



SORBONNE UNIVERSITÉ

M1 SESI

Modélisation SystemC d'un RISC-V pipeline

Rapport d'avancement
dans le cadre du projet PSESI

Etudiants:

M. Timothée Le Berre
M. Louis Geoffroy Pitailler
M. Kevin Lastra

Encadrante :

Mm. Daniela Genious

Nous souhaitons remercier
Mme. Daniella Genius, M. Pirouz Bazargan Sabet et M. Franck Wajsbürost
qui nous ont énormément aidé lors de la réalisation de ce projet.

Table de Matières

1	Introduction	2
2	Objectifs :	5
3	Situation actuelle du projet :	6
3.1	Résumé graphique :	6
4	Problèmes rencontrés lors de la conception :	6
5	Objectifs restants	7
6	Bibliographie :	8

1 Introduction

En 1971 Intel sort son premier micro-processeur, l'Intel 4004 basé sur une architecture 4 bits CISC (Complex Instruction Set Computing).

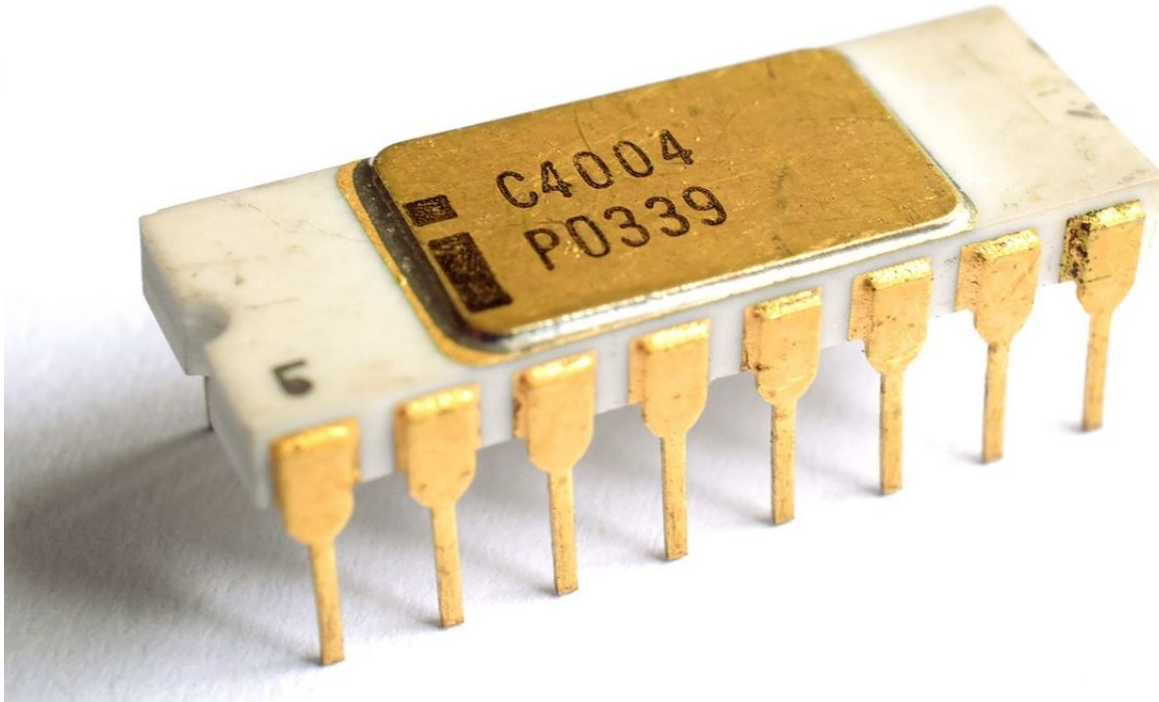


Figure 1: Intel 4004, source : <https://www.lesnumeriques.com/cpu-processeur/l-intel-4004-premier-processeur-du-fondeur-fete-aujourd-hui-ses-50-ans-n171113.html>

Les Processeurs CISC vont largement dominer le marché jusque dans les années 80 où une nouvelle architecture va faire son apparition : l'architecture RISC.

