# FripGa: A Prototypical Design Tool for Embedded Multi-Core Systems-on-Chip



TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Computer Science Integrated Circuits and Systems Lab

# **Alexander Biedermann and Sorin A. Huss**

biedermann@iss.tu-darmstadt.de, huss@iss.tu-darmstadt.de

## Challenges

- Today's FPGAs offer plenty of logic resources
- Complex Systems-on-Chip may consist of several dozens of processors and HW IP Cores
- Design tools often still tailored to "single processor with (HW) co-processors" designs
- How to enable smooth, fast, and safe creation of complex Systems-on-Chip?
- How to avoid repetitive, error-prone user inputs?
- How to visualize complex Systems-on-Chip?
- Develop prototypical design tool with usability features which future commercial tool suites should provide soon
- How to support easy design space exploration?
- How to include methodologies that abstract from underlying architectures?

# FripGa: Key Features

Idea



#### Multi Chip Designs

- Scatter designs over multiple FPGAs
- Automatic mapping of modules to FPGAs based on user constraints (max. number of FPGAs, max communication speed, or minimal resource consumption)

File Scene Look & Feel Language Info	
🗋 🔐 🕼 🔗 😰 🝙 🧟 鞲 🔕 🍰 🖉 🖉 🤶 👬 👬	
Abstract Process	Connection Templates



#### **User Interface**

- Build HW/SW designs from scratch
- Instantiate, duplicate, connect, and configure embedded processors and IP cores
- TCL scripting engine for all features

File Scene Look & Feel Language Info				



#### **Partial Reconfiguration**

- Avoids manual floor planning
- Automatic instantiation of partial reconfigurable regions on FPGA chip area
- Automatic resource analysis to determine sizes of partial reconfigurable regions

File	Scene	Look & Feel	Language	Info						
			2	Q #	o e	p p p	12 5 8 8 5 2 1			
-									Y	

(1)	microblaze_1	microblaze_2	Default Microblaze Elements
Block Ram			Software Projects
/	S" XILINK	ST XILINX:	No software project
ork			muSoftwareProject
	MicroBlaze	MicroSlaze	inysonwarenoject
	- III		
PGA	Software Project: muS	oftwareProject	
XILINX®	Software Project. Higs		
ToUart	Properties	1 /" resample.c "/	Ê
	Processors	3 #include <mvheader.b></mvheader.b>	
	Sources	4 #include <fsl.h></fsl.h>	
in 🚽	src/mySourceFile	5 #include "filter_params.	h" ⊨
	- D Headers	6 #include "file_params.h"	
	src/myHeader.h	7	
AB Controller		8	
٠		10 int resamplePoint(int sa	mpleO, int sample1, int sample2.
icroblaze		11 {	
		12 int resPoint;	
		• 13	
icroblaze Block	2) Scr	14 resPoint = sampleO	* coeffTab[phase][0] + sample1 *
۵.		15	
\$		$\frac{10}{17}$ if (resPoint < 0)	
oC_Process \$		18 resPoint = 0:	
		19	
bus \$		20 if (resPoint > 255)	
.0.		21 resPoint = 255	;
•		22	
econfigurable E 🔻			
A 19.4 4		😋 Add 🛛 🚫 Remove 🔄 🛃 Read	Load file

#### Handling of Embedded Software Projects

- Import and edit legacy software projects for embedded processors
- Create and edit new software projects
- Easy binding of SW projects to processors



#### Switching between Software and Hardware Implementations

- Allows for smooth and fast design space exploration
- Guided by a wizard to match interface connections



Editor to integrate Models of Computation (MoC)

- MoC processes may be partitioned to either SW or HW by the designer
- MoC processes may be connected to common modules such as embedded processors or HW IP cores

Additional Features	Comparison of Workflow Efficiency					
<ul> <li>Visual representation, editable at any time</li> </ul>	Xilinx XPS FripGa					
<ul> <li>XML file structure for efficient file handling</li> </ul>	<b>Common operations</b> MC <sup>1</sup> Time MC <sup>1</sup> Time					
<ul> <li>Wizard to import legacy IP cores or to create new ones</li> </ul>	Add embedded processor to design47180 s11 s					
- Grouping of modules to functional units to use them as hierarchical design primitives	Establish module-to-module connection 12 28 s 1 1 s					
– Full compatibility to existing commercial tool suites, e.g. Xilinx Platform Studio	Duplicate group of modules n.a. n.a. 2 6 s					
<ul> <li>Built-In IP cores for data path manipulation</li> </ul>	Add partial reconfigurable modules >100 >1800 s 9 25 s					
	$^{1}$ MC = mouse clicks					

# **Application Example**

- Parallelized FIR filter for image scaling designed in FripGa
- Sub-pixel resampling done either in hardware or in software
- Design with up to 18 soft-core processors created in less than one hour
- Easy switching between different implementations or levels of parallelization



## **Future Work**

- Expand functionality of MoC editor, e.g., by exploiting known MoC transformations from system level to component level
- Exploit virtualization schemes to further abstract from strict software-processor bindings and automatically choose optimal number of processors employed in design (see poster "Scalable Multi-Core Virtualization for Embedded System-on-Chip Architectures" on DATE'12 Friday Workshop: Quo Vadis Virtual Platforms?)

# Dipl.-Inform. Alexander Biedermann Technische Universität Darmstadt Integrated Circuits and Systems Lab Hochschulstraße 10 64289 Darmstadt, Germany

#### Prof. Dr.-Ing. Sorin A. Huss

Technische Universität Darmstadt Integrated Circuits and Systems Lab Hochschulstraße 10 64289 Darmstadt, Germany