Implementation of Scalable *e*mbedded FPGA for SoC

Hayder Mrabet, Zied Marrakchi and Habib Mehrez
Université Paris VI, Laboratoire LIP6-ASIM
4 Place Jussieu, 75252 Paris, France

André Tissot
CEA-DAM IDF
Commissariat à l'Energie Atomique

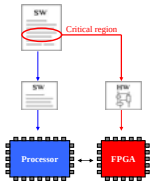
Abstract

- We propose an SRAM-based eFPGA architecture
- We present the configuration flow and the hardware design flow
- An example of implementation of eFPGA as a VCI IP

1. Introduction

Hardware/Software Partitioning

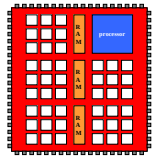
Achieves better results than software optimizations:
- Large speedup
- High performance



1. Introduction

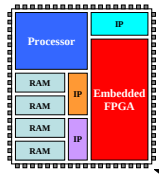
SoPC (System on Programmable Chip):
- Hard silicon cores embedded into Programmable architecture
- Limited to prototyping and applications requiring low volume production.
  - High power consumption
  - Lower performance compared to the standard cell ASIC

Examples: Excalibur ARM-based of Altera
Xilinx PowerPC405-based



***e*FPGA in SoC . ASIC implementation technology:**
- More efficient communication
- Smaller size
- Higher performance
- Low power
- Adds flexibility and suppleness
- Post-fabrication design changes

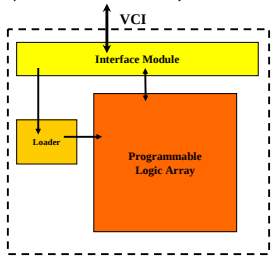
Examples: FlexASIC 0.13 Core of eASIC
M2000 FLEXOStm Configurable IP Core



2. *e*mbedded FPGA core

- Scalable Programmable Logic Array, in Island Style with fine granularity
- A loader to configure the SRAM-based *e*FPGA
- VCI (Virtual Component Interface Module) Interface Module

Model exists on:
- SystemC
- Synthesizable VHDL



3. SRAM-based Programmable Logic Array

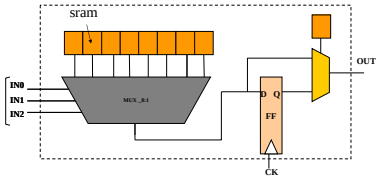
Configurable Logic Bloc CLB

-4-input Look-Up-Table
- One Flip-Flop

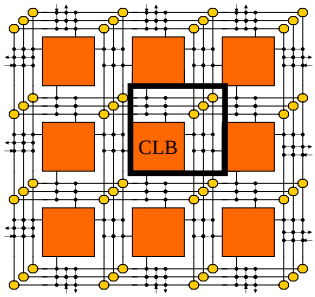
=> Can Implement any function of 4 inputs

Basic Tile consists on:

- One CLB
- Horizontal and vertical connection Boxes (programmable multiplexer+ bidirectional routing channels)
- One switch box (disjoint topology)



Programmable component



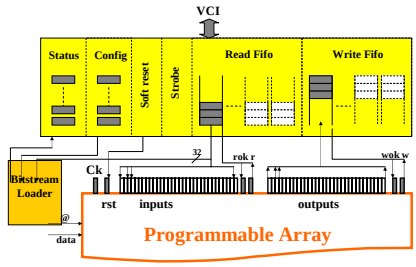
→**Architecture**
Island style

→**eFPGA size**
NX & NY

→**In/out flexibility**
In/out ratio

4. Interface Module PLA/VCI

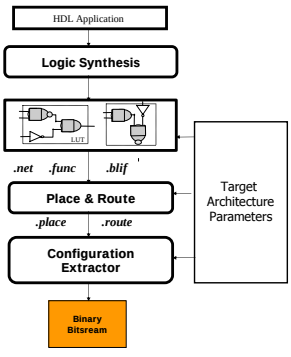
Generic Target Multi-FIFO

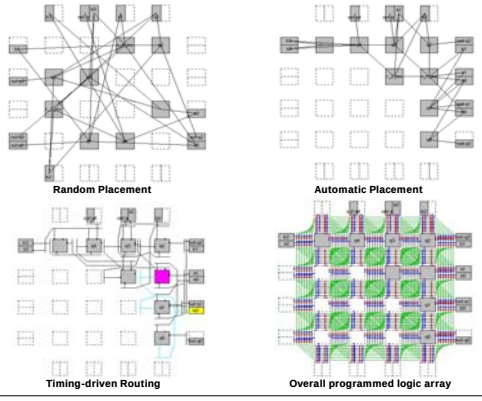


5. Configuration Flow

Tools:

Boog @lliance (Lip6)
Sis (Univ Berkley)
VPR (Univ of Toronto)
Generic_extractor (Lip6)

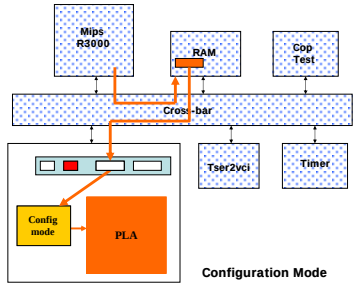




6. AOC (Asim On Chip) : MIPS Processor Based SoC

AOC include *e*FPGA as a hardware Accelerator

- Configuration mode driven by the Mips
- Functional mode:
  Computation-intensive tasks are mapped on the eFPGA



7. Hardware design flow

SystemC Model
Bottom-Up technique
CLB – SwitchBox – Base Tile
-> Programmable Logic Array
Loader
Interface Module

Simulation

VHDL Model
- Structural VHDL Model for the PLA
- VHDL state machine for the Loader
- generic VHDL Model for the Interface Module

Co-Simulation
Co-Simulation VHDL-SystemC with ModelSim:
- VHDL Model of the entire eFPGA
- AOC(MIPS+CrossBar+RAM+...) SystemC

Logic Synthesis
- Loader
- Interface module

using Synopsys and sxlib target library

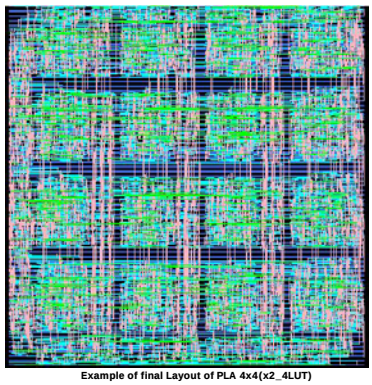
Co-Simulation
*e*FPGA implemented on Stratix FPGA(ALTERA)
Co-simulation Hardware/Software with ModelSim

8. Programmable Logic Array: final Layout

Automatic Placement thanks To PLA_generator, then a Global Routing with 4 layers of Metal

Matrix size CLB = 2x4LUT	Logic Capacity (gate)	Measure in Symbolic (2x2)	Area in CMOS 0.13µ (mm^2)
4x2	192	3770x2000	0.109
2x6	288	1972x3800	0.159
4X4	384	3770x3800	0.216
4x8	768	3770x7400	0.402
8x8	1536	7370x7400	0.786

Example of different sizes of PLAs



Example of final Layout of PLA 4x4(x2_4LUT)

9. Conclusion

- *e*FPGA :
  from architecture parameters to SystemC model then synthesizable VHDL model.
- Complete embedded FPGA Design Flow:
  - Configuration Flow
  - Hardware Flow
- Realizations:
  - Flexible and scalable *e*FPGA architecture
  - Placed & Routed Layout of the programmable Logic Array.
  - SoC Application