Automated PCB Component Placement and Copper Trace Layout for Power Electronics

Gus Cheng Zhang, The University of Manchester

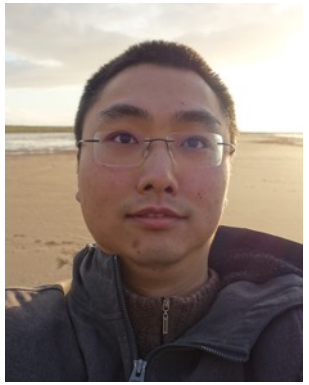
cheng.zhang@manchester.ac.uk

CPE2023

4 July 2023

About us

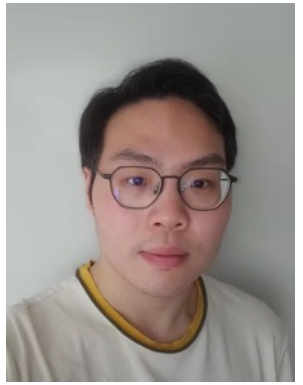
Design and Optimisation Automation for Power Electronics (DOAPE)
Power Conversion Group, Department of Electrical and Electronic Engineering



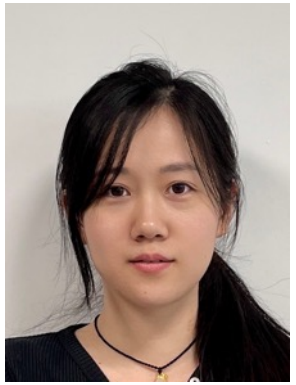
Dr Gus Zhang (me)
Lecturer

Prof. Andrew Forsyth
Chair Professor

Dr Judith Apsley
Senior Lecturer



Mr Yidong Tian
PGR Student



Miss Zhuoru Li
PGR Student

About this presentation

It is about our explorations in

Design and optimisation automation of PCB layouts

for Power Electronics

and we share the *experiences*,

initiate the *discussions*,

tell what doesn't work and what may work,

and look forward to future *collaborations*.

Outline

- Background, Aims and Objectives – why we're doing this
- Methodologies – what are the ideas
- Implementations – what we have encountered
- Examples – what we have done
- Conclusion – next plans
- Q & A

Background – the trend

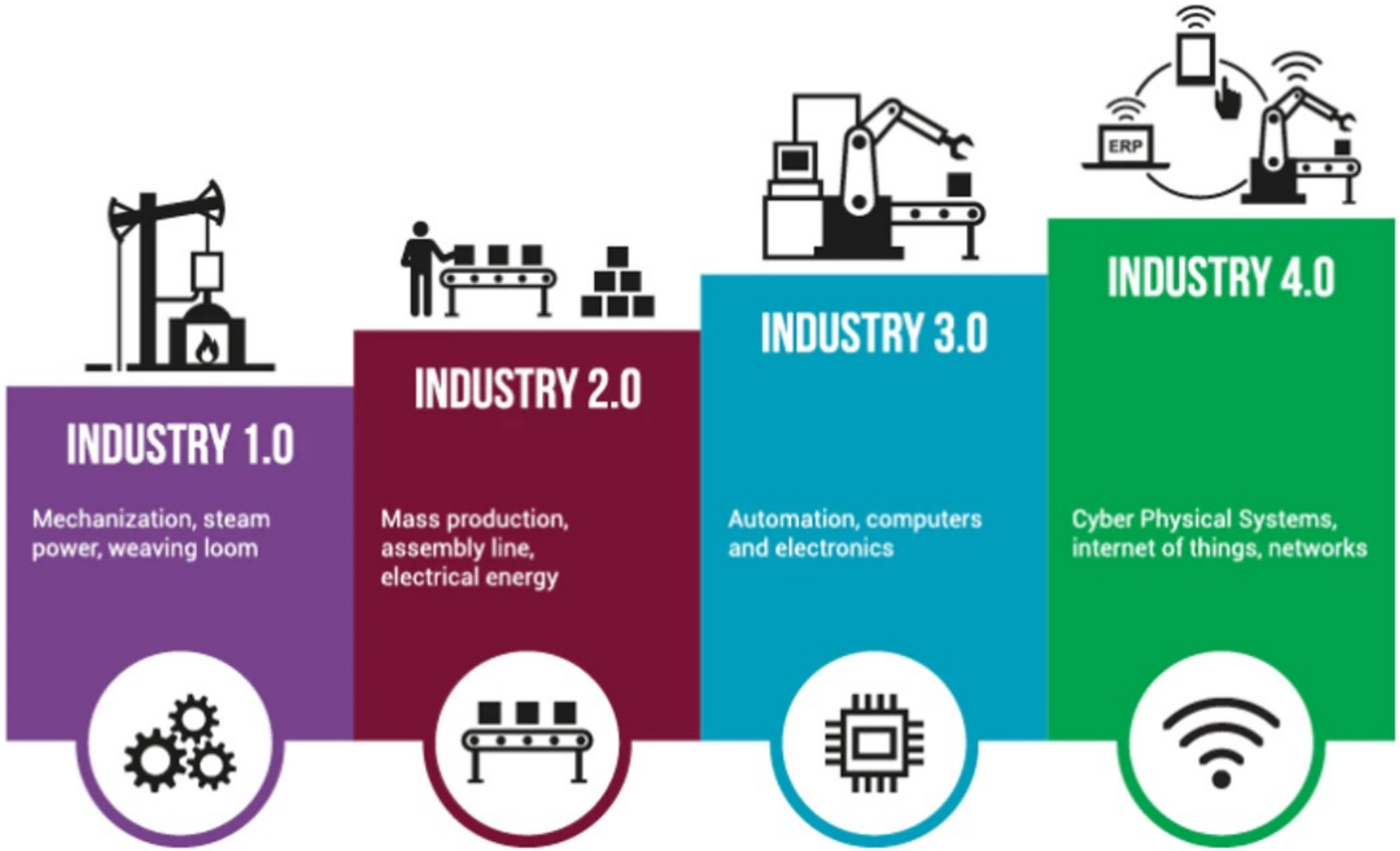


Image: automationworld.com

My interpretation:

1.0 – New power sources

2.0 – New production tools

3.0 – Automated production processes

4.0 and future –

Automated **design** processes

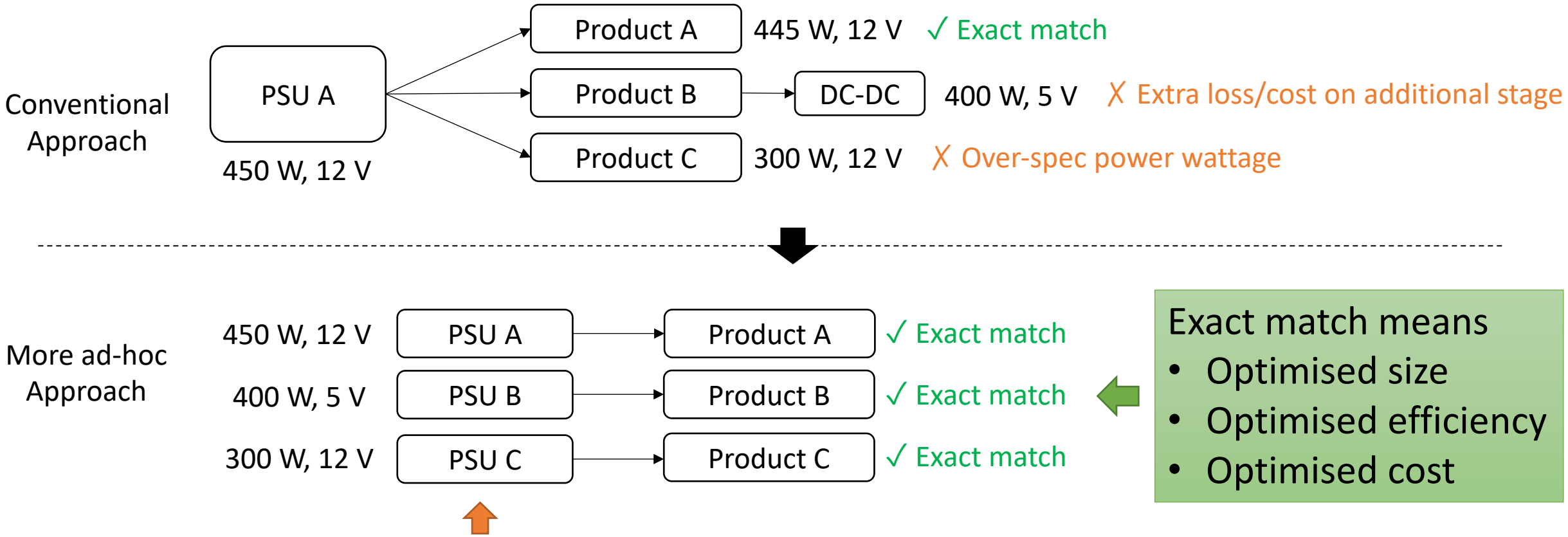
The linkage to Power Electronics

- Almost all electrical/electronic products have a part of power electronics – as long as there is energy conversion!