Les architectures CISC sont beaucoup plus complexes que les risques, en effet ces derniers implémentent des fonctions très complexes en matériel. On peut par exemple citer le Intel 8086 qui implémente matériellement des instructions permettant de faire des comparaisons entre des chaînes de caractères.

Les architectures RISC au contraire implémentent uniquement des fonctions basiques et simples de manière matérielle, la philosophie du RISC étant en effet de laisser les tâches complexes au compilateur.

RISC était à l'origine un projet mené par David Patterson à l'Université de Berkeley en Californie entre 1980 et 1984.

Cette architecture va vite montrer de gros avantages par rapport à l'architecture CISC et de nombreux projets vont se développer en se basant dessus.

En 1981 le MIPS (Microprocessor without Interlocked Pipeline Stages) -qui se base sur une architecture de type RISC- va faire son apparition à l'Université de Stanford.

La technologie MIPS va être commercialisée à partir de 1984 et sa première implémentation, le R2000, va devenir l'un des processeurs les plus utilisés pour l'embarqué.

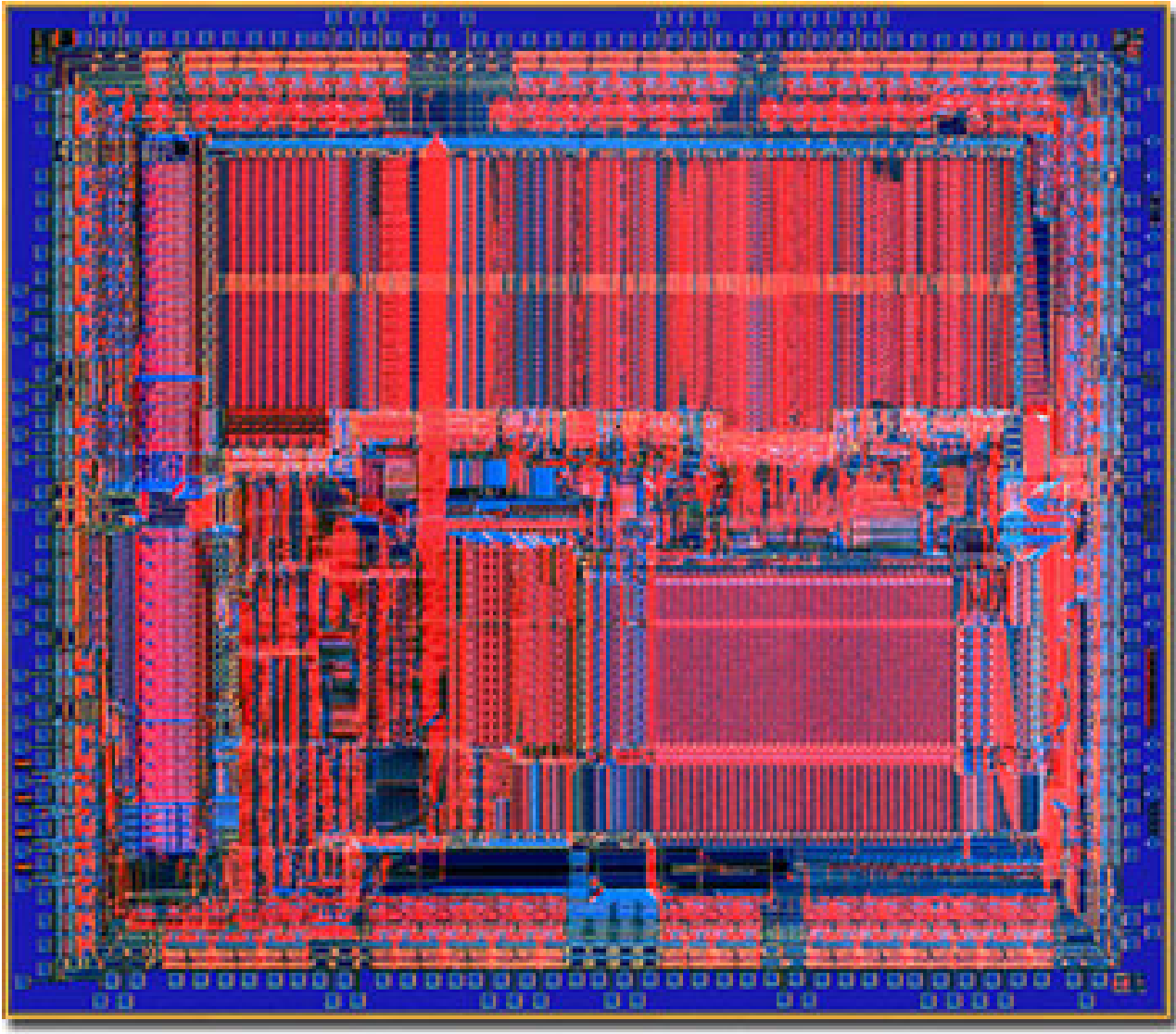


Figure 2: Intel 4004, source : <https://www.cpushack.com/MIPSCPU.html>

Grâce à sa simplicité de compréhension et son efficacité le MIPS s'impose comme un standard dans l'enseignement de l'architecture processeur, c'est pourquoi le lip6 l'a choisi comme base de son enseignement.

Néanmoins cette architecture a fait son temps et commence peu à peu à se montrer trop vieille. C'est pourquoi il a été décidé de changer la base des cours de Sorbonne Université utilisant en utilisant l'architecture RISCv, une architecture libre, facile à implémenter, proche du MIPS et surtout utilisé à l'heure actuelle.

