

# AMS Circuit Generation, It's in the



John Crossley, Alberto Puggelli, Hanh-Phuc Le,  
Lingkai Kong, Rachel Nancollas, Bonjern Yang, Elad Alon



HOME > NEWS > LOCAL

## Plastic Bag Ban in 14 Alameda County Cities

Bags will cost you a dime apiece

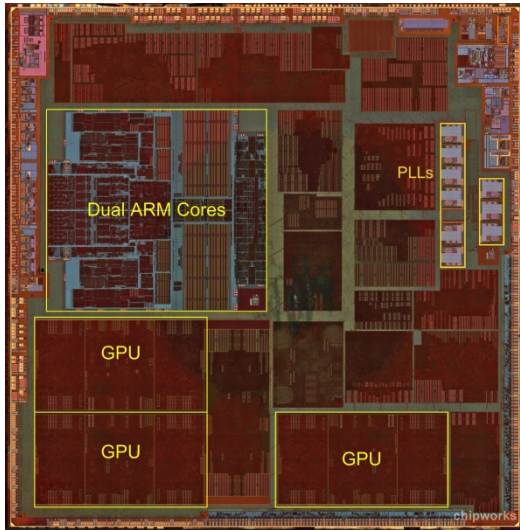
By Lisa Fernandez | Tuesday, Jan 1, 2013 | Updated 7:35 AM PST



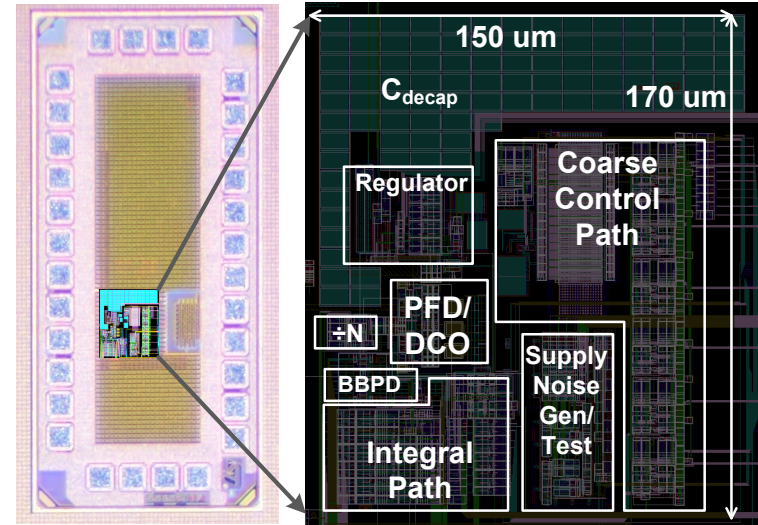
**BAG motivation similar to bag ban motivation**



# Analog Circuit Design Reuse



Apple A6



Personal Example - Digital PLL

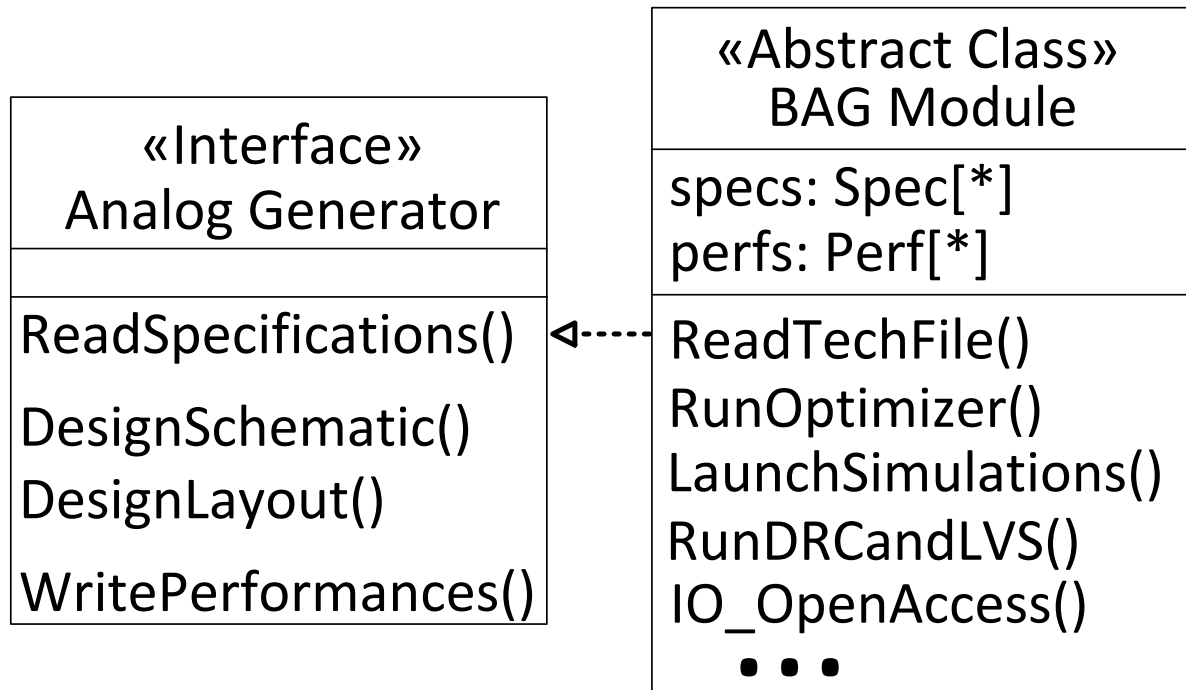
- Need for reuse in industry and academia: TTM vs. TTG
- Reuse is hard due to highly manual analog design flows
  - New process/specs → need to re-design/re-verify
- **What we need for reuse: Scriptable design flow**
  - Old idea that didn't catch on<sub>1</sub> but complexity is higher now

# Outline

- BAG basics:
  - What is BAG?
  - How is layout handled?
- BAG details:
  - “Standard” layout styles
  - Case Study: DCDC Regulator
- Conclusions

