

# Environnement d'exécution y86+HCL

- BANDET Alexis
- GAISSET Valentin
- GUISET Romain
- SIMBA Florian

# Introduction

Y86 Créé par R. E. Bryant et D. R. O'Hallaron

14	Main:
15	irmovl 4,%eax
16	pushl %eax # Push 4
17	irmovl array, %edx
18	pushl %edx # Push array
19	call Sum # Sum(array, 4)
20	popl %ebx
21	popl %ebx
22	halt

0x001c: 30f004000000	Main: irmovl 4,%eax
0x0022: a00f	pushl %eax
0x0024: 30f20c000000	irmovl array, %edx
0x002a: a02f	pushl %edx
0x002c: 8036000000	call Sum
0x0031: b03f	popl %ebx
0x0033: b03f	popl %ebx
0x0035: 00	halt

HCL : Hardware Controller Langage

```
86 ##### Decode Stage #####
87
88 ## What register should be used as the A source?
89 int srcA = [
90     icode in { RRMovl, RMMovl, OPl, PUSHl } : rA;
91     icode in { POPl, RET } : RESP;
92     1 : RNONE; # Don't need register
93 ];
94
95 ## What register should be used as the B source?
96 int srcB = [
97     icode in { OPl, IOPl, RMMovl, MRMovl } : rB;
98     icode in { PUSHl, POPl, CALL, RET } : RESP;
99     icode in { LOOP } : RECX;
100     1 : RNONE; # Don't need register
101 ];
```

# Description de l'existant

	Simulateur Web JS	Simulateur Web Angular	Simulateur CS:APP	Simulateur Node JS
Editeur	✓			✓
Etat du processeur		✓	✓	✓
Object code	✓	✓	✓	✓
Etat Mémoire	✓	✓	✓	✓
Version séquentielle	✓		✓	✓
Version pipelinée		✓	✓	✓
Ajout d'instructions			✓	✓
Design Moderne	✓	✓		✓
Requêtes paramétrées				✓
Multi-langues				✓

# Technologies employées



TypeScript



Jison \*



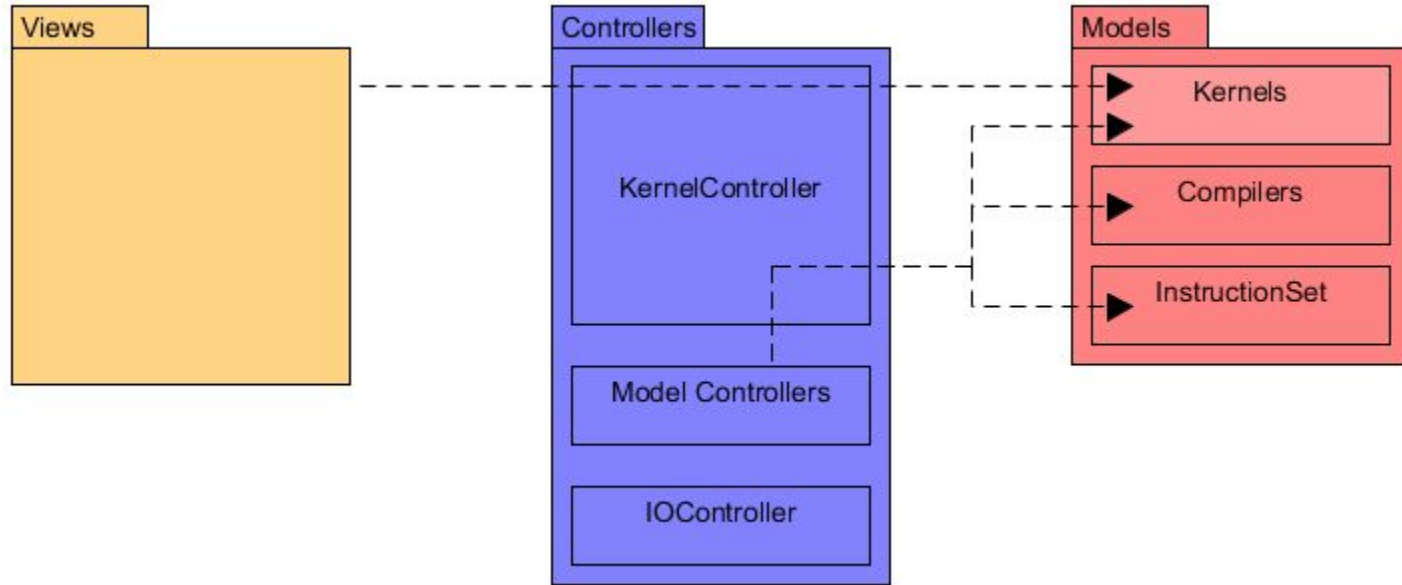
Backbone.js



Browserify

*\* Jison ne possède pas de logo. Celui présenté est là à titre d'illustration pour un parser (GNU bison)*

