Microelectronics Journal 45 (2014) 217–225

Contents lists available at ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

A regular fabric design methodology for applications requiring specific layout-level design rules

Sophie Dupuis [a], Ludovic Noury [b,*], Nicolas Fel [c]

[a] LIRMM - UM2/CNRS, 161 rue Ada, 34095, Montpellier cedex 5, France
[b] ESYCOM, ESIEE Paris, Université Paris-Est, F-93162 Noisy-le-Grand, France
[c] CEA, DAM, DIF, F-91297 Arpajon, France

ARTICLE INFO

Article history:
Received 21 January 2013
Received in revised form
2 November 2013
Accepted 6 November 2013
Available online 22 November 2013

Keywords:
Regular fabric
Structured ASIC
ASIC
FPGA
Design flow

ABSTRACT

Regular fabrics have been introduced as an approach to bridge the gap between ASICs and FPGAs in terms of cost and performance. Indeed, compared to an ASIC, by predefining most of the manufacturing masks, they highly reduce time-to-market, non-recoverable engineering costs and lithography hazards. Also, thanks to hardwired configuration and interconnections their performance is closer to those of ASICs than those of FPGAs. They are therefore well suited to many applications requiring low to medium volume applications or higher performance than those provided by FPGAs.

In this paper, we evaluate the interest of using a regular fabric to reduce time and design cost significantly in applications involving specific transistor level design (radiative/spacial conditions, side-channel attacks, NMR environment, etc.). With this aim in view, after a broad state of the art overview with an emphasis on architectures and design flows, we develop our approach of a regular fabric designed to limit layout level design, ad-hoc tools and technological migration cost. Then, we evaluate its performance in a 65 nm process versus FPGA and standard cell based ASIC implementations. For sequential designs, our proposed solution is on average 2.5 × slower and 2.3 × bigger than a standard cell implantation, but also on average 13 × faster than a FPGA.

1. Introduction

The two main solutions for implementing an integrated circuit are either manufacturing a standard cell based application-specific integrated circuit (ASIC) or configuring a field-programmable gate array (FPGA). On one hand, ASICs provide higher clock frequency, much smaller area and reduced power consumption, compared to FPGAs. On the other hand, designing for FPGAs targets is simpler and faster. Moreover FPGAs are well suited to applications' late evolution since they can be reconfigured [18].

Also, today's highly submicronic processes (<90 nm) induce dramatically increasing masks cost and multiple prototyping runs due to sub-wavelength lithographic complexity [31], making the design of an ASIC a very expensive and time consuming process, suitable only for very high volume applications [20]. To return to cost-effective manufacturing, the International Technology Roadmap for Semiconductors [13], insists on *design for manufacturability* development to obtain lithography-friendly designs by using regular layout styles to comply with increasingly more restrictive design rules. FPGAs could be an interesting alternative, but only for small production runs due to their high per-unit cost [33].

These constraints lead to the problem of chips production in quantities between 10k and 50k units and/or needing more performance than achievable with FPGAs [26]. To solve this problem, the exploration of the design space existing between high performance ASICs and low non-recoverable engineering (NRE) costs FPGAs has received much attention during the last decade [3,26]. This has led to today mask-programmable regular fabrics, the successors of yesterday's gate arrays and seas of gates [8].

Regular fabrics are usually based on a regular array of one fundamental logic element, called a tile. The manufacturing masks defining these tiles are designed only once. Then, for each application, the functionality is determined by grouping and/or configuring the tiles with the definition of the remaining metalization layers. Therefore, these technologies rely upon a two steps design: (1) define most of the masks (at least, the masks up to the metal 1, which are the most complex and expensive masks to manufacture) and eventually manufacture the corresponding partially processed generic wafers (some interconnection masks can also be predefined and specialized later through vias), (2) specialize the generic dies by defining the 1–10 remaining metalization masks (vias and/or wires) and manufacture the corresponding complete functional chips. This evolution from an ASIC

* Corresponding author. Tel.: +33 1 45 92 60 75.
E-mail addresses: sophie.dupuis@lirmm.fr (S. Dupuis), ludovic.noury@esiee.fr (L. Noury), nicolas.fel@cea.fr (N. Fel).

0026-2692/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.mejo.2013.11.002

design-flow (20–35 masks to develop, validate and manufacture for each new circuit) to a regular fabric design-flow (only 1–10 masks to develop, validate and manufacture for each new circuit) greatly reduces cost [33] and optical proximity correction (OPC) hazards [9,15] while maintaining performance close to those of a standard cell ASIC.

Yet, many applications are usually implemented with ASICs, even if they involve only limited production volume and require only performance comparable to those of FPGAs devices. In fact, these applications usually require specific design at transistor level (or cell level) to comply with non standard environments operational constraints such as radiative environments (space and military applications [23]), secure computing (countermeasure against side-channel attack by designing cells with controlled power signature [10]), strong magnetic fields (medical Nuclear Magnetic Resonance spectroscopy [7]), high temperature/pressure or corrosive environment, thus excluding the use of commercial FPGAs. As a matter of fact, because the standard cell libraries of manufacturers are unsuitable for those applications, a complete library must be developed, characterized and validated by physical characterization in its specific environment after a test run, for every technological process. As the availability of an advanced technological process for low volumes is about 3 years, with more and more complex drawing rules, the required development effort is unsustainable. This leads to the use of a regular fabric, which can be designed with cells complying with specific constraints with a lower cost. To the best of our knowledge, the use of regular fabrics in this context has not been studied yet; most of the research effort is dedicated to general applications with the goal to get as close as possible to ASIC performance while minimizing the number of necessary masks to personalize the circuit.

