

Ανάπτυξη γεννήτριας κυκλωμάτων
υλοποίησης αριθμητικών πράξεων σε
γλώσσα περιγραφής υλικού

Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Ιωάννης Πετροσόβ

Επιβλέπων Καθηγητής: Μηνάς Δασυγένης

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών
<http://arch.ict.e.uowm.gr>

Νοέμβριος 2016

Πίνακας περιεχομένων

Εισαγωγή

Θεωρητικό υπόβαθρο

Η υλοποίηση του λογισμικού μέρους

Frontend

Backend

Πειραματικές μετρήσεις

Συμπεράσματα και Μελλοντικές βελτιώσεις

Επίδειξη λειτουργίας

Εισαγωγή

Ορισμός προβλήματος

- ▶ Η σημαντικότητα του παράγοντα της κατανάλωσης ενέργειας
- ▶ Αυξανόμενη πολυπλοκότητα του λογισμικού και του υλικού
- ▶ Το 80% των τσιπ δεν μπορεί να ικανοποιήσει τις αρχικές προθεσμίες διάθεσης στην αγορά
- ▶ Χρονοβόρα διαδικασία: εξερεύνηση χώρου (Digital Space Exploration-DSE)

Η πρότασή μας

- ▶ Δημιουργία μίας γεννήτριας επιταχυντών υλικού προσαρμοσμένων αριθμητικών κυκλωμάτων
- ▶ Τα κυκλώματα παράγονται σε VHDL
- ▶ Αλάνθαστες περιγραφές (σύνταξη, λειτουργία)
- ▶ Υποστήριξη ανεξάρτητου αριθμού μεταβλητών, σταθερών και εύρους bit

Περιπτώσεις παρόμοιων εργαλείων

Παρόμοια εργαλεία

- ▶ Altera
- ▶ Xilinx
- ▶ Mathworks
- ▶ SPIRAL online tool
- ▶ FloPoCo

Μειονεκτήματα

- ▶ Ακριβό κόστος
- ▶ Τοπική εγκατάσταση
- ▶ Ιδιόκτητος (proprietary) κώδικας και netlist
- ▶ Γνώση DSL ή γλώσσας προγραμματισμού
- ▶ Δεν παράγονται προσαρμοσμένες συναρτήσεις

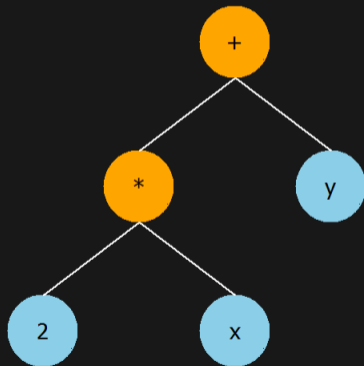
Θεωρητικό υπόβαθρο

Χρησιμοποιούμε δέντρα έκφρασης

Αλγόριθμος

```
Result: expression tree
create_empty_stack();
current_node = create_root_node();
for character in expression do
if character == ( then
// case 1
insert_node_left();
push_current_node_to_stack();
current_node = get_left_node();
else if character not in [+ , - , * , / , )] then
// case 2
current_node_value = character;
current_node = stack:pop();
else if character in [+ , - , * , /] then
// case 3
current_node_value = character;
insert_node_right();
push_current_node_to_stack();
current_node = get_right_node();
else if character == ) then
// case 4
current_node = stack:pop();
```

$$f(x, y) = ((2 * x) + y)$$



Τεχνολογική στοίβα του εργαλείου μας

Frontend

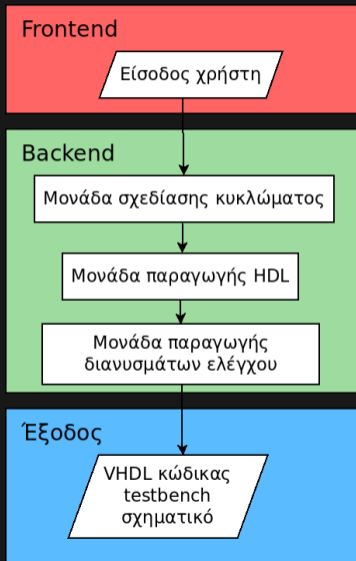
- ▶ HTML
- ▶ CSS
- ▶ Javascript
- ▶ PHP

Backend

- ▶ Python
- ▶ Cython
- ▶ PuDB
- ▶ Virtualenv
- ▶ Ipython
- ▶ GHDL
- ▶ Xilinx ISE
- ▶ Git

