

Conception du microprocesseur

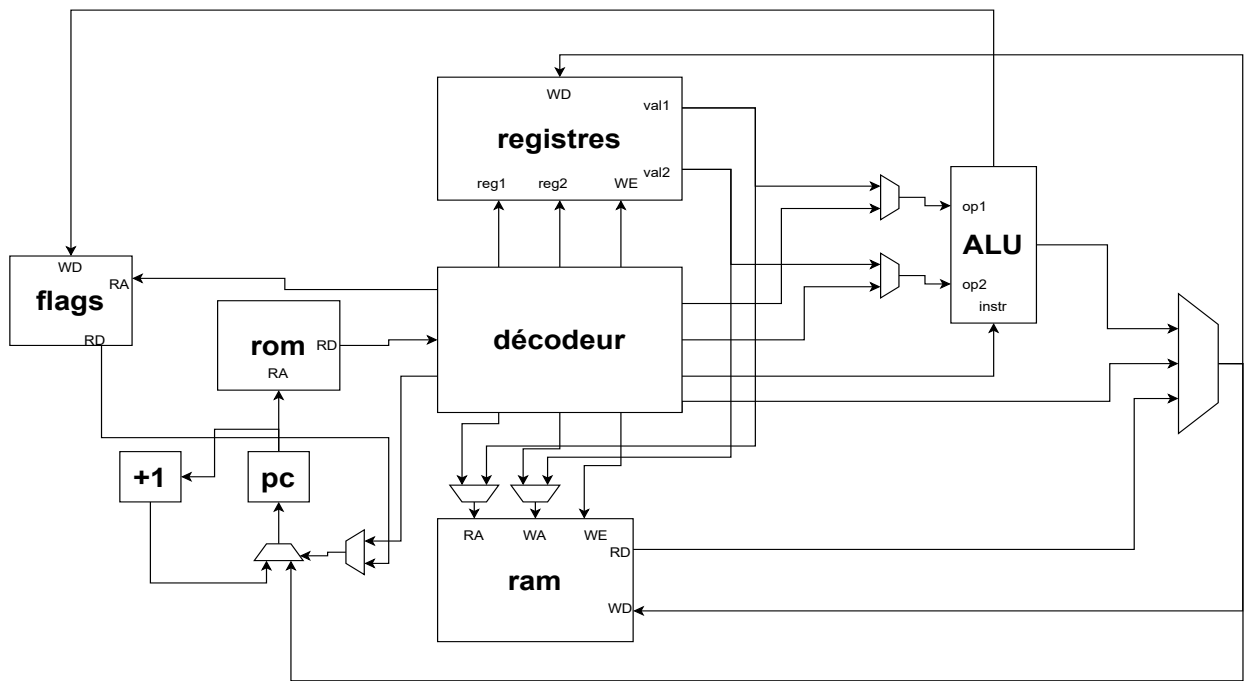
Florian Tousnakhoff

Fabio d'Ortoli Galerneau

Félix Lebrat

13/12/2021

Architecture générale



- RA : read address
- WE : write enable
- WA : write address
- RD : read data
- WD : write data

les multiplexeurs dont l'entrée de contrôle n'est pas précisée sont contrôlés par le décodeur

Registres

Le microprocesseur contient 16 registres généraux de 32 bits :

- rax
- rbx
- rcx
- rdx
- rsi
- rdi
- rbp
- rsp
- r8
- r9
- r10
- r11
- r12
- r13
- r14
- r15

qui sont donc adressés sur 4 bits

RAM

L'adressage de la RAM se fait sur 16 bits, et la taille des mots est de 32 bits.

ALU

L'ALU prends deux entrées sur 32 bits et sort le résultat, sur 32 bits également, de l'opération spécifiée par l'entrée **instr** sur 3 bits selon le code suivant :

1. non bit à bit
2. et bit à bit
3. ou bit à bit
4. ou exclusif bit à bit
5. multiplication
6. addition
7. soustraction

et définit les drapeaux **neg** et **null** en fonction du résultat.

