

# Hotspot Mitigation through Multi-Row Thermal-aware Re-Placement of Logic Cells based on High-Level Synthesis Scheduling

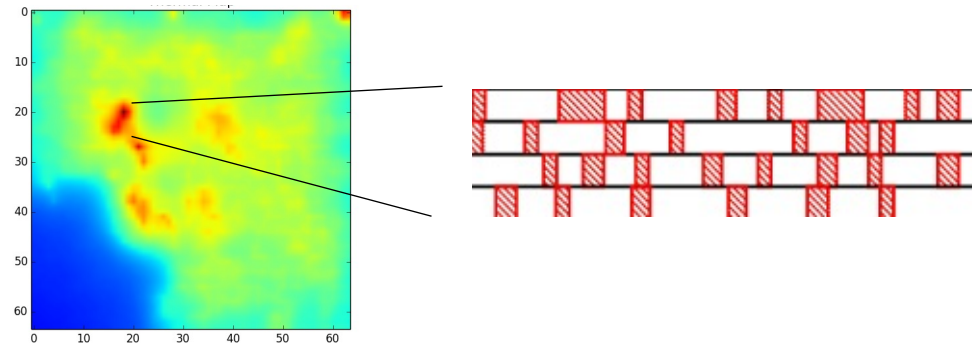
Benjamin Carrion Schafer

Assistant Professor

[schaferb@utallas.edu](mailto:schaferb@utallas.edu)

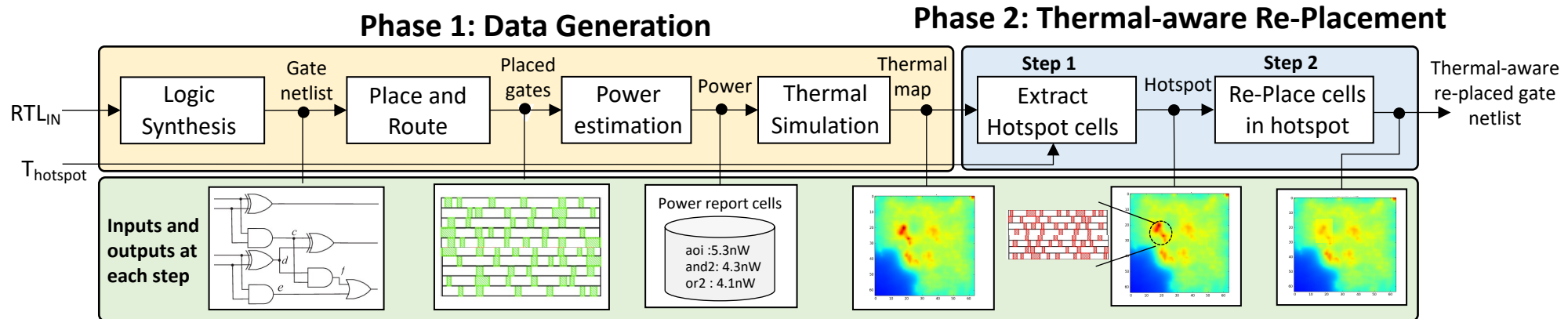
# Introduction

- Place and route tools do only consider area and timing when placing a synthesized netlist
- This can lead to a placement with high-density power regions, which in turn lead to hotspots



- ➔ This work presents a method to re-place logic cells locally, within the hotspot, to reduce the peak temperatures, leveraging the fact that placed rows contain fillers between cells
- ➔ Approach is based on Linear Programming also used in HLS scheduling

# Overview of Proposed Flow



## Inputs:

- RTL code ( $RTL_{in}$ )
- Workload (Test vectors)
- Maximum allowable temperature ( $T_{hotspot}$ )
- Technology library and synthesis constraints

