

ARMv7

Cortex M0

Si3 - Groupe 1

Antoine STEYER

Axel AIELLO

Gabriela CAVALCANTE DA SILVA

Raquel LOPES DE OLIVEIRA

Thomas JALABERT

Loïc ROSE

7 janvier 2016

Schedule

- 1 Extension de l'ULA
- 2 Contrôleur-Mémoriel
- 3 Registres - MémoireD
- 4 Assembleur/Désassembleur
- 5 Outils logiciels
- 6 Ecriture de jeux de test
- 7 Conclusion
- 8 References

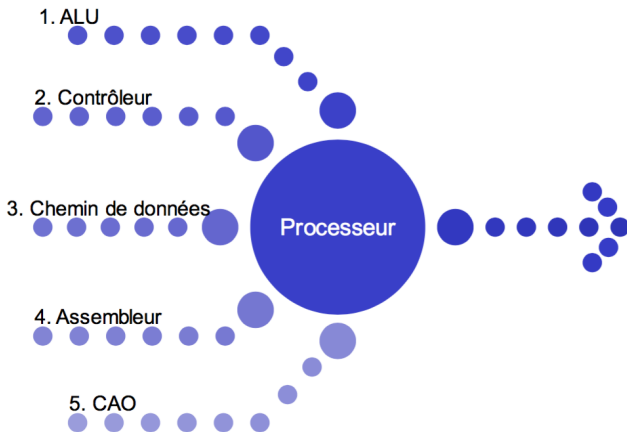


Figure: Component Schema

- ▶ Réaliser les blocs d'opérateurs arithmétiques et logiques
- ▶ Générer les Flags