# Architecture MVC - Vue simplifiée



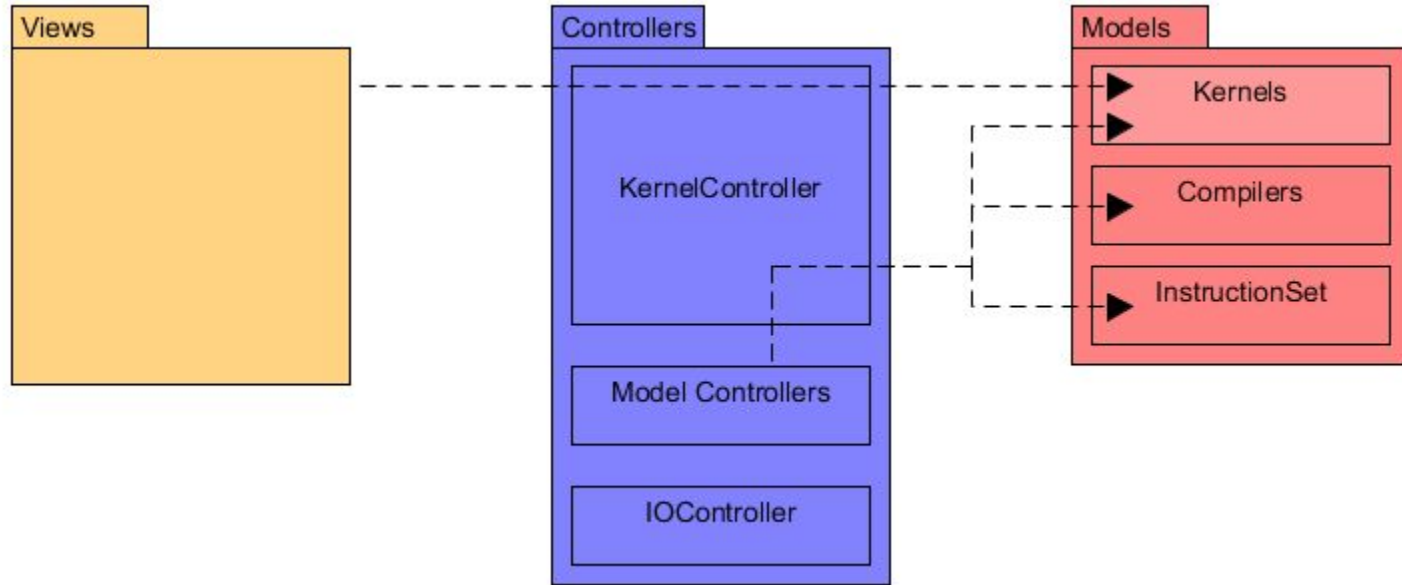


# Architecture MVC - Vue simplifiée

## Toolchain :

- ❖ noyau
- ❖ yas
- ❖ hcl2js
- ❖ jeu d'instruction

# Architecture MVC - Vue simplifiée



# Interface

Existant

Y86 Simulator ⚡ Assemble ⏮ Reset > Step » Continue 📁 Load Examples Wiki GitHub

SOURCE CODE	OBJECT CODE	MEMORY																																																																								
<pre>1   .pos 0 2   Init: 3     irmovl Stack, %ebp 4     irmovl Stack, %esp 5   6   .pos 0x100 7   Stack:</pre>		<table><thead><tr><th>ADDRESS</th><th>VALUE</th></tr></thead><tbody><tr><td>0000 00000000</td><td></td></tr><tr><td>0004 00000000</td><td>EBP ESP</td></tr><tr><td>0008 00000000</td><td></td></tr><tr><td>000c 00000000</td><td></td></tr><tr><td>0010 00000000</td><td></td></tr><tr><td>0014 00000000</td><td></td></tr><tr><td>0018 00000000</td><td></td></tr><tr><td>001c 00000000</td><td></td></tr><tr><td>0020 00000000</td><td></td></tr><tr><td>0024 00000000</td><td></td></tr><tr><td>0028 00000000</td><td></td></tr><tr><td>002c 00000000</td><td></td></tr><tr><td>0030 00000000</td><td></td></tr><tr><td>0034 00000000</td><td></td></tr><tr><td>0038 00000000</td><td></td></tr><tr><td>003c 00000000</td><td></td></tr><tr><td>0040 00000000</td><td></td></tr><tr><td>0044 00000000</td><td></td></tr><tr><td>0048 00000000</td><td></td></tr><tr><td>004c 00000000</td><td></td></tr><tr><td>0050 00000000</td><td></td></tr><tr><td>0054 00000000</td><td></td></tr><tr><td>0058 00000000</td><td></td></tr><tr><td>005c 00000000</td><td></td></tr><tr><td>0060 00000000</td><td></td></tr><tr><td>0064 00000000</td><td></td></tr><tr><td>0068 00000000</td><td></td></tr><tr><td>006c 00000000</td><td></td></tr><tr><td>0070 00000000</td><td></td></tr><tr><td>0074 00000000</td><td></td></tr><tr><td>0078 00000000</td><td></td></tr><tr><td>007c 00000000</td><td></td></tr><tr><td>0080 00000000</td><td></td></tr><tr><td>0084 00000000</td><td></td></tr><tr><td>0088 00000000</td><td></td></tr></tbody></table>	ADDRESS	VALUE	0000 00000000		0004 00000000	EBP ESP	0008 00000000		000c 00000000		0010 00000000		0014 00000000		0018 00000000		001c 00000000		0020 00000000		0024 00000000		0028 00000000		002c 00000000		0030 00000000		0034 00000000		0038 00000000		003c 00000000		0040 00000000		0044 00000000		0048 00000000		004c 00000000		0050 00000000		0054 00000000		0058 00000000		005c 00000000		0060 00000000		0064 00000000		0068 00000000		006c 00000000		0070 00000000		0074 00000000		0078 00000000		007c 00000000		0080 00000000		0084 00000000		0088 00000000	
ADDRESS	VALUE																																																																									
0000 00000000																																																																										
0004 00000000	EBP ESP																																																																									
0008 00000000																																																																										
000c 00000000																																																																										
0010 00000000																																																																										
0014 00000000																																																																										
0018 00000000																																																																										
001c 00000000																																																																										
0020 00000000																																																																										
0024 00000000																																																																										
0028 00000000																																																																										
