DESIGN AUTOMATION CONFERENCE
FROM CHIPS TO SYSTEMS – LEARN TODAY, CREATE TOMORROW

Developing Synthesis Flows without Human Knowledge

Cunxi Yu[1], Houping Xiao[2], Giovanni De Micheli[1]

EPFL[1], SUNY Buffalo[2]

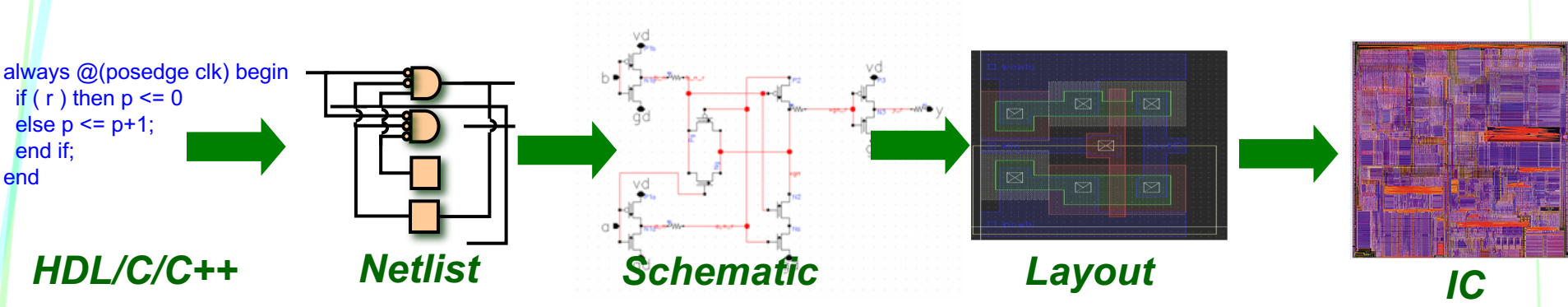
June 26, 2018

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

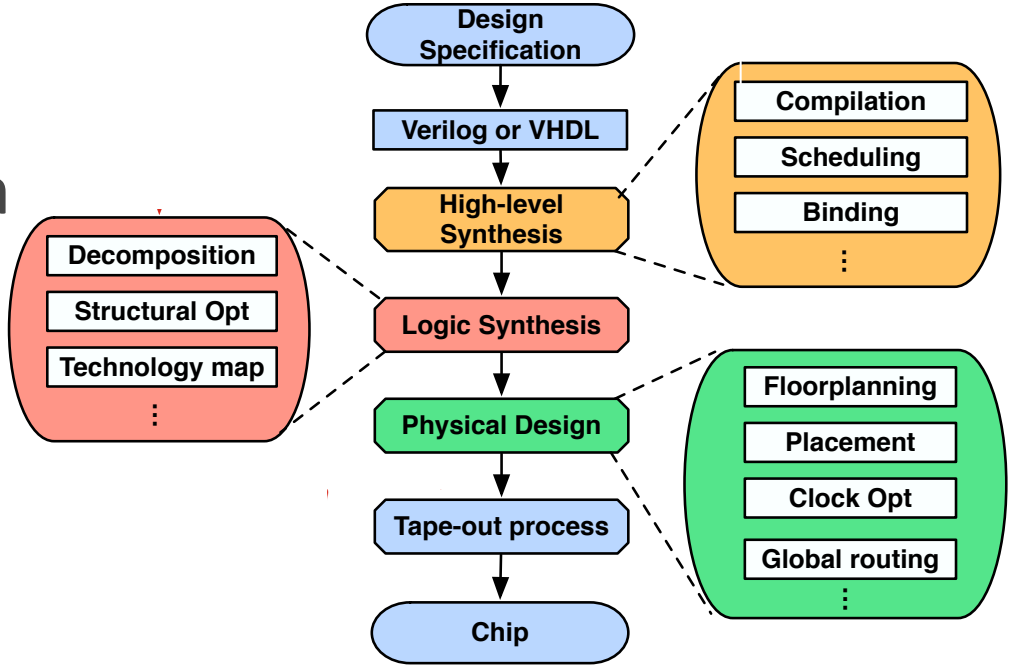
LSI Integrated Systems Laboratory

University at Buffalo
The State University of New York

Overview of Design Flows

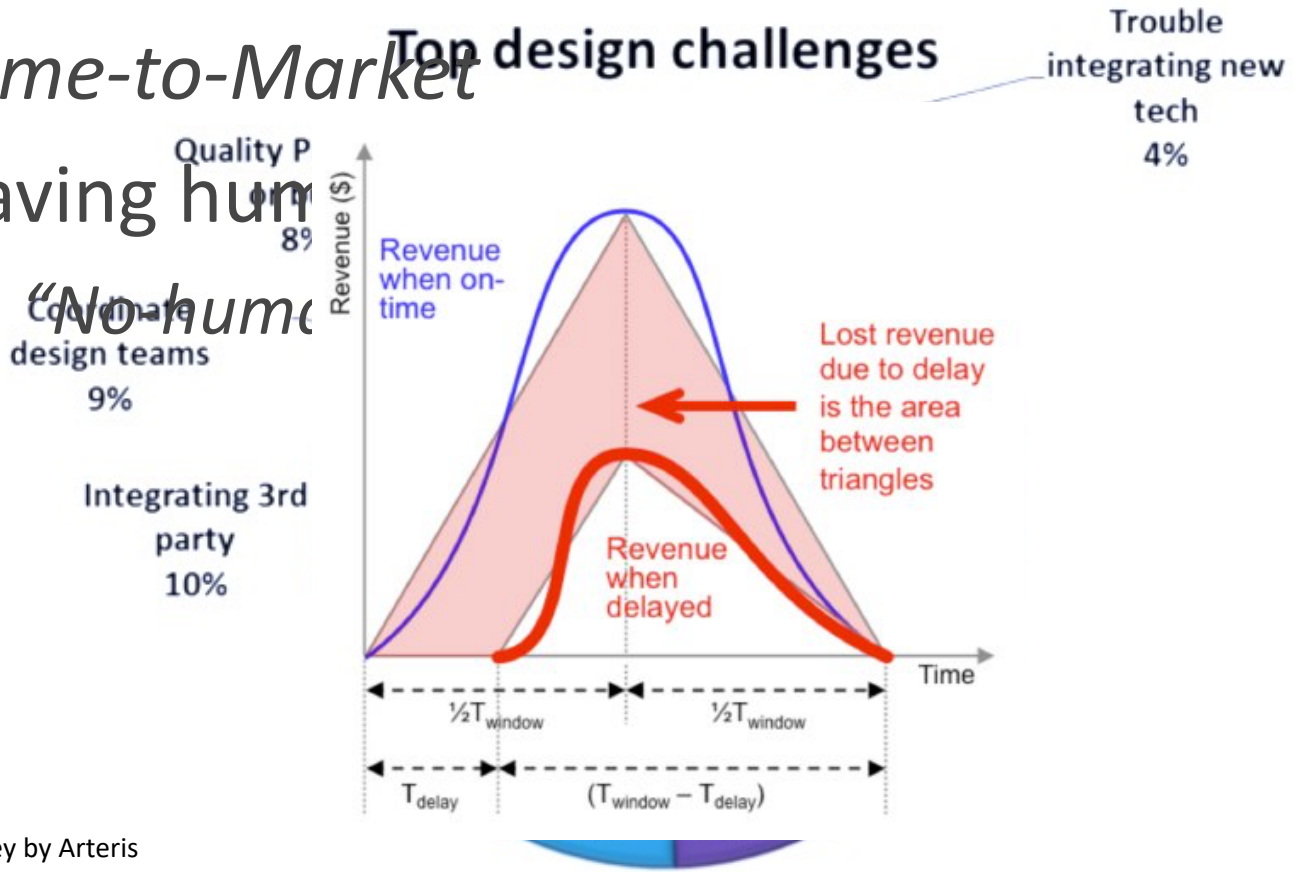


- Design flow
  - A ***sequence*** of design transformations



Introduction

- Quality of Designs
  - Design flows play the main role
- *Time-to-Market*
- Saving human
  - *"No-human*

