SmartPDN: Forging the Future of Scalable Power Delivery Networks

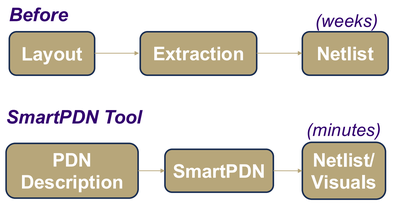
STUDENTS: BRENDA LIN, CHIA-EN WU, FERGUS XU, JERRY CHUNG, PANDHA PANTHAWANGKUN

CHALLENGE IN CHIP POWER DESIGN

Chips require **power delivery networks (PDN)**, the system of metal wires that distributes power from the chip's pins to its circuits. Verifying PDN properties requires running **simulations**, and to run simulations you need a circuit model called a **netlist**.

Traditional Netlist Generation

1. Engineers spend months designing the complete physical layout of the chip
2. A commercial extraction tool then reads the finished layout and generates the netlist
3. Simulations are run using the generated netlist

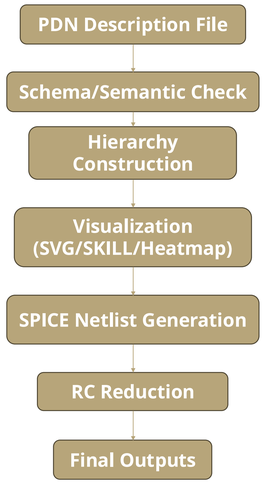


Essentially, physical layout (placing and routing every wire on every metal layer of the chip) takes engineers weeks before any extraction tool can start. This means PDN problems are not discovered until very **late**, when fixing them is expensive and time-consuming.

SmartPDN lets engineers skip the wait. Instead of a finished layout, engineers provide a simple chip description file. SmartPDN reads it and produces simulation-ready outputs in minutes.

REQUIREMENTS

Architecture Pipeline



Inputs

- A chip description file (YAML format) describing the chip's layers, connections, and geometry

Expected Behavior

- Validates the input file and reports any errors before processing
- Runs a pipeline: validate → interpret → generate outputs
- Errors halt execution; warnings are logged

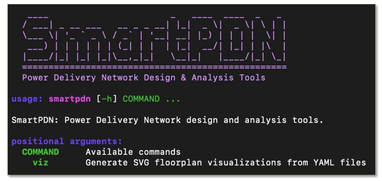
Outputs

- On-screen floorplan visualization (SVG)
- Layout script for Cadence Virtuoso (SKILL file)
- Simulation-ready circuit netlist (SPICE format, unreduced and reduced)
- Spatial resistance heat map

IMPLEMENTATION

SmartPDN Workflow:

Step 1: Write a chip description file The engineer describes the chip's power network in a structured text file: which metal layers exist, how wide the wires are, how cells are arranged, and how they connect.



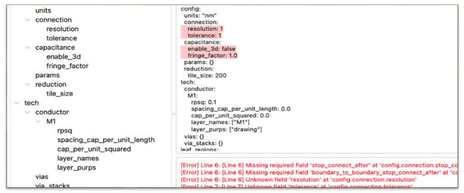
Command Line Interface

```
leaf_regions:
  cell_LA:
    bbox: [17.368, 9.962, 308.64, 30.45] #llx, lly, urx, ury
    tracks_horizontal:
      case1:
        conductor: metal4_normal
        feed_through_lr: [True, True]
        count_width_pitch: [2, "core_m4_width", 11.84]
        start_y: [10.945]
```

Excerpt of Chip Description File

Step 2: Input grammar check If the file has formatting errors or invalid references, the tool reports exactly what is wrong and where, so the engineer can fix it before proceeding.

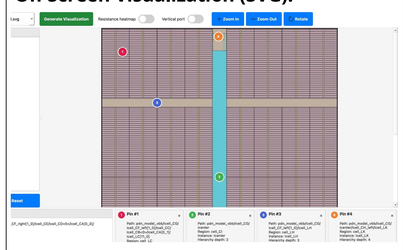
Step 3: Automatic processing pipeline Once the file is valid, SmartPDN parses it into an internal model of the chip.



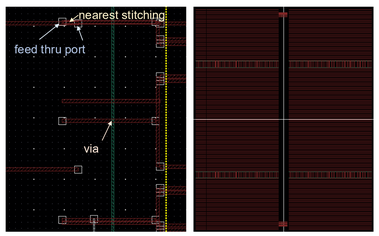
Grammar-Checker Feature

Step 4: Choose your output From the command line, the engineer can generate any of the included **outputs**:

On-Screen Visualization (SVG):



SKILL File for Cadance Simulation:



SPICE Netlist Generation (PDN Modeling):

