PnR and STA Flow for a Structured ASIC Platform

Phases 1&2 Checkpoint Presentation

Mazin Bersy
Malek Mahmoud
Mostafa Elshamy

Phase 1: Database, Validation, & Visualization

Database

```
# Merge into unified DB
fabric_db = {
    "fabric": {
        "cells_by_tile": enriched_cells_by_tile,
        "pin_placement": pin_placement,
        "site_dimensions_um": site_dimensions
    }
}
```

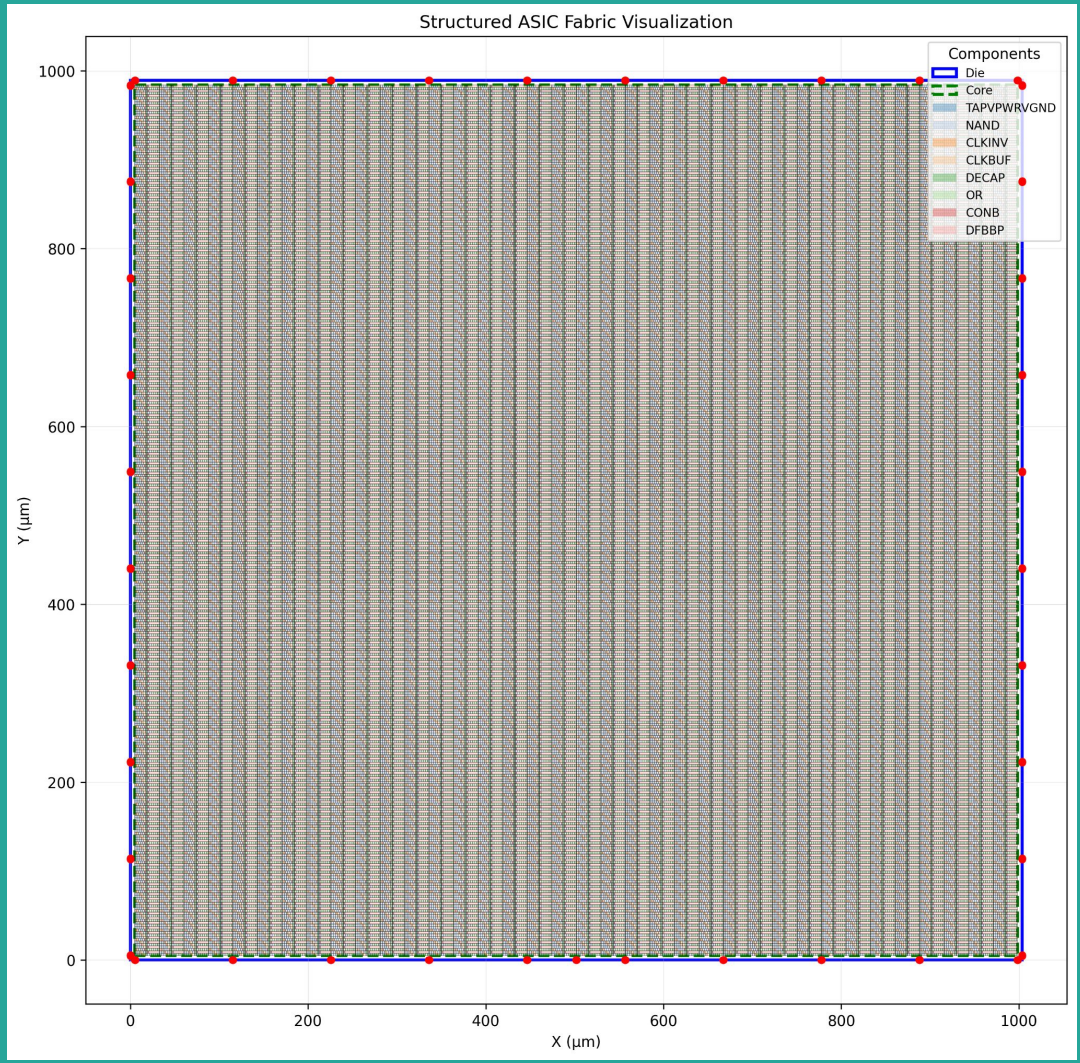
```
logical_db = {
    "cells": instances,
    "cells_by_type": dict(instances_by_type),
    "nets": nets,
    "ports": ports,
    "meta": {
        "top_module": top_name,
        "source_file": os.path.basename(json_path),
        "multi_bit_warnings": multi_bit_warnings,
    },
}
```

```
def _build_netlist_graph(logical_db: Dict[str, Any]) -> nx.Graph:
    """
    Build an undirected graph of the netlist:
      • Each instance and port is a node
      • Each shared net creates edges between all connected nodes
    """
    G = nx.Graph()

    # Add instance nodes
    for inst_name, inst_info in logical_db["cells"].items():
        G.add_node(inst_name, type=inst_info["type"], node_type="cell")

    # Add port nodes
    for pd in ("inputs", "outputs"):
        for pname in logical_db["ports"].get(pd, {}):
            G.add_node(pname, type="PORT", node_type="port", direction=pd)

    # Add edges
    for net_id, net_info in logical_db["nets"].items():
        endpoints = net_info.get("connections", [])
        node_list = [n for (n, _) in endpoints]
        if len(node_list) <= 1:
            continue
        for i in range(len(node_list)):
            for j in range(i + 1, len(node_list)):
                u, v = node_list[i], node_list[j]
                if not G.has_edge(u, v):
                    G.add_edge(u, v, nets=[net_info["name"]], net_id=net_id)
                else:
                    G[u][v]["nets"].append(net_info["name"])
    return G
```

