

CY Cergy Paris Université

# **RAPPORT**

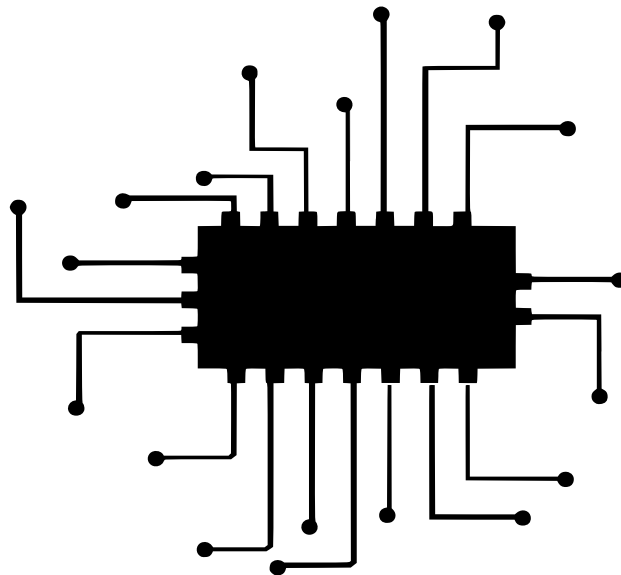
pour le projet d'Architecture des Ordinateurs

“Conception d'un processeur”

**Licence d'Informatique deuxième année**

rédigé par

**Adel ABBAS, Kévin BERNARD**



Mai 2020

## Table des matières

<b>1 Spécifications du processeur.....</b>	<b>3</b>
<b>2 Organisation du processeur.....</b>	<b>4</b>
2.1 Organisation générale.....	4
2.2 Unité arithmétique et logique.....	5
2.3 Banc de registres.....	6
2.4 Registre d'instruction.....	8
2.5 Unité d'adressage.....	9
2.6 Unité de contrôle.....	10
<b>3 Jeu d'instruction du processeur.....</b>	<b>14</b>
3.1 Format Registre.....	14
3.2 Format Immédiat.....	14
<b>4 Améliorations apportées.....</b>	<b>15</b>
4.1 Extension des opérations de l'UAL.....	15

## Index des figures

Figure 1: Circuit principal du processeur.....	4
Figure 2: Central Process Unit.....	4
Figure 3: UAL de 1 bit.....	5
Figure 4: UAL de 4 bits.....	5
Figure 5: Banc de registres.....	7
Figure 6: Registre de 4 bits.....	8
Figure 7: Registre d'instruction.....	8
Figure 8: Unité d'adressage.....	9
Figure 9: Unité de contrôle.....	11
Figure 10: Aperçu de l'UAL de 1 bit.....	15
Figure 11: Circuit de l'inverseur ("Inverter").....	15
Figure 12: Signal Z de sortie au niveau de l'UAL.....	16

## Index des tableaux

Tableau 1: Tableau de correspondance sortie du décodeur / état.....	12
Tableau 2 :Tableau des mnémoniques d'instruction.....	14

## 1 Spécifications du processeur

Notre processeur est un processeur RISC, conçu selon une **architecture à registres**, plus précisément **chargement-rangement**, ce qui signifie que les opérandes sont dans des registres (excepté pour les instructions load et store). Il dispose de 4 registres généraux sur 4-bits.

De plus, la communication avec la mémoire programme et la mémoire de données s'effectue via deux bus distincts, ce qui correspond à une **architecture de Harvard**.

Le processeur supporte des **instructions** d'une **taille fixe** de 12-bits, et 2 formats d'instructions.

Le contrôle du chemin de données est **câblé**.

