

Rapport PSI

Encadré par:

Adelin Rémi & Alata Eric & Yuxiao

Toumzine Rachid

Quatrième année en Informatique et Réseaux (4IR SI)

INSA Toulouse, France

Juin 2019

1. Développement d'un compilateur en utilisant LEX et YACC

1.1. LEX et YACC

Pour le lex et le yac le projet qu'on a créé il couvre tous les concepts demander dans le cahier des charges .mais il y a des petits problème au niveau de traitement de while et if ,le programme couvre aussi les opérations supérieur ,égale ,inférieur ...les tests marche très bien .

1.2. table des symboles

On a créé deux fichiers **tabsymbole.c** et **tabsymbole.h** pour mettre en place le concept de table de symbole qui contient les information suivant pour chaque symbole:

```
(char id[1600] /*A ,B ou TEMPORAIRE ..*/;int type /*on a seulement le type entier */;int addr /* adress dans la mémoire */ ;int deph /* profondeur*/;)
```

on a ajouté tous les fonctions que on a besoin pour le traitement et aussi une fonction pour l'affichage de tableau .

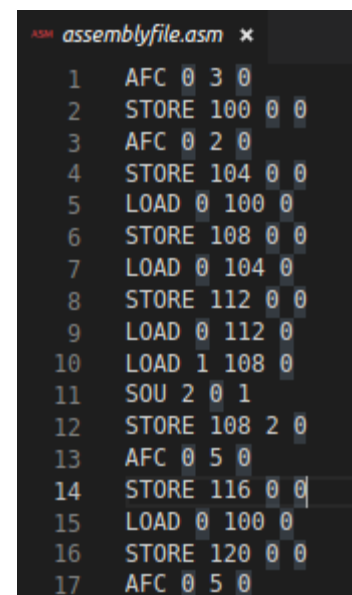
=> les tests marche très bien .

1.3. table des instruction

Pour les instructions on a met en place un tableau qui contient 4 colonne le type d'instruction

(store,mul,div,sou...) on a met en place tous les instruction sauf le jmp ,on a choisi d'utiliser le jump conditionnel que on utilise souvent dans le while et le if .le fichier généré par le code final est de type .asm

=> les tests marche très bien .



```
1  AFC 0 3 0
2  STORE 100 0 0
3  AFC 0 2 0
4  STORE 104 0 0
5  LOAD 0 100 0
6  STORE 108 0 0
7  LOAD 0 104 0
8  STORE 112 0 0
9  LOAD 0 112 0
10 LOAD 1 108 0
11 SOU 2 0 1
12 STORE 108 2 0
13 AFC 0 5 0
14 STORE 116 0 0
15 LOAD 0 100 0
16 STORE 120 0 0
17 AFC 0 5 0
```

1.3. interpréteur

Pour l'interpréteur c'est un fichier qui permet de lire le dernier fichier généré **assemblyfile.asm** et de spécifier les données que on va mettre dans la mémoire et les registres .

il y un problème au niveau de compilation même que le code est correct et logique il affiche a la fin la table de mémoire et de registre .

