Workflow for an Electronic Configurable Prototyping System

Etienne Lepercq[1], Yves Blaquière[2], Richard Norman, Yvon Savaria[1]
École Polytechnique de Montréal (1), Université du Québec à Montréal (2)
Montréal, Canada
{etienne.lepercq, yvon.savaria}@polymtl.ca (1) blaquiere.yves@uqam.ca (2)

***Abstract*— A recently proposed rapid electronic system prototyping technology based on a WSI active configurable circuit board is capable of programmably interconnecting integrated circuit packages deposited on its surface, which comprises over one million contacts. This technology has some similarities, but also some key distinctive constraints when compared to conventional printed circuit boards. A workflow that supports the design with such configurable circuit boards is proposed. As part of this workflow, algorithms and tools for package recognition and for routing through a multi-dimensional mesh interconnection network is proposed and implemented. Results reported in this paper confirm the feasibility of the proposed workflow and several architectural choices made with respect to the configurable circuit board technology. Using the prototype tools reported in this paper, packages are successfully recognized and netlists are routed even though they use up to 50% of the contact point resources, which corresponds to an extremely dense circuit board.**

I. Introduction

An active reconfigurable circuit board using a Wafer-Scale Integrated Circuit [1] has been recently proposed for rapid electronic system prototyping [2]. It includes an interconnection network linking a large number of contact points (called NanoPads), each one being configured as a signal I/O or as a programmable power supply (Fig. 1). A user simply places integrated circuits (ICs) on the active surface and an array of sensors detects component pins contacting NanoPads. ICs are then programmably interconnected through a defect tolerant network, and the system is ready to run. In this system, signal integrity is maintained by internal repeaters, mitigating cross-talk and attenuation, as well as eliminating many passive components. The network can also capture any signal and export it out for debugging. With this prototyping system, the development time can be reduced significantly.

Hardware design and fabrication of the WaferIC, a core building block of the proposed configurable circuit board technology, has been extensively studied and its feasibility confirmed [2]. Analysis and design of the required defect tolerant interconnection network have been presented in [3].

This paper explores the tools and workflow needed to support rapid system prototyping based on the configurable circuit board. It involves notably diagnosing defects that can be presented in the WaferIC (a wafer scale circuit) to avoid using such defective resources. The system must also detect and recognize balls (or pins) of components placed on its surface. It must process user netlists comprising hundreds to thousands of nets, and user constraints definition files, to build a bit-stream comprising megabytes of data to configure several million interconnect segments in the configurable circuit board. These tasks cannot be done without the support of suitable tools, and the purpose of this paper is to define a workflow for the required software tool-chain. It is based on the specification provided by the Workflow Management Coalition (WfMC), which includes a workflow reference model as a guideline for developing workflow processes [4], and have been widely used to build workflows in general [5] and specifically in computer-aided design tools [6-7]. Two critical tools that are part of our proposed workflow, the package recognition and the routing modules, are presented in this paper. They support and confirm the feasibility of the proposed workflow.

The paper is organized as follows. Section II describes steps involved in the proposed workflow. Section III describes two of its critical tools. Section IV presents results of the package recognition process, and of the routing algorithm. Section V concludes the paper.

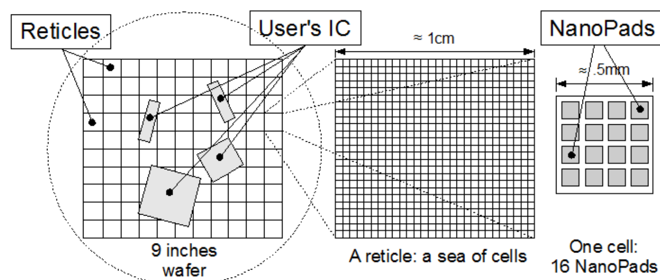

Figure 1: The DreamWafer board: user's IC are deposited on the wafer's surface, external connections are possible as well as probing digital on-wafer interconnection signals in real time.