## 2 Organisation du processeur

### 2.1 Organisation générale

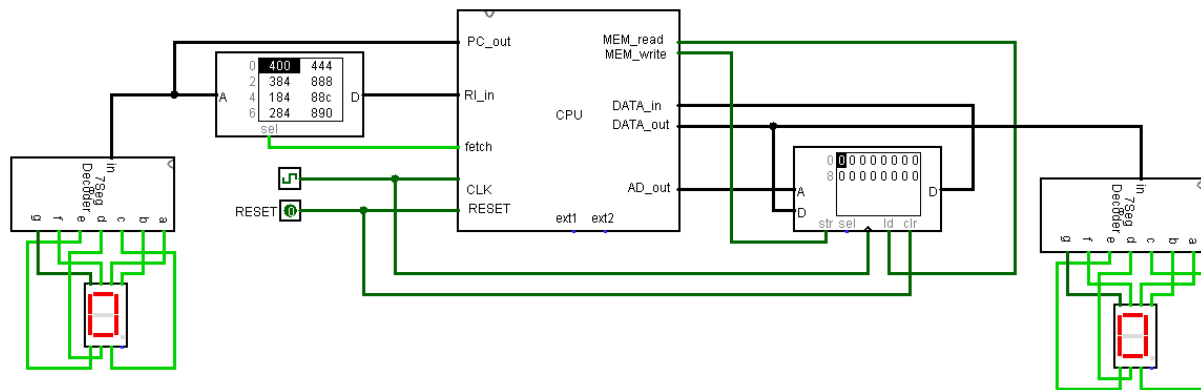


Figure 1: Circuit principal du processeur

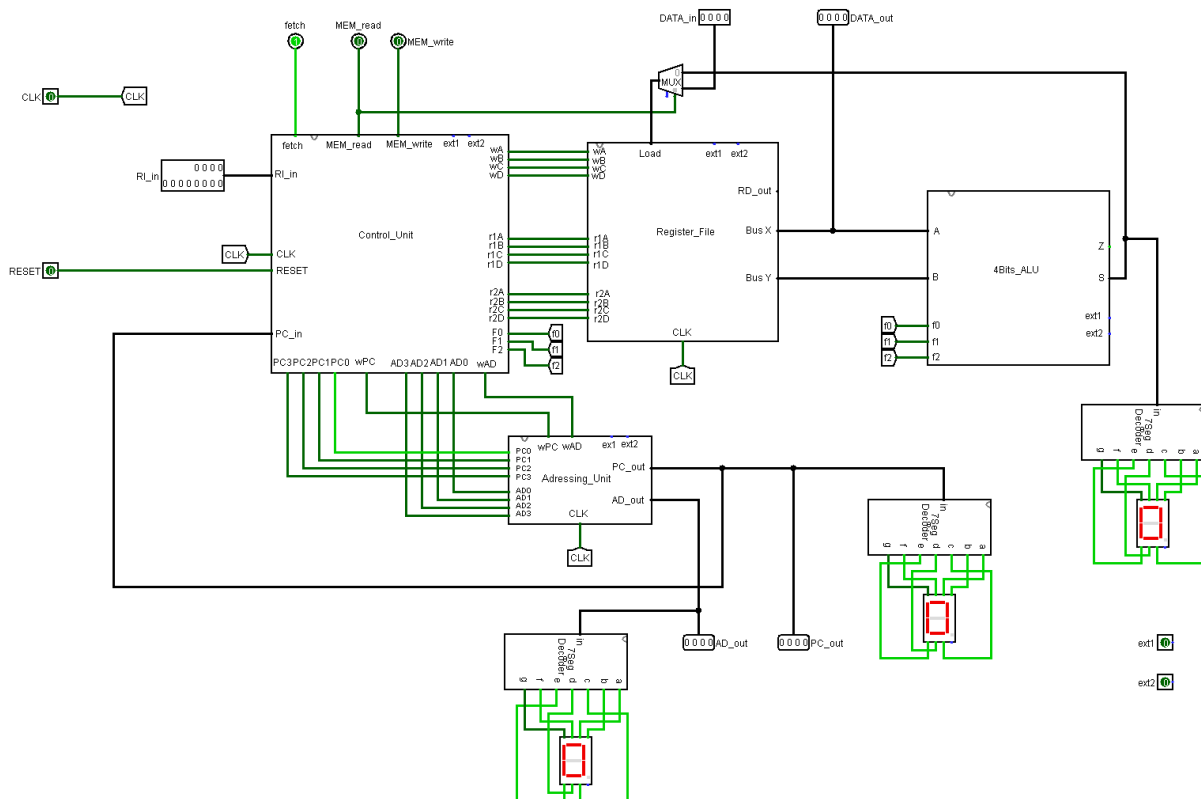


Figure 2: Central Process Unit

## 2.2 Unité arithmétique et logique

Notre Unité Arithmétique et Logique de 4 bits est composée de 4 UAL de 1 bit reliées en séquence.