- Need to redesign them every time?
  - Need to optimise for efficiency, cost, volume and weight …

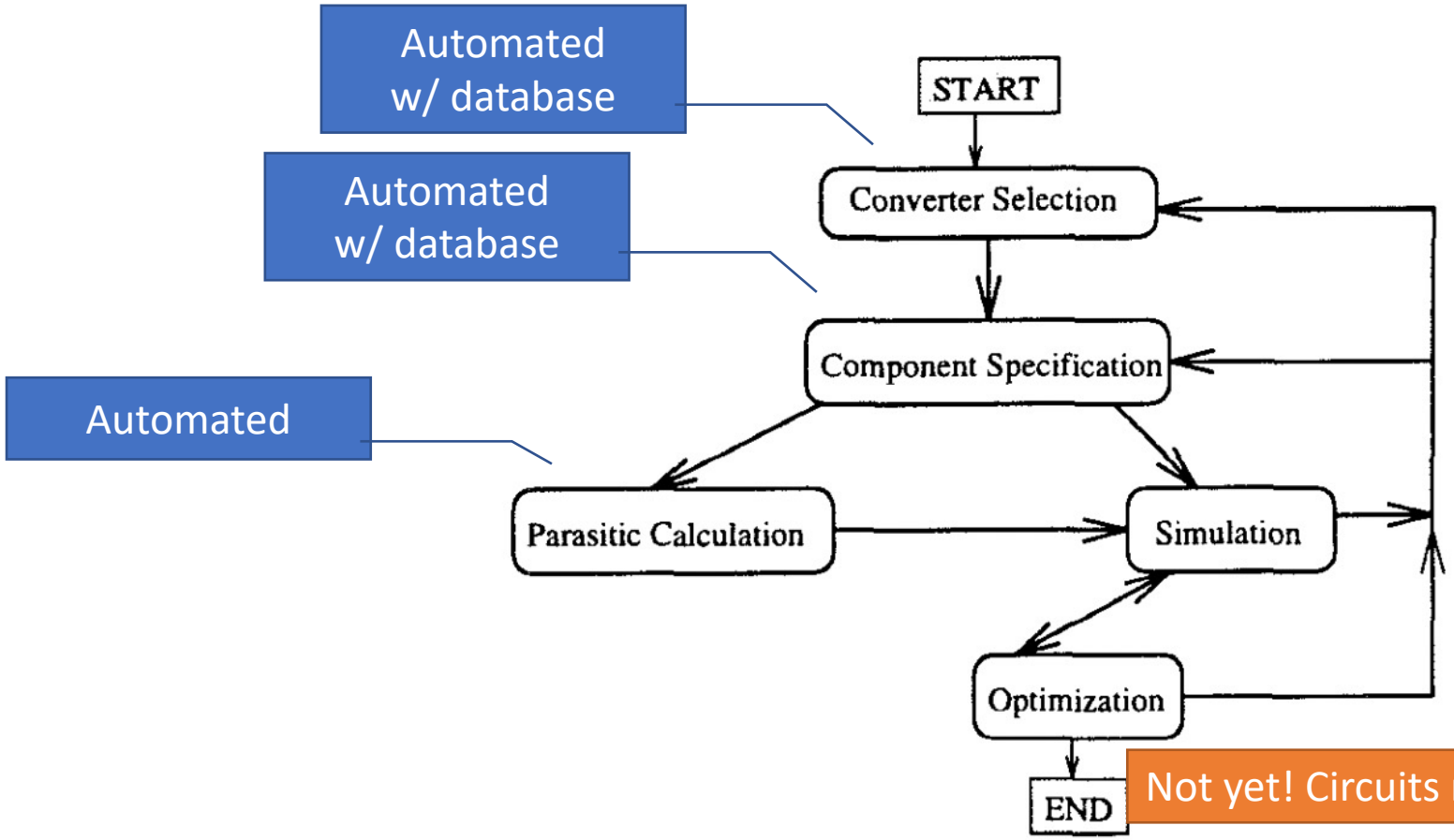
Source: consumertechtips.com, portlandgeneral.com

An emerging demand of more circuit designs



3x the work of design, development and testing!

Run-down of a generic PE design

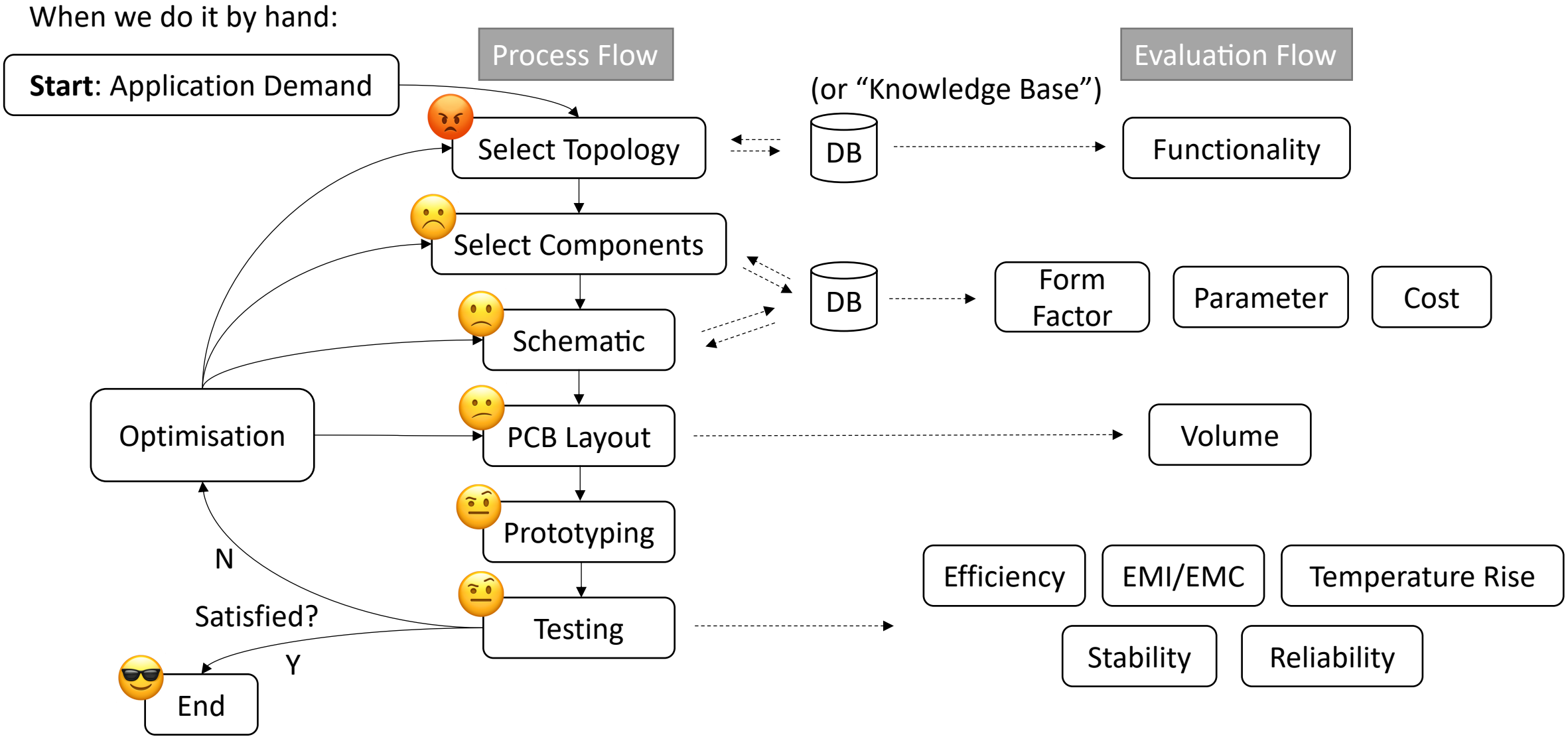


Most of these steps have been well studied.

