SAND2015-10890C

Improving ASIC Reuse with Embedded FPGA Fabrics

John Teifel, Matthew E. Land, Russell D. Miller
Sandia National Laboratories
Albuquerque, NM 87185, U.S.A.
jteifel@sandia.gov

Abstract: *We investigate an embedded FPGA fabric that may be able to improve ASIC reuse across multiple applications. This fabric is automatically generated from a high-level architecture specification, synthesized to existing standard-cell libraries, and laid out using commercial tools. The area and performance of the fabric is evaluated across several silicon technology nodes.*

Keywords: Embedded FPGA; hybrid ASIC; IP reuse.

Introduction

As ASIC development cycles and fabrication costs continue to escalate with each successive technology generation, government applications are increasingly less able to pursue custom ASIC designs. Mask configurable ASICs have helped to significantly reduce ASIC NRE costs and development schedules [1], but such devices still suffer from the inability to be reconfigured after manufacturing to address new application requirements. While embedding hard FPGA blocks within a larger ASIC to provide low-cost post-fabrication re-configurability is the dream of many system-on-chip designers [2], these hard FPGA blocks are not widely available as commercial IP. As a result, soft FPGA blocks [3][4][5][6], those that can readily be implemented with standard cell libraries, have been proposed as an alternative to the hard FPGA layouts. In this paper, we investigate the feasibility of using these synthesizable embedded FPGA fabrics in government ASIC applications.

A notional embedded FPGA chip is illustrated in Figure 1, where the FPGA block is allocated 17% of the total ASIC die area. It is assumed that some fraction of the FPGA's I/O signals will interface with internal ASIC circuits and some fraction will go off-chip. Additionally, it is assumed that the embedded FPGA block is reconfigurable upon power up and that its configuration bits are stored in shift register logic, compatible with standard cell library implementation flows. The remaining sections of this paper outline the design of this embedded FPGA block and its physical realization.

Embedded FPGA Fabric

The embedded FPGA fabric uses a traditional island-style programmable logic architecture, as shown in Figure 2a, where configurable logic blocks (CLBs) and bidirectional I/O pad resources are connected together with routing blocks (RBs). The CLBs contain LUTs (lookup tables) and DFF registers to realize digital logic. The RBs, which implement both connection box and switch box routing functions [7], wire the CLBs into useful applications. Unidirectional routing is used in the RBs so that they can be realized with standard-cell ASIC libraries (bidirectional routing requires tristate buffer logic that is incompatible with most standard-cell libraries and synthesis flows, and in most cases requires more area [8]). The configuration bits for the CLBs, RBs, and I/O resources are stored in simple shift registers, as shown in Figure 2b, where a configuration_done signal is used to tristate the I/O ports until the FPGA's programming is complete (to prevent drive contention during shifting). Almost all of the circuit features of the embedded FPGA fabric are parameterized, as illustrated in Figure 2c-g, including the size of the LUTs, the number of LUTs in the CLB, the number of input and outputs to the CLB, and the number of routing tracks and crossbar connectivity in the RBs. The routing length, the number of CLBs a routing wire crosses before having to go through a multiplexor in an RB, is also parameterized.

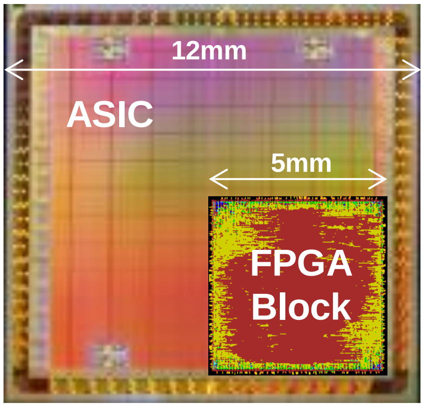


Figure 1: Notional 5x5-mm^2 FPGA block embedded within a larger 12x12-mm^2 ASIC.