2. Conception d'un System on Chip (SoC) avec microprocesseur de type RISC avec pipeline

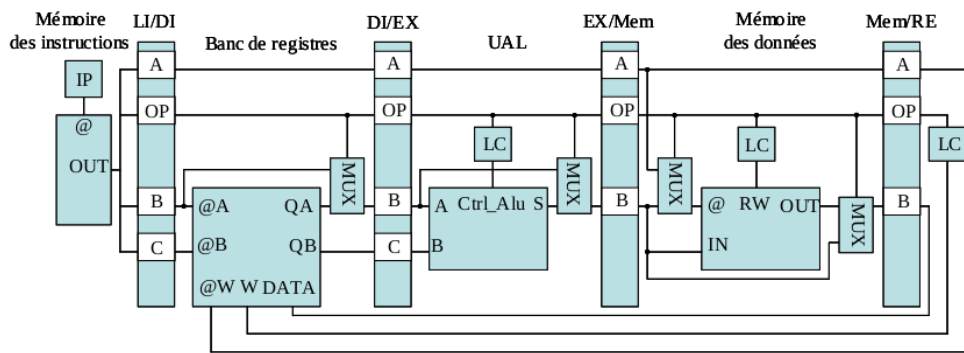


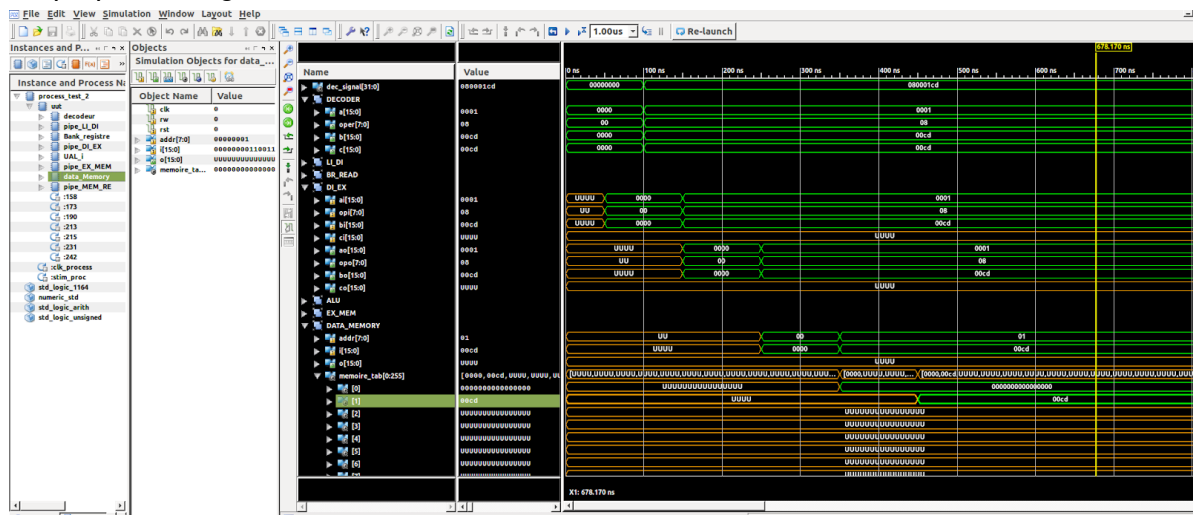
Illustration 6 : Chemin des données

on a utilisé dans cette partie la schéma de 2017 car il est plus compréhensible pour moi tous les composant sont testé et il marche très bien vous pouvez vérifier les fichier ci-joint.mais le problème que j'avais est comment le compteur de 16 bits il choisit une instruction parmi celle proposée par la mémoire d'instruction et pour cela j'ai choisi de faire le test a partir de de décodeur ; proposer un signal de 32 bits au décodeur et après on test le fonctionnement generale .(il y a deux fichiers de test de schema generale PROCESS_TEST_2.vhd shema_general_test.vhd) parmi les teste que on a déjà fait le ADD ,DIV, AFC , LOAD,STORE

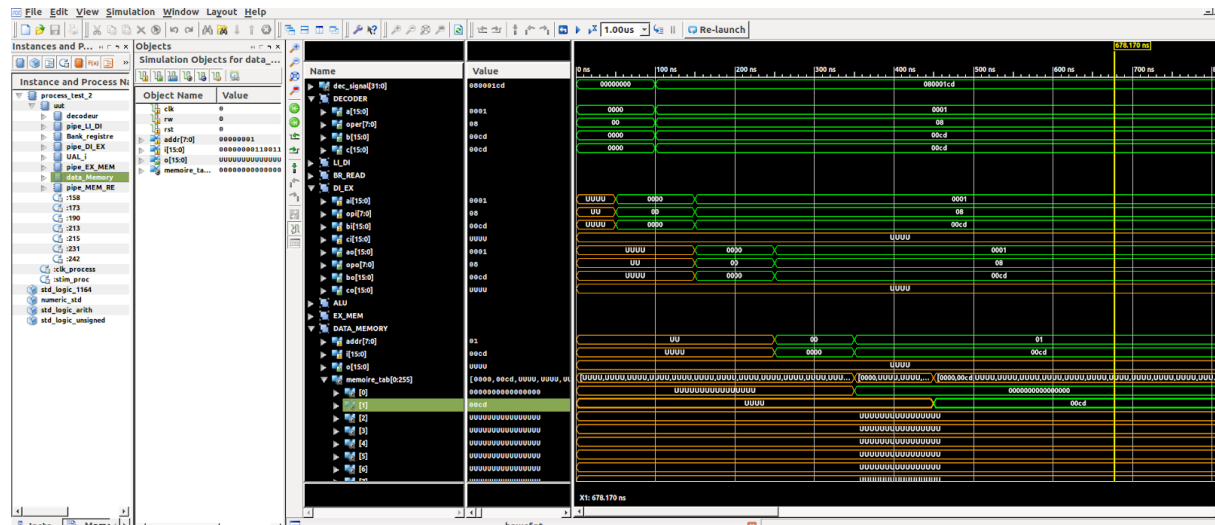
Exemple :

