

# **ERI Design: POSH**

*Automated AMS IP generator for CMOS technologies*

FA8650-18-2-7853

Under contract starting June 25, 2018

---

Tony Levi  
Mike Chen  
Sandeep Gupta

Review meeting at USC:  
Andreas Olofsson: [andreas.olofsson@darpa.mil](mailto:andreas.olofsson@darpa.mil)  
Christal Gordon: [christal.gordon.ctr@darpa.mil](mailto:christal.gordon.ctr@darpa.mil)  
Abi Sivananthan: [abirami.sivananthan.ctr@darpa.mil](mailto:abirami.sivananthan.ctr@darpa.mil)  
Len Orlando: [pompei.orlando@us.af.mil](mailto:pompei.orlando@us.af.mil)

19 September, 2018 12.30pm-3.30pm PHE 223, University of Southern California





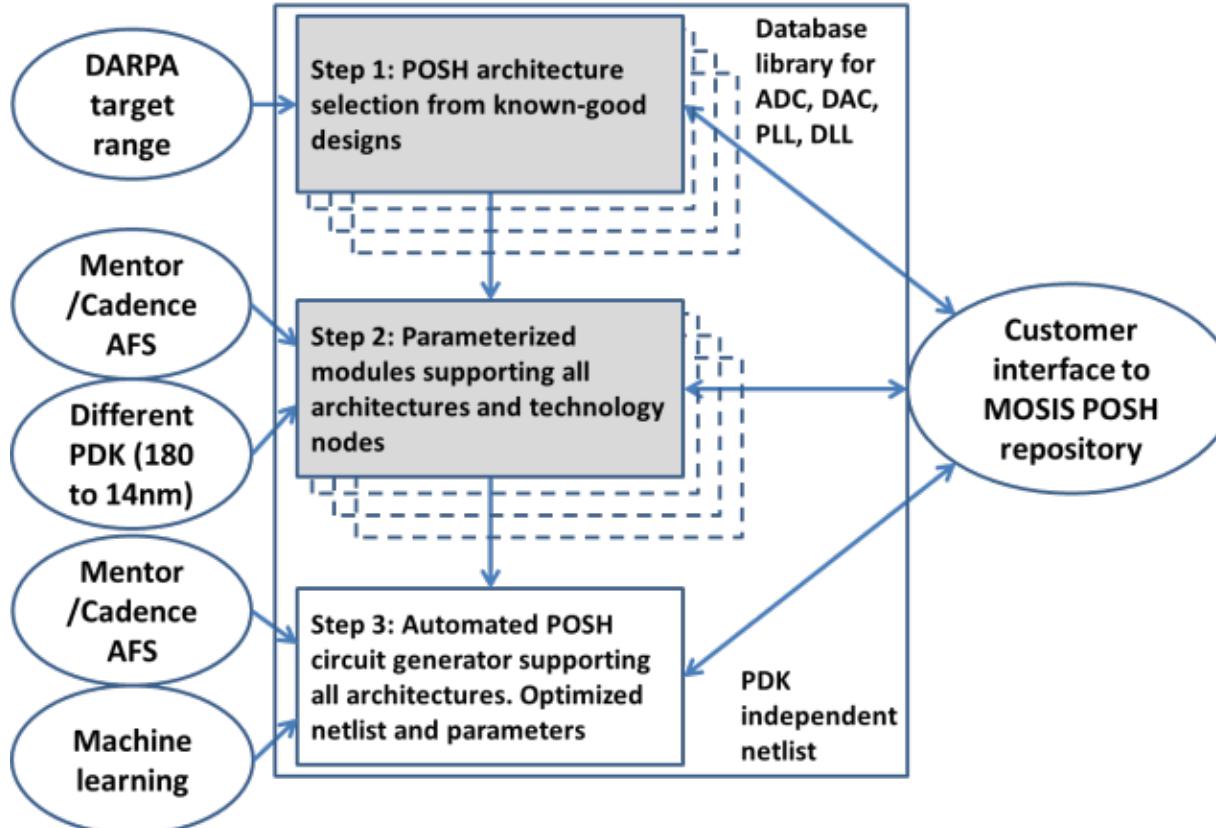
# Agenda

---

- **Discussion and presentation of automated analog mixed signal (AMS) schematic circuit generator for CMOS technologies will include:**
- **item 1:** overview of program at 2+ months and reminder of 6-month goals
  - Accomplishments/Highlights summary – Design examples using machine learning, development of meaningful objectives, and some progress with lite version, progress with systematic tools for machine learning design
  - Detailed activity report “diary” – are in monthly reports. Weekly meetings at USC
  - Detailed description of technical discoveries/breakthroughs – development of insight and approach
  - Schedule status compared to original SOW – have followed expectations set in original SOW. Progress with software design, specific worked examples, tape outs in 14nm continue
  - Resource/staffing/contracting status – list of individuals involved
  - Top challenges/Issues that concerns you – additional resources will enable potentially productive avenues of exploration (e.g., submitted SRC proposal)
  - Papers/presentations accepted for publication – ICCAD invite (MC) for a AMS related talk
  - Goals for the next reporting period – by September 30th anticipate technical report will show worked example of team approach
  - Integration exercise – Anticipate a venue that enables a show-and-tell approach using material available at MOSIS repository and presenting worked examples to others on how to use. There will likely be a netlist sample (in each category, ADC/DAC/PLL/DLL) with documentation of how to use them
- **item 2:** technical presentation of worked design examples
  - ADC and DPPLL
- **item 3: summary and discussion**
  - May include examples of TSMC65nm Time-based ADC, SAR ADC, DAC, Digital PLL, DLL

# Program overview of automated AMS schematic CMOS circuit generator

- Develop automated analog mixed signal (AMS) CMOS schematic design tools
- Open source design tools
- Potentially of great value, in part because contemporary AMS is an expensive manual (human in the loop) task
- Adopt a narrow focus on key AMS elements: PLL, DLL, ADC, and DAC to enhance probability of early success
- Architecture choices are *limited* and mostly digital
- Supported by library of validated known good designs (KGD) including GF14LPP-XL