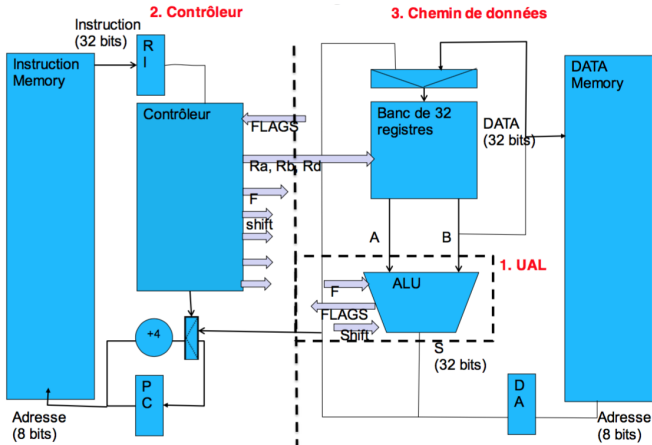


Figure: Architecture générale

ULA

Arithmetic Logic Unit

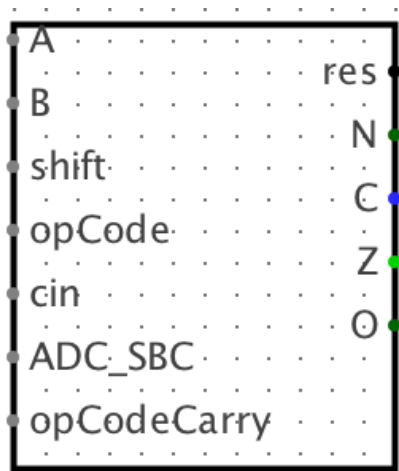


Figure: ALU Component

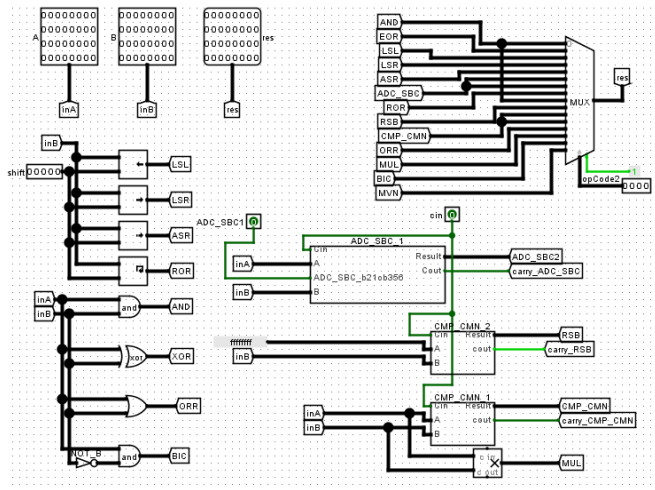


Figure: Detail de fonctionnement

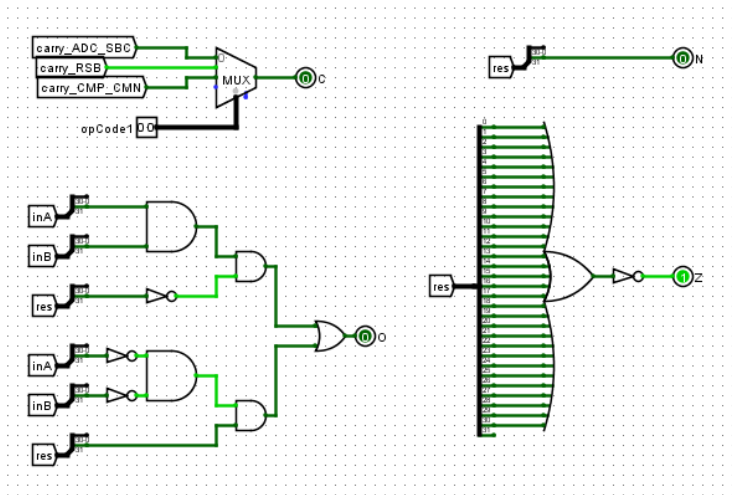
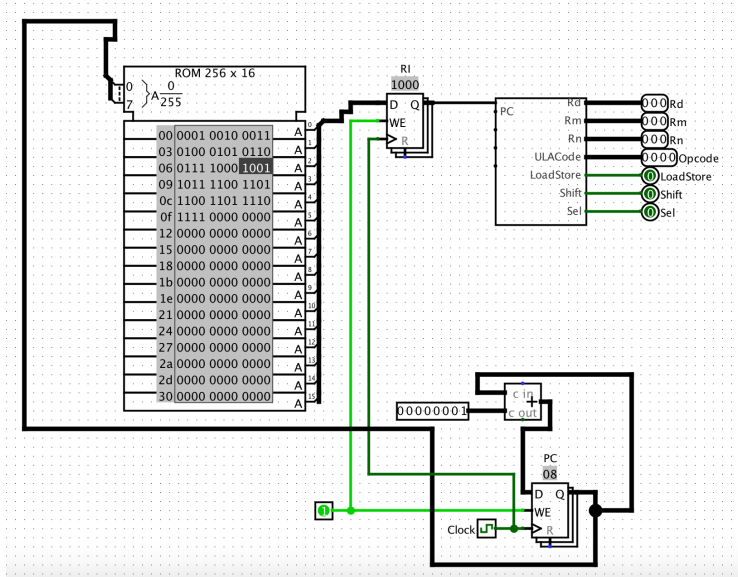