Figure 3a shows the software tool flow that is used to convert RTL designs into embedded FPGA programming bit streams. First, a commercial ASIC tool (Synopsys DesignCompiler) is used for front-end logic synthesis to optimize the RTL and transform it into a simplified Verilog netlist of gates with LUT-style functions. Second, the open-source VTR tools [9] are used to map, place, and route the LUT-style netlist onto the embedded FPGA fabric. The VTR tools enable the architecture of the embedded FPGA fabric to be described using a concise, high-level XML-based specification, allowing it to be easily parameterized. It should also be noted that while the VTR tools accept Verilog RTL as an input format, only a very limited subset of the language is supported and hence the reason for the front-end logic synthesis stage using a commercial

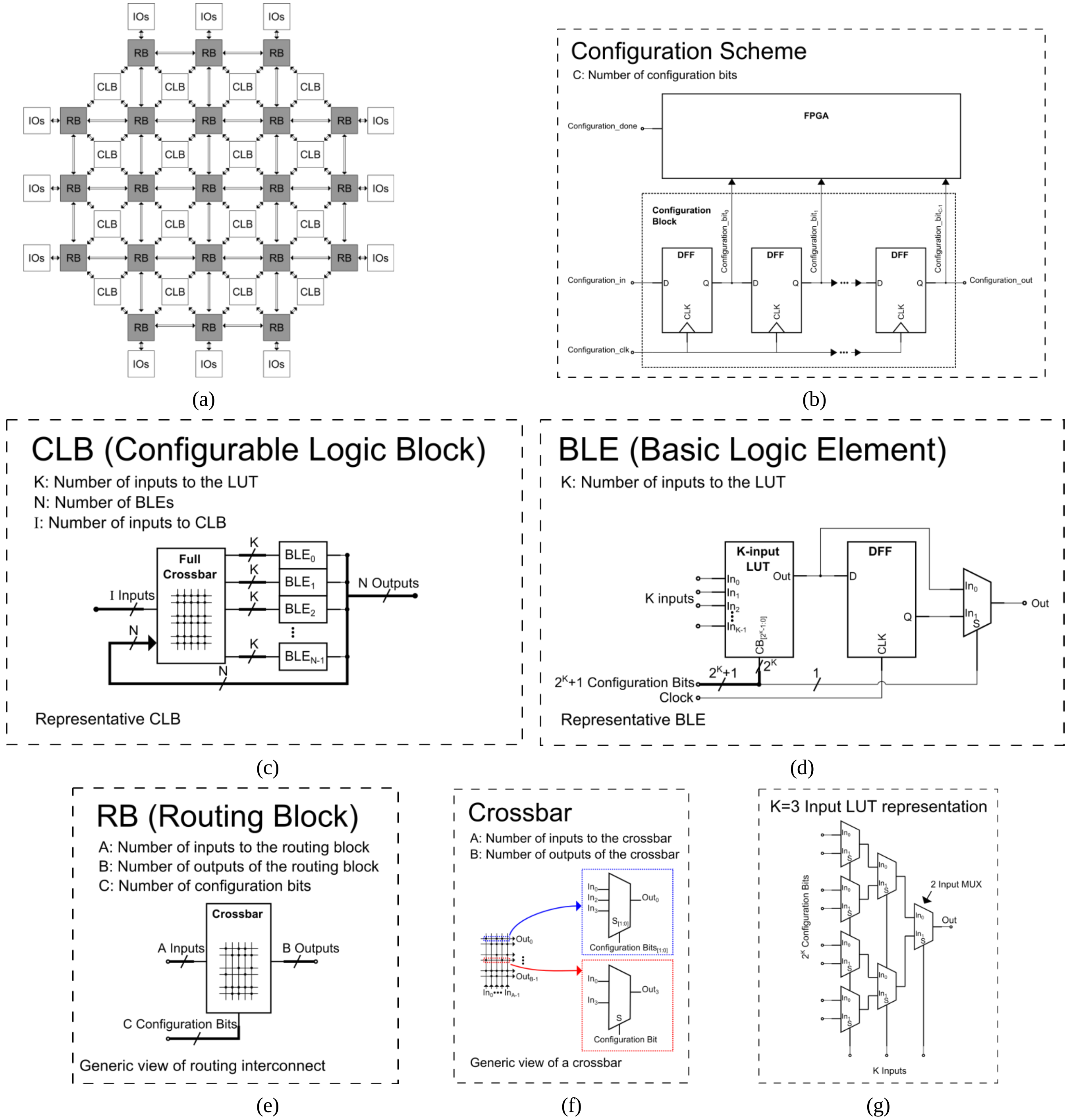


Figure 2: Flexible, parameterized embedded FPGA fabric (a) architecture, (b) configuration circuit, (c) Configurable Logic Block, (d) Basic Logic Element, (e) Routing Block, (f) crossbar circuit , and (g) Look-Up-Table circuit.

synthesis tool. Finally, a custom Sandia tool (gen_efpga_bitstream) is used for the back-end programming bit-stream generation.

Advanced FPGA features, such as arithmetic carry chains, hard multipliers, and block memories were not considered in this paper, but would be straightforward to incorporate into the circuit and software framework described above.

Architecture Studies

Using the benchmark circuits listed in Table 1, detailed area studies were performed to determine optimal values for the embedded FPGA's circuit parameters. In the CLB, the LUT size was varied from 3-5, the number of LUTs was varied from 3-5, and the number of inputs was varied from 4-15. The F_c in the RBs, which is the fraction of routing wires an individual port can connect to in a given