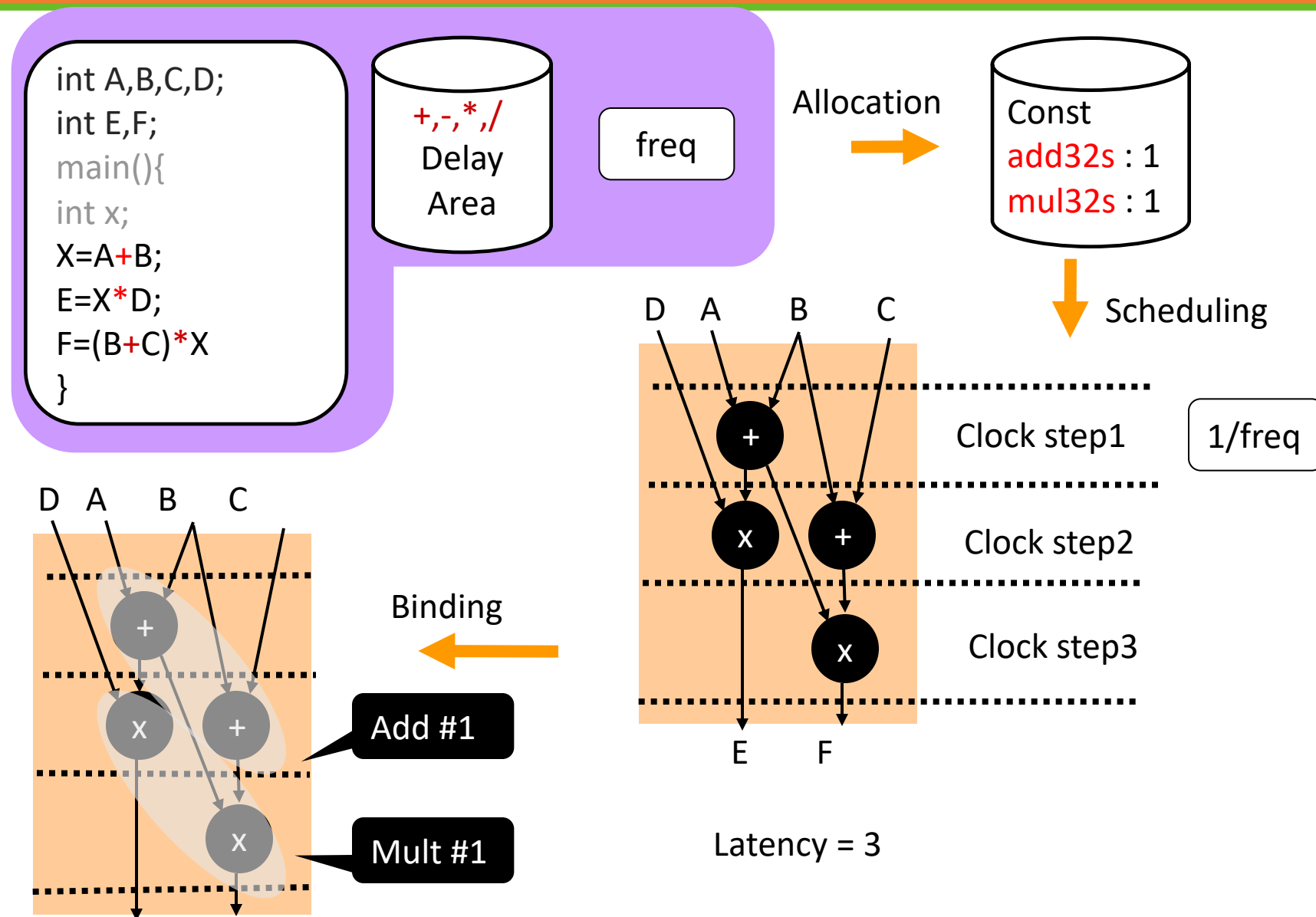
## Output:

- Thermal-aware re-place netlist

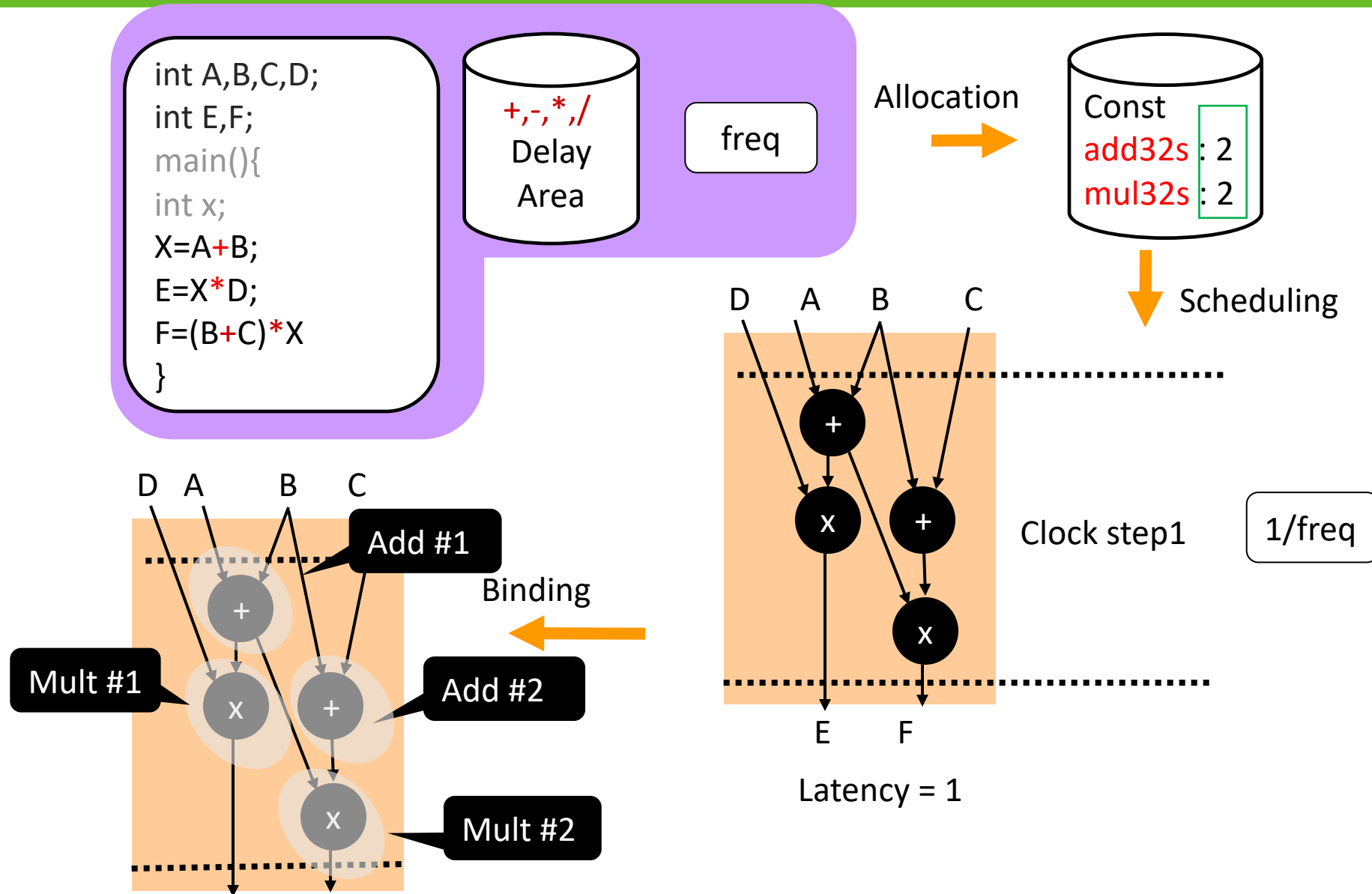
**Phase I – Data Generation:** Generates the thermal map of given synthesizable description ( $RTL_{in}$ ) given in either Verilog or VHDL

**Phase II – Thermal-aware Re-placement:** Replace logic cells to reduce peak temperature using HLS scheduling LP programming method called: System of Difference Constraints (SDC)

# High Level Synthesis Overview



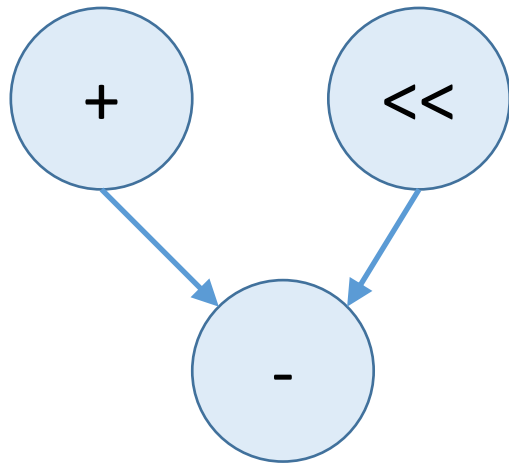
# High Level Synthesis Overview cont.



# One Popular Way of Scheduling: “SDC Scheduling”

- SDC  $\leftrightarrow$  System of Difference Constraints
  - Cong, Zhang, “An efficient and versatile scheduling algorithm based on SDC formulation”, DAC 2006: 433-438.
- Basic idea: formulate scheduling as a mathematical optimization problem
  - Linear objective function + linear constraints  
( $=$ ,  $\leq$ ,  $\geq$ )
- The problem is a linear program (LP)
  - Solvable in polynomial time with standard solvers

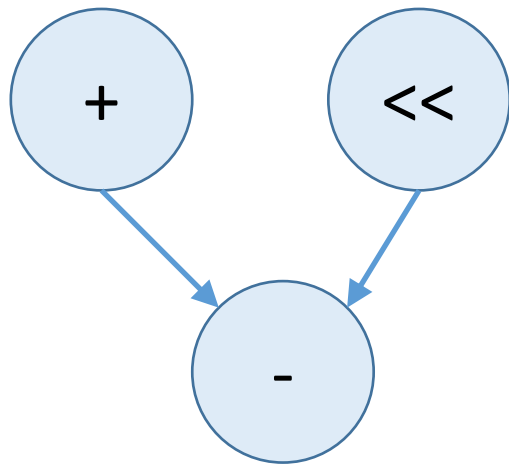
# Define Variables



Data flow graph (DFG)

