Projet PARM

Groupe Mozzarella
Michael Mc Carthy, Robin Viollet, Thomas Guiot, Yaël Tramier

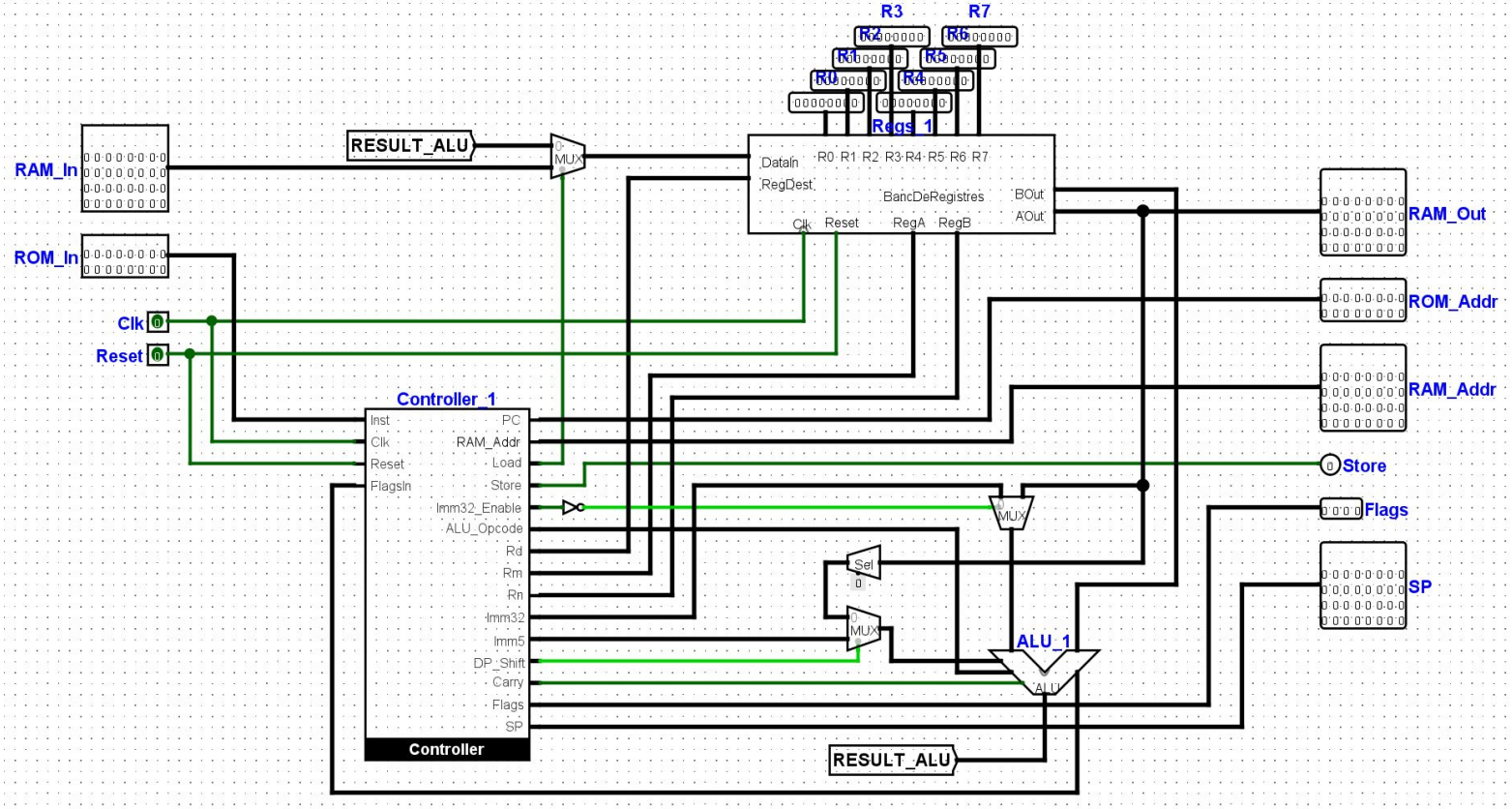
Objectifs du projet

Objectif du projet

- Reproduire un microprocesseur ARM Cortex-M0
- Fonctionnement simplifié :
  - Instructions sur 16 bits
  - Seulement 16 instructions dans l'ALU
  - Données sur 32 bits
  - Adressage RAM/ROM sur 9 bits
  - Adressage RAM uniquement sur la pile (en utilisant le Stack Pointer)