II. Workflow Description

The proposed workflow includes eight steps, as shown in figure 2. The boot-up and diagnostic process (1) automatically powers up the wafer and scans the whole wafer to extract a defect map to ensure that the application will not make use of such resources. A map of connected NanoPads is then extracted (2). A connected NanoPad is defined as a NanoPad in contact with an IC package ball. The map is extracted from a sea of about a million contact sensors embedded into the WaferIC, based on shorts created by an IC solder ball between adjacent NanoPads.

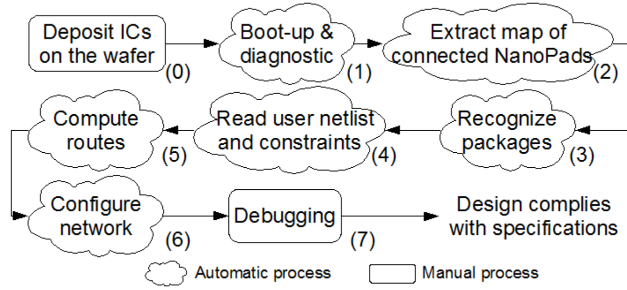

Figure 2: Proposed workflow: from WaferIC characterization up to a working system prototype.

This contact map is then used by the IC package pin/netlist recognition process (3). This tool is based on pattern matching algorithms, as proposed by the six-step workflow in figure 3. Its implementation is detailed in the next section.

A routing process (5) is then executed according to the netlist and constraints provided by the user (4). This process follows the workflow proposed in figure 4. The user netlist defines the connections that must be made between ICs deposited on the surface, which can be manually defined or read from a standard netlist file (e.g. EDIF, GRB, Protel). The routing process consists of reading performance and implementation constraints and then computing a list of routes for each net in the netlist. These constraints have some similarities when compared to that used in PCB routers. However, routes must be assigned to predefined wire

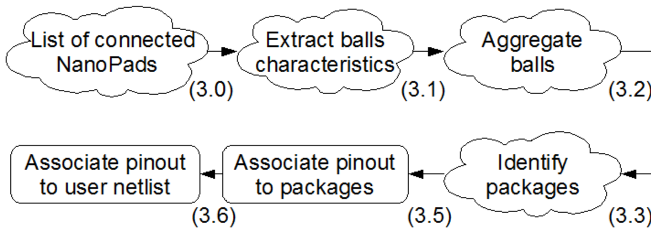

Figure 3: The package recognition workflow, from a map of connected NanoPads to netlist assignment, six steps are proposed.

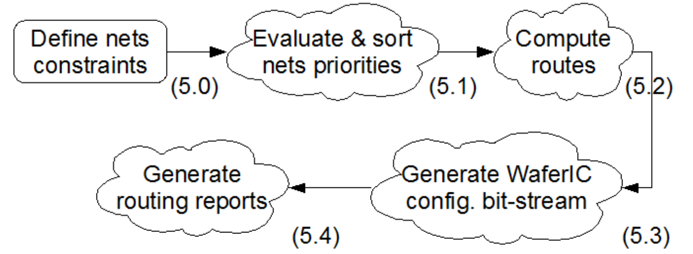

Figure 4: The routing workflow, from constraints definition to wafer bit-stream configuration and results log.

segments of a defect-tolerant multi-dimensional mesh interconnection network (WaferNet^{TM}) described in [3]. Other routing constraints are, for example, bus latency (timing constraints), skew, and bandwidth. Finally, the tool generates the bit-stream to configure the WaferIC^{TM} (6) and routing reports.

This paper introduces in section III a solution to the most common net constraint that consists in finding the shortest path (that exhibits the minimum delay) between NanoPads.

III. Implemented Tools

The implementation of two critical tools of the workflow proposed in the previous section is presented in this section: the package recognition algorithm and the routing algorithm for one type of constraints.

