

# Electronic Design Automation (EDA) tool development: Performance enhancements to circuit extraction

---

Rithvik Bhogavilli

January 10, 2021

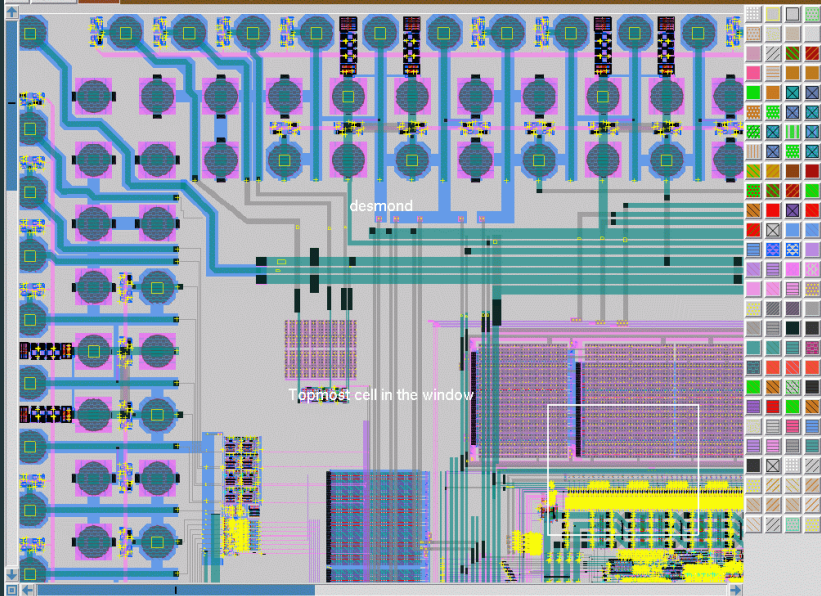
Mr. R. Tim Edwards

# Background

---

# Overview of Magic

- Very Large Scale Integration (VLSI) Layout Tool
- Open source
  - Used by universities and open source developers
- Composed of design viewer and console



```
tkcon 2.3 Main
File Console Edit Interp Prefs History Help
microns: 0.09 x 0.09 ( 0.00, 0.00 ), ( 0.09, 0.09 ) 0.01
lambda: 1 x 1 ( 0, 0 ), ( 1, 1 ) 1
Main console display active (Tc:18.4.8 / Tk:8.4.8)
adc_allrings: 10000 rects
adc_allrings: 20000 rects
adc_allrings: 30000 rects
adc_allrings: 40000 rects
adc_allrings: 50000 rects
adc_allrings: 60000 rects
adc_allrings: 70000 rects
adc_allrings: 80000 rects
adc_allrings: 90000 rects
Processing timestamp mismatches: adc_allrings.
Processing timestamp mismatches: bump_bond_pwr.
```

# Comparison to Other VLSI Layout Tools

Caesar or KIC2:

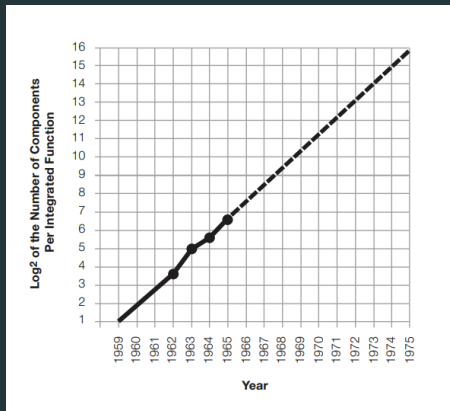
- Minimal support in routing features
- Hard to change a design once loaded

Magic:

- Increased design knowledge
- Easier to modify designs

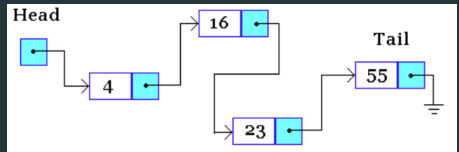
# Issues

- Magic is built for smaller designs
  - Moore's Law
- Labels
  - Store information on their associated cell
  - Must be referenced by name or location
- Labels are stored in a linked list



