

Rapport architecture microprocesseur

HORNERO Baptiste
HERZLICH Raphaël
HELARY Axel

Architecture du processeur

L'architecture du processeur est une architecture 32 bit proche de l'architecture Risc-V, les instructions sont codées sur 32 bits. Le programme sera lu sur une ROM, avec un pointeur se décalant de 1 à chaque tour de netlist. On a 16 registres accessibles, numérotés de 0 à 15, chacun de taille 32. On ajoute un registre de 32 bits correspondant à la position du pointeur sur la ROM, qu'on appelle P. On se munit de trois flags : ZF (dernière opération a renvoyé 0 ou non), SF (positivité de la dernière opération), OF (overflow de la dernière opération arithmétique) tous de taille 1.

Instructions

Le set d'instructions est le suivant:

Instruction	Encodage	Description	Arguments
NOP	0000 0000	No opération	
ADD	0000 0001	Addition	rs1 rs2
SUB	0000 0010	Soustraction	rs1 rs2
MUL	0000 0011	Multiplication	rs1 rs2
AND	0000 0100	Et logique	rs1 rs2
NOT	0000 0101	Non	rs1
OR	0000 0110	Ou logique	rs1 rs2
XOR	0000 0111	Xor logique	rs1 rs2
SLL	0000 1000	Décalage gauche logique	rs1
SRL	0000 1001	Décalage droite logique	rs1

et :

Instruction	Encodage	Description	Arguments	Description formelle
MOV	0000 1010	Met un registre à la valeur d'un autre registre	rs1 rs2	$rs1 \leftarrow rs2$
MOVI	0000 1011	Met une valeur immédiate dans un registre	rs1 immediate	$rs1 \leftarrow immediate$
JMP	0000 1100	Jump à une valeur	immediate	$P \leftarrow immediate$
CMP	0000 1101	Compare deux valeurs	rs1 rs2	Mise à jour des flags
JE	0000 1110	Jump si égal (ZF = 1)	immediate	Si ZF = 1 alors $P \leftarrow immediate$
JNE	0000 1111	Jump si non égal (ZF = 0)	immediate	Si ZF = 0 alors $P \leftarrow immediate$
JGE	0001 0000	Jump si plus grand ou égal	immediate	Si OF=SF alors $P \leftarrow immediate$
LOAD	0001 0001	Lit dans la ram	rs1 rs2	$rs1 \leftarrow R[rs2]$
STORE	0001 0010	Stocke dans la ram	rs1 rs2	$R[rs1] \leftarrow rs2$

Si le temps nous le permet

Interface graphique

Implémentation d'une petite interface graphique et des instructions suivantes pour gérer l'interface

Instruction	Encodage	Description	Arguments
PIX	0010 0001	Change la valeur du pixel	rs1 rs2
CLEAR	0010 0010	Met toute l'interface graphique à 0	