

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ**

**ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΠΑΡΟΥΣΙΑΣΗ / ΕΞΕΤΑΣΗ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

**Γιαουρτάς Μιχαήλ  
Μεταπτυχιακός Φοιτητής**

**Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης  
Επόπτης Μεταπτ. Εργασίας: Καθηγητής, Μ. Κατεβαίνης  
Β. Παπαευσταθίου, Ερευνητής Γ` (επιβλέπων)**

**Τρίτη, 25/02/2020, 16:00**

**Αίθουσα Β106, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης**

**“Εξερεύνηση Αρχιτεκτονικών Προγραμματιζόμενης Λογικής για Αποδοτική  
Επιτάχυνση Εφαρμογών Υψηλών Επιδόσεων ”**

#### **ΠΕΡΙΛΗΨΗ**

Οι Διατάξεις Προγραμματιζόμενης Λογικής (Field Programmable Gate Arrays - FPGAs) έχουν αυξανόμενο αντίκτυπο σε όλο και περισσότερες εφαρμογές, από τα νευρωνικά δίκτυα (Deep Neural Networks) έως επεκτάσεις του ρεπερτορίου εντολών (Instruction Set Extensions) σε στενώς συνεζευγμένα συστήματα με ενσωματωμένες FPGAs (eFPGAs). Καθώς οι εφαρμογές αποκλίνουν στην πολυπλοκότητα, τις επιδόσεις, τις ανάγκες μνήμης και τους περιορισμούς σε έκταση, υπάρχει ανάγκη για ένα ευρύτερο φάσμα αρχιτεκτονικών FPGA. Ωστόσο, η ανάπτυξη και υλοποίηση των αρχιτεκτονικών αυτών παραμένει δύσκολη και απαιτεί πολύ χρόνο, λόγω των υψηλών απαιτήσεών τους σε ειδικευμένες σχεδιάσεις (custom layout) και της ανάγκης ανάπτυξης λογισμικού

προσαρμοσμένου για τον προγραμματισμό κάθε αρχιτεκτονικής, οδηγώντας στην παραγωγή προϊόντων πιο γενικού σκοπού.

Πολλές ακαδημαϊκές εργασίες επικεντρώνονται στην αυτοματοποιημένη διαδικασία παραγωγής αρχιτεκτονικών FPGA, σε μια προσπάθεια να προωθηθεί η εξατομίκευση και να μειωθεί η χρονική περίοδος μέχρι την αγορά. Σε άλλες προσεγγίσεις, οι ερευνητές στοχεύουν στη διαδικασία εξερεύνησης, στην οποία αναζητούν τη βέλτιστη αρχιτεκτονική για ένα συγκεκριμένο σενάριο, χρησιμοποιώντας μοντέλα εκτιμήσεων μεγέθους και καθυστέρησης.

Στην εργασία αυτή επιλέγουμε να συνδυάσουμε τις δύο αυτές προσεγγίσεις. Αναπτύσσουμε μια επέκταση για το δημοφιλές εργαλείο ανοιχτού κώδικα Verilog-to-Routing (VTR) προκειμένου να εξάγουμε σε Verilog την αναπαράσταση των αρχιτεκτονικών FPGA που έχουν οριστεί από τον χρήστη, να υποστηρίξουμε ζητούμενες μονάδες ειδικού σκοπού (hard blocks - RAMs, DSPs, FP Units) και να παράξουμε αρχεία προγραμματισμού της FPGA (Bitstreams) για δοθέντα benchmarks. Στόχος μας είναι να δημιουργήσουμε συνθέσιμο κώδικα RTL ανεξάρτητο από τεχνολογία, ικανό να συντεθεί με οποιαδήποτε βιβλιοθήκη Standard Cells. Ανακαλύπτουμε τις πραγματικές σχεδιαστικές ιδιότητες μιας αρχιτεκτονικής FPGA χρησιμοποιώντας μια προτεινόμενη ροή σχεδιασμού υλικού (ASIC flow) και ανακτούμε πραγματικές μετρήσεις μεγέθους και καθυστέρησης και τελικά προχωρούμε στην εξερεύνηση των βέλτιστων αρχιτεκτονικών FPGA για συγκεκριμένα σύνολα από benchmarks.

