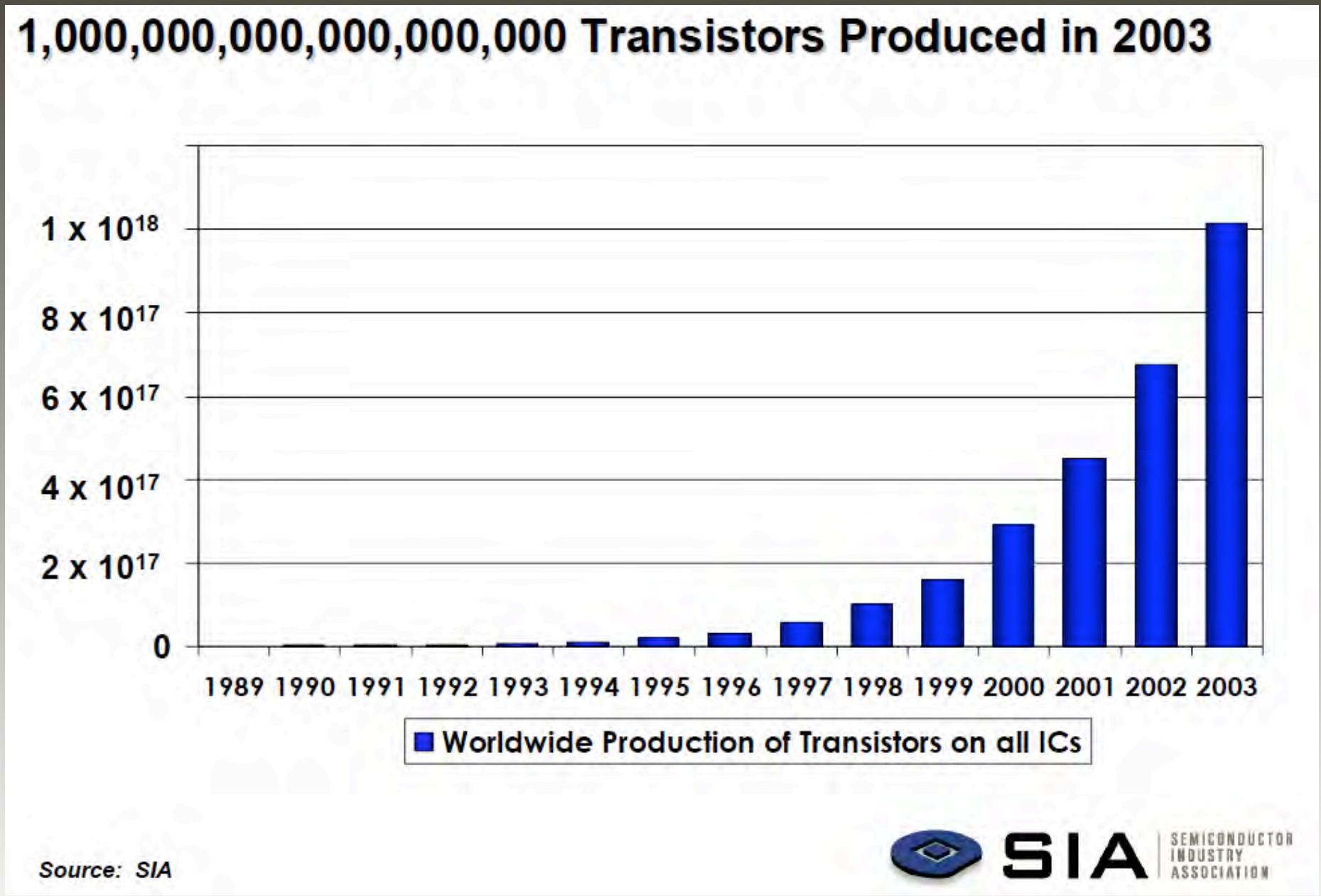
Preamble



1 Quintillion of Transistors

Semiconductor Growth: 2006

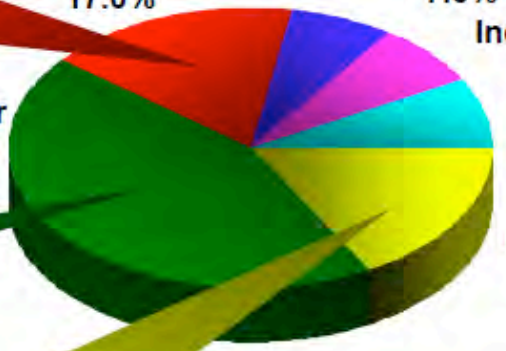
2006
Cell Phone Shipments
+13% (Units)

2006
PC Shipments
+10% (Units)

2006
Digital Camera
+9% (Units)

2006
MP3 Player
+52% (Units)

2006
Digital TV
+52% (Units)



$245B / +7.9%
2006

Sources:Semico/Nokia/Gartner/IDC/iSuppli/SIA November 2005 Forecast

SIA SEMICONDUCTOR INDUSTRY ASSOCIATION

NM

Electronic Design Automation at Transistor Level

Ricardo Reis

Universidade Federal do Rio Grande do Sul
Instituto de Informática - Porto Alegre - RS - Brasil
reis@inf.ufrgs.br

UFRGS

OUTLINE

1. Introduction
2. History
3. Standard Cell
4. CMOS Complex Gates
5. Layout Synthesis
6. Layout Strategies
7. Experimental Results
8. Conclusions

History

Logic Design Evolution

Years 70 : **microprocessors “hand made”**
computer used just as graphical input

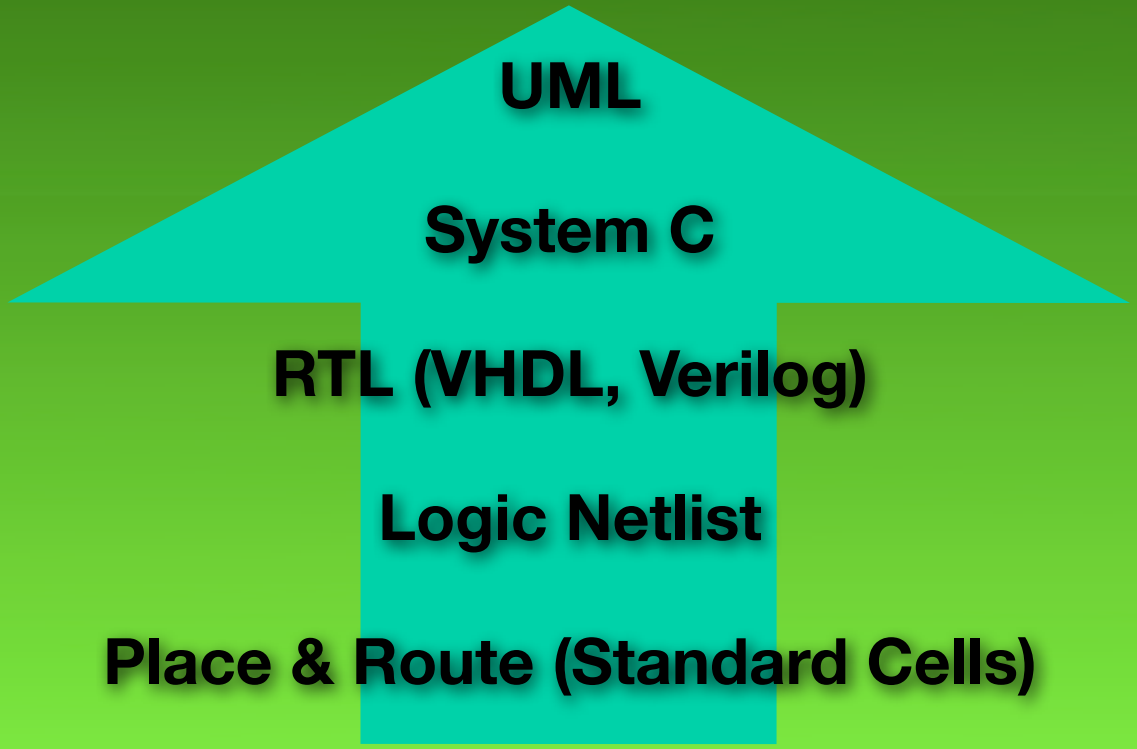
End Years 70: Random Logic → ROMs, PLAs
Z8000 → **M68000**

Years 90: ROMs, PLAs → Standard Cell
→ **486, Pentium**

History

Design Automation

ESL - Electronics System Level Design Automation



Introduction

Nowadays
Physical Design using Standard Cell
is a common practice

Should we look for another approach?

Why?

Standard Cell Approach

Is it a layout automated approach ?

NO !

Standard Cell Approach

Cell characterization
Cell performance predictability

But nowadays cell predictability is not anymore sufficient to have circuit predictability

Connections becomes the central problem !

Standard Cell Approach

Logic Options Limited to Cells Available in the Library

No optimal logic minimization

Cells oversized

Area

Standard Cell Approach

Far from Minimization on:

- Area
- Number of Transistors
- Wirelenght
- Delay
- Power

Change of Paradigm

Cell Generation on-the-fly

History

Logic Design Evolution

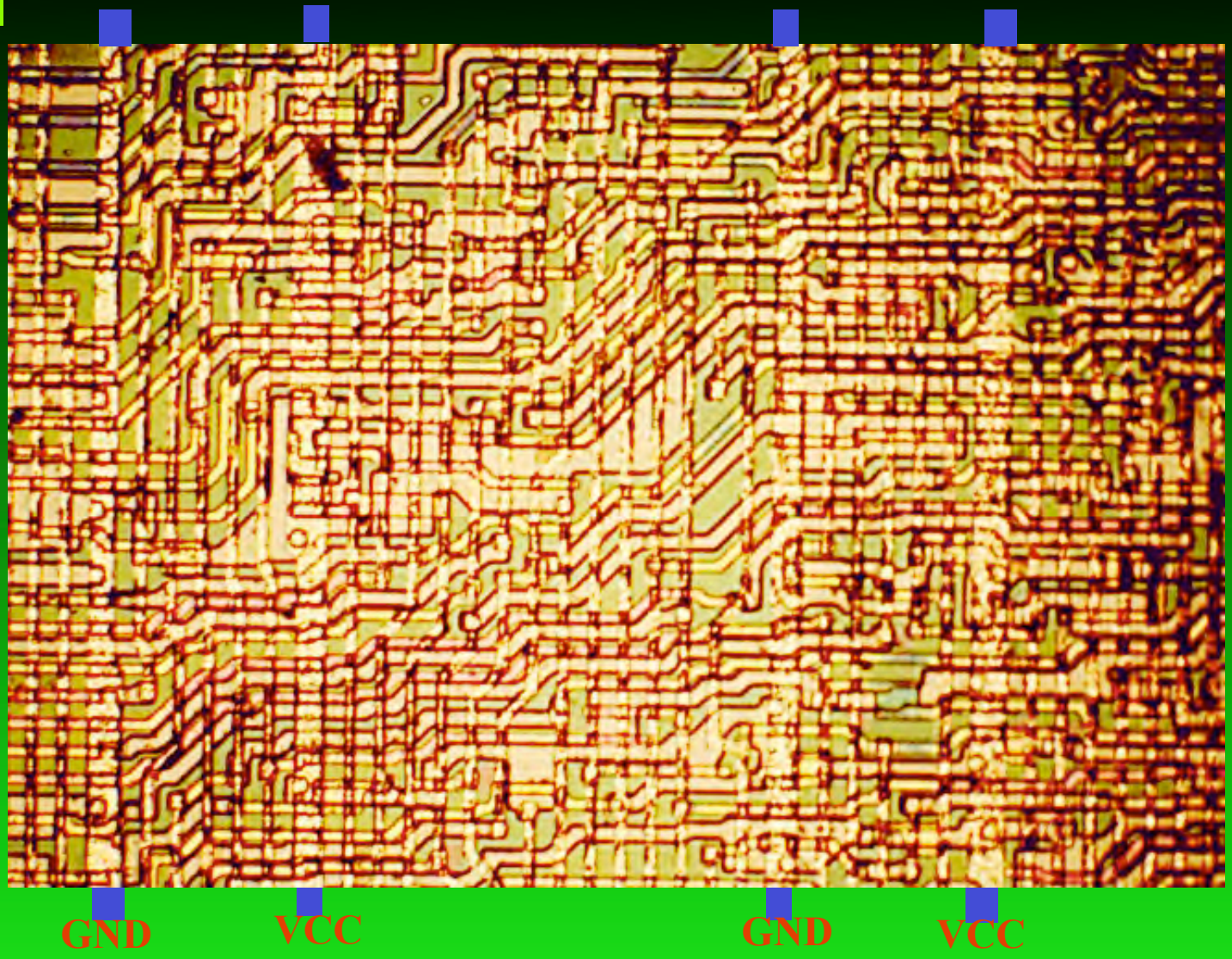
Years 70 : **microprocessors "hand made"**
computer used just as graphical input

End Years 70: Random Logic → ROMs, PLAs
Z8000 → **M68000**

Years 90: ROMs, PLAs → Standard Cell
486, Pentium

Next Step: Standard Cell → Random Logic
Automatic Layout of Cells-on-the-fly

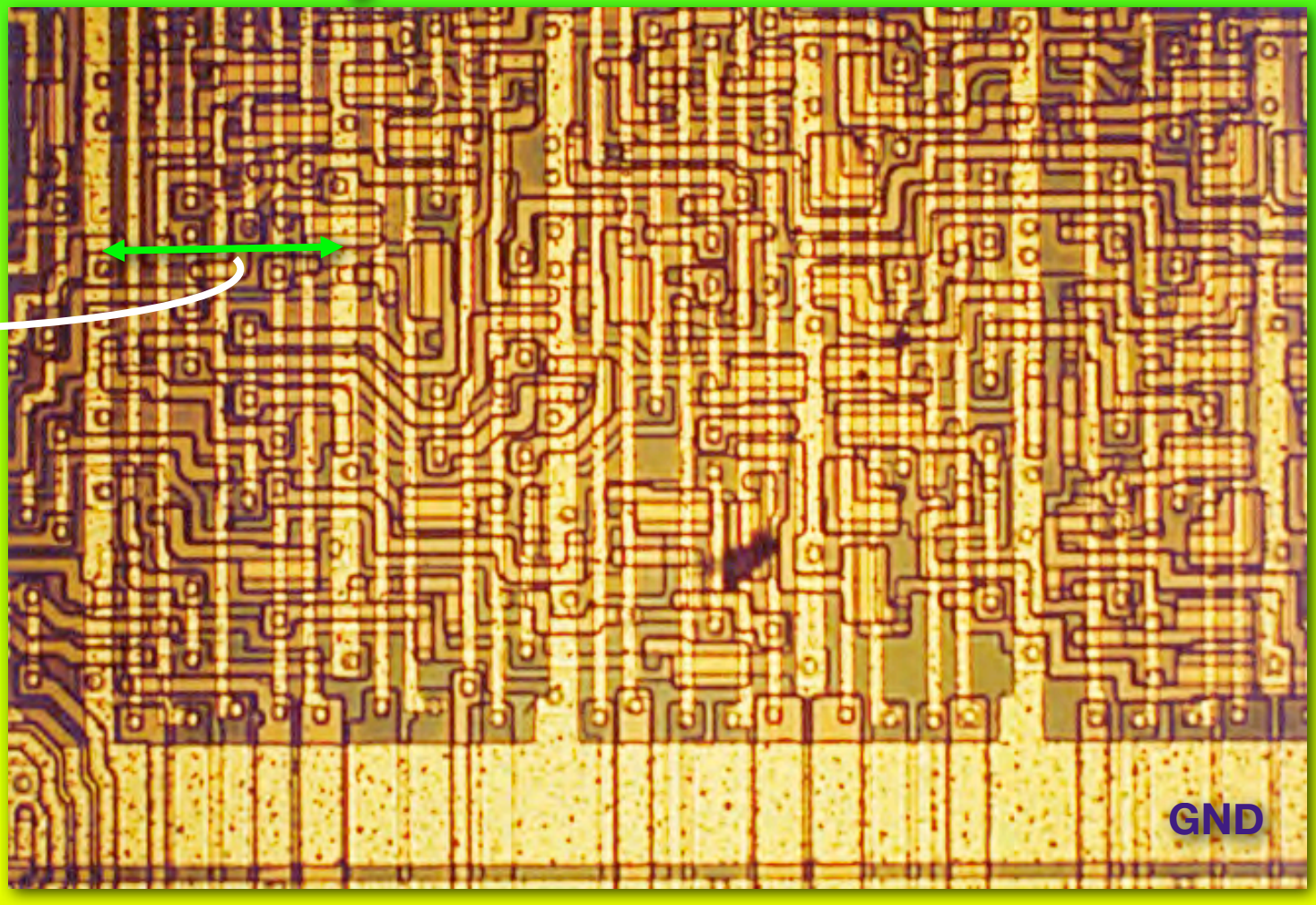
Full Custom



Zilog Z8000

detail of the control part using random logic

Full Custom



Detail of the control part of TMS7000 implemented with random logic

Standard Cell Approach

Cell On-the-fly Approach

Transistor Level Design Automation

challenge

To develop a CAD system for the automatic physical design of integrated circuits tuned for the requirements of submicron technologies:

smaller area → (wirelenght reduction)
smaller delay ← (wirelenght reduction)
less power consumption

Connections becomes
the central problem !

Challenge:
how to reduce wirelength?

Challenge:

how to reduce wirelength?