002c 00000000																																																																										
0030 00000000																																																																										
0034 00000000																																																																										
0038 00000000																																																																										
003c 00000000																																																																										
0040 00000000																																																																										
0044 00000000																																																																										
0048 00000000																																																																										
004c 00000000																																																																										
0050 00000000																																																																										
0054 00000000																																																																										
0058 00000000																																																																										
005c 00000000																																																																										
0060 00000000																																																																										
0064 00000000																																																																										
0068 00000000																																																																										
006c 00000000																																																																										
0070 00000000																																																																										
0074 00000000																																																																										
0078 00000000																																																																										
007c 00000000																																																																										
0080 00000000																																																																										
0084 00000000																																																																										
0088 00000000																																																																										
<table><thead><tr><th>REGISTERS</th><th>FLAGS</th></tr></thead><tbody><tr><td>%eax 0x00000000</td><td>SF 0 ZF 0 OF 0</td></tr><tr><td>%ecx 0x00000000</td><td></td></tr><tr><td>%edx 0x00000000</td><td>STATUS</td></tr><tr><td>%ebx 0x00000000</td><td>STAT AOK</td></tr><tr><td>%esp 0x00000000</td><td>ERR</td></tr><tr><td>%ebp 0x00000000</td><td>PC 0x0000</td></tr><tr><td>%esi 0x00000000</td><td></td></tr><tr><td>%edi 0x00000000</td><td></td></tr></tbody></table>		REGISTERS	FLAGS	%eax 0x00000000	SF 0 ZF 0 OF 0	%ecx 0x00000000		%edx 0x00000000	STATUS	%ebx 0x00000000	STAT AOK	%esp 0x00000000	ERR	%ebp 0x00000000	PC 0x0000	%esi 0x00000000		%edi 0x00000000																																																								
REGISTERS	FLAGS																																																																									
%eax 0x00000000	SF 0 ZF 0 OF 0																																																																									
%ecx 0x00000000																																																																										
%edx 0x00000000	STATUS																																																																									
%ebx 0x00000000	STAT AOK																																																																									
%esp 0x00000000	ERR																																																																									
%ebp 0x00000000	PC 0x0000																																																																									
%esi 0x00000000																																																																										
%edi 0x00000000																																																																										

