

Implementing the beta machine on a Terasic DE10 SoC + FPGA development board

Auteur : Polet, Quentin

Promoteur(s) : Fontaine, Pascal; Mathy, Laurent

Faculté : Faculté des Sciences appliquées

Diplôme : Master : ingénieur civil électricien, à finalité spécialisée en "electronic systems and devices"

Année académique : 2020-2021

URI/URL : <http://hdl.handle.net/2268.2/11612>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.

Implementing the beta machine on a Terasic DE10 SoC+FPGA development board

Author: Quentin Polet
Section: Electrical Engineering
Promoter: Laurent Mathy, Pascal Fontaine

2020 - 2021

The goal of this work was to design a harvard 32bits CPU, the beta machine, for the laboratories of the Computer Structures [INFO0012] course at the University of Liège. The machine is implemented with arithmetic, shift, conditional branch and jump operations. As it is a Harvard architecture, it has two memories: one for data (65kB) and one for instructions (131kB). In addition to all this, a register file provides 32 registers of 32 bits to the machine. This is more than enough for many applications.

Access to some IOs as well as a graphical accelerator called GPU in the work has been implemented. They allow to add a lot of interaction for the students. The IO access unit is easily accessible through the CPU Load and Store operations and allows the reading and/or writing of three IOs: two push buttons, four switches and eight LEDs. Concerning the GPU, it allows rendering on a 16:9 60Hz screen with a resolution of 848x480 pixels (of which 576x432 can be written). The writing is done using masks that the user can program. The masks allow to write 8x8 pixels per operation with two different colors (it can also be chosen to clear a pixel or to keep its current value). This allows for a more efficient GPU, in fact in the best case it accelerates the writing by 64, which is a great saving in terms of clock cycles.

In addition, a facility to write and read the values in different memories from the ARM processor has been provided. The ARM processor was chosen as the access to the system developed in this work in order to simplify the access. Thanks to this, it is enough to know how to navigate in a Linux terminal and to launch a program to be able to use the system and the beta machine, which is very simple. The whole work was done with simplicity in mind. Finally, several demonstrations were made to show that everything worked as it should.