Validation

```
================================================================================
FABRIC VALIDATION REPORT
================================================================================
Design: 6502_mapped.json
Top Module: sasic_top

Design Statistics:
  Total Cells: 2899
  Total Nets: 3086
  Input Ports: 42
  Output Ports: 80

Fabric Utilization by Cell Type:
--------------------------------------------------------------------------------
Cell Type            Required     Available    Usage            Status
--------------------------------------------------------------------------------
BUF                  322          9720         3.31%            OK
CONB                 3            9720         0.03%            OK
DFBBP                143          6480         2.21%            OK
INV                  360          12960        2.78%            OK
NAND                 1228         48600        2.53%            OK
OR                   843          25920        3.25%            OK
--------------------------------------------------------------------------------

Overall Fabric Utilization: 2899/113400 (2.56%)
```

Visualization



Phase 2 - Placement (Assignment) & Analysis

Greedy Placement

Stage 1: Fixed I/O Placement

- Directly place input/output ports using pre-defined pin coordinates.
- Establishes anchor points for early connectivity.

Stage 2: SEED Placement (Pin-Connected Cells)

- Identify all cells directly connected to I/O pins.
- Place them first using **barycenter of pin locations**.
- Ensures logic close to external interfaces is well-localized.

Greedy Slot Assignment

- For every cell to place, choose the **nearest free physical slot** to the target position.
- Minimizes immediate wirelength.

Stage 3: GROW Phase (Most-Connected-First)

- Iteratively select the **unplaced cell with the highest number of already-placed neighbors**.
- Compute barycenter of those neighbors → guides placement toward existing clusters.
- Greedy expansion of connected components across the fabric.

HPWL Calculation

Measures total estimated wire length needed to connect all cells
Computed per-net as follows:

- Identify all placed cells on the net
- Draw the smallest bounding box around them
- HPWL = (box width + box height)

HPWL Example

- Net connects 3 cells at positions:
  - A: (2, 3)
  - B: (5, 7)
  - C: (8, 4)
- Bounding box:
  - Width = 8 − 2 = 6
  - Height = 7 − 3 = 4
- HPWL = 6 + 4 = 10 units

SA Optimizer

Purpose

- Enhance greedy placement
- Reduce total wirelength (HPWL)

Key Concepts

- Geometric Cooling
  – Temperature decays each step
- Move Types
  – REFINE: swap two cells
  – EXPLORE: shift cell to new slot (shrinking window)

Acceptance Rule

- Always accept improvements
- Accept worse moves with probability exp(–Δ / T)

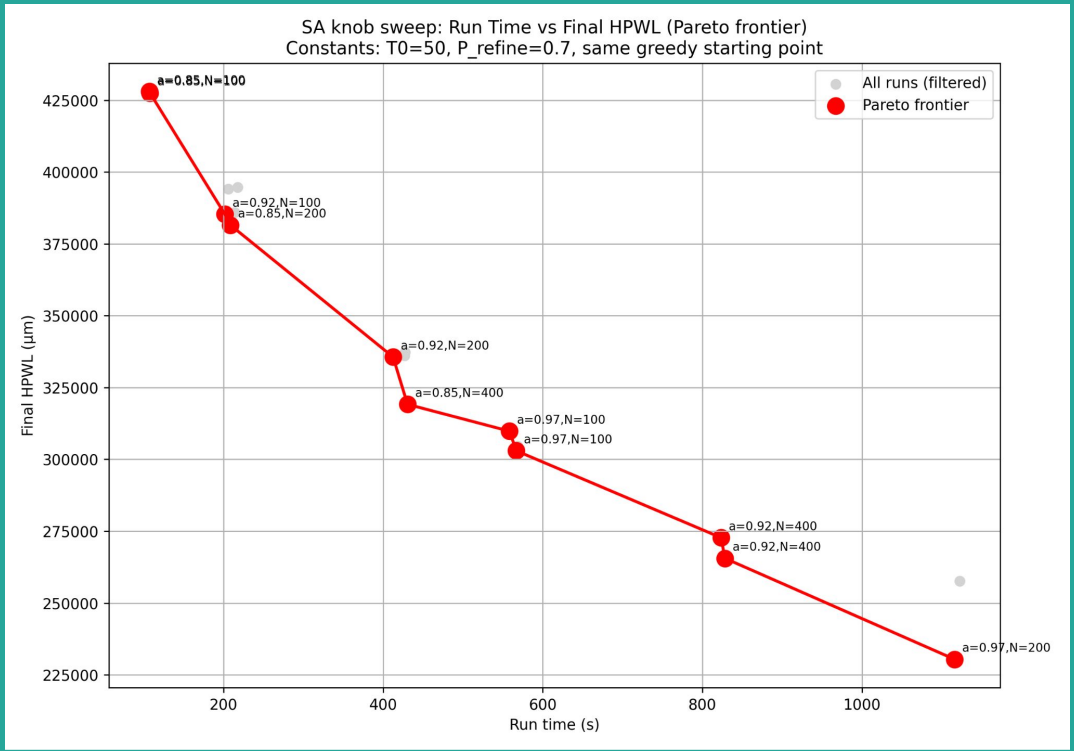
```
============================================================
OPTIMIZATION COMPLETE
============================================================
Initial HPWL:     466403.70 µm
Final HPWL:       214716.20 µm
Improvement:      53.96%
Total Iterations: 112000
Accepted Moves:   19278 (17.2%)
Rejected Moves:   92722 (82.8%)
REFINE moves:     78238
EXPLORE moves:    33762
Improvements:     9954
============================================================

[RUN] cfg=9 alpha=0.97 N=400 T0=50.0 pref_refine=0.7 seed=2
```