- area reduction
- use of complex gates (SCCG)
- improvement of routing and placement algorithms

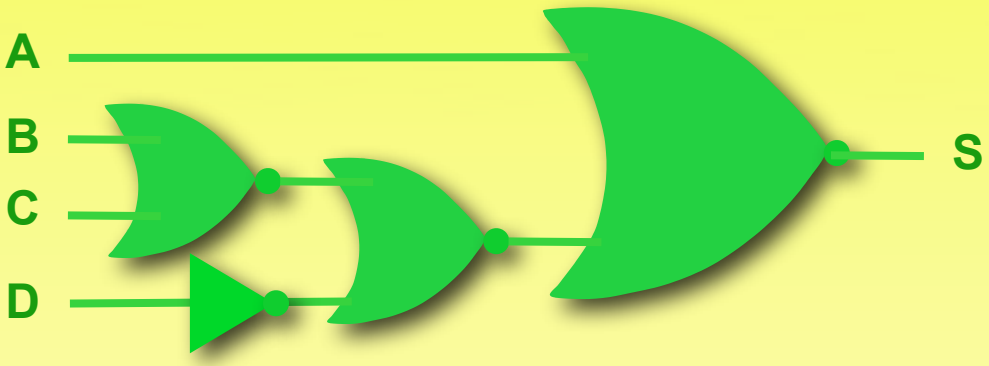
Using Static CMOS Complex Gates (SCCG)

with cell generation on-the-fly

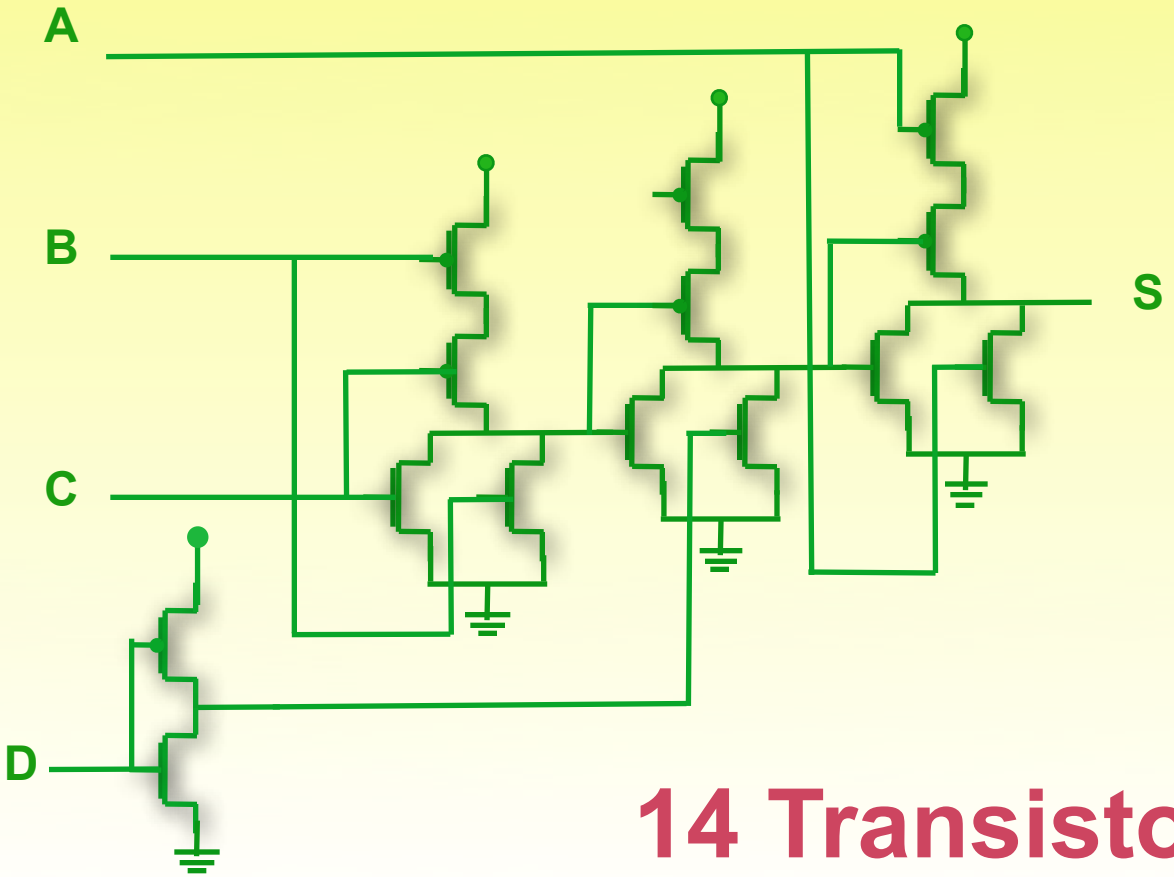
It is possible do to an extreme logic minimization

Freedom to Logic Designers !!!!

Example



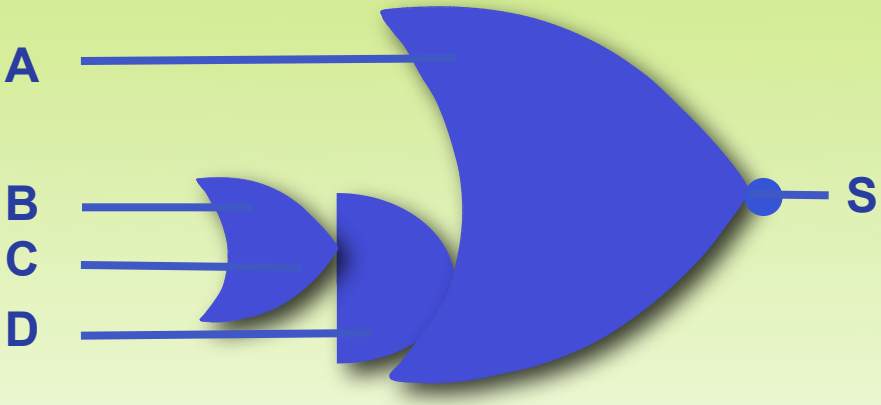
$$S = \overline{A + (\overline{B} + \overline{(C+D)})}$$

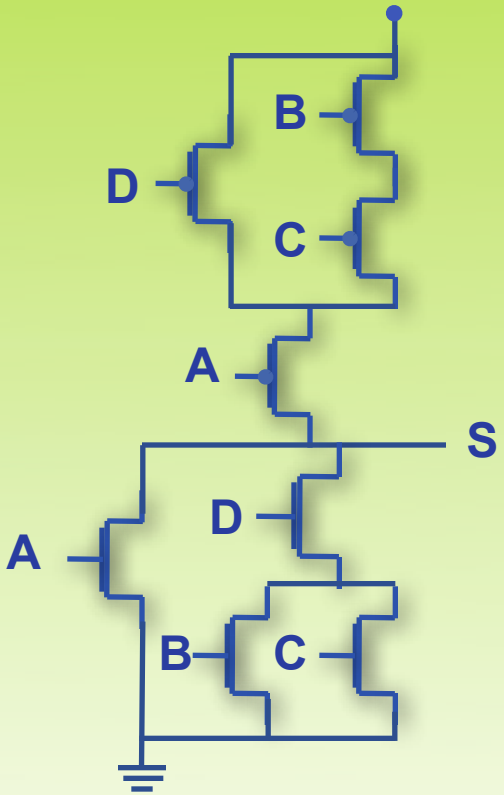


14 Transistors

Use of SCCG

$$S = \overline{A + (\overline{B} + \overline{(C+D)})} \quad \Rightarrow \quad S = \overline{A + (B \,.\, (C+D))}$$



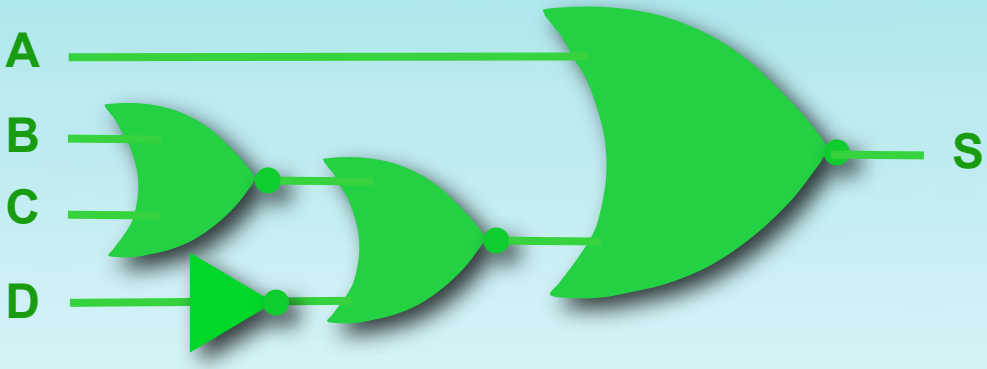


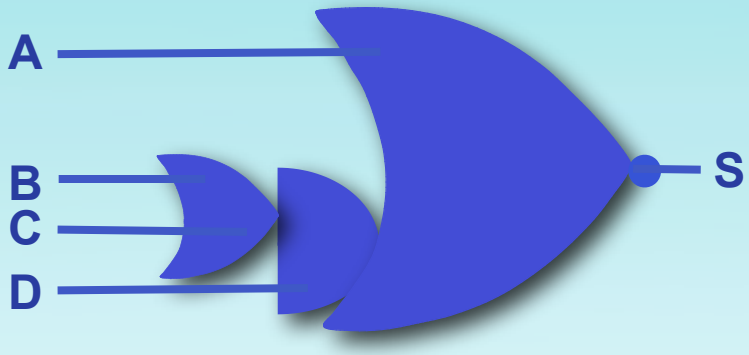
8 Transistors

Use of SCCG

$$S = \overline{A + (\overline{B} + \overline{(C+D)})}$$

$$S = \overline{A + (B \,.(C+D))}$$





14 Transistors

8 Transistors

Automatic Layout Synthesis Using Complex Gates (SCCG)

NUMBER OF SERIAL NMOS TRANSISTORS \ NUMBER OF SERIAL PMOS TRANSISTORS	1	2	3	4	5
1	1	2	3	4	5
2	2	7	18	42	90
3	3	18	87	396	1677
4	4	42	396	3503	28435
5	5	90	1677	28435	125803

Power Reduction

$I_{leakage}$ is become important in submicron circuits.

It is function of the number of transistors

Routing

the solutions produced by academic and industrial tools are in average within **1.43 to 2.38 times** the optimal solutions considering **wirelength**

C-C. Chang, J. Cong, M. Xie.
"Optimality and Scalability of Existing Placement Algorithms".
ASPDAC 2003

FOTC Routing

FOTC approach (Full-Over-The-Cell Routing)

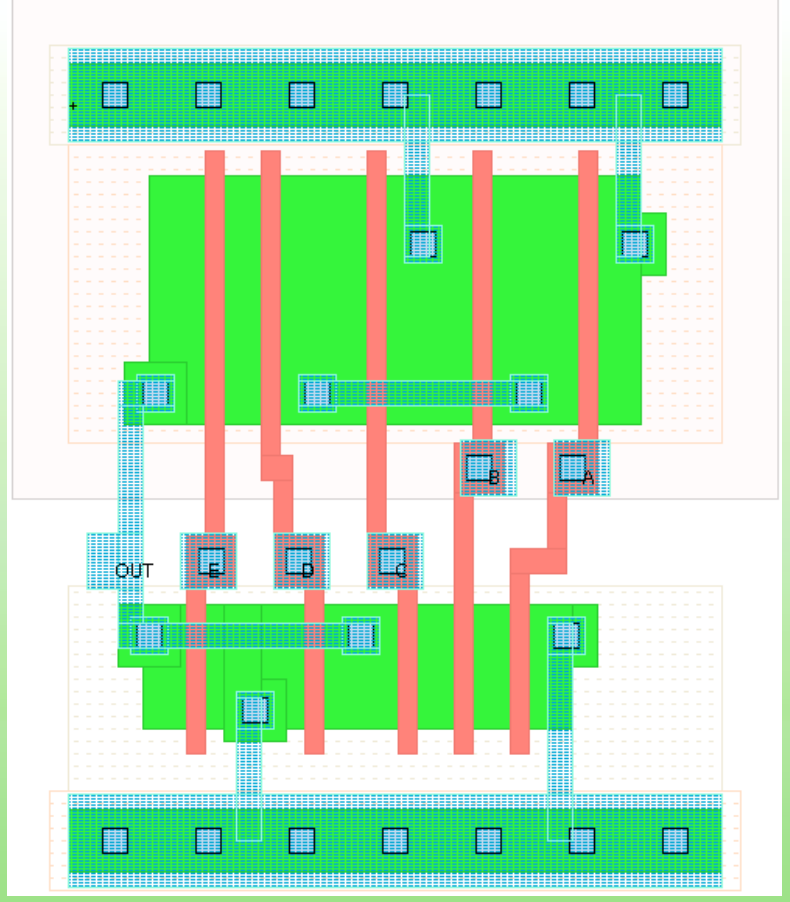
All conections are over the active zones

Layout Strategies

Layout Strategies

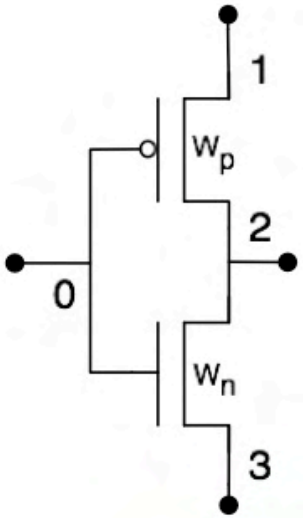
- transistor topologies
- management of routing in all layers
- VCC and Ground distribution
- clock distribution
- contacts and vias management
- body ties management
- transistor sizing and folding

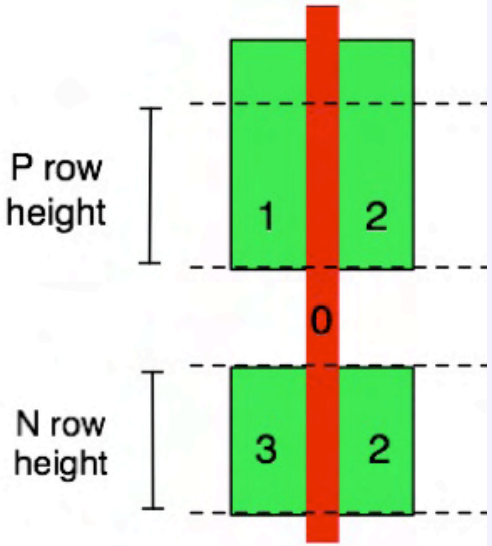
Layout Strategies



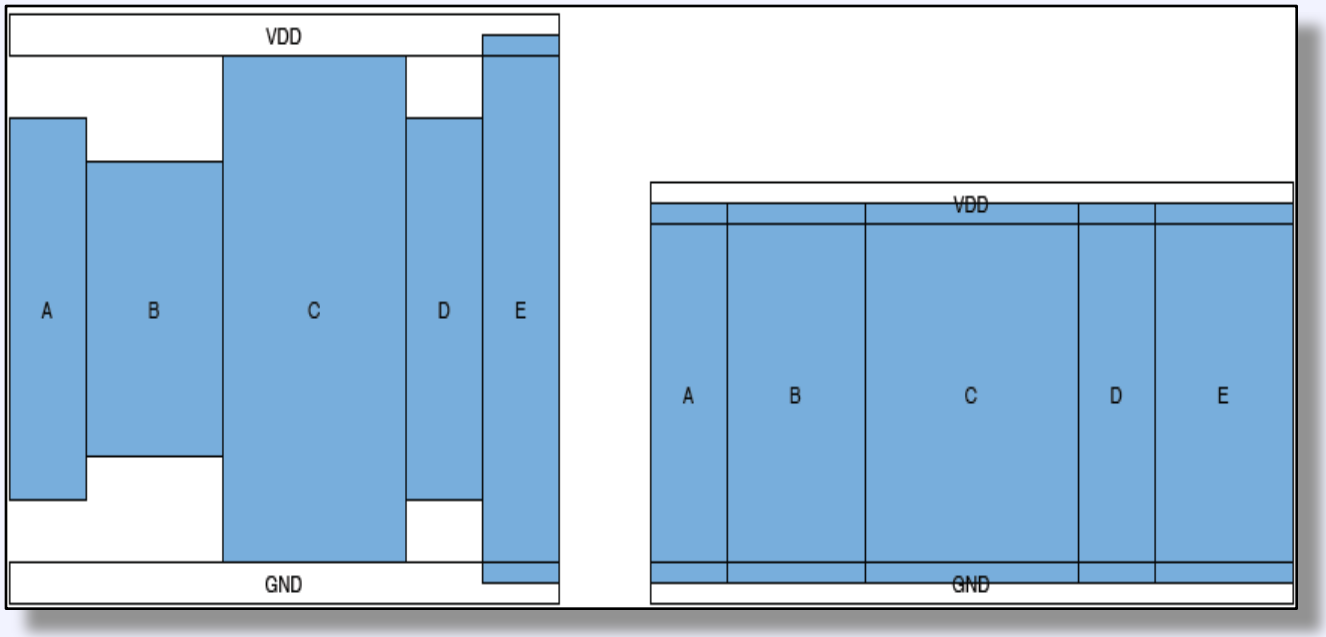
Transistor topologies
- horizontal
- vertical
- doglegs (different directions)
- folding