Most state-of-the-art regular fabrics require a complex full-custom design often incompatible with some application specific layout constraints, in addition to time consuming dedicated EDA software development and validation. Hence our motivation to investigate the potential of regular fabrics for applications requiring specific layout level designs rules. To this end, our aim is to target satisfactory performance associated with a minimized design time of the regular fabric in order to minimize cost and facilitate any future technological migration. Another potential advantage of regular fabrics in this context is that when the current technological process is no longer available, the final metalization process could be carried out by another foundry; provided that the lower layers have already been pre-manufactured. In this article, we extend the work presented in [24] which aimed at presenting a regular fabric designed to be used as a reference to evaluate the cost-performance trade-off of regular fabric architectural enhancements.

Our contributions are independent of any specific layout level technique: (1) we give an extensive state of the art including published regular fabric ad-hoc tools and layout level design complexity, (2) we define a specific regular fabric design methodology (using standard industrial EDA tools only and minimizing layout level development time), (3) we demonstrate our concept using a real 65 nm process with results generated from post place and route level layout extraction. Furthermore, we focus on performance/metal levels trade-offs to see if our regular fabric can be routed with the same constraints as the standard cell equivalent.

This paper is organized as follows. In Section 2, we present an extensive state of the art of regular fabric technology. For each approach, we give detail about the architectures, the corresponding design flows and performance. In Section 3 we describe our proposed methodology to convert a standard cell industrial ASIC design flow into a regular fabric design flow with the minimum amount of modifications. Then in Section 4 we present a detailed performance analysis in terms of delay and area results obtained from layout level extraction in a 65 nm process, along with a comparison with standard cell ASIC and FPGA implementations. Our goal is to explore the trade-offs between performance and the number of routing layers. Finally we conclude and present our reflexions about future work.

2. Previous work

In this section, we describe and analyze recently published regular fabric designs. As summarized in Table 1, for each architecture we study not only the parameters influencing its performance (tile architecture, configuration mechanism, sequential design support, etc.) but also those relative to the regular fabric development and migration cost (ad-hoc software and standard cell library equivalent).

2.1. Regular fabrics architectures

A regular fabric is an array of identical mask-configurable predefined elements called tiles. Vendors often integrate hard IP such as PLL and programmable I/O in a regular fabric, resulting in a structured ASIC. However, the properties of a regular fabric are mostly defined by its tile architecture.

We differentiate three types of tiles: (1) "*homogeneous configurable tiles*" i.e. tiles using only one type of configurable mechanism (PLA, LUT, etc.), (2) "*heterogeneous configurable tiles*" i.e. tiles using several types of configurable mechanisms, and (3)"*homogeneous non-configurable tiles*" i.e. non configurable tiles.

2.1.1. Homogeneous configurable tile

PLA-based tiles: In [21], Mo and Brayton present two regular structures based on programmable logic array (PLA) tiles. After synthesis with a dedicated software, neither placement nor routing are necessary since they are directly determined by the synthesis. Customization is done by using via and metal layers. These structures can only handle up to 10k-gate combinational circuits.

In [14], Jayakumar and Khatri add sequential design support by co-locating each PLA output with a D flip–flop (DFF), but their architecture also requires a standard place and route step to connect the configured tiles.

Pass-transistor-based tiles: In [11], the tile is an if-then-else (ITE) logic cell based on pass-transistor logic (cf. Fig. 1(e)). Local and global routing are done by metal and via customization. Sequential designs are handled by predefined DFF embedded in the ITE array.

LUT-based tiles: In [25], Patel et al. study a regular fabric with via-configurable look up table (LUT) tiles (cf. Fig. 1(a)); three to five inputs LUT are evaluated. Each LUT functionality is determined by adding (or not) vias at predetermined locations using a single via mask. To handle sequential design, each tile also includes a DFF.

In [34], Yuen et al. present an alternative with an array that includes one DFF tile for eight LUT tiles.

In [6], the tile is called a complementary universal logic gate. It is constructed with cross-coupled PMOS pull-up loads, complementary NMOS pull-down logic networks and output inverters. It realizes sequential functions without predefined DFFs by configuring two LUT tiles as latches.

Gate-based tiles: A via-configurable tile called gain-based logic block (GLB) is presented by Hu et al. in [12]. The tile consists in two NAND2, one NAND3 and one NOR3 gates. Most inputs/outputs of each cell can be directly accessed through via-programming (cf. Fig. 1(b)). Also, the inputs of the tile can be negated and the capacitance of its outputs can be configured. Sequential designs are not supported.

An alternative presented in [22] uses a via-programmable exclusive OR gate called VPEX (cf. Fig. 1(c)) and handles sequential designs with flip–flops built using 5 tiles.