- For each operation  $i$  to schedule, create a variable  $x_i$
- The  $x_i$ 's will hold the cycle # in which each op is scheduled
- Here we have:
  - $x_{\text{add}}$ ,  $x_{\text{shift}}$ ,  $x_{\text{sub}}$

# Dependency Constraints



- In this example, the subtract can only happen after the add and shift

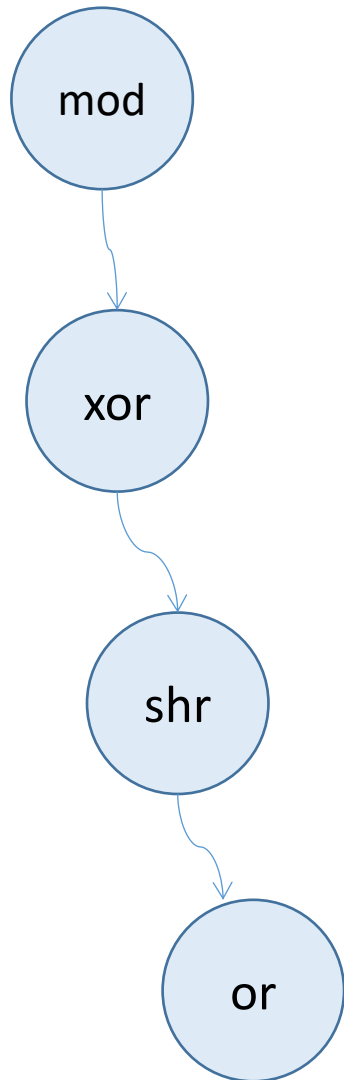
$$X_{\text{sub}} - X_{\text{add}} \geq 0$$

$$X_{\text{sub}} - X_{\text{shift}} \geq 0$$

- Hence the name *difference constraints*



# Handling Clock Period Constraints



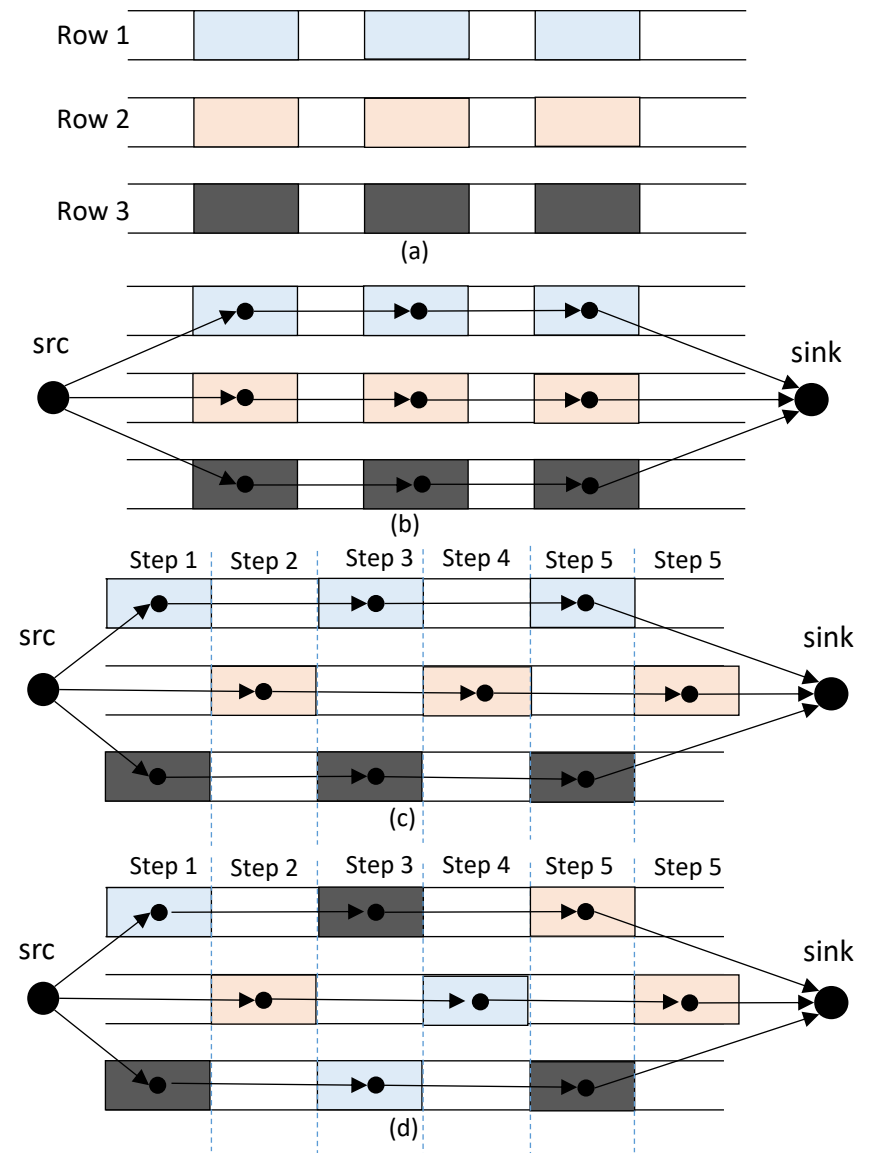
- Target period:  $P$  (e.g., 10 ns)
- For each chain of dependant operations in DFG, find the path delay  $D$ 
  - E.g.:  $D$  from  $\text{mod} \rightarrow \text{or} = 23$  ns.
- Compute:  $R = \text{ceiling}(D/P) - 1$ 
  - E.g.:  $R = 2$
- Add the *difference constraint*:
  - $X_{\text{or}} - X_{\text{mod}} \geq 2$

