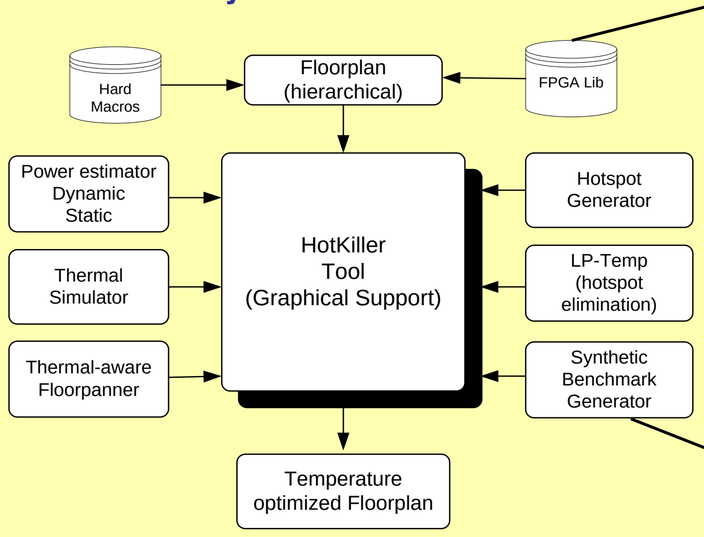
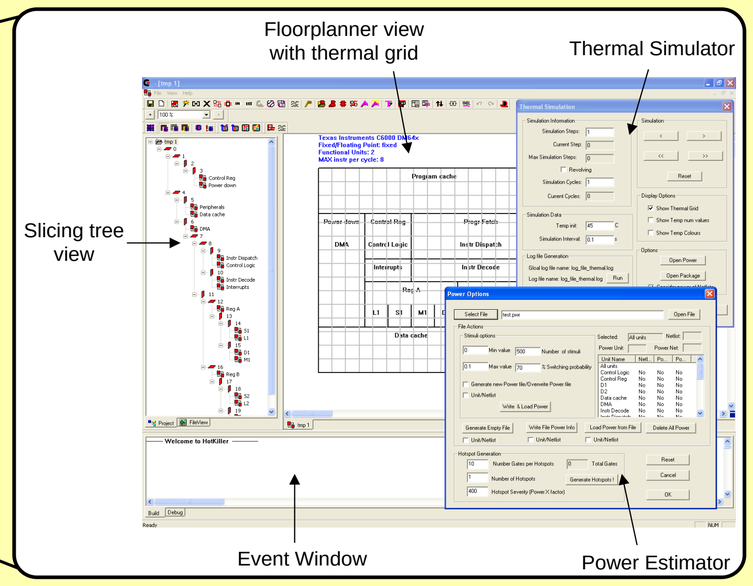
# Temperature Reduction and Control Techniques for VLSI circuits

Dr. Benjamin Carrión Schäfer and Prof. Taewhan Kim
System Synthesis Laboratory http://ssl.snu.ac.kr (02-880-9134)
School Of Electrical Engineering and Computer Science , Seoul National University

## Hotkiller Project:

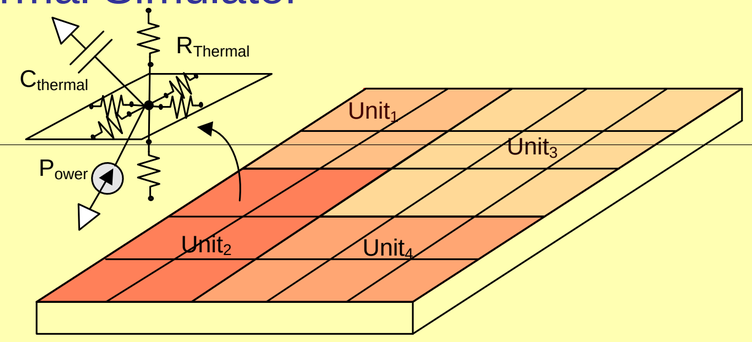



## Objectives:

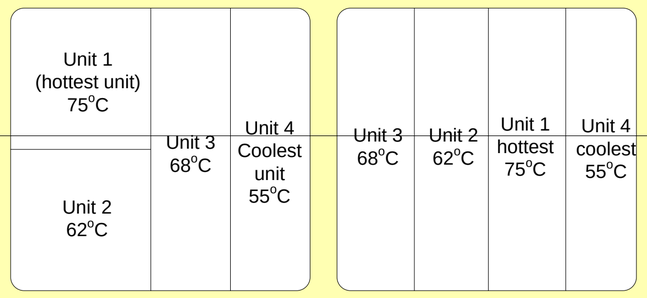
- Study the thermal behavior of VLSI circuits
- Eliminate Hotspots
- Flatten overall temperature
- Reduce leakage power

