GRIP: Graph-Rewriting Based IP-Integration Tool

An EDA Tool For Software-Defined SoC

IP-XACT IEEE-1685
Eclipse Epsilon and EMF
Java Freemarker

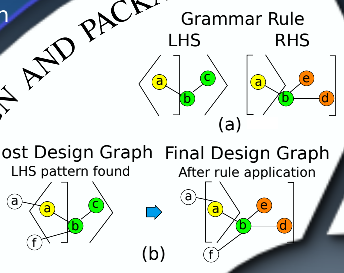
IP-SUPPLIER

- * IP-supplier knows the most about the IP and its integration
- * IP-supplier to encode HW IP knowledge in IP-XACT and IP-integration knowledge via GRIP integration rules described in IP-XACT

Graph-Grammar

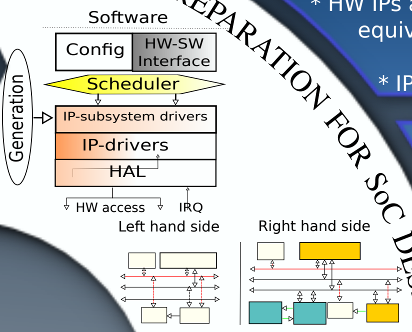
- * Grammar-rules to encode IP-integration process
- * Mathematical modeling for exchanging IP-knowledge

1. IP-DESIGN AND PACKAGING



* Graph transformation process,
MATCH >> CHOOSE >> APPLY

2. IP-LIBRARY PREPARATION FOR SoC DESIGN



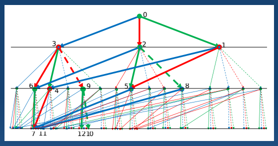
IP-SUPPLIER

- * IPs and IP-subsystems integration rules are compiled in the IP-library
- * HW IPs are mapped to their equivalent SW methods
- * IP-XACT Library validation

Drivers Generation

- * SW-drivers hierarchy,
  - > HW-SW Interface
  - > IP-subsystem drivers
  - > IP-drivers (HAL)

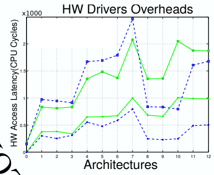
SoC ARCHITECT



CONCLUSIONS

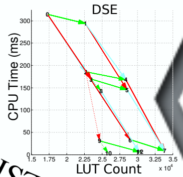
* GRIP tool enables,
= Automated IP-integration for SoCs
= SoC HW code for the targeted platform
= SW drivers generation for SW application
= Validation of IP-XACT design & IP-library

4. SoC DESIGN SPACE EXPLORATION

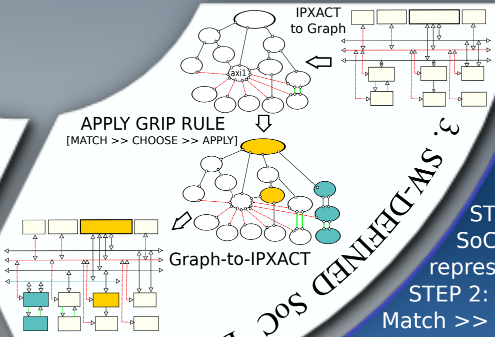


0.28% drivers overhead

Video-processing
Case Study:
Generated
12 SoCs



3. SW-DEFINED SoC DESIGN



SoC ARCHITECT

- * GRIP tool automates IP-integration utilizing the prepared IP-library

STEP 1: Transform IP-XACT SoC design to Graph representation
STEP 2: Graph transformation, Match >> Choose >> Apply
STEP 3: Transform new graph to IP-XACT
STEP 4: Code generation from IP-XACT to target platform

***Abstract*—The GRIP tool - Graph-Rewriting-Based IP-Integration - provides system engineers with a comprehensive platform that takes care of their IP-integration concerns for IP-centric SoC designs, also referred to as SW-defined SoCs. The tool uses the standardized meta-data IP-XACT format for electronic system descriptions and encodes the design IP-integration knowledge as a set of integration rules based on graph rewriting and graph grammar theory. The tool automates and encodes the step-by-step integration of IPs to build a desired system architecture. Multiple sequential IP-integration steps can be compiled to iteratively generate new system architectures. For design space exploration (DSE) constraints can be given to generate a desired subset of candidate SoC architectures. Code generation is used to generate the design files for each architecture. This is demonstrated for integrating HW acceleration into an OpenCV computer vision application running on a Xilinx Zynq chipset based Zedboard. GRIP additionally generates the HW-drivers for both bare-metal and Linux-based systems [1] [2] [3].**

1. IP-DESIGN AND PACKAGING - One major challenge in the IP-centric design is IP-supplier's lacking knowledge of targeted SoCs and SoC architect's lacking knowledge of the supplied hardware IPs. Using the principles of graph-grammar and graph-rewriting, GRIP provides a way to encode IP-integration knowledge by grammar-rules, namely GRIP integration rules. An IP supplier can write these rules in IP-XACT, and GRIP internally transform those to graphs [1]. GRIP generates SW drivers using from the rules for IP-packaging[3].

2. IP-LIBRARY PREPARATION FOR SoCs - An SoC architect assimilate the supplied IP-packages to prepare an IP-library. Using GRIP, an SoC architect can customize the library IPs, rules, and SW drivers. He/She can also generate the drivers for custom IP-subsystems, using IPs from various vendors. GRIP supports validation of IP-XACT IP-library and integration rules.

3. SW-DEFINED SoC DESIGN - In the GRIP platform, using the IP-library, an SoC architect can cumulatively apply the rules to design desired SoC. He/She can utilize generated drivers to HW accelerate SW methods of the SW application. An IP-integration in GRIP takes four steps, as explained in the figure above. GRIP validates each generated SoC.

4. SoC DESIGN SPACE EXPLORATION (DSE) - While performing DSE for a SW application, an architect can opt for incrementally adding new HW IPs, in the step-by-step exploration mode. Otherwise, an architect can constrain the design space by architecture constraints, and GRIP follows these constraints to generate candidate SoC architectures.

A case study was executed to perform DSE for a video-processing application on the Xilinx Zynq FPGA (ZedBoard). The IP-library had 3 HW accelerators and 8 GRIP rules. With DSE constraints, GRIP generated 12 candidate architectures [1].

References

[1] M. Jassi, D. Mueller-Gritschneder, and U. Schlichtmann. GRIP: Grammar-Based IP Integration and Packaging for Acceleration-Rich SoC Designs. *Design Automation Conference (DAC)*, June 2015.

[2] M. Jassi, B. Bordes, D. Mueller-Gritschneder, and U. Schlichtmann. Automation of FPGA Performance Monitoring and Debugging Using IP-XACT and Graph-Grammars. *Internation Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, Sept 2015.

[3] M. Jassi, U. Sharif, D. Mueller-Gritschneder, and U. Schlichtmann. Hardware-Accelerated Software Libraries Drivers Generation for IP-Centric SoC Designs. *Great Lake Symp. VLSI (GLSVLSI)*, 2016 (SUBMITTED).

Munish Jassi, Yong Hu, Jian Lyu, Daniel Müller-Gritschneder, Ulf Schlichtmann
{munish.jassi, yong.hu, jian.lyu, daniel.mueller, ulf.schlichtmann}@tum.de
Institute for Electronic Design Automation, Technische Universität München