# Example of Multi-row Logic cell Scheduling

- Each placed cell is mapped to a node in the DFG
- A DFG is generated by connecting the different cells
- Two approaches are investigated

**Approach 1:** cells are only moved within its own row

**Approach 2:** cells can move to neighboring rows

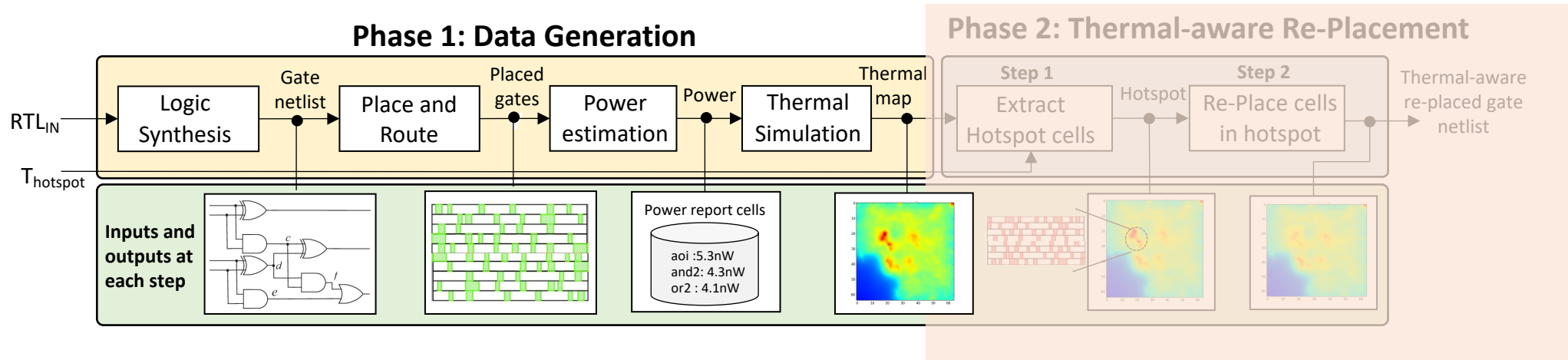


# HLS vs. Thermal-aware Re-placement Equivalence

- Number of resources = cell density

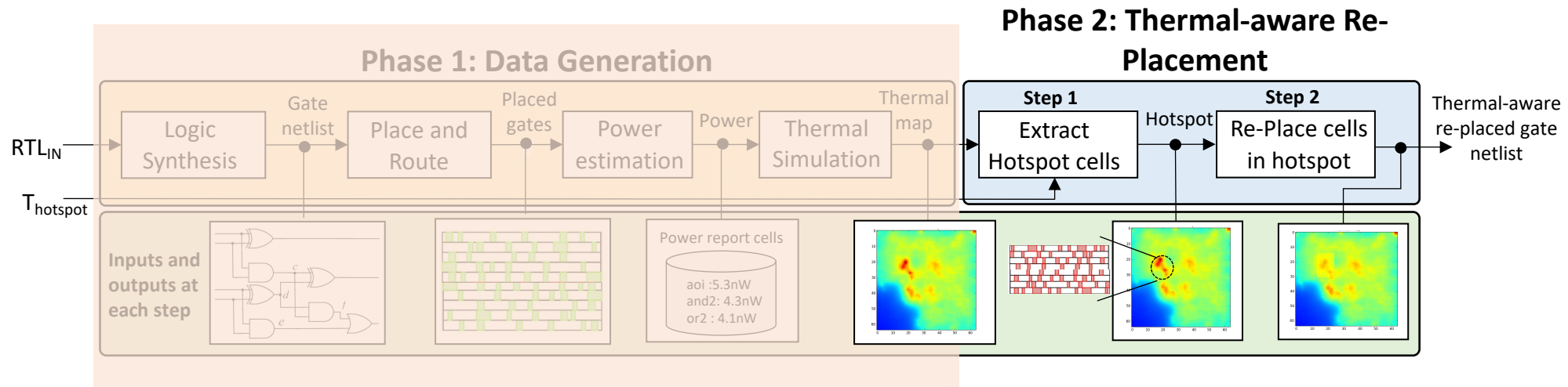
High-Level Synthesis	Thermal-aware Cell Re-placement
Functional Units Delay	Cell power