# Berkeley Analog Generator (BAG)

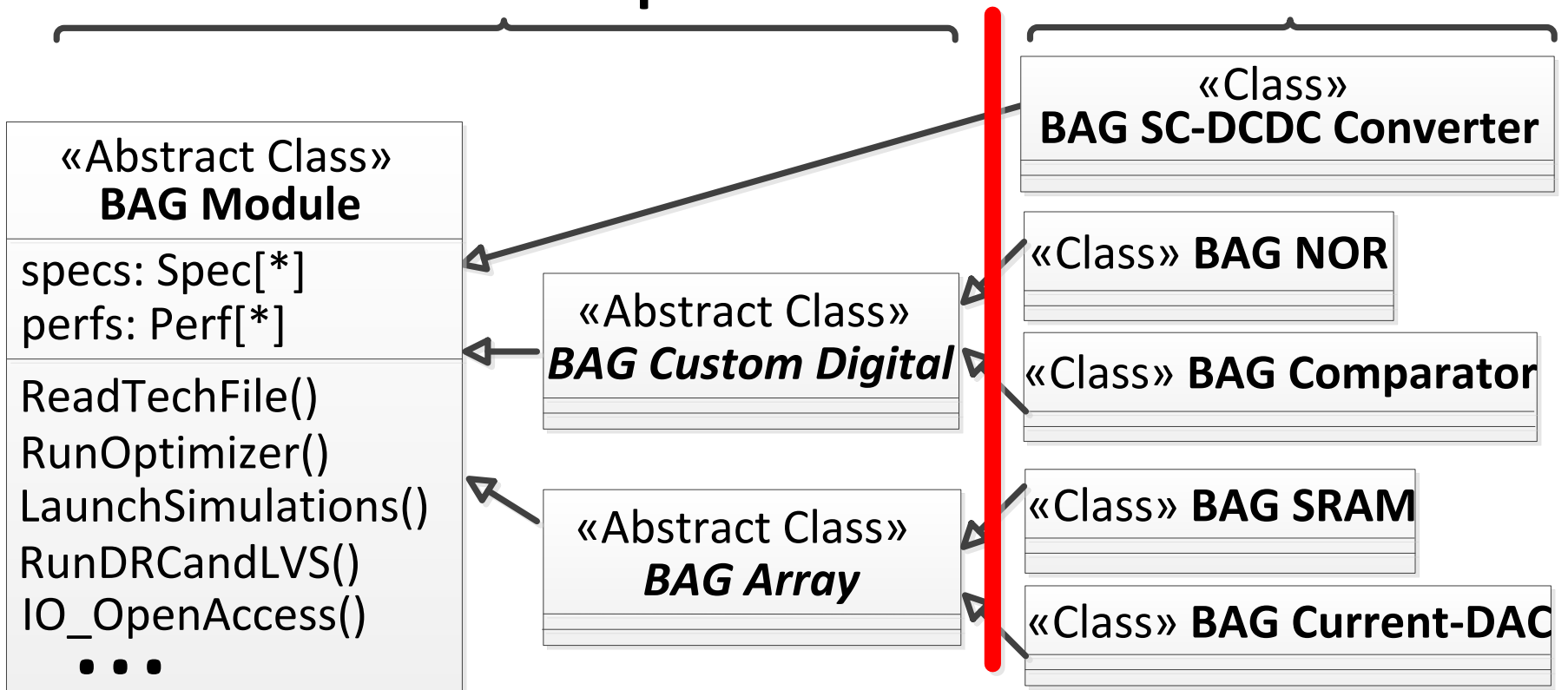
- **Goal: codify analog designer's methodology**
  - Designer's output is a generator rather than an instance.
- **Python-based, hierarchical framework**



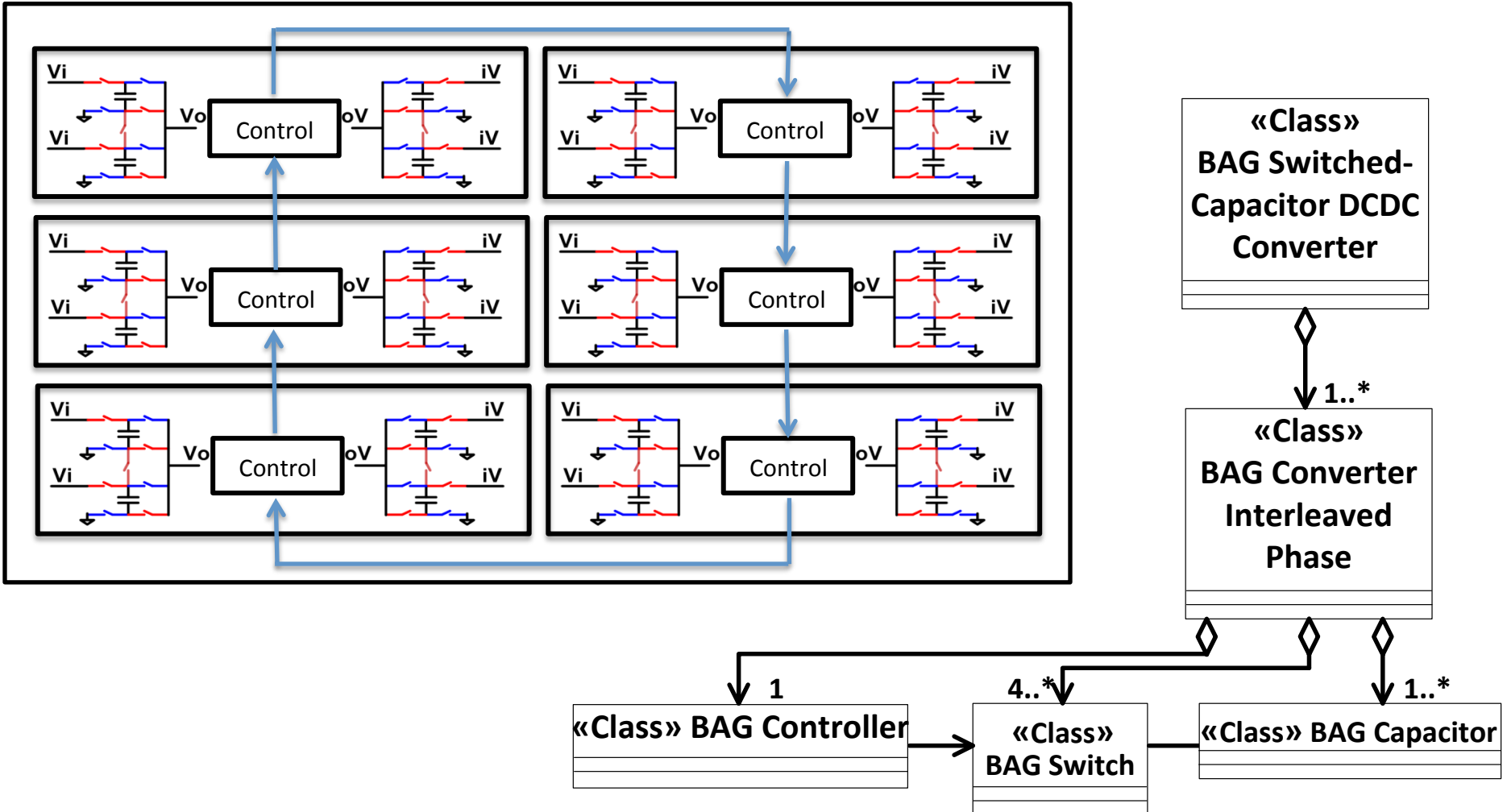
# BAG Structure Diagram

**BAG Developer**

**Generator Writer**



# Hierarchical Example: DCDC Regulator



# BAG: Typical Generator Writer's Flow

## Schematic Entry

Creation of Parametric Schematic



Creation of Parametric Testbench

## BAG framework

Template Generation

Interactive Schematic Exploration



<class>.DesignSchematic() Codification



Layout Exploration



<class>.DesignLayout() Codification

## Python Code

Welcome to pylab, a matplotlib-based Python environment.  
For more information, type 'help(pylab)'.

```
In [1]: execfile('setup.py')
```

