement pu réussir à faire un sorte que les éléments de mon CPU fonctionne parfaitement. Cependant, j'ai éprouvé des difficulté à utiliser la ROM, étant donné que son utilisation se faisait en hexadécimal. Pour améliorer mon CPU, je pense que je pourrais utiliser plus de splitter, ce qui peut me permettrait de mieux me retrouver dans l'exploration des composants et du CPU, dans l'optique de pouvoir apporter d'autres extension à mon CPU, tel que l'ajout d'instruction de saut de branchement.

VI Annexes

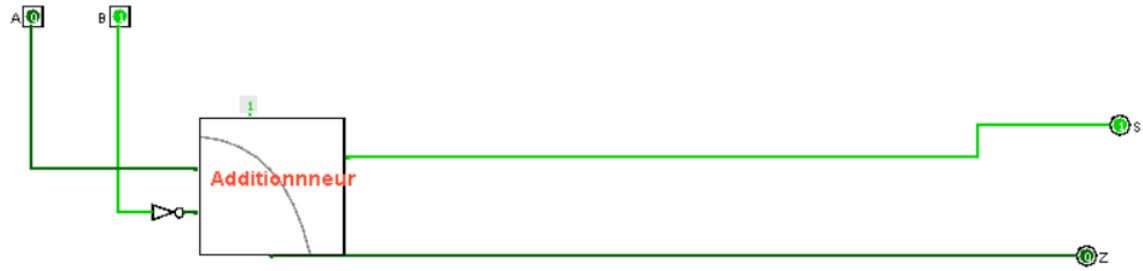
1 Additionneur complet 1 bits



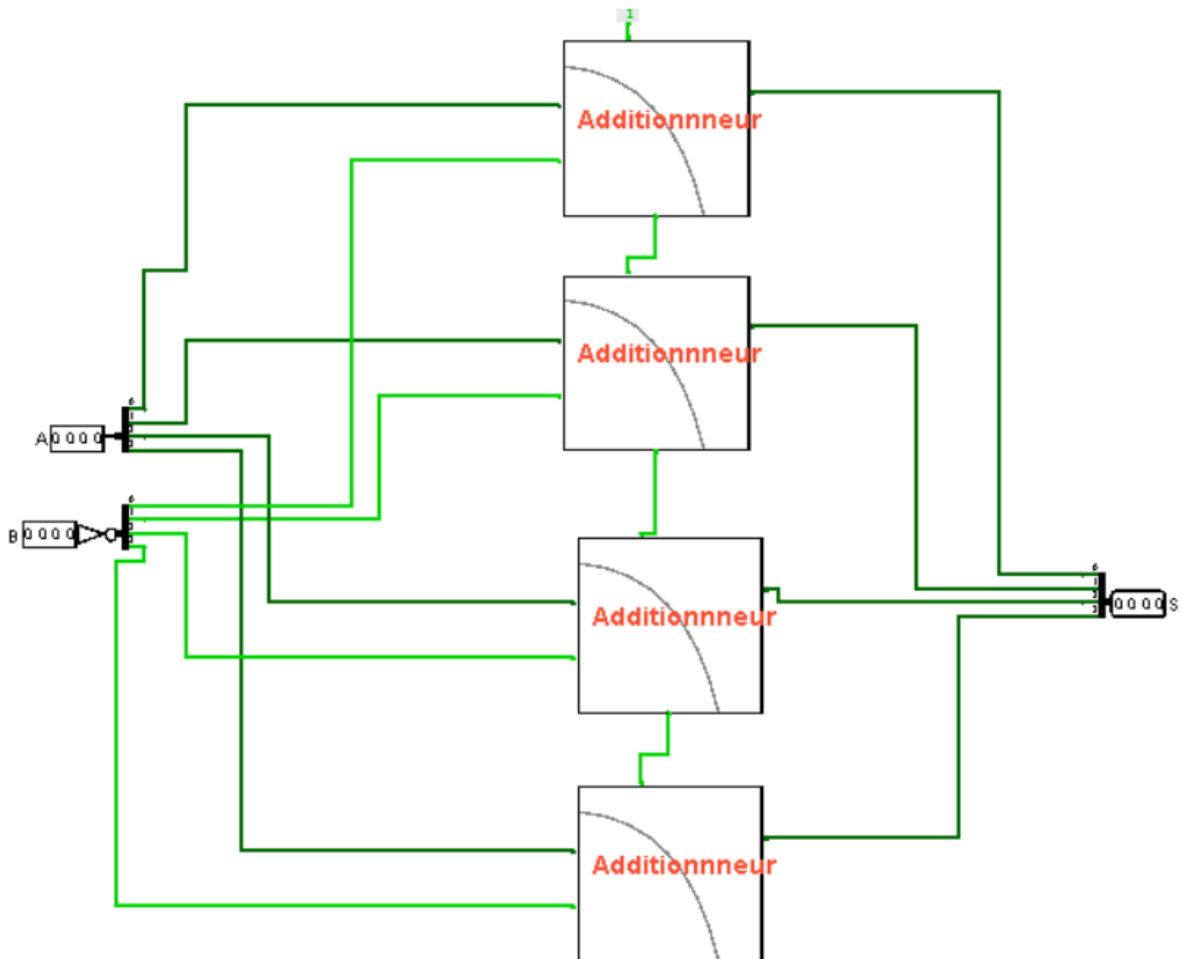
2 Additionneur complet 4 bits