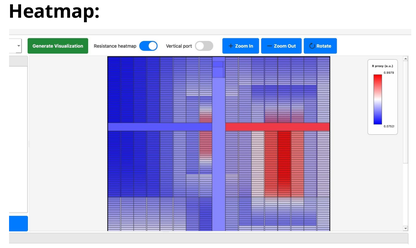
Leaf Cell:

```
* SPICE File
*
* Creator : SmartPDN hierarchical RC emitter
* Date : Fri Mar 13 13:01:34 2026
* Top : pdn_model_vbb
*
.subckt cell_LB left_ports<0> left_ports<1> right_ports<0> right_ports<1> top_ports<0> top_ports<1> top_ports<2> top_ports<3> top_p
* resistor model node2 resistance(ohm) layer X Y
R1 left_ports<0> IX_0_metal4_36447_37531 1.58123981481 $[metal4] $X=27.807250 $Y=37.531000
C1 left_ports<0> 0 2.55521175e-15 $[metal4] $X=37.727500 $Y=37.531000
C2 IX_0_metal4_36447_37531 0 2.55521175e-15 $[metal4] $X=96.447000 $Y=37.531000
R2 IX_0_metal4_36447_37531 IX_0_metal4_70847_37531 6.28444444444 $[metal4] $X=53.247000 $Y=37.531000
C3 IX_0_metal4_36447_37531 0 4.58646e-15 $[metal4] $X=35.447000 $Y=37.531000
C4 IX_0_metal4_70847_37531 0 4.58646e-15 $[metal4] $X=79.847000 $Y=37.531000
R3 IX_0_metal4_70847_37531 IX_0_metal4_105547_37531 6.28444444444 $[metal4] $X=66.047000 $Y=37.531000
C5 IX_0_metal4_70847_37531 0 4.58646e-15 $[metal4] $X=79.847000 $Y=37.531000
```

Container:

```
.subckt cell_CC Icell_CB<0>__Icell_CA__0_0__Icell_LB__0_0__case1_softy_0 Icell_CB<0>__Icell_CA__0_0__Icell_LB__0_0__case1_softy_0
XIcell_CB<0> Icell_CB<0>__Icell_CA__0_0__Icell_LB__0_0__case1_softy_0 Icell_CB<0>__Icell_CA__0_0__Icell_LB__0_0__case1_softy_0
XIcell_CB<1> Icell_CB<1>__Icell_CA__0_0__Icell_LB__0_0__case1_softy_0 Icell_CB<1>__Icell_CA__0_0__Icell_LB__0_0__case1_softy_0
XIcell_CB<2> Icell_CB<2>__Icell_CA__0_0__Icell_LB__0_0__case1_softy_0 Icell_CB<2>__Icell_CA__0_0__Icell_LB__0_0__case1_softy_0
XIcell_CB<3> Icell_CB<3>__Icell_CA__0_0__Icell_LB__0_0__case1_softy_0 Icell_CB<3>__Icell_CA__0_0__Icell_LB__0_0__case1_softy_0
.ends cell_CC
```

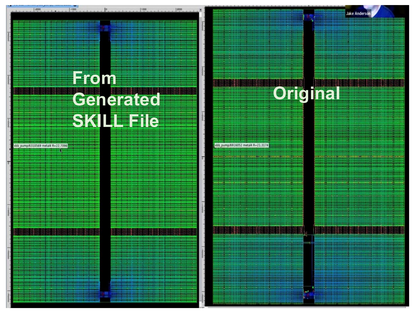
Heatmap:



RESULTS

The full generation pipeline was tested using a real chip description file, demonstrating that the tool can we be used in to process **realistic workloads**. All **validation checks were successfully run**, confirming that the input was well-defined and consistent.

The pipeline successfully completed, generating both **SVG visualizations and SKILL layout** with consistent geometries. A **spatial resistance heat map** was also created, allowing users to visually identify high resistance regions in the generated power delivery network.



RSCALC (for verification)

A **simulation-ready netlist** was also generated. Resistance measurements from the generated outputs closely matched expected values, with the **SKILL layout producing 26.6 Ω and the SPICE netlist producing 26.4 Ω**, demonstrating acceptable accuracy compared to industry-standard tools for early-stage design exploration.

In addition, the full pipeline execution time was significantly reduced: the SKILL layout was generated in approximately **30 seconds**, the netlist in **10 seconds**, and HSPICE simulation completed in **~160 seconds**. This end-to-end flow is driven entirely by the chip description input and eliminates the need for manual layout design, which traditionally takes weeks, enabling rapid iteration and early PDN validation. The project also produced **detailed documentation and example files**, making the project easy to use, test, and expand in the future.

FUTURE WORK

Future work will focus on **improving the accuracy, usability, and integration** into real-world environments:

- Explore adding support for a richer set of power delivery network descriptions, including adding **support for multiple networks at once** and **modeling relevant effects**. A stretch feature could also make the tool support 3D geometries, which is used in advanced packaging.
- Explore improving the **usability and integration** of this program. Future work could look to make the process of describing a chip easier and more accessible.
- Exploring the possible **integration into larger workflows**, such as simulation, making the process of simulating a prototype seamless.

CONCLUSION

SmartPDN enables engineers to model and verify a chip's power delivery network before the physical layout is finished. By accepting a simple chip description file and automatically generating visualizations, layout scripts, and simulation-ready netlists, the tool moves PDN analysis from the end of the design cycle to the beginning, reducing risk and enabling faster iteration.

The core pipeline is complete and validated, ready for real-world application.

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISERS: PROFESSOR ANG LI
INDUSTRY MENTOR: JAKE ANDERSON
SPONSOR: MICRON TECHNOLOGY