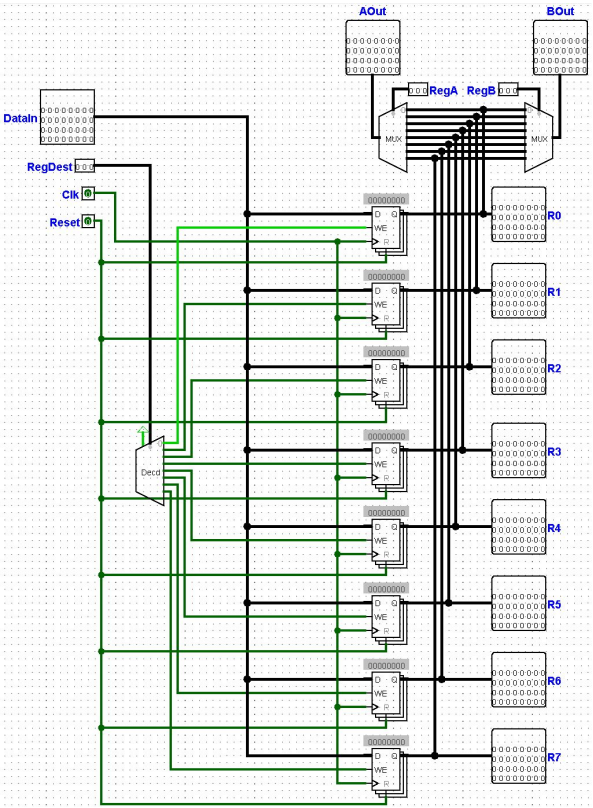
Vue générale du CPU

Chemin de données



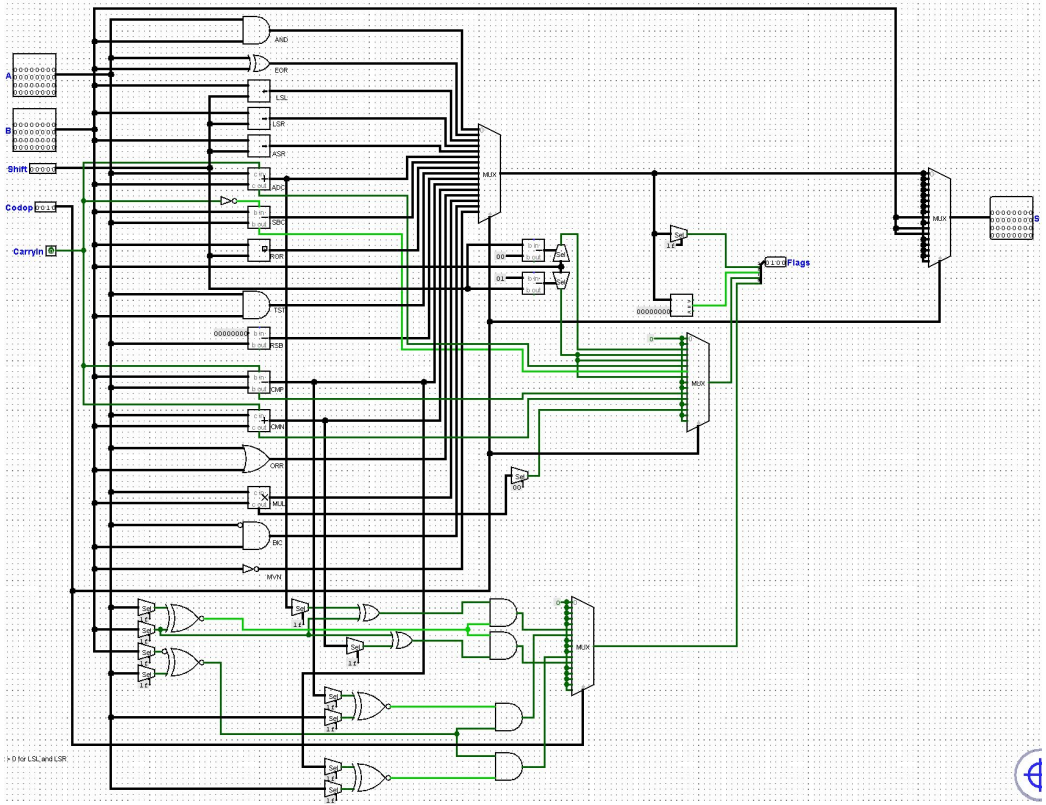
Banc de registres