3 Soustracteur 1 bits



4 Soustracteur 4 bits



5 Décodeurs

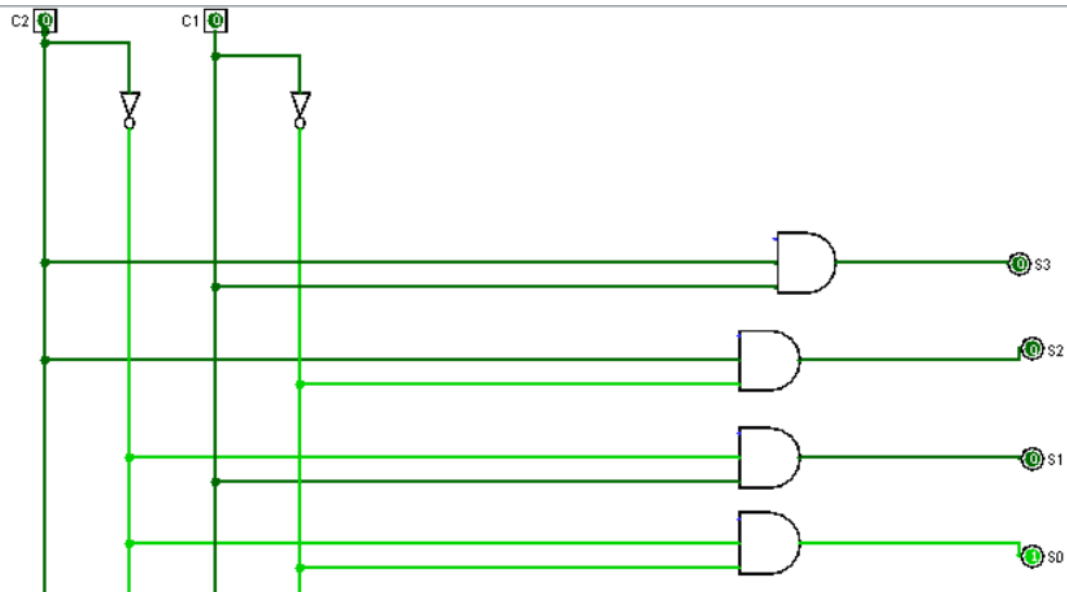


Illustration 6: Décodeur 2bits

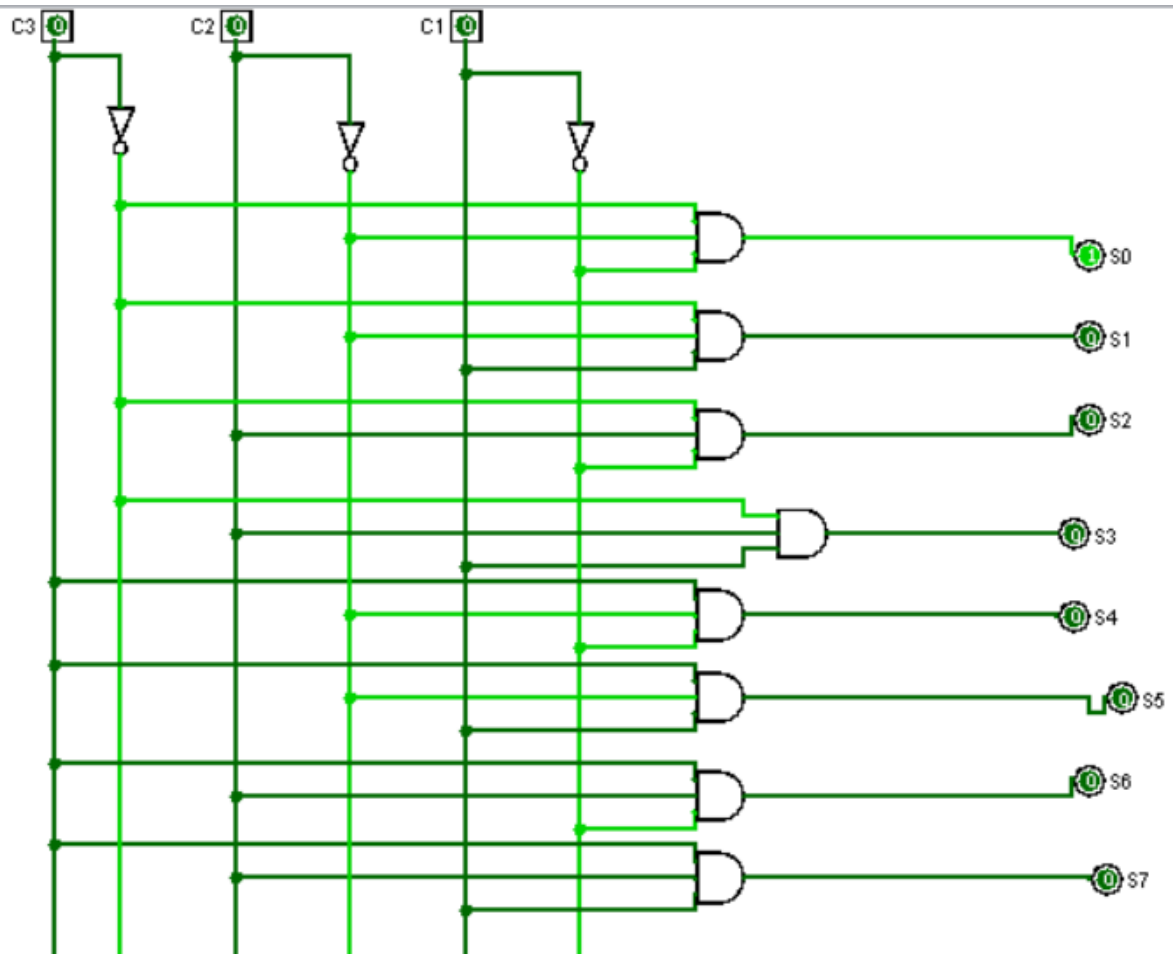


Illustration 7: décodeur 3 bits

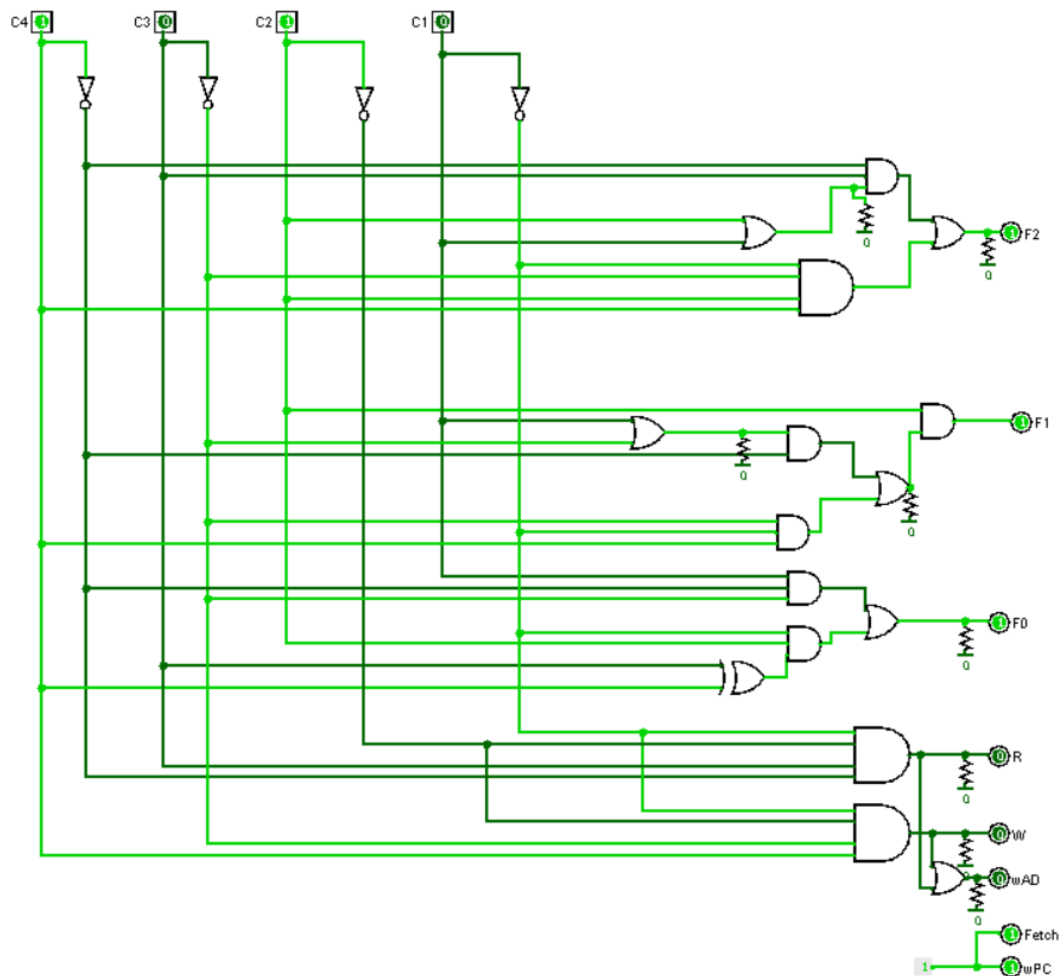


Illustration 8: Décodeur d'instruction sur 4bits

6 Registres 4 bits