**USC** University of  
Southern California



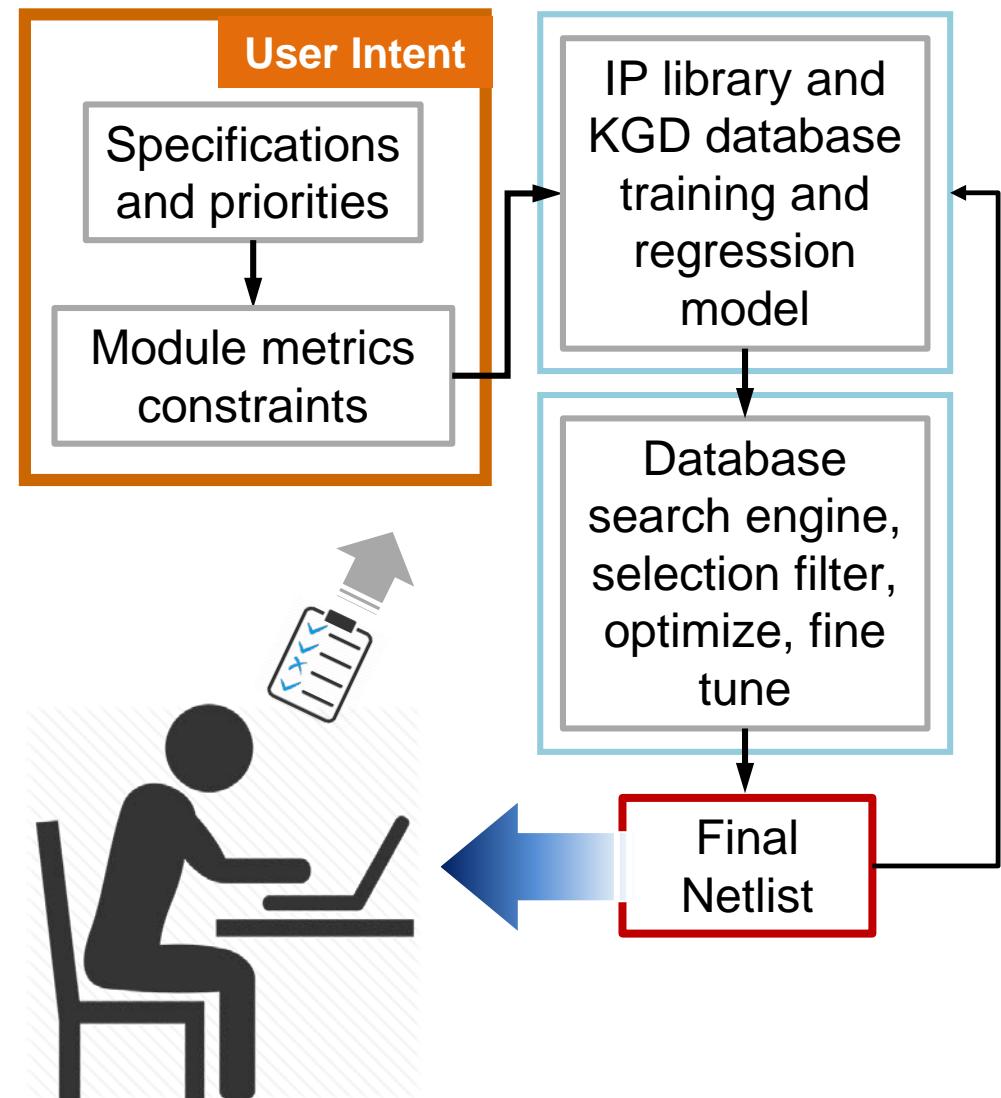
The **MOSIS** Service



**GLOBAL**  
**FOUNDRIES**

# Summary

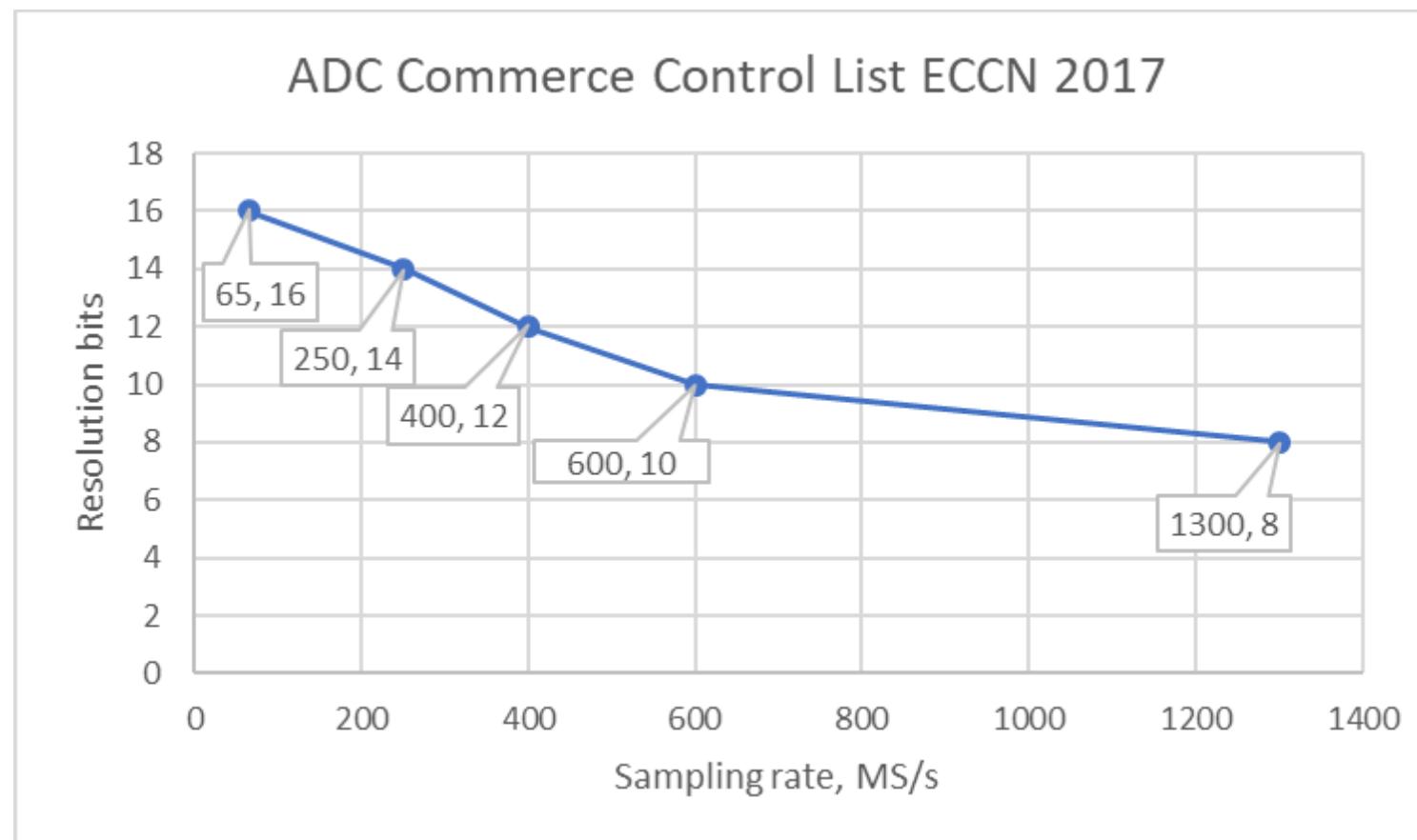
- POSH TA-2: *Automated open-source mixed-signal IP schematic generation and distribution via MOSIS POSH repository*
- Leverage existing USC circuit designs in 180nm, 65nm and 14nm FinFET with access to advanced GF technology nodes
- Architectures for advanced technology IP blocks that meet or exceed performance metrics at month 48:
  - PLL Range: 10 MHz – 10 GHz
  - DLL Range: 10 MHz – 10 GHz
  - Analog to Digital Converters (ADC) range: 1 – 10,000 MS/s
  - Digital to Analog Converters (DAC) range: 1 – 10,000 MS/s
- Key aspect of technical approach is use of machine learning (e.g., ANN) and optimization (e.g., convex) around known-good design (KGD) points
- Library of *validated* open-source mixed-signal IP circuit modules with initial delivery to repository at month 6





# Export control and choice of specifications

- Specification limitations
  - Since IP will be open-source, specifications limited by export control
  - Example: ADC, notice implied specification 10,000 MS/s includes 8b resolution





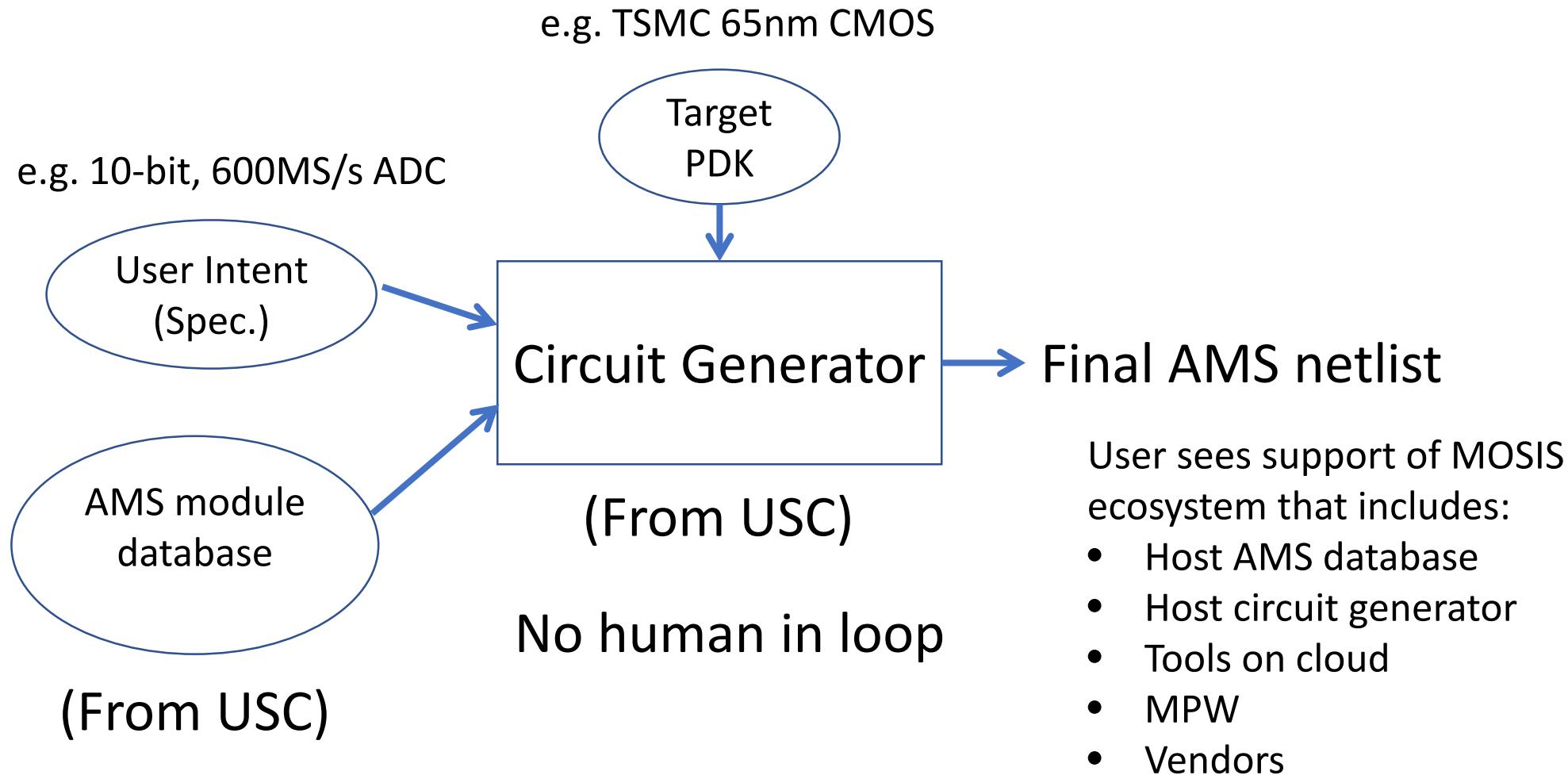
# Program schedule

Deliverables described by task.		Milestones at month after start of program is <b>documentation released to MOSIS POSH repository</b>							
Task #	Description	Phase 1				Phase 2			
1	Background IP documentation of known-good AMS circuits	6	12	18	24				
2	Documented physical validation of 14nm AMS circuit blocks (tape out month 6 and 18)	6	12	18	24				
	Option: Documented physical validation of 7nm AMS circuit blocks (tape out month 30)					30	36	42	48
3	AMS IP design methodology and fast circuit generator for PLL, DLL, ADC, DAC	6	12	18	24				
	Updates and maintenance of AMS IP design methodology and fast circuit generator					30	36	42	48
4	Collaboration and management	Monthly video conference collaboration calls. Monthly and quarterly reporting requirements to government. Final report Phase 1 (month 24) and final report Phase 2 (month 48)							
5	Documented technology transition plan including description of fee-based added-value services				24				48



## Detail USC personnel

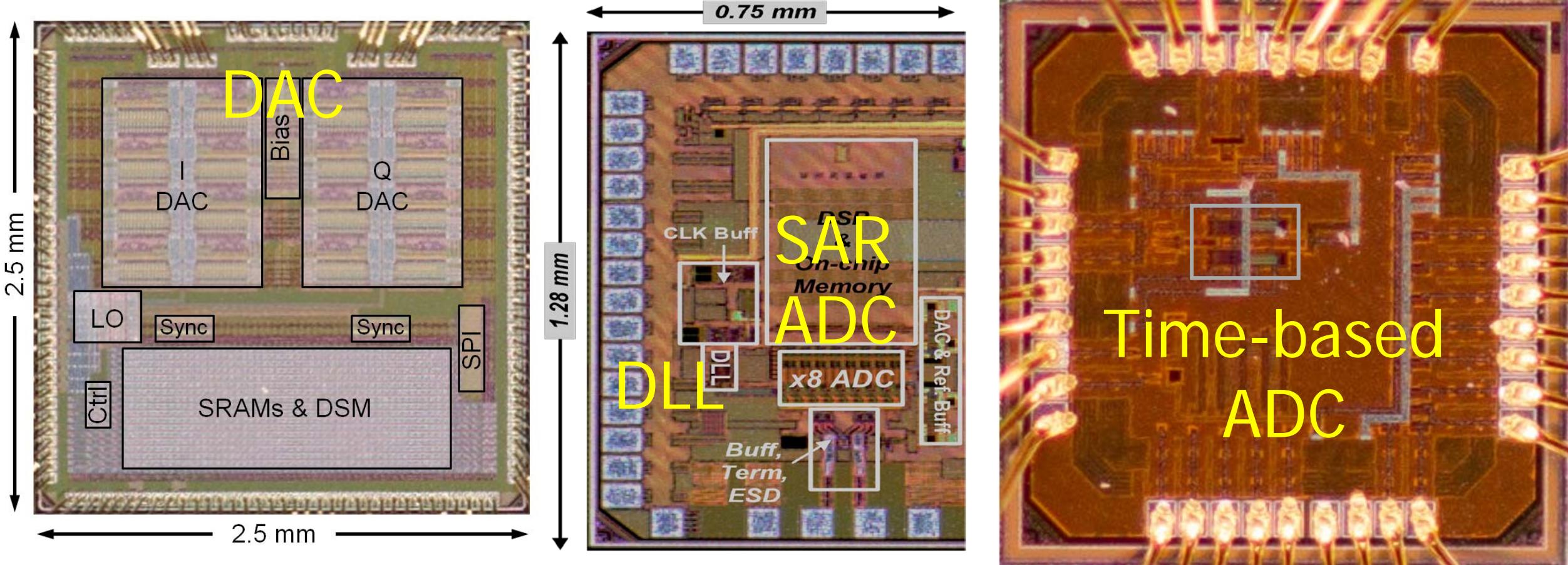
Name	Roles and responsibilities
Tony Levi	PI and overall management. Optimization algorithms and AMS testing
Mike Chen	AMS archive, AMS KGD, architecture selection and circuit generator
Sandeep Gupta	Circuit generator
<b>Students:</b>	
Arjun Sathyamoorthy	Circuit generator, local convex optimization, lite tool
Mutian Zhu	Circuit generator, starting with systematic tools to help ML-based design
Mohsen Hassanpourghadi	ADC (library, circuit generator, KGD, ML)
Praveen Sharma	ADC (library, circuit generator, KGD, ML)
Jaewon Nam	ADC, DLL (library, circuit generator, KGD)
Qiaochu Zhang	PLL (library, circuit generator, KGD, ML)
Subhajit Chowdhury	DLL (library, circuit generator, KGD)
Shiyu Su	DAC (library, circuit generator, KGD)



## Highlights of Our POSH Approach

1. Build parameterized **module library** for pre-selected ADC/DAC/PLL/DLL architectures
2. Machine-learning based AMS **circuit generator** (netlist) that expands large design space
3. **Known good designs** (silicon proven) of ADC/DAC/PLL/DLL in TSMC 65nm CMOS as the bases for #1 and #2.

## Baseline: Known Good Designs (TSMC 65nm CMOS)



- Working silicon chips and more measured data to come ☺
- Lab demo is available today!

## Off to an exciting start

1. Built initial parameterized module library (in Cadence) for single pre-selected ADC/PLL architecture
2. Successfully applied machine-learning based AMS circuit generator for two design examples from end to end!

More advanced algorithms and broader libraries are being lined up to enhance our current capability.