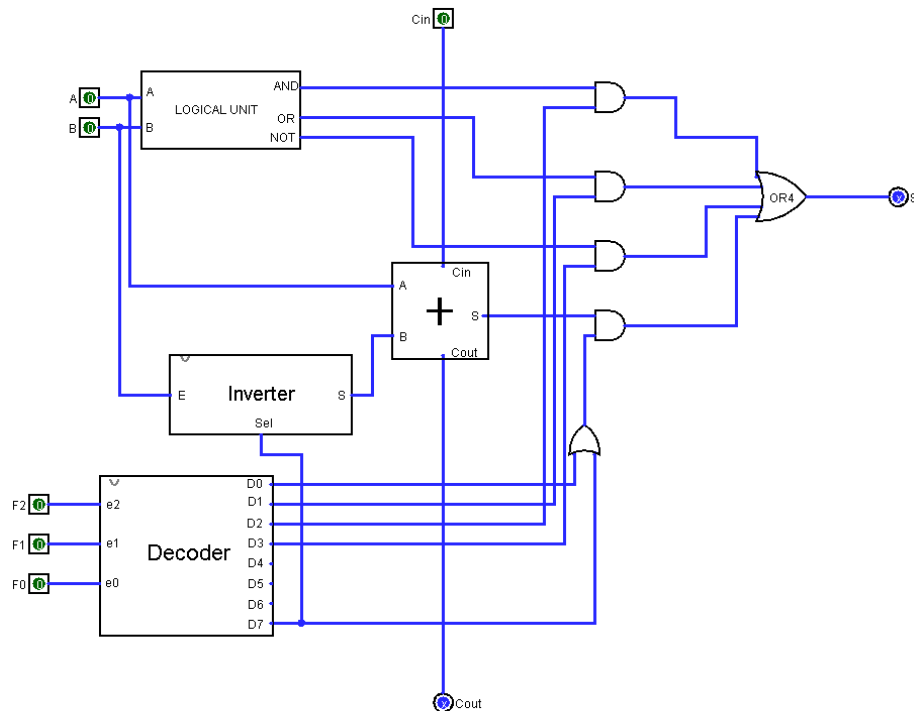


Figure 3: UAL de 1 bit

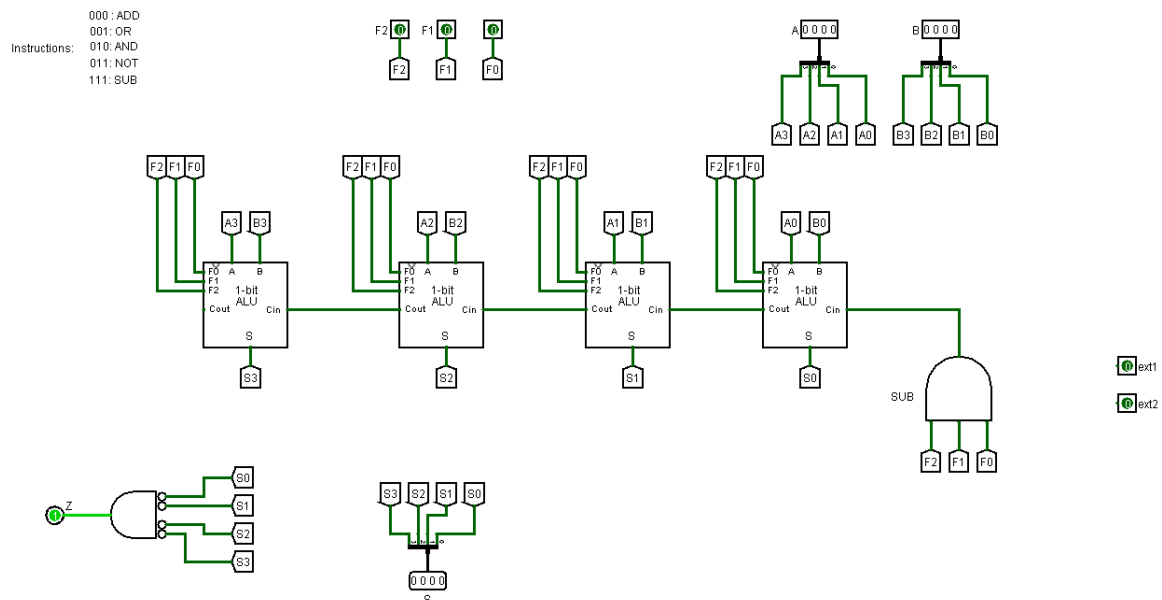


Figure 4: UAL de 4 bits

Elle effectue les opérations suivantes:

### **Opérations logiques**

Les opérations logiques sont effectuées bit à bit. Les opérations supportées sont le ET, le OU et la négation de l'entrée B.

### **Opérations arithmétiques**

L'UAL peut effectuer l'addition et la soustraction. Le mécanisme de soustraction sera présenté dans la partie sur les améliorations apportées.

## **2.3 Banc de registres**

Le banc de registre est composé de 4 registres 4 bits :A, B, C et D. Son interface propose notamment :

- 1 signal de contrôle en écriture par registre
- 2 signaux de contrôle en lecture par registre
- 1 entrée de donnée sur 4 bits
- 2 sorties de données générales sur 4 bits
- 1 sortie de données sur 4 bits spécifique au registre D

Des résistances de pull-down sont placées au niveau des sorties de données pour prévenir les erreurs.