A. *Package Recognition Algorithm*

The recognition of IC packages deposited on the WaferIC^{TM} surface is preceded by a step estimating all ball positions, which are not known *a priori*. Each IC solder ball typically connects to multiple NanoPads, regardless of its position on the WaferIC^{TM} surface. For example, black squares in figure 5 represent NanoPads connected to the same ball and white squares, represent unconnected NanoPads, for which the ball sensor did not trigger. Most feature extraction algorithms make use of the scale-space theory [11] to observe an image at various scales and extract relevant information from that image. For a package using 3x3 reticules (about 3 cm^2), using such techniques involves processing nearly 40,000 pixels, while using extracted ball positions is equivalent to an image of at most 2,500 pixels, which is the number of balls of the biggest FPGA on the market. Thus, as the number of considered elements is much smaller with the proposed method, its processing complexity is also much lower.

The position and size of a ball are estimated from the set of connected NanoPads (Fig. 4), which is equivalent to the geometrical problem of finding the smallest bounding circle for n points in a two dimensional space [12]. We assume that reliable package recognition is possible if and only if all IC balls are connected to at least two NanoPads. If this assumption is not true, it means that some pins have not made a valid contact, and therefore the user IC is not usable. In this case, corrective actions are required such as applying more pressure to the IC or moving it to another place on the surface.

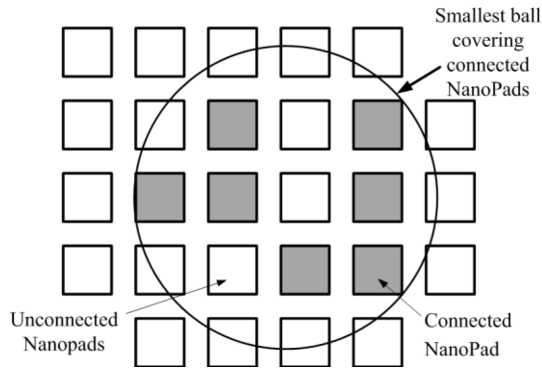

Figure 4: Example of estimated ball size and position from NanoPads connected to the same IC ball.

To aggregate IC balls into a set associated with an IC package, the algorithm first uses an empirical distance threshold by grouping close-enough balls into one package instance. This threshold is determined by the largest known pitch in the package library. Secondly, a package orientation is extracted, based on two IC characteristics: most IC packages are rectangular and most IC balls are aligned, sometimes according to a checkerboard pattern. Finally, package recognition is performed by searching in a library of known IC packages and discriminating according to the number of balls, the ball spacing, position and size. The algorithm that was implemented is based on [14], but the details are beyond the scope of the present paper.

B. Routing algorithm

To find the shortest path (that exhibits the minimum delay) between connected NanoPads, the interconnection network is modeled as a dense graph G(V,E) with $E >> V$, where E is a WaferNet^{TM} segment, and V a cell. Each cell supports up to two IC balls through 16 NanoPads (Fig. 1). There are many well-known algorithms for such task [8-9]. The basic algorithm implemented as a proof of concept is based on Dijkstra's algorithm [8] that can be considered as providing a practical upper bound for the routing time, as more efficient algorithms are available [9] and will be implemented as needed in the future.

Two heuristics are proposed to manage conflicts. A conflict reflects either a detected defect in the WaferIC (on NanoPad, crossbar, segment, etc.) or several routes temporarily assigned to the same WaferNet^{TM} segment. A conflict is solved if the router finds an alternate path that meets the routing constraints. The first implemented heuristic, called *In-Order*, routes each net of the user netlist incrementally. After sorting nets by priority, set from constraints, it computes the shortest path of each net by taking into account the list of already used vertices in the graph. This behavior is not optimal, since no congestion prediction is made: it could lead to some increasing difficulties to route nets towards the end of the netlist. A second considered heuristic is slightly different: a route is computed independently for each net, assuming an ideal and fully functional WaferNet^{TM}. A conflict map is then generated and only conflicting routes are recomputed using the *In-Order* heuristic, by taking into account routing resources that are already used (or unavailable due to defects).