Χρησιμοποιώντας την επέκτασή μας, εξερευνούμε το χώρο σχεδιασμού των FPGA για ένα σύνολο από benchmarks υψηλών επιδόσεων που εξάγονται από την πλατφόρμα High Level Synthesis (HLS) της Xilinx. Η εξερεύνηση μας αρχίζει με τον εντοπισμό των βέλτιστων αρχιτεκτονικών FPGA, ξεκινώντας από το μέγεθος των προγραμματιζόμενων πυλών (Lookup Tables - LUTs) και τον αριθμό τους ανά ομάδα (Configurable Logic Block - CLB) και έπειτα εξετάζοντας το μέγεθος και μήκος των καναλιών διασύνδεσης τους. Συγκρίνουμε επίσης τις βέλτιστες αρχιτεκτονικές FPGA που προκύπτουν κατά τη χρήση των benchmarks υψηλών επιδόσεων με τις αντίστοιχες αρχιτεκτονικές που προκύπτουν όταν χρησιμοποιούμε τα MCNC benchmarks γενικού σκοπού.

Τέλος, δημιουργούμε σύνολα εντολών TCL για τη σύνθεση και την υλοποίηση της τοποθέτησης και διασύνδεσης (place and route) που μπορούν να προσαρμοστούν σε οποιοδήποτε μέγεθος και χαρακτηριστικό αρχιτεκτονικής και να αυτοματοποιήσουν τη ροή ASIC για νέα τσιπ FPGA.

**Giaourtas Mixail**

**M.Sc. Thesis**

**Computer Science Department**

**University of Crete**

**Master's Thesis Supervisor: Professor, M. Katevenis**

**V. Papaefstathiou, (Thesis Co- Advisor)**

**Tuesday, 25/02/2020, 16:00**

**Room B106, Computer Science Dept., University of Crete**

**“Design-space Exploration of FPGA Architectures for Efficient HPC Acceleration”**

### **ABSTRACT**

Field Programmable Gate Arrays (FPGAs) have an ever-expanding impact to more and more applications, ranging from Deep Neural Networks to High-Performance Computing (HPC) and other uses such as customization of Instruction Set Extensions and computation offloading in systems with tightly coupled embedded FPGAs (eFPGAs). As applications diverge in complexity, performance, memory needs and area limitations, there is a need for a wider variety of FPGA architectures. However, developing and implementing new FPGA architectures remains challenging and requires a lot of time, due to their high content in custom layout designs and the need for design software and flows tailored for each specific architecture, leading to the production of more generic products.

Many academic works are focusing on the automated FPGA design generation process, in an attempt to promote customizability and reduce time-to-market. In other approaches, researchers target only the exploration process, in which they seek for the optimal architecture for a specific case scenario, using area and delay estimation models.

In this thesis we choose to combine the two approaches. We develop an extension for the popular open-source tool Verilog-to-Routing (VTR) in order to export in Verilog the representation of user-specified FPGA architectures, develop support for custom user

hard-blocks (RAMs, DSPs, FP Units), and generate Bitstreams for given benchmarks. Our objective is to create synthesizable and technology independent RTL design code, able to be synthesized with any standard cell library. We discover the real design properties of an FPGA architecture using our proposed ASIC flow and retrieve real area and delay measurements and eventually proceed with the exploration of optimal FPGA architectures for given sets of benchmarks.

Using our VTR extension, we perform FPGA design-space exploration for a set of HPC oriented benchmarks that are derived using Xilinx's High Level Synthesis (HLS). Our exploration starts by identifying pareto-optimal FPGA architectures starting with the size of Lookup Tables (LUTs) and the number of LUTs per Configurable Logic Block (CLB) and then explore the size of routing channels and wire segments' configurations. We also compare the optimal FPGA architectures derived when using the HPC benchmarks with the respective architectures derived when we use the generic MCNC benchmarks.

Finally, we create TCL scripts for synthesis and back-end implementation (place and route) which can adjust to any architectural characteristic and size and automate the ASIC flow for new FPGA chips.