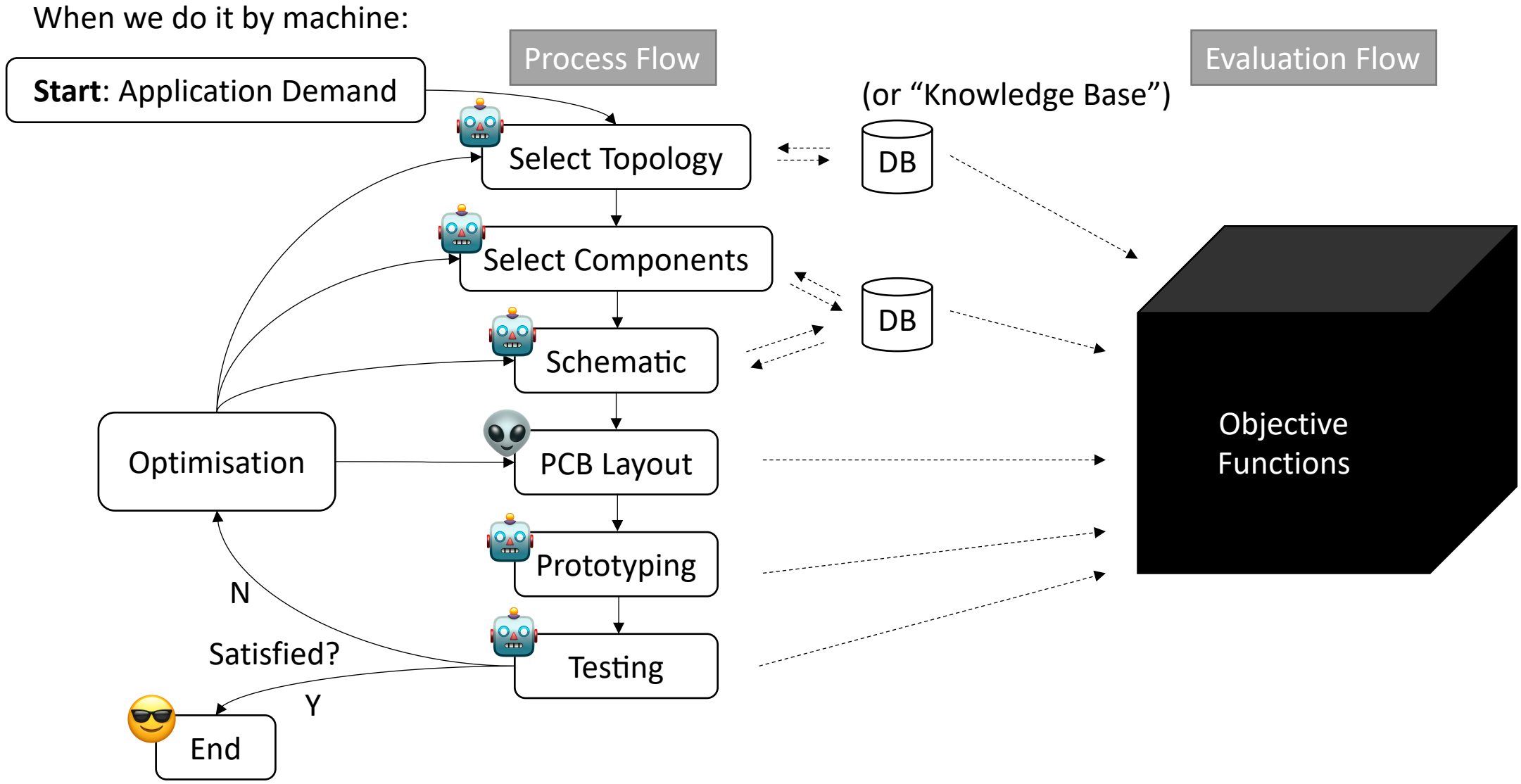
Not yet! Circuits need to be built on PCBs

L. E. Amaya, P. T. Krein and F. N. Najm, "A design methodology for power electronics," 5th IEEE Workshop on Computers in Power Electronics, 1996, pp. 52-57, doi: 10.1109/CIPE.1996.612336.

A resketch, rough and messy…



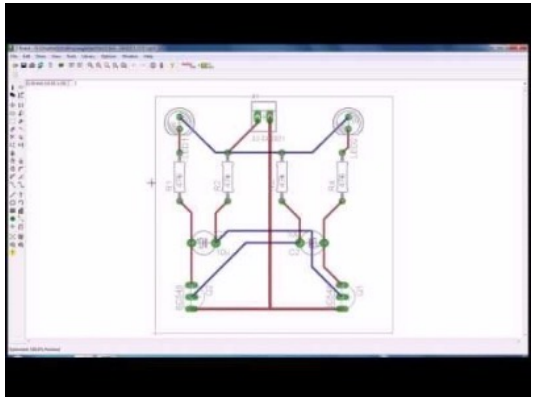
A resketch, rough and messy…



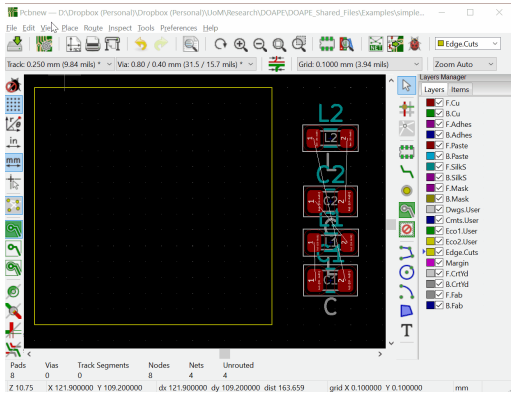
Existing auto-routing and auto-placement

- Problems:
  - Simplified models assuming PCB tracks are ideal conductors
  - No or little considerations on component interconnections
  - Work for simple, low frequency circuits
  - Integrated calculators mainly for impedance matching and latencies → for high-speed mixed signal circuits

Altium Designer



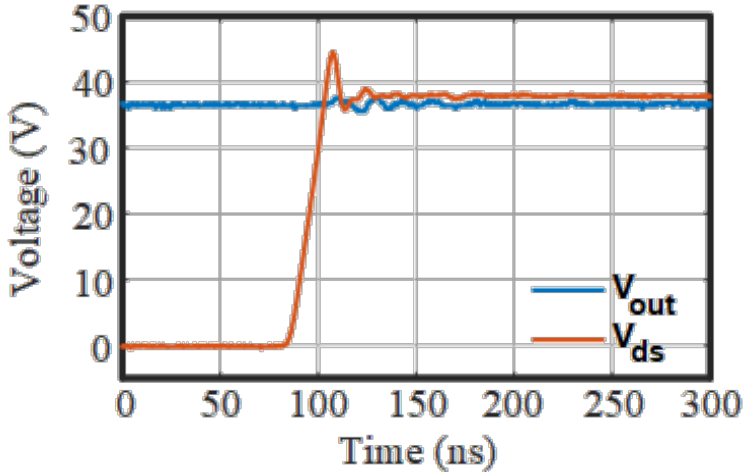
EAGLE



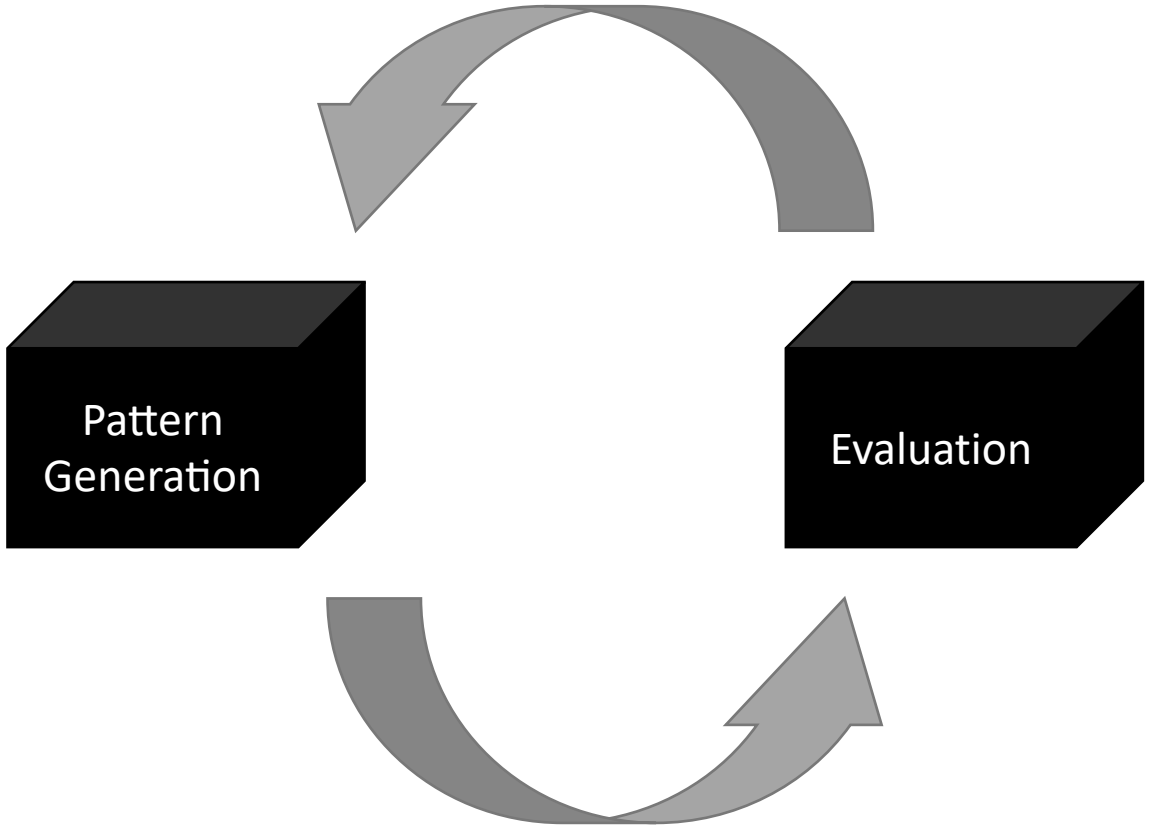
KiCAD

Unsuitable for Power Electronics!