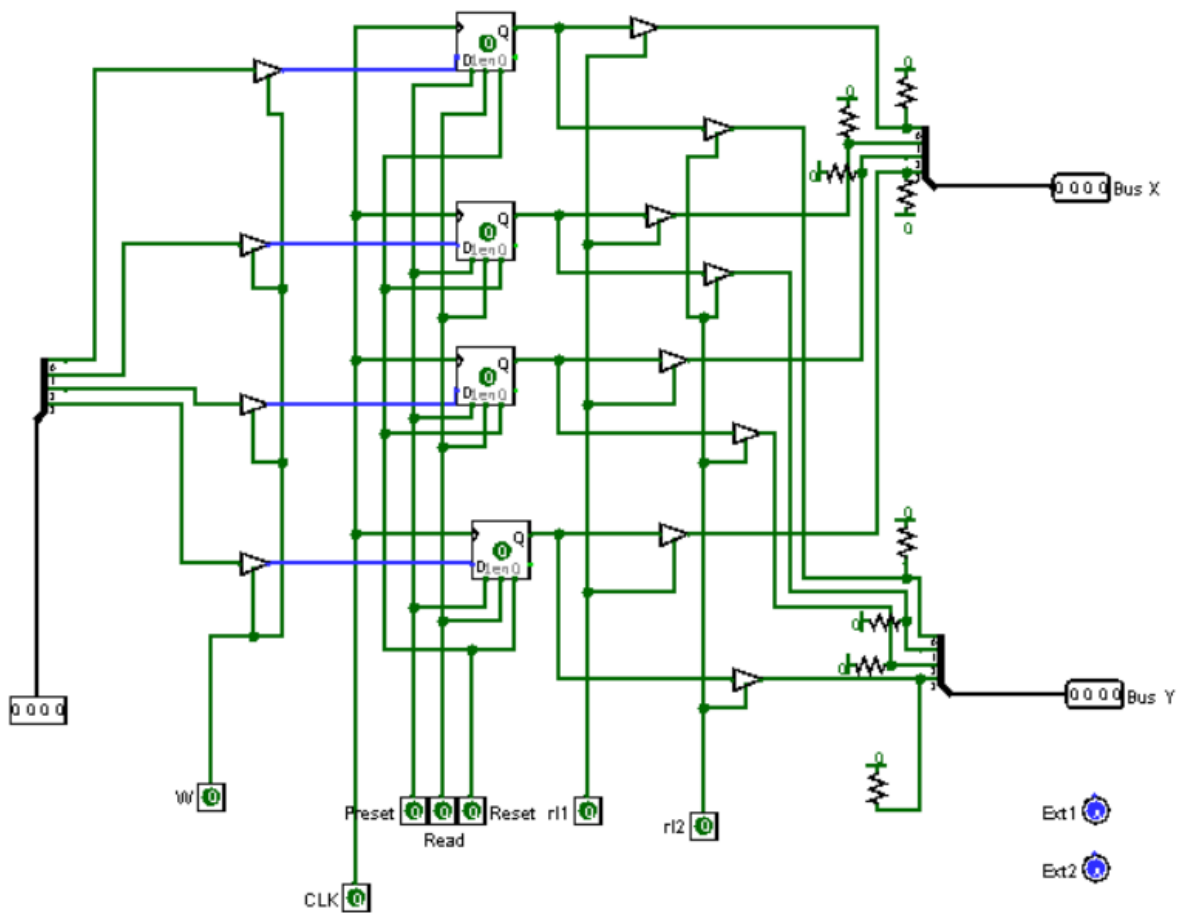


Illustration 9: Pour le registre d'instruction, le principe est le même que pour la conception d'un registre 4 bits, avec 12 bascules D au lieu de 4

VII Sources externe au cours

[Vidéo youtube exemple d'utilisation de la ROM](#)

[Vidéo youtube exemple d'utilisation RAM](#)