### Implications:

- Validate our direction
- Upcoming integration exercise

# Automated Analog and Mixed Signal Design Synthesis

Low-Resolution Medium-speed ADC Example

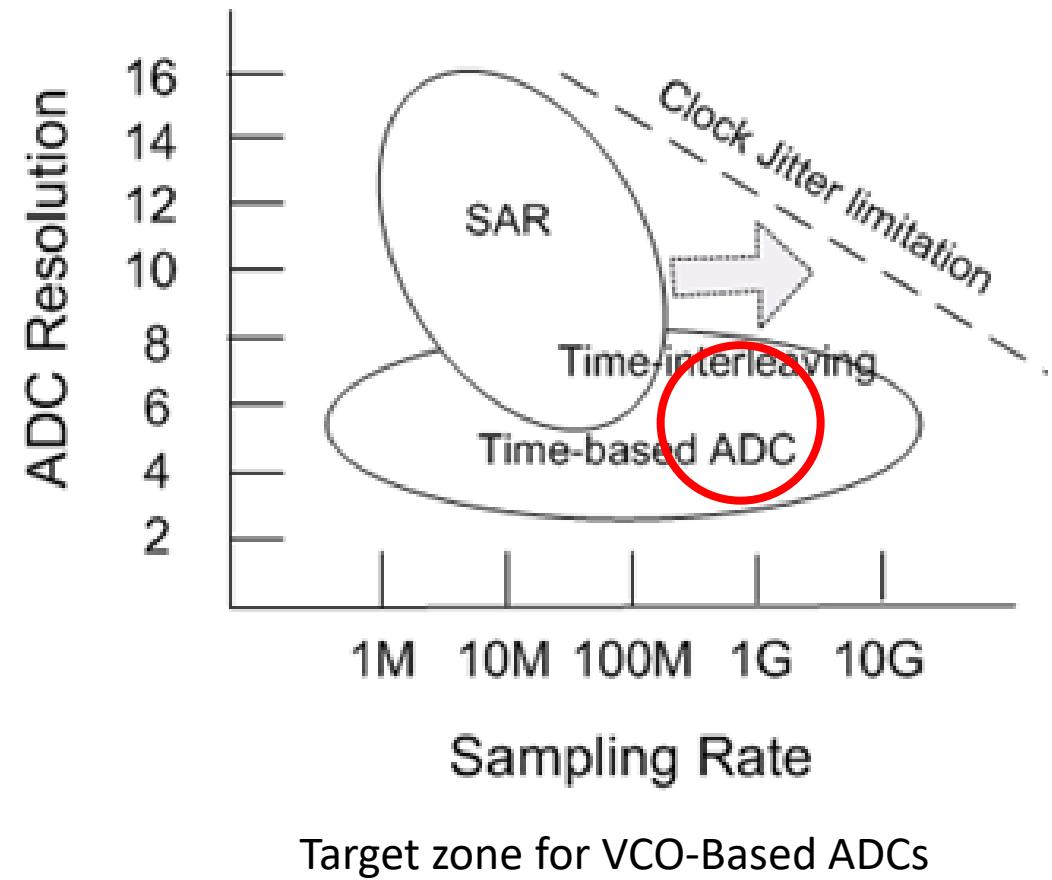
VCO-Based ADC in TSMC 65nm

Praveen Sharma

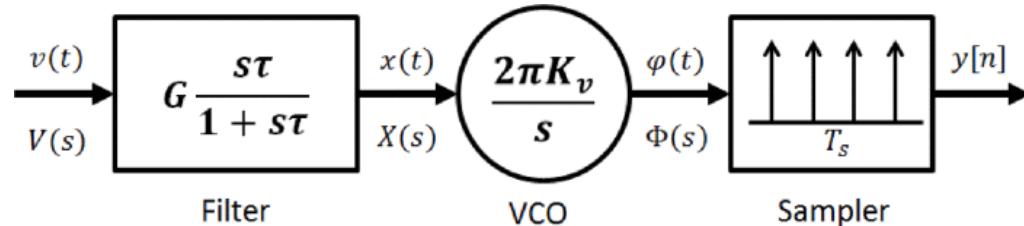
# Outline

- Introduction
- VCO-Based ADC
- Machine-Learning Based AMS Circuit Generator
- Module Breakdown
- Characterization of modules
- ANN based regression model
- Search from specifications

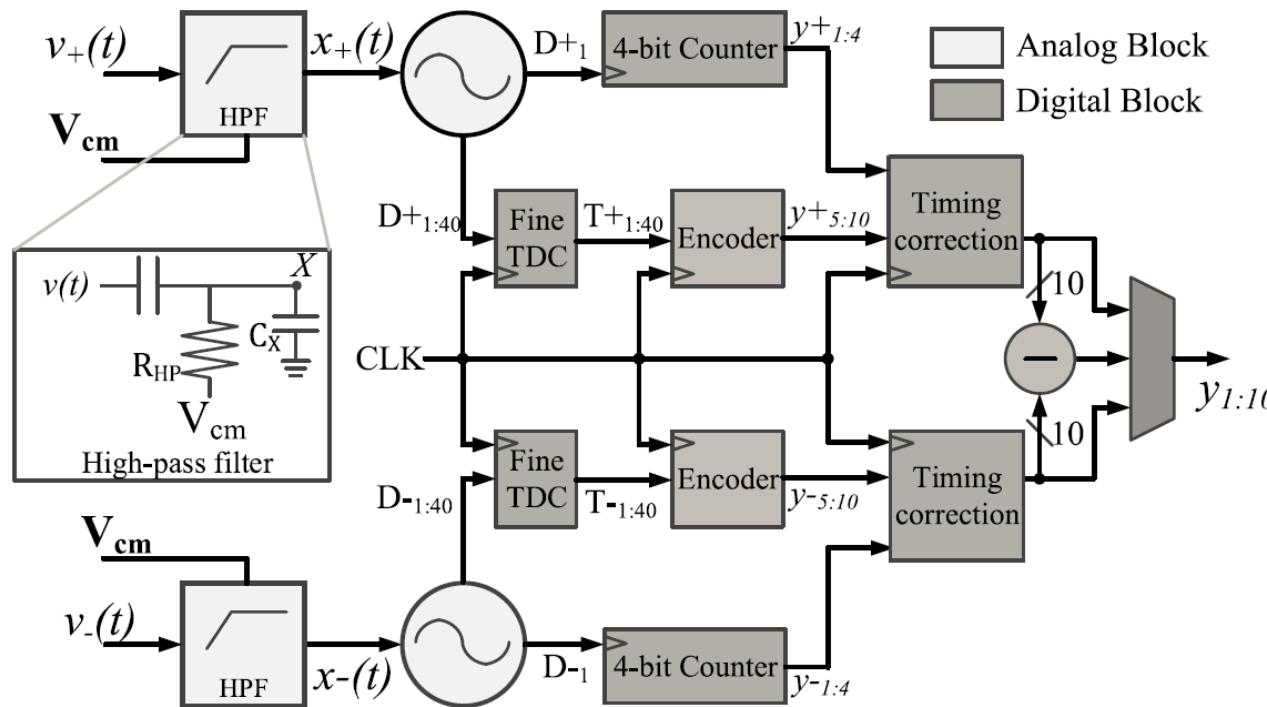
- Low resolution ( 4-8 bits)
- Medium speed ( 500MHz - 2GHz)
- Time based measurement



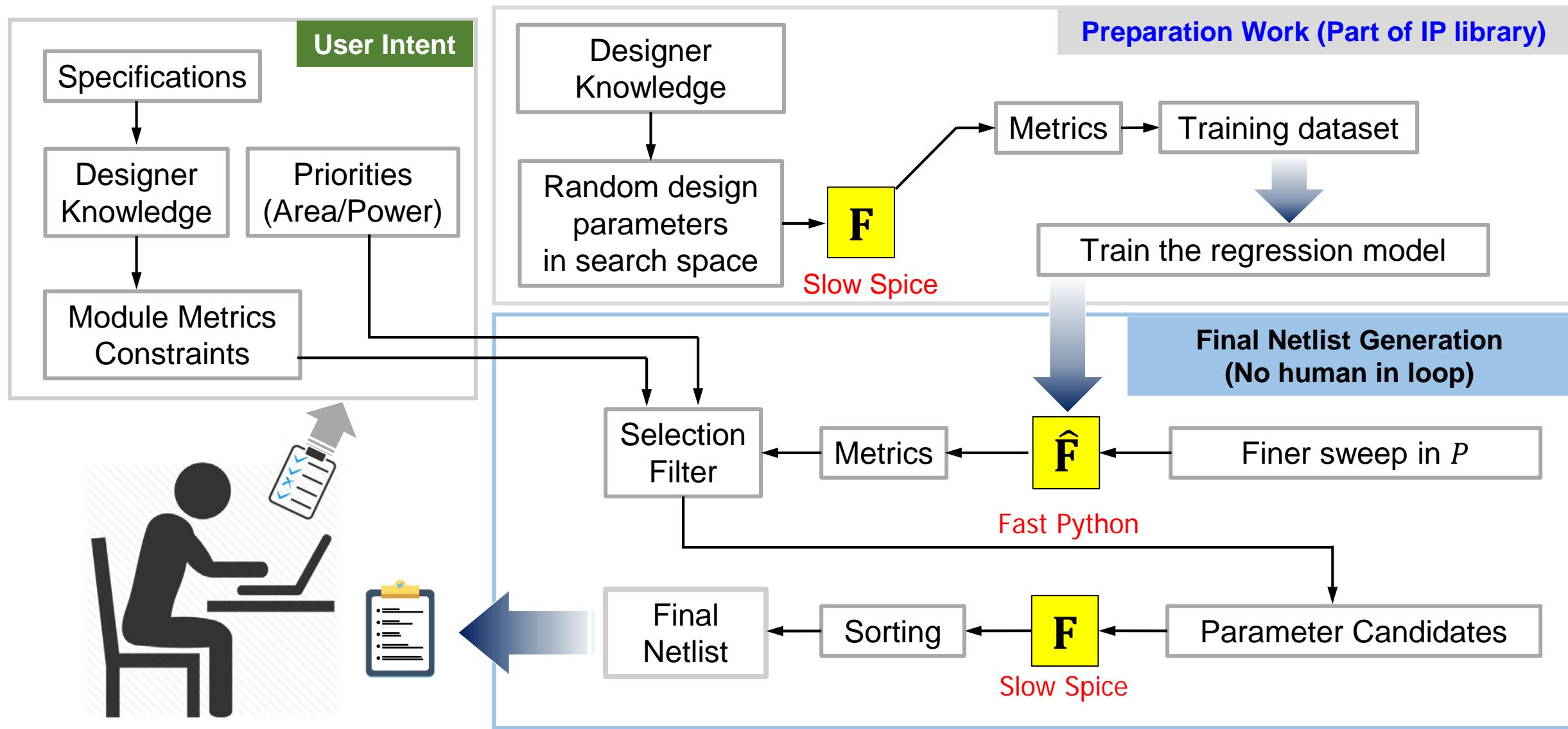
# Introduction



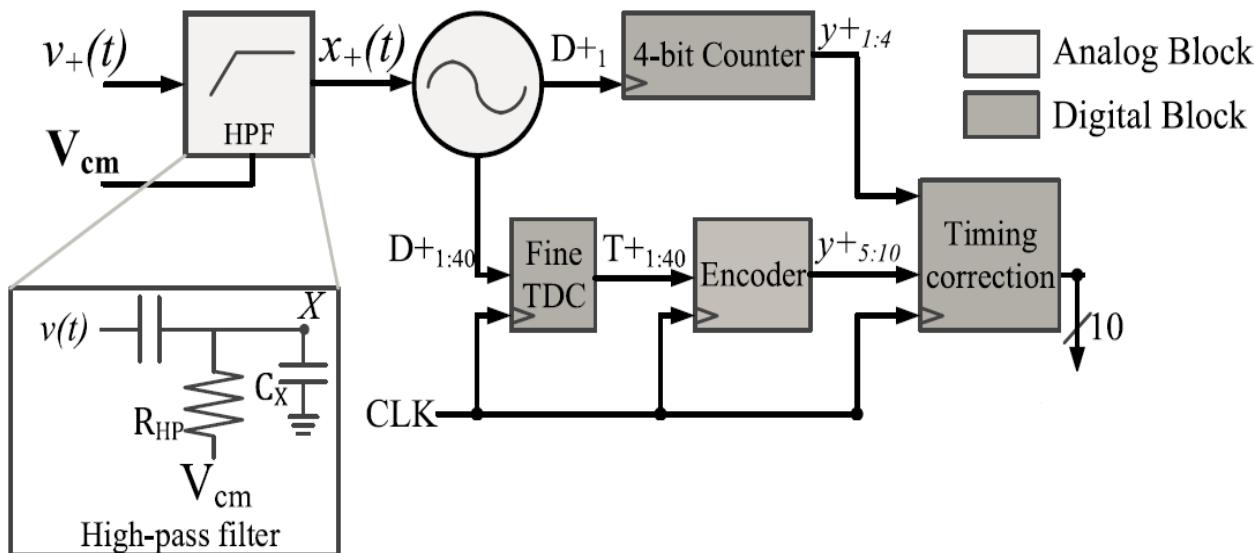
- Strength : Mostly Digital
  - Scales well with feature size
  - Easier to automate design flow
- Conceptually use VCO as integrator, and keep track of phase changed per sampling period
- Overall Known Good design Architecture in TSMC 65nm
- Mix of analog and digital circuit elements