Transistor Folding

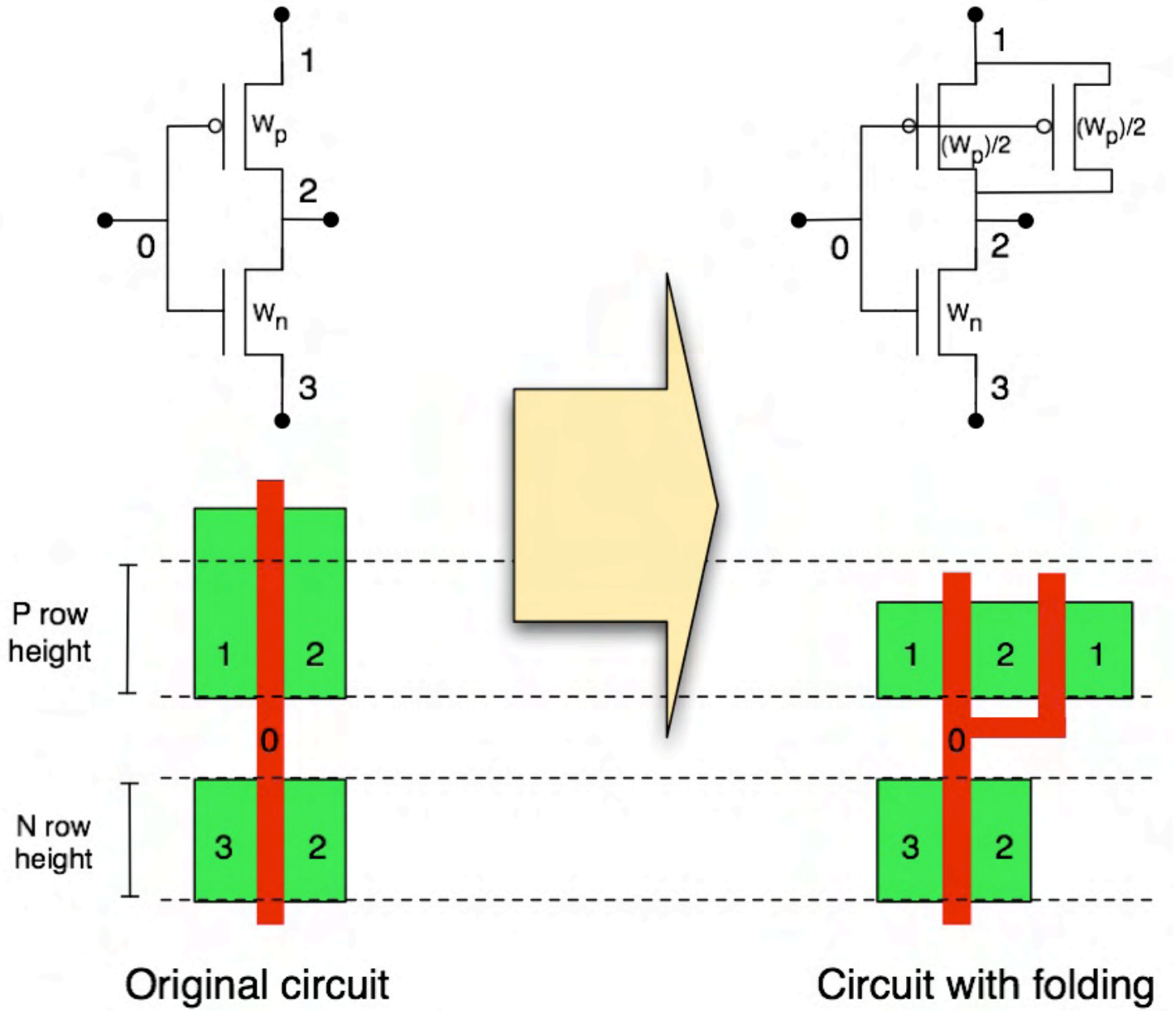




Original circuit



Transistor Folding



Layout Strategies

Routing Management
- priority tracks schema
- routing layers priority
- routing layers directions

Layout Strategies

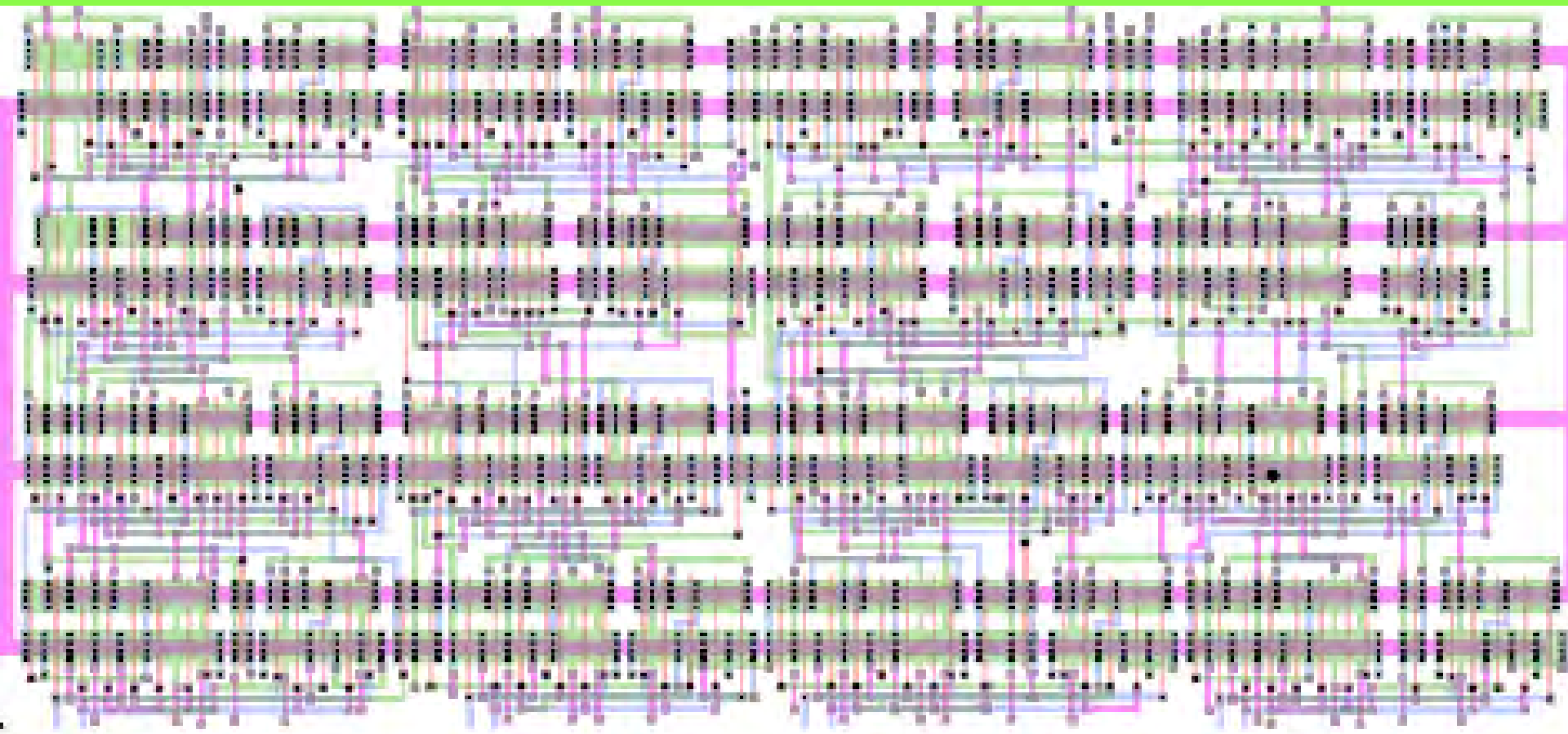
VCC and Ground Distribution

- borders of the strip
- middle of strips

  (between P and N diffusions)
- over the transistors

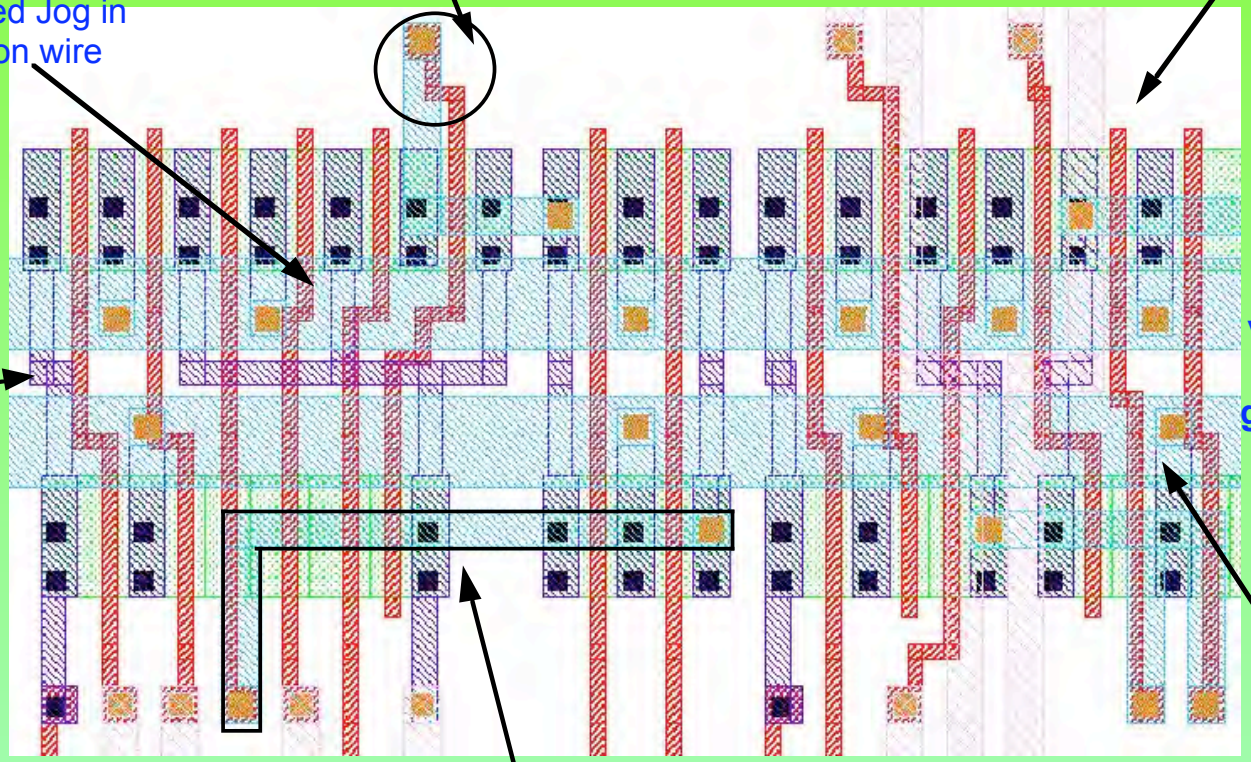
Layer (metal 1, metal 2,...)