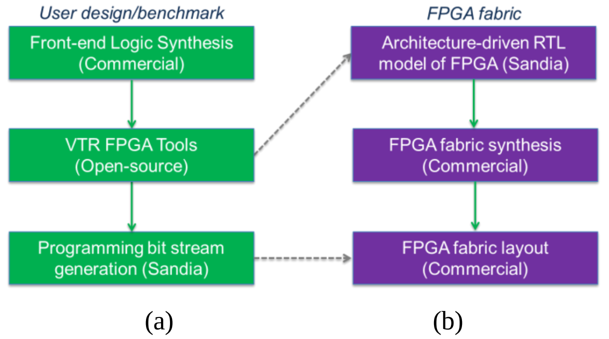

User design/benchmark
Front-end Logic Synthesis (Commercial)
VTR FPGA Tools (Open-source)
Programming bit stream generation (Sandia)
FPGA fabric
Architecture-driven RTL model of FPGA (Sandia)
FPGA fabric synthesis (Commercial)
FPGA fabric layout (Commercial)
(a)
(b)

Figure 3: (a) FPGA software tool flow for benchmark designs. (b) FPGA fabric physical realization flow.

crossbar, was set to 0.15, 0.25, 0.35, 0.45, or 1.0 for the CLB inputs, CLB outputs, and I/O routing. By leveraging VTR's automatic-size-scaling capability, where the number of CLBs and the number of routing tracks are minimized for smallest area, each benchmark was implemented onto a given FPGA architecture configuration. The area of the benchmark was then estimated, using a custom Sandia tool, based on the number of required flip flops and the number of equivalent two-input multiplexors. The accuracy of this estimator tool was verified to be within +15% of the synthesized gate area. Figure 4 plots the geometric mean of the normalized area across all of the benchmarks for 248 of the most promising FPGA architecture configurations. The most area-optimal architecture setting was for a CLB with nine inputs and four three-input LUTs.

Table 2 summarizes the results of these detailed architecture experiments, and the values chosen for use in this paper. Most of these parameters were set to minimize the embedded FPGA's area. The routing length, is one exception, which was set to span four CLBs to improve clock speeds. Also, without loss of generality, the number of routing tracks and the CLB array size were set to their values to support the physical design experiments in the next section, and would most likely be sized differently depending on the end-use application.