- For power electronics we care about:
  - Energy efficiency!
    - Less I2R loss
    - Faster and clean dv/dt and di/dt edges
    - In other words, need traces to be as close as ideal
  - Box volume
  - Thermal dissipation (e.g. balanced distribution)



Method: Generative and Evaluative



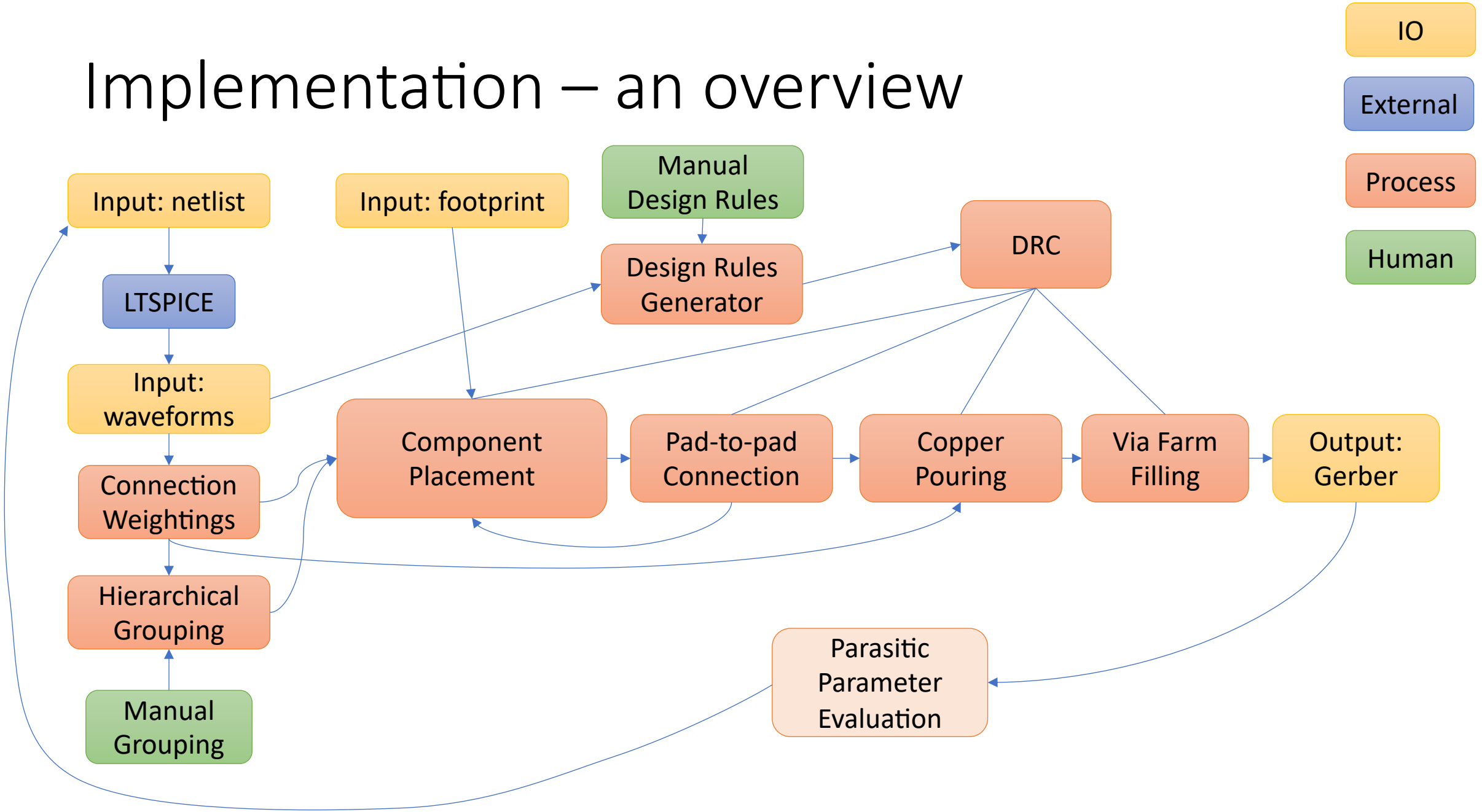
The scope

- We target on PCB layout automation only
  - Automated placement
  - Automated trace layout, and copper pours
  - Automated via placement
  - CAM files generation
- In future developments we would like to integrate with other design automation tools
  - Topologies
  - Component selection
  - Control
  - …

Assumptions

- Circuit topology is known
- Components, including values and packages are determined
- Circuit operation is known
  - Therefore, all node voltages and terminal currents are known, through SPICE simulation
- Targets in layman's words:
  - Least Ohmic loss on PCB
  - Least parasitic impact
  - Use intuition to define what is the best PCB layout

Implementation – an overview



Input: Netlist

- Widely used to represent circuit – the machine readable schematic
- Used for both simulation and topology extraction
- Examples:

```
* C:\Users\Frank Tian\Desktop\DC-DC_design\boost_des
L1 IN N001 330
R1 N002 N003 2
V1 IN 0 100
Rload OUT 0 3200
C1 OUT 0 0.27 Rser=370m
V2 N003 0 PULSE(-5 5 0 1e-9 1e-9 1.47e-6 2e-6)
XU2 N001 OUT C3D1P7060Q
XU1 N002 N001 0 0 GaN_LTspice_GS66504B_L1V4P1
.tran 2m startup
.lib C:\Users\Frank Tian\Desktop\DC-DC_design\LTspic
.lib GaN_LTspice_GS66504B_L1V4P1.lib
.backanno
```

.net file for SPICE

```
<signal name="VSN_SENSE">
<contactref element="R3" pad="2"/>
<contactref element="R2" pad="1"/>
<contactref element="C5" pad="1"/>
<contactref element="C4" pad="2"/>
<wire x1="20.75" y1="12.9" x2="21.65" y2="12.9" width="0.4" layer="1"/>
<wire x1="20.75" y1="11.9" x2="21.65" y2="11.9" width="0.4" layer="1"/>
<wire x1="20.75" y1="11.9" x2="20.9" y2="11.9" width="0.4" layer="1"/>
<wire x1="20.9" y1="11.9" x2="21.2" y2="12.2" width="0.4" layer="1"/>
<wire x1="21.2" y1="12.2" x2="21.2" y2="12.7" width="0.4" layer="1"/>
<wire x1="21.2" y1="12.7" x2="21.45" y2="12.7" width="0.4" layer="1"/>
<wire x1="21.45" y1="12.7" x2="21.65" y2="12.9" width="0.4" layer="1"/>
<wire x1="20.75" y1="11.9" x2="21.7" y2="11.9" width="0.4" layer="1"/>
<wire x1="21.7" y1="11.9" x2="21.8" y2="11.8" width="0.4" layer="1"/>
<wire x1="21.8" y1="11.8" x2="21.8" y2="10.9" width="0.4" layer="1"/>
<via x="21.8" y="10.9" extent="1-16" drill="0.3" diameter="0.4"/>
<wire x1="21.8" y1="10.9" x2="21.8" y2="7.9" width="0.4" layer="15"/>
<wire x1="21.8" y1="7.9" x2="19.7" y2="5.8" width="0.4" layer="15"/>
<via x="19.7" y="5.8" extent="1-16" drill="0.3" diameter="0.4"/>
<contactref element="JP1" pad="10"/>
<wire x1="19.7" y1="5.8" x2="18.7" y2="4.8" width="0.4" layer="1"/>
<wire x1="18.7" y1="4.8" x2="18.7" y2="3.88" width="0.4" layer="1"/>
```