IV. RESULTS

A. Package recognition

The proposed tools were implemented and the package recognition tool was found able to successfully identify multiple packages on the wafer, at any orientation. The algorithm has been implemented in C++ using a standard library. By using a visualization tool, specifically written in QT/C++, one can simulate packages placement and orientation. This tool generates images of (un)connected NanoPads, which are then processed by the proposed algorithm to identify packages. As shown in figure 5, these packages typically have a regular footprint and symmetry from their center. The algorithm is able to process as many packages as the user can place on the wafer, whatever their size, as long as they are separated by more than the maximum ball pitch in the library (see section III.A). The total processing time for a set of 74 package images [15], is less than 3 seconds on a 2GHz computer.

B. Routing algorithm

A netlist generator tool has been written to benchmark the routing algorithm, where different Wafer occupation rates, net fanout and path length distribution (similar to that found in high end PCBs) can be set [13]. The two heuristics defined in the previous section have been implemented in C++. Results are shown in table 1 with four benchmarks for a 9 cm^2 segment of a WaferIC. The results for the first heuristic (*In-Order*) are highlighted in gray.

Table 1: Routing results for several WaferIC occupation rates.

Wafer Occupation Rate				**Definition**
5%	**10%**	**25%**	**50%**	
733	1445	3555	7171	Number of nets in the user netlist
906	1831	4638	9215	Number of routes
0	2	360	2395	Number of conflicts (routes)
166	452	1921	2640	
2	7	21	41	Execution time (min.)
4	10	44	72	

The current active reconfigurable circuit board design supports a maximum of two balls per cell (650 µm), providing a maximum density of 9.46 IC package balls per square millimeter. A 50% occupation rate represents one ball per cell, or several times denser than the use of today's densest FPGAs on the market fully covering the whole wafer. Even 5% occupation is equivalent to a design using nearly 1,600 pins on a 9 cm^2 board, which would represent a dense benchmark from a user standpoint : a 50% occupation rate is equivalent to 16,000 pins on the same area, which is by far more than the densest existing chip package. As routing such densities is clearly feasible, the routing network appears over-designed and possibilities to reduce cost will be studied in further work.

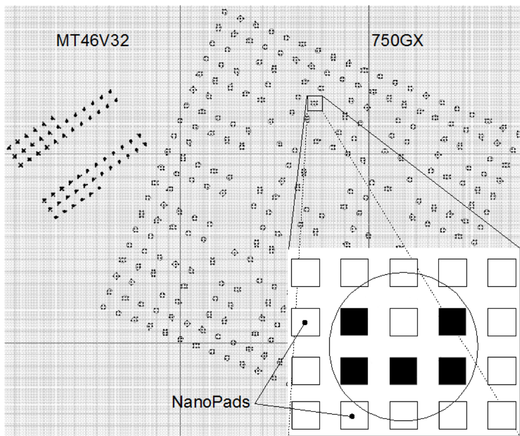

Figure 5: Results produced by the package recognition tool with a 750GX processor and a memory IC deposited on the surface.

Both heuristics have an execution time linearly proportional to the total number of routes, even with very high density netlists. This means that the interconnection network provides so many routing possibilities that nets in congested areas are still easy to route. The execution time of the second heuristic is larger than that of the *In-Order* heuristic, partly because it computes twice the paths involved in each conflicting route. Note that we keep exploring the second heuristic as it has the advantage of extracting information on routing quality by comparing final route lengths to the shortest existing path, allowing further analysis for general and local optimizations. Table 1 shows that the total number of conflicts is fairly low compared to NanoPad density: using 50% of all connection points (2 balls per cell in the current design) is equivalent to one IC ball every 0.65 mm over the whole wafer. If such density is possible locally , for example when using MSOP/µSOP packages, this density is unlikely to be observed over a full wafer.. Both implemented heuristics successfully resolve all conflicts. It is of interest that we tried routing a large number of nets over a whole 8” wafer, but it takes between a few hours to a week as a function of the density. It clearly confirms the need for improved routing methods that will be explored in future work