Banc de registres



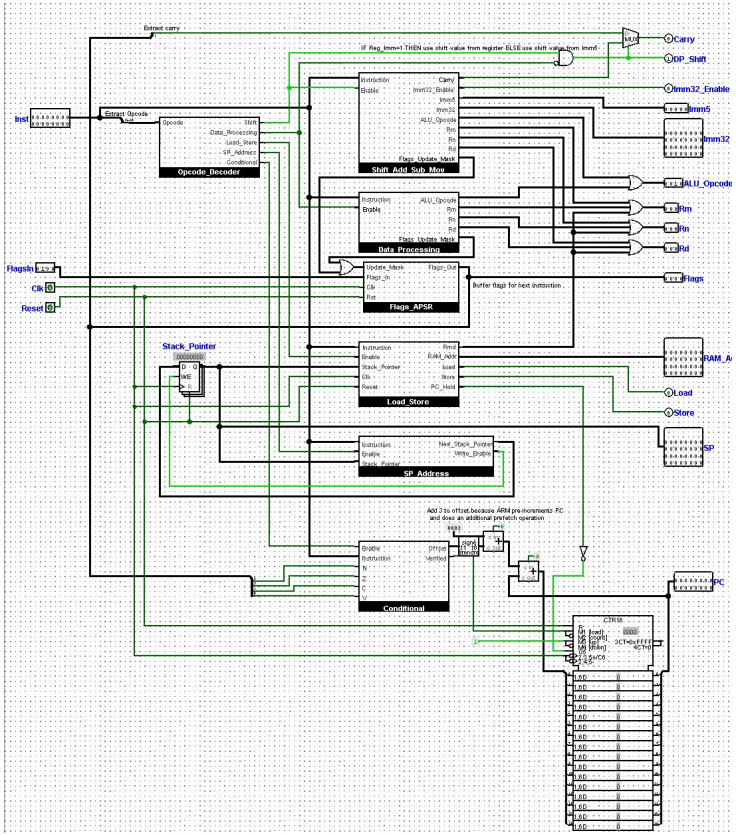
ALU

The Arithmetic and Logical Unit

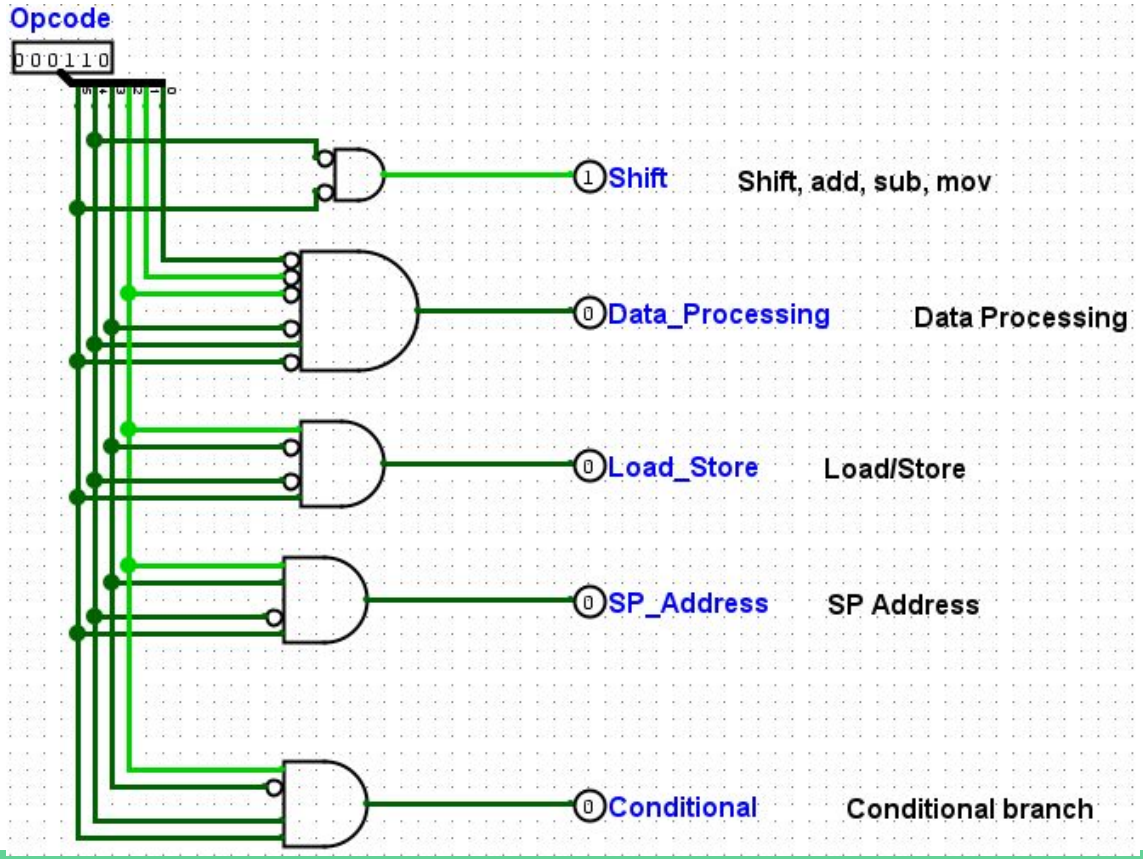


Le contrôleur