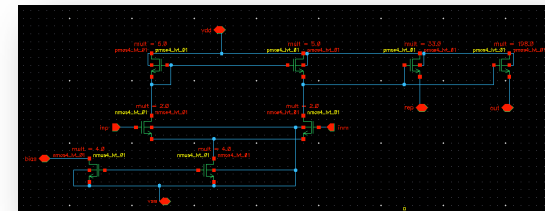
```
In [2]: cfg = BAG.BagCfg('BAG.cfg')  
log file for virtuoso run is located here: /tmp/fileks8IeZ
```

```
In [3]: from reg_vco import reg_vco
```

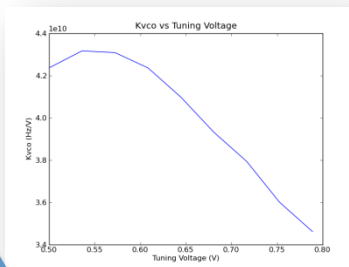
```
In [4]: rvco = reg_vco(cfg)
```



## Sized Schematic



## Sim. Results

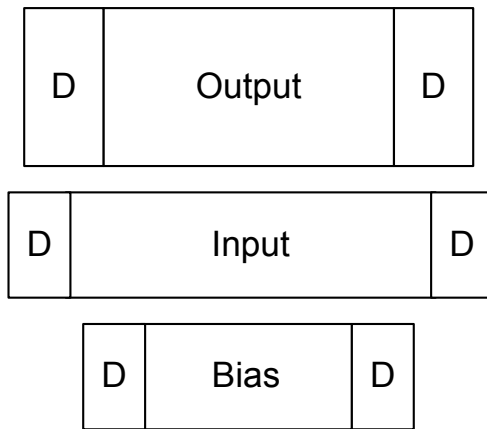


# Outline

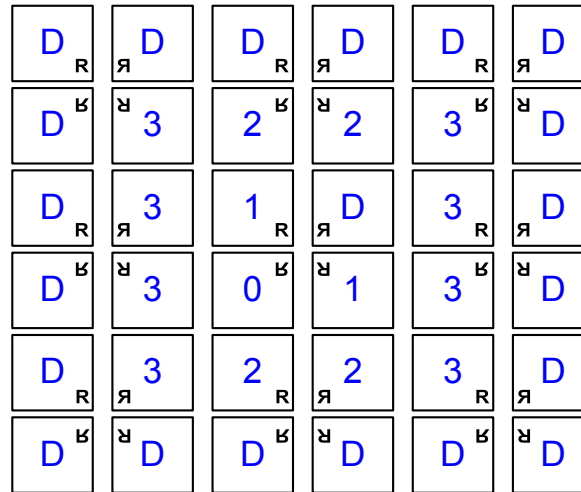
- Review:
  - What is BAG?
  - **How is layout handled?**
- Update:
  - “Standard” layout styles
  - Case Study: DCDC Regulator
- Conclusions

# Layout Styles

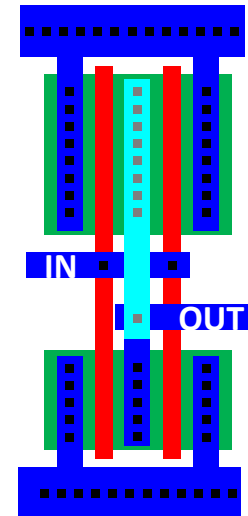
- Many cells can be mapped to a handful of “styles”
- Complete (or nearly complete) parameterized layouts generated automatically