# Machine-Learning Based AMS Circuit Generator

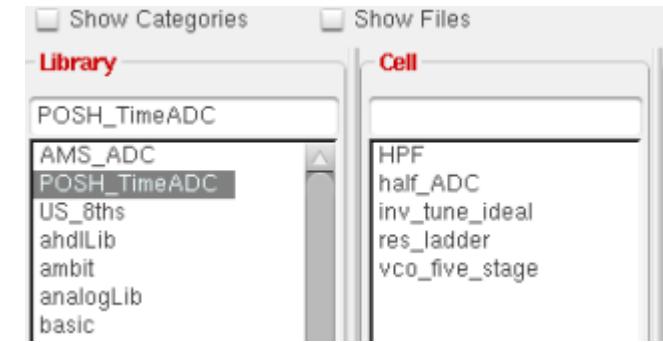
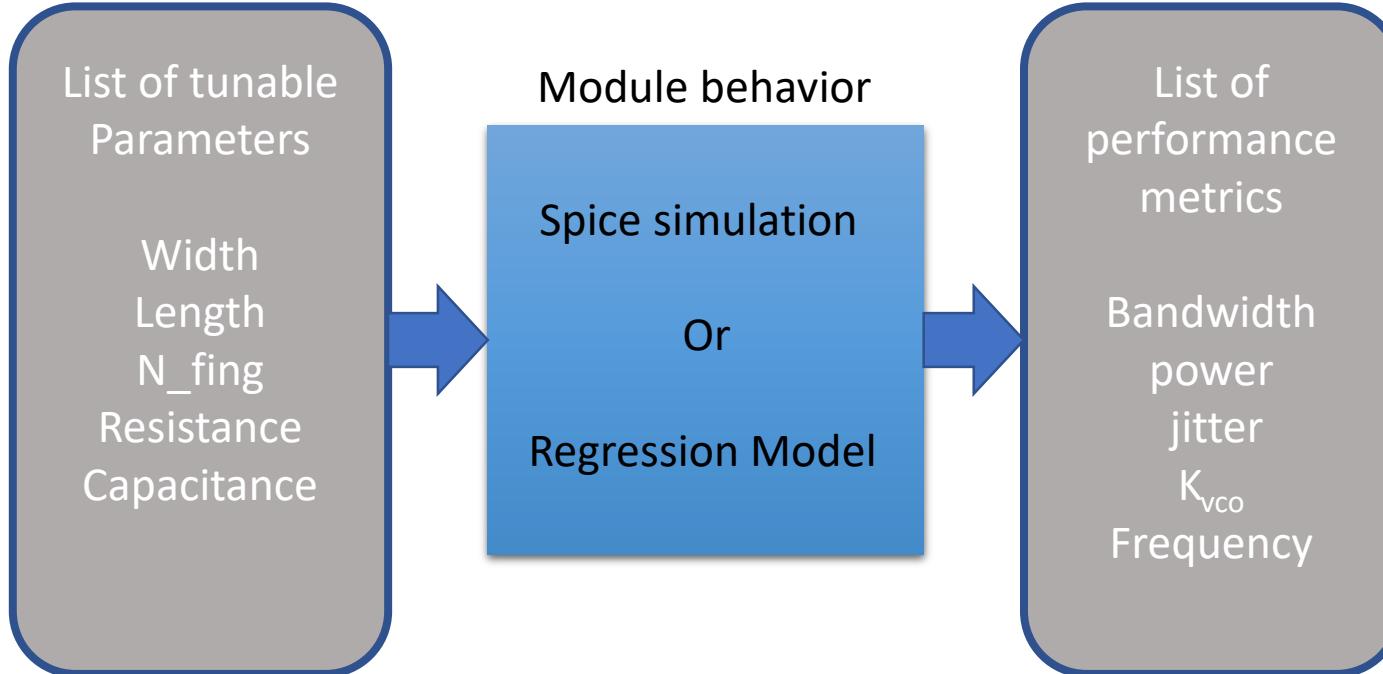


# Module Breakdown



- Simplified Architecture for proof of concept
- Analog Modules
  - High Pass filter
  - VCO
- Digital Modules
  - VCO phase sampler (boundary)
  - Counter
  - ROM bubble encoder

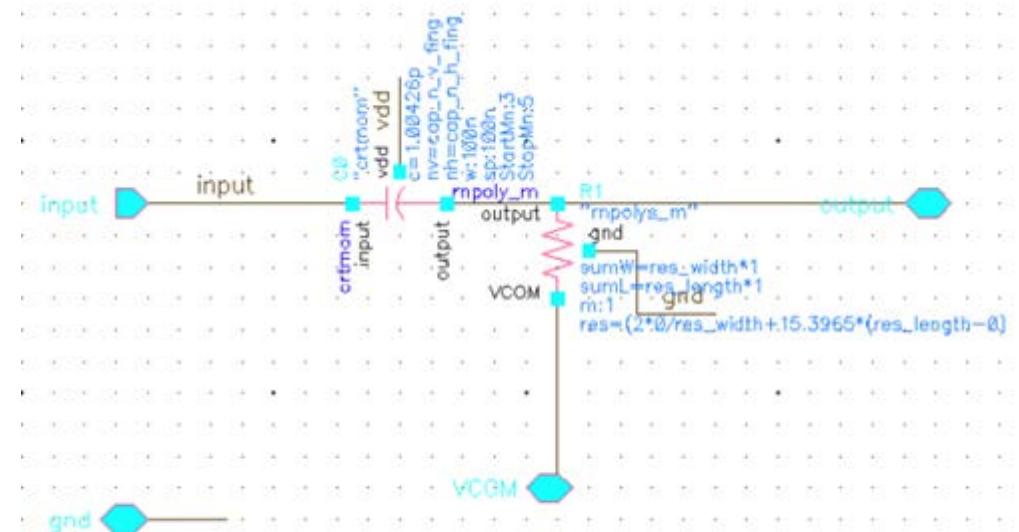
# Characterization of Modules



Parameterized library in TSMC 65nm technology

## Differentiator ( High pass filter)

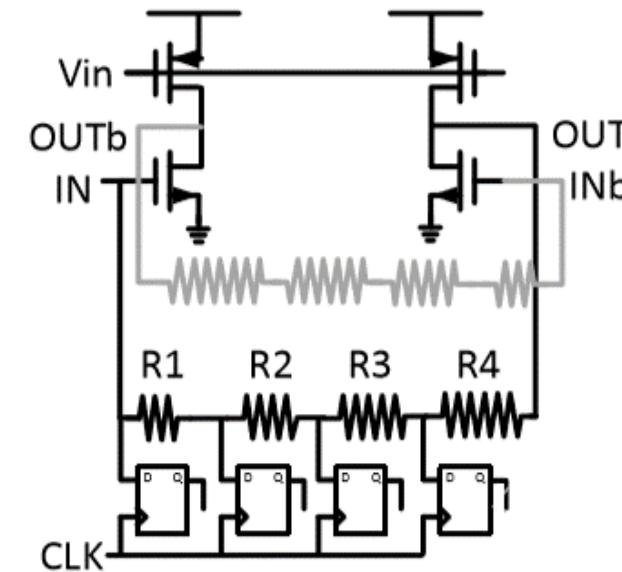
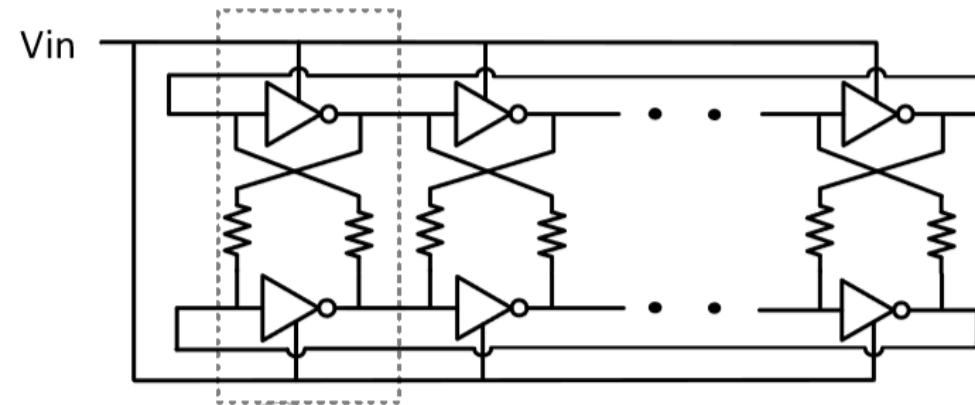
- Parameters
  - Resistance
  - Capacitance
  - length/width
  - $Z_{load}$  (next stage load)
- Metrics
  - Bandwidth
  - leakage power
  - peak attenuation