## Thermal Simulator




- Based on the duality between electricity and thermal flow ($C_{thermal}$ models transient behavior, $R_{thermal}$ the heat flow)
- Thermal mesh of different sizes is generated on top (Finer mesh yields more precise results; Coarser mesh faster simulation speeds)
- Power profile of each unit passed. Temperature of each cell is updated every time step

## Thermal-aware Floorplanner



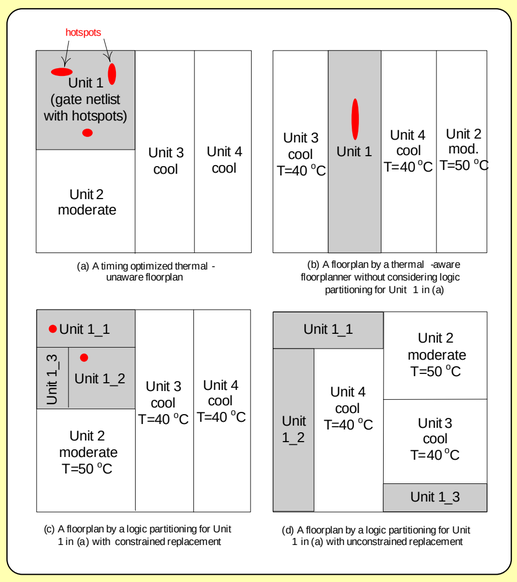

(a) Initial floorplan (b) Final floorplan

- Slicing floorplanner
- Simulated Annealing
- COST=$\alpha A+\beta W-\gamma D$ (A: Area; W: Wirelenght; D Thermal diffusion) Thermal diffusion wants to be maximized.

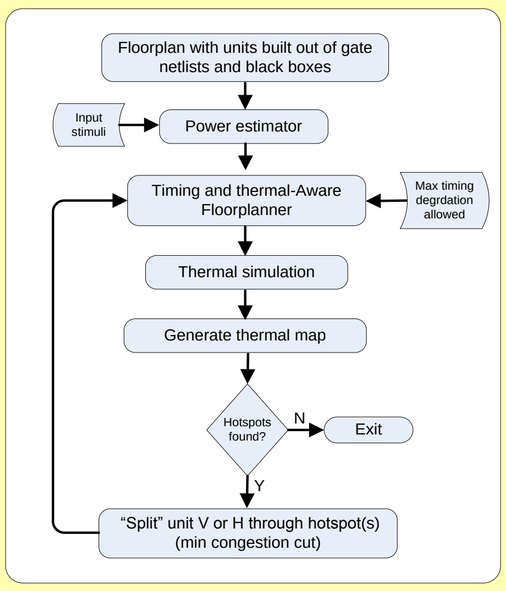
# Temperature Reduction Techniques

## Logic Partitioning Technique

### Motivation:

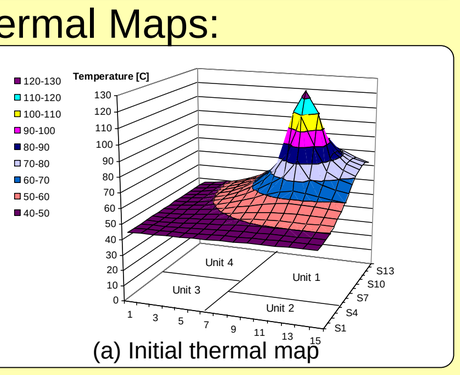



### Flow Graph:

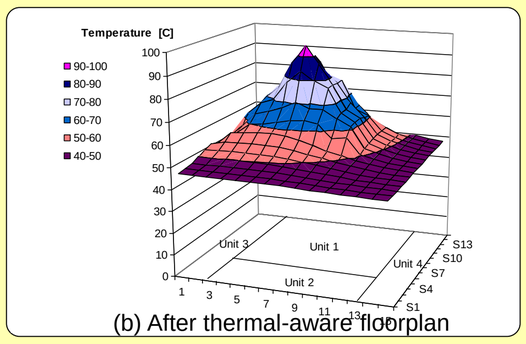



- Place netlist in floorplan minimizing delay
- Start performing thermal simulation to obtain the initial thermal map
- Build isothermal logic clusters
- Identify the hottest cluster and partition logic netlist thorough its hottest point (V or H; min congestion)
- Re-place the new floorplan using thermal-aware floorplanner allowing x degree of timing degradation