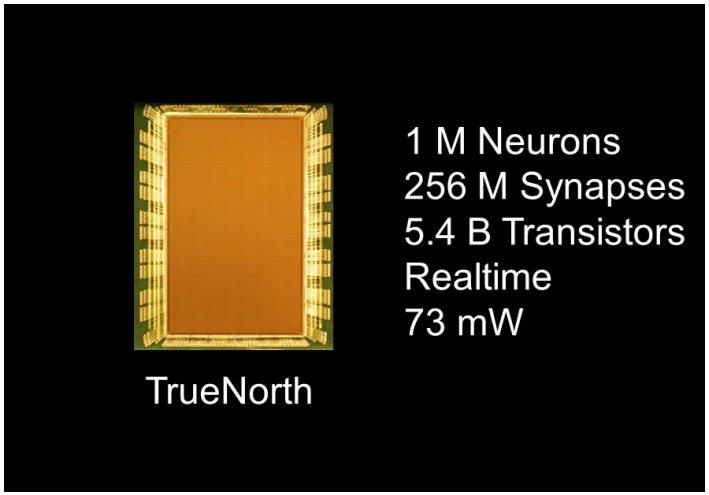


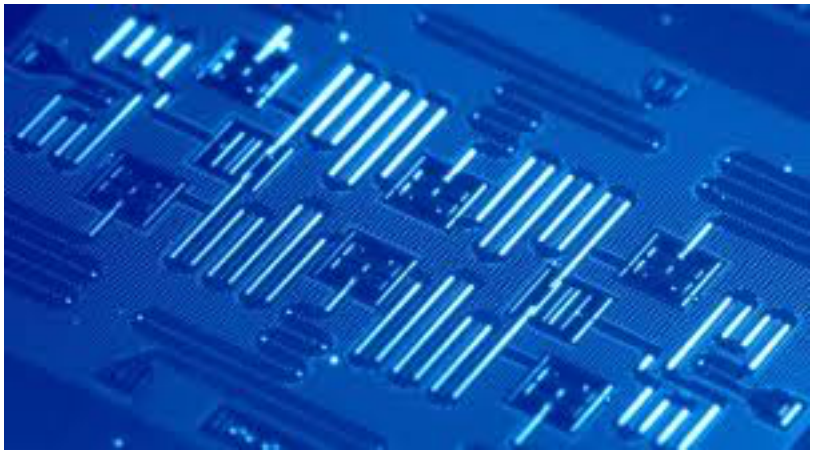
[1] Survey by Arteris

[6] Kahng, A. B. New directions for learning-based IC design tools and methodologies. ASP-DAC'18

Introduction

- Challenges in new design styles
  - e.g., Neuromorphic chip [2], Quantum chip [3]
    - Limited knowledge compared to ICs

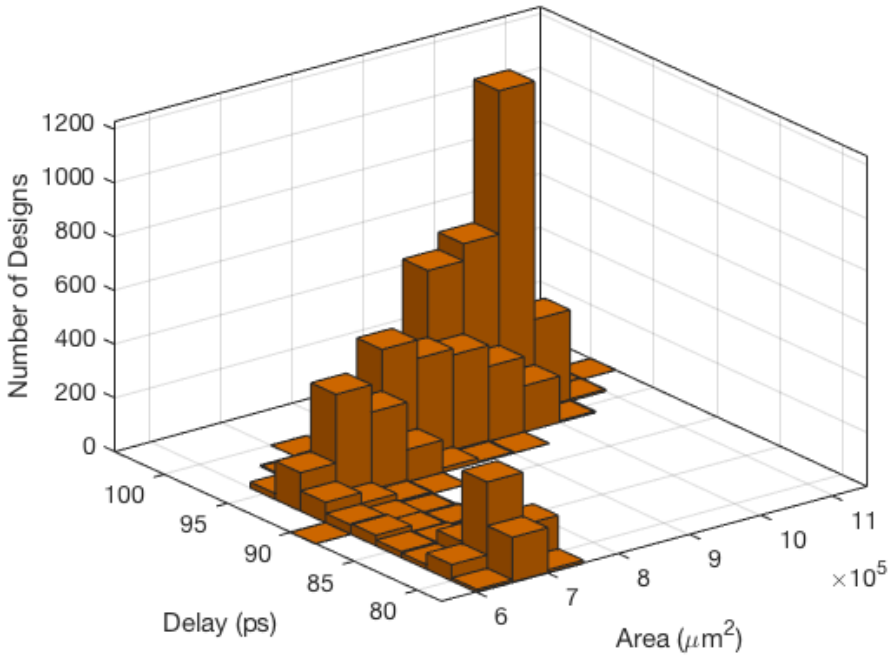


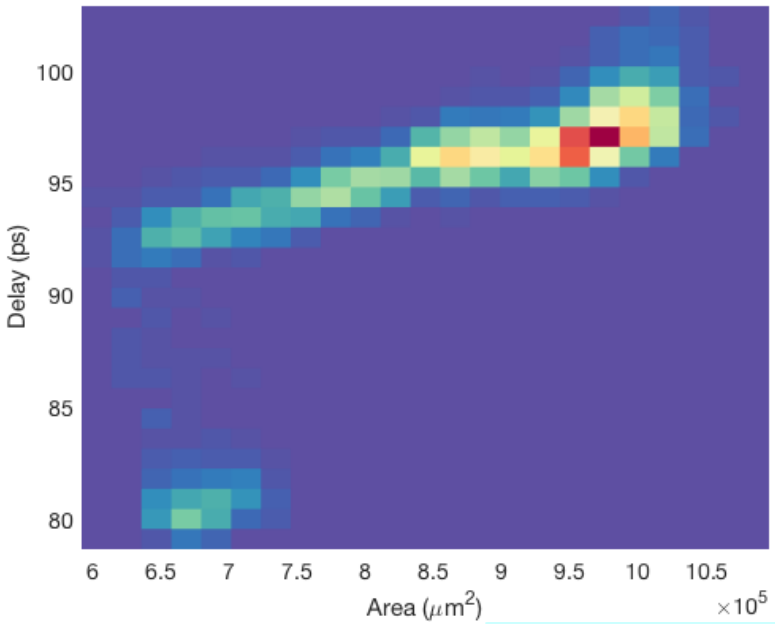


[2] IBM Research, TrueNorth: Design and Tool Flow of a 65 mW. 1 Million Neuron Programmable Neurosynaptic Chip, IEEE TCAD 2015.
[3] IBM 20-QBit