V. CONCLUSION

This paper proposed a workflow defined as a set of manual and automatic processes to support a novel configurable circuit board for rapid system prototyping. Two critical tools, the package recognition and the routing algorithm have been implemented and the results confirm the feasibility of this tool-chain. The IC package recognition algorithm results shows that it can identify packages deposited at any orientation and position on the wafer, for typical applications. Further work is on-going to improve the robustness of the method and to support irregular ball patterns. With respect to the second key considered step of the tool flow, the routing tool, our results show that shortest paths can be found for most routes, even when the user needs to implement very large number of connections. The proposed algorithm handles in a reasonable time the hundreds of conflicting nets produced with the densest benchmark on a 9 cm^2 portion of the configurable board. Seeking the optimal level of redundancy for routability, and for defect-tolerance will be explored as part of future work. A routing method that can process a complete wafer in an acceptable time will also be explored.

ACKNOWLEDGMENT

The authors would like to thank NSERC and Gestion TechnoCap Inc. for their financial support, and CMC Microsystems for providing design tools and access to technologies for prototyping the proposed system.

REFERENCES

[1] Swartzlander E.E. editor, “Wafer Scale Integration”, Kluwer Academic Publishers, 1989, 503 pages.

[2] Norman, R; Valorge, O.; Blaquière, Y.; Lepercq, E.; Basile-Bellavance, Y.; El-Alaoui, Y.; Prytula, R.; Savaria, Y.; “An Active Reconfigurable Circuit Board”; NEWCAS-TAISA 2008; pp 351-354; June 2008.

[3] Norman, R.; Lepercq, E.; Blaquiere, Y.; Valorge, O.; Basile-Bellavance, Y.; Prytula, R.; Savaria, Y.; “An interconnection network for a novel reconfigurable circuit board”; NEWCAS-TAISA 2008; pp 129-132; June 2008.

[4] WfMC, Workflow Management Coalition Terminology and Glossary, document number WfMC-tc-1011, document status – issue 3.0, 1999. http://www.wfmc.org.

[5] Georgakopoulos, D.; Hornick, M.; “An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure”; in Special issue on software support for work flow management; Vol. 3, Issue 2, pp 119-153; 1995.

[6] Huang, C.J.; Liao, L.M.; “An intelligent agent-based collaborative workflow for inter-enterprise PCB product design”; in Industrial Engineering and Engineering Management, 2007 IEEE International Conference on; pp 189-193; Dec. 2007.

[7] Huang, C. J.; Trappey, C. V.; Ku, C. C.; “A JADE-based Autonomous Workflow Management System for Collaborative IC Design” in The 11th International Conference on CSCW in Design, Melbourne, Australia, April 26-28, 2007.

[8] Dijkstra, E.W, “A note on two problems in connexion with graphs”; in Numerische Mathematik; Vol1, pp 269-271; 1959.

[9] Dechter, R.; Pearl, J.; “Generalized best-first search strategies and the optimality of A*”; in Journal of the ACM; Vol. 32, Issue 3, pp 505-536; 1985.

[10] ITRS Team; “Assembly & Packaging”; ITRS 2007 Edition; p 17; 2007.

[11] J, Babaud and A P Witkin and M Baudin and R O Duda; “Uniqueness of the Gaussian kernel for scale-space filtering”; in IEEE Trans. Pattern Anal. Mach. Intell.; pp26-33; 1986.

[12] Arvo, J., An Easy Bounding Circle, Jon Rokne, Graphics Gems II, Academic press, Boston, U.S.A, 1991.

[13] Roy, J.A., Lu, J.F. and Markov, I.L.; “Seeing the forest and the trees: Steiner wirelength optimization in placement” in Proceedings of the 2006 International Symposium on Physical Design; pp 78-85; 2006.

[14] Tuytelaars, T.; Mikolajczyk, K.; “Local invariant feature detectors: a survey”; Foundations and Trends in Computer Graphics and Vision, Vol. 3, Issue 3, pp 177-280, 2008.

[15] http://www.grm.polymtl.ca/~savaria/DreamWafer