Instructions

Les instructions sont codées sur 32 bits, dont 6 pour l'opcode et 13 par opérande. Elles sont stockées dans une ROM, adressée sur 16 bits, et lues par le décodeur. Le jeu d'instructions est le suivant :

- MOV op1 op2
- NOT op
- XOR op1 op2
- OR op1 op2
- AND op1 op2
- ADD op1 op2
- SUB op1 op2
- MUL op1 op2
- LSL op
- LSR op
- CMP op1 op2 (définit les drapeaux pour op2-op1)
- JMP op

- JZ op (jump if null)
- JNZ op (jump if not zero)
- JL op (jump if less than)

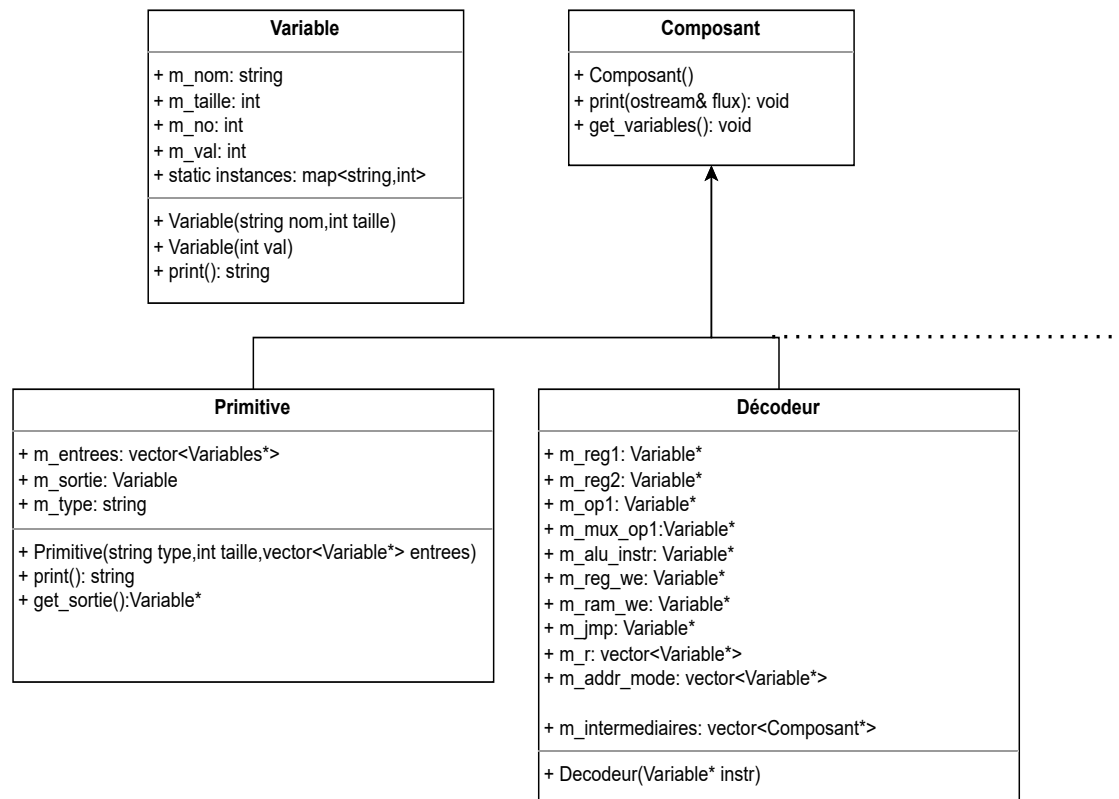
Modes d'adressage :

- MOV : op1 peut être une constante, une adresse RAM ou un registre, et op2 peut être une adresse RAM ou un registre.
- Pour les opérations arithmétiques ou logiques à une opérande, op doit être un registre
- Pour les opérations arithmétiques ou logiques à deux opérandes, op1 doit être une constante ou un registre et op2 un registre.
- CMP : op1 et op2 sont des constantes ou des registres
- pour les instructions de branchement, op est une constante ou un registre

Choix techniques :

Les netlists sont écrites directement, à l'aide du C++ (ou d'un autre langage).

Génération de netlist en C++



Le dictionnaire `instances` dans la classe **Variable** associe un nom de variable au nombre de variables portant ce nom. Lorsqu'une variable est créée, on peut donc lui associer un numéro pour éviter la duplication des noms. Une netlist est représentée par un ensemble de **Composant**. Tous les **Composant** ont des méthodes `get_variables` et `print`. La première renvoie l'ensemble des variables déclarées dans le **Composant** et la deuxième sert à écrire les équations qui les définissent.

dans un flux. De manière générale, les entrées et les sorties des composants sont représentées par des **Variable**.