Example 1 – Permutation of Flows

- 128-bit AES
  - Same transformations
  - 50,000 random flows
    - Random permutation





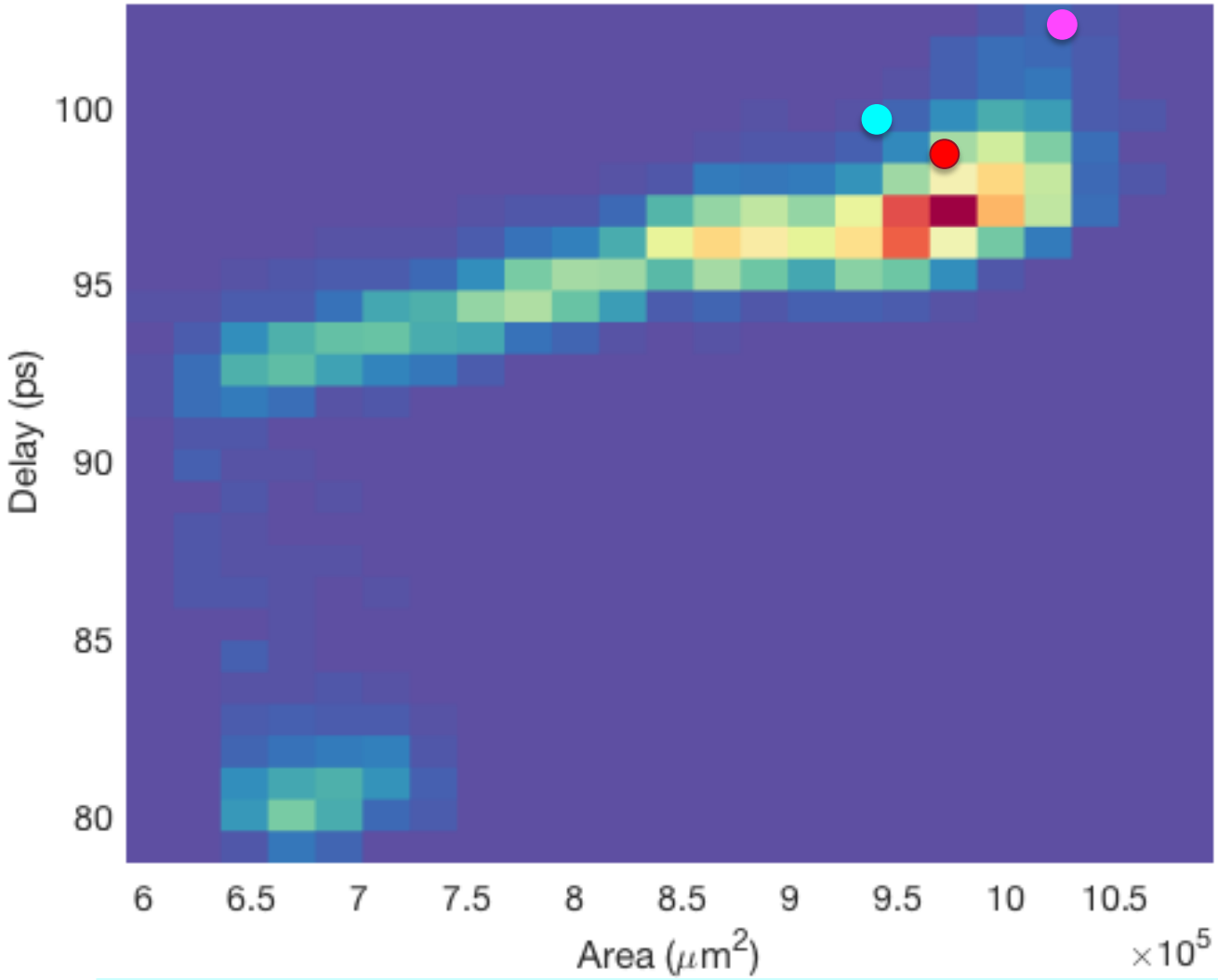
- QoR collected after tech-map
  - Delay and Area
  - 14nm

Example 1 – Permutation of Flows

resyn * 4 (24)

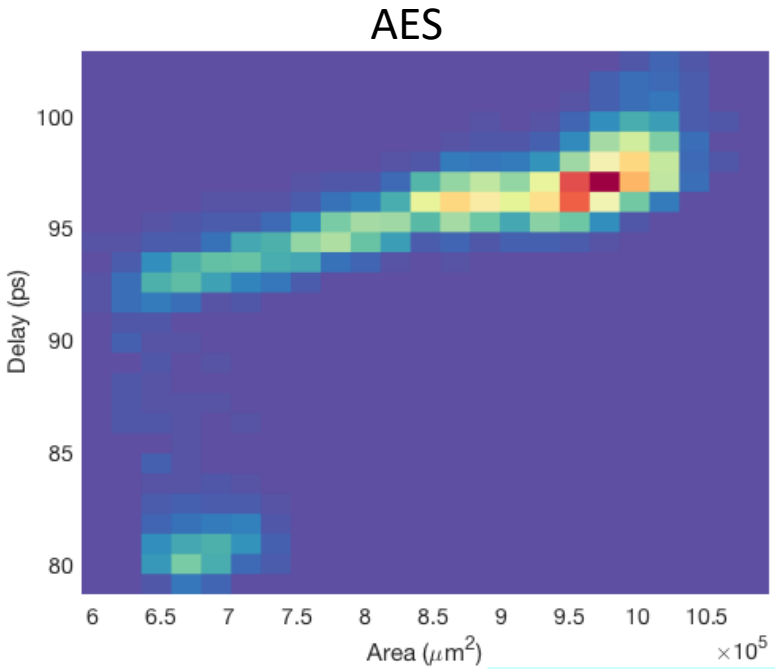
resyn2 * 4 (40)

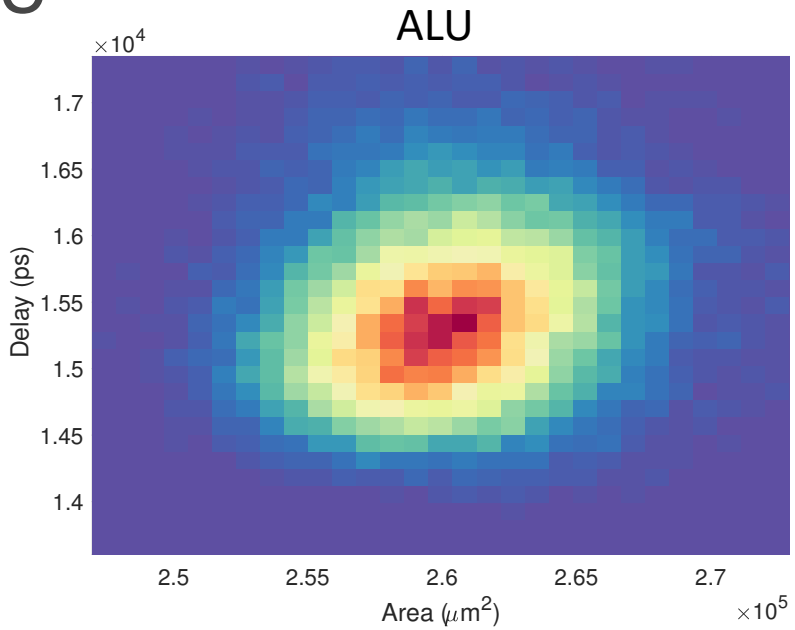
resyn3 * 4 (36)



