

# HORLOGE

Aimeric DUCHEMIN, Amédée LEBERRE, Paul RAPHAEL,  
Yann Viegas

ENS Ulm

13 janvier 2025

- 1 Fonctionnement du microprocesseur
  - Instruction set

- 1 Fonctionnement du microprocesseur
  - Instruction set

# Instruction set

## Proposition 1.1 : Conventions

On choisit une architecture la plus proche possible de celle du cours. La taille d'un mot est de 32 bits. On représente les addresses sur 16 bits.

<i>Instruction</i>	<i>Encoding</i>	<i>Description</i>
<i>ADD</i>	01 rs1 rs2 00	rs1 <- \$rs1 + \$rs2
<i>SUB</i>	02 rs1 rs2 00	rs1 <- \$rs1 - \$rs2
<i>XOR</i>	03 rs1 rs2 00	rs1 <- \$rs1 ^ \$rs2
<i>OR</i>	04 rs1 rs2 00	rs1 <- \$rs1   \$rs2
<i>AND</i>	05 rs1 rs2 00	rs1 <- \$rs1 & \$rs2
<i>ADDIMM</i>	45 rs1 rs2 00	rs1 <- imm   \$rs1
<i>NOT</i>	06 rs1 rs2 00	rs1 <- ~\$rs2
<i>LSHIFT</i>	07 rs1 rs2 00	rs1 <- \$rs1 « \$rs2
<i>RSHIFT</i>	08 rs1 rs2 00	rs1 <- \$rs1 » \$rs2
<i>LOADROM</i>	09 rs1 rs2 00	rs1 <- M[\$rs2]
<i>LOADRAM</i>	14 rs1 rs2 00	rs1 <- M[\$rs2]

<i>Instruction</i>	<i>Encoding</i>	<i>Description</i>
<i>MOVIMM</i>	49 rs1 rs2 00	rs1 <- \$imm
<i>STORE</i>	2A rs1 rs2 00	M[\$rs2] <- \$rs1
<i>MOV</i>	0B rs1 rs2 00	rs1 <- \$rs2
<i>NONZERO</i>	82 rs1 rs2 00	rs1 <- \$rs2
<i>JMP</i>	80 rs1 00 00	PC += \$rs1
<i>JMPIMM</i>	81 00 imm	PC += imm

# Structure

Le projet est structuré en plusieurs fichiers. `alu.py` contient les fonctions relatives à l'ALU. `registers.py` permet de créer et connecter les registres. Le `main` gère la lecture du code dans la ROM. Enfin, `utils.py` contient des fonctions telles que `gigamux` (permettant de faire la sélection de registres, d'instructions etc)

## Exemple : Fibonacci

```
MOV rdx $15  
MOV rbx $0  
MOV rcx $1
```

```
MOV r2x $1  
fibo :  
MOV r1x rcx  
ADD rcx rbx  
MOV rbx r1x
```

```
MOV rax rcx  
SUB rdx r2x  
NONZERO rdx  
JMP 'end  
JMP 'fibo
```

```
end :  
MOV rbx $42
```

```
.data
```



# Circuit de fonctionnement

dessin du circuit (j'en ai un dans mes notes de cours ; faut juste que je le redessine au propre)

# Carry Lookahead

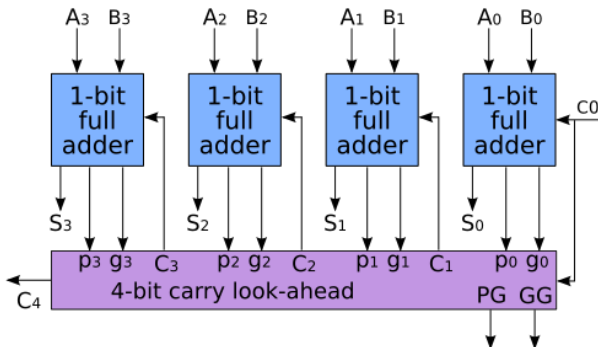


Figure – Circuit fabriquant un carry lookahead

# La Clock

fonctionnement clock

# Conclusion

- sysnum c a chier
- iris c une grosse pute