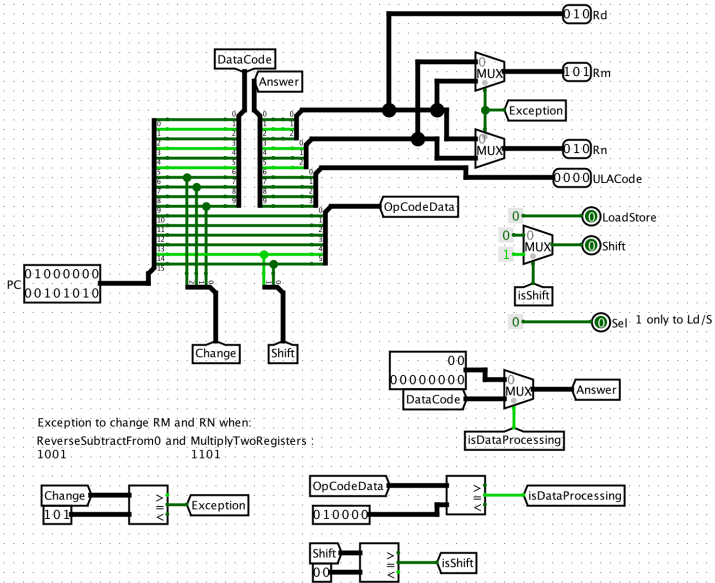
Figure: Detail de fonctionnement

Contrôleur



Contrôleur

Data Processing



Registre

Banc de registres

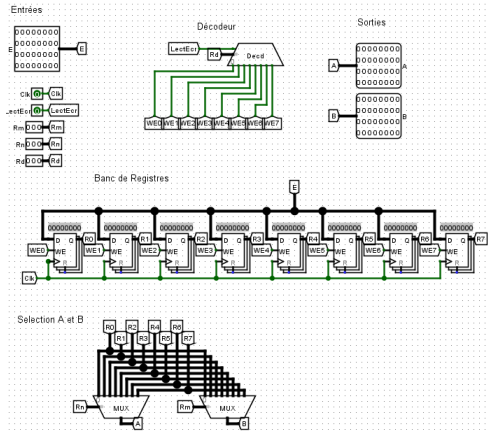


Figure: Banc de registre

Registre

Chemin de données

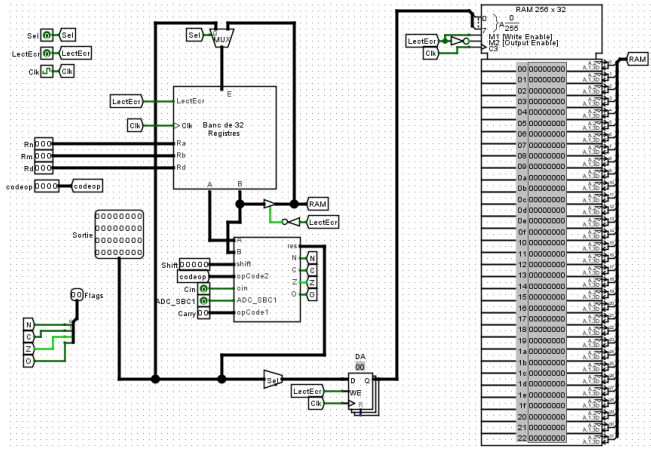


Figure: Chemin de données

- 1 Lecture du fichier assembleur
- 2 Identification du opcode 1
- 3 Identification du opcode 2
- 4 Ecriture du fichier binaire.

```
AND R2,R7  
EOR R1,R5  
LSL R5,R6  
ADC R4,R3  
LSL R1,R2,#23|
```

Figure: Code assembleur

- 1 Utilisation de logisim-evolution
- 2 Utilisation de Quartus II (v13.0 sp1)
- 3 Synthèse sur FPGA (DE2-BOARD)
- 4 Validation de l'ALU

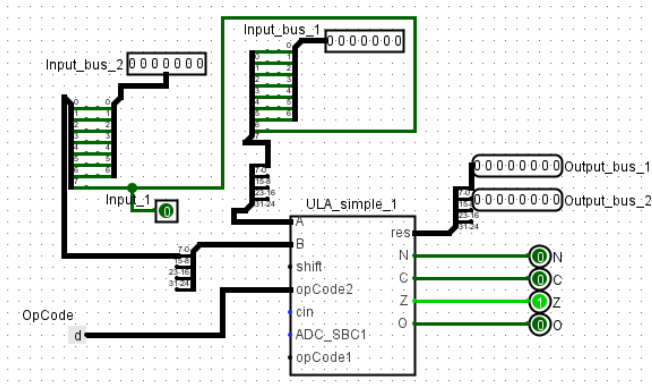


Figure: Multiplication avec l'ALU

Encore beaucoup de travail !

- ❶ Le banc de registre, le contrôleur sont encore à vérifier
- ❷ Comment entrer les données binaires dans le processeur ?
- ❸ Simplifier les équations logiques

- 1 ARM®v7-M Architecture Reference Manual
https://web.eecs.umich.edu/~prabal/teaching/eecs373-f10/readings/ARMv7-M_ARM.pdf