Example 2 – Design Specific

- 128-bit AES vs. 64-bit ALU
  - Same 50,000 flows
  - *Statistically significant*

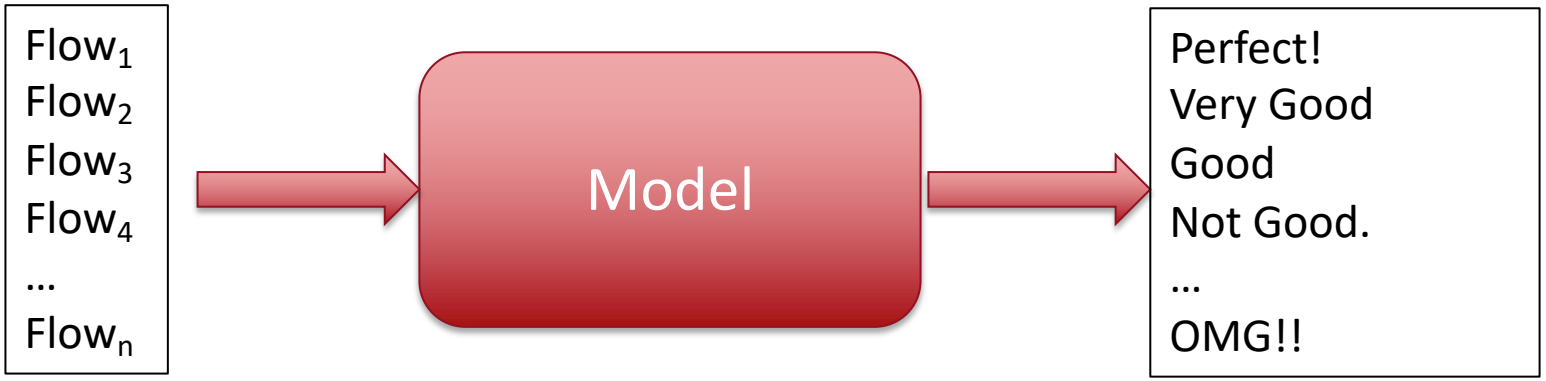




- QoR collected after tech-map
  - Delay and Area
  - 14nm

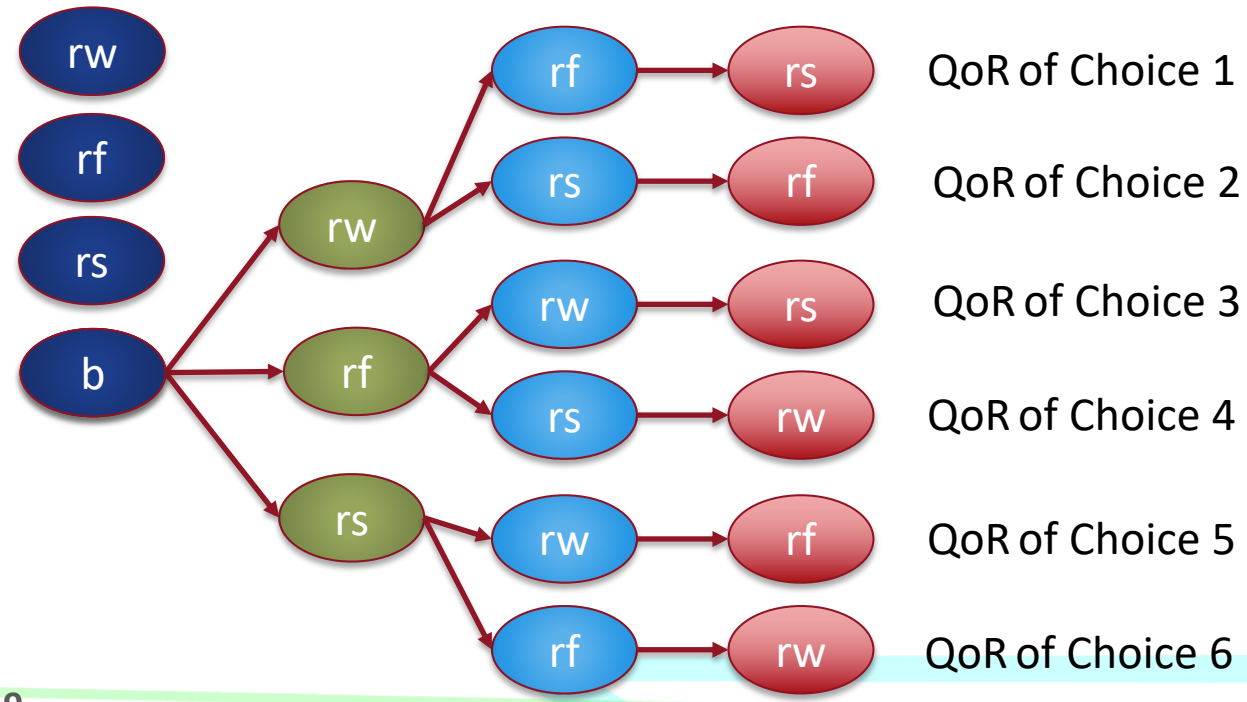
Objective

- Performance estimation



Definition and Search Space

- Decision make in design flows
  - Case 1: none-repetition flow
    - Each transformation appears **only once** (+ map)
      - Example: n=4,balance (b), rewrite (rw), refactor (rf), resub (rs)



