ABSTRACT

The explosive proliferation of artificial intelligence (AI), cloud computing, and the Internet of Things (IoT) has transformed power consumption in semiconductor integrated circuits into one of the most pressing technical and environmental challenges of the 21st century. Data centers supporting large-scale AI models now rival small nations in electricity demand, with projections estimating that AI-related infrastructure could account for 8% of global power by 2030. Simultaneously, billions of IoT and wearable devices operate under stringent battery or energy-harvesting constraints, where micro-watt-level efficiency improvements directly translate to extended operational life and enhanced user experience.

Adiabatic logic, first conceptualized by Landauer and later formalized by Charles Bennett in 1973, offers a radical alternative: reversible computation that preserves information and enables energy recovery. By charging and discharging capacitive loads through resonant power-clock networks, adiabatic circuits can theoretically eliminate $CV^2 f$ dissipation during switching. Early explorations in the 1980s and 1990s demonstrated logical reversibility in CMOS, but practical energy recovery remained elusive due to the absence of efficient resonant mechanisms and significant circuit overhead.

Among adiabatic families, *Two-Level Adiabatic Logic (2LAL)* stands out for its near-ideal energy recovery, robust noise margins, and compatibility with standard CMOS processes. Employing dual-rail encoding, transmission gates (T-gates), and four-phase power clocks, 2LAL achieves charge recycling with minimal non-adiabatic losses. Recent industry breakthroughs, such as Vaire Computing's Ice River test chip in 2025, have demonstrated net energy recovery with a 1.77× efficiency factor over conventional circuits in 22 nm CMOS, validating resonant adiabatic switching in silicon for the first time.

Even though 2LAL has an excellent potential in power dissipation, in its fully pipelined gate-level configurations, mandatory "decompute" operations for charge recovery cause severe buffer overhead. Input and output signals of each logic gate are buffered across pipeline stages. When the role of an output signal is completed, more specifically, after driving the decomputes of all fanout gates of the output signal, its replica is regenerated from the input and placed it in opposition to the target output signal for returning charge to the power supply. This decompute must be performed for every gate, leading to an explosive increase in buffer count and impractical circuit area.

This thesis aims to systematically reduce this buffer overhead by determining and optimizing which logic gates should receive early decompute and at what timing it should be applied. Early decompute involves decomputing a gate output immediately after computing the subsequent output and recomputing it when the subsequent output is decomputed. Although this doubles decompute operations per gate, it eliminates buffers between the first decompute and recompute phases, enabling substantial area reduction. While prior work proposed only simple heuristics for this concept, this research focuses on the structural properties of adiabatic circuits and formulates the problem of selecting early decompute targets and their timing as an integer linear programming (ILP) task from three distinct approaches:

1. Formulating early decompute insertion as a pipeline stage assignment problem under fixed scheduling to achieve optimization.
2. Reformulating early decompute target selection as a weighted maximum stable set problem on an Extended And-Inverter Graph (E-AIG), significantly reducing the search space in terms of total pipeline stages and candidate gate count, thereby shortening computation time.
3. Extending the first approach to jointly optimize early decompute timing and logic gate pipeline stage assignment, enhancing scheduling flexibility.

These methods were applied to all 11 circuits in the ISCAS-85 benchmark suite, a standard for combinational logic synthesis, and evaluated using the area efficiency metric $E_{\text{area}} = M_{\text{2LAL}}/M_{\text{CMOS}}$. Results showed up to 79.1% area reduction compared to 2LAL circuits without any early decompute, and up to 55.9% improvement over circuits obtained by prior heuristic methods. Notably, the second method achieved optimal solution extraction for all 11 circuits in under one second. This enabled analysis of the trade-off between area performance and computation time, clarifying the applicable scope of each method.

A rigorous energy-area-leakage trade-off analysis, validated through nanometer-scale LTspice simulations across 45 nm, 35 nm, and 22 nm nodes, confirmed that optimized 2LAL maintains

orders-of-magnitude energy superiority over CMOS even with residual overhead, with advantages strengthening in leakage-dominated advanced nodes due to exponentially rising static power. Tolerable area overheads reach 500×–1200× at practical frequencies, providing vast headroom for deployment.

Additionally, the appendix proposes two LC-resonant power-clock generators (2N2P-1 and 2N2P-2) for trapezoidal waveforms. Design equations and theoretical power consumption formulas were derived for each, and characteristics were evaluated and compared via LTspice simulation. This provides a theoretical guideline for selecting the optimal power-clock configuration from a power dissipation perspective, establishing a foundation for designing adiabatic logic circuits as a complete system encompassing both logic and power components.

This research establishes a comprehensive design automation framework for practical synthesis of Two-Level Adiabatic Logic (2LAL), simultaneously achieving circuit area optimization, computational efficiency, power system integration, and verified energy superiority in modern technology nodes—paving the way for deployment in ultra-low-power IoT, edge AI, implantable devices, and energy-harvesting systems.

Keywords: adiabatic logic, reversible computing, Two-Level Adiabatic Logic (2LAL), early decompute scheduling, buffer minimization, integer linear programming, stable set problem, pipeline rescheduling, energy-area trade-off, low-power VLSI, sustainable computing