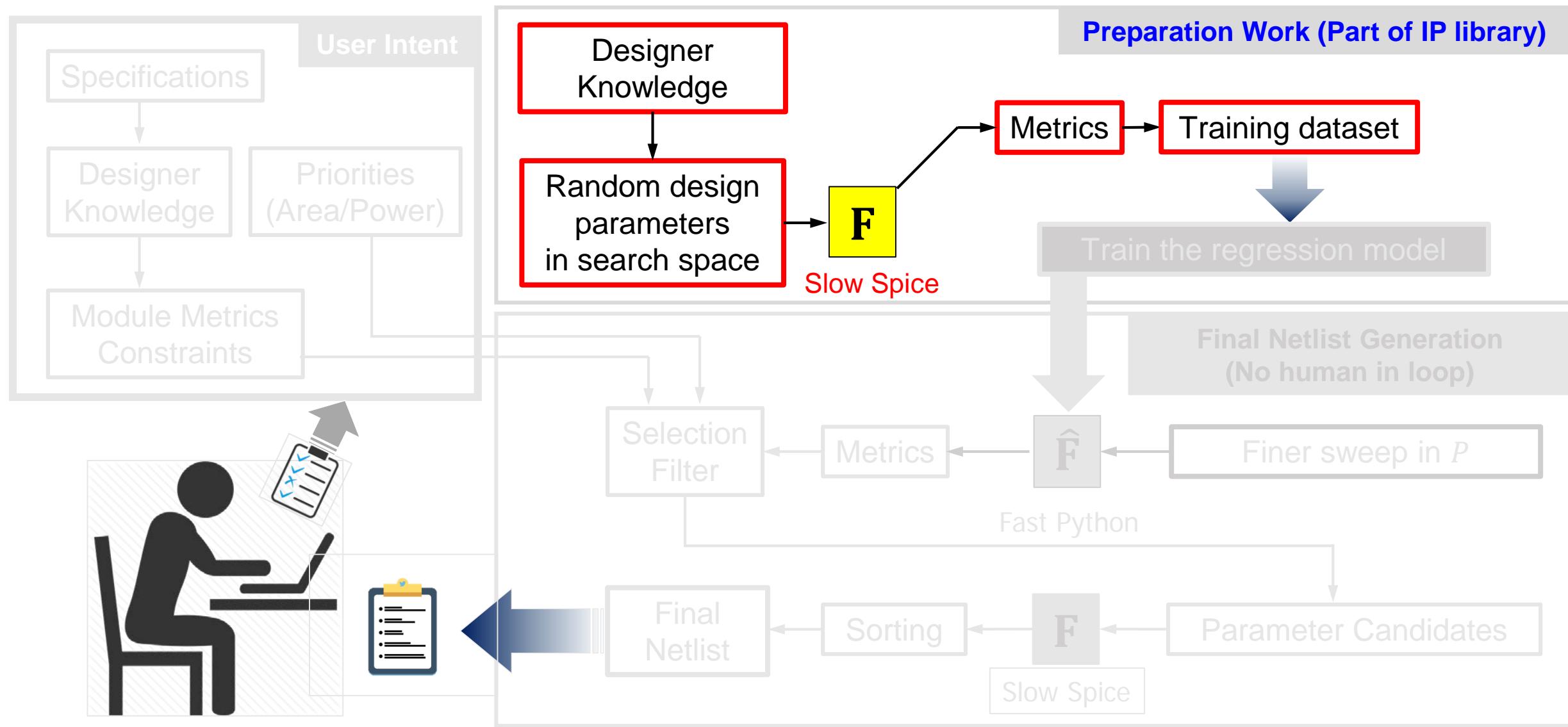
Zout	Cap	length_cap	width_cap	n_fingers_h	n_fingers_v	length_res	width_res	seg spacing	Res Value	Bandwidth
80k	1.00426p	100n	100n	132	132	3.285u	1u	180n	49.9758	1.14Ghz
85k	1.00426p	100n	100n	132	132	3.285u	1u	180n	49.9758	1.09Ghz
90k	1.00426p	100n	100n	132	132	3.285u	1u	180n	49.9758	1.10Ghz

## 5 stage ring oscillator

- Parameters
  - Transistor sizing ( width and n\_fing)
  - $V_{bias}$  to control PMOS current
- Metrics
  - Input linear range
  - $k_{vco}$
  - $f_{nominal}$
  - Output amplitude
  - Power
  - $Z_{load}$  (loading on the previous stage)



# Library Creation



# Training set generation

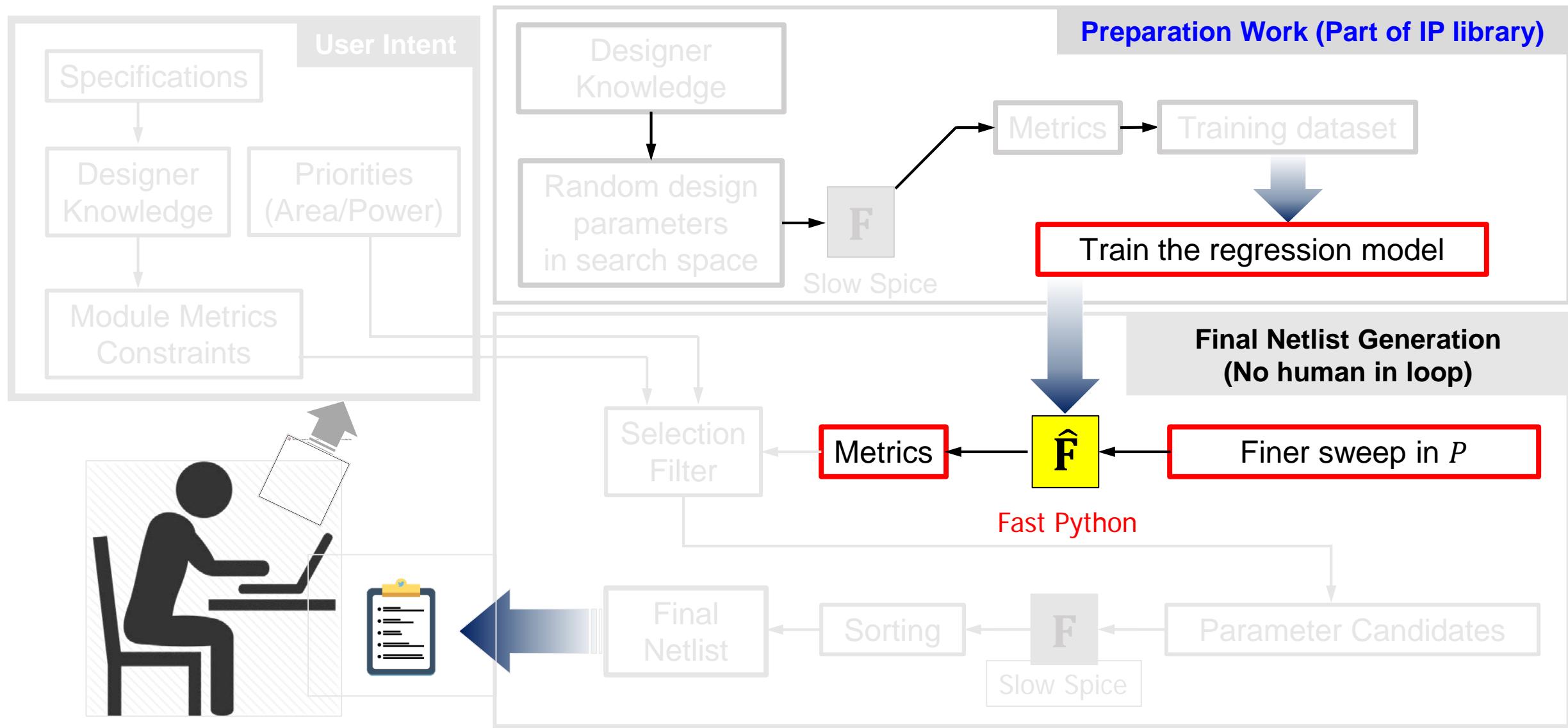
High Pass  
Filter

Design parameter	Sweep range
cap_n_h_fing	50-288 integer
cap_n_v_fing	50-288 integer
res_width	0.25-2 um (multiple of 100nm)
res_length	0.25-10 um (multiple of 100nm)
capacitance	142fF to 4.7pF (derived from above)
resistance	15-150 Ohms (derived from above)

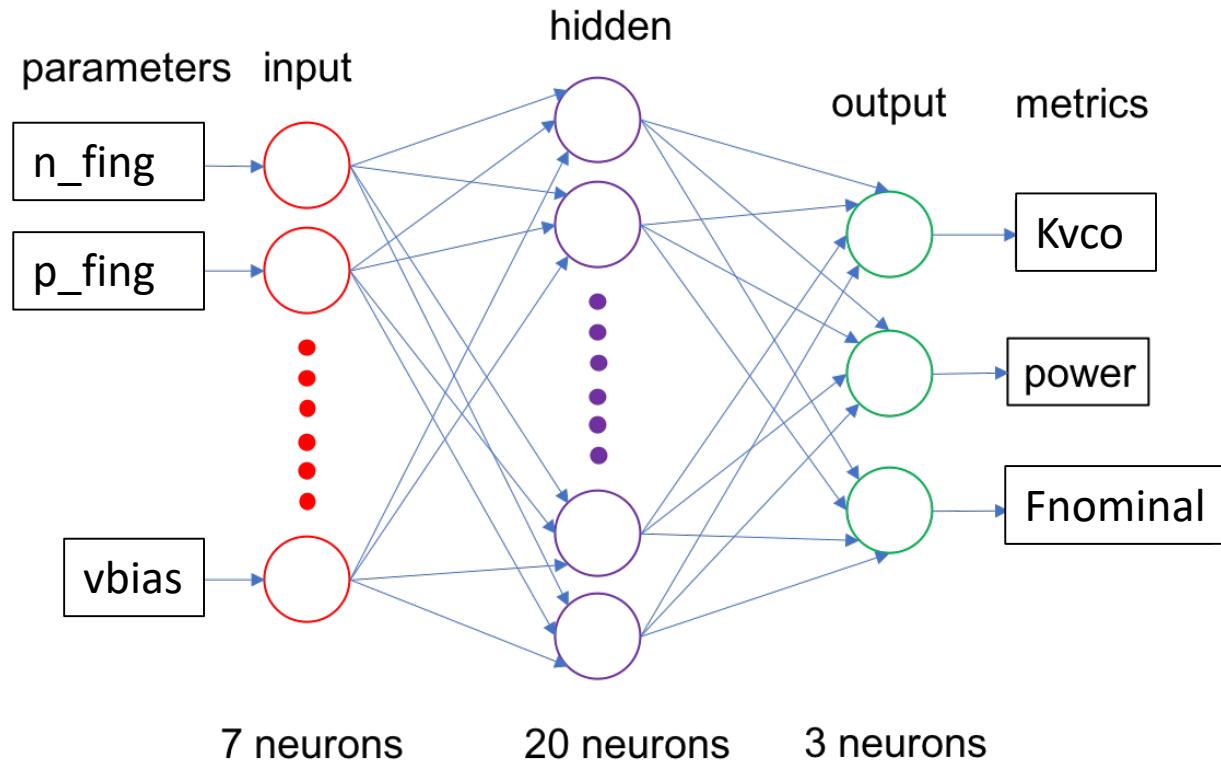
Voltage  
Controlled  
Oscillator

Design parameter	Sweep range
N_fing_n	2-20 int
N_fing_p	2-20 int
Wn	200-600n steps of 100n
Wp	200-600n steps of 100n
Z <sub>load</sub> (following stage)	5-50fF Random floating point

# Library Creation



# Regression using Artificial Neural Network (ANN)



- The optimal number of hidden layers and neurons in each layer to be explored along with activation functions
- Other regression algorithms are possible (linear, logistic).

# Regression Model Evaluation

1700 "Spice" High Pass Filters

1500 designated as Training set

200 taken as Test set

188/200 (94%) within 5% error (bandwidth)

21000 final candidate HPFs with fine sweep

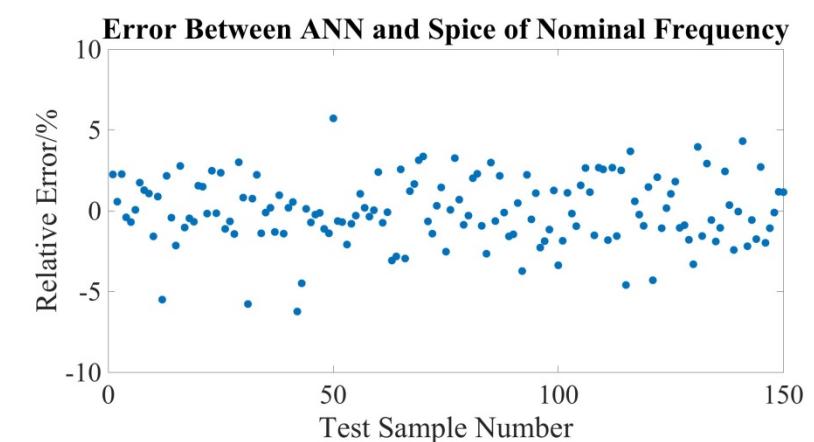
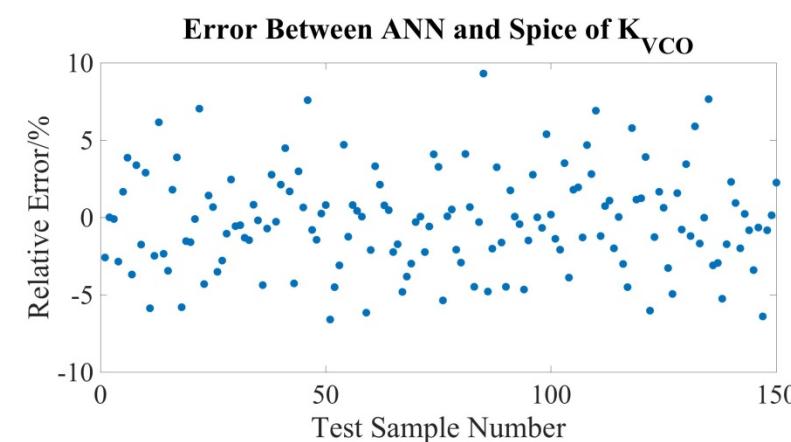
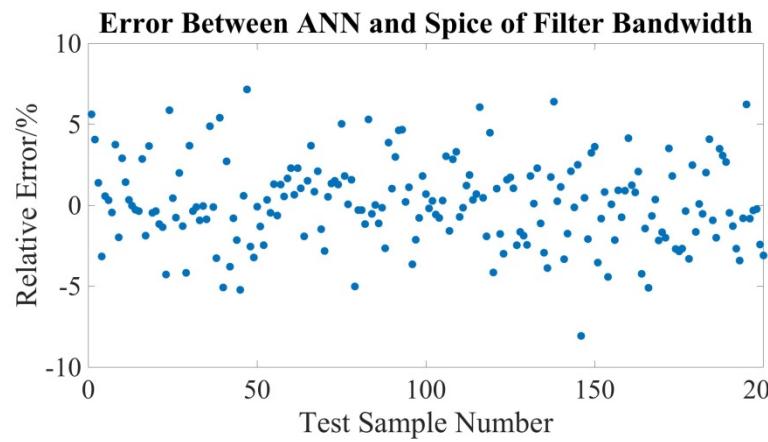
1450 "spice" VCOs

1300 designated as training set.