test1: Ce test d'affectation est déjà expliqué dans la vidéo :

On propose le signal suivant au décodeur: x"0601ABCD"



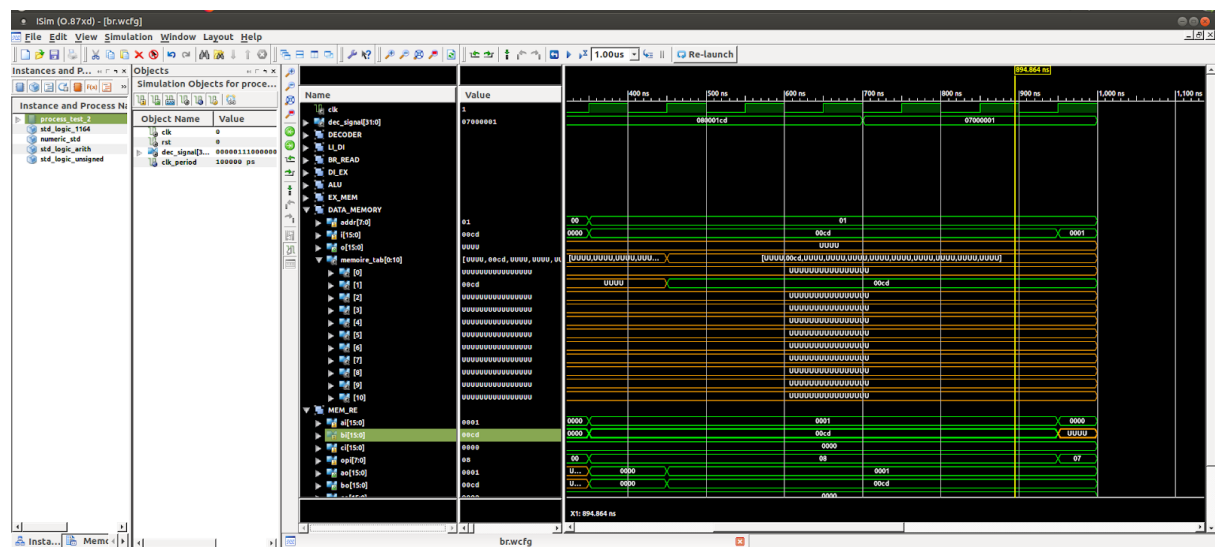
Test2: Ce test de Store et déjà expliquer dans la vidéo
On propose le signal suivant au décodeur: `x"080001cd"`



Résultat: la valeur est enregistré dans la mémoire de donne dans l'adress 1

Test2: Ce test de load et déjà expliquer dans la vidéo

On propose le signal suivant au décodeur: x"080001CD" après en attend 600 ns en propose x"07000001



Résultat: la valeur a lu la valeur enregistrer dans le mémoire de donne .

conclusion

Ce projet s'est révélé très enrichissant dans la mesure où il a consisté en une approche concrète de langage machine . En effet, le sujet était très intéressant à résoudre, j'ai essayé d'avancer le maximum grâce a votre encadrement ,malgré les difficultés d'interprétation des méthodes à utiliser pour solutionner le problème, et une non-maîtrise du langage de programmation (assembleur).

En fin, Je ne pouvais achever ce projet sans exprimer ma gratitude à vous **Adelin Rémi& Alata Eric & Yuxiao** ,

Lien Repository Github :

Dépot github <https://github.com/RachidToumzine/PSI>

drive des videos et document

<https://drive.google.com/drive/folders/1Ou2IzHq9fRNI-Kcjk05SO0zpva4VH2B3>