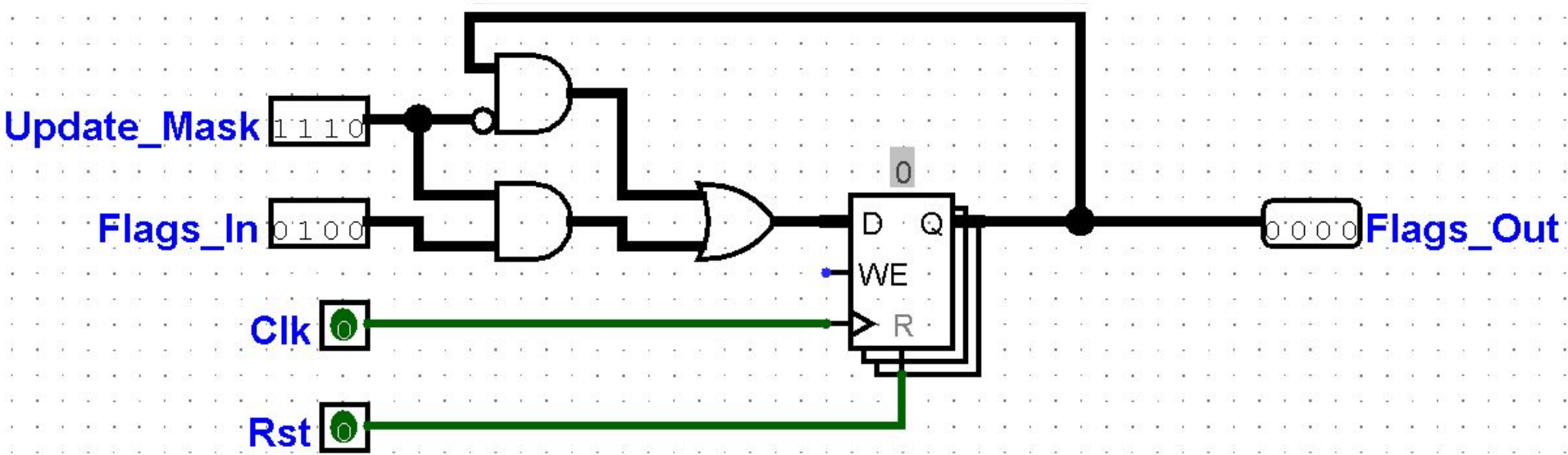
Vue globale du Controller



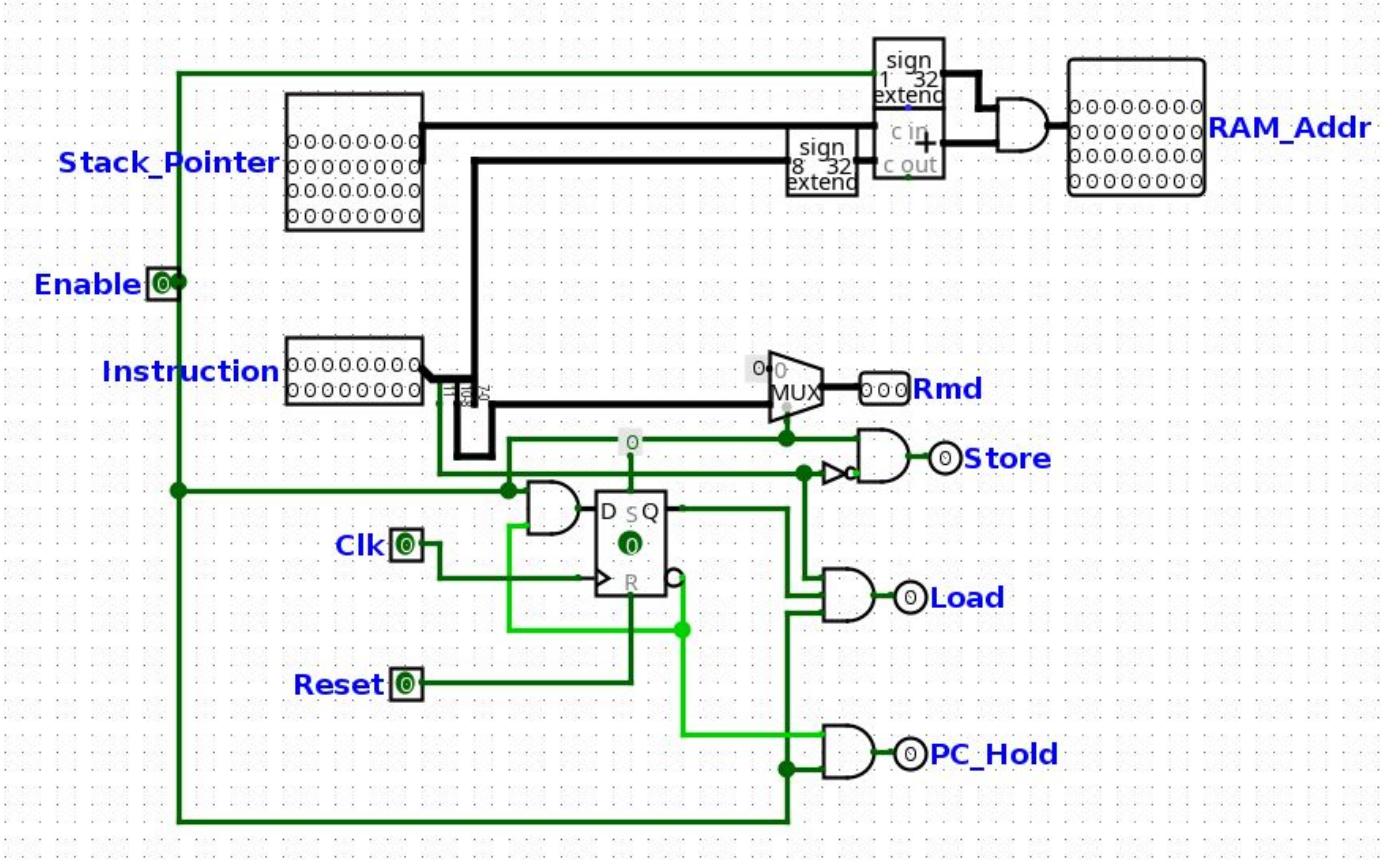
Opcode Décodeur