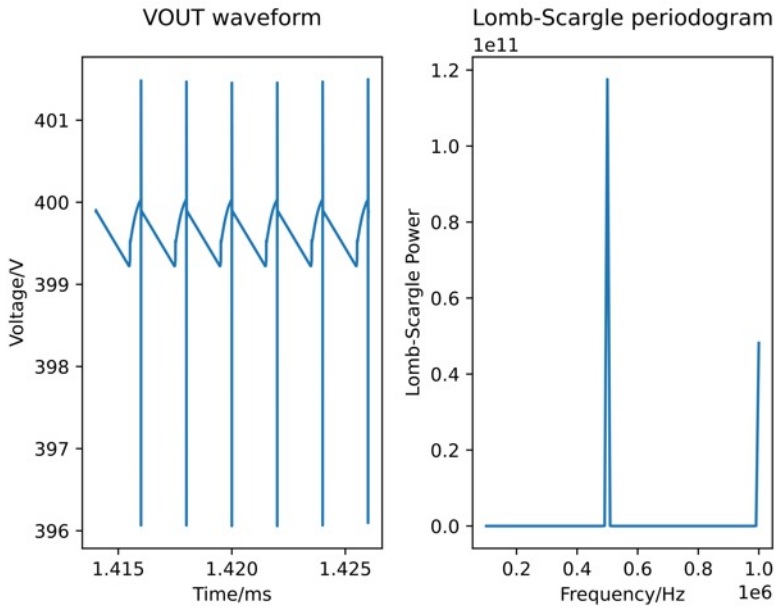
.brd file from EAGLE

Simulation result extraction

LTSPICE outputs .raw waveform files includes all

- Voltages on nodes
- Currents on component terminals

Unfixed step-size: Lomb-Scargle periodogram to find out the period




```
Title: * C:\Users\Frank Tian\Desktop\DC-DC_design\boost_design\boost_converter_50W.asc
Date: Tue Sep 07 11:03:52 2021
Plotname: Transient Analysis
Flags: real forward
No. Variables: 18
No. Points:      1156432
Offset:   0.0000000000000000e+000
Command: Linear Technology Corporation LTspice XVII
Backannotation: u2 a k
Backannotation: u1 gatein drainin sourcein source_s
Variables:
    0   time    time
    1   V(in)   voltage
    2   V(n001) voltage
    3   V(n002) voltage
    4   V(n003) voltage
    5   V(out)  voltage
    6   I(C1)   device_current
    7   I(L1)   device_current
    8   I(Rload)    device_current
    9   I(R1)   device_current
    10  I(V2)   device_current
    11  I(V1)   device_current
    12  Ix(u2:A)    subckt_current
    13  Ix(u2:K)    subckt_current
    14  Ix(u1:GATEIN)   subckt_current
    15  Ix(u1:DRAININ)  subckt_current
    16  Ix(u1:SOURCEIN) subckt_current
    17  Ix(u1:SOURCE_S) subckt_current
Binary:
```

We fetch:

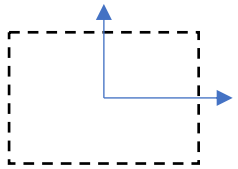
- Maximum absolute value and RMS value of all node pairs voltages, and their derivatives
- Maximum absolute value and RMS value of all terminal currents, and their derivatives

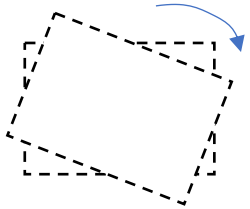
Simulation result extraction

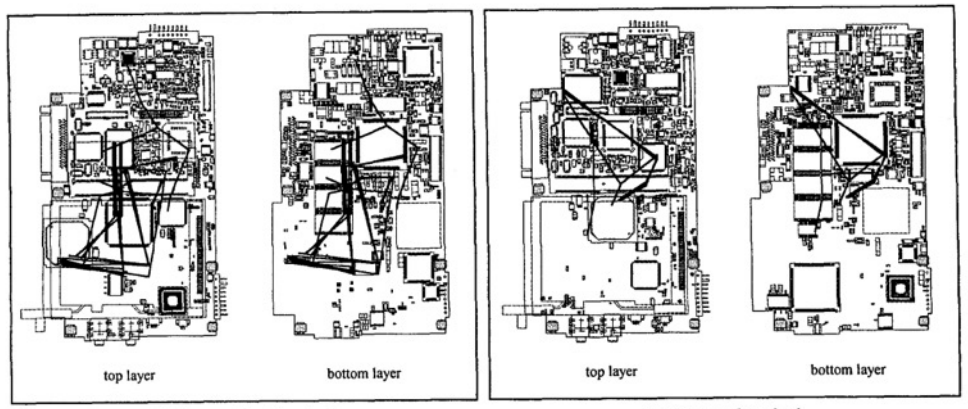
- For voltages between any pair of nodes (nets)
- v_{max} → related to clearance distances
- $\mathrm{d}v/\mathrm{d}t$ → related to stray capacitance, need to be minimised
- n nodes → $n(n-1)/2$ voltage differences
- For current between any pair of terminals (pads) – “connections”
- i_{max}, i_{rms} → related to I2R losses
- $\mathrm{d}i/\mathrm{d}t$ → related to impacts from inductances

Placement solution space

- For each component: origin offset (x, y) and angle (θ), all $\in \Re$
  - Regardless of the shape of the component







(a) Conventional method (b) Proposed method

Figure 7. Comparison between proposed automatic layout and conventional layout for the main board of a multimedia equipment.

Ref & most CAD: only limited to orthogonal placement

In practice:

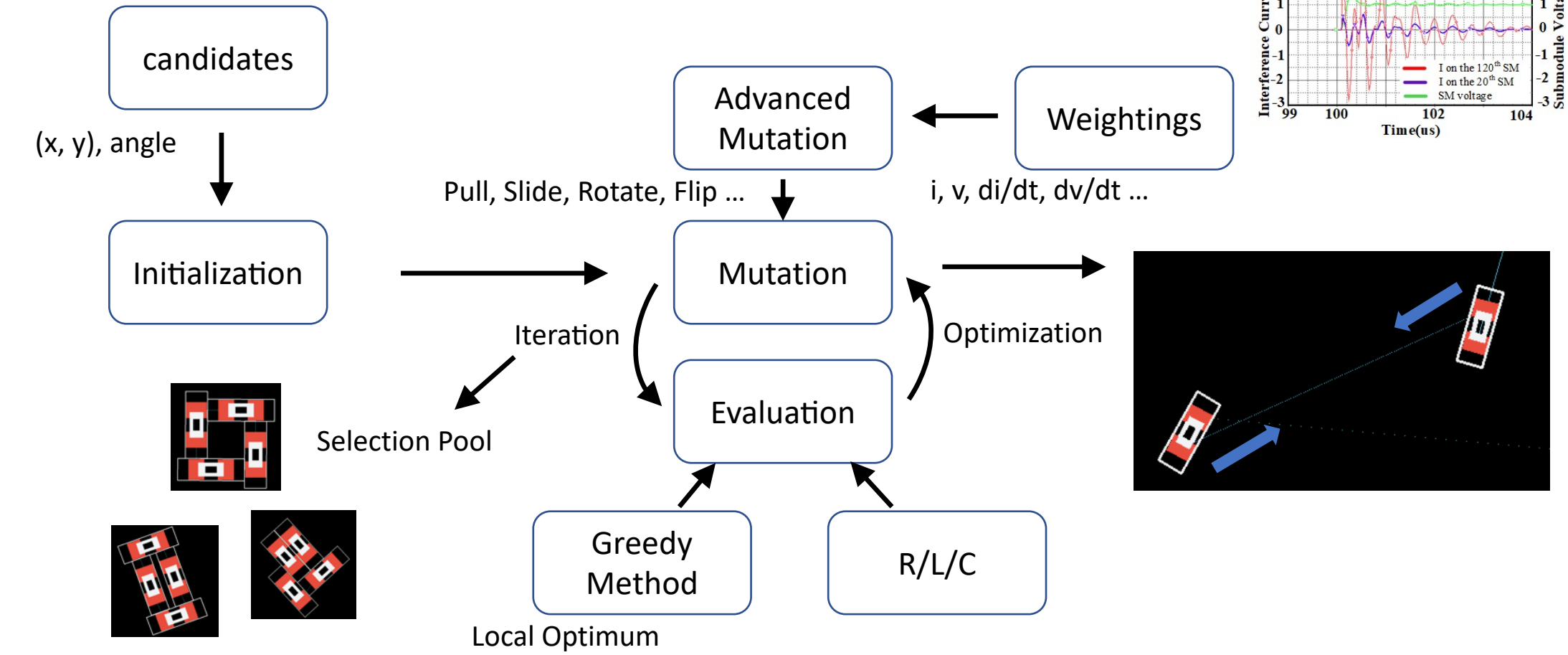
Image: Ken Shirriff

- Still limited degree-of-freedom, 3x (component count-1)

Y. Fukumoto, S. Miura, H. Ikeda, T. Nakayama, S. Tanimoto and H. Uemura, "A method of automatic placement that reduces electromagnetic radiation noise from digital printed circuit boards," IEEE International Symposium on Electromagnetic Compatibility. Symposium Record (Cat. No.00CH37016), Washington, DC, 2000, pp. 363-368 vol.1, doi: 10.1109/ISEMC.2000.875594.