Power Lines over the transistors

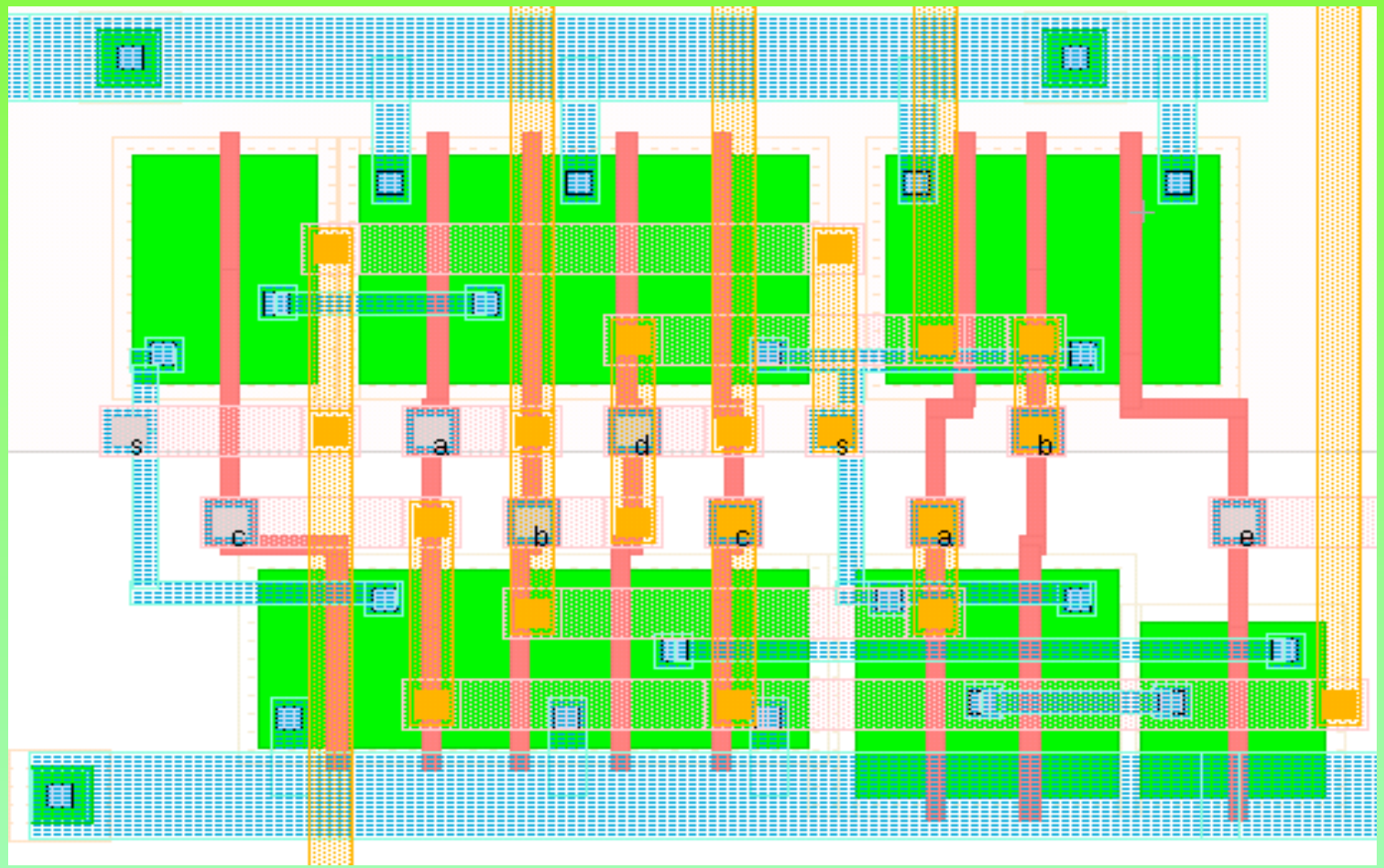


TROPIC3

Power Lines between P and N plans

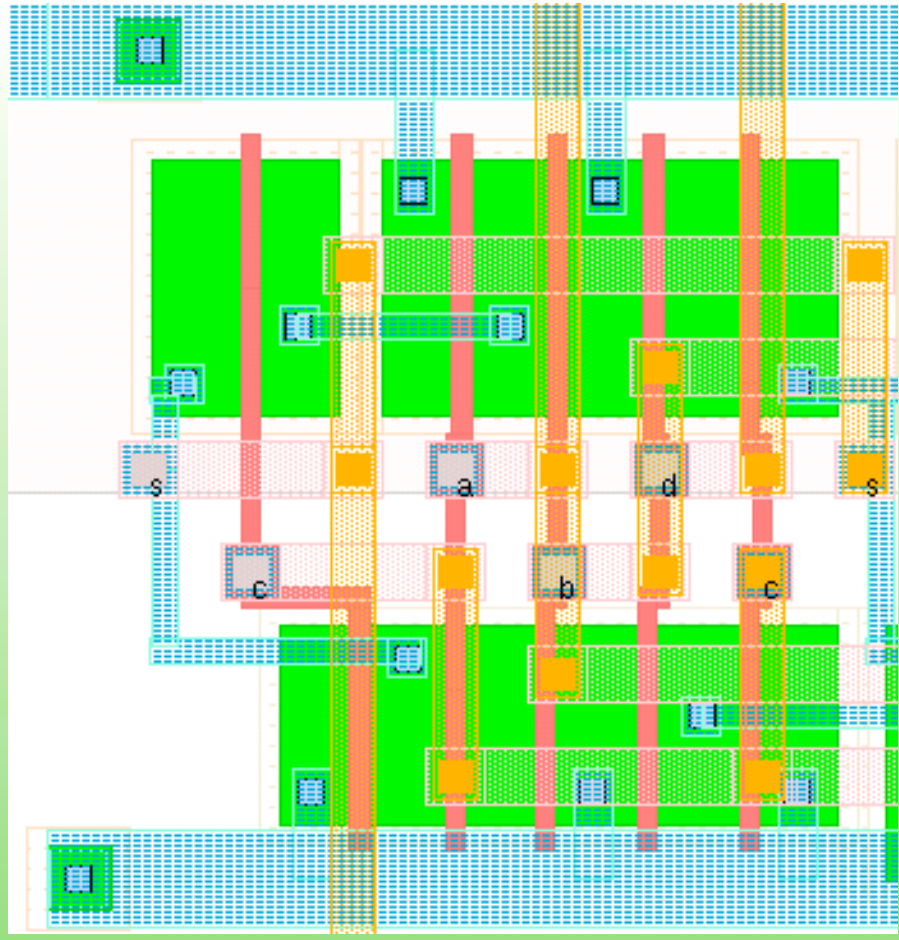


Power Lines at the Strip Borders



Parrot1

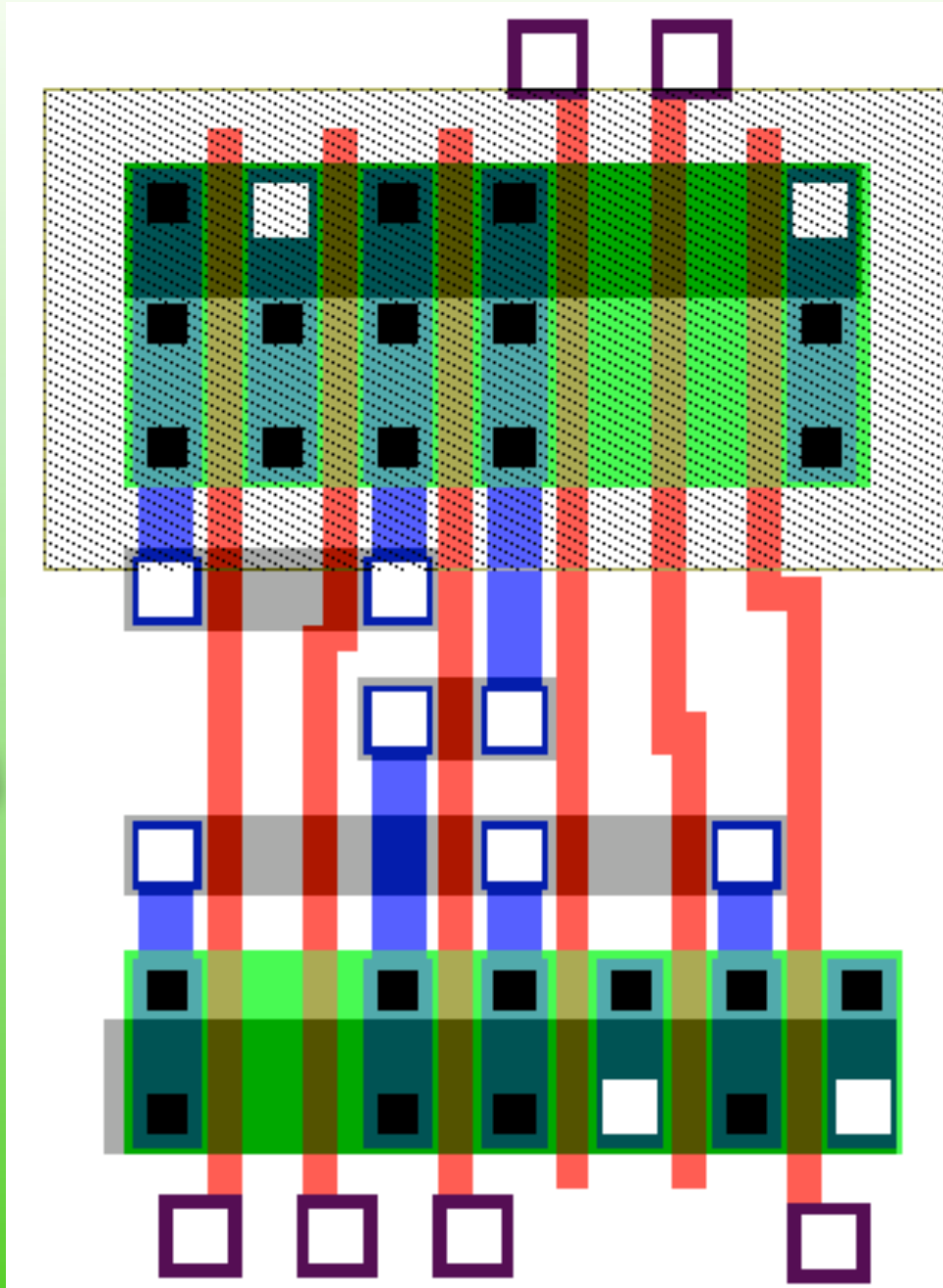
Layout Strategies



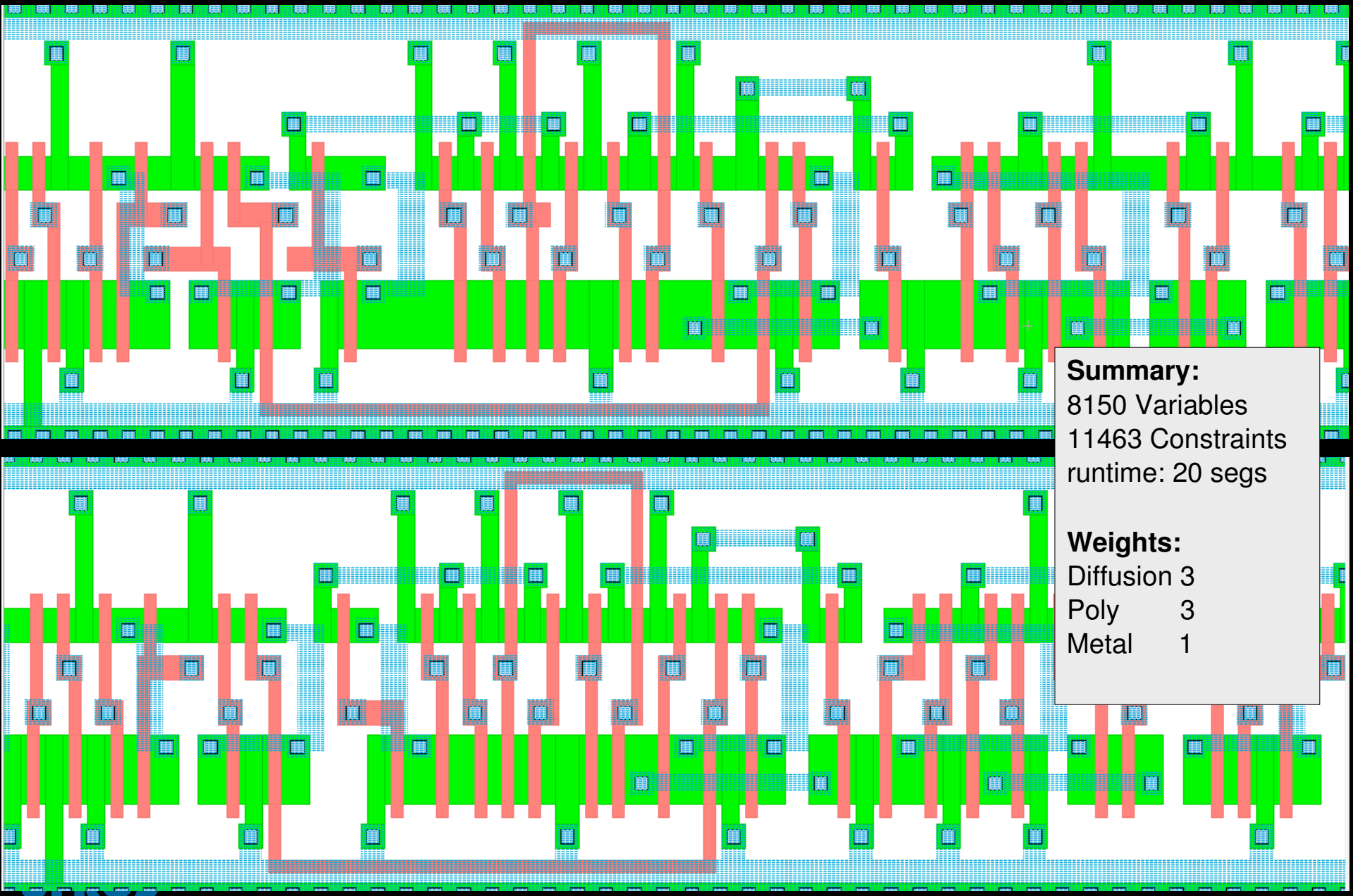
- contacts and vias management
- body ties management

Layout Strategies

contacts and vias management
(towers of vias)



Compaction Results

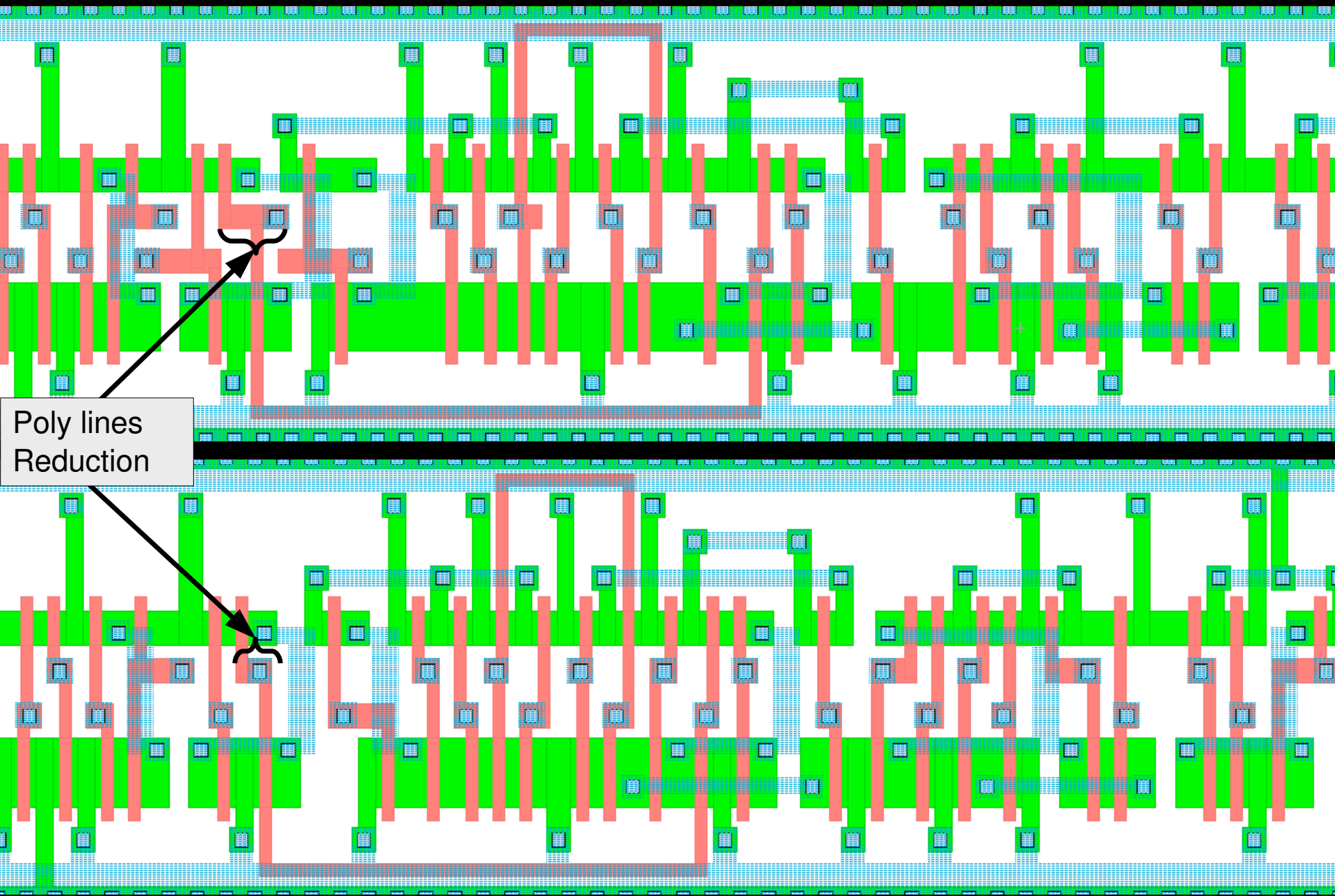


Summary:
8150 Variables
11463 Constraints
runtime: 20 segs

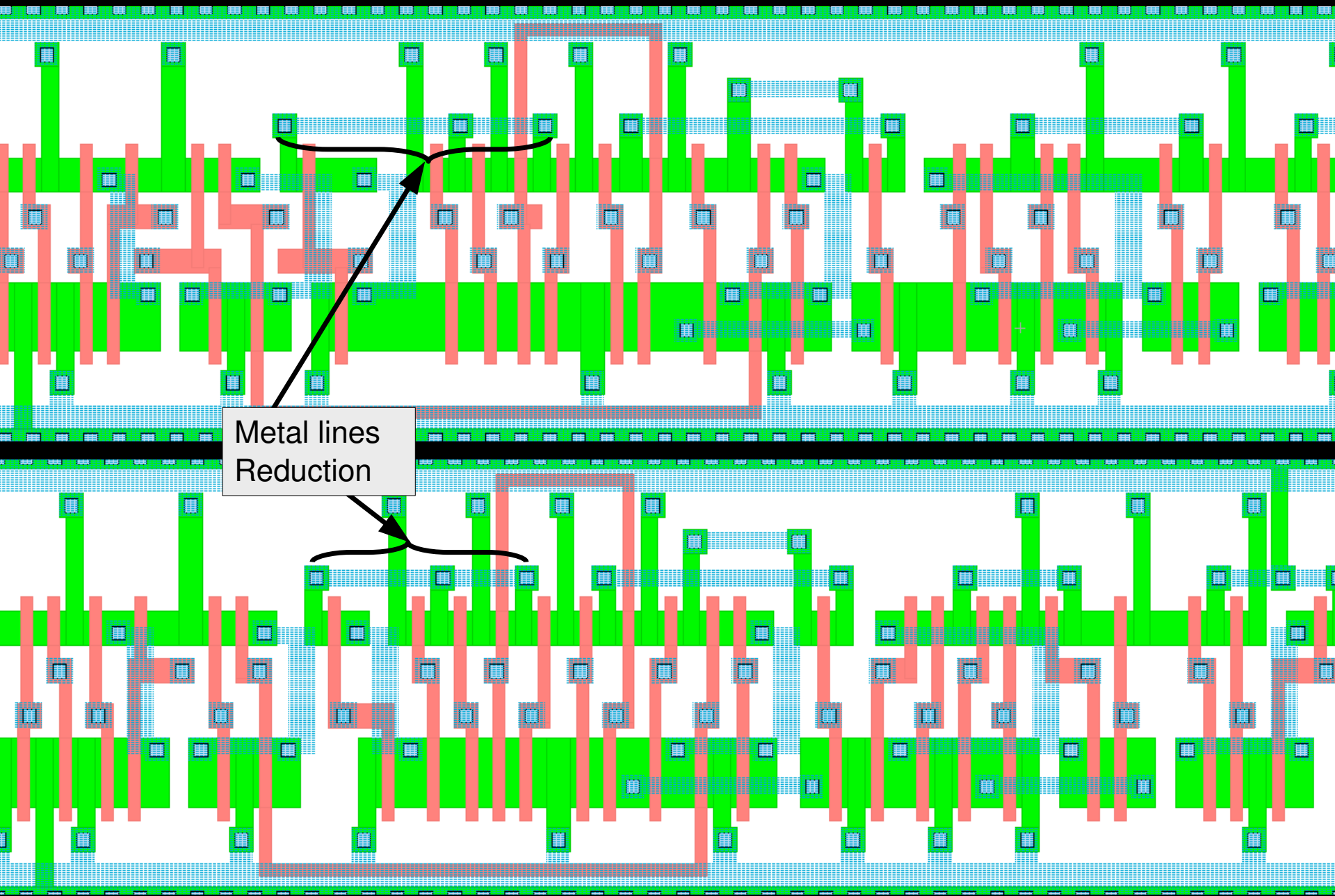
Weights:

Diffusion	3
Poly	3
Metal	1

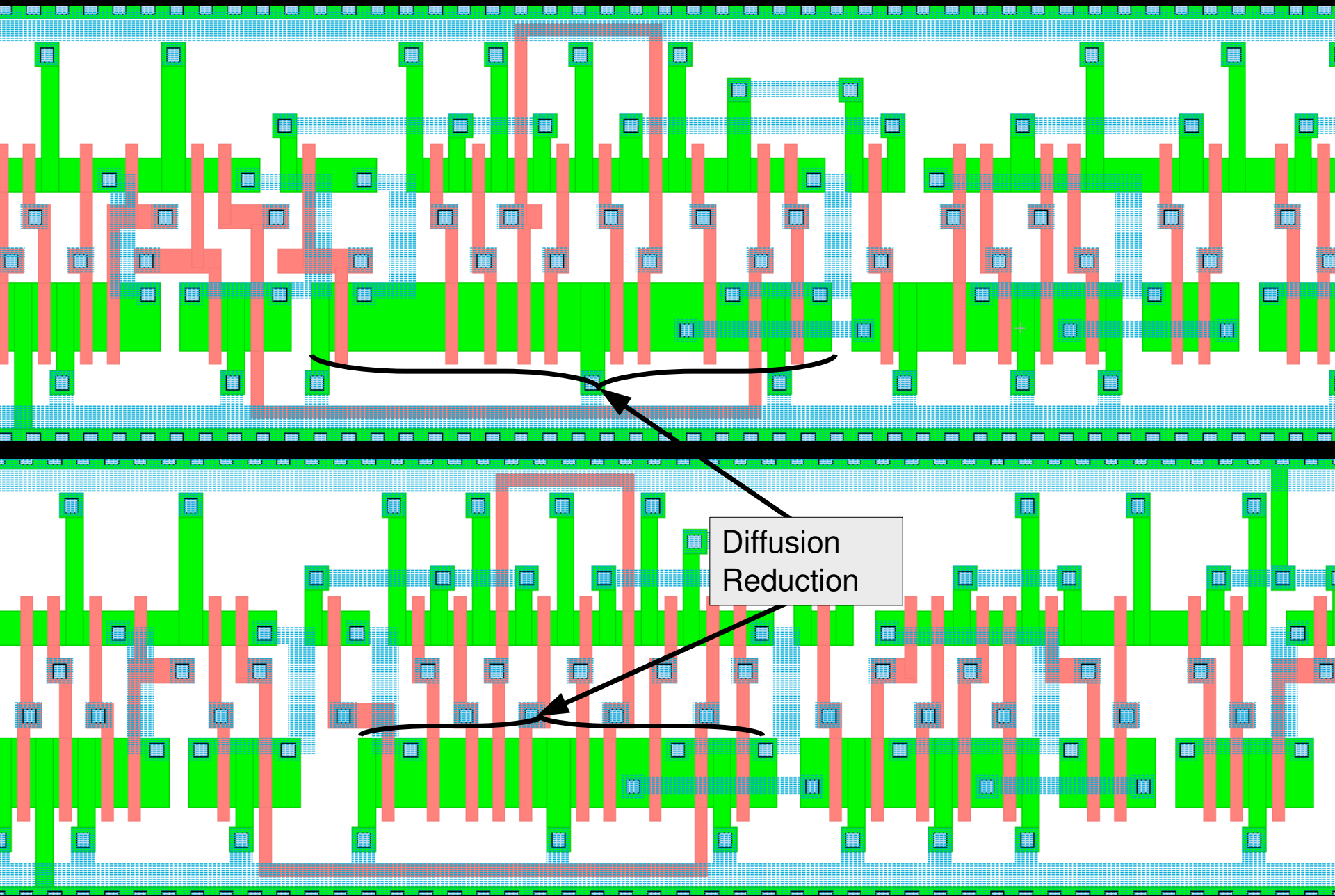
Compaction Results



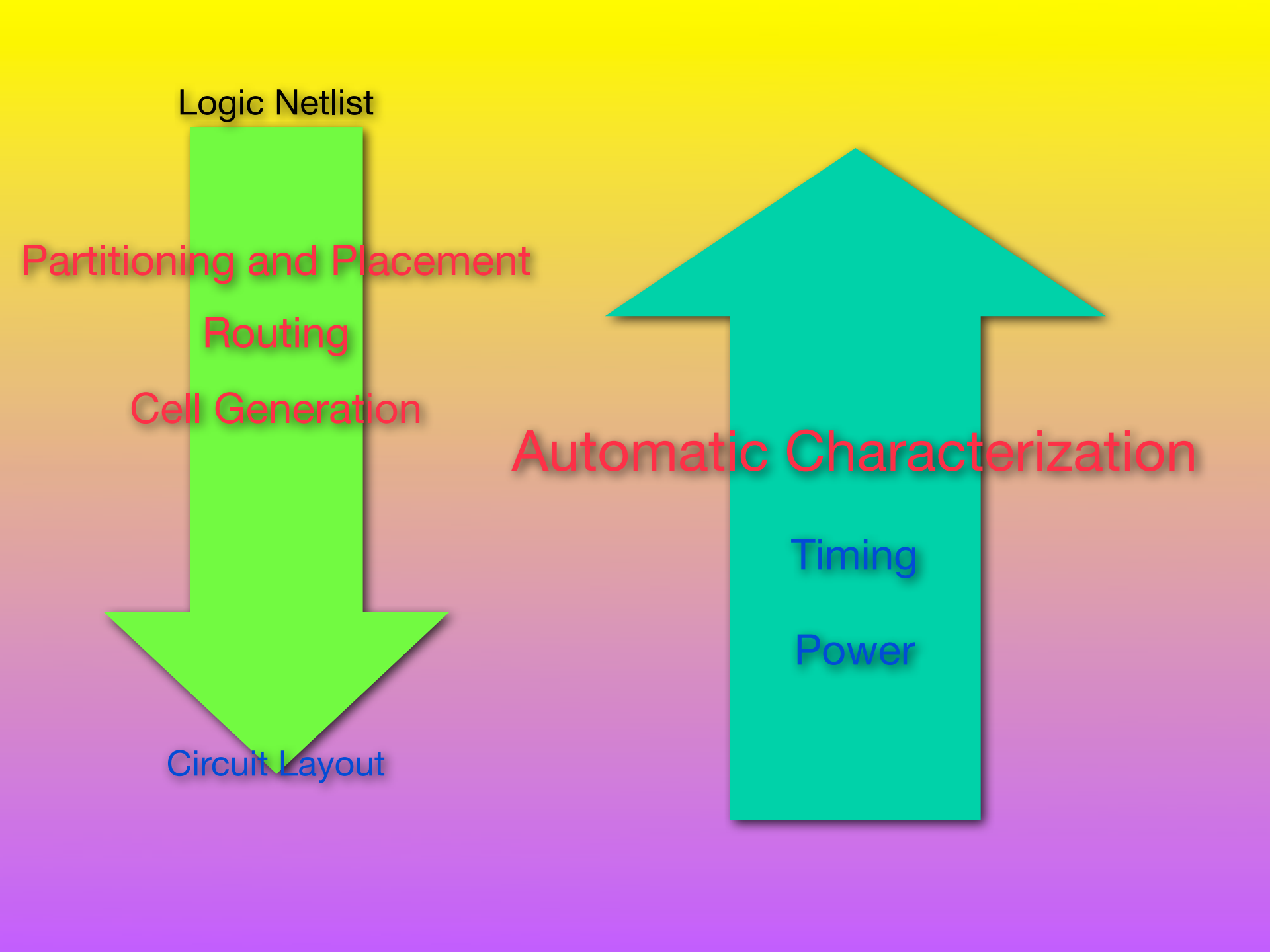
Compaction Results



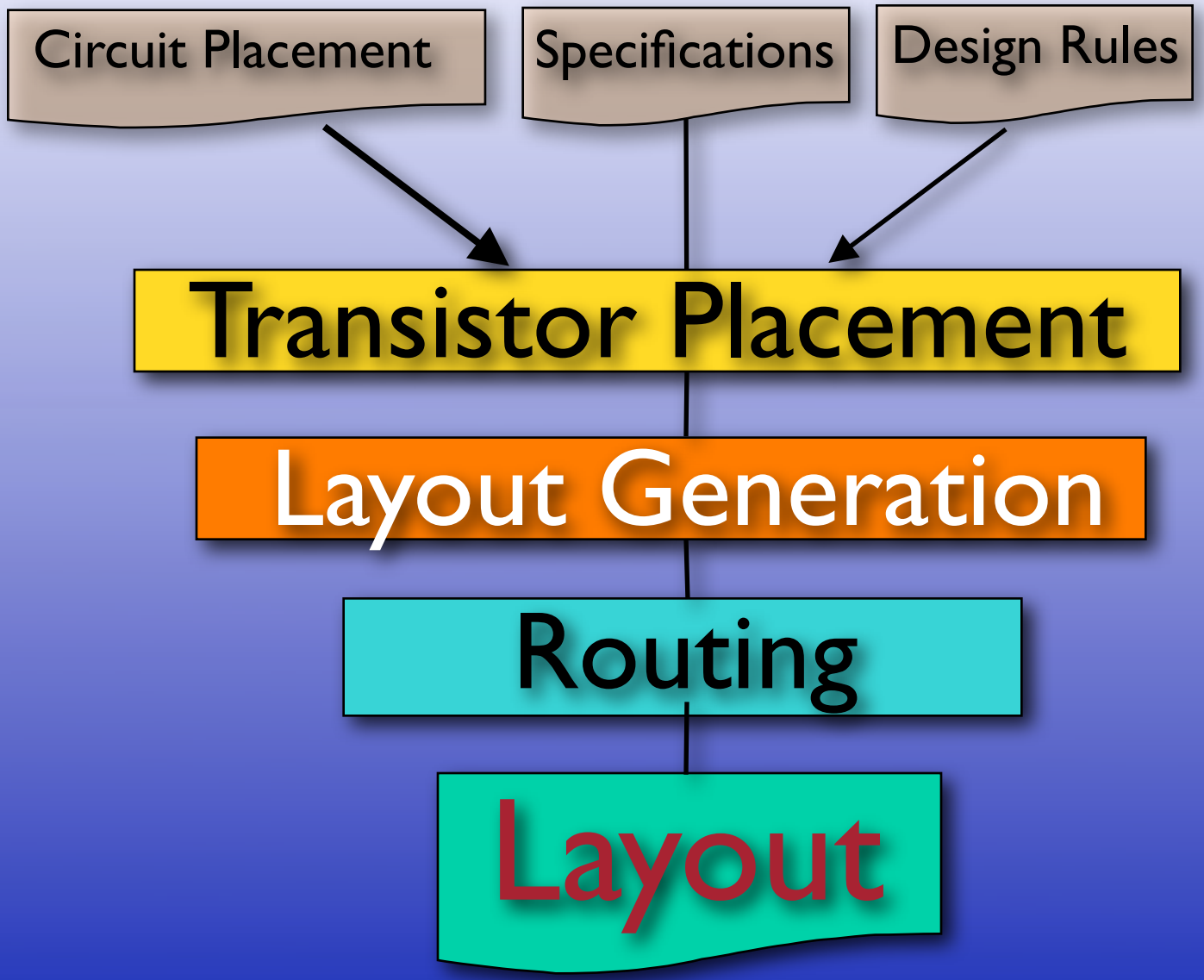
Compaction Results



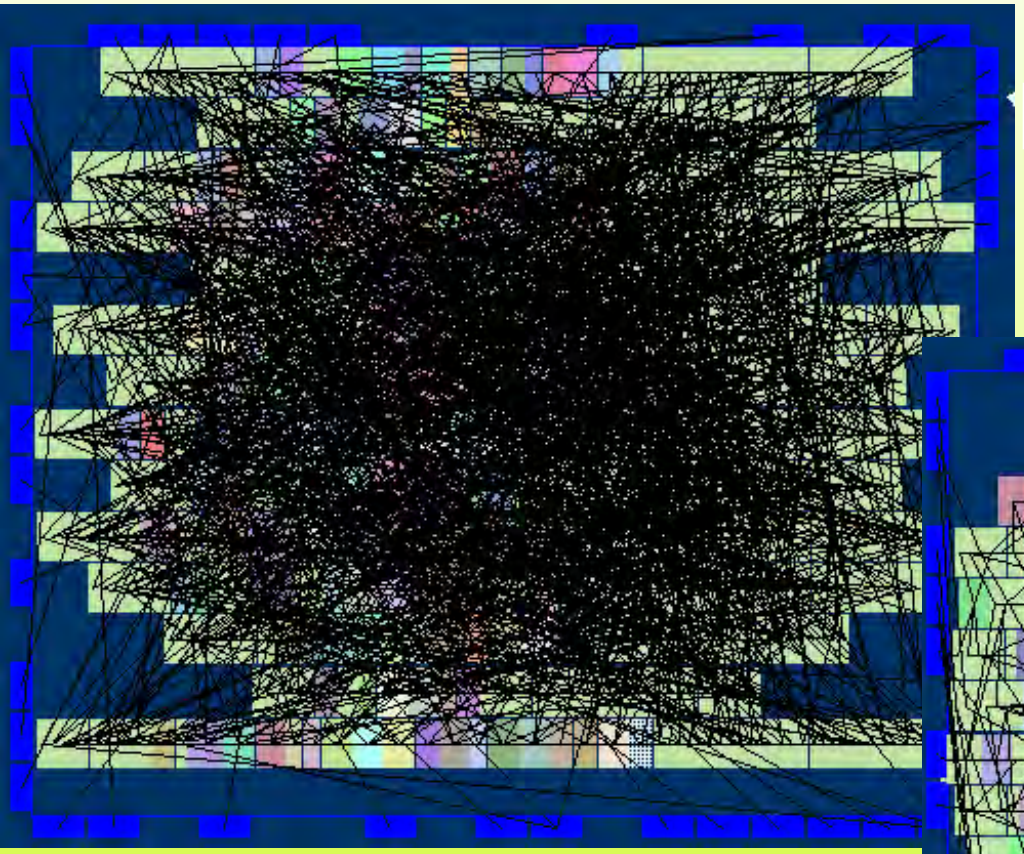
Physical Design Flow


Logic Netlist
Partitioning and Placement
Routing
Cell Generation
Circuit Layout
Automatic Characterization
Timing
Power