Η υλοποίηση του λογισμικού
μέρους

Γενική ροή εργαλείου μας



Frontend

Αρχείο εισόδου

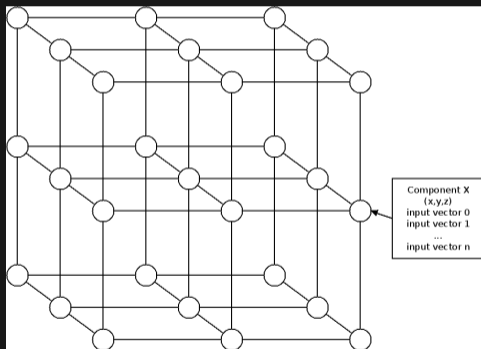
- ▶ JSON μορφή
- ▶ Δημιουργείται από τον χρήστη ή υπολογιστικά
- ▶ Δηλώνεται η συνάρτηση και τα εύρη bit των μεταβλητών
- ▶ Επικοινωνία frontend-backend

```
{  
  "function": "( ( in0 + in1 ) * ( in2 + 73 ) )",  
  "in0": 2,  
  "in1": 2,  
  "in2": 5  
}
```

Backend

α -HDL netlist

netlist



Ιδιότητες

- ▶ Τρισδιάστατη μορφή
- ▶ Αποτελείται από 3 δομές
 - components
 - interconnections
 - componentlist
- ▶ Δημιουργείται αυτόματα από τη μονάδα σχεδίασης
- ▶ Επικοινωνία μεταξύ μονάδων στο backend

Μονάδα σχεδίασης κυκλώματος

- ▶ Δημιουργεί την α-HDL netlist
- ▶ Αποτελείται από 4 βήματα

1. Δημιούργησε το δέντρο έκφρασης



2. Διόρθωσε το εύρος bit των κόμβων



3. Εκτέλεσε αντίστροφη διάσχιση



4. Τοποθέτησε τα στοιχεία

Μονάδα παραγωγής HDL



Μονάδα παραγωγής διανυσμάτων ελέγχου

Παράγει τα διανύσματα ελέγχου που επιβεβαιώνουν την ορθότητα λειτουργίας του κυκλώματος του επιταχυντή

1. Δημιουργεί τυχαίο αριθμό για κάθε μεταβλητή
2. Οδηγεί τον αριθμό στη σωστή θύρα εισόδου
3. Υπολογίζει το σωστό αποτέλεσμα
4. Συγκρίνει το σωστό αποτέλεσμα με την έξοδο του κυκλώματος

Παράδειγμα testbench της συνάρτησης $f(x, y) = x + y$

```
signal1<="10001" AFTER 4 ns ;-- input vector: 17^M
signal0<="001" AFTER 4 ns ;-- input vector: 1^M
wait for waittime * 1 ns ;^M
-- output: 18^M
assert (vec2int(signal2) = 18 ) report "TESTBENCH3-N Output:"&
integer'image(vec2int(signal2))&" Expected:"&integer'image(18) severity error ;
assert (vec2int(signal2) /= 18 ) report "TESTBENCH3-N OK" severity note ;
```

Εξοδος γεννήτριας

- ▶ Περιγραφή κυκλώματος σε VHDL
- ▶ Βιβλιοθήκη VHDL με τα υποκυκλώματα

Προαιρετικά:

- ▶ Σχηματικό κυκλώματος (PNG και dot)
- ▶ Αρχείο επιβεβαίωσης ορθότητας λειτουργίας (testbench)

Πειραματικές μετρήσεις επιταχυντών

Πίνακας : Πειραματικές μετρήσεις συνάρτησης $f(x,y) = ((2 * x) + y)$.

in0 bits	in1 bits	Αριθμός τρανζίστορ	Αριθμός τομών	Κρίσιμο μονοπάτι (ns)	Συχνότητα (MHz)	Ενέργεια (W)
2	2	70	1	1.060	943,39	3.422
4	4	126	3	1.690	591.71	3.422
8	8	238	5	2.409	415.11	3.422
16	16	462	9	4.701	212.72	3.422
24	24	686	14	6.993	143	3.422
12	22	602	13	5.218	191.644	3.422

Πειραματικές μετρήσεις γεννήτριας

Πίνακας : Χρόνοι παραγωγής $f(x,y)=((2*x)+y)$.