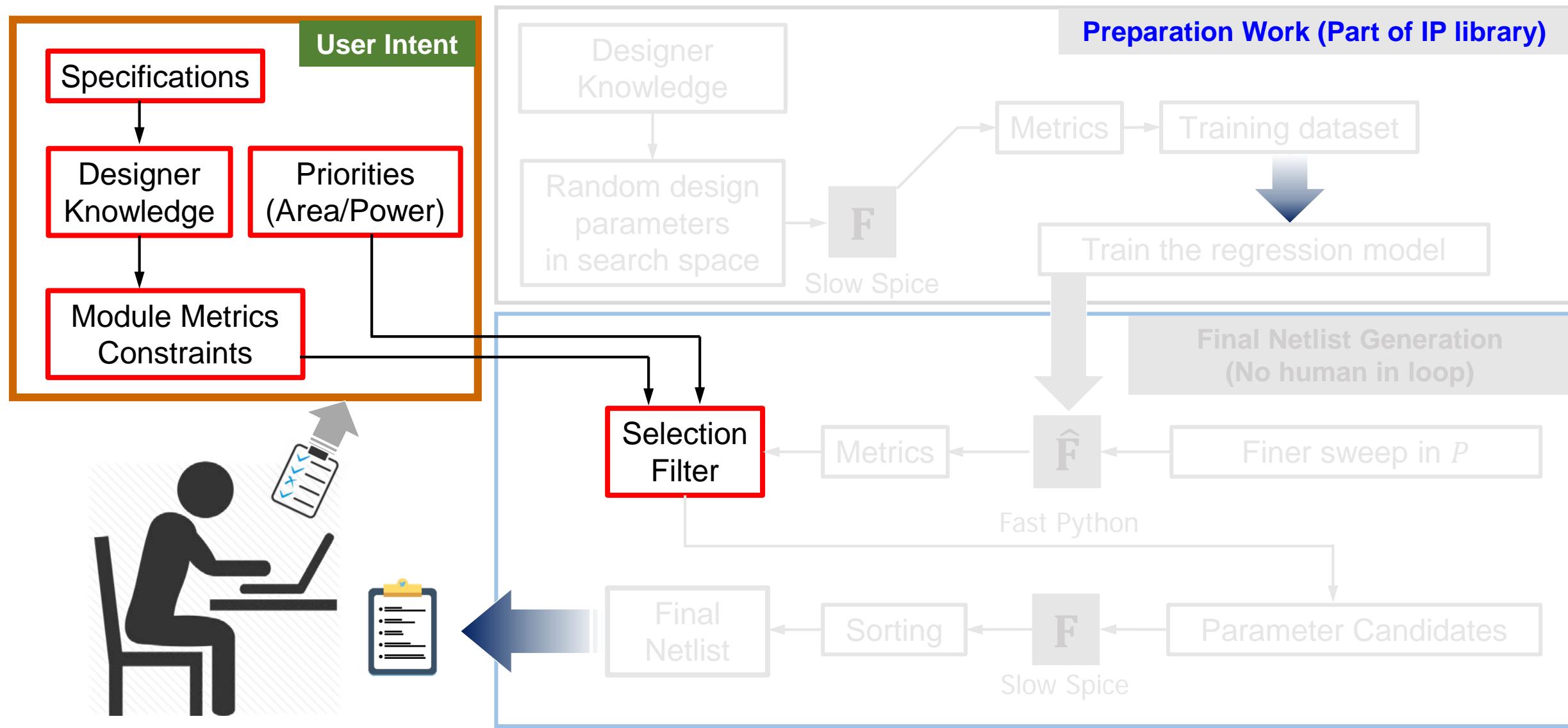
150 taken as test set.

137/150 ( 91.3%) within 5% error ( $F_{\text{nominal}}$  and  $K_{\text{vco}}$ )

10000 final candidate VCOs with fine sweep



# Specifications to Netlist



## 1. Breakdown features from design equations

1.  $H_{\text{Diff}}(s) = \frac{s\tau}{1+s\tau}$ . // get minimum bandwidth of filter required
2.  $N = \log_2(2AG\tau K_v M)$  // resolution from macro parameters
3.  $\text{SNR}_{\text{jitter}} = -20 \log(\omega\sigma_{tj})$  // SNR limitation due to jitter

## 2. Get relations between module parameters

- Equation 1 gives constraint on HPF b/w
- Equation 2 gives relation between filter and needed  $K_{\text{VCO}}$
- Equation 3 adds constraint on VCO choice based on max permissible jitter

## 3. Based on specs, estimate digital core requirements ( frequency, bits)

# Algorithm to find metrics in VCO-Based ADC

$$\tau = 1/(2\pi f_{in}) = RC$$

get filters with at least this minimum time constant

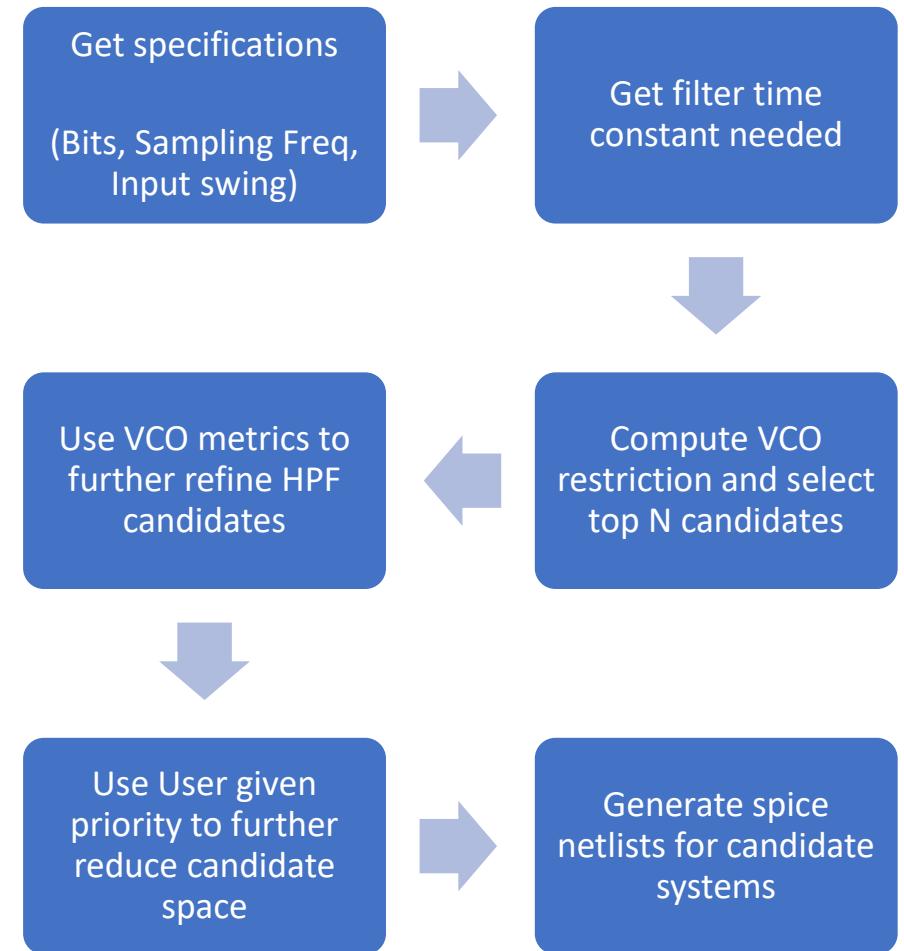
$$A^*K_{vco} \leq 2^n/(G\tau M)$$

Get VCOs which match this criteria

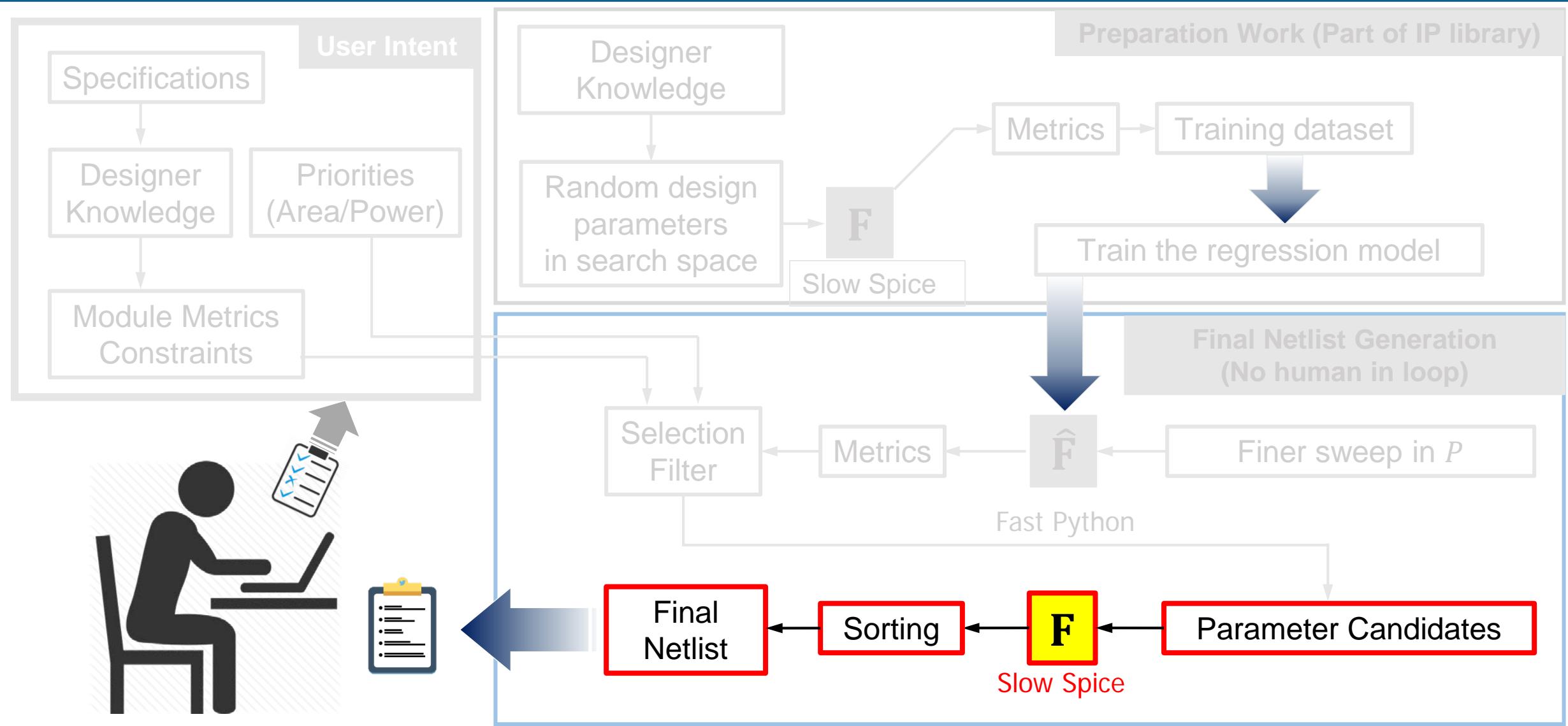
Pick top3 minimum power VCOs ( power priority)