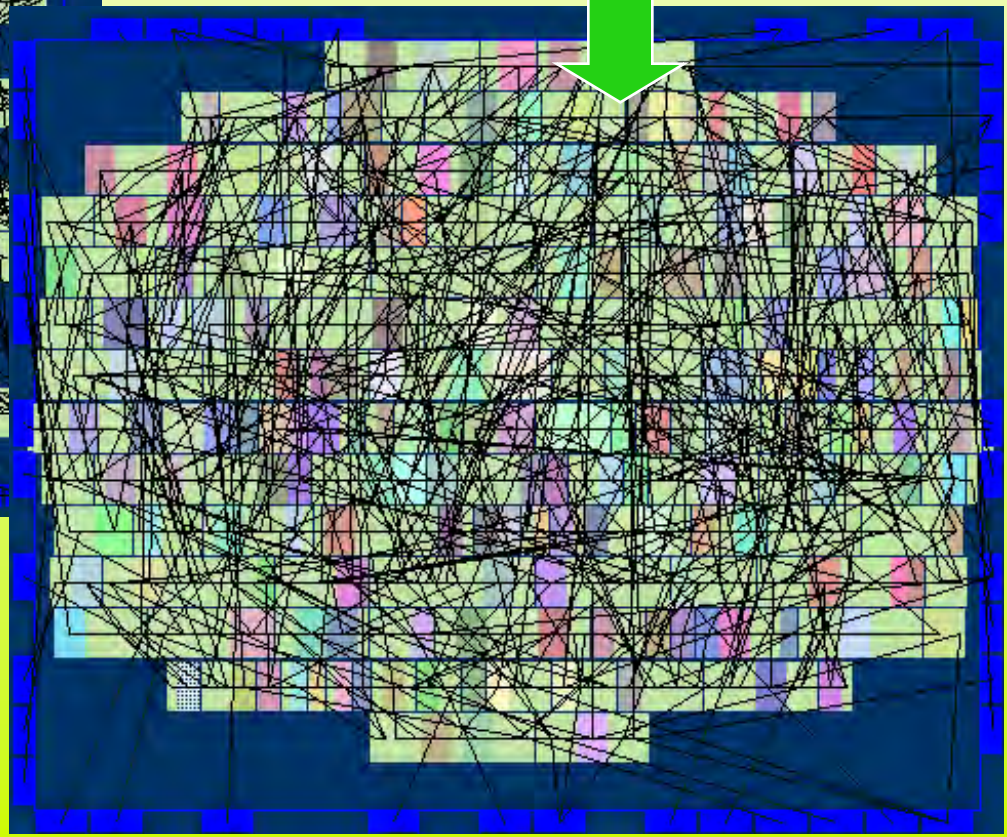
Layout Generation Flow

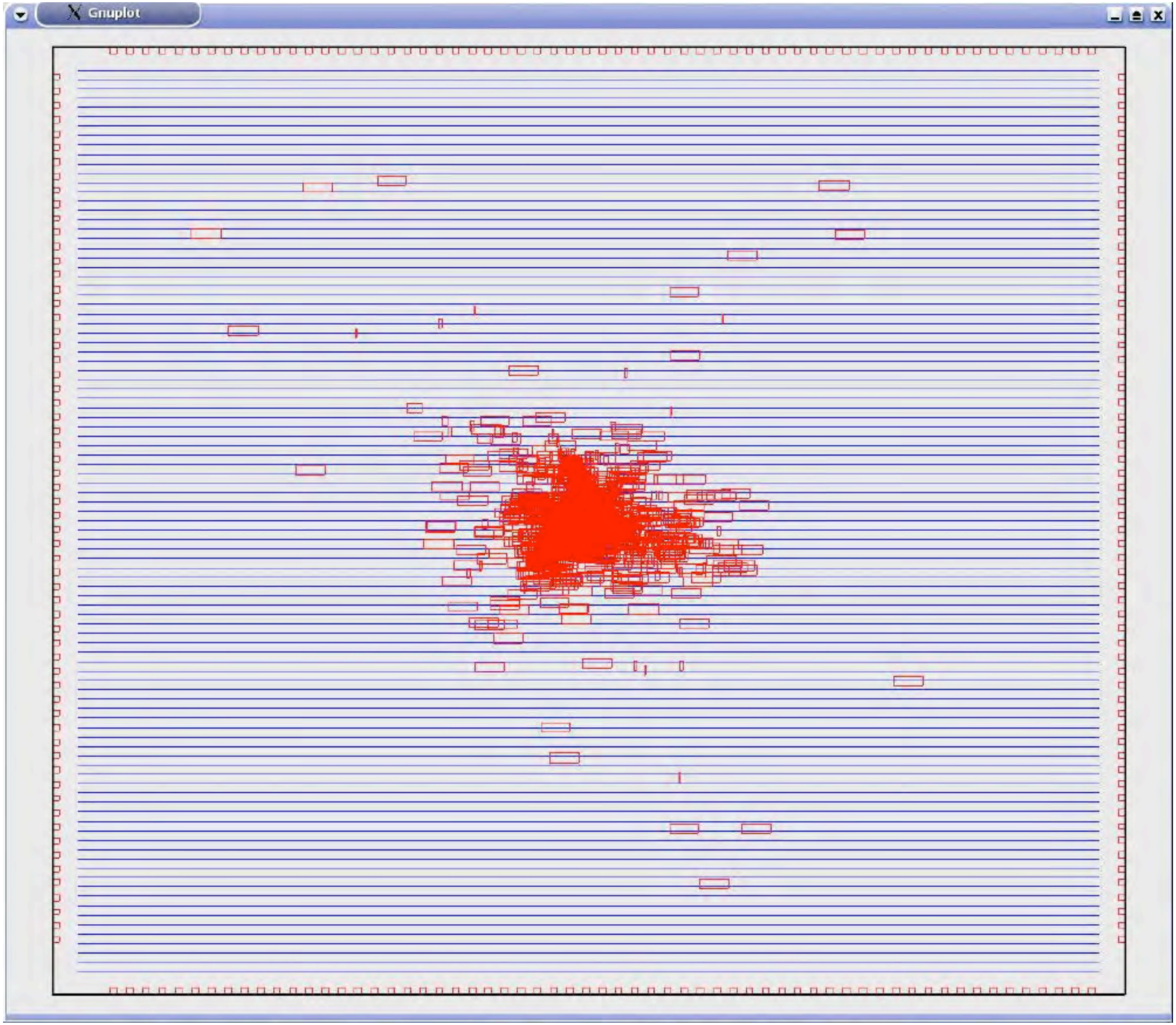


Wirelength

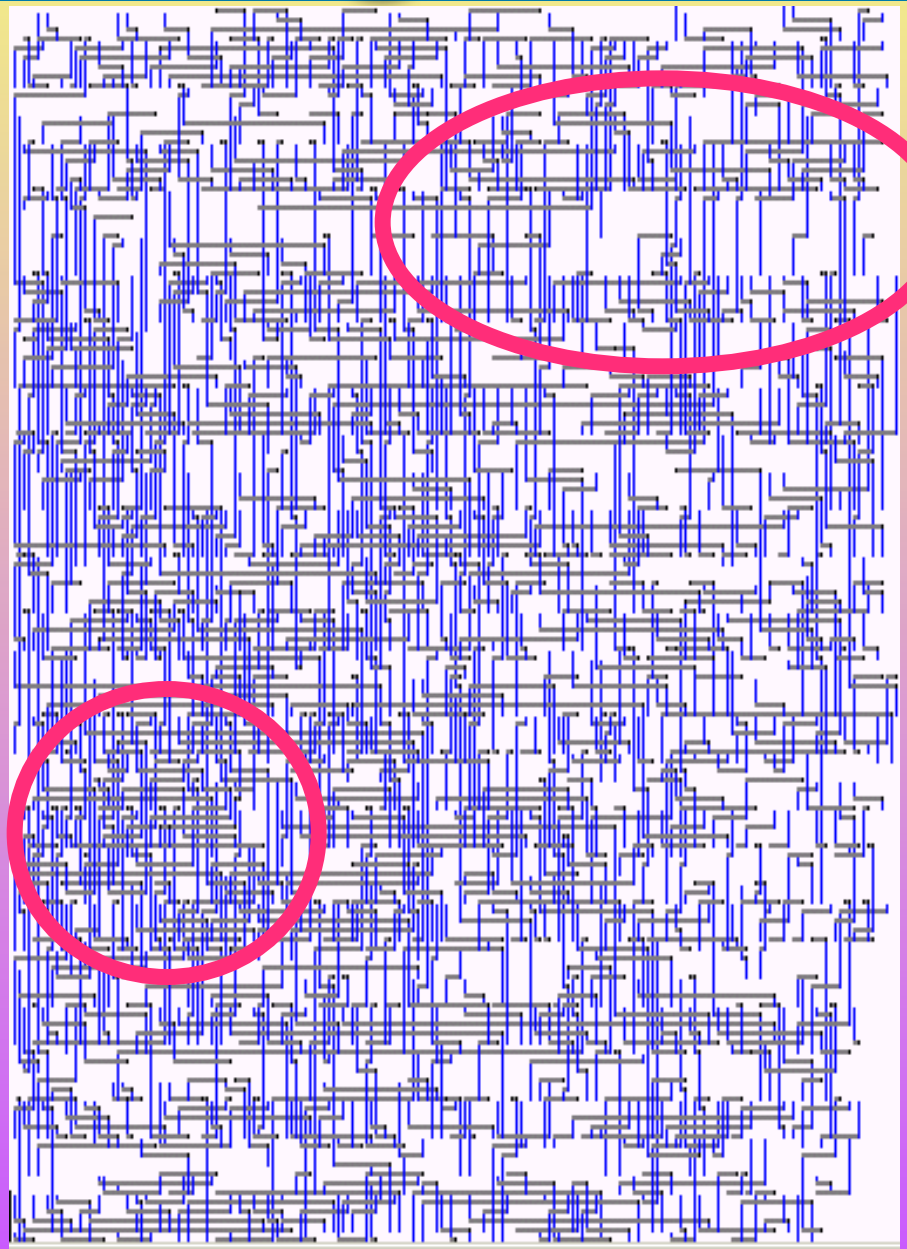


Placement with wirelength reduction




Gnuplot

Congestion



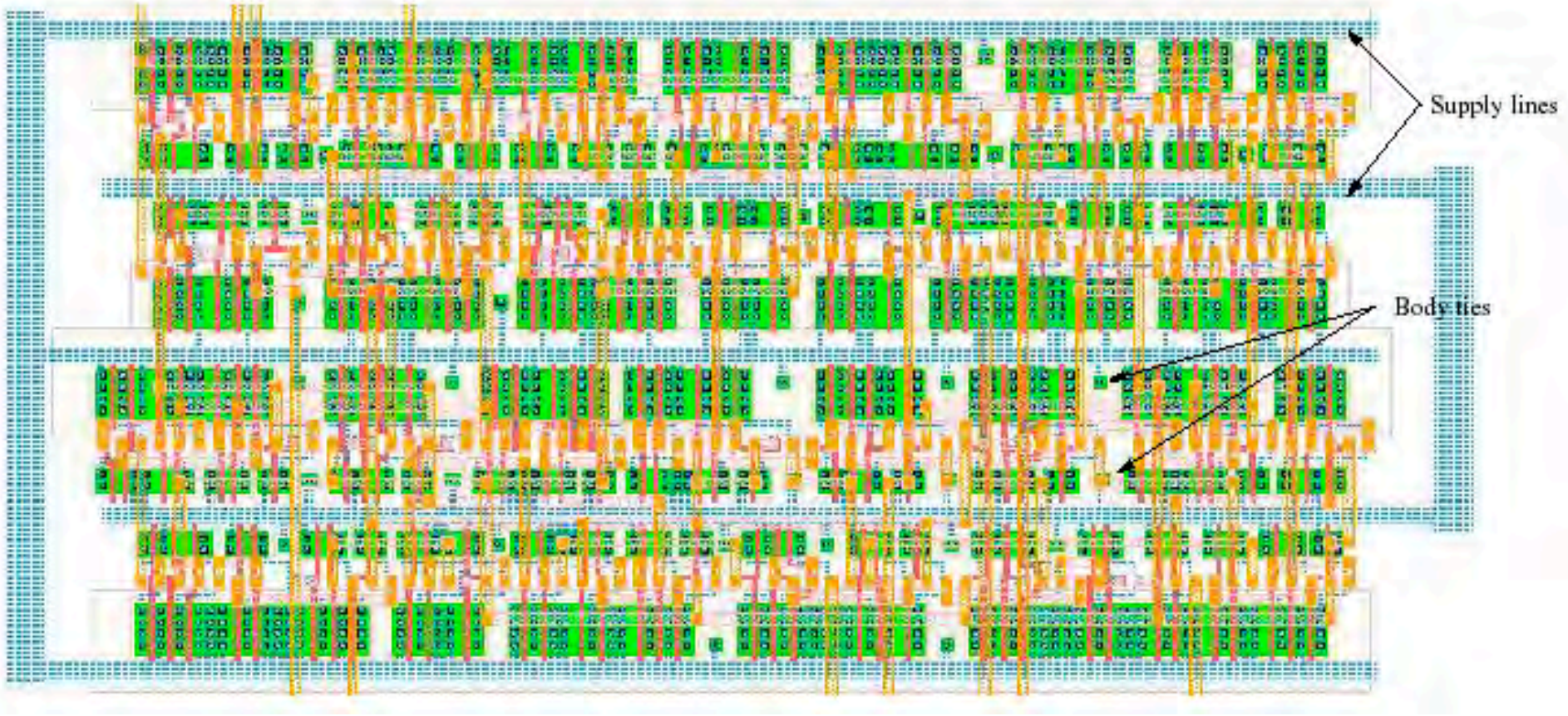
Congestion is an important problem because it can forbid a complete routing

Routability

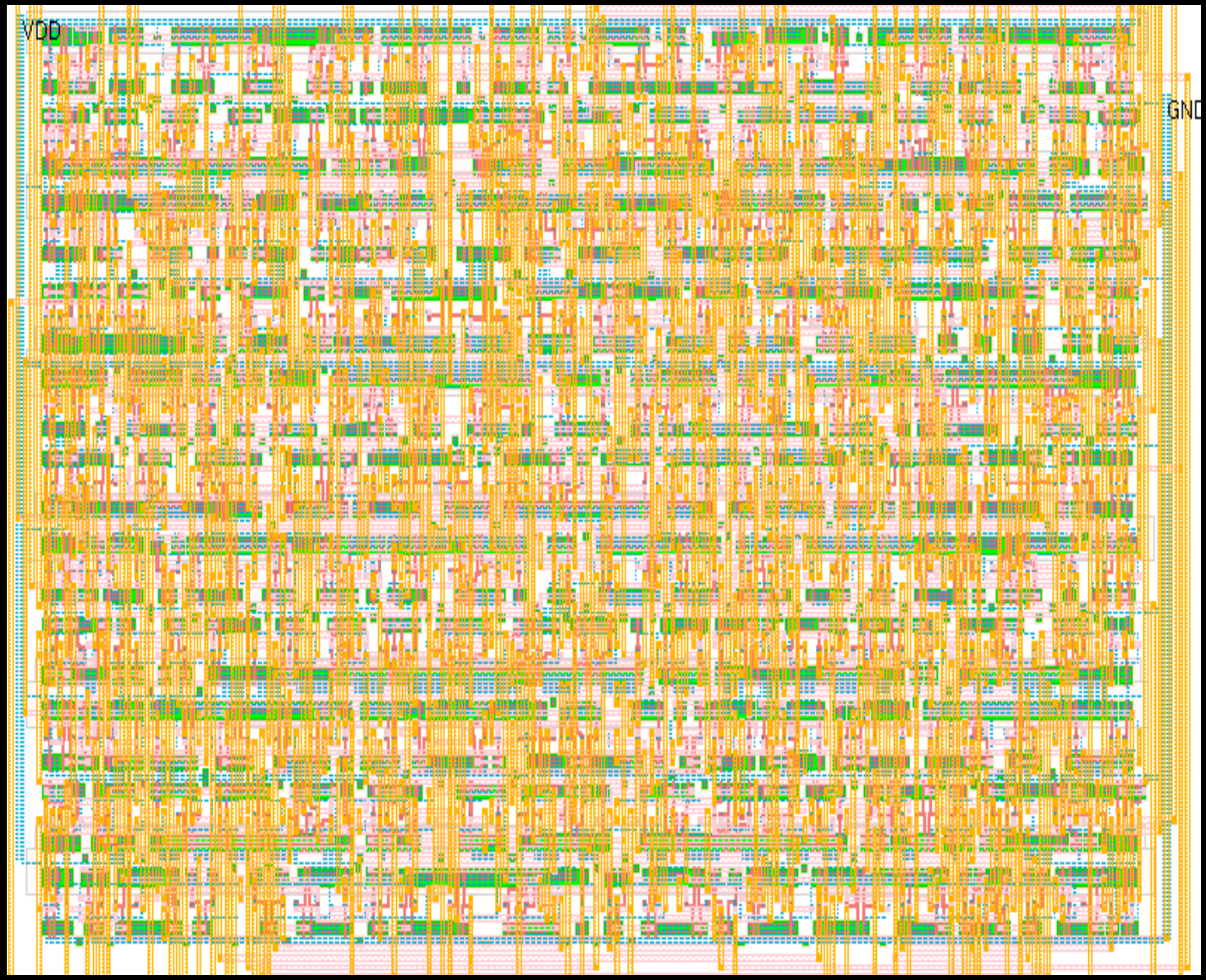
Compromise:

Routability and Wirelength Reduction

Parrot Layout Style



Layout Generated Automatically with Parrot Tool Suite

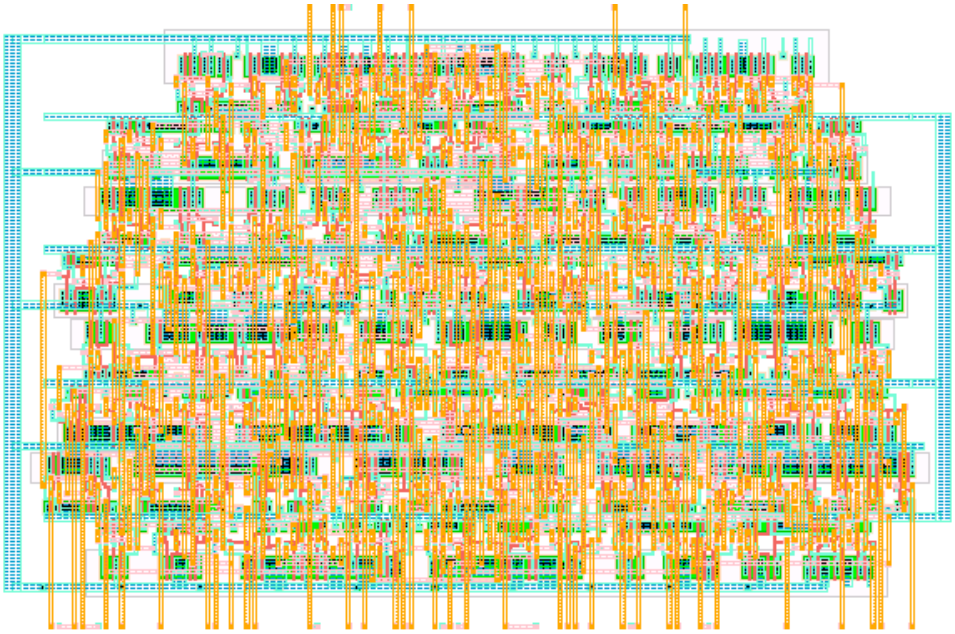


UFRGS

C432

TROPIC

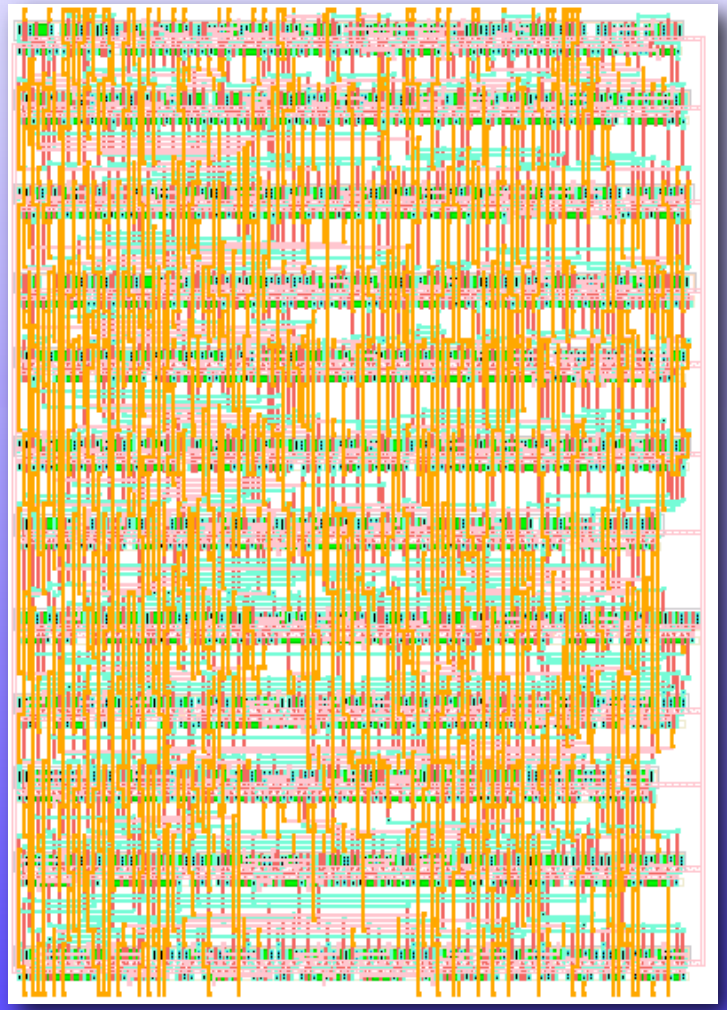
PARROT



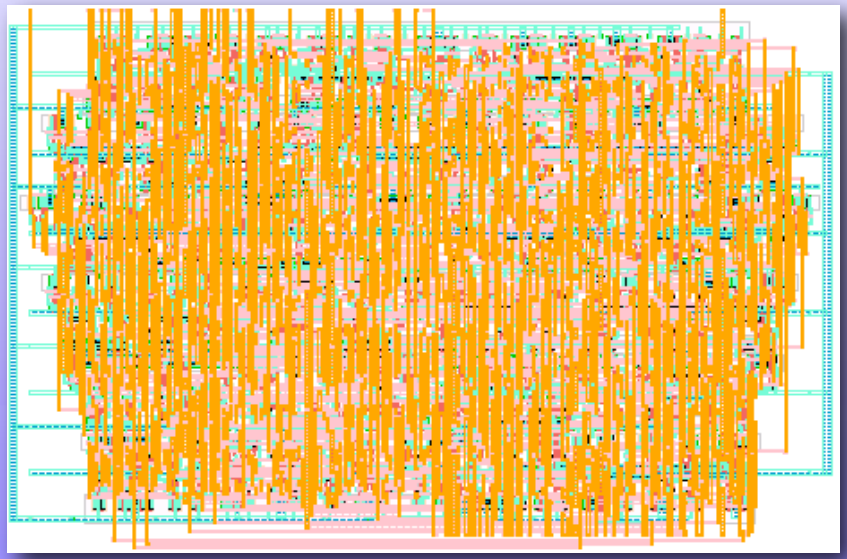
804 transistors
Delay: -26.6 %
Area: -41.3 %