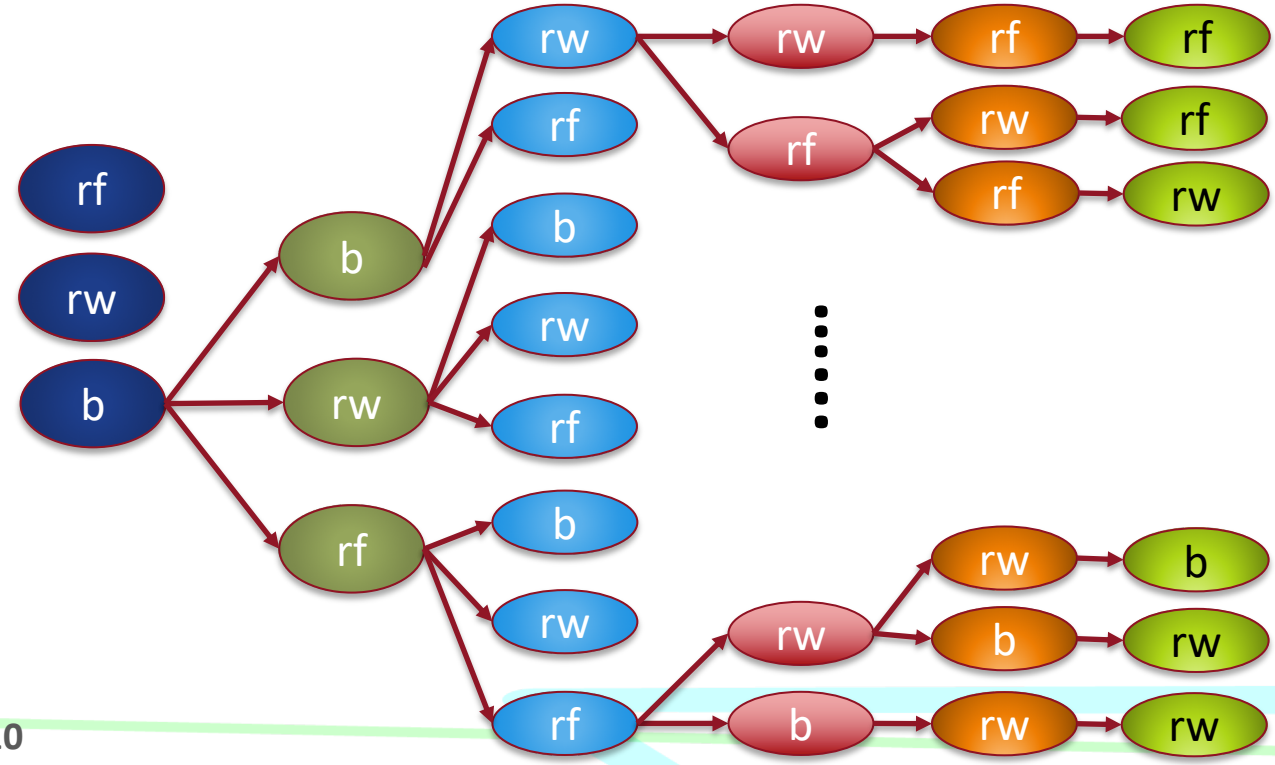
Search space

$$f(n) = n!$$

4!=24

Definition and Search Space

- Decision make in design flows
  - Case 2: *m*-repetition flow
    - Each transformation appears ***m* times** (+ map)
      - Example: m=2, n=3, balance (b), rewrite (rw), refactor (rf)



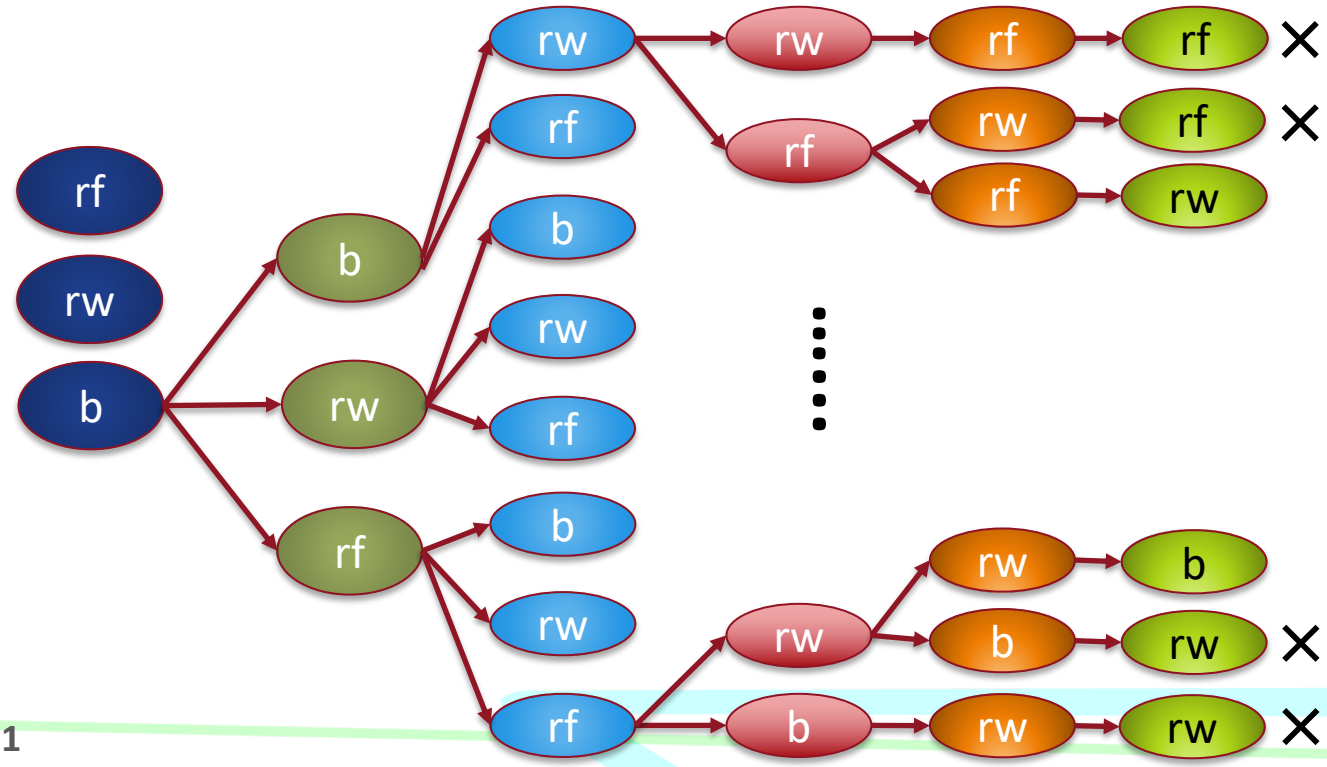
Search space

$$f(n) = \frac{(n \cdot m)!}{(m!)^n}$$

(2*3)!/(2!)^3 = 90

Definition and Search Space

- Decision making in design flows
  - Different flow likely produces different QoR
    - Find the Best (Angel) and Worst (Devil)



n=6

m=4

≈ 3.25e+15

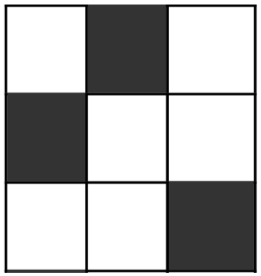
Approach

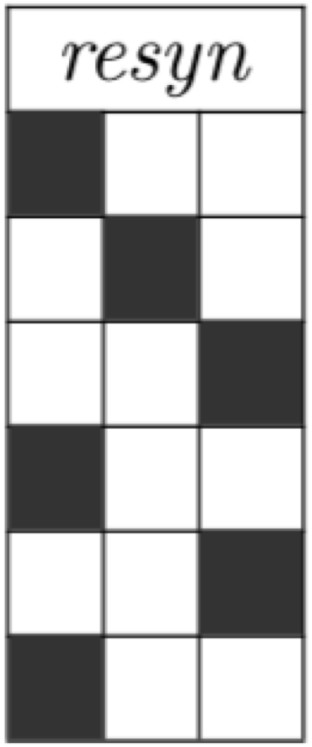
- MultiClass Classification
  - Modeling
    - Features
    - Ground truth
  - CNNs-Classifier
    - Why Deep Learning ? Why CNNs ?
    - Architecture
    - Training
    - Inference: finding best/worst flows by

Features – Flow in One-hot Matrix

- Flow in One-hot Matrix
  - Example: balance (b), rewrite (rw), rw -z (rwz)
    - b = [1 0 0]
    - rw = [0 1 0]
    - rwz= [0 0 1]
  - Flow: rw -> b -> rwz = [rw;b;rwz]

