Implementació d'un Sistema RISC-V en *FPGA* amb suport per *Linux*

Informe inicial

Oscar Lostes Cazorla

Octubre 2021

1 Context i problemàtica a resoldre

RISC-V (RV, a partir d'ara) és un conjunt d'estàndards d'arquitectura de processadors, de naturalesa oberta i lliure, en contraposició amb arquitectures privatives com poden ser x86 o ARM.

Al llarg dels anys, s'han desenvolupat diverses implementacions de RV, principalment en entorns universitaris (com la universitat ETH de Zürich o Berkeley), tot i que també hi ha iniciatives comercials (com SiFive o Andes).

Si bé l'ecosistema RV és robust i prou madur, alhora presenta un problema d'usabilitat important si pretén guanyar adeptes més enllà de les esferes acadèmiques (on actualment l'arquitectura ja va tenint un pes important): els sistemes operatius.

Actualment, la majoria de sistemes implementats amb processadors RV cauen en tres categories principals:

- **Sistemes *baremetal*:** software corrent directament sobre el processador, sense sistema operatiu, i compilat de forma creuada en una plataforma tradicional.

  Aquests acostumen a ser els usos més acadèmics de RV, de cara a demostrar el seu funcionament. Però aquesta forma d'ús és poc *user friendly*.

- **Sistemes encastats:** plaques amb SO de baix impacte (estil Zephyr o FreeRTOS) orientats a aplicacions encastades.

  Si bé aquesta configuració és molt més versàtil que la *baremetal*, els SO encastats són bastant limitats i orientats a usos més industrials.

- **Plaques privatives:** la llicència de l'estàndard RV permet un ús comercial d'aquest. Aquest factor ha fet aparèixer en el mercat alguns productes amb RV, i amb capacitat d'executar *Linux* de forma relativament completa i fàcil.

  Si bé el fet d'emprar *Linux* salva l'objecció inicial que hem presentat sobre usabilitat, aquestes plaques són normalment costoses i per tant trenquen aquest avantatge.

2 Plantejament del TFG

Partint del problema presentat, s'estableix el següent objectiu de resolució: **implementar un sistema RISC-V sobre una *FPGA***, i donar-li una capacitat real mitjançant l'execució d'un **sistema operatiu *GNU/Linux***.

Dins d'aquest enunciat inicial, s'acoten els següents conceptes inicials, que s'haurien d'intentar mantenir dins del possible:

- S'opta per emprar *softcores* (és a dir, processadors RV que no es fabriquen *ad-hoc*) sintetitzats en *FPGA*.

  L'ús d'una *FPGA* en aquest cas permet fer proves i prototipat de forma àgil sense haver de comprar o fabricar múltiples plaques concretes.

- Per qüestions logístiques, es prefereixen plaques Altera/Terasic amb *FPGA* Cyclone d'Intel.

- L'objectiu del treball no és construir cap dels components individuals que conformen el sistema descrit, sinó integrar-los per a complir l'objectiu desitjat.

  Això significa que aquest treball **no** consisteix a dissenyar un processador RV o un SO *Linux*.

  Ara bé, aquest fet no impedeix que, a falta de components ja construïts que serveixin, es pugui modificar o refer material ja existent.

- Per tal d'acostar-se el màxim possible a l'usuari, l'objectiu és poder executar un gestor de finestres i entorn d'escriptori, i poder utilitzar-lo directament a través de teclat/ratolí i sortida de vídeo disponible a la *FPGA* emprada.

  Si no es pogués assolir, també serien acceptables entorns textuals o, com a últim recurs, connexions a terminal mitjançant el port sèrie.

3 Estat actual

El treball es troba actualment en una fase inicial d'investigació, per determinar les tecnologies més adequades a emprar. En concret, s'està buscant informació sobre:

- Requeriments del *Kernel* de *Linux* sobre RV i distribucions compatibles.
- *Softcores* que suportin *Linux* i *FPGA*s on s'hagin implementat.

Paral·lelament, es disposa d'una placa Terasic DE0 (amb *FPGA* Intel Cyclone III), candidata a ser el *host* del projecte. Per tant, i també com a pràctica addicional, s'estan investigant i testejant les següents qüestions:

- Mostra d'imatge per *VGA*.
- Càrrega de dades per SD. *(Pendent actualment)*
- Interacció amb PS/2. *(Pendent actualment)*

3.1 Requeriments del *Kernel* de *Linux* sobre RV i distribucions compatibles

Es conclou que, el suport estàndard de *Linux* per a RV és de 64 bits (perquè, en general, *Linux* ja té bastant abandonat els 32 bits).

D'altra banda, *Linux* requereix una sèrie de capacitats mínimes del processador. A RV, aquestes capacitats es materialitzen en diferents extensions de l'arquitectura base (l'arquitectura base només suporta manipulació d'enters). En aquest cas, *Linux* requereix l'extensió de propòsit general (extensió G), que comprèn un conjunt d'altres extensions (multiplicació/divisió d'enters, aritmètica de punt flotant de simple i doble precisió, i operacions atòmiques, principalment). També se suporta, en alguns casos, l'extensió per instruccions comprimides (extensió C).

Seguint la nomenclatura de RV, això implica que un processador compatible amb *Linux* ha de ser un processador **RV64GC**.

El següent pas és identificar quines distribucions (orientades a usuari, i no d'ús industrial) són compatibles amb RV. A continuació, es mostren les més importants (tot i haver-hi d'altres):

- Debian
- Ubuntu
- Fedora
- Gentoo

D'aquestes quatre distribucions, les que estan més al dia són Debian i Ubuntu.