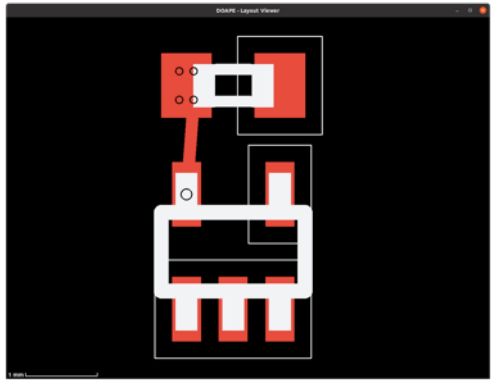
Implementation – Component placement

Genetic Algorithm

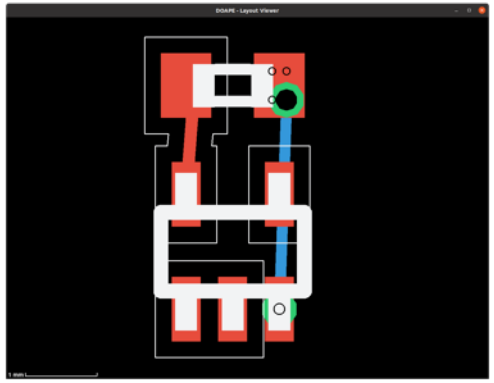


Connecting pads with traces

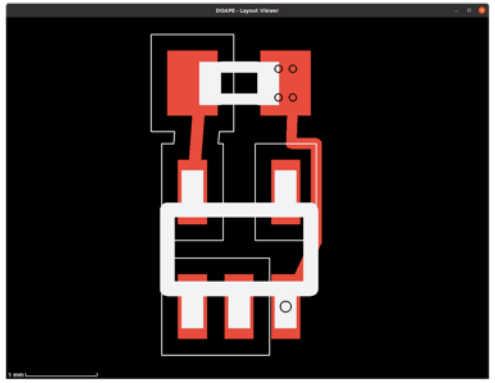
Bug algorithm – walk around obstacles
A* search algorithm – existing and predicted cost weightings via key points



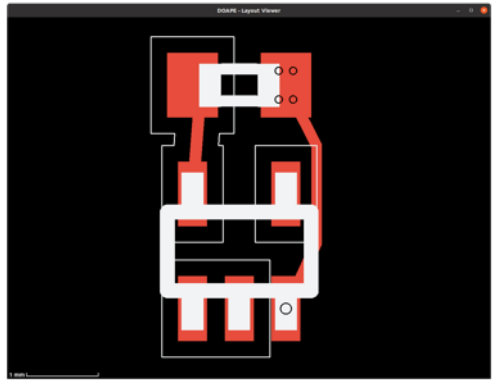
Left trace laid



Connect through bottom layer

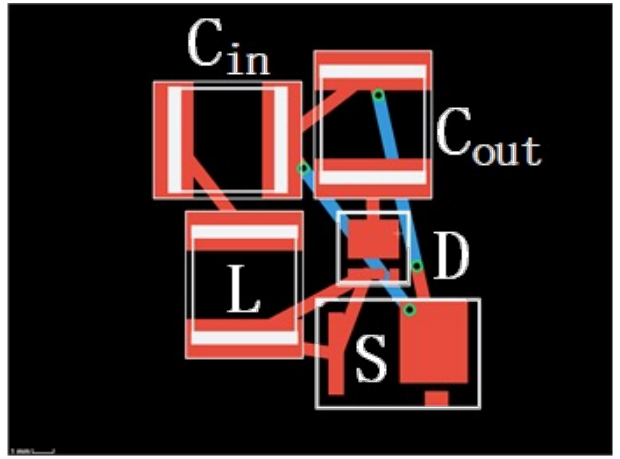


Connect on the same layer

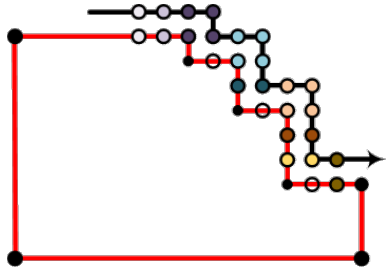


Straightening some corners

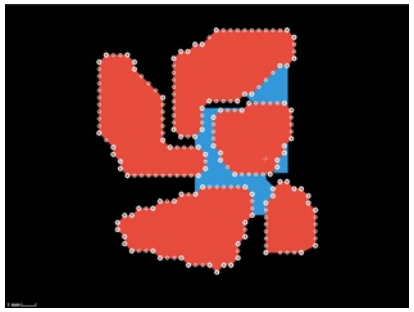
Copper pouring

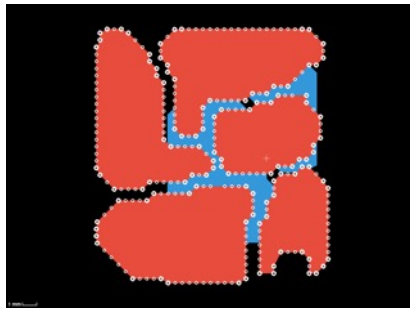


Rasterised approach

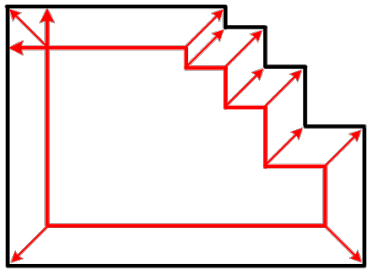


Acceptable resolution and execution efficiency, easy to control speed of expansion. 👍

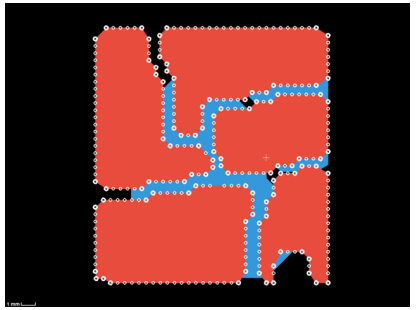


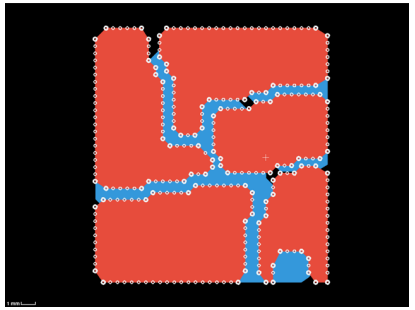


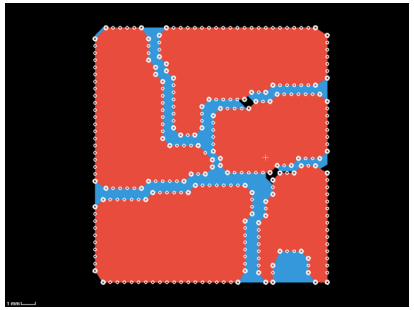
Vector approach



Very complicated computation near narrow and sharp corners. 👎



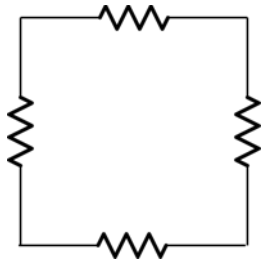


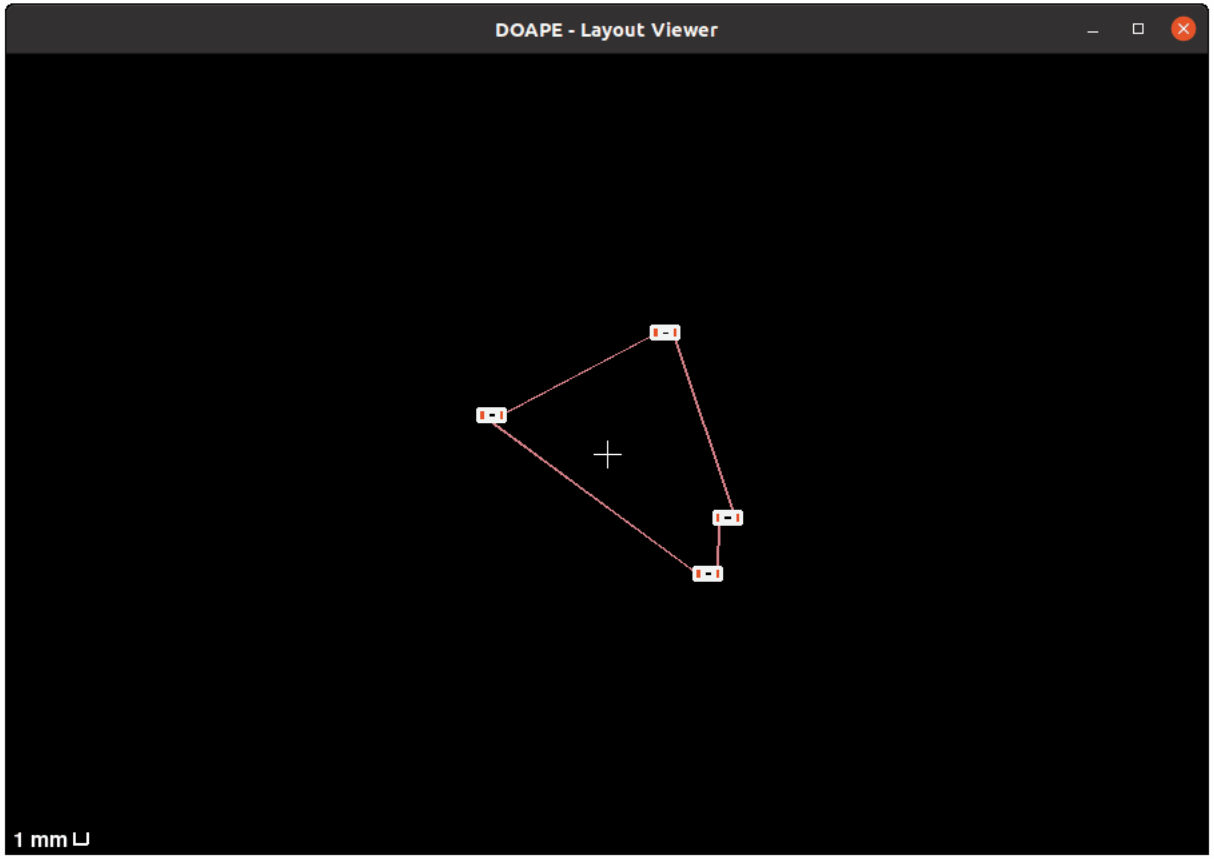


After pouring, the rasterised copper areas will be converted back to the vector formats.