**Amplifier Style**

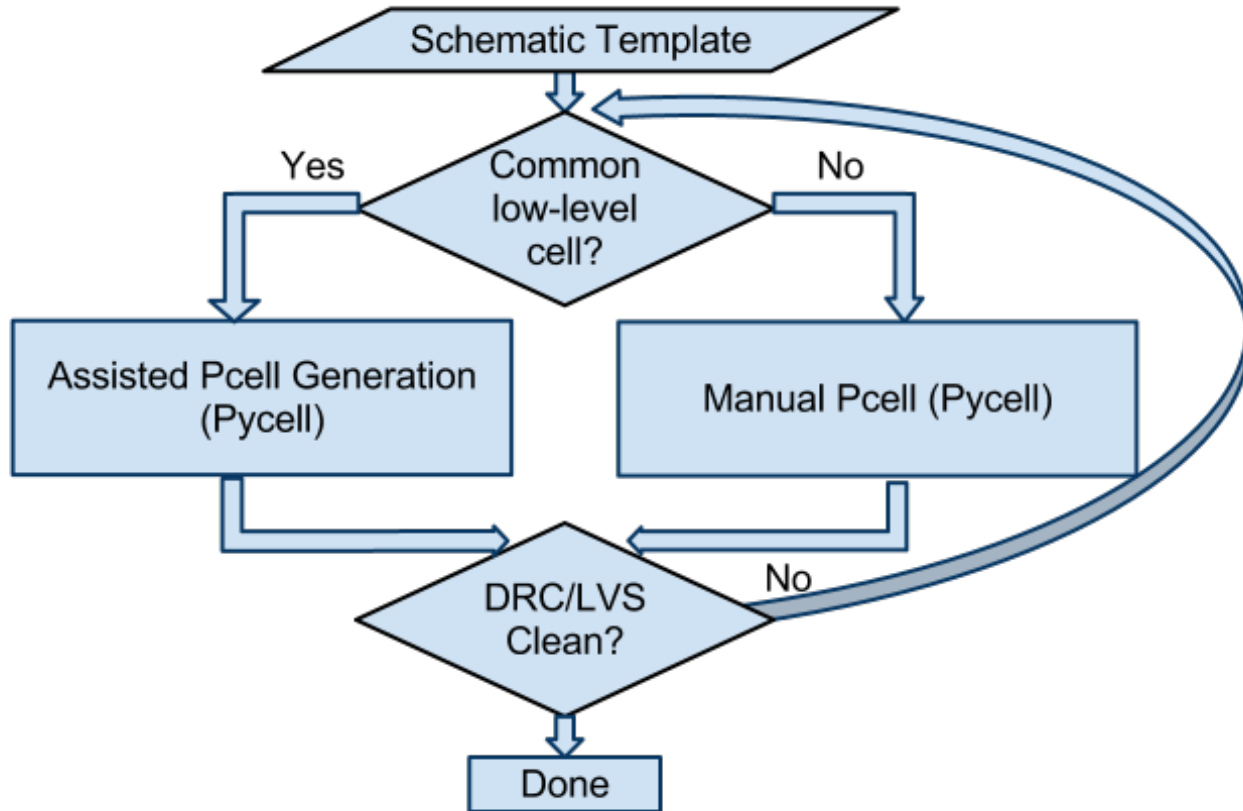


**Array Style**



**“Standard” Style**

# Pycell Based Layout



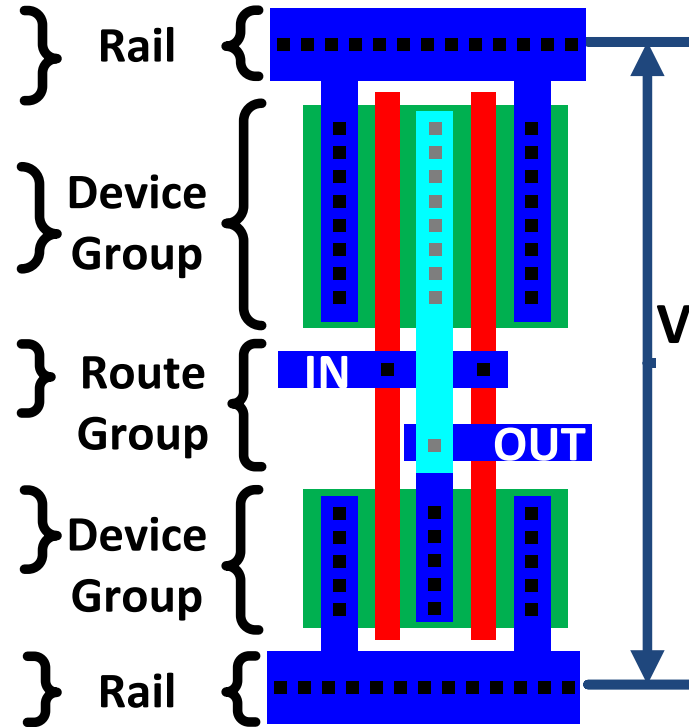
- Pycells provide more control over implementation
- Assisted pycell generation using layout *styles*

# Outline

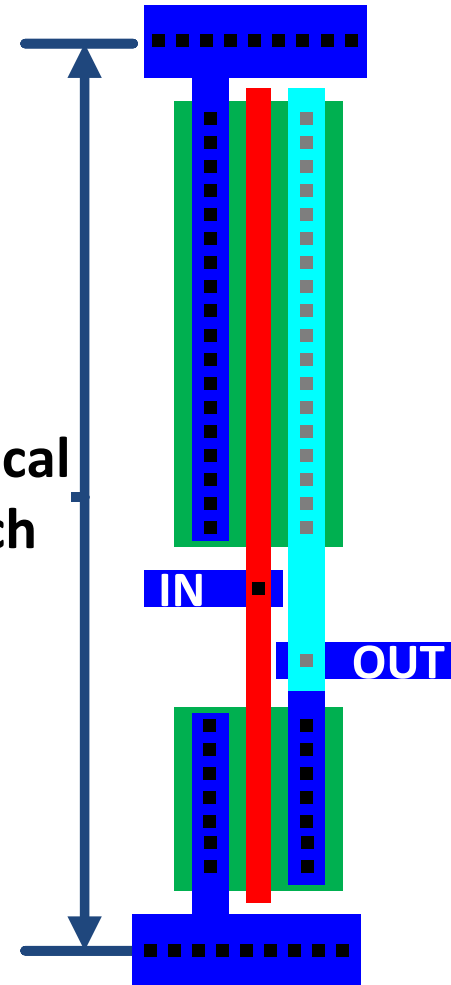
- Review:
  - What is BAG?
  - How is layout handled?
- Update:
  - **“Standard” family of layout styles**
  - Case Study: DCDC Regulator
- Conclusions

# Standard Cell Style

```
layout_info = [ dict(type='rail', name='VDD', layer='metal1', -), dict(type='device_group', name='p_row1', tran_type='pmos', -), dict(type='route_group', name='route_group1', -), dict(type='device_group', name='n_row1', tran_type='nmos', -), dict(type='rail', name='GND', layer='metal1', -), ]
```

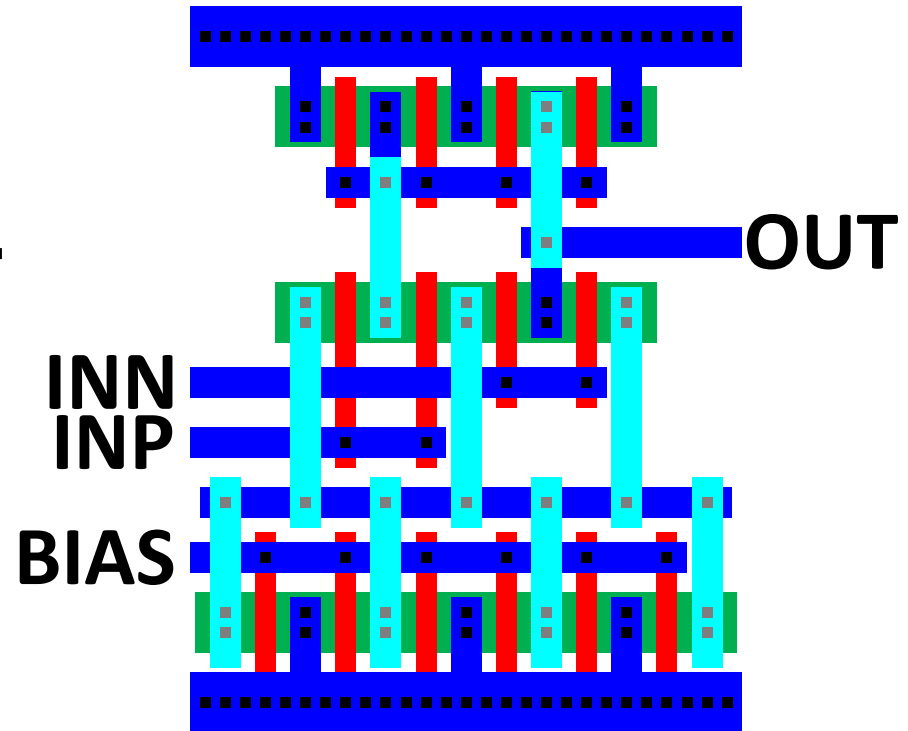
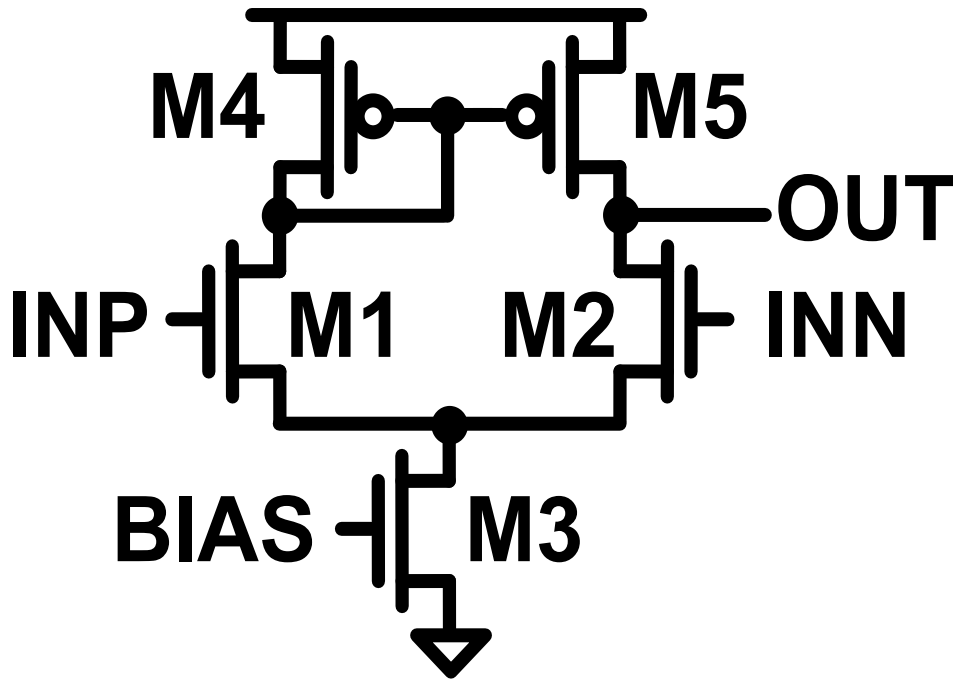


