

Présentation du projet de sysnum

Ryan LAHFA, Constantin GIERCZAK-GALLE, Julien MARQUET, Gabriel DORIATH DÖHLER

Introduction

Parce que c'est notre projet !

Le projet se divise en deux sous-projets :

- Le processeur Minecraft avec l'ISA V-RISC-V¹ ;
- Le processeur RISC-V écrit en System Verilog et simulé avec Verilator

¹Invention de cerveaux malades.

Plan pour Minecraft

- Motivations

Plan pour Minecraft

- Motivations
- Redstone

Plan pour Minecraft

- Motivations
- Redstone
- ISA

Plan pour Minecraft

- Motivations
- Redstone
- ISA
- Détails d'implémentation

Plan pour RISC-V

- Fonctionnalités principales du processeur : extensions, entrées-sorties

Plan pour RISC-V

- Fonctionnalités principales du processeur : extensions, entrées-sorties
- Prototypes

Plan pour RISC-V

- Fonctionnalités principales du processeur : extensions, entrées-sorties
- Prototypes
- Icarus Verilog → Verilator et Verilog → System Verilog

Plan pour RISC-V

- Fonctionnalités principales du processeur : extensions, entrées-sorties
- Prototypes
- Icarus Verilog → Verilator et Verilog → System Verilog
- Caches, MMU

Plan pour RISC-V

- Fonctionnalités principales du processeur : extensions, entrées-sorties
- Prototypes
- Icarus Verilog → Verilator et Verilog → System Verilog
- Caches, MMU
- Wishbone

Plan pour RISC-V

- Fonctionnalités principales du processeur : extensions, entrées-sorties
- Prototypes
- Icarus Verilog → Verilator et Verilog → System Verilog
- Caches, MMU
- Wishbone
- Vérification formelle avec SymbiFlow

Plan pour RISC-V

- Fonctionnalités principales du processeur : extensions, entrées-sorties
- Prototypes
- Icarus Verilog → Verilator et Verilog → System Verilog
- Caches, MMU
- Wishbone
- Vérification formelle avec SymbiFlow
- Contrôleur VGA

Minecraft

Motivations

Minecraft : circuits logiques avec de la redstone².

²cf slide suivante

Motivations

Minecraft : circuits logiques avec de la redstone².

Déjà quelques implémentations existantes de CPU plus ou moins complexes.

²cf slide suivante

Motivations

Minecraft : circuits logiques avec de la redstone².

Déjà quelques implémentations existantes de CPU plus ou moins complexes.

But : implémenter un CPU 8-bits simple dans Minecraft ; contraintes surtout liées au jeu.

²cf slide suivante

Redstone

Redstone : poudre qui, placée au sol, forme des fils. Valeurs : 0 ou 1³.

³Subtilité : il y a des histoires de puissance... Out of the scope pour cette présentation

Redstone

Redstone : poudre qui, placée au sol, forme des fils. Valeurs : 0 ou 1³.

Un agencement d'éléments (fils de redstone, torches de redstones, blocs, etc.) forme un **circuit logique combinatoire**. Propagation non instantanée : facteur à prendre en compte (naïvement, ≥ 0.1 seconde pour qu'un signal parcourt 16 blocs) → limitation en taille.

³Subtilité : il y a des histoires de puissance... Out of the scope pour cette présentation

Redstone

Redstone : poudre qui, placée au sol, forme des fils. Valeurs : 0 ou 1³.

Un agencement d'éléments (fils de redstone, torches de redstones, blocs, etc.) forme un **circuit logique combinatoire**. Propagation non instantanée : facteur à prendre en compte (naïvement, ≥ 0.1 seconde pour qu'un signal parcourt 16 blocs) → limitation en taille.

Quelques timings ajustés et des fonctionnalités de Minecraft permettent de faire des latchs : sauvegarde de données.

³Subtilité : il y a des histoires de puissance... Out of the scope pour cette présentation

Exemple redstone



Fig. 1 : AND gate

Exemple redstone



Fig. 2 : Clock

ISA : V-RISC-V

V-RISC-V = Very Reduced Instruction Set Computer (-V pour le jeu de mot)

Données sur 8 bits, instructions sur 32 bits (pour l'instant 27 bits utilisés).

ISA : V-RISC-V

V-RISC-V = Very Reduced Instruction Set Computer (-V pour le jeu de mot)

Données sur 8 bits, instructions sur 32 bits (pour l'instant 27 bits utilisés).

- STORE
- LOAD
- ADD
- OR
- XOR
- LOADI
- JMP conditionnel

ISA : V-RISC-V

V-RISC-V = Very Reduced Instruction Set Computer (-V pour le jeu de mot)

Données sur 8 bits, instructions sur 32 bits (pour l'instant 27 bits utilisés).