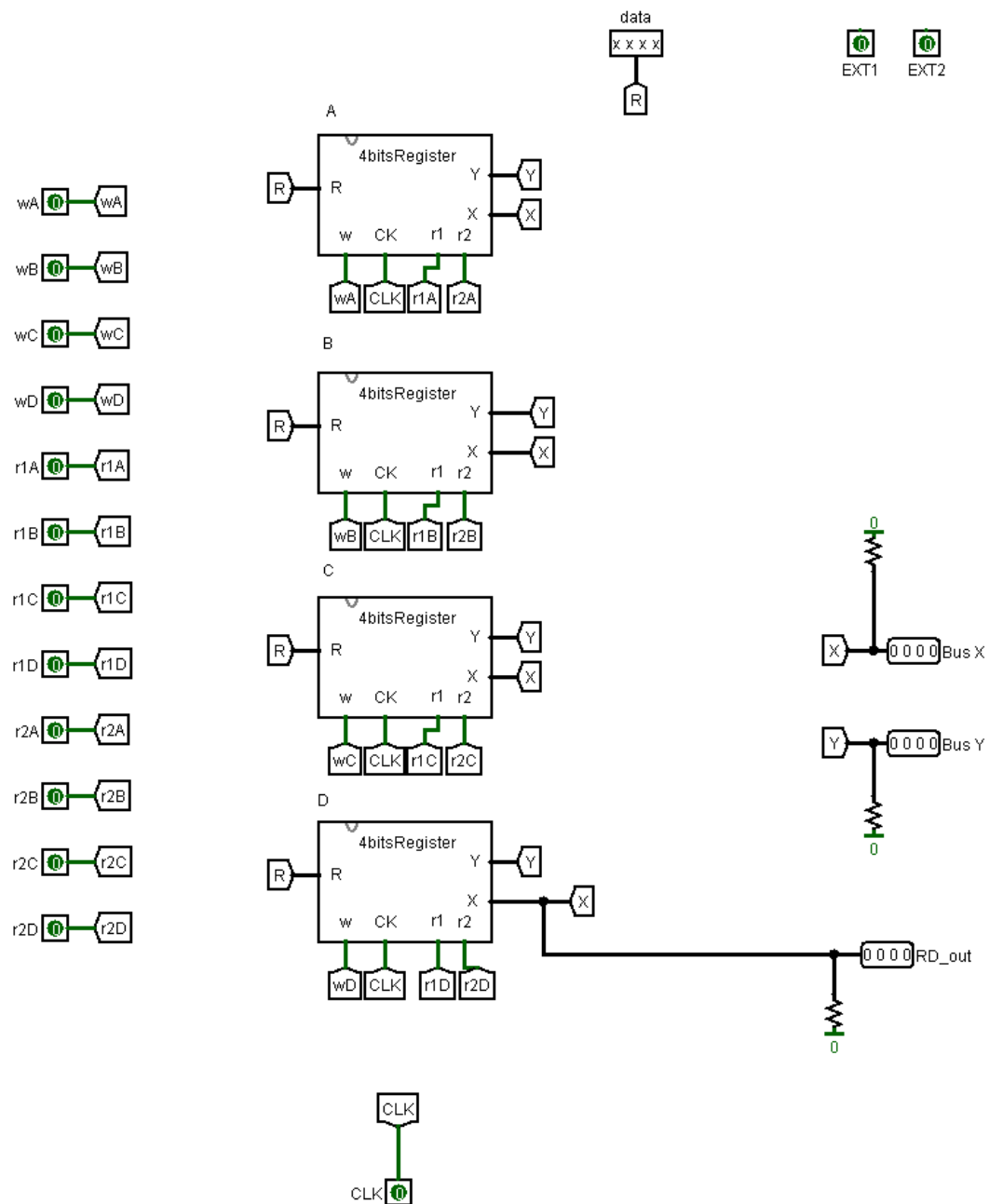


Figure 5: Banc de registres

### Registre 4 bits

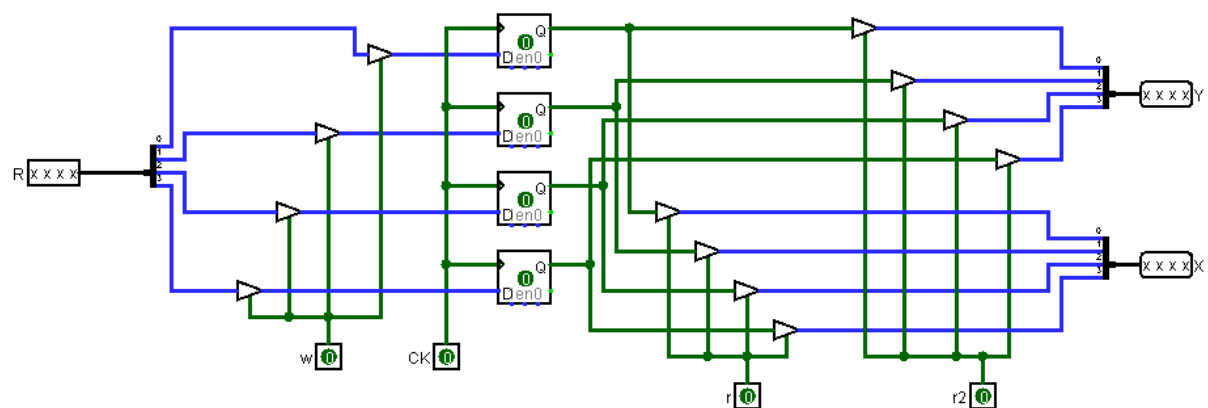


Figure 6: Register de 4 bits

Des portes 3 états permettent le contrôle des accès en lecture et en écriture.

## 2.4 Registre d'instruction

Le registre d'instruction a une taille de 12 bits, ce qui correspond au format d'instructions de notre processeur. Il est composé de 3 registres de 4 bits basés sur le modèle précédent.