Vertical Pitch



- *Nearly* automated
- Number of fingers and finger width adjusted automatically based on vertical pitch

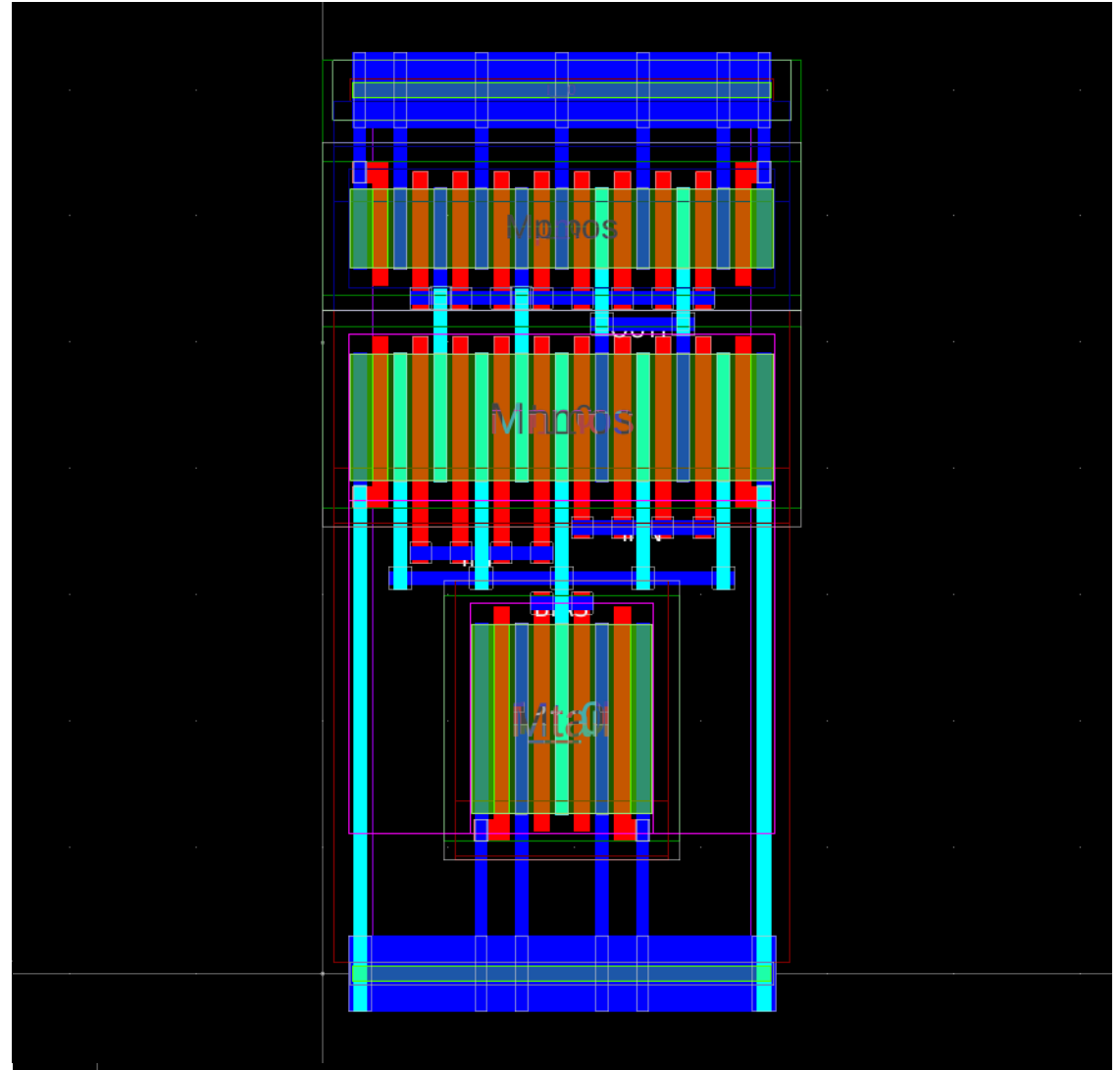
# Standard Cell Style – Analog Cells



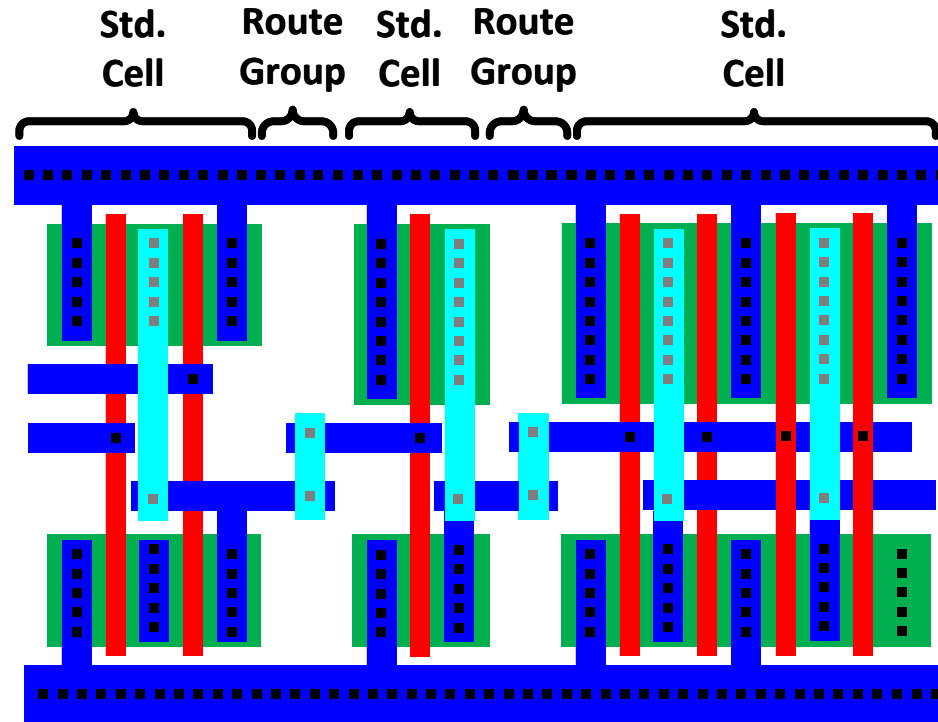
- Standard cell style based on digital BUT
- Also useful for analog/mixed circuits