$f=((2*x)+y)$		Χρόνος (seconds) Σχηματικό	
in0 bits	in1 bits	Ναι	Όχι
2	2	0.632	0.212
4	4	0.904	0.313
8	8	1.651	0.212
16	16	3.440	0.252
24	24	7.767	0.293
12	22	5.764	0.280

Συμπεράσματα

- ▶ Προσθέσαμε ακόμα ένα εργαλείο στη φαρέτρα των μηχανικών και σχεδιαστών κυκλωμάτων
- ▶ Το πρόγραμμα μας επιτρέπει την εύκολη επέκταση με νέες λειτουργίες
- ▶ Εξοικονόμηση χρόνου στην εξερεύνηση χώρου (DSE)
- ▶ Χρήση της Python για προτυποποίηση κυκλωμάτων
- ▶ Η παραγωγή κυκλωμάτων, απαιτεί πολύπλοκα εργαλεία σύνθεσης υψηλού επιπέδου
- ▶ Προώθησης ενδιαφέροντος προς άλλους φοιτητές

Μελλοντικές επεκτάσεις

- ▶ Προσθήκη περισσότερων αριθμητικών πυρήνων
- ▶ Υποστήριξη αριθμών κινητής υποδιαστολής
- ▶ Προσθήκη επιλογής για διασωλήνωση
- ▶ Επιτάχυνση του κώδικα της γεννήτριας (Cython, PyPy)
- ▶ Μετάβαση από Python 2 σε Python 3
- ▶ Συγγραφή πλήρους οδηγού ανάπτυξης
- ▶ Μεταφορά του αποθετηρίου (repository) στο cloud
- ▶ Δημιουργία REST API

Αναγνώριση ερευνητικού έργου



Minas Dasygenis and Giannis Petrousov.

A generic moduli selection algorithm for the residue number system.

In 10th International Conference on Design & Technology of Integrated Systems in Nanoscale Era, DTIS 2015, Napoli, Italy, April 21-23, 2015.



Minas Dasygenis and Ioannis Petrousov.

A networking EDA tool for multi-vector multiplication IP circuits.

In 10th International Conference on Design & Technology of Integrated Systems in Nanoscale Era, DTIS 2015, Napoli, Italy, April 21-23, 2015.



Ioannis Petrousov Minas Dasygenis.

Online generation of constant multiplication accelerators.

In Panhellenic Conference on Electronics and Telecommunications, PACET, 2015.



Giannis Petrousov and Minas Dasygenis.

Designing optimized forward residue number systems IP blocks converters from a network interface.

In 18th Panhellenic Conference on Informatics, PCI '14, Athens, Greece, October 2-4, 2014.



Giannis Petrousov and Minas Dasygenis.

A unique network EDA tool to create optimized ad hoc binary to residue number system converters.

In 24th International Workshop on Power and Timing Modeling, Optimization and Simulation, (PATMOS), Palma de Mallorca, Spain, September 29 - Oct. 1, 2014.



Ioannis Petrousov and Minas Dasygenis.

A CAD tool for custom magnitude comparators.

In Proceedings of the 19th Panhellenic Conference on Informatics, PCI 2015, Athens, Greece, October 1-3, 2015.



Ioannis Petrousov and Minas Dasygenis.

Generating custom bitwidth 4:2 compressors in hardware description language from an online tool.

In 4th International conference on Modern Circuits and System Technologies, 2015.

Επίδειξη λειτουργίας

http://arch.icte.uowm.gr/hdl/equationparser_json.php

Ευχαριστώ
για την προσοχή σας

petrousov@gmail.com
<http://petrousov.net/>

http://arch.icte.uowm.gr/hdl/equationparser_json.php