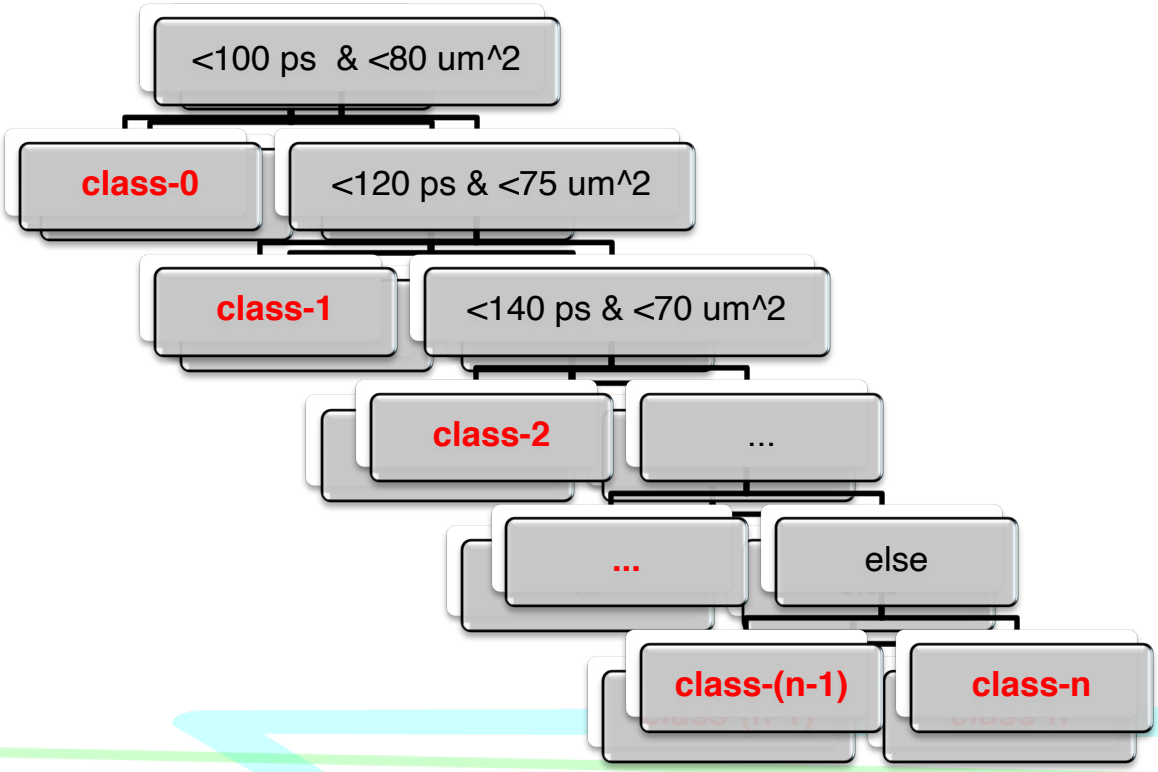
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$





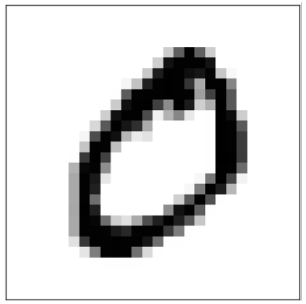
Ground Truth – Performance Class of Flows

- Labels of flows
  - Labeling rule 1: single metric (e.g. delay)
  - Labeling rule 2: multi-metric (e.g. delay+area)



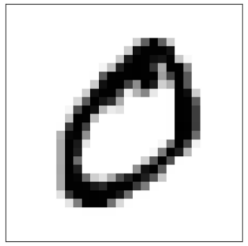
Convolutional Neural Network

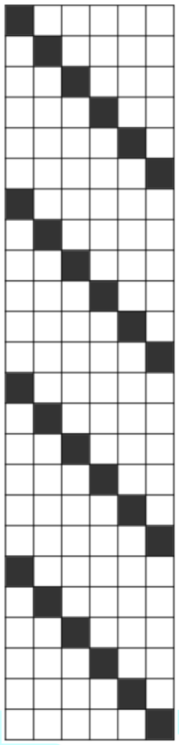
- Convolutional Neural Networks (CNNs)
  - Deep, feed-forward artificial neural networks
  - Multilayer perceptron
    - Conv, Pool, Fully connected layers
- Why CNNs Classifier?
  - “0” of hand writing



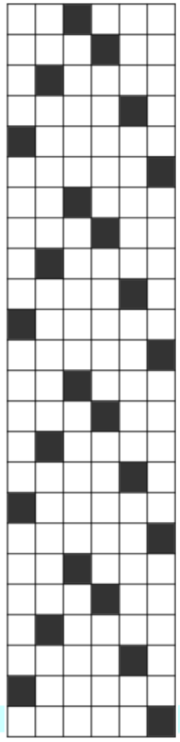
Convolutional Neural Network

- Why CNNs–classifier?
  - “0” of hand writing
  - “0” of Logic Synthesis (ABC)





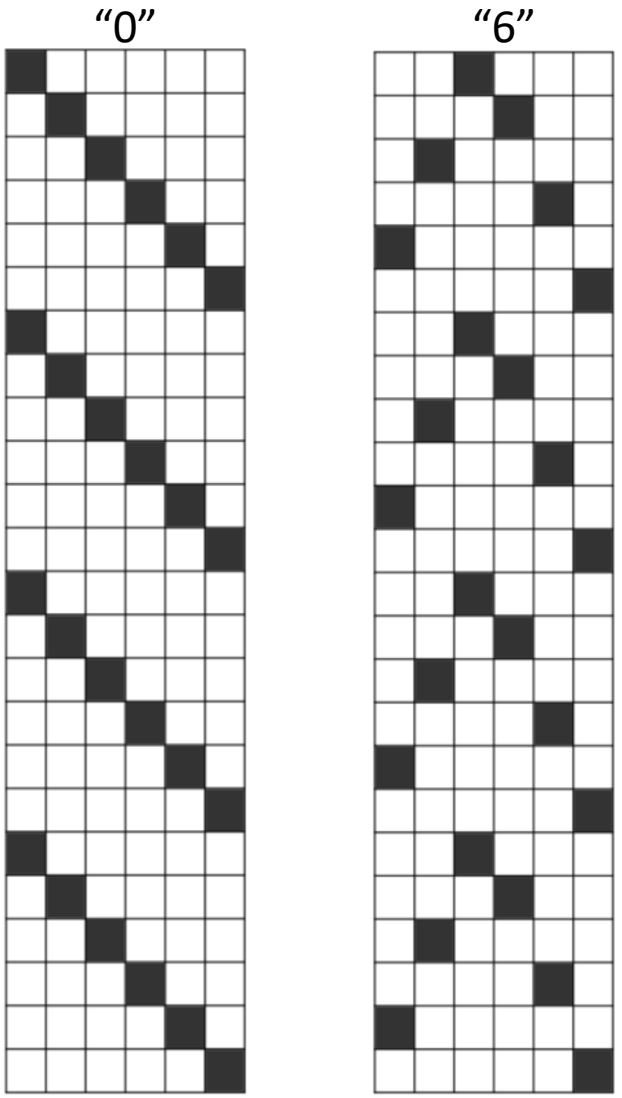
“0”



“6”

Convolutional Neural Network

- Why CNNs–classifier?
  - "digits" of flows
    - shape of best flow – "0"
      - Global or local shape
    - shape of worst flow – "6"
- CNN classifier
  - 2 Conv + 3 Dense
    - Softmax dim = 6
  - Training
    - RSMProp + SELU