3.2 *Softcores* que suportin *Linux* i *FPGA*s on s'hagin implementat

Analitzant els requisits de *Linux* anteriorment esmentats, es presenten els principals *softcores* (i els entorns *SoC* al seu voltant) que s'han considerat més adients per la tasca:

Podem veure una marcada tendència cap a les *FPGA* de Xilinx. Per tant, en un futur, i si els altres mètodes no funcionen, potser s'ha d'investigar les plaques Xilinx concretes, trobar una Altera similar i intentar convertir projectes d'una placa a l'altra manualment.

Ara bé, com a mínim en una fase inicial, sembla que es podrà evitar, ja que els mitjans de síntesi mostrats haurien de servir. Tant Chipyard com RocketChip sembla que generen codi Verilog independent de la *FPGA* de destí, i la llibreria GRLIB de Gaisler suporta diverses plaques Altera/Terasic.

Val la pena remarcar, que Rocket i BOOM són pràcticament el mateix *softcore* (com a mínim pel que fa a l'usuari). La diferència fonamental és que BOOM és un processador fora d'ordre (*Berkeley Out-Of-Order Machine*) i Rocket no.

Softcore	Developer	HW "Oficial"	Mitjà de síntesi
Rocket	UC Berkeley	Artix-7 35T Arty FPGA Xilinx Virtex-7 FPGA VC707 Xilinx Virtex UltraScale+ FPGA VCU118	Rocket Chip Generator Chipyard
BOOM			
Ariane / CVA6	ETH-Zurich	Digilent Genesys 2 Xilinx Vertex 7 – VC707	Projecte de Vivado Chipyard
NOEL-V	Gaisler	Digilent Arty-A7 Microsemi PolarFire Xilinx KCU105	Llibreria d'IP GRLIB

3.3 Mostra d'imatge per *VGA*

Ja que aquesta serà una tasca important del sistema, i com a forma de comprovar que la placa funciona amb el PC de desenvolupament, s'ha intentat generar imatges a través de la sortida *VGA* integrada a la placa DE0 de la que es disposa actualment.

Partint del software de demostració de la mateixa placa, s'ha pogut obtenir un codi Verilog que genera una imatge *hardcoded* i la representa correctament per pantalla. El codi s'ha testejat a diferents resolucions, i si bé totes funcionen i es veuen, el monitor de proves només ha reconegut correctament aquelles on el *timing* és una freqüència entera (aquest comportament es pot atribuir a la incapacitat del PLL de generar freqüències no enteres).

En una següent iteració de les proves, estaria bé poder carregar la imatge de memòria o, inclús millor, des de la SD. Tot i així, la utilitat final del codi actual (tal com està actualment) encara està per veure, ja que és molt probable que els *SoC* que se sintetitzin incorporin el seu propi sistema de vídeo.

4 Objectius i full de ruta actual

D'acord amb tota la informació exposada, es poden prendre una sèrie de decisions de cara a començar a moure el projecte cap a un terreny més pràctic. Evidentment, aquestes decisions en cap cas són finals, i estan subjectes a les dificultats que puguin sorgir al llarg de l'execució del projecte.

- **Distribució:** seguint l'argument principal de la usabilitat, s'intentarà emprar **Ubuntu** com distribució objectiu. Si això no fos possible, la següent opció acceptable seria Debian (per proximitat tècnica).
- ***Softcore/SoC*:** el *softcore* **NOEL-V** (mitjançant la **GRLIB** de Gaisler) és el que té un suport explícit per a plaques Altera/Terasic, i per tant sembla l'opció més adequada en primera instància. La següent opció que sembla viable és l'ús de Chipyard per a generar un *SoC* amb qualsevol dels altres 3 *softcores*.

Per tant, en les properes iteracions del projecte, l'objectiu serà començar a treballar segons aquestes eleccions de disseny, si la placa DE0 actual no serveix buscar-ne una que si serveix, i si no es pot seguir amb les decisions preses, replantejar-les.

Bibliografia web

[1] Alon Amid et al. "Chipyard: Integrated Design, Simulation, and Implementation Framework for Custom SoCs". A: *IEEE Micro* 40.4 (2020), pàg. 10 - 21. ISSN: 1937-4143. DOI: 10.1109/MM.2020.2996616.

[2] Krste Asanović et al. *The Rocket Chip Generator*. Inf. tèc. UCB/EECS-2016-17. EECS Department, University of California, Berkeley, abr. de 2016. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.html.

[3] Debian Community. *RISC-V - Debian Wiki*. 2021. URL: https://wiki.debian.org/RISC-V.

[4] Fedora Community. *Architectures/RISC-V*. 2021. URL: https://fedoraproject.org/wiki/Architectures/RISC-V.

[5] Gentoo Community. *Gentoo RISC-V architecture support project*. 2021. URL: https://wiki.gentoo.org/wiki/Project:RISC-V.

[6] Ubuntu Community. *RISC-V - Ubuntu Wiki*. 2021. URL: https://wiki.ubuntu.com/RISC-V.

[7] Gaisler. *GRLIB IP Library User's Manual*. 2021. URL: https://www.gaisler.com/products/grlib/grlib.pdf.

[8] Gaisler. *NOEL-V Processor*. 2021. URL: https://www.gaisler.com/index.php/products/processors/noel-v.

[9] RISC-V International. *RISC-V Cores and SoC Overview*. 2021. URL: https://github.com/riscvarchive/riscv-cores-list.

[10] UC Berkeley Architecture Research. *RISC-V BOOM / The Berkeley Out-of-Order RISC-V Processor*. 2020. URL: https://boom-core.org/.

[11] F. Zaruba i L. Benini. "The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-Ready 1.7-GHz 64-Bit RISC-V Core in 22-nm FDSOI Technology". A: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.11 (nov. de 2019), pàg. 2629 - 2640. ISSN: 1557-9999. DOI: 10.1109/TVLSI.2019.2926114.