



Contents lists available at ScienceDirect

Integration, the VLSI Journal

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

A pattern-based routing algorithm for a novel electronic system prototyping platform

Étienne Lepercq^a, Yves Blaquière^{b,*}, Yvon Savaria^a

^a Electrical Engineering Department, Polytechnique Montréal, C.P. 6079, Succ. Centre-ville, Montréal, Québec, H3C 3A7, Canada

^b Electrical Engineering Department, École de Technologie Supérieure, 1100 Notre-Dame Ouest Street, Montréal, Québec, H3C 1K3, Canada

ARTICLE INFO

Keywords:

Routing algorithm
Interconnection network
Rapid prototyping
Wafer-sized integrated circuit

ABSTRACT

A recently proposed wafer-sized active integrated circuit capable of programmably interconnecting integrated circuits deposited on its surface needs a routing tool with computation time in the order of minutes. In this paper, a first algorithm computes the shortest route in $O(n)$, n being the number of edges between source and destination. The second algorithm performs a parallelized random search to resolve conflicting routes. Our algorithm can route high density PCB-like netlists (25% vertices occupancy) on an 80,000 vertices regular interconnection network in about 9 min, while typical density netlists (5–15%) are routed in times ranging from 0.4 to 11 s.

1. Introduction

An active reconfigurable platform using a Wafer-Scale Integrated (WSI) Circuit called WaferICTM was proposed for rapid system prototyping by Norman *et al* [1]. This WSI circuit and parts of the prototyping platform were designed, fabricated and successfully tested as reported in Ref. [2]. A main benefit of using WSI in this prototyping system is to obtain a large enough smart active surface and interconnect network at an affordable cost, which is made possible by using mature 180 nm CMOS technologies with 6–8 inch wafer sizes [1]. Another application for neural network was implemented using WSI as part of the European research project “FACETS” [3]. Apart from these two recent applications, multiple WSI implementations were done in the '80s, as briefly summarized in Ref. [4]. Thus, there are applications where WSI or large area integrated circuits have unique features that enable new classes of systems that are feasible and affordable.

Today's electronic system prototyping workflows imply a sequence of iterative steps alternating between PCB development, prototype manufacturing, manual debugging, and PCB fixes. One iteration through these steps can take several weeks on complex systems, and multiple iterations may be needed. This has a negative impact on time-to-market and productivity. While Field Programmable Interconnect Chips (FPICs) [5] are capable of reconfigurably interconnecting electronic components, a printed circuit board must be built to match the contact patterns of those components. Furthermore, the limited capacity of FPICs

requires the use of multiple chips for the development of high-end circuit boards. This imposes strong limitations on system density and dramatically increases the cost of each prototype.

To overcome these limitations and to significantly reduce electronic system development time, the prototyping platform providing a framework for this research was proposed by Norman *et al* [1,6]. This prototyping system is based on a wafer-sized CMOS circuit that includes a field programmable interconnection network FPIN (see schematic Fig. 1) [2] configured with a reconfigurable and defect tolerant JTAG scan chain [7]. This network links a large number of contact points (called NanoPads), each one being configurable as an I/O signal or as a programmable power supply. The NanoPads are grouped by 16 in a simple building block called a cell, where each one supports up to two integrated circuit (IC) pins, i.e. each cell has two “access points”, using the terminology used later in this paper. A user simply places integrated circuits on the active surface and an array of NanoPads detects the IC pins. ICs are then programmably interconnected through the defect-tolerant network and the system is ready to run. This step requires a routing algorithm to compute non-conflicting routes through the FPIN. Signal integrity is maintained by internal repeaters, mitigating crosstalk and attenuation, and ensuring a propagation delay mostly proportional to the wire length. The FPIN implemented with integrated wires and repeaters introduces delays several times longer than same length PCB copper traces, exacerbating the necessity for an algorithm that focuses on minimizing propagation delays.

* Corresponding author.

E-mail addresses: e.guepe@gmail.com (É. Lepercq), Yves.Blaquiere@etsmtl.ca (Y. Blaquière), Yvon.Savaria@polymtl.ca (Y. Savaria).

<https://doi.org/10.1016/j.vlsi.2018.03.005>

Received 14 October 2016; Received in revised form 3 March 2018; Accepted 7 March 2018

Available online XXX

0167-9260/© 2018 Elsevier B.V. All rights reserved.

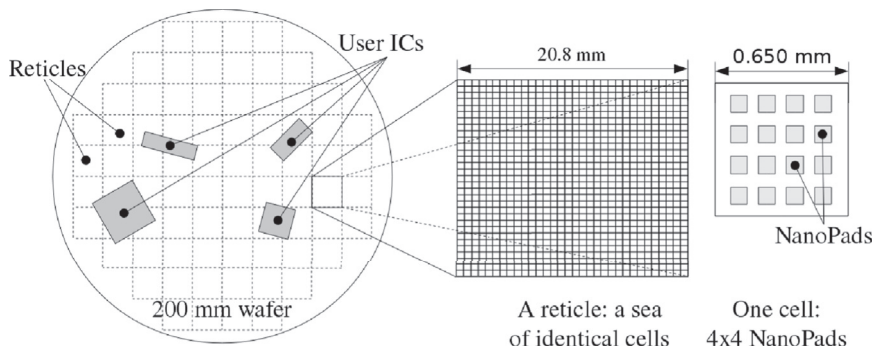


Fig. 1. WaferIC - the core of the active reconfigurable platform: user's IC are manually deposited on the wafer's surface, external connections are possible as well as probing digital on-wafer interconnection signals in real time.