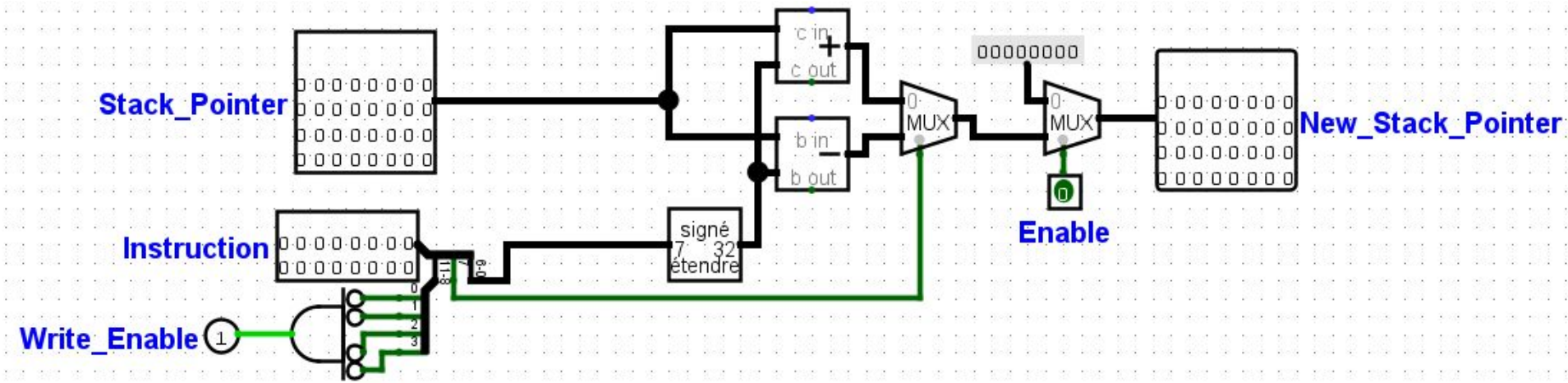
Flag ASPR


Update_Mask
1 1 1 0
Flags_In
0 1 0 0
Clk
0
Rst
0
0
D
Q
WE
R
0 0 0 0
Flags_Out

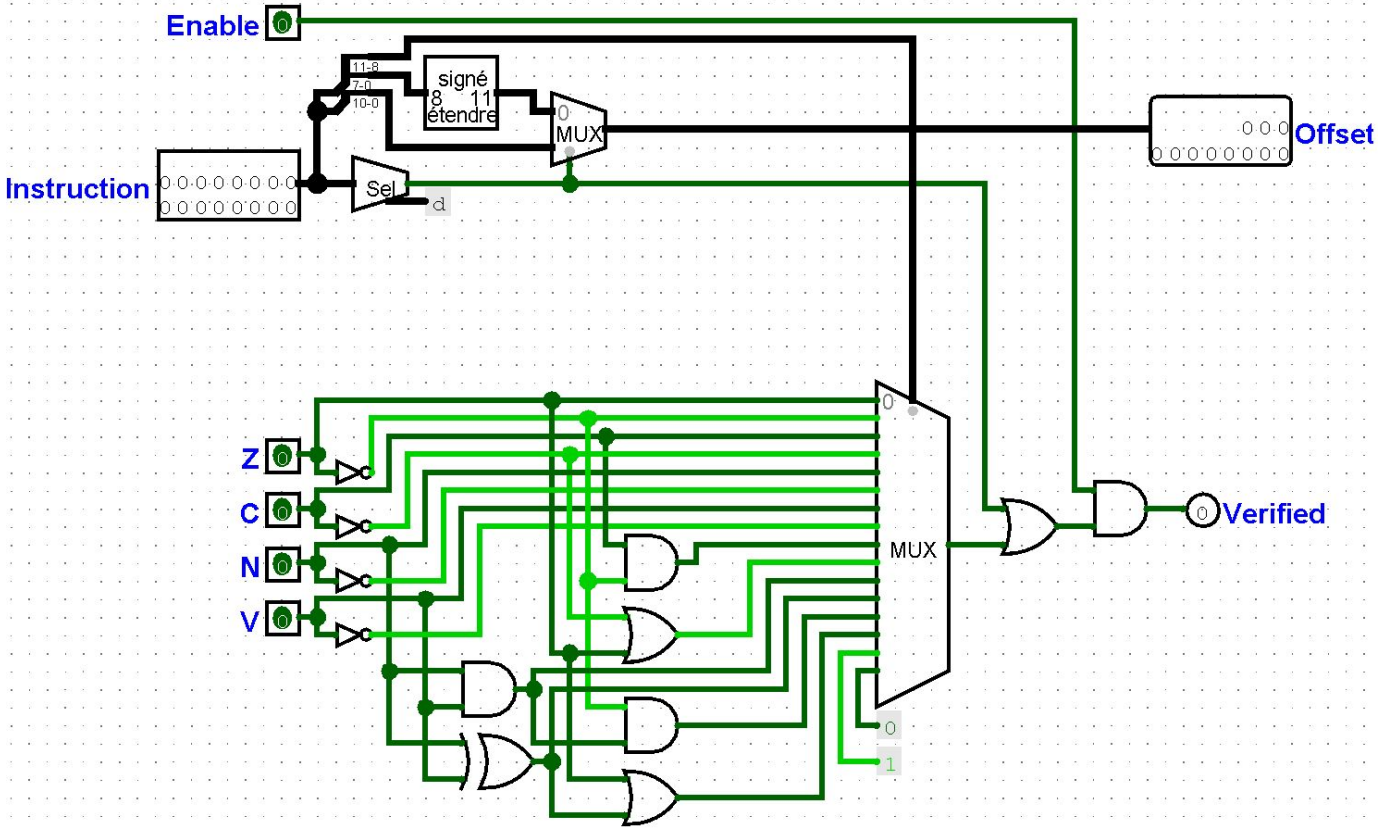
Load/Store


Stack_Pointer
Enable
Instruction
sign
1 32
extend
sign
8 32
extend
c in
c out
RAM_Addr
MUX
Rmd
Store
Clk
D S Q
R
Load
Reset
PC_Hold