- STORE
- LOAD
- ADD
- OR
- XOR
- LOADI
- JMP conditionnel

| pc : 1 | flag : 2 | or,carry,xor : 3 | read1 : 4 |
imm : 0:3 | write : 4 | imm : 4:7 | read2 : 4

Pseudo-instructions

Avec les instructions de base et les registres spéciaux :

- NOP
- SUB
- HALT
- PRINT
- JMP (inconditionnel)
- MOV
- NOT
- CMP

Registres

16 general purpose⁴ registers : %0 to %15.

⁴Enfin dont 3 registres spéciaux

Registres

16 general purpose⁴ registers : %0 to %15.

Largeur : 8 bits

⁴Enfin dont 3 registres spéciaux

Registres

16 general purpose⁴ registers : %0 to %15.

Largeur : 8 bits

Registres spéciaux :

- %0 = 0 → NOP
- %1 = -1 → NOT
- %15 = random(0, 255)

⁴Enfin dont 3 registres spéciaux

Assembler

On a un assembler de notre ASM vers des commandes MC permettant de programmer des ROM dans Minecraft.

Aspects techniques

Laines



Fig. 3 : Laines

Protections hardware



Fig. 4 : Protection side-channel attack

Conclusion et ouvertures

Achievements :

CPU V-RISC-V avec ROM, registres, ALU, instructions arithmétiques et logiques.

⁵O : afficheurs 7-segments ; I : sélecteurs à leviers

⁶En fait déjà presque possible...

Conclusion et ouvertures

Achievements :

CPU V-RISC-V avec ROM, registres, ALU, instructions arithmétiques et logiques.

TODO :

RAM, I/O utilisateur⁵, découpage de l'espace mémoire, pipeline⁶

⁵O : afficheurs 7-segments ; I : sélecteurs à leviers

⁶En fait déjà presque possible...

Conclusion



Fig. 5 : Serveur

Le processeur RISC-V (Sakaido, le brillant)

Fonctionnalités principales

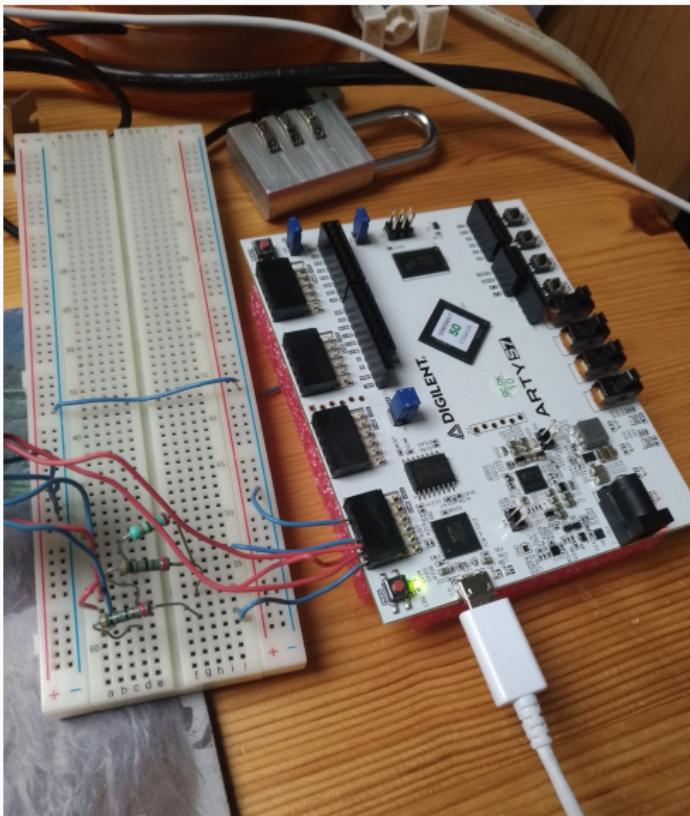
Il s'agit d'un processeur RISC-V

- Implémente RV32I⁷
- Implémentation pipelinée
- Communication par Wishbone avec la mémoire

⁷RV32IM était disponible à un moment

Contrôleur VGA

Contrôleur VGA 640x480.



Contrôleur VGA



Contrôleur VGA

Affichage d'une matrice de caractères



Pipeline

- 5 étages : IF, ID, EXE, MEM, WB
- Forwarding
 - Essentiellement entre MEM et EXE
- Prédiction : « predict not taken »
 - Signal KILL

Pipeliner RISC-V

RISC-V est conçu pour des implémentations selon ce modèle de pipeline.

Communication avec la mémoire

- Les accès mémoire peuvent prendre un temps arbitraire
- Il faut donc mettre le processeur en pause le temps qu'ils soient traités
- Signal STALL
- Petite machine à états (exécution normale / attente de données)