Code: https://github.com/ycunxi/FLowGen-CNNs-DAC18

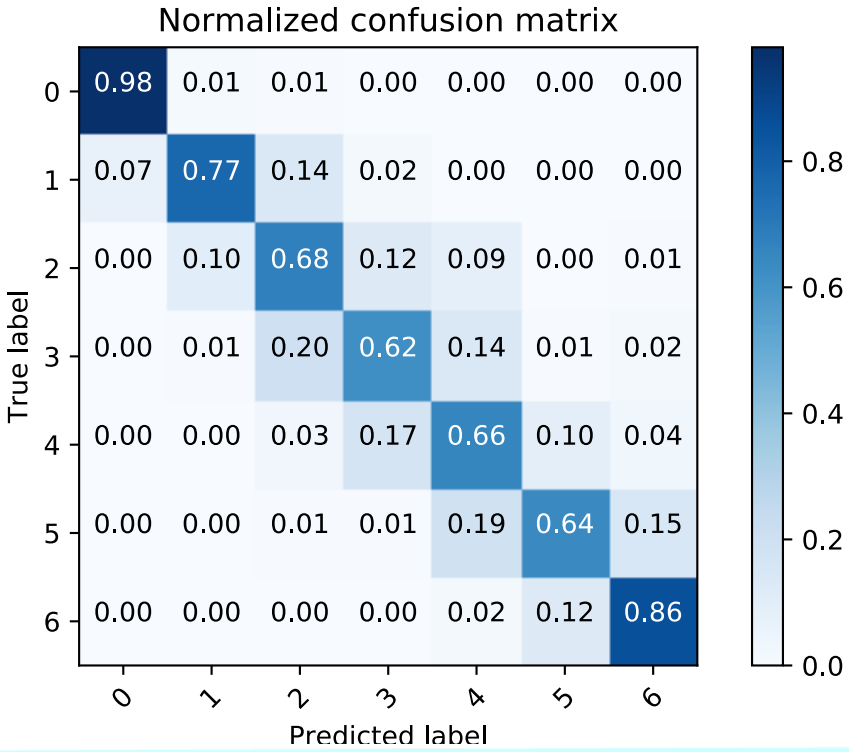
Datasets

- Training/testing inputs
  - Three designs
  - ABC “map” mapper + 14nm
  - ~300,000 data points
    - Failure cases eliminated
  - 20% for training, 80% for testing

Design	**64-bit Mont**	**64-bit ALU**	**128-bit AES**
Data points	99,997	100,000	99,737
Features		**Labels**	
Flow	(1, 144) → (24,6)	Delay, Area	(1,2)