C499

TROPIC



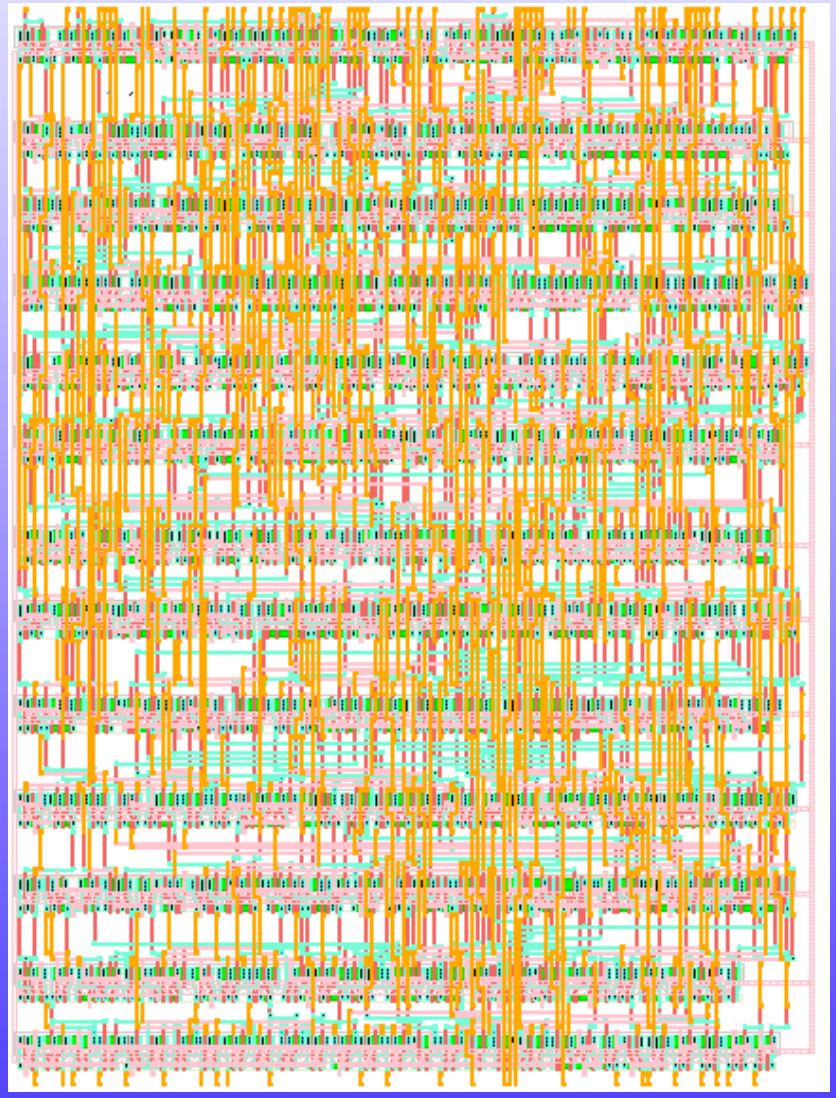
PARROT



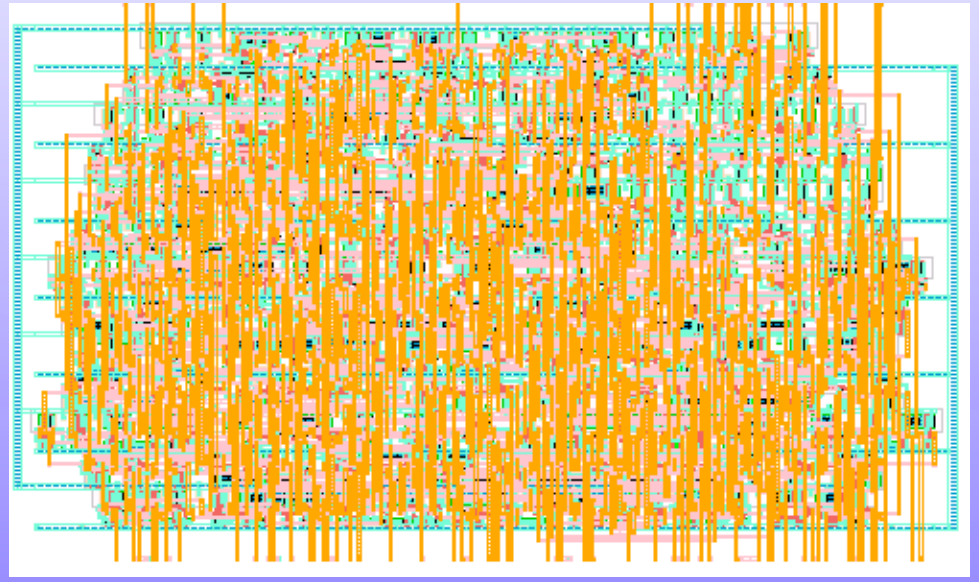
1556 transistors
Delay: - 26.0 %
Area: - 37.3 %

C880

TROPIC



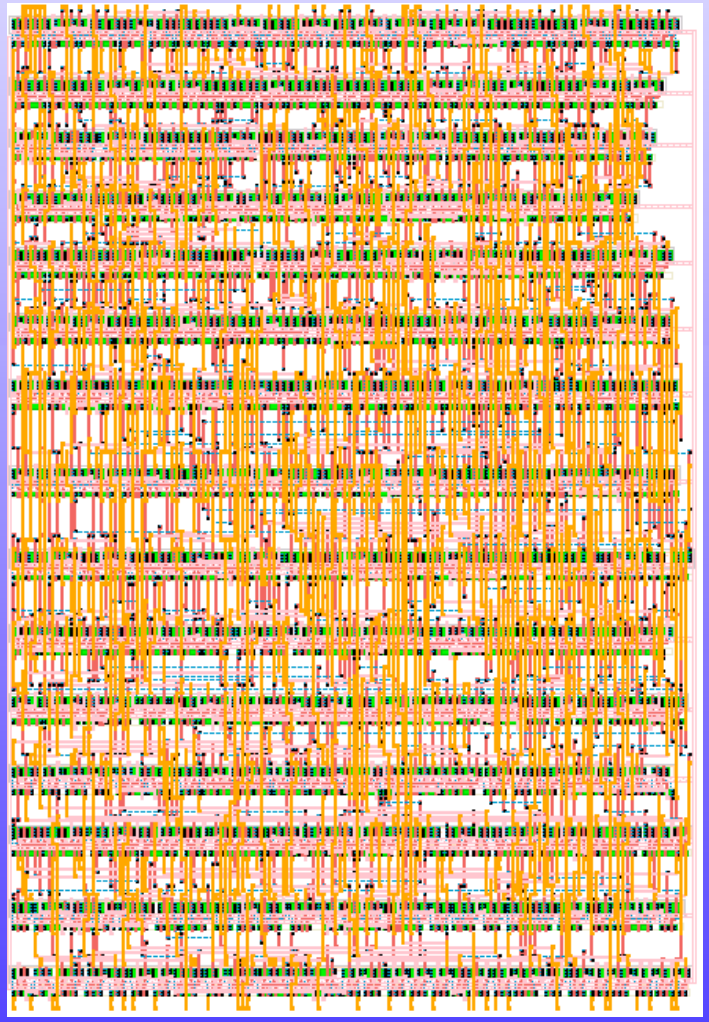
PARROT



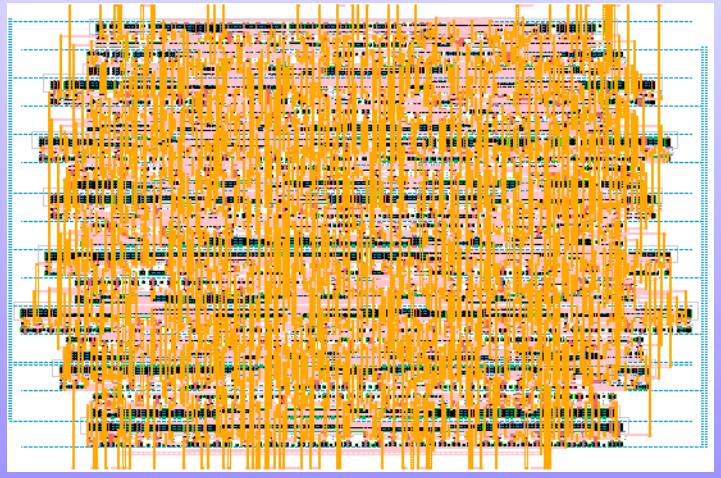
1802 transistors
Delay: -22.0 %
Area: -37.5 %