Y86 Simulator ⚡ Assemble > Step » Continue 📁 Load 📁 Save Examples Wiki GitHub

Editor	Execution	InstructionSet	HCL
<b>SOURCE CODE</b>			
<pre>1   2   3   Init:   irmovl Stack, %esp   # Set up Stack pointer 4          jmp Main           # Execute main program 5   6          .align 4 7   array: .long 0xbad 8          .long 0xbad 9          .long 0xbad 10         .long 0xbad 11   12   Main:  irmovl 4, %eax       # Push 4 13          pushl %eax 14          irmovl array, %edx   # Push array 15          pushl %edx 16          call Sum             # Sum(array, 4) 17          popl %ebx 18          popl %edx 19          halt 20   21   Sum:   irmovl 4(%esp), %edx  # edx = Start 22          irmovl 8(%esp), %ecx  # ecx = Count 23          irmovl 0, %eax       # sum = 0 24   25   Loop:  irmovl (%edx), %ebx   # get *Start 26          addl %ebx, %eax      # add to sum 27          addl 4, %edx         # Start++ 28          loop Loop           # Stop when 0 29   30   End:   ret                  # result must be abcd in %eax 31   32         .pos 0x95 33   Stack: 34   35  </pre>			

En développement





# Découpage en onglets

❖ **Editor**

❑ Assemble

❖ **Execution**

❑ Reset

❑ Step

❖ **InstructionSet**

❑ Continue

❑ Load

❖ **HCL**

❑ Save

# Affichage des composants



Backbone.js

```
<div><%= composant %></div>
```



# Modularité

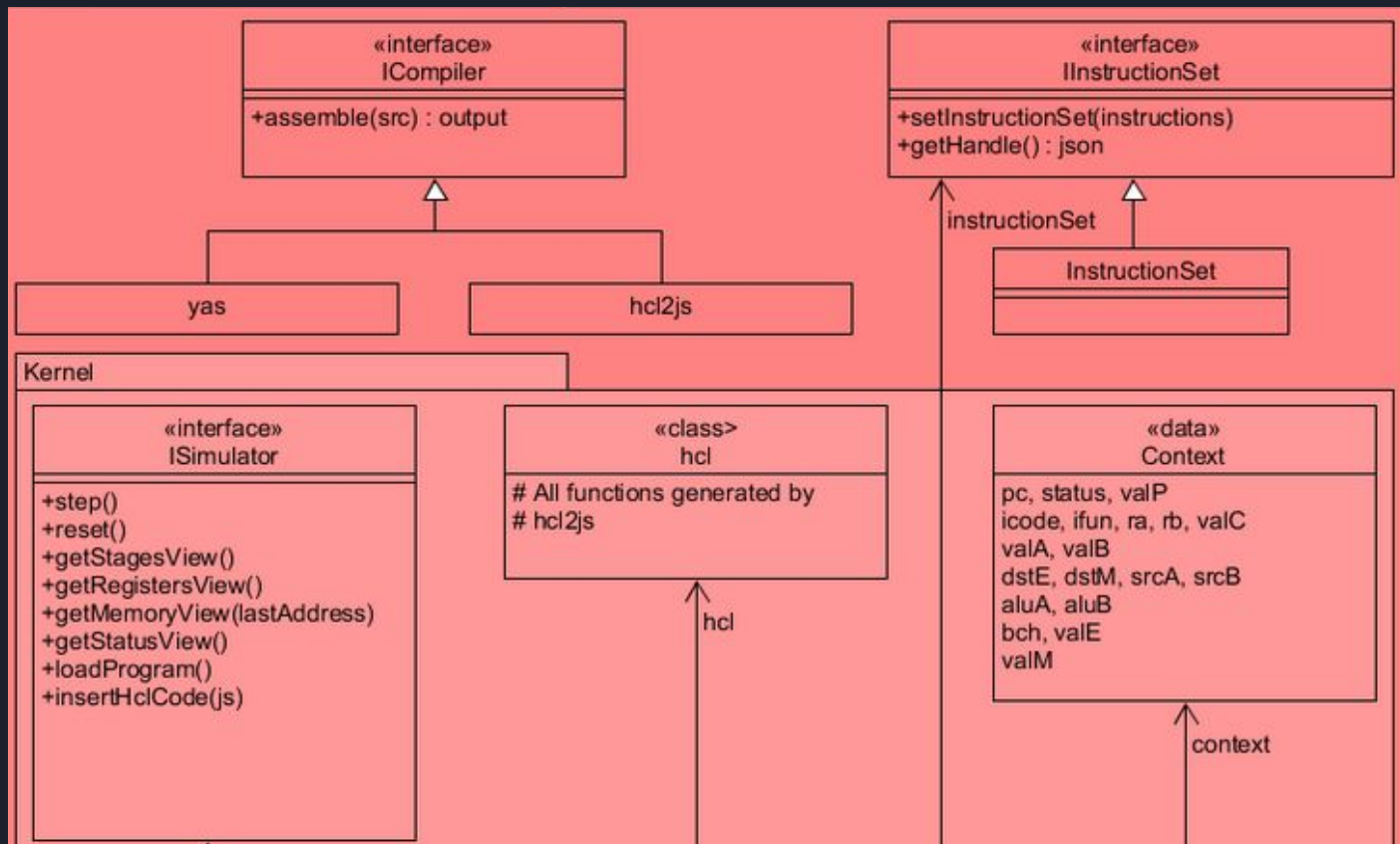
- ❖ Langue
- ❖ Apparence
- ❖ Paramétrage