# Standard Cell - Amplifier

1. Rail width
2. Dummies
3. Channel length
4. Balance PMOS/  
NMOS
5. Vertical pitch

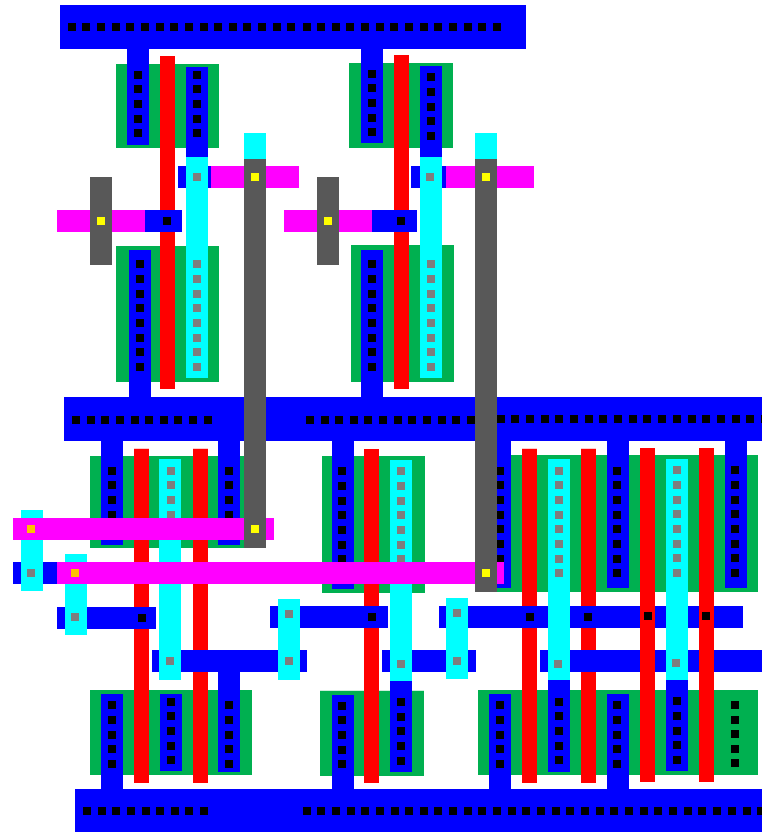


# Standard Row Style



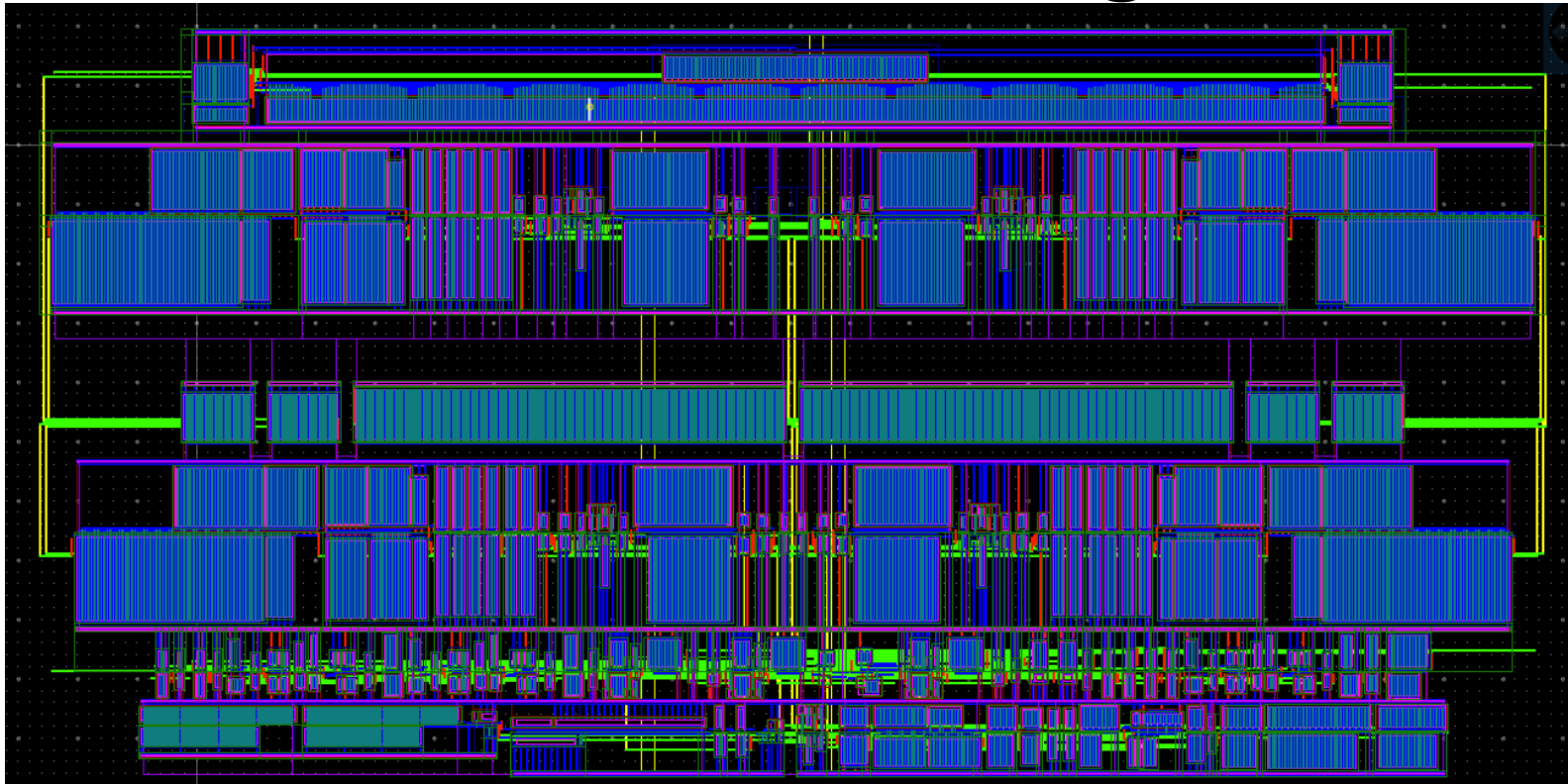
- Combine std. cells using vertical routing channels
- Vertical pitch and pmos/nmos ratio passed to lower level standard cells

# Standard Block Style



- **Combine groups of standard cells and rows**
- **Routing channels shared within each row**