User's ICs are powered through the NanoPads, each cell directly supporting several power levels (fed through wafer vias and tied by suitable power grids). Therefore, the routing algorithm works with PCB-like netlists as input, but with the notable exception that all power and ground nets are not routed. The router uses the predefined FPIN resources and architecture. This is somewhat similar to FPGA routers linking logic cells using a network architecture different from the one used in our prototyping platform. Most FPGAs use a multi-level architecture (usually, with two levels), as described in Refs. [8] and [9] with intra-cluster and inter-cluster architectures, where each logic block connects to communication channels and shares switching blocks resources. In the targeted technology, each cell has access to routing resources with its own embedded crossbar, connected to different cells both as inputs and outputs.

Besides these differences, the proposed platform and FPGA interconnection networks share some common characteristics related to VLSI and PCB routers. Two families of algorithms exist in the literature. Concurrent ones like k-SAT [10] and Integer Linear Programming [11,12] or those based on Lagrangian Relaxation approaches [13] that are not widely used when short computation times are needed. This is the case when the cost of computing the solution outweighs the benefits derived from an improved solution quality. By contrast, algorithms of the second family are sequential. They are based on problem specific heuristics for PCB, VLSI or FPGA routing. Several FPGA routing algorithms are derivatives of the famous PathFinder [14,15] and VPR [16–18] algorithms. They are designed to find a good balance between performance and routability, especially on networks with scarce resources. Sequential algorithms are mainly focused on congestion management heuristics, based on votes and rip-and-reroute techniques. Some algorithms exploit a negotiation-based global router [19] for standard cell design, in addition to a maze router [20] to interconnect source-destination pairs. Our implementation uses the A^* algorithm, which is fast and provides optimal solutions when the heuristic used to estimate distance between source-destination pair never overestimates the distance to the destination (*admissible* theorem defined in Ref. [21]). Such approach is common also in VLSI routers such as [22], where sequential routers use iterative rip-up and re-route techniques with penalty cost functions based on the congestion history of each edge. Their goal is to minimize total wire length in very dense benchmarks, like [23–25], while still providing acceptable computation times. In this paper, the target computation time is expected to be in the order of minutes. This is justified by the set up time and development iteration loop time for the prototyping platform (also called WaferBoard), which is expected to be a few minutes. This contrasts with VLSI development time (measured in weeks or months).

This paper proposes an efficient linear-complexity algorithm to compute optimal delay paths for any given pair of nodes in power-of-two multi-dimensional interconnection networks. This algorithm was successfully implemented into a new routing tool for the novel electronic systems prototyping platform that programmably interconnects any NanoPad to any others. This tool can route high density PCB-like netlists with computation time in the order of minutes without generating con-

flicting routes. This rapid conflict free routing is an essential feature of the target application. It is proven that it computes delay-optimal solutions, and can serve as an *admissible* heuristic in an A^* algorithm. Two approaches are proposed to significantly reduce the computation times of sequential routing algorithms: a permutation approach for solving routing conflict by minimizing the number of routes processed by the A^* algorithm, and an “edge-based” routing approach, which reduce the processing time while having a very low impact on the routing quality.

This paper presents the FPIN and its associated graph representation in Section 2. In Section 3, the proposed algorithm called PermFinder is formulated and its important features are analyzed. Section 4 is dedicated to our results, which show how the algorithm can trade off propagation delays and computation times, key parameters influencing these goals, and comparisons to the PathFinder algorithm. A conclusion closes the paper.

2. The wafer-scale field programmable interconnection network (FPIN)

The FPIN embedded in the targeted application is a multi-dimensional mesh. The FPIN needs to cope with generally long interconnects compared to FPGA or VLSI circuits. Since crossing nodes are roughly ten times slower than the wire delay between two nodes in the technology used, long wires avoid this penalty. However since interconnection delays are non-linear in a CMOS circuit for such long connections, repeaters are regularly inserted to achieve a linear delay with distance. This network architecture provides faster paths for long distances than using neighbor-to-neighbor connections, which makes minimum-delay routes not necessarily the shortest in terms of distance. To our knowledge, this non-linear relationship between delay and distance is unique among published FPGA, VLSI and PCB interconnection networks. This contrasts with the PCB or VLSI routing problems that can be modeled with a simple mesh, where each vertex is fully connected to its 4 (2 dimensional mesh) or 6 neighbors (3 dimensional mesh). Global routers for FPGAs use such model because the delay of a path is directly proportional to its length. In the case of the targeted application, the delay model is slightly different, consequently, a new mesh model is needed for the proposed routing algorithm.

Definition 1. The field programmable interconnection network FPIN is a regular grid of crossbars, where each crossbar is modeled as a vertex. For each vertex v , let us define a set S_v of distant neighbors in the four directions: the distance between v and its neighbors is expressed by 2^d , $d \in [0; D]$. D is the dimension of the network and is a positive integer. These distances are power-of-two, and an edge is a wire that links v and a vertex in the set S_v .

Such construction is depicted in Fig. 2, where direct connections (i.e. wires) from the vertex at coordinates (0;0) to its neighbors at distance $2^0, 2^1, 2^2$ are shown. These wires exist for each vertex in the FPIN. D , the dimension of the network, is sufficient to define λ_{\max} , the length of the longest wire. For example in the network of Fig. 2, $D = 2$ with the longest wire being $\lambda_{\max} = 2^D = 4$. Repeaters are placed on long wires to

Download English Version:

<https://daneshyari.com/en/article/6942158>

Download Persian Version:

<https://daneshyari.com/article/6942158>

[Daneshyari.com](https://daneshyari.com)