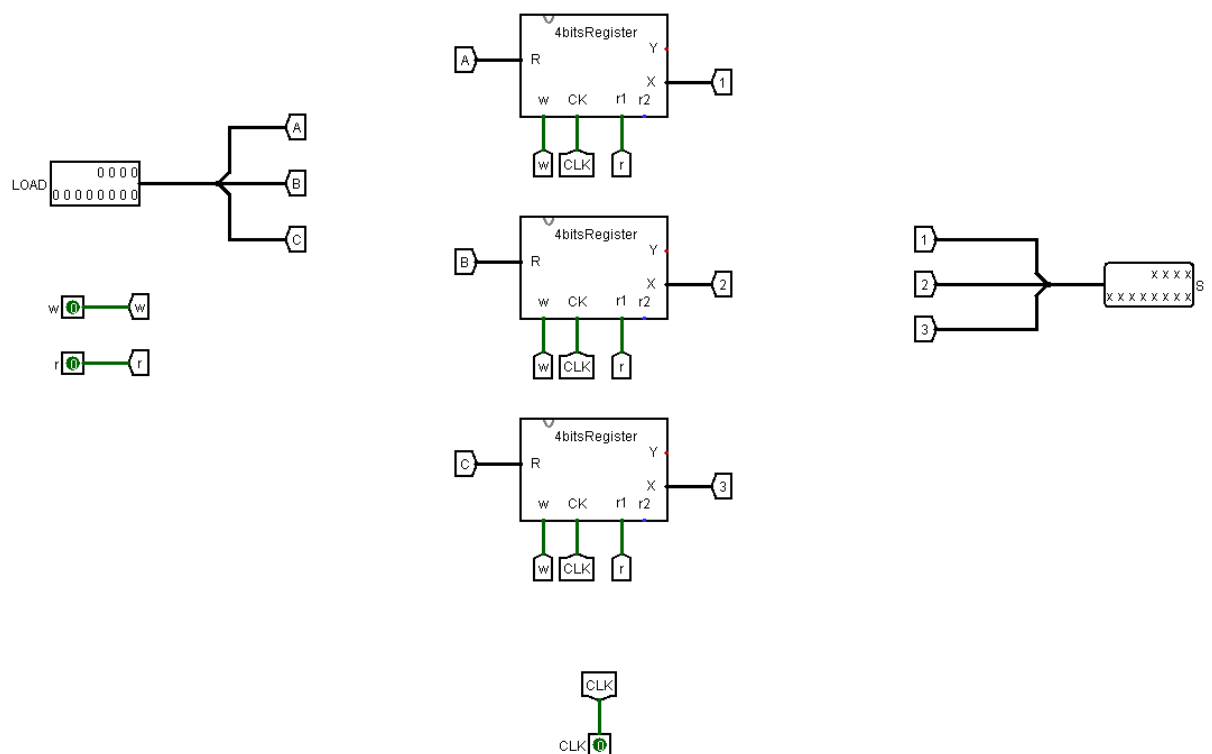


Figure 7: Registre d'instruction



### Signaux de contrôle

L'interface du registre d'instruction propose des signaux de contrôle pour la lecture et l'écriture.

### Sauvegarde dans le registre d'instruction

La donnée entrante sur 12 bits répartie entre les 3 registres de 4 bits à l'aide d'un splitter.

