Evolutionary Placement and Diffusion Sharing for a Standard-Cell D Flip-Flop

k34ding

***Abstract*—We present a lightweight evolutionary-algorithm (EA) approach that generates standard-cell style placements with diffusion sharing for small custom cells. The method operates directly on transistor-level connectivity derived from a SPICE netlist and outputs integer x-positions, per-device flip bits, and contiguous sharing segments (derived). We demonstrate feasibility on a positive-edge triggered D flip-flop and visualize the result with a stick diagram. Under time constraints, we focus on the DFF and show a valid placement with multiple sharing segments and a compact diffusion arrangement.**

I. Introduction

Compact device placement with diffusion sharing is a key step in custom standard-cell design. Traditional formulations often require handcrafted templates or complex combinatorial optimization with many constraints. We investigate a pragmatic alternative: an EA that searches over discrete placements and per-device flips, guided by feasibility checks and a simple objective structure. Our goal is to *automatically* obtain a legal placement that exhibits significant diffusion sharing.

II. Method

A. Input Representation

We parse a SPICE netlist to obtain a vector of devices $V = \{v_i\}$, each labeled $\{\text{PMOS, NMOS}\}$, and a set E that maps each device to its (drain, gate, source) nets. Standard-cell assumptions allow a single row for PMOS and a single row for NMOS; only x-positions (integers) and flips are optimized.

B. Genome

The genome encodes: (i) PMOS x-positions, (ii) NMOS x-positions, and (iii) flip bits per device (0: drain-left/source-right, 1: source-left/drain-right). Groups are *not* explicitly encoded; instead, we *derive* contiguous sharing segments from (V, E) plus (placement, flip).

C. Feasibility and Objectives

Feasibility requires unique x-positions within each row. Given (placement, flip), we sort by x and scan adjacent devices to form sharing segments when the right net of the left device equals the left net of the right device. We compute:

- **Share count** S: number of shared adjacencies (higher is better).
- **Span** L: total horizontal span (lower is better).

We use a bi-objective formulation: maximize S and minimize L. In practice, we minimize $(-S, L)$.

D. Evolutionary Algorithm

We use NSGA-II with integer-compatible sampling/mutation and a simple *repair* that (a) fixes duplicate positions by assigning unused slots, and (b) clamps flip bits to $\{0, 1\}$. This steers the search toward feasible regions without rejecting large portions of the population.

E. Visualization

A stick-diagram renderer summarizes results: diffusion strips (contiguous, net-consistent), poly sticks with gate labels, and interface tags (`SHARE`/`break`/`gap`). This provides a quick visual sanity check of sharing quality and placement compactness.

III. Results

We evaluated the approach on a positive-edge triggered D flip-flop. The parser identified 13 PMOS and 13 NMOS devices. The EA produced a legal integer placement with multiple sharing segments. A representative run yielded:

- **PMOS**: 5 breaks (7 shares),
- **NMOS**: 7 breaks (5 shares),
- **Total**: 12 breaks across both rows.

The x-positions were unique and formed several disjoint sharing chains, consistent with the derived-group report. The stick diagram confirms contiguous diffusion strips and correctly labeled gate polys.

IV. Discussion and Future Work

This EA-based method successfully generated a feasible, share-rich placement for the DFF under standard-cell assumptions. The derived-group strategy avoided explicitly modeling groups and simplified the genome. Given time constraints, experiments were limited to the DFF; **future work** includes:

- applying the method to other cells (e.g., half/full adders, AOI/OAI),
- incorporating wirelength and pin-access into the objectives,
- exploring ILP or matching-based post-processing to further improve sharing,
- extending the visualizer with contacts and metal routing sketches.

V. Conclusion

An EA with a repair-and-derive design can produce legal placements with meaningful diffusion sharing from only device connectivity and simple objectives. Our DFF study demonstrates practicality and provides a foundation for broader cell libraries.