# Linked Lists

- Each node keeps reference to next
- Time complexity:
  - **Search:**  $\mathcal{O}(n)$
  - **Insertion:**  $\mathcal{O}(1)$
  - **Deletion:**  $\mathcal{O}(n)$



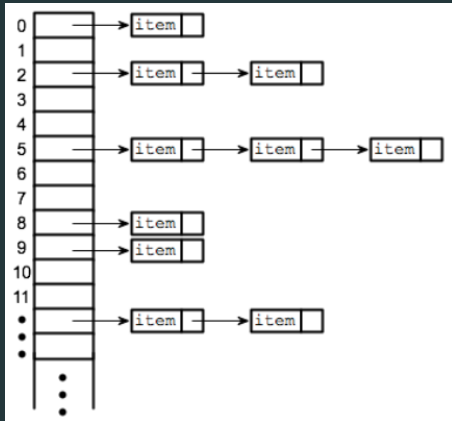


# Solution

---

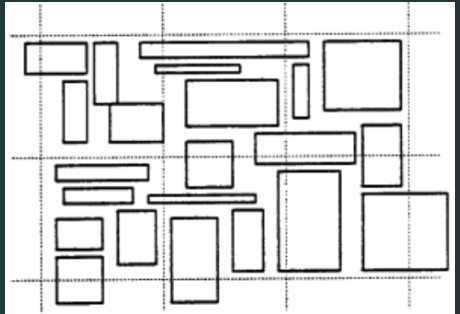
# Hash Table

- Hash algorithm sorts items by index
  - Collisions resolved through chaining
- Time complexity:
  - Search:  $\mathcal{O}(1)$
  - Insertion:  $\mathcal{O}(1)$
  - Deletion:  $\mathcal{O}(1)$



## Binned Plane (bplane)

- 2D hash table
- Cells track information about labels in the region
- Labels have near equal size and spatial distribution



# Methods

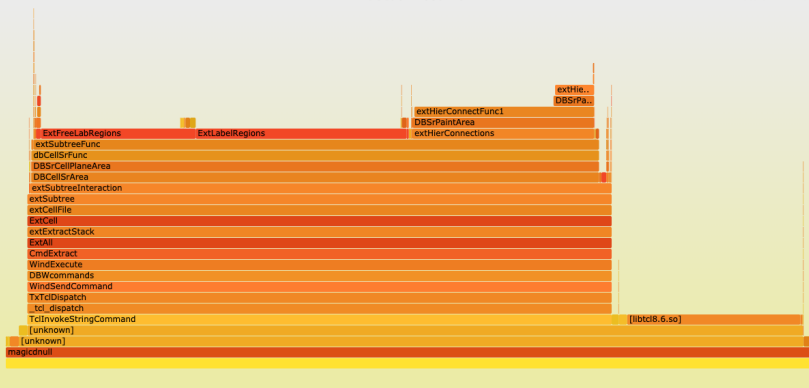
---

# Measuring Performance

- Linux Perf tool used
  - Sampling rate of 100 per second
  - Generates time spent per function
- Flame graph generated
  - Gives proportion of time spent on given function

# Extraction Baseline

Search



# Methods

1. striVe chip loaded into Magic
2. Command of interest is run
3. Performance measured at 100 samples per second
4. Magic is recompiled with optimizations
5. Performance measured at 100 samples per second
6. Flame graph generated for analysis

# Selecting Areas for Optimization

Based on previous experience in loading large designs

Chosen Areas:

- Extraction
- Net Selection
- Label Search by Content



# Areas of Optimization

---

# Extraction

- **extract all** command extracts into .ext file
- Functions were found to be searching all labels
- plane added to CellDef
  - Properties for the cell, includes label storage
- extSubtreeFunc and extHierConnections found to be of concern

# Net Selection

- **select clear** used to analyze
- DBTreeSrLabels found to take most time
- TF\_LABEL\_ATTACH and TF\_LABEL\_DISPLAY flags for labels
  - Requires use of bplane and linked list