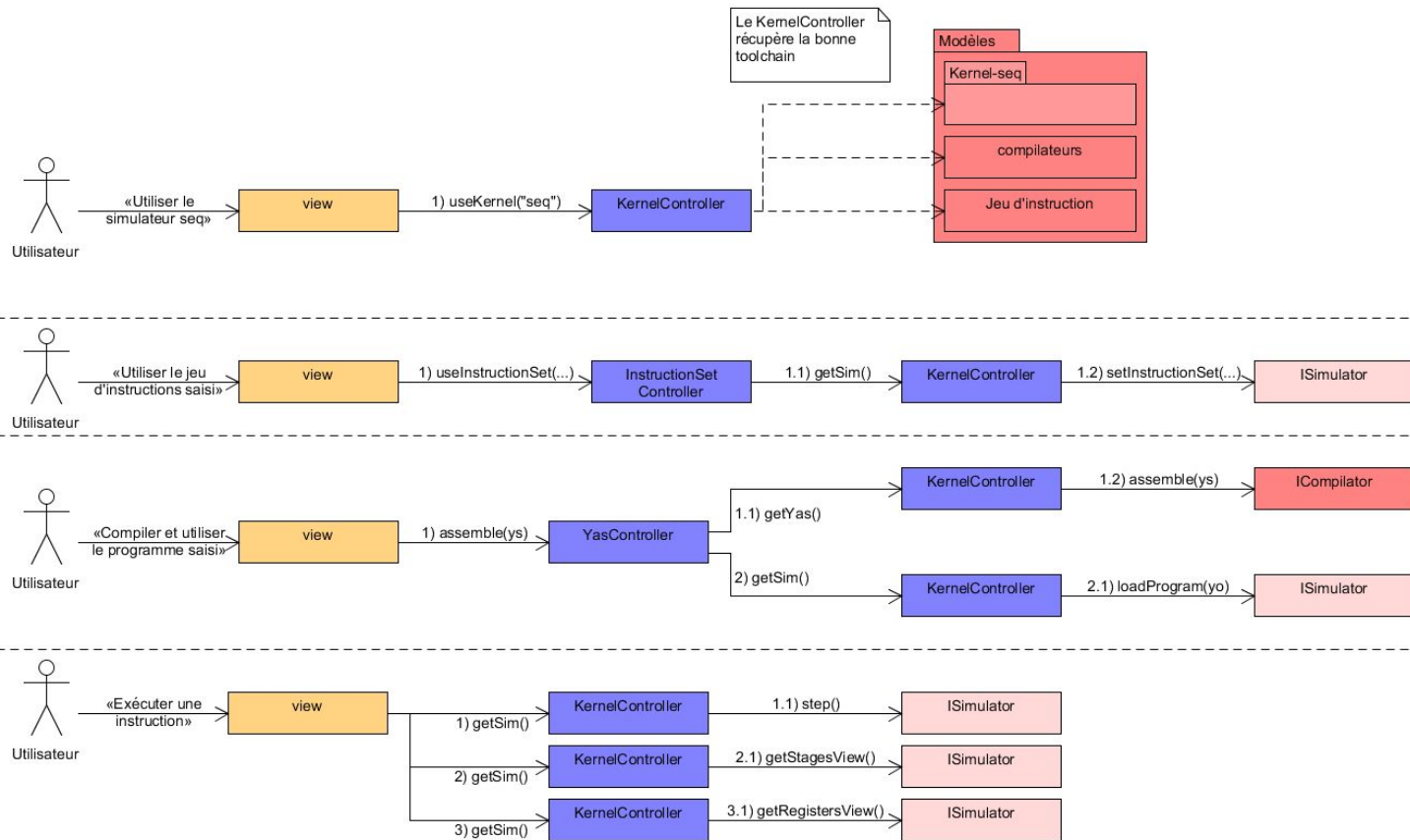
# Suite de tests sur l'interface



Jasmine

# Interfaces



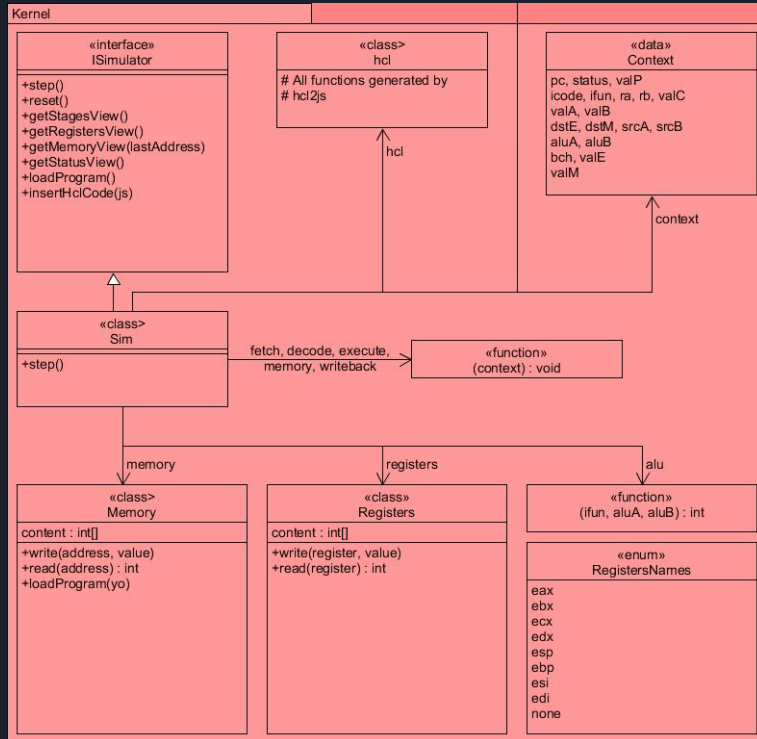


# Représentation d'un CPU

«interface» ISimulator
+step() +reset() +getStagesView() +getRegistersView() +getMemoryView(lastAddress) +getStatusView() +loadProgram() +insertHclCode(js)

- Chaque kernel hérite d'une interface
- Un kernel par CPU (seq, pipeline, 64bits...)
- Le câblage, bus, du CPU sont pré-déterminé pour chaque stage
- Chaque CPU possèdent ses tests unitaires.

# Kernel séquentiel



Composé de 3 unités :

- Mémoire
- Registres
- ALU

Le contexte pour les valeurs des bus.

Simulation complète des stages grâce aux fonction compilé du HCL





# Roadmap

- hcl2js
- kernel-seq : ALU, mémoire, registres
- Squelette de l'interface

Fait

- yas
- Jeu d'instruction
- Contrôleurs
- Test du kernel seq
- Modularité de l'interface
- Onglets :
  - Éditeur
  - Exécution
  - HCL

En cours

- Dump de la mémoire
- Interface pour les requêtes paramétrées
- Onglet Jeu d'instructions
- Fenêtre de paramétrage

À faire



# Questions

