

Les processeurs embarqués dans les FPGA couplés à Linux

email : kadionik@enseirb.fr
http : <http://www.enseirb.fr/~kadionik>
<http://www.enseirb.fr/cosynux/>

Patrice KADIONIK
IMS – ENSEIRB – Université de Bordeaux 1 - France



FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 1 -

INTRODUCTION



FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 2 -

Introduction

- Cette présentation a pour objectif de préciser comment sont conçus actuellement les circuits numériques complexes avec l'approche système et le système sur silicium SoC (*System on Chip*).
- La méthodologie de conception appelée *codesign* sera précisée. Elle autorise un développement conjoint du matériel (*hardware*) et du logiciel (*software*).
- Les principaux processeurs « logiciels » ou *softcore* seront passés en revue et mettant en avant leur adéquation à exécuter du logiciel libre.
- Un exemple couplant μ Linux à un système SoPC sera donné pour montrer l'intérêt du codesign lors de la conception de systèmes numériques complexes.



FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 3 -

LA CONCEPTION DES SYSTEMES NUMERIQUES COMPLEXES

ims **FPGA, softcore et Linux embarqué** EVERIS

pk@univ-stb/2007 - 4 -

Conception des systèmes numériques

- Les systèmes numériques deviennent aujourd'hui de plus en plus complexes au niveau intégration et fonctionnalités et l'on est en mesure d'intégrer tout dans un même composant.
- C'est le concept du *single chip*.
- Ceci est en fait lié à la loi empirique de Moore qui stipule que pour une surface de silicium donnée, on double le nombre de transistors intégrés tous les 18 mois !

ims **FPGA, softcore et Linux embarqué** EVERIS

pk@univ-stb/2007 - 5 -

Conception des systèmes numériques

- On travaille maintenant au niveau système (ou fonctionnalité) et non au niveau porte logique (pour le grand bien des électroniciens).
- Les fonctionnalités peuvent être implantées dans des composants spécifiques de type ASIC (*Application Specific integrated Circuit*). On parle alors de Système sur Silicium SoC (*System on Chip*).
- Les fonctionnalités peuvent être implantées dans des composants logiques programmables de type FPGA (*Field Programmable Gate Array*). On parle alors de système SoPC (*System on Programmable Chip*).

ims **FPGA, softcore et Linux embarqué** EVERIS

pk@univ-stb/2007 - 6 -

Conception des systèmes numériques

- L'approche « schématique » au niveau porte logique ou fonctionnalités de base RTL (*Register Transfer Logic*) est délaissée pour la conception des systèmes complexes au profit d'une approche « textuelle ».
- On utilise des langages de description de matériel comme VHDL (*Very high speed integrated circuit Hardware Description Language*) ou Verilog pour synthétiser une fonctionnalité numérique.
- Ces langages de description de matériel sont en fait de véritables langages de programmation informatiques, orientés objet. Ils sont utilisés conjointement avec un synthétiseur (compilateur) ou un simulateur.



FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 7 -

Conception des systèmes numériques

- Ces langages ont permis de travailler avec un niveau d'abstraction plus grand laissant les basses besognes au synthétiseur.
- On a pu rapidement développer des bibliothèques de fonctionnalités comme une interface USB, un contrôleur MAC Ethernet que l'on appelle blocs IP (*Intellectual Property*).
- On peut les acheter ou bien utiliser des blocs IP libres (comme du logiciel libre) dont le site phare de référence est <http://www.opencores.org>.



FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 8 -

Conception des systèmes numériques

- Lorsque l'on conçoit un système numérique complexe, on met en œuvre généralement un processeur embarqué.
- Ce processeur embarqué est :
 - Soit un bloc IP : on parle de processeur *softcore*.
 - Soit déjà implanté dans le circuit électronique en « dur » : on parle de processeur *hardcore*. Le processeur de ce type est généralement plus performant que le processeur du type précédent.



FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 9 -

Conception des systèmes numériques

- Le processeur embarqué allie la souplesse du logiciel à l'accélération du temps d'exécution du matériel.
- Une fonctionnalité particulière peut donc être composée d'une partie matérielle couplée à une fonctionnalité logicielle dédiée : on a donc une conception conjointe matérielle-logicielle ou *codesign*.
- Le *codesign* implique donc une conception en même temps du matériel et du logiciel, ce qui est une nouvelle méthodologie par rapport à la méthodologie de conception classique (conception matérielle puis conception logicielle)...



FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 10 -

LES PROCESSEURS POUR LE SOPC



FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 11 -

Les processeurs pour le SoPC

- Le choix d'un processeur pour le SoPC peut se faire sur différents critères :
 - Processeur *hardcore* : pour ses performances au détriment de la flexibilité.
 - Processeur *softcore* : pour sa flexibilité de mise à jour au détriment de performances moindres que le précédent. La portabilité vers n'importe quel circuit FPGA est assurée en étant donc circuit FPGA indépendant. Il est aussi possible de migrer vers un circuit de type ASIC en cas d'une production en grande série.
- Généralement, on privilégie les processeurs *softcore* pour s'affranchir des problèmes d'obsolescence et pour pouvoir bénéficier facilement des évolutions apportées en refaisant une synthèse.



FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 12 -

Les processeurs pour le SoPC

- Le processeur *softcore* peut être libre :
 - Il est décrit en langage de description de matériel (VHDL, Verilog).
 - Le code source peut être librement distribué et implanté dans n'importe quel circuit programmable FPGA.
 - On est alors indépendant du type de circuit FPGA.



FPGA, softcore et Linux embarqué



- 13 -

Les processeurs pour le SoPC

- Le processeur *softcore* peut être propriétaire :
 - Il est distribué par exemple sous forme d'une *netlist* pour être implantée dans un circuit FPGA.
 - Il est généralement lié à un fondeur de circuit FPGA particulier (comme Altera ou Xilinx).
 - On ne peut pas l'utiliser dans un circuit FPGA autre que celui pour lequel il est prévu. On a donc ici une boîte noire.



FPGA, softcore et Linux embarqué



- 14 -

Les processeurs pour le SoPC

- On trouvera principalement au niveau des processeurs *softcore* libres :
 - Le processeur Leon <http://www.gaisler.com/index.html>.
 - Le processeur OpenRisc <http://www.opencores.org/projects.cgi/web/or1k/overview>.
 - Autre processeur : F-CPU <http://www.f-cpu.org>.
 - Autres processeurs : clones de 6800, 68HC11, 68K, PIC : http://www.opencores.org/browse.cgi/filter/category_microprocessor



FPGA, softcore et Linux embarqué



- 15 -

Les processeurs pour le SoPC

- On trouvera principalement au niveau des processeurs *softcore* propriétaires :
 - Le processeur NIOS et NIOS II d'Altera <http://www.altera.com>.
 - Le processeur Microblaze de Xilinx <http://www.xilinx.com>.



FPGA, softcore et Linux embarqué



pk@univ-stb/2007

- 16 -

LES OUTILS LOGICIELS POUR LES PROCESSEURS SOFTCORE



FPGA, softcore et Linux embarqué



pk@univ-stb/2007

- 17 -

Le processeur softcore LEON

LEON : LOGICIELS

- Chaîne d'outils GNU (compilation croisée).
- Simulateur LEON TSIM.
- OS supportés :
 - μ Clinux.
 - Linux.
 - eCos.
 - RTEMS.



FPGA, softcore et Linux embarqué



pk@univ-stb/2007

- 18 -

Le processeur softcore OPENRISC

OPENRISC 1200 : LOGICIELS

- Chaîne d'outils GNU (compilation croisée).
- Simulateur or1Ksim.
- OS supportés :
 - µClinux.
 - Linux.
 - eCos.
 - RTEMS.



FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 19 -

Le processeur softcore MicroBlaze

MicroBlaze : LOGICIELS

- Plateforme de développement Xilinx XPS.
- Chaîne d'outils GNU (compilation croisée).
- Simulateur XMD.
- OS supportés :
 - µClinux. Portage GPL
 - <http://www.itee.uq.edu.au/~jwilliams/mblaze-uclinux/>.
 - microC/OS II. Produit commercial.
 - Noyau ATI. Produit commercial.



FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 20 -

Le processeur softcore NIOS II

NIOS II : LOGICIELS

- Plateforme de développement Quartus II.
- Chaîne d'outils GNU (compilation croisée).
- IDE Eclipse
- Simulateur ModelSim.
- OS supportés :
 - µClinux. Portage GPL
 - Version intégré à l'IDE Eclipse : <http://www.niosforum.com/>
 - Version sous Linux : <http://nioswiki.jot.com/WikiHome>
 - microC/OS II. Produit commercial.
 - Noyau Nucleus. Produit commercial.
 - eCos.