Examples – Placement

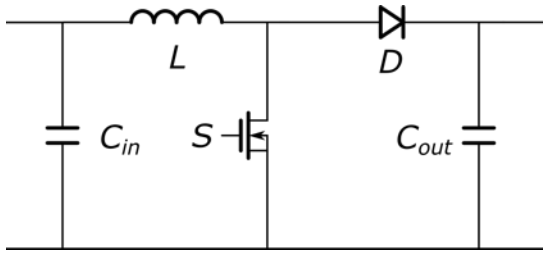
A simple case with four chip resistors head-to-head

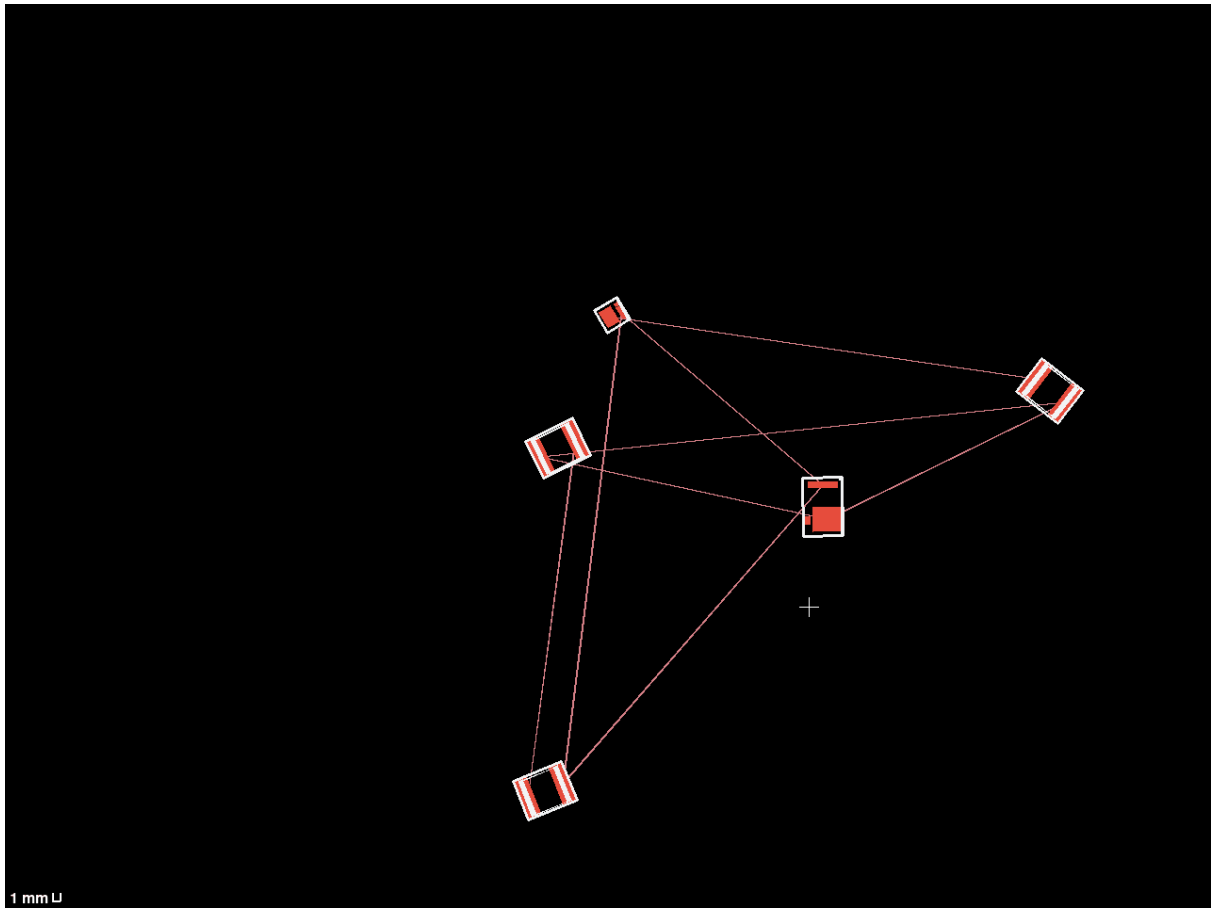




Examples – Placement and Copper Pouring

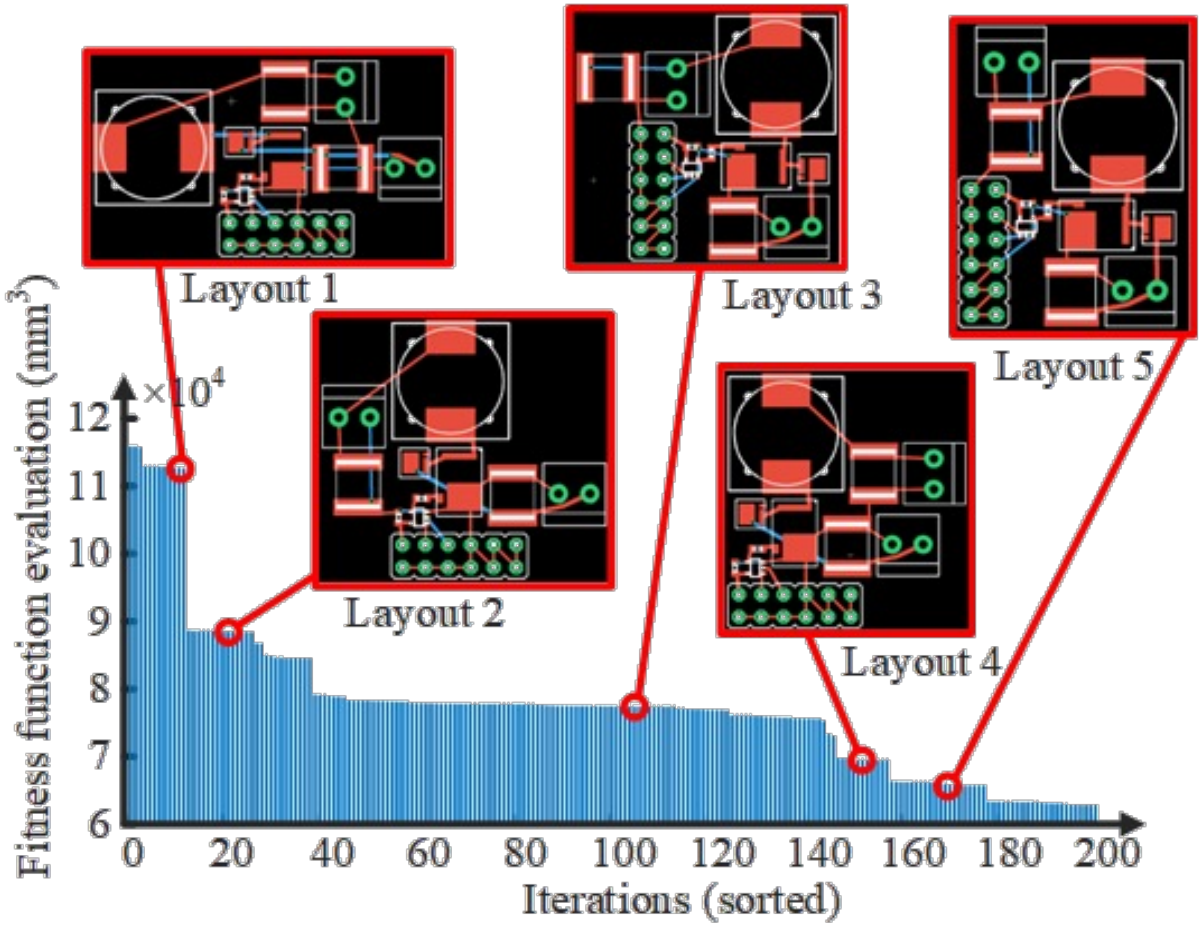
A simple case with five major components in a boost converter.