### Lecture dans le registre d'instruction

L'instruction sur 12 bits fournie en sortie est formée à partir des contenus des 3 registres de 4 bits et d'un splitter.

## 2.5 Unité d'adressage

L'unité d'adressage est composée de 2 registres de 4 bits: AD et PC

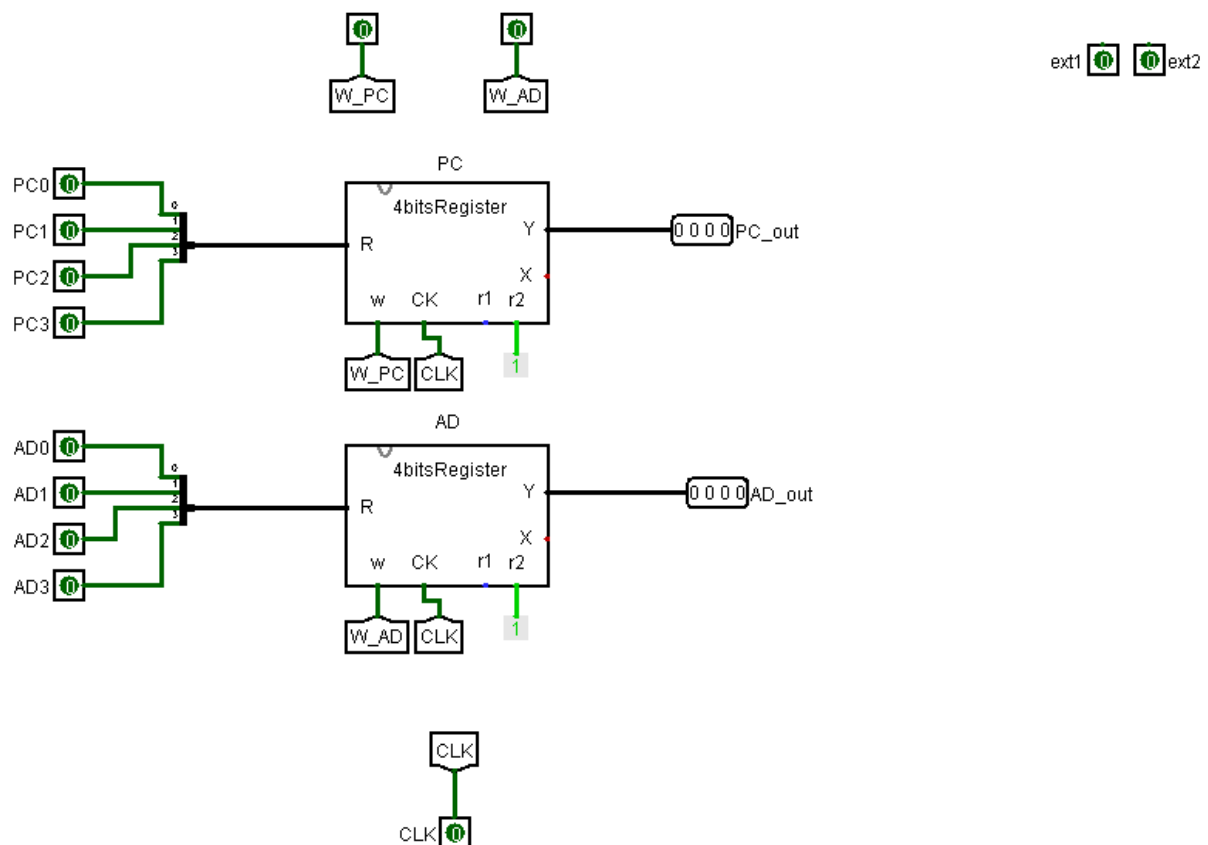


Figure 8: Unité d'adressage

### **Signaux de contrôle**

L'accès en lecture est toujours autorisé pour les 2 registres. L'accès en écriture est conditionné par un signal de contrôle.

### **Registre AD**

Le registre AD permet de stocker des adresses de données sur 4 bits. Ces adresses correspondent au format immédiat de données. Plus précisément, il s'agit des adresses des données en mémoire (RAM). Elles sont utilisées lors de l'exécution des instructions LOAD et STORE.

### **Registre PC**

Le registre PC correspond au compteur programme: il contient l'adresse d'une instruction dans la mémoire programme, sur 4 bits.

## **2.6 Unité de contrôle**

L'unité de contrôle réalise un automate à états finis. Ces états correspondent aux grandes étapes du cycle d'exécution d'une instruction : fetch, decode et execute.