FPGA, softcore et Linux embarqué



pk@enseirb.fr/2007

- 21 -



**EXEMPLE DE MISE EN ŒUVRE LOGICIELLE : PROCESSEUR
SOFTCORE NIOS II SOUS μ CLINUX**

 **FPGA, softcore et Linux embarqué** 

pk@enseiuh/2007 - 22 -

Exemple : NIOS II/ μ Clinux

- Un exemple de *codesign* est donné en mettant en œuvre le processeur *softcore* NIOS II sous μ Clinux (Linux embarqué).
- La carte cible est la carte Altera Stratix 1S10.
- Exemple logiciel : serveur web embarqué.



 **FPGA, softcore et Linux embarqué** 

pk@enseiuh/2007 - 23 -

Exemple : NIOS II/ μ Clinux

IDE SYNTHÈSE

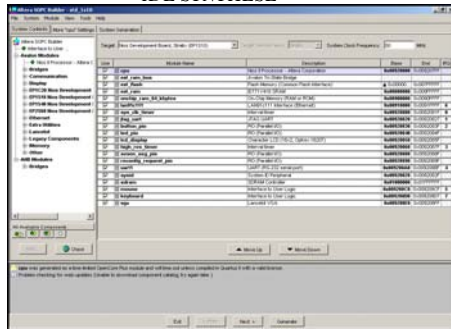
- L'IDE mis en œuvre pour la synthèse matérielle du processeur softcore NIOS II est Quartus II d'Altera.
- On est tributaire des outils de fondeurs de circuits FPGA.

 **FPGA, softcore et Linux embarqué** 

pk@enseiuh/2007 - 24 -

Exemple : NIOS II/μClinux

IDE SYNTHÈSE



FPGA, softcore et Linux embarqué



- 25 -

Exemple : NIOS II/μClinux

CARTE CIBLE

- Circuit FPGA Stratix II EP2S30F672C5.
- 1 Mo de SRAM 16 bits, 16 Mo de SDRAM 32 bits, 16 Mo de mémoire Flash.
- 1 support CompactFlash type I.
- 1 interface Ethernet 10/100 Mb/s.
- 2 ports série (RS-232 DB9).
- 1 connecteur JTAG.
- 4 boutons poussoirs, 8 leds utilisateurs.
- 2 afficheurs 7 segments.
- 1 afficheur LCD 2x16.
- ...



FPGA, softcore et Linux embarqué



- 26 -

Exemple : NIOS II/μClinux

DEVELOPPEMENT LOGICIEL

- On utilise l'environnement de développement Eclipse.
- Le portage de μClinux est disponible pour NIOS II et est intégré à Eclipse.
- Les principales étapes sont :
 - Etape 1 : design de référence
 - Etape 2 : génération du noyau μClinux
 - Etape 3 : construction du système de fichiers root
 - Etape 4 : téléchargement et boot



FPGA, softcore et Linux embarqué



- 27 -

CONCLUSION

ims **FPGA, softcore et Linux embarqué** EVERIS

pk@enscm.fr/2007 - 31 -

Conclusion

- La conception des systèmes numériques complexes nécessite maintenant de mettre en oeuvre une nouvelle méthodologie de conception : le *codesign*.
- L'intégration grandissante sur le silicium a permis d'avoir une approche de conception orientée système et l'on développe maintenant des systèmes sur silicium SoC et SoPC.
- Les systèmes SoC et SoPC intègrent généralement un processeur embarqué si bien qu'il semble naturel d'embarquer Linux dans ces systèmes !

ims **FPGA, softcore et Linux embarqué** EVERIS

pk@enscm.fr/2007 - 32 -

Conclusion

- Le *codesign* apparaît dans le développement conjoint matériel-logiciel pour la conception des systèmes numériques complexes :
 - On développera une partie de l'application sous forme d'un bloc IP (en VHDL par exemple) pour bénéficier d'une accélération matérielle du traitement. Cela revient en fait à développer un coprocesseur matériel spécifique.
 - On développera l'autre partie de l'application sous forme logicielle pour bénéficier de la souplesse de la « logique programmée » exécutée par le processeur softcore embarqué dans le système cible.

ims **FPGA, softcore et Linux embarqué** EVERIS

pk@enscm.fr/2007 - 33 -