# Standard Block Example: Control for DCDC Regulator



## Subcircuits

Asynchronous comparator

Decoupling capacitors

Flip flops

Level shifters

Skewed logic gates

Phase splitter

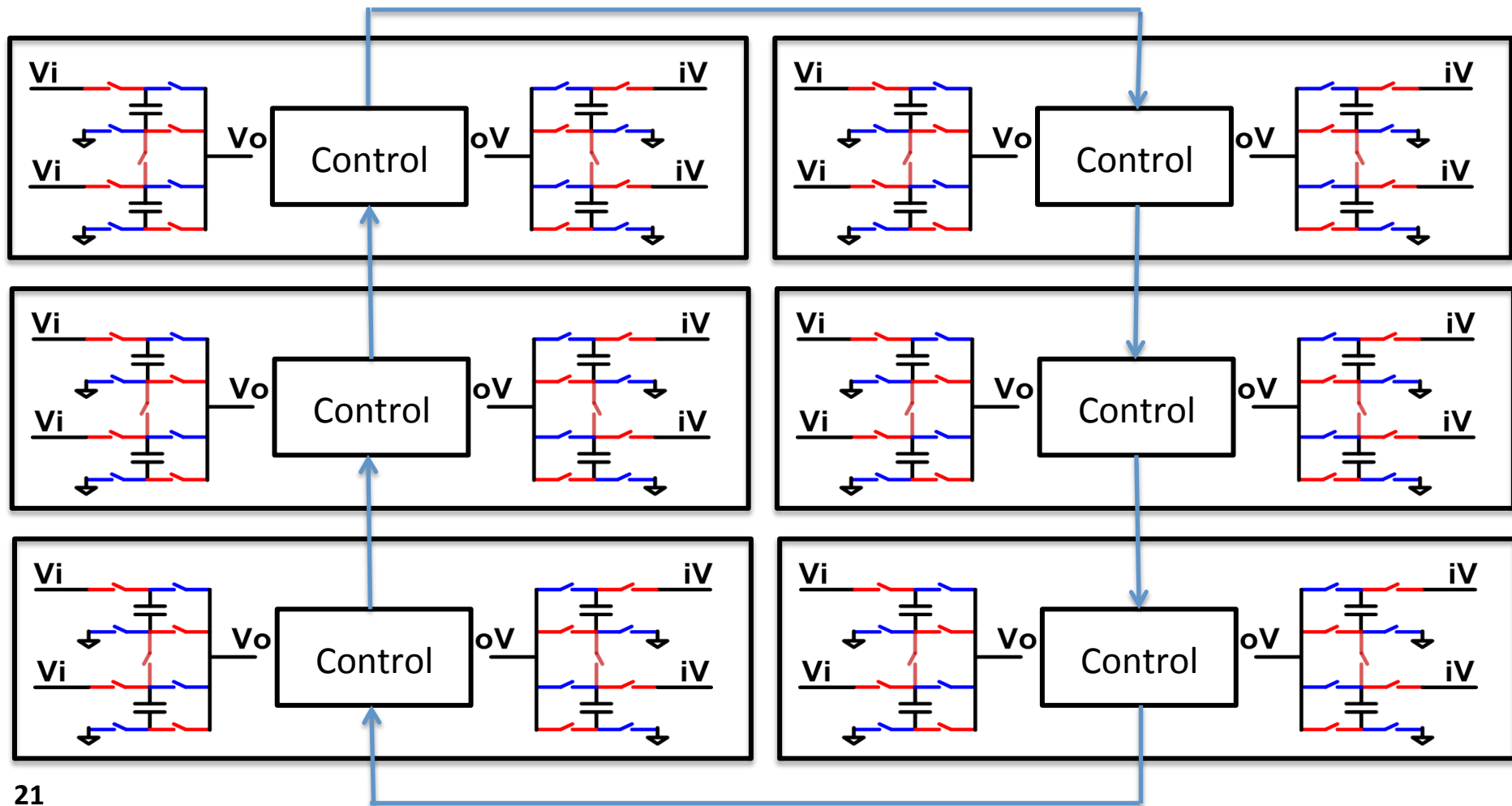
# Outline

- Review:
  - What is BAG?
  - How is layout handled?
- Update:
  - “Standard” family of layout styles
  - **Case Study: DCDC Regulator**
- Conclusions

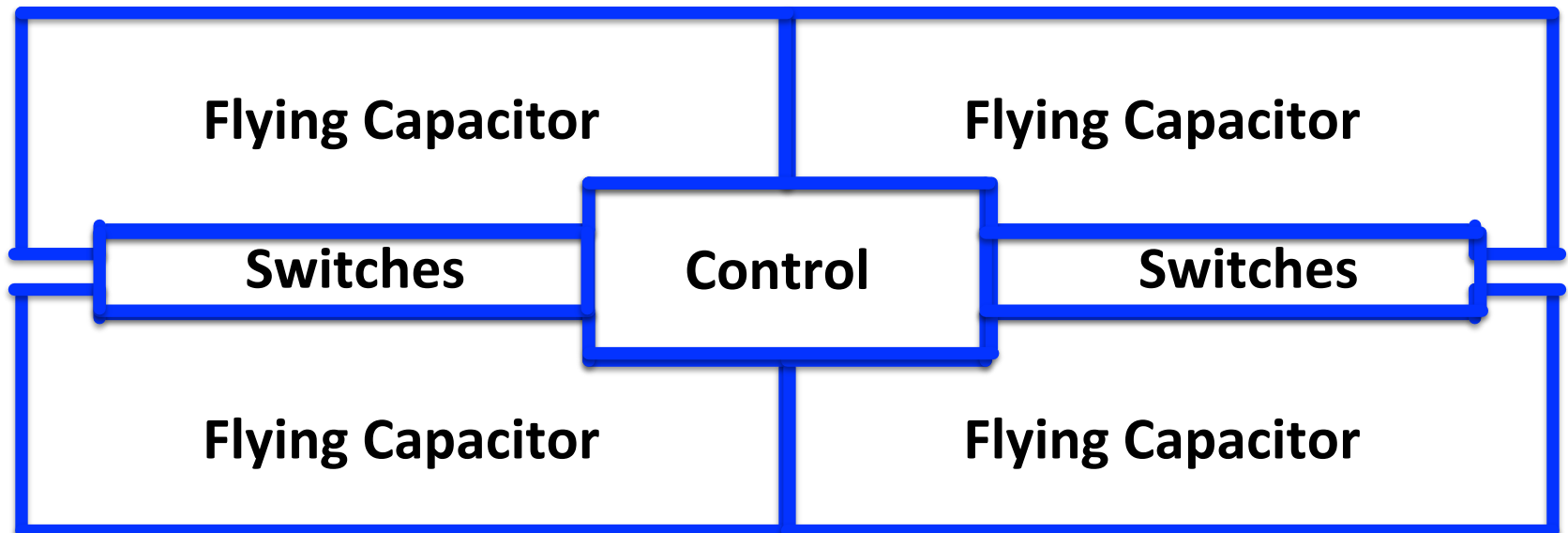
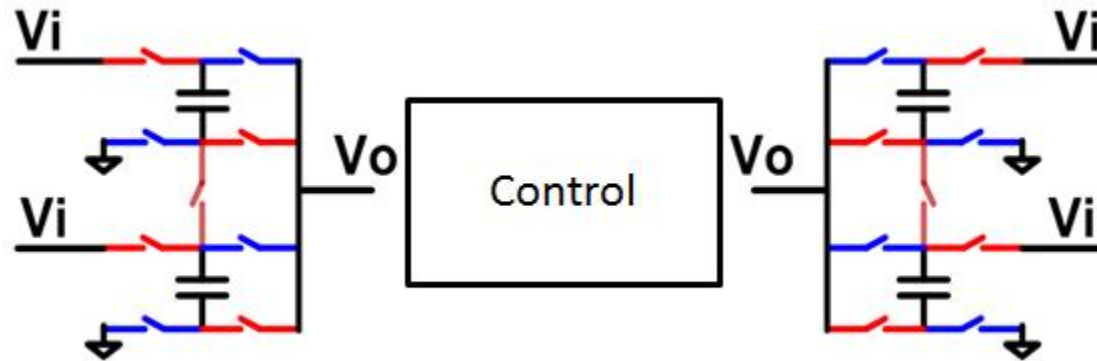
# Switched Cap DCDC Regulator