Au cours de notre premier semestre de M1 SESI nous avons eu l'occasion d'implémenter une architecture ARMv2a en VHDL. Cela nous a énormément plu et lorsque Mme. Genius a proposé une implémentation d'une architecture RISCv en systemC nous avons tout de suite postulé afin de pouvoir participer à ce projet.

Ce projet nous a permis de nous familiariser avec le jeu d'instruction RISCv et avec le langage SystemC. RISCv étant l'une des implémentations de RISC les plus importantes - au côté de l'architecture ARM -, il est très intéressant d'en étudier son fonctionnement.

Ne voulant pas simplement réaliser une architecture scalaire et voulant aller plus loin que ce que nous avons

déjà eu l'occasion de faire en VLSI avec l'architecture ARM, nous avons décidé d'avancer rapidement sur le projet dans le but de finir début mars l'implémentation scalaire et d'ensuite pouvoir nous concentrer sur une implémentation SS2.

Néanmoins après discussion avec notre encadrente il a été décidé de commencer d'abord par l'ajout d'une partie Kernel à notre design et d'ensuite passer à l'implémentation SS2 s'il nous restait du temps.

2 Objectifs :

L'objectif premier de notre projet était d'implémenter une architecture RISC-V 32bits pipeliné à 5 étages sans extension, dans le but de remplacer l'architecture MIPS du Lip6. Les étages de notre implémentation sont les mêmes qu'une MIPS R3000, à savoir :

- IFETCH
- DECODE
- EXECUTE
- MEMORY
- WRITE-BACK

Le langage imposé pour cette réalisation est systemC. Notre implémentation devait donc être proche de l'architecture MIPS que nous avons étudié en cours, d'où l'implémentation sous forme d'un pipeline 5 étages. Une fois l'implémentation terminée nous devions également mettre en place une plateforme de TP.

Finalement il a été décidé que la mise en place de la plateforme de TP n'était pas prioritaire et que l'implémentation de la partie Kernel primait.