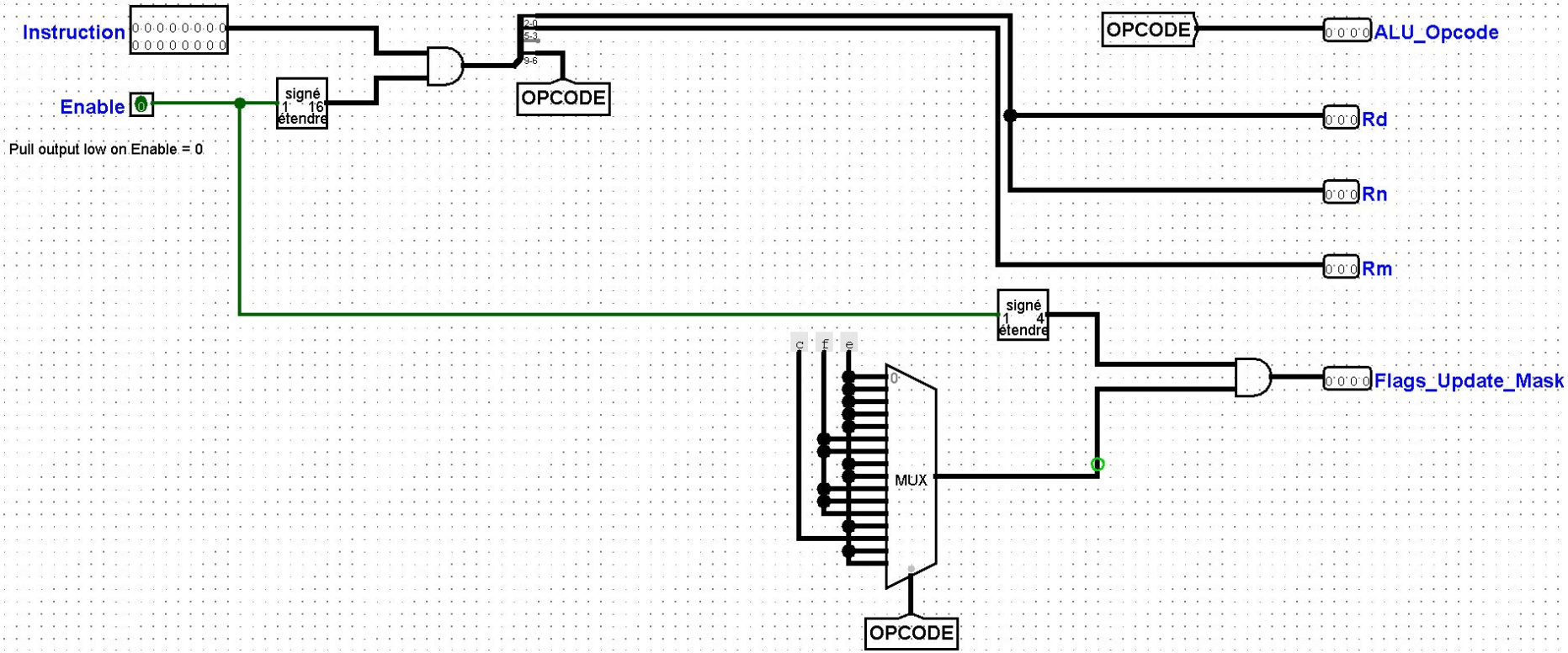
SP Address


Stack_Pointer
Instruction
Write_Enable
1
c in
c out
b in
b out
signé
7 32
étendre
0
MUX
00000000
0
MUX
0
Enable
New_Stack_Pointer

