RISC-V : un ISA 32 bits libre !

Ce jeu d'instructions (ISA : Instruction Set Architecture), développé initialement par l'université de Berkeley, est maintenant soutenu par une fondation regroupant quasiment tous les grands noms de l'industrie informatique, parmi lesquelles *Google*, *Nvidia*, *Western Digital*, *NXP Semiconductors*, *Microsemi*, *Bluespec*, *AMD*, *Qualcomm*, *IBM*, etc (liste complète sur https://riscv.org/members/)
En 2018, la Fondation Linux et la Fondation RISC-V ont annoncé leur collaboration conjointe pour permettre une nouvelle ère d'architecture ouverte. En 2019, certains membres étant « préoccupés par une possible perturbation géopolitique », la fondation RISC-V s'est installée en Suisse.

Présentation

C'est beaucoup plus qu'un énième processeur libre venant s'ajouter à la pile de processeurs «softcore» à utiliser sur un FPGA, comme l'on peut en trouver sur ***opencore*** et autre ***librecore***.

RISC-V (se prononce « risque failleve ») est une standardisation libre d'un jeu d'instructions avec une série d'extensions permettant de couvrir l'ensemble des domaines d'application des architectures 32/64 bits (voir même 128 bits, même si la définition n'est pas encore stabilisée, le temps d'en trouver une utilité). Pas de micro en 4, 8 ou 16 bits donc. Mais, même pour la très basse consommation, le 32 bits est désormais tres utilisé, comme on peut le voir avec les STM32. De plus, il existe une extension permettant d'utiliser des instructions compressées sur 16 bits, ainsi qu'une base avec moitié moins de registres (RV32E).

Cours Linux embarqué sur MangoPi MQ Pro

Le Mango Pi MQ Pro embarque le petit processeur Allwinner D1, une puce monocœur de 1 GHz, 64 bits, construite sur l'architecture RISC-V. La RAM peut aller de 512 Mo à 2 Go, et si le MQ Pro embarque suffisamment de ports (2x USB Type-C, HDMI, Wi-Fi et Bluetooth), ainsi que son GPIO 40 broches (compatible RaspberryPi Zero), pour fonctionner en solo, il existe également une carte porteuse en test pour ajouter Ethernet et USB Type-A.

On ne le trouve pas encore chez nos vendeurs habituels dont les fournisseurs fabriquent surtout autour des architectures ARM et voient d'un mauvais œil l'arrivée du RISC-V. Mais on en obtient à moins de 25 euro sur des sites comme AliExpress.com.

Evidemment ce n'est pas aussi « sportif » qu'avec des cartes Cortex M3 ou M4. Mais cela permet aux élèves d'apprendre à créer une image complète, comme avec le cours sur STM32F429I, et d'y développer ensuite des applications réseaux, en utilisant les sockets et TCP/IP, et d'autres graphiques avec C++/Qt par exemple.

Pour ceux qui souhaitent se pencher sur le code assembleur du RISC-V pour l'étudier en détail, ils pourront le faire avec des IDE comme :

- RARS l'adaptation de MARS à RISC-V par Benjamin Landers,
- AndeSight, l'IDE développé par AndesTech, l'un des membres de la Fondation RISC-V.

Un petit peu d’histoire

La formalisation de l'architecture RISC est née à Berkeley au début des années 80, à une époque où les principaux fondeurs de microprocesseurs menaient une course à qui aura le plus d’instructions, les plus grosses, les plus complexes... À cette époque, les constructeurs de processeurs ajoutaient des registres et des instructions de tailles et de formats différents, pour augmenter la puissance de calcul de leurs machines.

Cet ajout de multiples registres et instructions avait pour conséquence une augmentation du nombre de transistors, ainsi qu'une augmentation de la consommation. Cette course à l’échalote fonctionnait plutôt bien, en tenant compte de la loi de Moore. Et l'architecture x86 d'Intel est la preuve que cela fonctionne encore, puisque le format x86 domine toujours le marché des ordinateurs personnels et des serveurs.

Cependant, à Berkeley, les chercheurs tentèrent une échappée et proposèrent un nouveau modèle de microprocesseur, basé sur la simplicité. L’idée était qu’on a tout à gagner avec un jeu d'instructions plus simple, plus petit. En faisant des statistiques sur les instructions généralement utilisées par les compilateurs, ils s'étaient aperçus que la plupart de ces grosses instructions ne sont utilisées qu’à la marge, et que seule une portion minimale était massivement utilisée dans les programmes génériques.

Que de silicium gaspillé pour implémenter des instructions quasiment jamais utilisées, que d'énergie gaspillée pour cadencer le décodage des instructions devenues très complexes, que de cycles d’horloges gaspillés pour cadencer le cycle d'exécution de chaque instruction...

Dans leur publication initiale, les chercheurs ont donc défini le concept de RISC pour Reduced Instruction Set Computer, en opposition à CISC pour Complex Instruction Set Computer. À Stanford, une autre équipe a elle aussi surfé sur cette vague, en lançant un jeu d'instructions qui a son petit succès encore aujourd’hui : le MIPS. Jeu d'instructions qui vient d’ailleurs d’être libéré, en réaction à la montée de RISC-V…

Qu’est-ce qu’un processeur RISC-V ?

Un processeur RISC-V est une implémentation du jeu d’instructions (ISA) car le standard RISC-V ne définit que le « langage » du processeur. L’implémentation est à la charge du concepteur qui va décider en combien de cycles d'horloge s’exécutera une instruction, la taille du pipeline qu’il veut implémenter, comment gérer la prédiction de branches ou le format du bus de donnée (AXI, Wishbone...).

Ce même jeu d'instructions peut donc être implémenté sur un petit processeur avec seulement deux étages de pipeline, ou sur un très gros processeur avec réordonnancement des instructions « à la volée » (Out of Order pipeline). Le binaire sera compatible avec les deux processeurs. Seules les performances changeront.

Pour toute information complémentaire : **secretariat@seriane.org**