

Présentation Systèmes Numériques

HERZLICH Raphaël, HELARY Axel, HORNERO Baptiste

14 janvier 2025

Sommaire

Principe général

Assembleur

Horloge

Principe général

Architecture

Jeu d'instruction

Assembleur

Horloge

Architecture

On implémente:

- ▶ 16 registres de taille 32
- ▶ P qui donne la position du pointeur sur la ROM
- ▶ ZF, SF, OF trois flags de taille 1

Jeu d'instruction

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

Jeu d'instruction

Instruction	Encodage	Description formelle
MOV	0110 0001	$rs1 \leftarrow rs2$
MOVI	0111 0001	$rs1 \leftarrow \text{immediate}$
CMP	1000 0001	
JMP	0100 0001	$P \leftarrow \text{immediate}$
JNE	0100 0010	Si $ZF = 0$ alors $P \leftarrow \text{immediate}$
JE	0100 0011	Si $ZF = 1$ alors $P \leftarrow \text{immediate}$
JGE	0100 0100	Si $OF = SF$ alors $P \leftarrow \text{immediate}$
LOAD	0101 0001	$rs1 \leftarrow R[rs2]$
STORE	0101 0010	$R[rs2] \leftarrow rs1$
LOADFIX	0101 0011	$rs1 \leftarrow R[\text{immediate}]$
STOREFIX	0101 0100	$R[\text{immediate}] \leftarrow rs1$

Principe général

Assembleur

Horloge

L'assembleur

- ▶ Fait en OCaml : Ocamllex + Menhir
- ▶ Syntaxe : [NOM DE L'OP] *arg1 arg2*;
- ▶ Ajout de variables nommées, de label et commentaires.
- ▶ Syntaxe similaire à l'appel machine pour toutes les opérations sauf pour les MOV

Fonctionnement des MOV

- ▶ $\text{reg} \leftarrow \text{reg} : \text{MOV } \%rnumReg1 \%rnumReg2;$
- ▶ $\text{var} \leftarrow \text{reg} : \text{MOV } \%rnumReg \text{ varName};$
- ▶ $\text{var} \leftarrow \text{reg} : \text{MOV } \text{varName } \%rnumReg;$
- ▶ $\text{reg} \leftarrow \text{var} \text{ (par pointeur)} : \text{LOAD } \%rnumReg1 \%rnumReg2;$
- ▶ $\text{reg} \leftarrow \text{var} \text{ (par pointeur)} : \text{LOAD } \%rnumReg1 \text{ numReg2};$
- ▶ $\text{reg} \leftarrow \text{immediate} \text{ (Valeur immédiate ou pointeur de variable)} : \text{MOV } \%rnumReg1 (\$immediate | * \text{varName});$

Fonctionnement des Variables

- ▶ Fonctionne avec la RAM (2^{16} variables possibles (32-bit))
- ▶ Pas besoin de déclarer les variables(sauf pour les tableaux)
- ▶ Remplacé pendant la compilation par leur numéro (avec un environnement)
- ▶ Même environnement utilisé pour les pointeurs

Principe général

Assembleur

Horloge

Utilité

Ce que l'on fait

Notre horloge permet de:

- ▶ passer les secondes, les minutes, les heures, les jours, les jours de la semaine, les mois et les années
- ▶ détecter les années bissextiles et de modifier le nombre de jours de février en conséquence
- ▶ choisir la date de début

Notre horloge débutera à l'année 1970.

Les problèmes

Nous avons rencontré les problèmes suivants:

- ▶ Netlist trop grande en utilisant des boucles for
- ▶ Netlist trop grande à cause des multiplications
- ▶ Implémentation peu efficace de SLL