## Séquenceur

Le séquenceur détermine à quelle étape du cycle d'instruction se trouve le processeur. Il est composé d'un compteur et d'un décodeur. Le compteur est réalisé avec un additionneur 2 bits et un registre 2 bits. La valeur du compteur est stockée dans le registre et incrémentée à chaque cycle d'horloge. Elle sert de signal de sélection du décodeur.

Sortie	Etat
0	1
1	2
2	2
3	3

Tableau 1: Tableau de correspondance sortie du décodeur / état

## Cycle d'exécution d'une instruction

### 1) Fetch

Lors de la phase de fetch (état 1), un signal de fetch est émis et le registre d'instruction est accessible en écriture. Le signal de fetch active la mémoire ROM qui renvoie une instruction sur 12 bits vers le registre d'instructions.

### 2) Decode

Les accès en lecture ou en écriture vers le banc de registre sont uniquement accordés lors de l'état 2.

Le mode de décodage est conditionné par le format de l'instruction présente dans le registre d'instruction. Le format est déterminé grâce au code opératoire:

S'il correspond à l'une des instructions LOAD ou STORE, l'instruction est au format immédiat, sinon, elle est au format registre.

En format registre, l'instruction fournit les adresses du résultat et des opérandes. Ces adresses servent de signal de sélection à 3 décodeurs qui activent respectivement l'écriture, la lecture de l'opérande A et la lecture de l'opérande B au niveau des registres correspondant. Les trois premiers bits du code opératoire servent de signaux de commande pour l'UAL.

En format immédiat, l'écriture dans le registre AD est autorisée et l'adresse de données présente dans l'instruction y est envoyée. Si le code opératoire correspond à l'instruction LOAD, un accès en écriture au registre de destination est accordé. Si le code opératoire correspond à l'instruction STORE, un accès en lecture vers le registre source est accordé.

### 3) Execute

A l'issue de l'exécution d'une instruction, l'accès en écriture vers le registre PC est autorisé et la valeur qu'il contient est incrémentée.

### **Réaction au signal RESET**

Si le signal RESET est perçu, l'accès en écriture au registre PC est activé et sa valeur est écrasée par une adresse câblée d'initialisation. En effet, la donnée envoyée vers le registre PC d'abord sélectionnée grâce à un multiplexeur.

De plus, le contenu du séquenceur est réinitialisé.

### **Communication avec la mémoire**

L'accès en lecture vers la mémoire de données (MEM\_read) est autorisé uniquement pour l'instruction LOAD et l'accès en écriture (MEM\_write) uniquement pour l'instruction STORE.

### 3 Jeu d'instruction du processeur

Le code opératoire est sur 4 bits soit  $2^4 = 16$  instructions possibles.

Mnémonique d'instruction	Code opératoire	Format (Registre ou Immédiat)
ADD	0000	R
OR	0001	R
AND	0010	R
NOT	0011	R
LOAD	0100	I
STORE	1000	I
SUB	0111	R

Tableau 2 :Tableau des mnémoniques d'instruction

#### 3.1 Format Registre

Traitement des données

Opérations logiques

Opérations arithmétiques

#### 3.2 Format Immédiat

Opérations de transfert (entre la mémoire, les registres...)

Déplacement des données entre mémoire et registres

## 4 Améliorations apportées

### 4.1 Extension des opérations de l'UAL

L'objectif des modifications est de permettre au processeur de réaliser l'opération de soustraction.  $A - B = A + (-B)$ , en d'autres termes, pour soustraire B à A il suffit de lui ajouter le complément à 2 de B. Le complément à 2 d'un nombre binaire est obtenu en inversant chacun de ses bits puis en ajoutant 1 au résultat.

Nous avons donc ajouté un inverseur pour l'entrée B au niveau de l'UAL de 1 bit. Au niveau de l'UAL de 4 bits, une retenue entrante est propagée lors de l'opération de soustraction.

