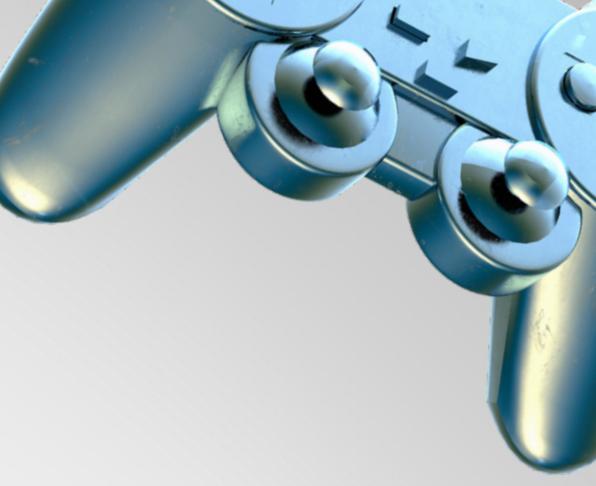


Groupe B

Processeur P-ARM

Florian Latapie - Loïc Le Bris - Marius Lesaulnier - Thomas Paul



Contexte

Objectif

- 1/ Comprendre le fonctionnement d'un processeur en se basant sur l'architecture ARM
- 2/ Simuler le comportement d'un microprocesseur basique (P-ARM) en le modélisant sur le logiciel Logisim

