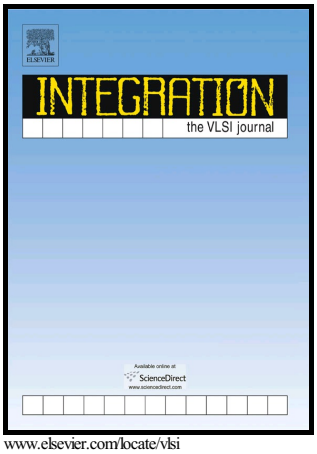


www.elsevier.com/locate/vlsi

# Synthesis of Dual Mode Logic

Lior Moyal, Itamar Levi, Adam Teman, Alexander Fish

PII: S0167-9260(16)30036-0
DOI: http://dx.doi.org/10.1016/j.vlsi.2016.07.004
Reference: VLSI1229

To appear in: *Integration, the VLSI Journal*

Received date: 16 September 2015
Revised date: 7 June 2016
Accepted date: 28 July 2016

Cite this article as: Lior Moyal, Itamar Levi, Adam Teman and Alexander Fish Synthesis of Dual Mode Logic, *Integration, the VLSI Journa* http://dx.doi.org/10.1016/j.vlsi.2016.07.004

# Synthesis of Dual Mode Logic

Lior Moyal, Itamar Levi, *Student Member, IEEE*, Adam Teman, *Member, IEEE*, and Alexander Fish, *Member, IEEE*

***Abstract*—In recent years, the major focus of VLSI design has shifted from high-speed to low-power consumption. While standard CMOS-based digital design provides substantial flexibility during pre-silicon design phases, the characteristics of the gates are set by fabrication variations and environmental conditions and cannot easily be changed at runtime. The recently proposed Dual Mode Logic (DML) family provides a novel approach to providing this capability by introducing two configurable operating modes, static and dynamic, that enable fine-grained control of the power-performance tradeoff of a logic path. However, the introduction of a new topology requires the development of both a design methodology and techniques for integration in a robust design automation flow. Standard synthesis tools do not support dynamic gates, and in particular, dual-characteristic gates. Therefore, until now, DML has been limited to small, custom-made blocks and components. In this paper, we present a novel approach for the integration of DML into standard electronic design automation tools, as part of the standard digital design flow. The development of this approach and the accompanying design methodology enables DML to be used in larger designs, such as state-of-the-art, high-speed and/or low-power SoCs. We demonstrate the employment of the proposed approach in order to benefit from DML properties, and reduce the power consumption, while simultaneously improving the operating frequency of a number of test designs.**

## I. Introduction

For many generations during the evolution of integrated circuits (ICs), the quest for high-frequency operation was at the forefront of system design. Many different approaches and techniques to accelerate the critical path were introduced and experimented with, including several digital logic families. One of the fastest and most popular topologies was the dynamic *Domino Logic* family that was carefully integrated into datapaths and other critical elements of state-of-the-art, high-performance systems. However, with the continuation of technology scaling below the micron and into the deep-nanoscale era, speed gave way to other factors, such as power and reliability, as the primary focuses of VLSI design. In the past decade, static CMOS digital logic has almost entirely displaced alternative logic families as the sole topology used in digital designs, trading-off speed for simple integration, robustness, and low power consumption.

Manuscript received September 16, 2015. Manuscript revised January 26, 2016.

L. Moyal is with the VLSI Systems Center, Ben-Gurion University of the Negev, P.O. Box 653, Beer Sheva, 84105. (phone: 972-8-647-7157; fax: 972-8-6477620; e-mail: moyall@ee.bgu.ac.il).

I. Levi, A. Teman and A. Fish are with the Emerging Nanoscaled Integrated Circuits (EnICS) Labs of the Faculty of Engineering, Bar-Ilan University, Ramat Gan (e-mail: itamarlevi@gmail.com, adam.teman@biu.ac.il, alexander.fish@biu.ac.il).

A. Teman is also with the Telecommunications Circuits Laboratory (TCL) of the Institute of Electrical Engineering, EPFL, Lausanne, VD, 1015 Switzerland.

This trend of exclusively static CMOS digital design has spanned several nano-scaled technology processes. However, the increasing ambition for power savings has led to continuous efforts to scale the supply voltage – often considered the most efficient way to reduce both static and dynamic power consumption. But as the supply voltage drops, performance is exponentially degraded, and maybe even more problematic.

In order to provide a solution to these modern challenges, *Dual Mode Logic* (DML) was recently proposed as an alternative to traditional CMOS [1], [2]. This logic family provides the ability to change the performance/power characteristics of each individual gate by controlling its mode of operation. In *dynamic* mode, a DML gate provides a speed advantage over CMOS at the expense of increased dynamic power, and in *static* mode, speed is relinquished for low power consumption. By switching between these two options at run-time, a system can adapt to the process, to the data-dependency and to the environment variations, as well as to the current power-performance requirements of the application.

However, the introduction of a new logic family does not come free of charge. Virtually all steps of the standard digital design flow (SDDF) assume that all digital components are static CMOS or compliant with the features of static CMOS, and based on this assumption, highly automated electronic design automation (EDA) tools are able to place millions of gates on a single die and quickly analyze their performance. Accordingly, any new logic family must propose solutions for integration with existing tools and design flows to accommodate smooth implementation within existing processes.

Logic synthesis, which is the process of converting a high-level hardware description language (HDL) into a gate-level netlist (GTL), is one of the primary components of the SDDF. Current synthesizers, which map behavioral register-transfer level (RTL) code to specific standard cells, only support CMOS compliant gates, featuring static output levels driven by low-resistance devices, high-resistance capacitive inputs, and asynchronous combinational logic. Dynamic logic solutions, such as DML, are characterized by different features and cannot be straightforwardly synthesized with current tools.

Several past attempts have proposed methods to integrate dynamic logic families into the SDDF [3]–[7]. Yee, et al. [3] proposed a method to synthesize dynamic logic, which along with the studies in [4] and [5], describes a CMOS based solution with complex timing races that need to be addressed. Chappel, et al. [6] introduced a system-level solution for integrating Domino Logic into the SDDF, followed by a particular solution for synthesis, proposed by Parmar [7]. However, none of these approaches consider logic, such as DML, that can function with several performance characteristics at the same operating corner, nor do they take into account any of the