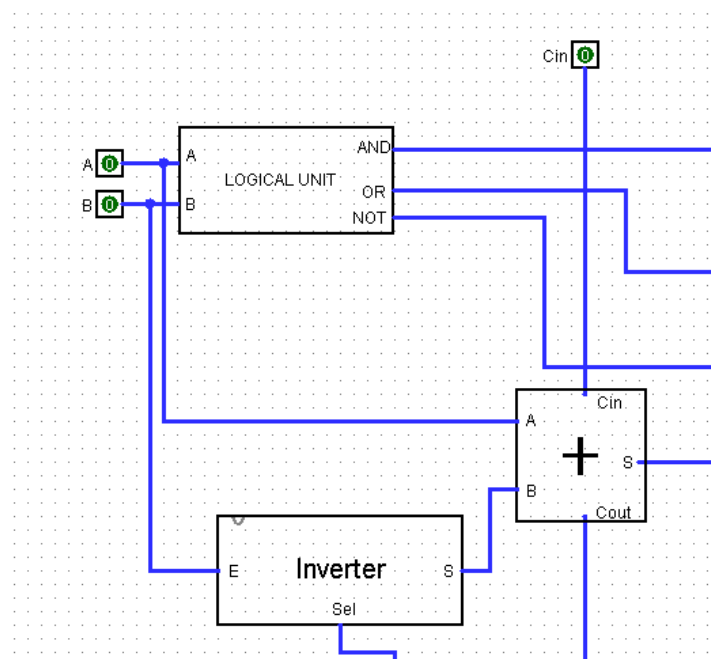


Figure 10: Aperçu de l'UAL de 1 bit

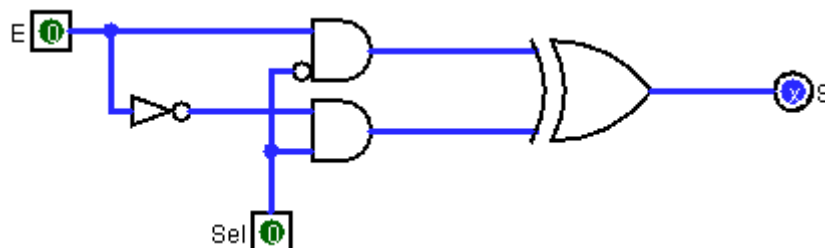
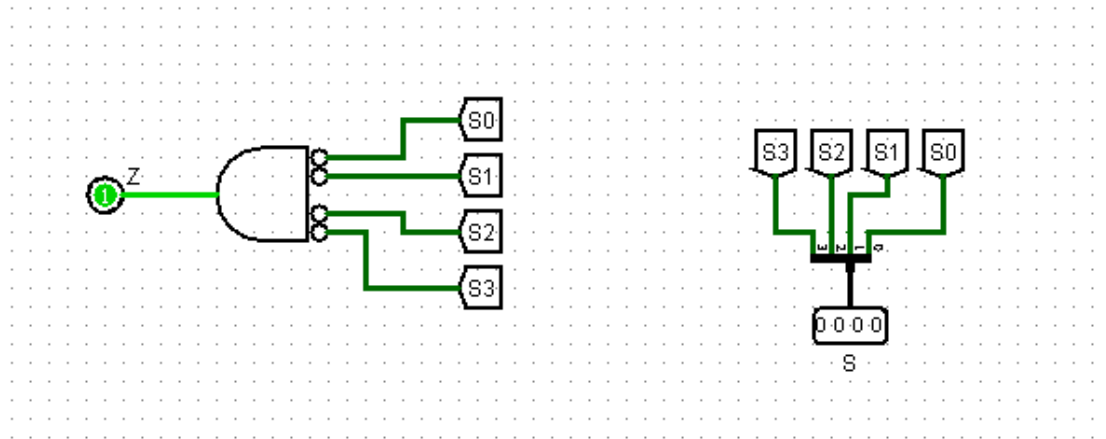


Figure 11: Circuit de l'inverseur ("Inverter")

Ajout d'un signal Z de sortie à l'UAL qui indique si le résultat des opérations de l'UAL est nul ou non.



Figure

12: Signal Z de sortie au niveau de l'UAL



## Annexes

### Programme test

Ce programme permet de tester toutes nos instructions.

- 1) LOAD A, 0x0 : 0100 00 0000 00 = 400
- 2) LOAD B, 0x1 : 0100 01 0001 00 = 444
- 3) NOT C, A, B : 0011 10 00 01 00 = 384
- 4) STORE C, 0x3 : 1000 10 0010 00 = 888
- 5) OR C, A, B : 0001 10 00 01 00 = 184
- 6) STORE C, 0x4 : 1000 10 0011 00 = 88C
- 7) AND C, A, B : 0010 10 00 01 00 = 284
- 8) STORE C, 0x5 : 1000 10 0100 00 = 890
- 9) ADD C, A, B : 0000 10 00 01 00 = 84
- 10) STORE C, 0x6 : 1000 10 0101 00 = 894
- 11) SUB D, A, B : 0111 11 00 01 00 = 7C4
- 12) STORE D, 0x7 : 1000 11 0110 00 = 8D8

Exemple pour 4 et 3 dans la RAM

$$\text{NOT } 0011 = 1100 = C$$

$$0100 \text{ OR } 0011 = 0111 = 7$$

$$0100 \text{ AND } 0011 = 0000 = 0$$

$$0100 + 0011 = 0111 = 7$$

$$0100 - 0011 = 0100 + 1100 + 0001 = 0001 = 1$$