Finalité

Exécuter des programmes simples démontrant notre compréhension du projet

Plan



I - Présentation des composants

II - Démonstration du processeur

III - Démonstration de l'assembleur

IV - Qualité

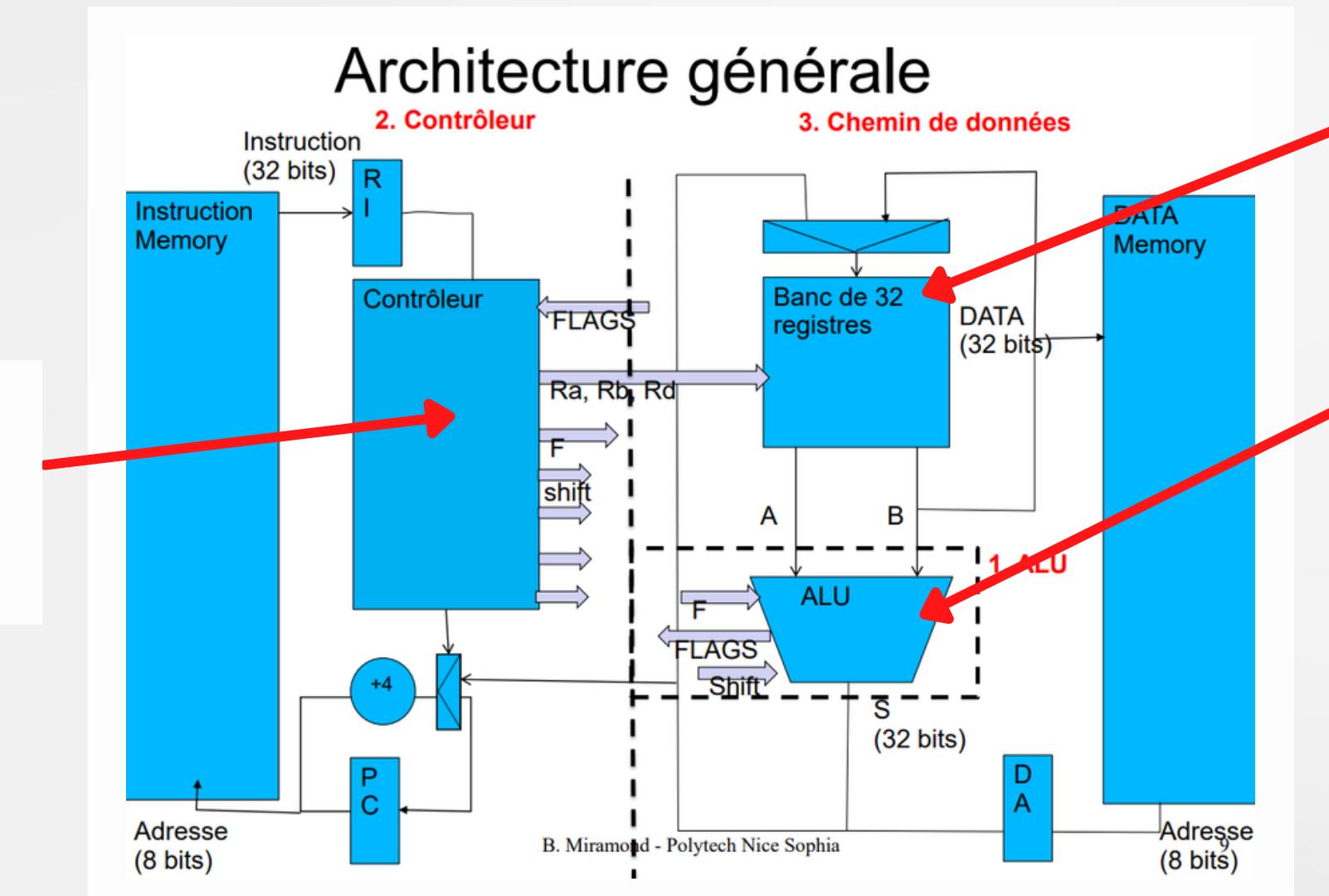
V - Organisation du travail

VI Démonstration de notre
programme sur le processeur

VII - Problèmes rencontrés

I - Présentation des composants

Contrôleur : chef d'orchestre au sein du processeur. Il lit les instructions et les chargent.