## Top Level Floorplan

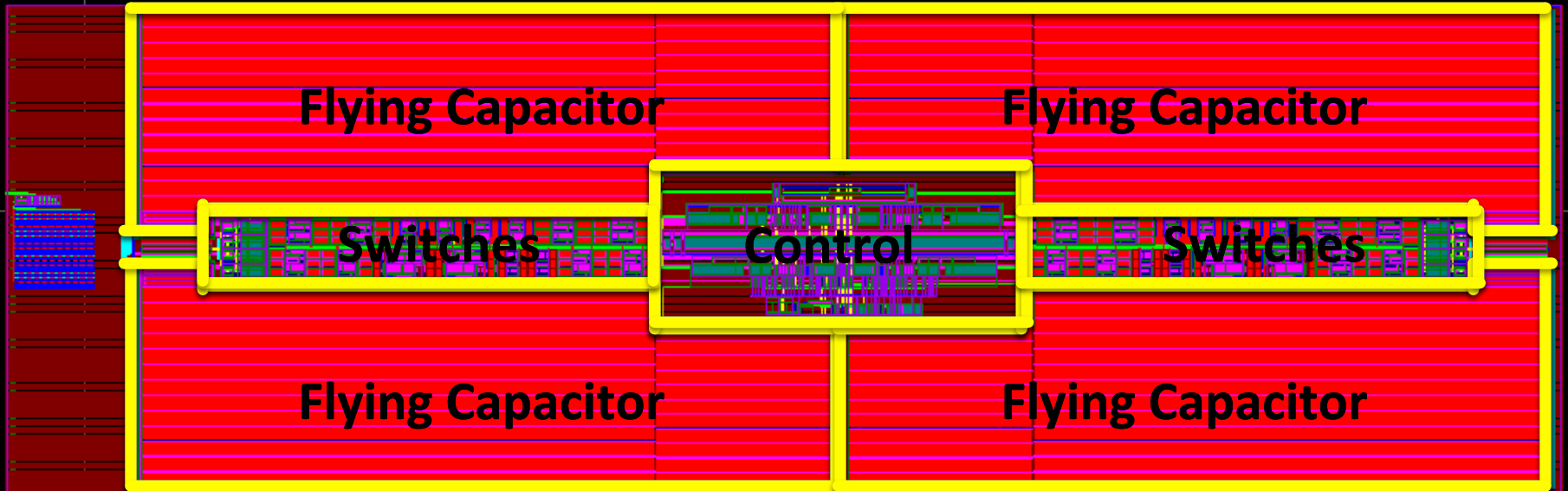
- *Manual pycell* used for upper levels of hierarchy



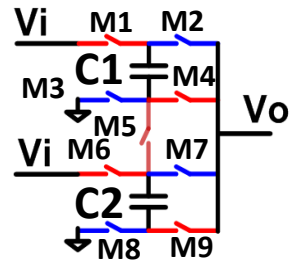
# Unit Cell Floorplan



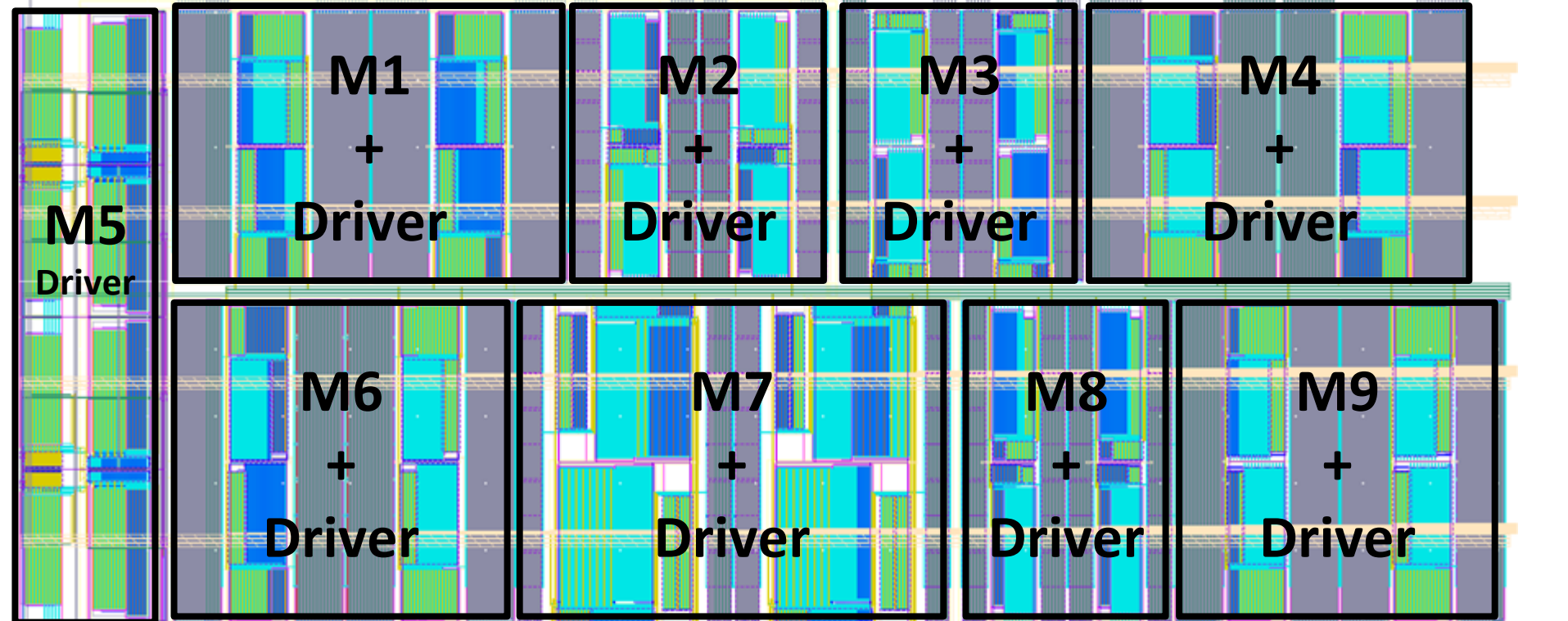
# Unit Cell Floorplan



# Power Switch Layout



C1



C2



# Conclusion

- BAG framework extended to enable automated layout
  - Pycells are powerful enough for full system layout
  - Standard layout styles ease creation of layout
- AMS circuit generation is in the bag!
  - We have real generators that handle sizing and layout
  - DCDC regulator taping out this month
  - Graduate analog circuit design class currently using BAG

# Acknowledgements

- UC Discovery Grant ele07-10283 under the IMPACT program
- NSF Infrastructure Grant No. 0403427.
- BWRC students, faculty and sponsors
- TSMC for the iPDK
- Ciranova/Synopsys for Helix License
- Matt Spencer for terrible jokes