Go back and choose filters with that  $Z_{load}$  added

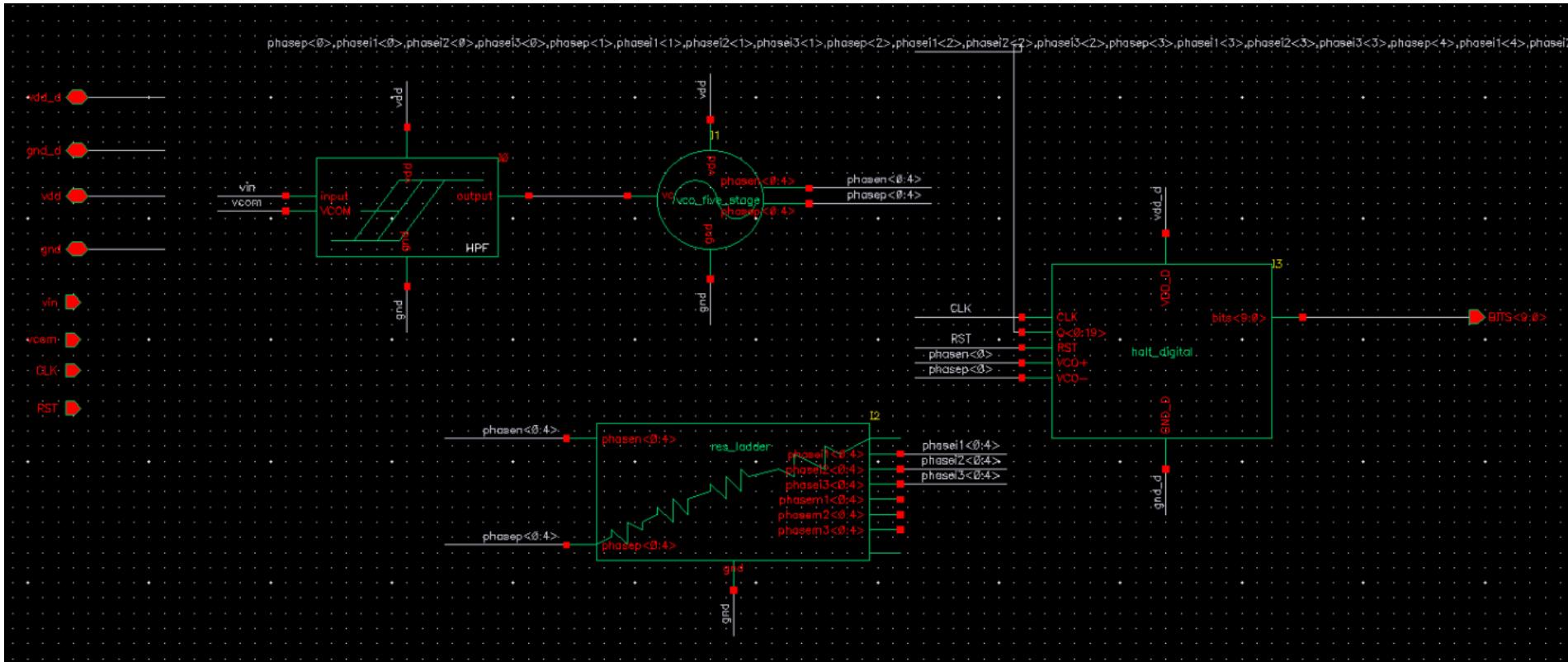
Choose filters with min R and C total area ( area priority)



# Verification



- Validation
  - Apply estimated module parameters into system netlist
  - Performance evaluation benches



# Test Cases in TSMC 65nm

6 bit 1GSPS

5715 filter choices

828 VCO choices

7 bit 800 MSPS

8293 filter choices

797 VCO choices

8 bit 700 MSPS

9834 filter choices

481 VCO choices

Apply Priority based Reduction to get best candidates

Parameter	Value
cap_n_h_fing	132
cap_n_v_fing	132
res_width	1u
res_length	3.5u
N_fing_n	10
N_fing_p	8
Wn	400n
Wp	400n

Parameter	Value
cap_n_h_fing	102
cap_n_v_fing	102
res_width	1u
res_length	2.8u
N_fing_n	7
N_fing_p	5
Wn	400n
Wp	400n

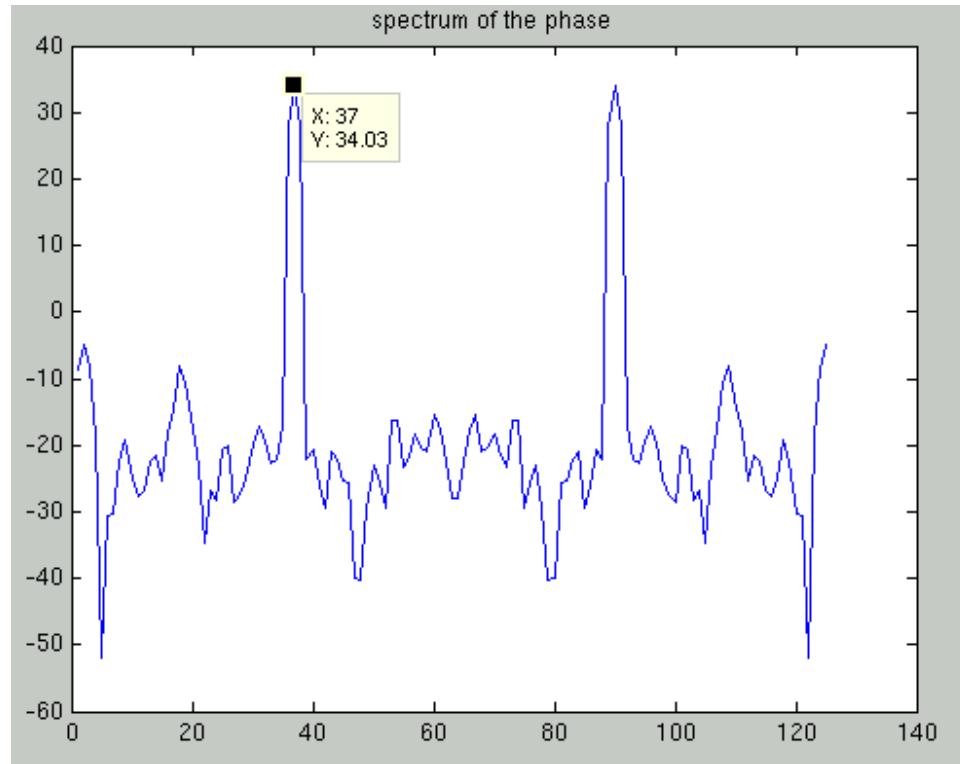
Parameter	Value
cap_n_h_fing	88
cap_n_v_fing	88
res_width	0.5u
res_length	4.2u
N_fing_n	12
N_fing_p	10
Wn	200n
Wp	200n

Digital section parameterized to match resolution of analog section. To prevent overflow errors, 1 extra bit added to MSB.

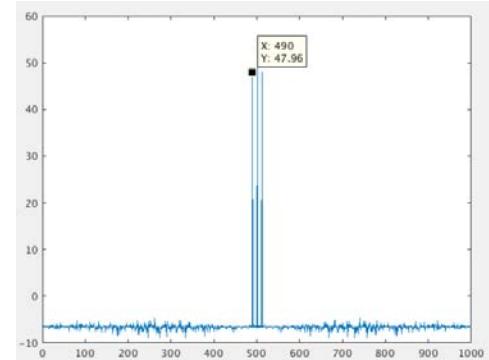
# Test Cases Measurement

Two level measurement.

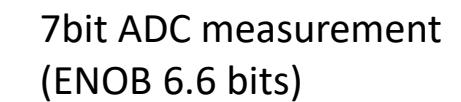
1. Metrics of modules in final netlist
  - $F_{\text{nominal}}$   $K_{\text{vco}}$  Bandwidth <3% error
  
2. High level measurement
  - SNR
  - SFDR ( currently ideal noiseless simulation)



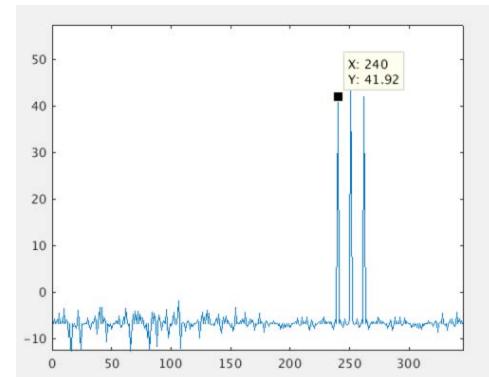
6 bit 1 Gsps ADC measurement (ENOB 6.1 bits)  
(nominal increase due to overflow bit usage )



8bit ADC measurement  
(ENOB 7.8 bits)



7bit ADC measurement  
(ENOB 6.6 bits)



- Demonstration of Automated Mixed Signal design
  - Parameterized library creation
  - Regression model to quickly explore design space
  - Break top level specs to module specifications
  - Schematic level netlist generation for varying specifications
- Next steps
  - Expand library for more variants of modules
    - Different metal layers in passive elements
    - Different VCO topology and number of stages



## Automated AMS IP Generator for CMOS Technologies

---

AMS Design Example: Digital Phase Locked Loop (DPLL) in TSMC 65nm CMOS Technology