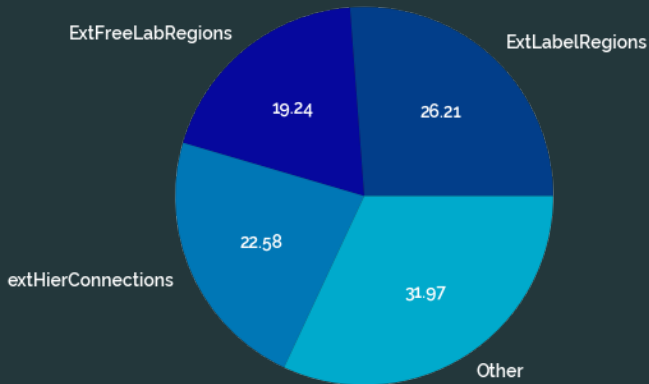
## Label Filtering by Content

- **select short** command used for profiling
  - Used because it relies on label names
- Hash table added to CellDef
- Hash table iteration implemented in DBCheckLabelsByContent

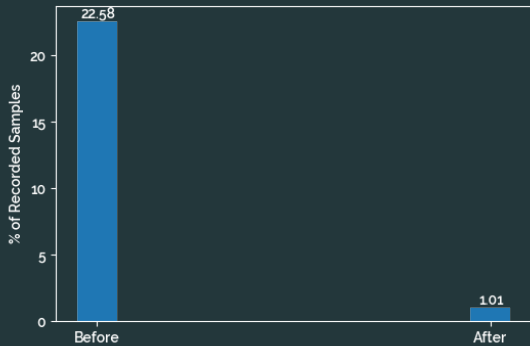
# Results

---

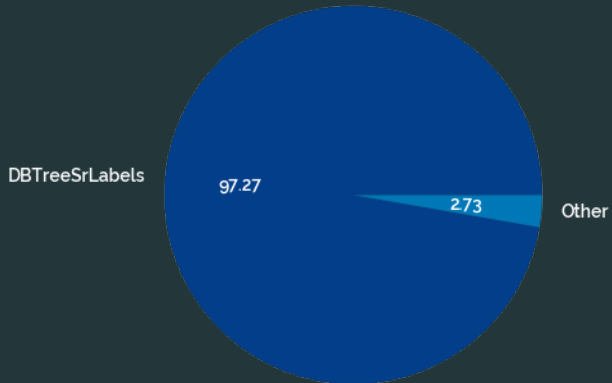
# Extraction



# Extraction

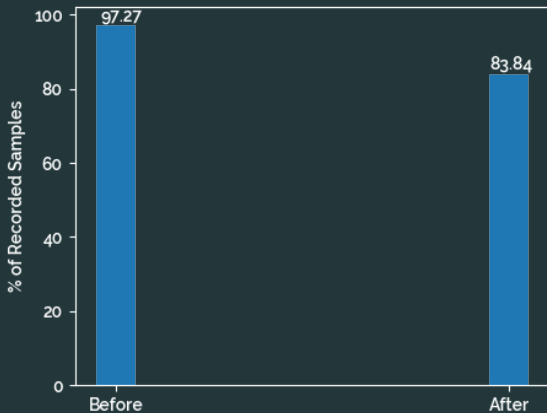


# Net Selection





# Net Selection



# Label Filtering by Content

- Perf did not detect content searching function
  - Sampling rate of 30,000 samples per second
- Confirmed to run using gdb
- Original complexity was  $\mathcal{O}(n)$ 
  - Likely to not detect  $\mathcal{O}(1)$

# Conclusions

---

# Conclusions

- General decrease in time spent on functions
- Added overhead in extraction
  - Added class initialization for bplane
- Net selection may not have been fully tested with striVe

# Future Work

- Use a wider variety of chips
  - Designs used might not test all cases
- Improve profiling
  - Profiling is manually stopped

# Acknowledgements

Thank you to Mr. Edwards for pointing me in the right direction for optimizations and providing feedback.

Thank you to my teachers and friends.

*Questions?*