Physical Design

Figure 3b shows the physical design flow used to realize the embedded FPGA fabric. First, the VTR tools are used to generate a resource graph of the embedded FPGA from a high-level XML architecture specification, and a custom Sandia tool (gen_efpga_structural_verilog) is used to generate a Verilog RTL model of the embedded FPGA block from the VTR resource graph. Second, the Verilog RTL model is synthesized using a commercial tool (Synopsys DesignCompiler) and laid out using a commercial auto-place-and-route tool (Cadence Encounter). Finally, a commercial tool (Synopsys PrimeTime) is used to determine the speed of the FPGA after it has been programmed with a benchmark or end-user design.

Table 1: Benchmarks used for architecture studies.

Benchmark Name	# Flip-flops	# LUTs (K=3)
SNL Design	1375	5671
ch_intrinsics [9]	489	1466
diffeq1 [9]	193	6268
diffeq2 [9]	96	6071
mkPkgMerge [9]	7380	17100
mkSMAdapter4B [9]	5431	14837
or1200 [9]	2739	15014
raygentop [9]	6784	25786
sha [9]	911	4738
stereovision0 [9]	13405	17147
stereovision3 [9]	102	400

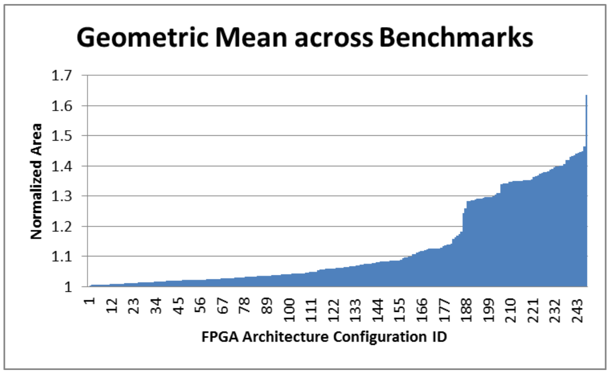


Figure 4: Area results of architecture parameter studies.

Table 2: Embedded FPGA fabric parameters.

Parameter	Fabric Support	Values used in this paper
K (LUT size)	>1	3
N (# LUTs per CLB)	>0	4
I (Inputs into CLB)	>1	9
F_c for CLB inputs	0.0 - 1.0	0.15
F_c for CLB outputs	0.0 - 1.0	0.25
F_c for IO inputs	0.0 - 1.0	0.25
F_c for IO outputs	0.0 - 1.0	0.25
W (# of Routing Tracks)	>1	38
CLB Array Size	Arbitrary	10x10
L (Routing Lengths)	Arbitrary	Length-4
Routing Hierarchy	Arbitrary	Single length
Clock Domains	1	1

Figure 5 shows the density and performance of the embedded FPGA fabric after it was synthesized and laid out in 350-nm, 90-nm, and 45-nm CMOS technologies. As expected, the area and speed is more competitive in smaller geometry technologies. For example, a 45-nm process technology is required to achieve 100-MHz clock speeds with 16-bit counter logic. The layouts for these embedded FPGA fabrics are shown in Figure 6, and a version of the 350-nm FPGA fabric is currently being fabricated.

Summary

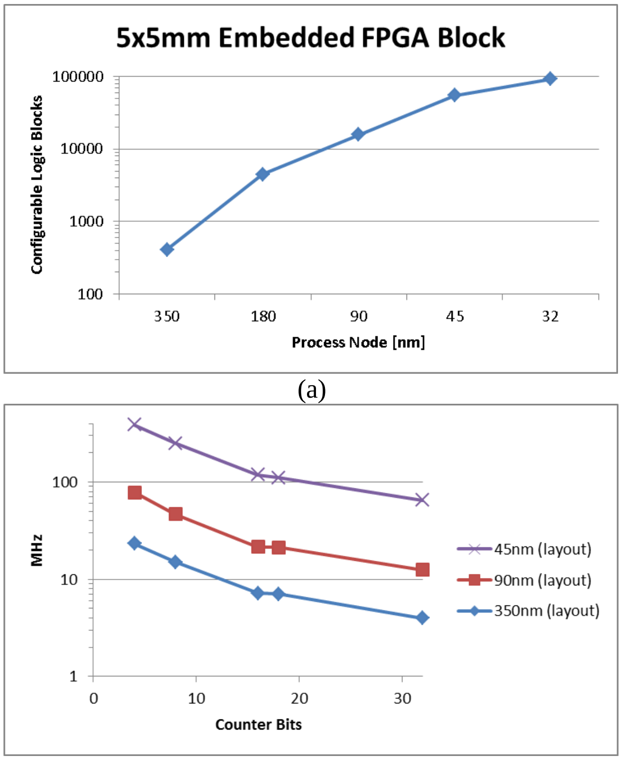
Embedded FPGA fabrics were evaluated as a method for improving ASIC-level IP reuse, and an automated design flow was developed to rapidly insert them into standard-cell ASICs using a combination of commercial and open-source tools. Ultimately, government applications will need to weigh the performance/density overheads of these programmable logic fabrics versus the benefit they provide in enabling portions of an ASIC to be reconfigurable.

