



Projet de CA

Un mini-JavaScript sur circuit FPGA

Rajith RAVINDRAN

12 mai 2024

Établissement / Formation : Sorbonne Université, Master 1 STL
Année de Formation : Semestre 2, 2023-2024

Table des matières

1	Introduction	2
2	vm.ecl	2
2.1	Présentation du fichier	2
3	value.ecl	3

1 Introduction

Le langage Eclat, présenté durant le cours, est un langage à la fois fonctionnel et impératif, conçu pour la programmation parallèle et synchrone de circuits. Les programmes écrits en Eclat sont convertis en descriptions VHDL, destinées à être synthétisées sur des cartes FPGA. À titre d'exemple d'utilisation pratique d'Eclat, une version simplifiée de la machine virtuelle OCaml a été développée dans ce langage. Cela permet non seulement l'exécution directe de programmes OCaml via un circuit spécialisé sur FPGA, mais également l'invocation de fonctions externes Eclat, c'est-à-dire des accélérateurs matériels, depuis les programmes OCaml.

Le projet actuel cherche à reproduire cette expérience avec microJS, un mini-langage fonctionnel et impératif inspiré de Javascript. La tâche consiste à réimplémenter en Eclat la machine virtuelle utilisée précédemment dans le cours de Compilation de 3I018. Cette machine, nommée VM3I018, est une machine à pile qui exécute du bytecode généré à partir du langage MicroJS.

2 vm.ecl

2.1 Présentation du fichier

Dans le cadre de l'étude sur la mise en œuvre de machines virtuelles pour des langages fonctionnels-impératifs parallèles et synchrones, cette recherche examine la conceptualisation et la simulation d'une machine virtuelle à l'aide du langage Eclat.

Le cœur de ce projet est la définition et l'exécution des instructions au sein de la VM, qui manipule un état interne complexe pour réaliser les opérations spécifiées par un bytecode. Les types de données définis, tels que *instr* et *vm_state*, jouent un rôle crucial en modélisant respectivement les instructions exécutables et l'état complet de la VM à tout moment de son cycle d'exécution.

Les fonctions comme *vm_run_instr* et *vm_run_code* sont au cœur de la logique d'exécution, permettant la simulation du comportement de la VM en traitant des instructions variées telles que la gestion de la mémoire, le contrôle de flux et l'exécution de fonctions.

Le développement de ce projet met en lumière les défis et les stratégies in-

hérents à la création de machines virtuelles spécifiques, révélant des insights pratiques pour l'implémentation efficace de langages spécialisés sur des architectures matérielles comme les FPGA, en utilisant des descriptions VHDL générées par des compilations d'Eclat.

Ajout des instr .

3 value.ecl

Ajout d'une value fermeture afin de pouvoir l'utiliser avec Call. Pas d'ajout de prim.