1. INTRODUCTION

The industry-standard design entry for synchronous digital integrated circuits (ICs) is behavioral, where a designer describes the behavior of the circuit using standard Hardware Description Languages (HDL) like Verilog or VHDL. After logical verification, these behavioral descriptions are submitted to logical synthesis tools such as Cadence Genus or Synopsys Design Compiler (DC), along with collateral that defines a set of standard cells for the target technology. These utilities transform the behavioral description into a structural implementation of the specified circuit, consisting of various logic cells and their interconnects. The synthesized netlist is then provided to a physical implementation utility like Cadence Innovus or Synopsys IC Compiler II (ICC2) along with additional collateral for the cell library. In the physical implementation phase, the tool places the cells from the netlist, synthesizes the synchronous clock tree, and routes the interconnects between the cells while performing various optimizations. After completing signoff verification steps, the layout generated through physical implementation is ready for fabrication. These Electronic Design Automation (EDA) tools provide numerous advantages, including shortened design time, reduced effort, enhanced performance, reduced power consumption, decreased area, and greater reliability. Without the assistance of these sophisticated EDA tools, the design of today's multi-Gigahertz (GHz), multi-core digital circuits would not be possible.

This synchronous design methodology has been used for decades, yielding high-performance and low-area circuits. However, despite the ubiquitous adoption of synchronous digital circuits, they possess a distinct set of disadvantages. Since all logic operations are synchronized, the circuits suffer from high peak power demand, increased noise, and greater electro-magnetic interference (EMI) [1]. Furthermore, extensive timing analysis is required to ensure that all

operations are complete and all results are latched within the allotted clock period. Consequently, the performance of synchronous circuits is bottlenecked by the worst-case scenario of several factors including process corner, supply voltage, temperature, data pattern, routing RC variation, and aging.

Asynchronous design methodologies, which replace global clocks with local handshaking protocols, were developed to address these limitations. However, existing synchronous EDA software does not directly support the development of circuits using these asynchronous methodologies. Multi-Threshold NULL Convention Logic (MTNCL) is one such example, having historically been implemented manually without the assistance of synthesis utilities [2]. In addition, while industry-standard physical implementation tools have been used for placement and routing of MTNCL circuits, these tools have been applied crudely, failing to harness the extensive optimizations available. This dissertation research introduces the first design flow that takes synchronous Register-Transfer Level (RTL) as input and performs synthesis and physical implementation using commercial EDA tools to produce highly optimized MTNCL circuit layouts, capitalizing on advanced optimizations throughout the process.