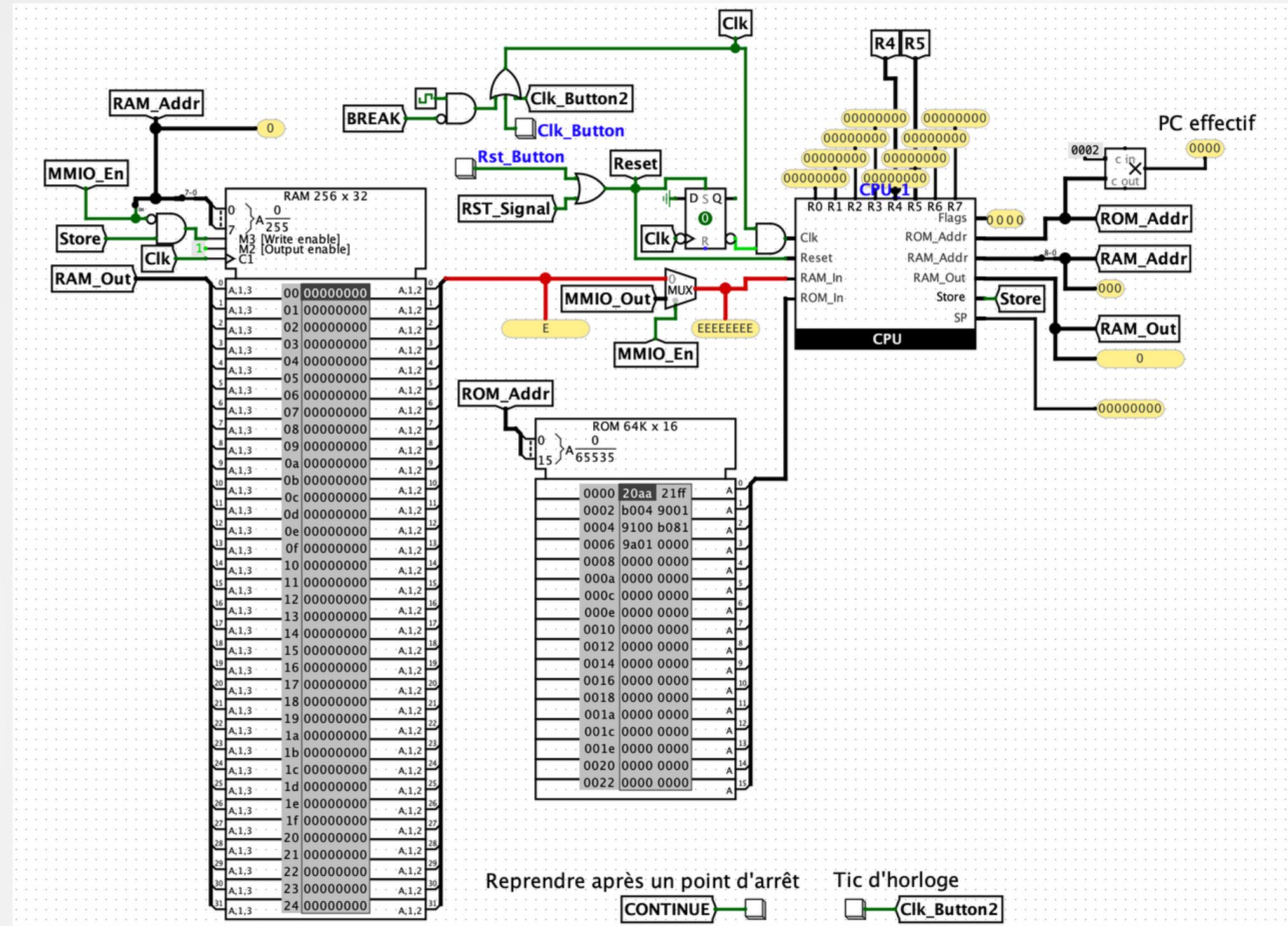


Banc de registre : Le processeur stocke le résultat d'un calcul dans un registre (mémoire rapide).

ALU (L'unité arithmétique et logique) : élément qui se charge des calculs au sein du processeur.

II - Démonstration du processeur

Exécution des tests inclus dans le projet



III - Démonstration de l'assembleur

Récupérer un code en .s de Clang-8 et de le convertir à nouveau vers un code binaire (Hexadécimal) utilisable dans la ROM de notre processeur

```
#include <parm.h>

void run () {
BEGIN ();
    int a = 5;
    int b = 10;
    RES = a+b;
END ();
}
```



Résultat de notre programme Java

```
run:
sub sp, #92
sub sp, #508
sub sp, #452
movs r0, #5
str r0, [sp, #4]
movs r0, #10
str r0, [sp]
ldr r0, [sp, #4]
ldr r1, [sp]
adds r0, r0, r1
str r0, [sp, #32]
```