Notre objectif est donc de mettre en place une architecture RISC-V scalaire avec partie Kernel et s'il nous reste du temps d'implémenter un SS2 et enfin de mettre en place une plateforme de TP.

3 Situation actuelle du projet :

A l'heure où nous écrivons ce rapport, nous avons fini la partie scalaire et le nettoyage du code en accord avec les conventions que nous avons pris pour le nom des signaux.

Notre processeur ne possède pas encore de bypass qui sont en cours de création mais il est capable de compiler du C ou de l'assembleur.

L'implementation RISCv que nous avons surnommé v1.0 est a ce jour totalement finalisé. Nous avons en effet réalisé multiples test C tel que la suite de Fibonacci recursive ou encore un algortihme de calcul de PGCD.

3.1 Résumé graphique :

Tous les membres de notre projet ont participé à l'ensemble de la conception mais certains membres se sont plus concentré sur certaines tâches. Ainsi le tableau ci dessous indique qui a majoritairement contribué à la réalisation de la tâche.

	Janvier	Février	Mars	Avril	Mai	Juin				
Documentation RiscV	X									
Conception CORE RiscV v1.0										
Etagé IFETCH		X								
Etagé DECODE		X								
Etagé EXEC		X								
Etagé MEMORY		X								
Etagé WRITEBACK		X								
Débogage CORE		X								
Nettoyage du programme			X							
Conceptions Caches										
Mis a jour Mips3000R		X								
Documentations Mips3000R		X								

X : Représente le mois où la tâche a été finalisé.

Timothée Le Berre	
Louis Geoffroy Pitailler	
Kevin Lastra	

4 Problèmes rencontrés lors de la conception :

Plusieur problematiques on surgi depuis le debut du projets mais grace au travail d'equipe et a la bonne repartition des tâches nous les avons surpasse.

Au tour de la mis a jour du modele MIPS32, le manque de commentaires et les signals avec acronymys pas evident en notament rendu la tâche plus complique.

Au tour du developpement du modele RISCv v1.0, nous avons trouver plusieurs problematiques, au debut utilisant les connaissance aquis dans l'UE vlsi au premier semestre nous avons implementer une architecture armv2a pour la base, mais rapidement les difference en apparu donc a du etudier les difference a fond utilisant les specification technique du RISCv du 2019.

5 Objectifs restants

Le tableau ci dessous fait office de Roadmap, il s'agit d'un récapitulatif graphique des objectifs que nous nous sommes fixés et les deadlines correspondantes.

	Fin Février	Fin Mars	Fin Avril	Fin Mai	Juin
Debugage & implémentation des bypass		X			
Implémentation de la partie Kernel				X	
Implémentation SS2 & debugage complet					X

6 Bibliographie :

https://en.wikipedia.org/wiki/IBM_System/370

https://en.wikipedia.org/wiki/IBM_document_processorsIBM_801

Waterman, A., Lee, Y., Patterson, D., Asanovic, K., level Isa, V. I. U. (2014). The RISC-V instruction set manual. Volume I: User-Level ISA', version, 2.

Waterman, A., Lee, Y., Avizienis, R., Patterson, D. A., Asanovic, K. (2015). The risc-v instruction set manual volume 2: Privileged architecture version 1.7. University of California at Berkeley Berkeley United States.

Asanović, K., Patterson, D. A. (2014). Instruction sets should be free: The case for risc-v. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2014-146.

Utting, M., Kearney, P. (1992). Pipeline specification of a MIPS R3000 CPU. Technical Report 92-6, Software Verification Research Centre, Department of Computer Science, University of Queensland.

David A. Patterson John L. Hennessy (2021). Computer organization and design RISC-V edition, second edition.

Jurij Šilc, Jurij Silc, Borut Robic, Theo Ungerer (1999). Processor architecture : From dataflow to super-scalar and beyond.