SA Parameters

- **Initial Temperature** – The starting temperature
- **Final Temperature** – Sets the convergence point
- **Cooling Rate** – Determines how quickly temperature decreases between steps.
- **Moves per Temperature** – Number of candidate moves evaluated at each temperature level.
- **Maximum Iterations** – Safety cap preventing the algorithm from running indefinitely.
- **REFINE Move Probability** – Probability of performing a local swap between two cells.
- **EXPLORE Move Probability** – Probability of shifting a cell to a different location for global exploration.
- **Initial Explore Window** – Limits how far EXPLORE moves can relocate a cell, shrinking as annealing progresses.

Analysis

Round 1

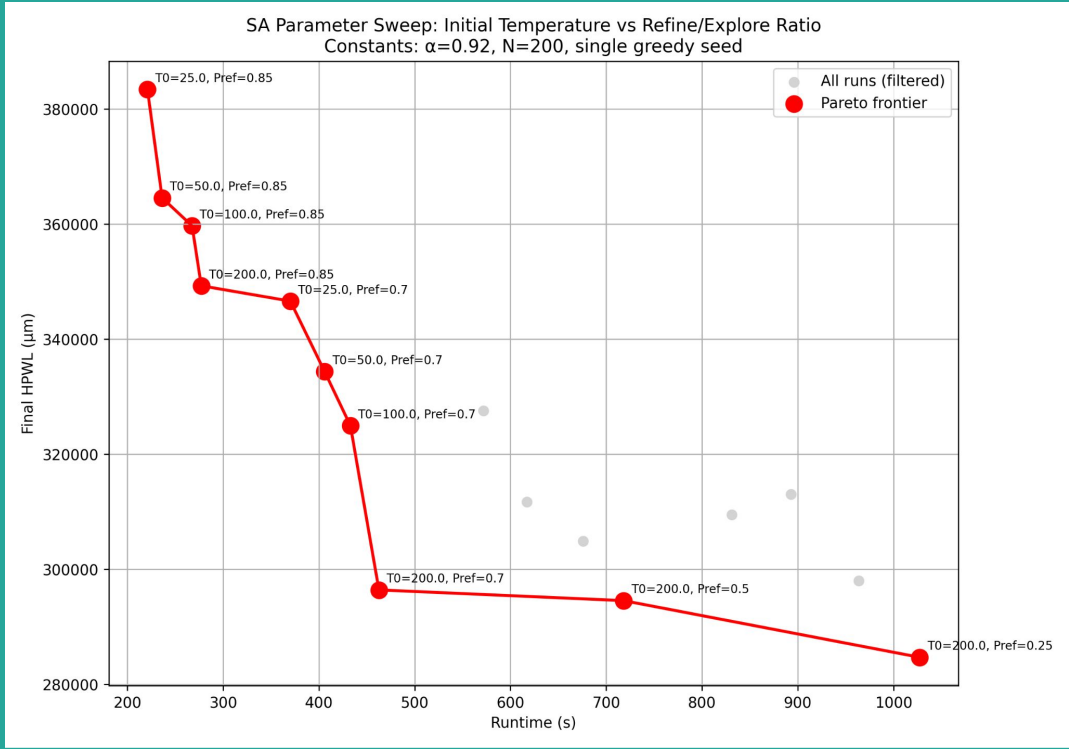


Testing:
Cooling rate
Moves per temperature

Conducted sweep of values, Repeated each experiment 3 times to ensure consistency

Developed pareto curve
Optimal Values:
$\alpha = 0.92$, $N = 100$

Round 2



Testing:
Initial Temperature
Refine Probability

Conducted sweep of values, Repeated each experiment 3 times to ensure consistency

Developed pareto curve

Optimal value:
Initial Temperature (T_0) = 200
Refine Probability = 0.50

Visualization - After Optimization