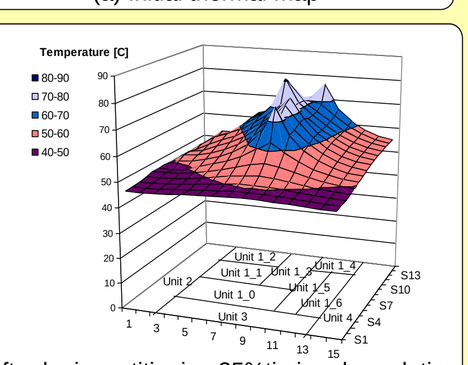
### Thermal Maps:



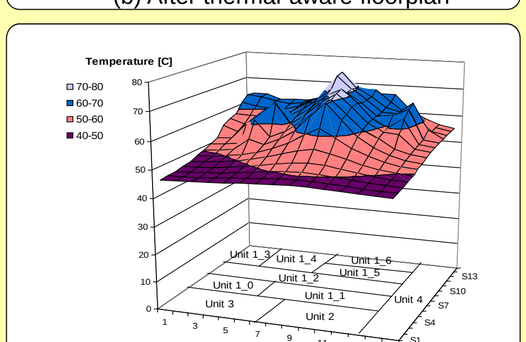

(a) Initial thermal map




(b) After thermal-aware floorplan

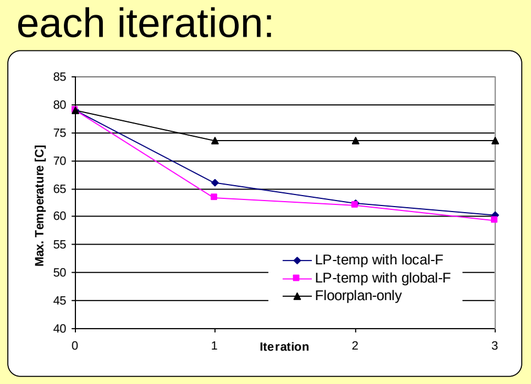



(c) After logic partitioning 25%timing degradation




(d) After logic partitioning 50%timing degradation

### Peak Temp reduction on each iteration:




## Thermal-aware Instruction Assignment for VLIW Processors

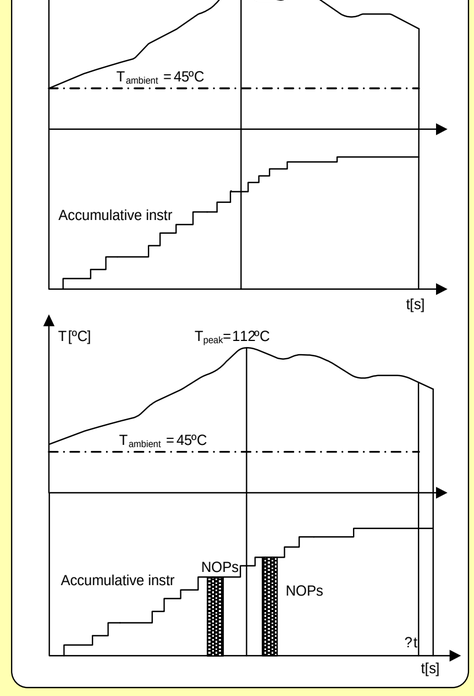
Instr1 .L1 || Instr2 .M1 || Instr3 .S1
Instr1:{L1,S1,L2,S2}
Instr2:{M1, M2}
Instr3:{L1,S1,D1,L1,S1,D1}

| Instr1 | Instr3 | Instr2 | | | | | |
|---|---|---|---|---|---|---|---|
| L1 | S1 | M1 | D1 | D2 | M2 | S2 | L2 |
| 86ºC | 84ºC | 82ºC | 80ºC | 74ºC | 72ºC | 73ºC | 75ºC |

(a)

| | | | | Instr3 | Instr2 | Instr1 | |
|---|---|---|---|---|---|---|---|
| L1 | S1 | M1 | D1 | D2 | M2 | S2 | L2 |
| 86ºC | 84ºC | 82ºC | 80ºC | 74ºC | 72ºC | 73ºC | 75ºC |

(b)




- 1st approach based on the rebinding of instruction on the coolest possible functional unit
- 2nd approach based on the insertion of NOPs in order to allow functional units to cool down