C1355

TROPIC

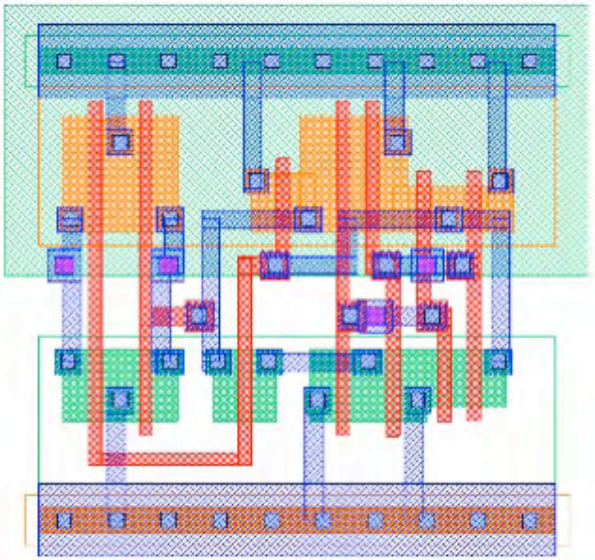


PARROT

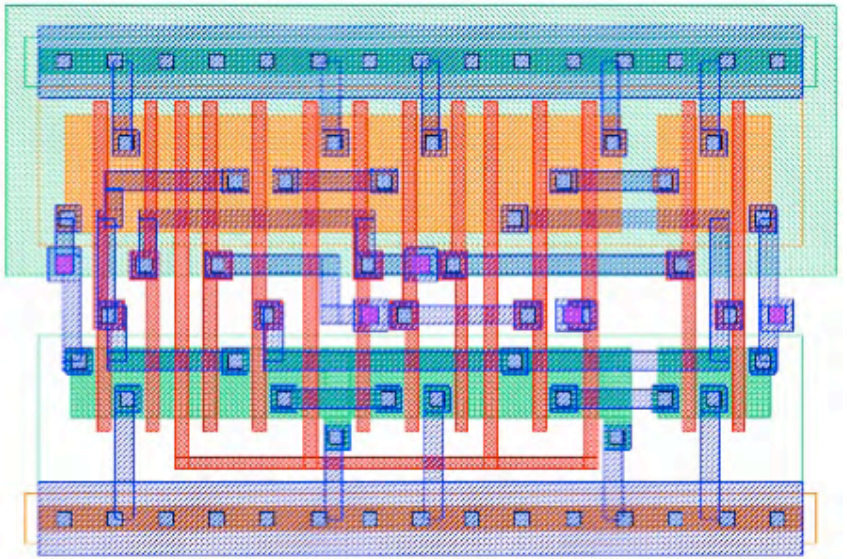


2308 transistors
Delay: -21.1 %
Area: -33.2 %

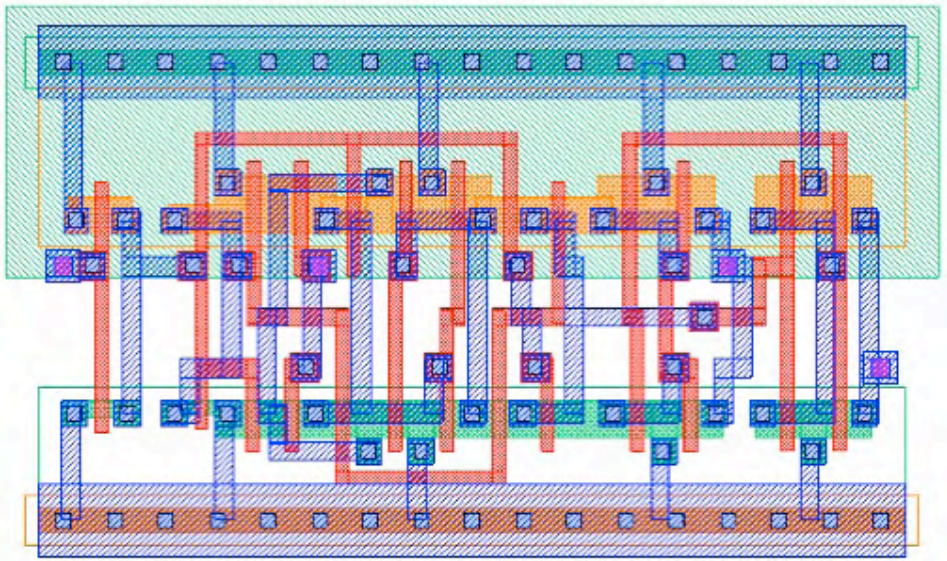
Results: Layouts



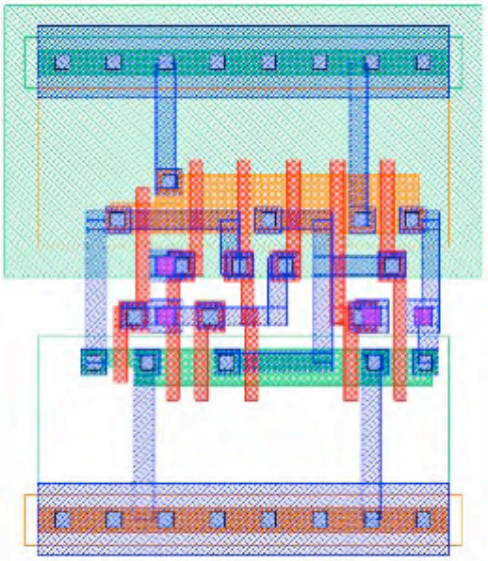
ADD22



ADD32

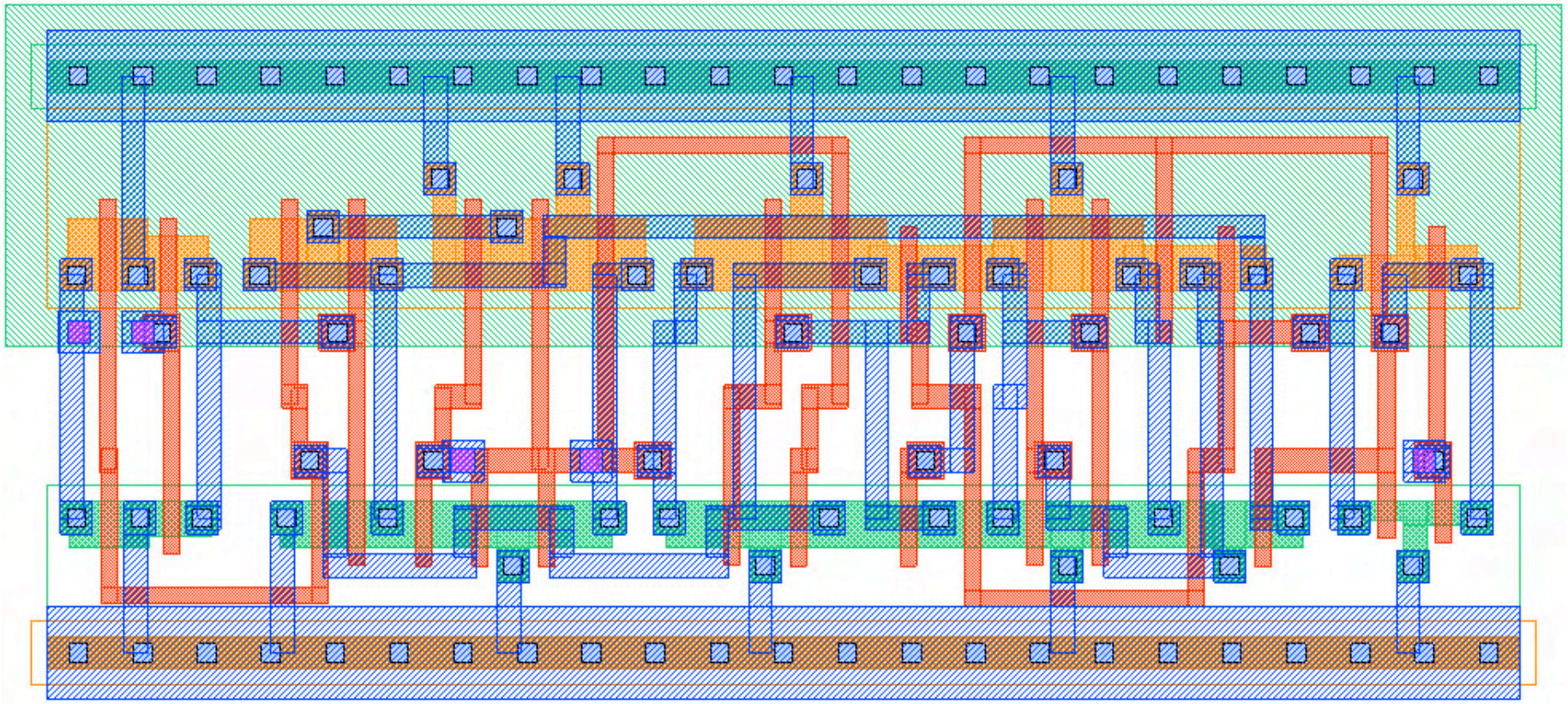


DF1



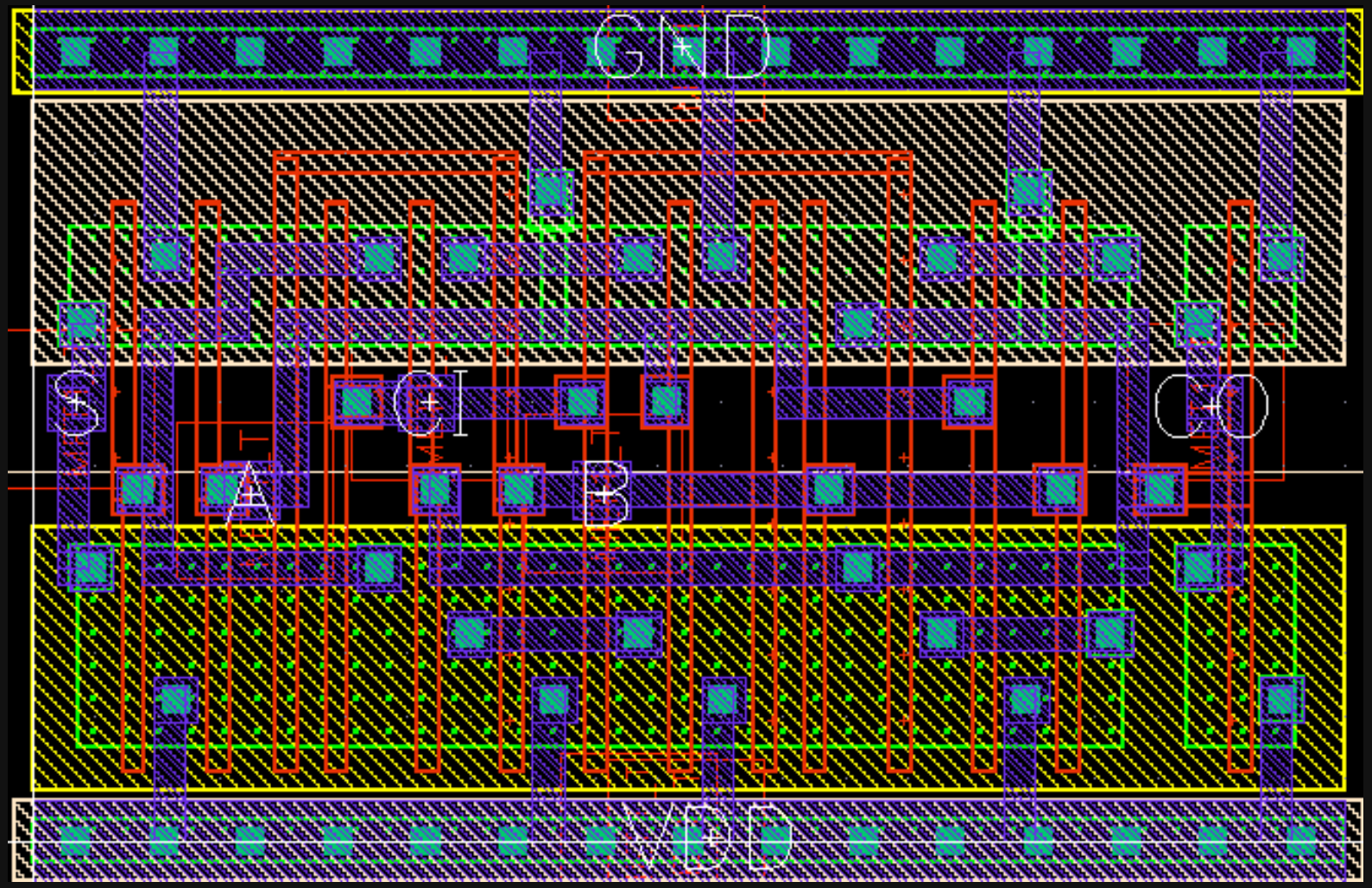
MUX21

Results: Layouts

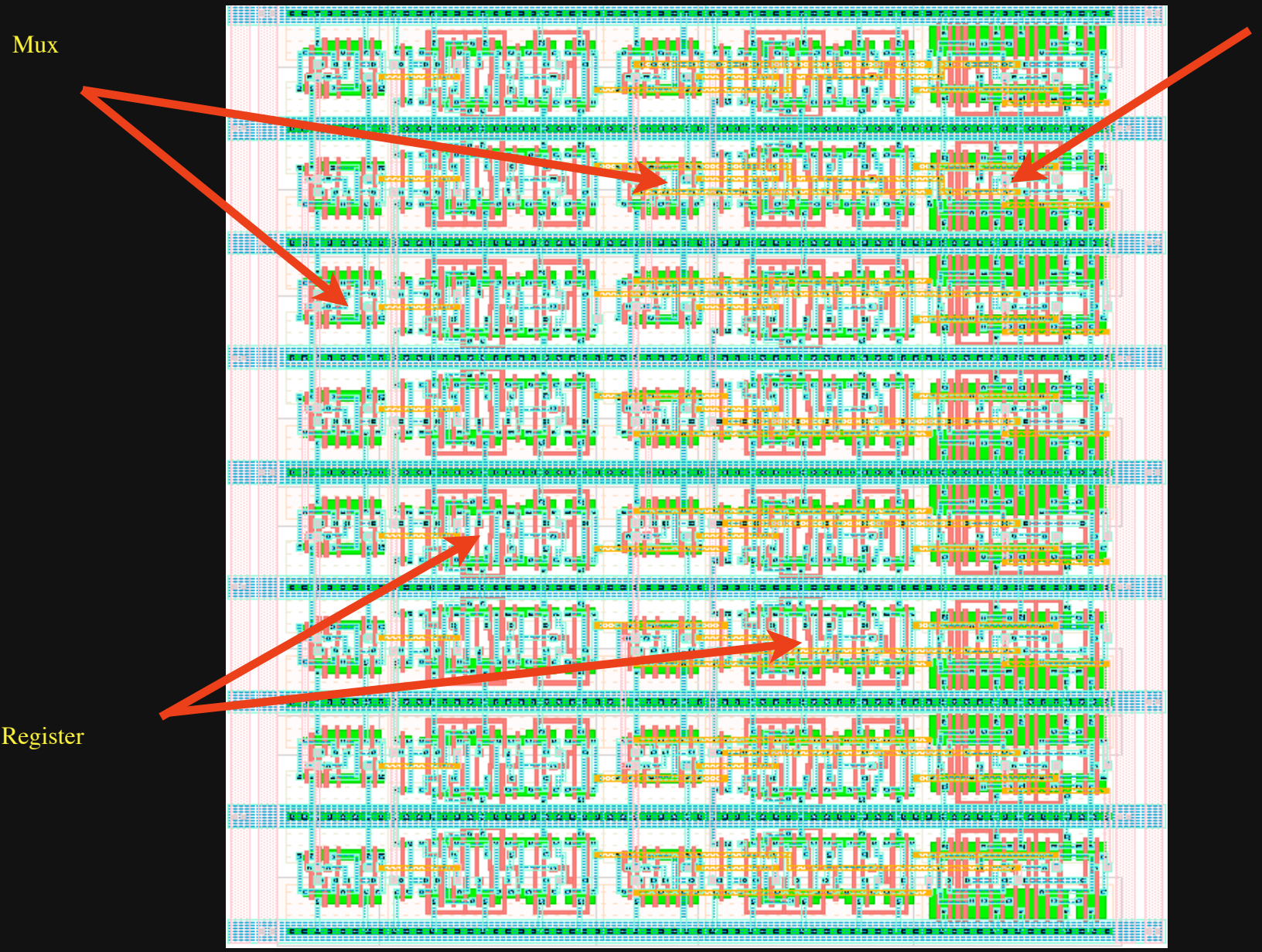


JK1 (34 transistors)

ADD32

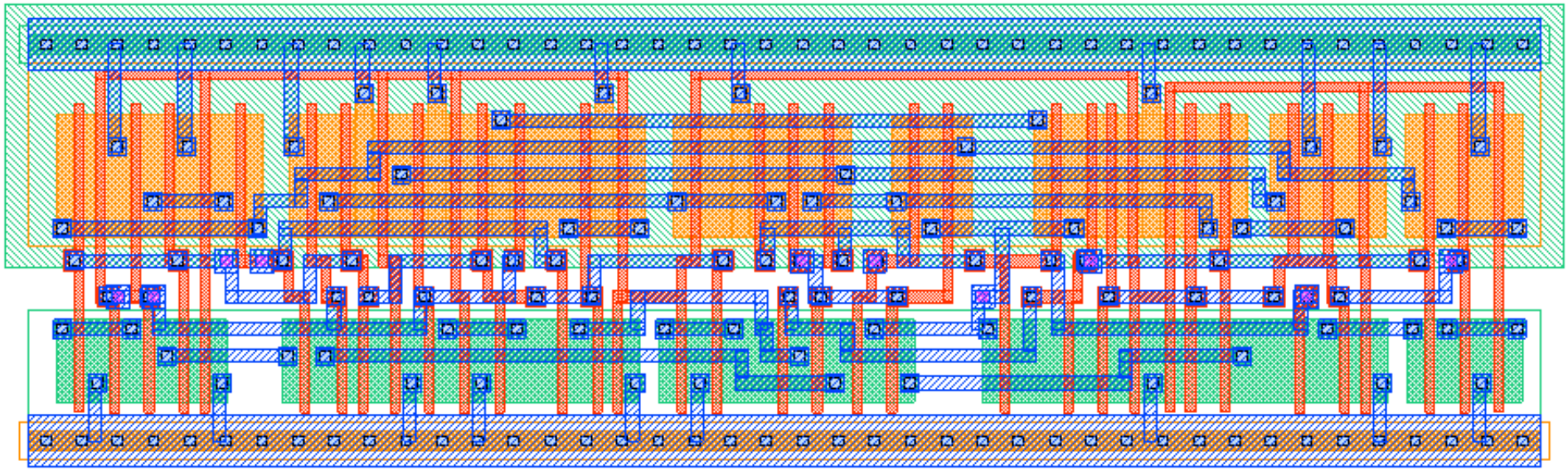


Adder



Results: Layouts

Non-Complementary Logic

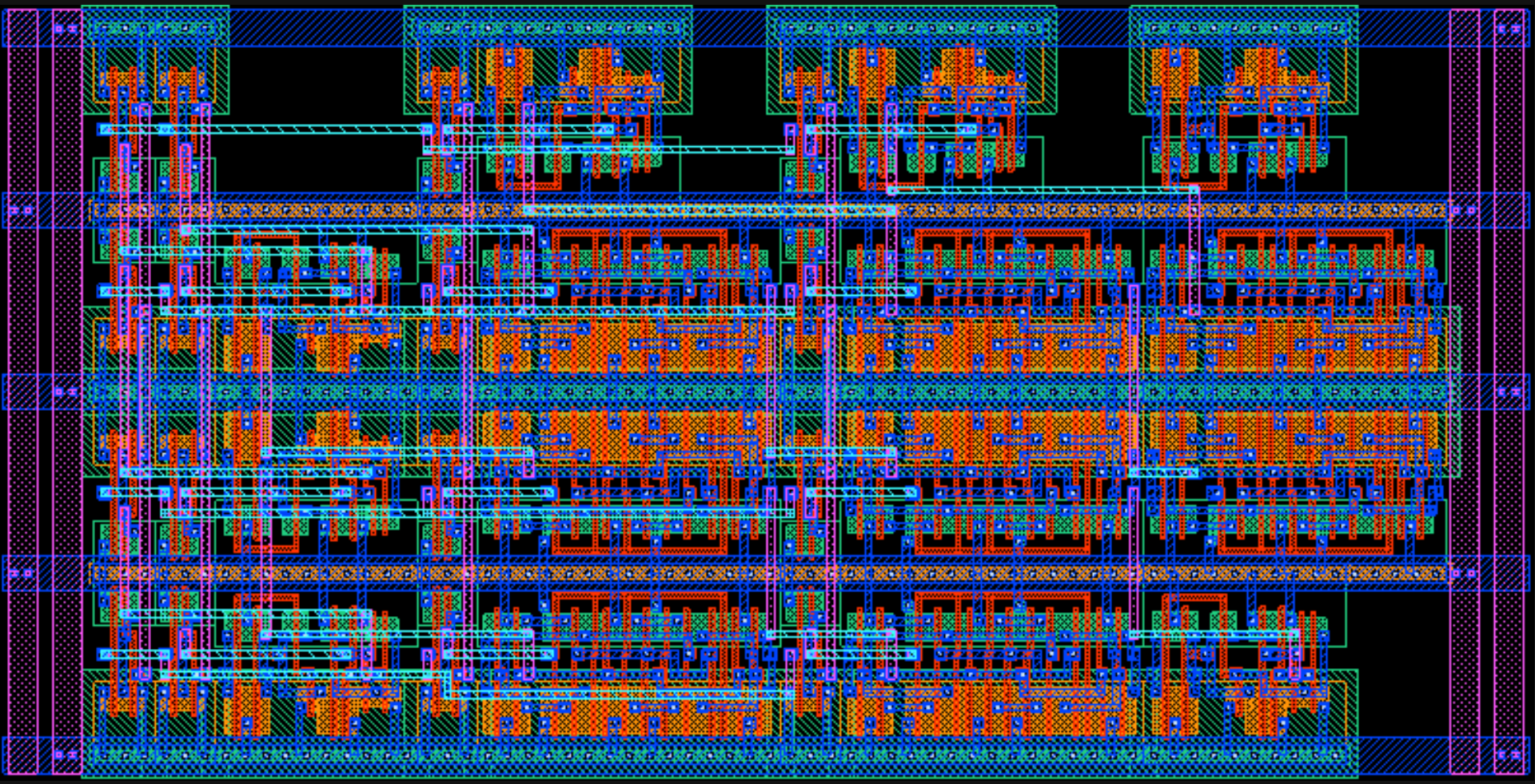


LBBDD_0117177F177F7FFF (68 transistors)

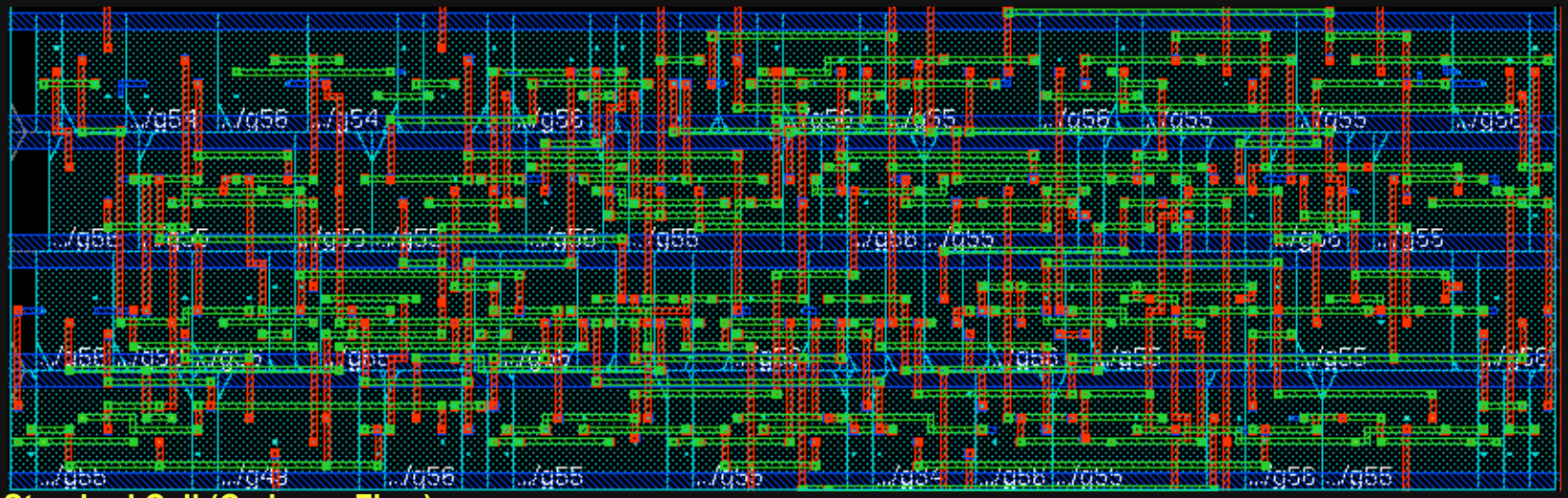
Runtime: 36 min

L.S.da Rosa Jr., F.Marques, T.M.G.Cardoso, R.P.Ribas, S.S.Sapatnekar, A.I.Reis, Fast Transistor Networks from BDDs. SBCCI 2006, pp. 137-142.

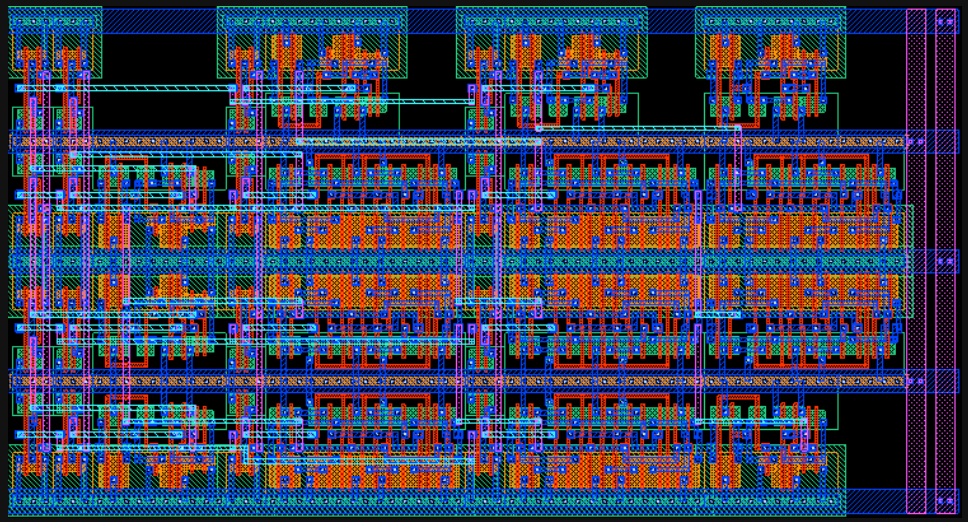
Data Path Design Automation



Multiplier Carry-Save 4x4



Standard Cell (Cadence Flow)



Generated with our Data Path Compiler

Multiplier Carry-Save 4x4

	Standard Cell	Cell Compiler	Gain (%)
Number of Cells	52	28	46
Number of Transistors	634	376	59.3
Area (μm^2)	6716	5070	24.50
Delay (ps)	2174	1896	12.8
Power (mW)	6.45	3.97	61.55

Conclusions

Conclusions

Cell generated on-the-fly target to their environment

Area reduction

Reduction on the number of transistors

Cell library free

Wirelength minimization

Power Reduction

Delay Reduction

Conclusions

Let's do

Transistor Level

Design Automation

Electronic Design Automation at Transistor Level

Ricardo Reis

Universidade Federal do Rio Grande do Sul
Instituto de Informática - Porto Alegre - RS - Brasil
reis@inf.ufrgs.br

UFRGS