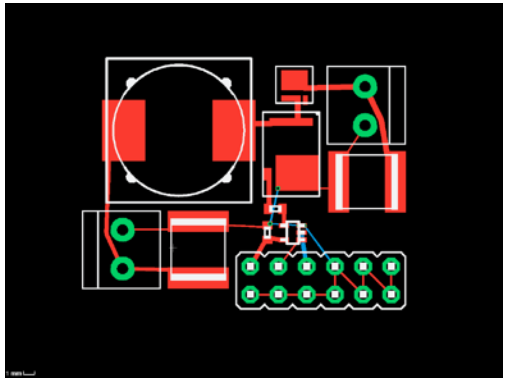




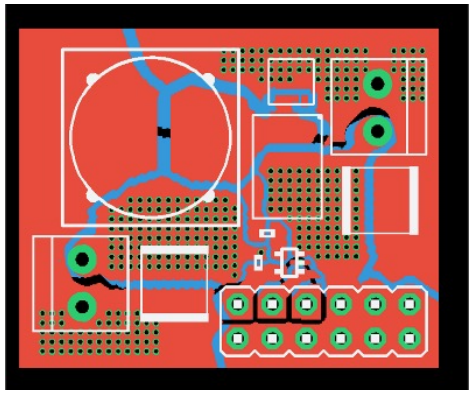
Example: A Simple Boost Converter

A more practical example: a boost converter with connectors and gate driver

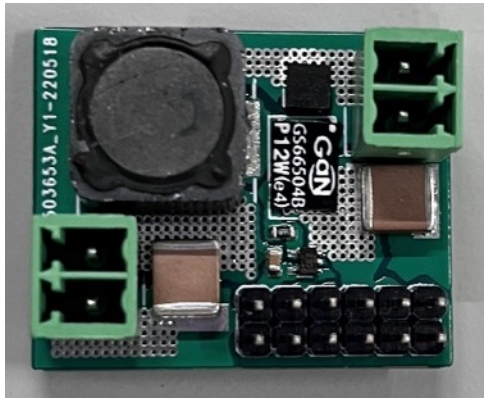




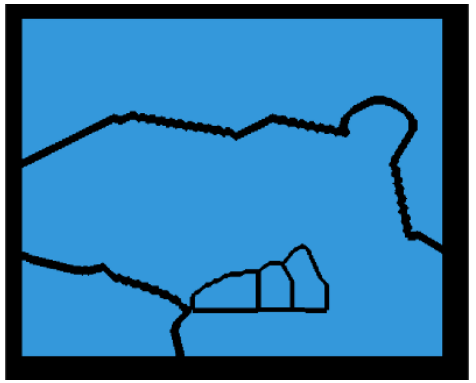
Placement



Layout (top layer)

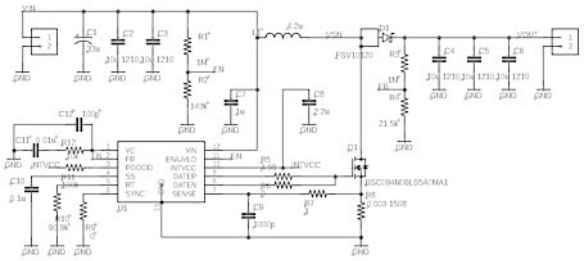


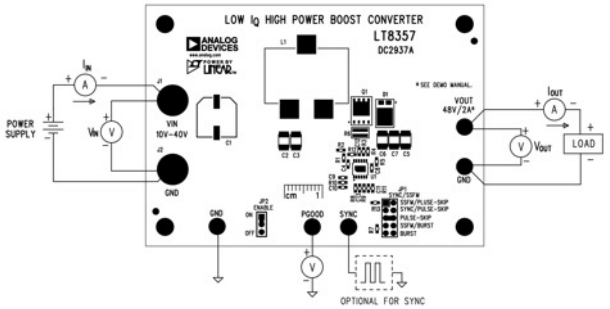
Product

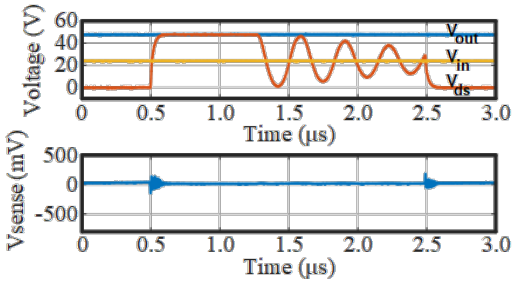


Layout (bottom layer)

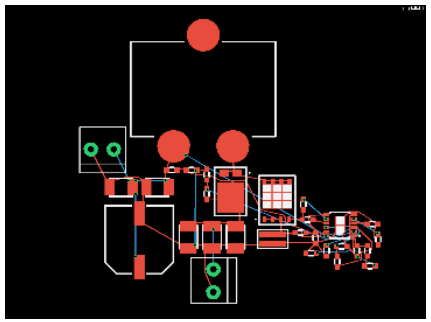
Example: LT2937A Boost Converter

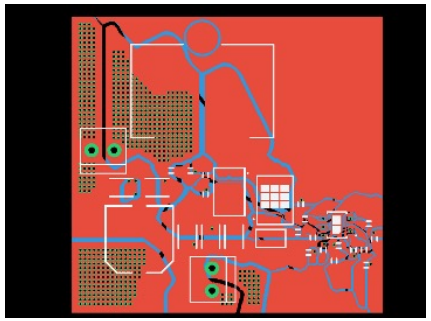


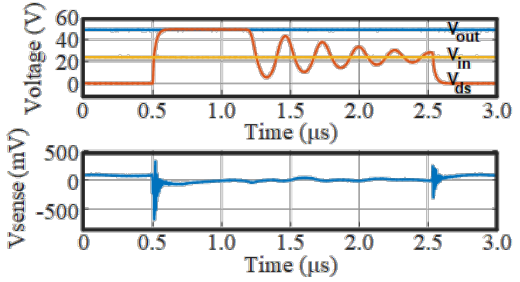




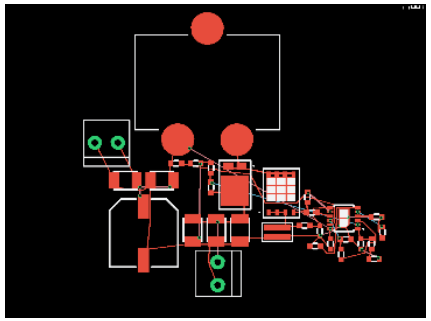
DC2937A Eval Board
4-Layer

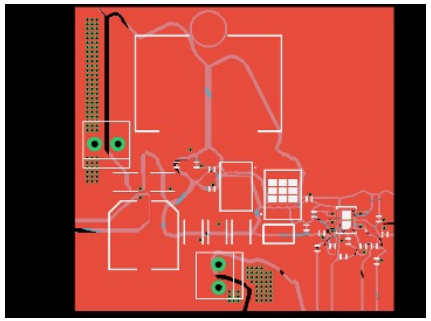


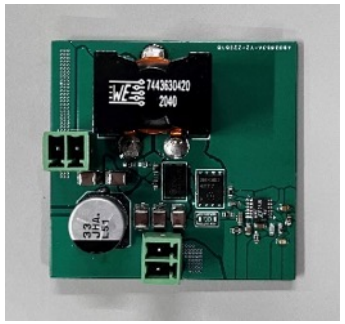


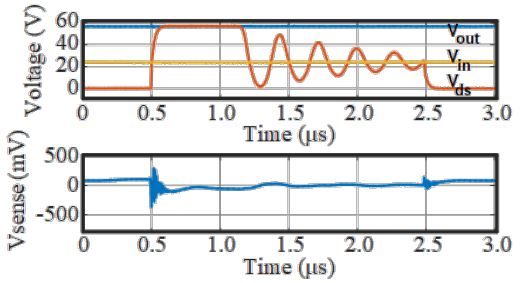


By PEPCB
2-Layer









By PEPCB
4-Layer

2.5x smaller than eval board, 30 minutes on i7-10510U+16G RAM

Conclusion

- We have explored automated PCB layout design including the steps of
  - Extraction of circuit topology and voltage/current information
  - Synthesising grouping hierarchy and weightings
  - Automatic placement subject to given rules and objectives
  - Automatic trace connections and copper pouring
- Big picture is still genetic algorithm, but lots of the works are done by hard-coding for better efficiency
- We’re still improving all steps through testing with more complicated circuits. Many techniques are constantly being evolved.

Additional notes

- We have used Python for generation most of the examples in this presentation. We are migrating the algorithms to C++ for better performance and more rigorous pre-compile checks.
- We have used floating number for vector coordinates. We will move to integers to avoid boundary ambiguity problems.
- Limited by the length of the presentation, we cannot include all details. Please feel free to contact us!

Thank you!

For any queries please contact cheng.zhang@manchester.ac.uk