Acknowledgements

Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

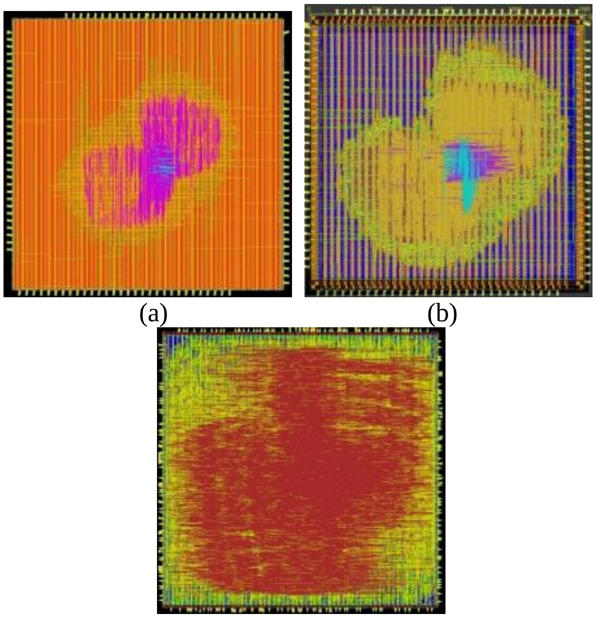
1. J. Teifel, et. al., ViArray Standard Platforms: Rad-Hard Structured ASICs for Digital and Mixed-Signal Applications. IEEE Aerospace Conference, 2012.
2. M. Borgatti, et. al., A 1GOPS Reconfigurable Signal Processing IC with Embedded FPGA and 3-Port 1.2GB/s Flash Memory Subsystem. IEEE ISSCC, 2003.
3. S. J. E. Wilton, et. al., Design Considerations for Soft Embedded Programmable Logic Cores. IEEE JSSC, vol. 40, no. 2, Feb 2005.
4. V. Aken'Ova, et. al., An Improved "Soft" eFPGA Design and Implementation Strategy, IEEE CICC, 2005.
5. www.adicsys.com
6. www.menta.fr
7. U. Farooq, et. al., "FPGA Architectures: An Overview," in Tree-based Heterogeneous FPGA Architectures, New York, NY, USA: Springer, 2012, pp. 7-48.
8. G. Lemieux, et. al., Directional and Single-Driver Wires in FPGA Interconnect, IEEE ICFPT, 2004.
9. J. Luu, et. al., VTR 7.0: Next Generation Architecture and CAD System for FPGAs, ACM Trans. on Reconfigurable Technology and Systems, vol. 7, no. 2, Jun 2014.



(a)

(b)

Figure 5: (a) Number of 3-input configurable logic blocks in a 5x5mm embedded FPGA for various process nodes. (b) Speed of various counter benchmarks using worst case military-spec conditions (slow transistors and wires, 125C temperature, 90% Vdd).



(a)

(b)

(c)

Figure 6: Embedded FPGA layouts (1st pass layouts; not optimized for smallest block size) in various standard-cell technologies: (a) 45-nm (1.5x1.5-mm^2) (b) 90-nm (2x2-mm^2), (c) 350-nm (6x6-mm^2).