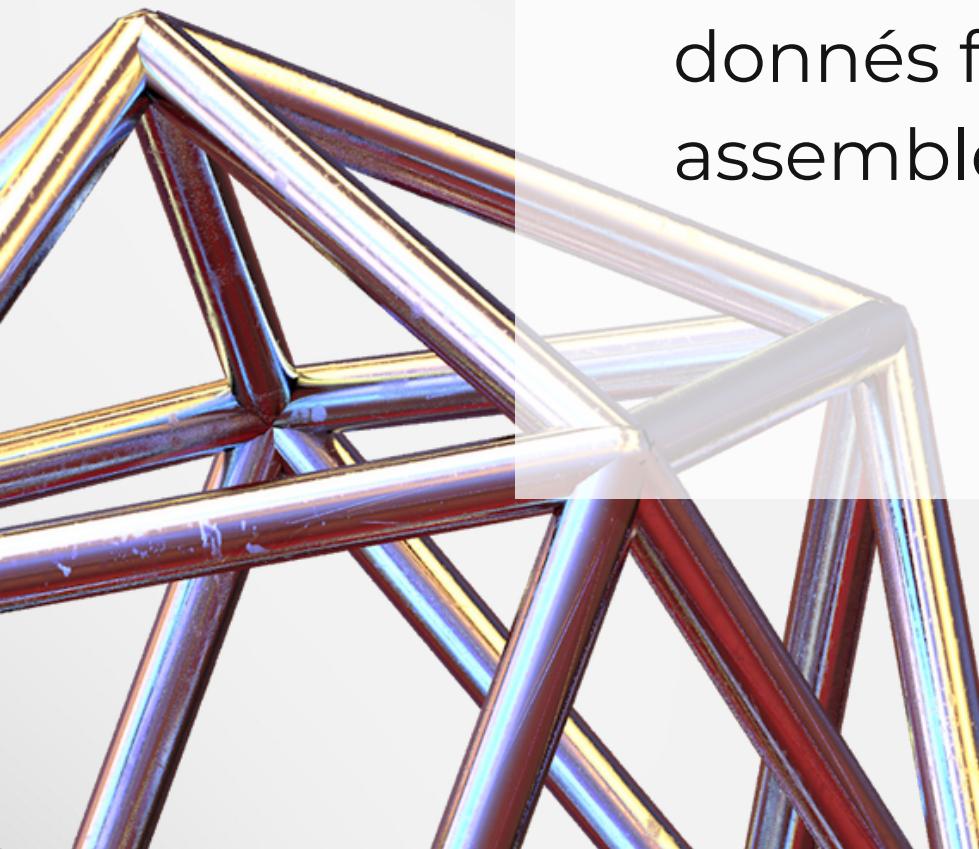


v2.0 raw
B097 BOFF BOF1 2005 9001
200A 9000 9801 9900 1840
9008 E7FE E7FE E7FD

IV - Qualité

Bonne qualité

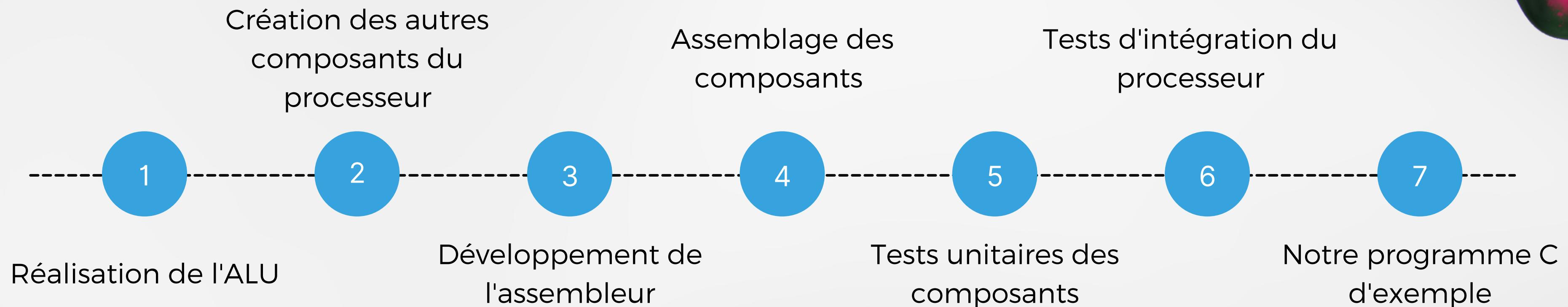
- Tous les composants
- Assembleur
- Tous les tests d'intégration donnés fonctionnent pour notre assembleur et notre processeur.