# Resources	Area cells/Area total x n rows
HLS frequency	Length row/Power row x Power cell

# Detailed Proposed Flow – Phase 1

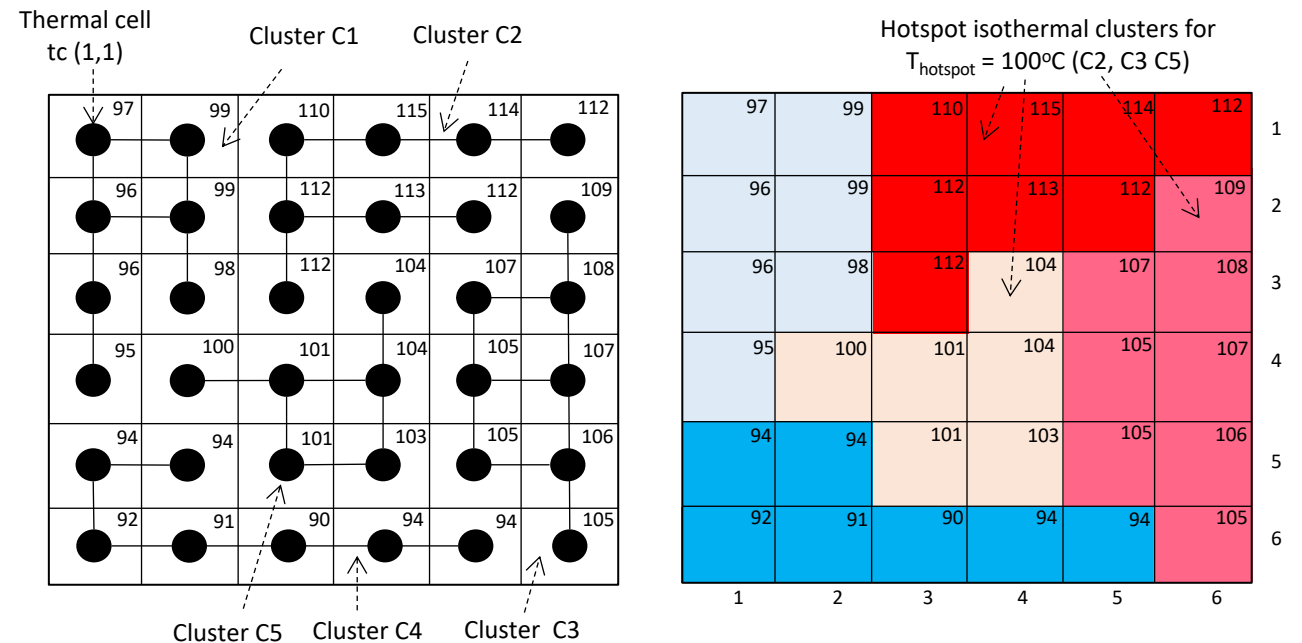


- Step 1: Logic synthesis of  $RTL_{in}$
- Step 2: Place and route gate netlist
- Step 3: Power estimation
- Step 4: Thermal simulation

# Detailed Proposed Flow – Phase 2



- Step 1: Extract cells in hotspot (build isothermal clusters and extract cells over  $T_{hotspot}$ )
- Step 2: Re-place cells in hotspot to reduce peak temperature formulating problem as SDC constraint scheduling