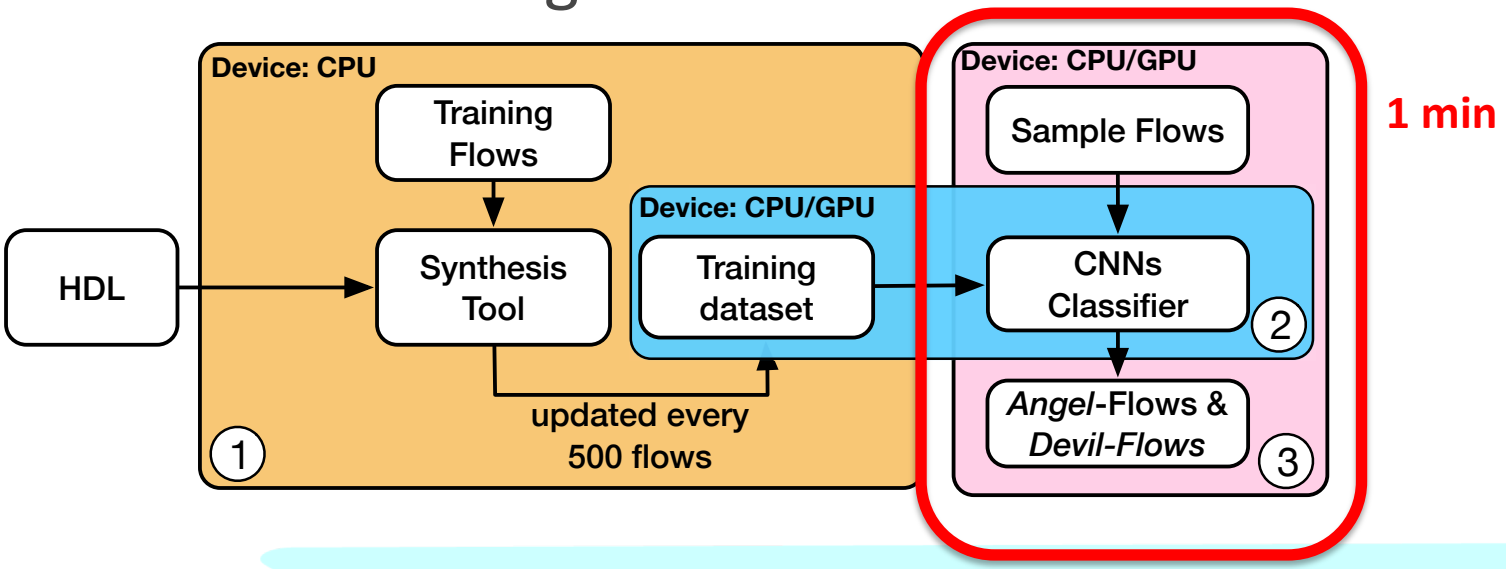
Results – Prediction Accuracy

- Confusion matrix
  - Design: 128-bit AES

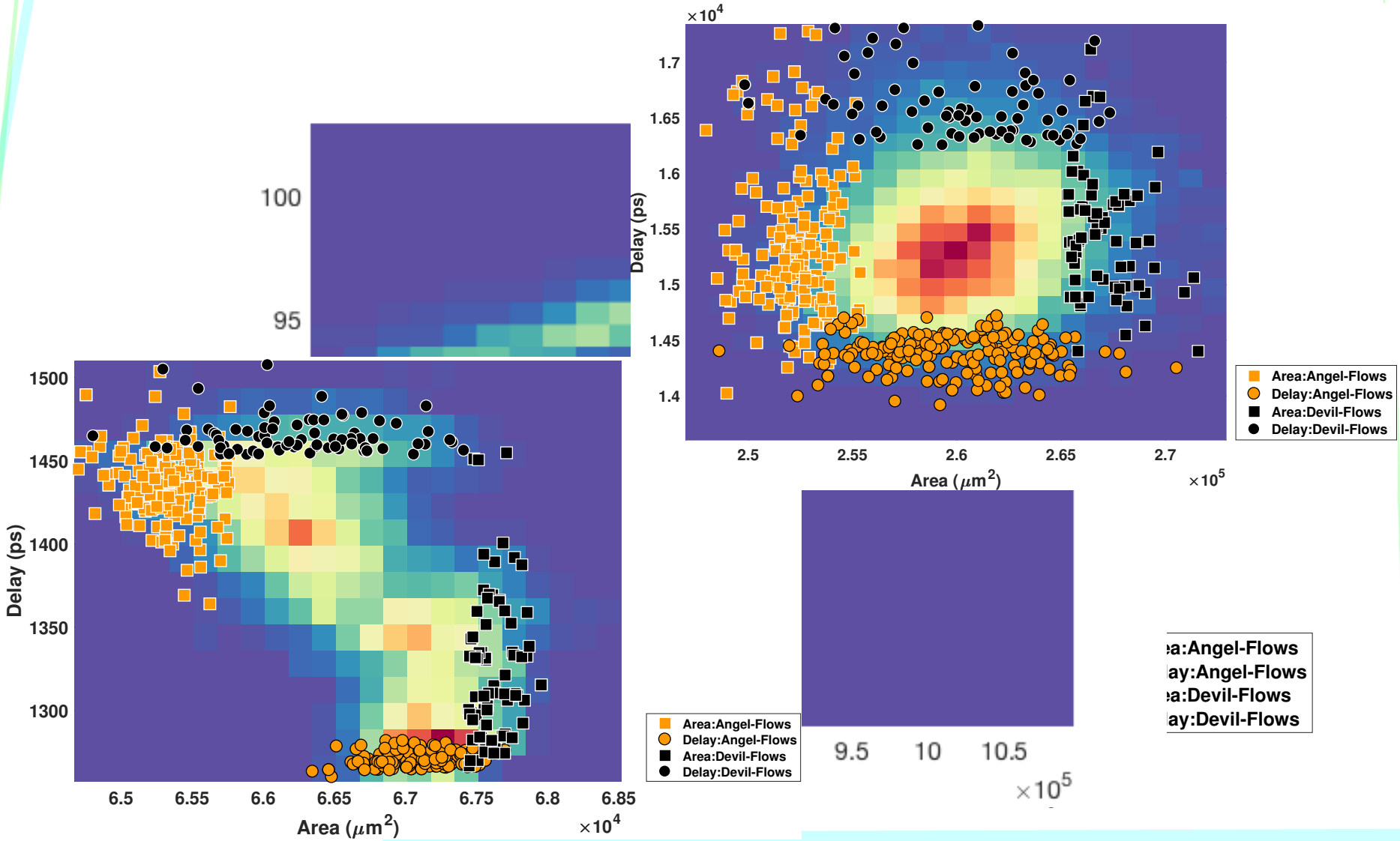


Results – Generated Flows

- System overview
  - Training set generation
    - Online collection
  - Training the classifier
  - Inference on a large number of untested flows



Results – Generated Flows



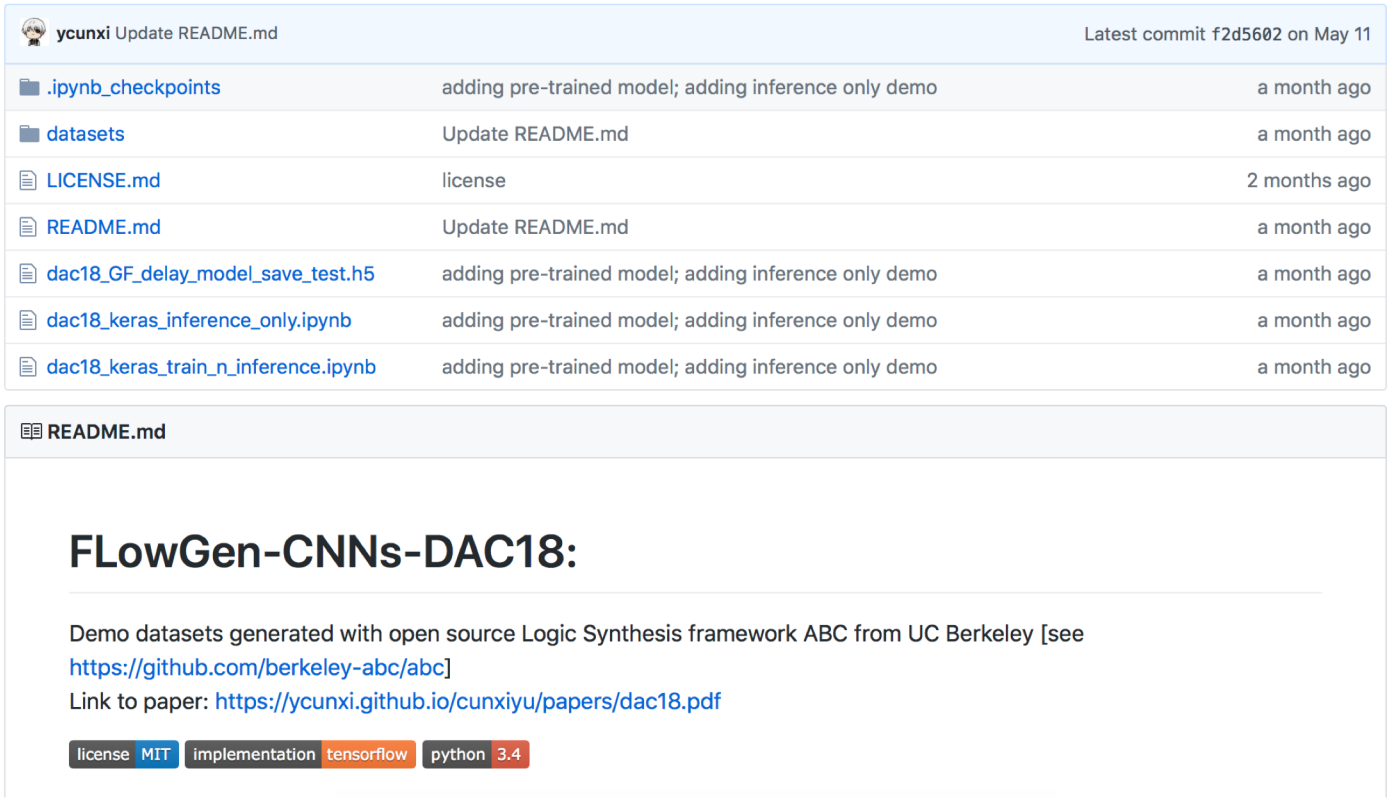
Conclusion

- Generic Flow Classification
  - Deep Learning, CNNs
  - Demonstration with Logic Synthesis Tool
    - Design specific
    - Overall >70%, best/worst class > 86%
      - Three realistic designs
- Future Work
  - Cross-layer design flows (end-to-end)
  - Transfer-learning cross technology/designs
  - Explainability
  - Open database

Open Source

- Code & Datasets

  https://github.com/ycunxi/FLowGen-CNNs-DAC18



ycunxi Update README.md — Latest commit f2d5602 on May 11

.ipynb_checkpoints	adding pre-trained model; adding inference only demo	a month ago
datasets	Update README.md	a month ago
LICENSE.md	license	2 months ago
README.md	Update README.md	a month ago
dac18_GF_delay_model_save_test.h5	adding pre-trained model; adding inference only demo	a month ago
dac18_keras_inference_only.ipynb	adding pre-trained model; adding inference only demo	a month ago
dac18_keras_train_n_inference.ipynb	adding pre-trained model; adding inference only demo	a month ago

README.md

FLowGen-CNNs-DAC18:

Demo datasets generated with open source Logic Synthesis framework ABC from UC Berkeley [see https://github.com/berkeley-abc/abc]
Link to paper: https://ycunxi.github.io/cunxiyu/papers/dac18.pdf

Thank you!