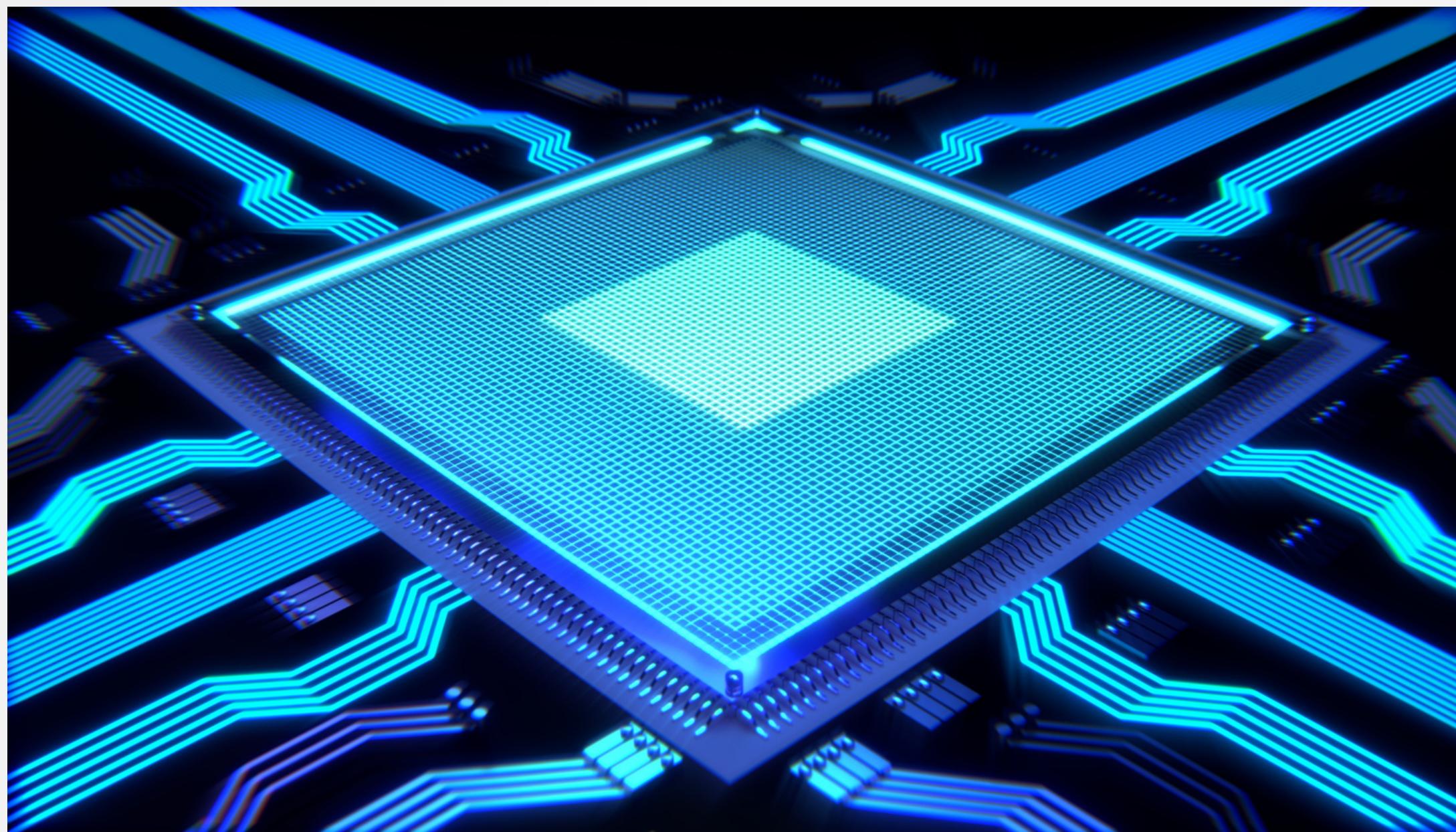
Mauvaise qualité

- L'architecture du Shift Add Sub Mov est difficile à comprendre : utilisation d'un multiplexeur plus simple
- Assembleur : il correspond aux attentes, MAIS il lit seulement ligne par ligne et sépare chaque morceau d'instruction par des espaces

V - Organisation du travail



VI - Démonstration de notre programme sur le processeur



VII Problèmes rencontrés



- Shift, Add, Sub, Mov : plein de petits problèmes qui ont pu être résolu grâce aux tests
- Mauvaise configuration des registres
- Conditional mal documenté : il doit gérer le conditionnel et l'inconditionnel
- Alu mal documenté pour l'instruction "MVNS" : Au niveau de l'ALU : not A au lieu de not B