# Experimental Setup

## Tools

- Logic Synthesis tool: Synopsys Design Compiler v.0-2016.02-SP3
- Placement tool: Cadence Innovus
- Power estimator: Synopsys Primetime 2016.12-SP3
- Thermal simulator: Hotspot 6.0
- Target technology: Nangate Opencell 45nm
- Solver : Ip\_solve 5.5.2.0

## Platform

- Computer platform
  - Intel(R) Xeon E7 with 16GByte of RAM
  - CentOS Linux release 7.8.2003 (Core)

## Evaluation

- Synthetic benchmark generator of different logic densities
  - Low-density (60%), medium-density (75%) and high-density (90%)
  - AES example
- Proposed two methods : Move cells within its own row and across rows
- Compared against a previously developed method that optimizes row by row\*

\*J. Song, Y. Lee, and C. Ho. 2016. **ThermPL: Thermal-aware placement based on thermal contribution and locality**. In VLSI-DAT. 1–4.

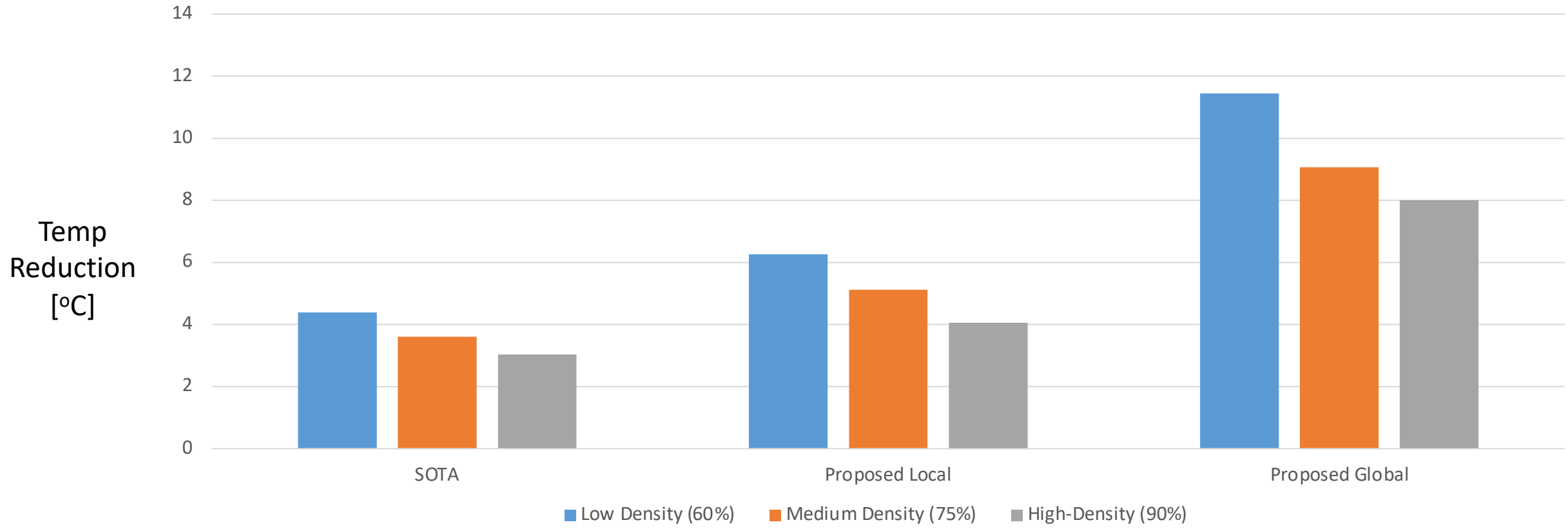
# Experimental Result – Overhead Analysis

Low-Density (60%)

Bench	#Cells	Original	Single row based[14]			Proposed local			Proposed global		
		Temp [°C]	Temp[°C]	Delay[%]	Run [s]	Temp[°C]	Delay[%]	Run [s]	Temp[°C]	Delay[%]	Run [s]
small1	400	60.01	55.2	0.04	74.2	55.17	0.06	110.36	50.66	0.13	130.79
small2	500	62.43	57.5	0.03	95.1	57.88	0.04	109.57	51.65	0.11	156.6
medium1	2,500	64.53	61.5	0.06	39.23	55.17	0.07	59.99	52.55	0.22	46.74
medium2	3,600	65.42	59.45	0.04	38.91	58.94	0.05	58.77	52.23	0.18	52.69
large1	10,000	67.36	62.40	0.05	40.95	62.45	0.08	57.64	55.4	0.22	58.24
large2	12,100	68.12	65.51	0.06	41.17	60.7	0.05	54.42	56.7	0.24	70.93
<b>Geomean</b>					<b>51.27</b>			<b>71.51</b>			<b>77.00</b>
<b>Avg.</b>		<b>64.65</b>	<b>60.26</b>	<b>0.05</b>		<b>58.39</b>	<b>0.06</b>		<b>53.20</b>	<b>0.18</b>	