Schéma du gestionnaire de registres

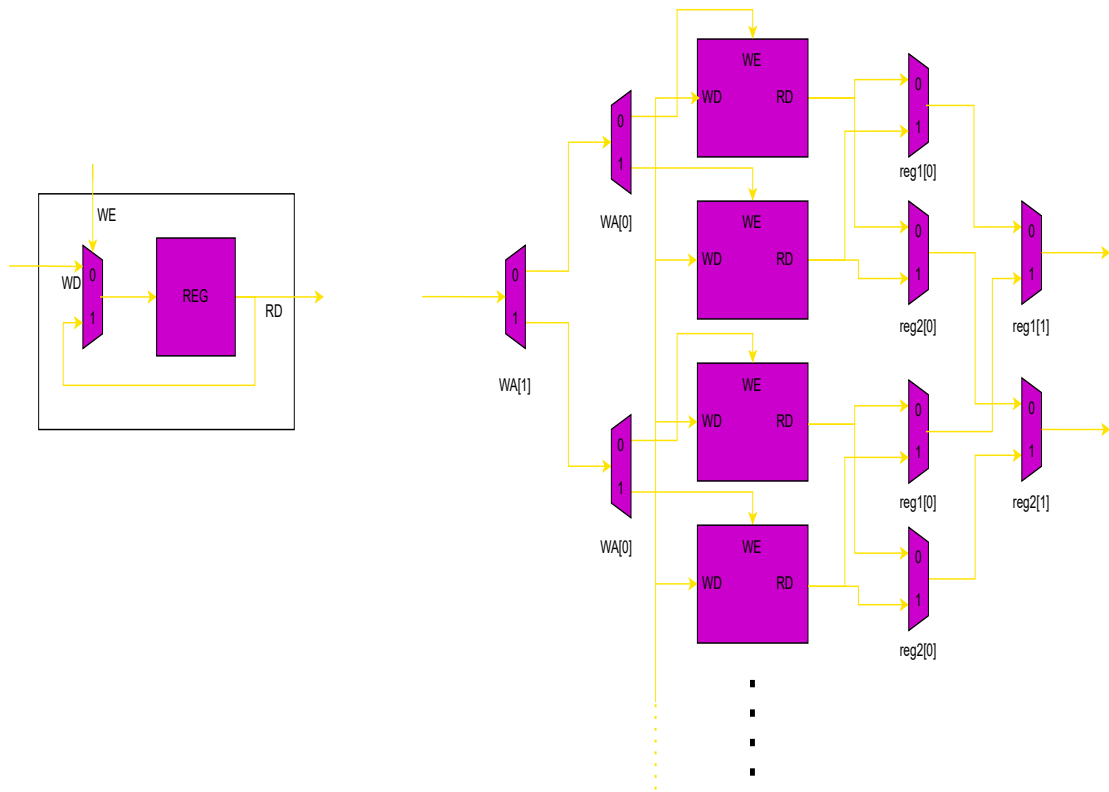
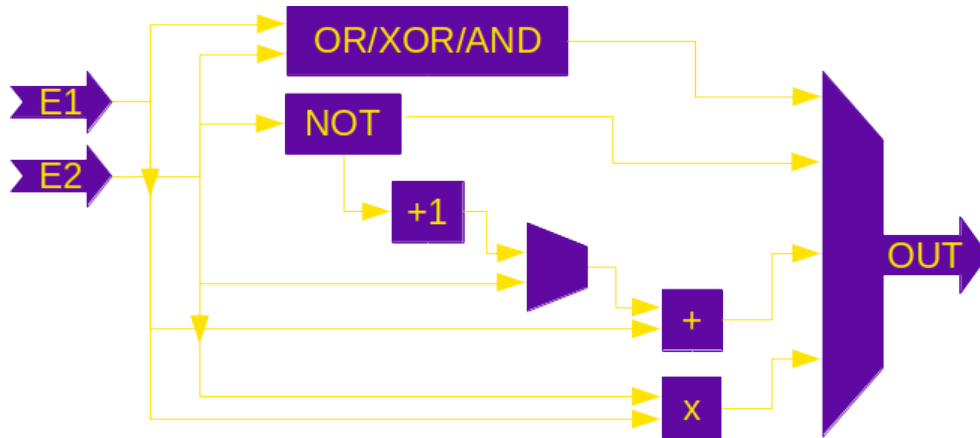


Schéma de l'ALU :



Assembleur et Horloge:

L'assembleur a été réalisé en OCaml, il consiste en un lexer, un parser et une fonction transformant l'arbre de syntaxe abstraite en code binaire lu et reconnu par le décodeur du microprocesseur.

L'Horloge a été faite avec l'assembleur, elle donne l'heure du Méridien de Greenwich.

Le signal d'horloge du microprocesseur est donné par le timestamp (nombre de secondes depuis le premier janvier 1970) et est transformé par le programme en la date.

Cela est ensuite mis en mémoire et on fait pointer le registre r14 vers cet endroit de la mémoire qui sera ensuite affiché par l'afficheur 7 segments.