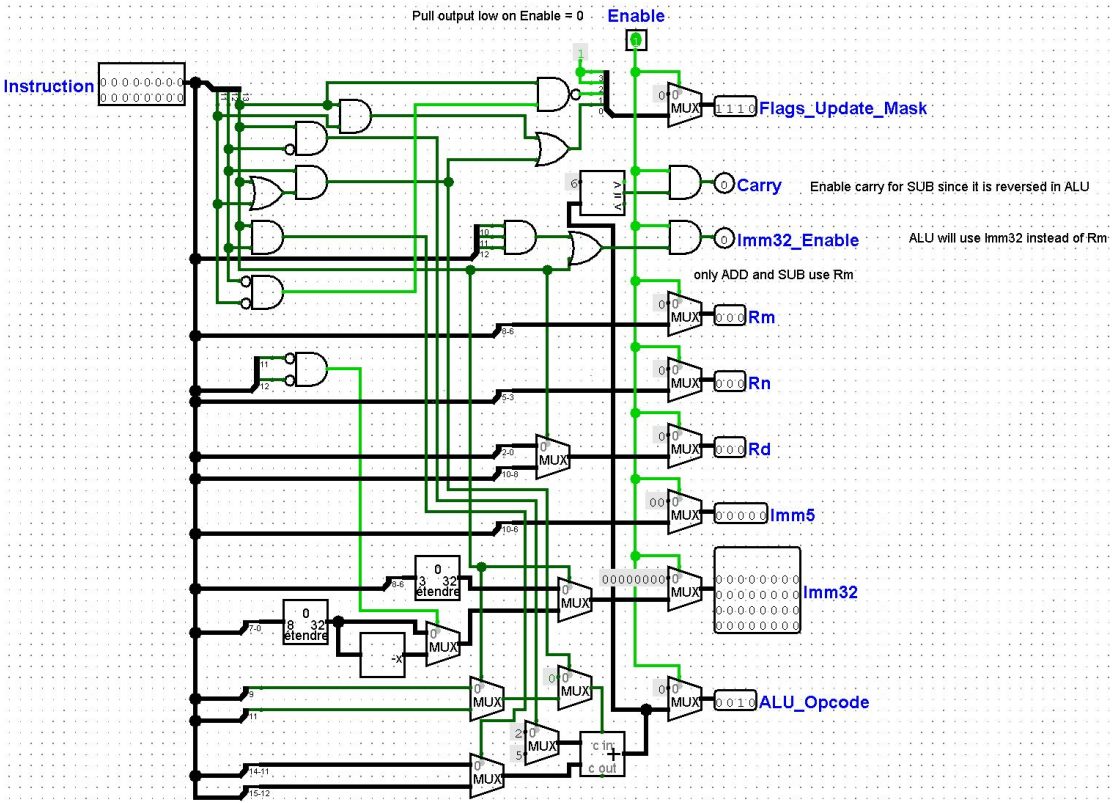
Conditional



Data Processing



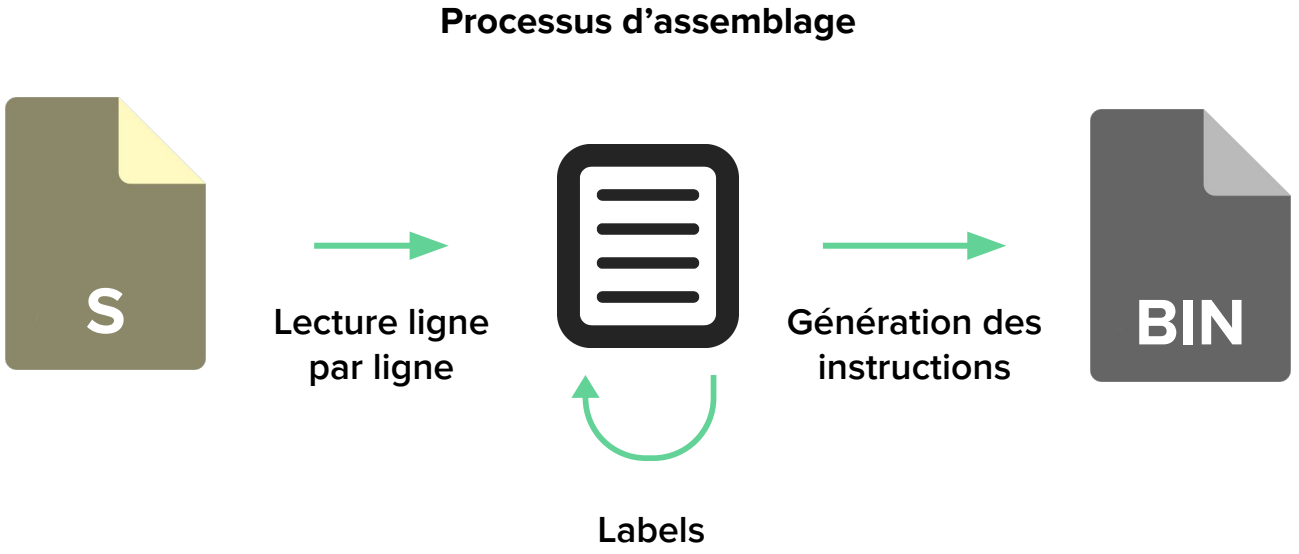
Shift, add, sub, mov



Avengers

Convertisseur d’assembleur ARM vers le format binaire de Logisim

Assembleur


Processus d'assemblage
S
Lecture ligne par ligne
Génération des instructions
BIN
Labels

Assembleur

```c
#include "parm.h"

void run()
{
 BEGIN();
 int a = 1;
 int b = 2;
 int c = a+b;
 RES = c;
 END();
}
```

simple_add.c

```
v2.0 raw
b098 b0ff b0f1
2001 9002 2002
9001 9802 9901
1840 9000 9800
9009 e7fe e7fe
e7fd
```

simple_add.bin

```
run:
  .fnstart
  .pad    #96
  sub     sp, #96
  @APP
  sub     sp, #508
  @NO_APP
  @APP
  sub     sp, #452
  @NO_APP
  movs    r0, #1
  str     r0, [sp, #8]
  movs    r0, #2
  str     r0, [sp, #4]
  ldr     r0, [sp, #8]
  ldr     r1, [sp, #4]
  adds    r0, r0, r1
  str     r0, [sp]
  ldr     r0, [sp]
  str     r0, [sp, #36]
  b   .LBB0_1
.LBB0_1:
  b   .LBB0_2
.LBB0 2:
  b   .LBB0 2
.Lfunc_end0:
  .size   run, .Lfunc_end0-run
  .cantunwind
  .fnend
```

simple_add.s

Conversion 100% conforme

0001100 001 000 000

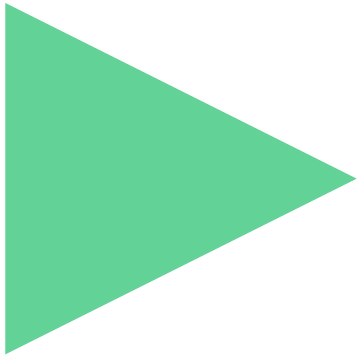
r1 r0 r0

Assembleur - Tests

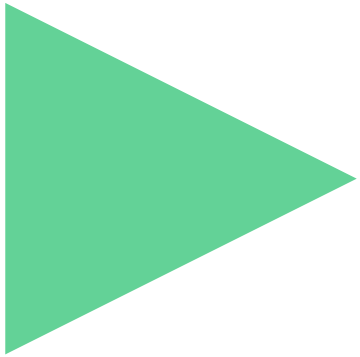
BASH

THE BOURNE-AGAIN SHELL

Assembleur - Démo



Processeur - Démo



Merci de votre attention