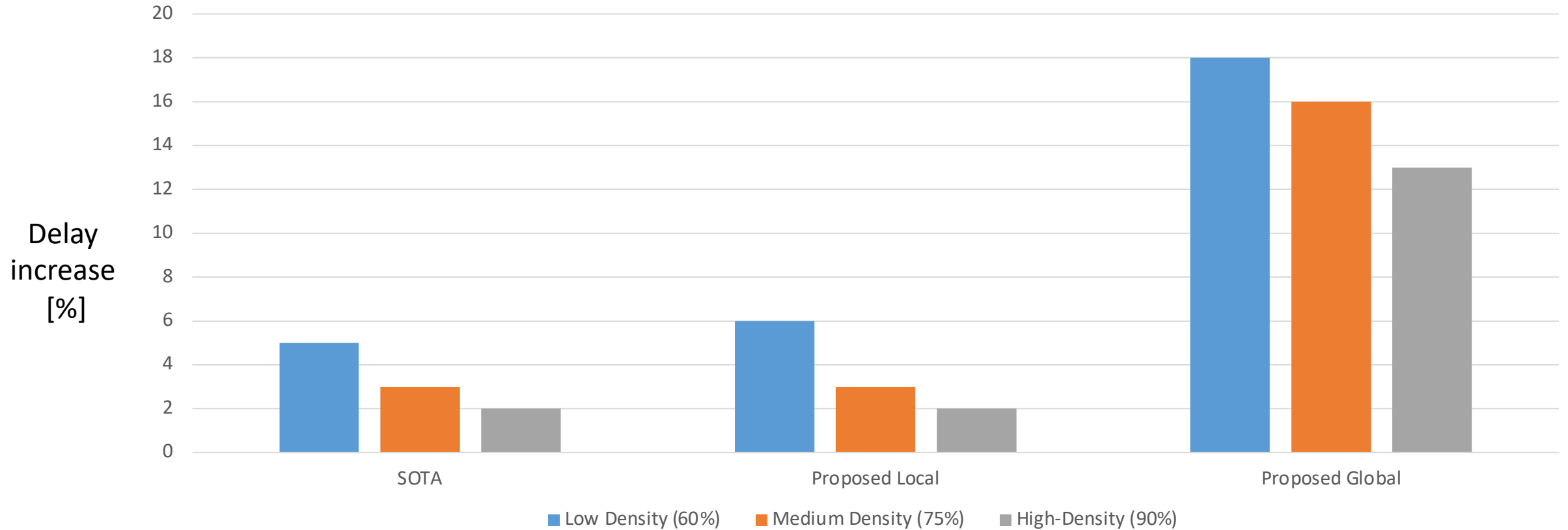
- **Observations:**
  - Our proposed technique works better with lower logic densities as it has more *room* to move the cells apart.
- **Temperature reduction:**
  - On average the temperature by 6.26 5.12 and 4.06°C for the low, medium and high-density cases for the local placement method (same rows)
  - On average the temperature is reduced by and 11.45, 9.06 and 8.01°C for the global re-placement method
  - Compared to the state of the art, on average across all three densities, our proposed method was able to further reduce the temperature by a factor of 1.6x and 2.9x compared the local and global optimization method respectively
- **Delay increase:**
  - local optimization method delay increase by 6%, 3% and 2% for each of the logic density scenario
  - global optimization approach it increases, as expected, to an average of 18%, 16% and 13%.

# Experimental Results : Logic Density vs. Temp





# Experimental Results : Delay vs. Temperature



# Experimental Result – AES Example

<b>Orig.</b>	<b>Single row [14]</b>		<b>Proposed local</b>		<b>Proposed global</b>	
Temp °C	Temp [°C]	Delay [%]	Temp [°C]	Delay [%]	Temp [°C]	Delay [%]
59.23	56.23	0.01	54.12	0.03	51.65	0.07

- Proposed local re-placement flow leads to ~5°C lower temperature and 2 °C less than SOTA
- Proposed Global re-placement : ~7.5 °C lower temperature and ~5 °C less than SOTA

# Summary

- We have presented a method to reduce the temperature of hotspots in placed and routed netlists
- Formulated the problem as a System of Difference Constraints (SDC) first introduced in the context of High-Level Synthesis (HLS) scheduling
- Results show the effectiveness of our proposed flow compared to the SOTA

Thank You