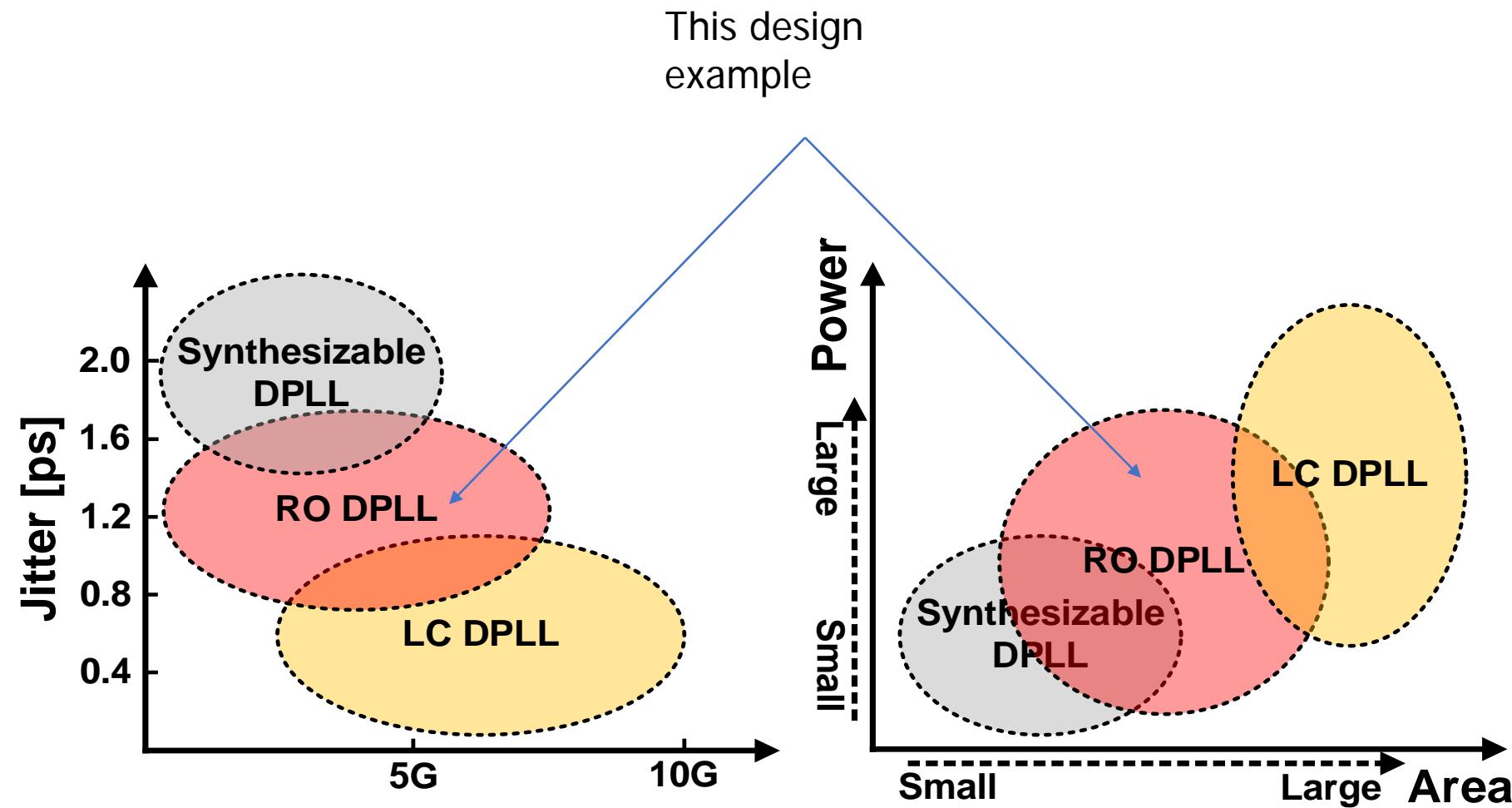
Qiaochu Zhang  
University of Southern California  
Sept. 2018



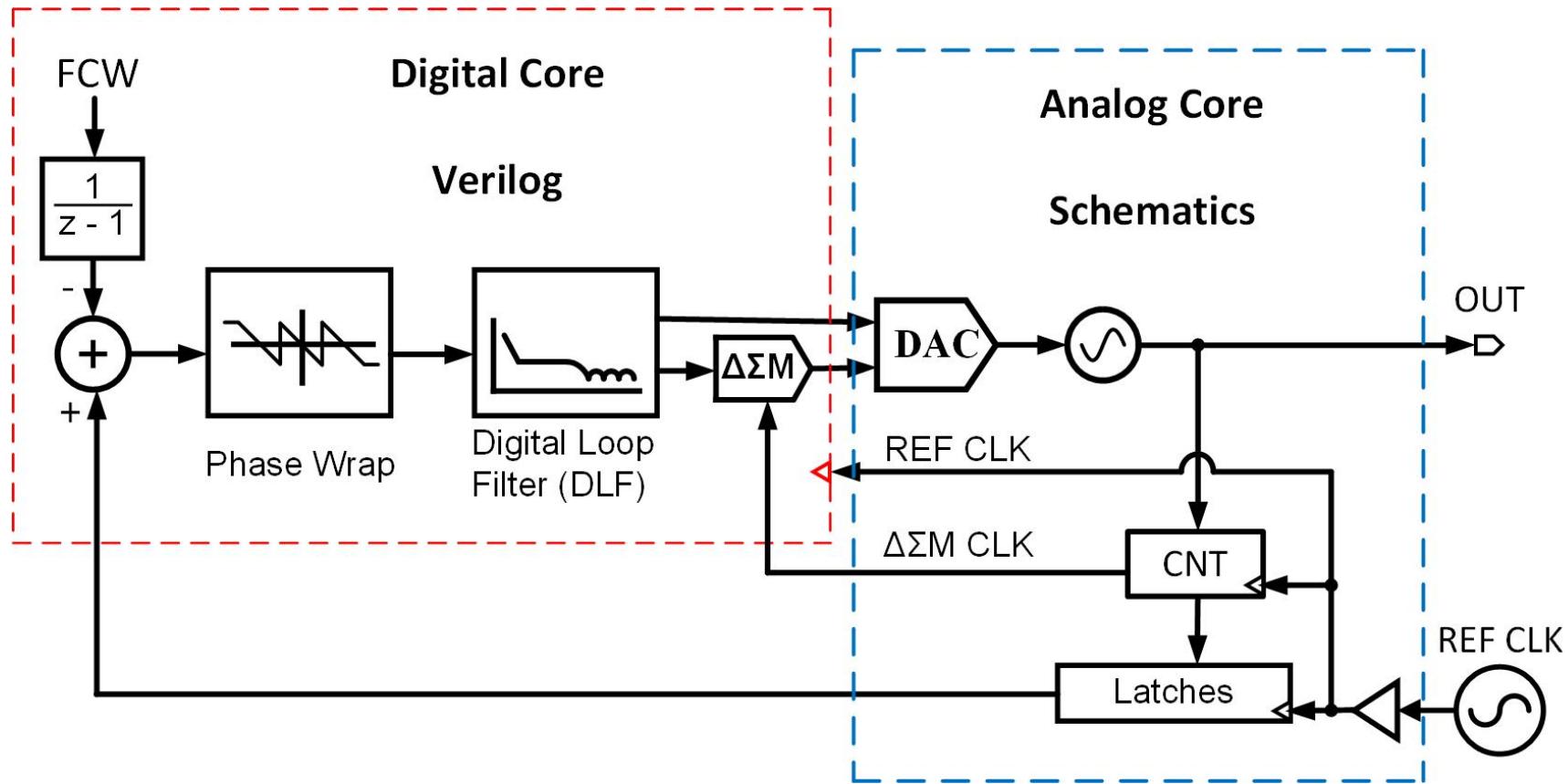
# Outline

- Introduction
- DPLL Architecture and Module break down
- System level consideration
- Module characterization
- Design search from specification
- Verification

# Choice of PLL Architectures



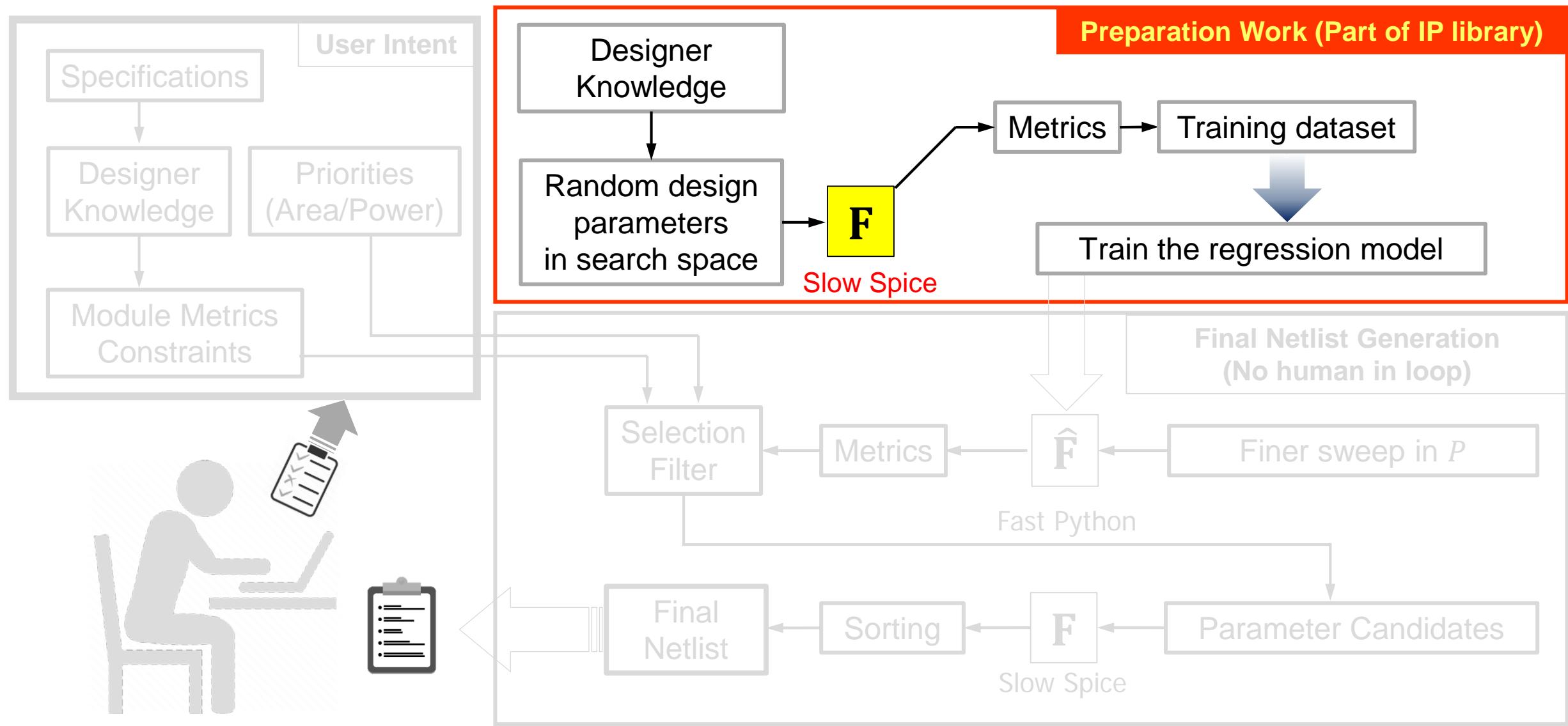
## Architecture of DPLL and Module Breakdown



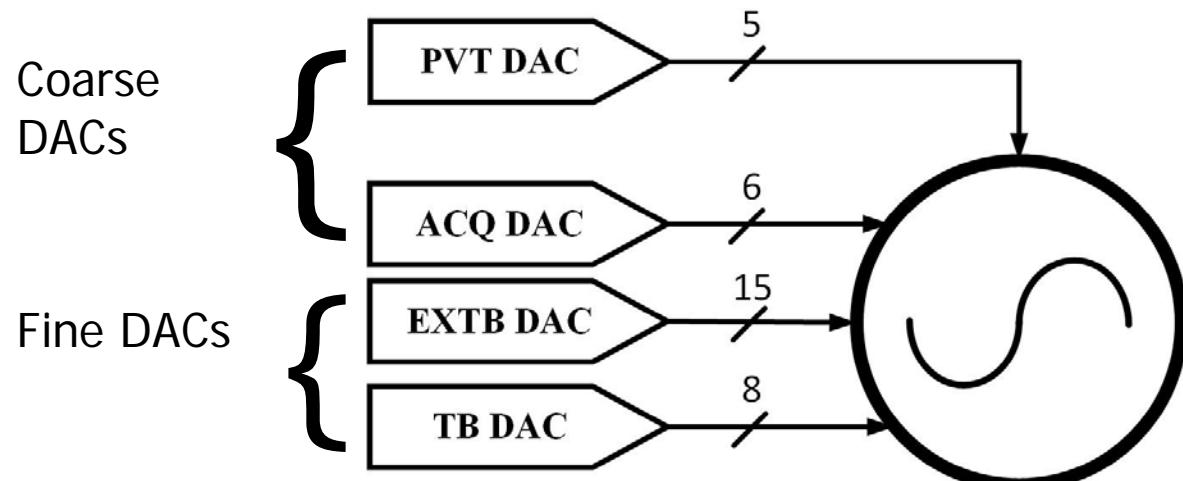
- Digital modules (modelled by RTL)
- Analog modules
  - DCO (VCO+Current DAC)
  - Buffer
  - Counter

- Specifications: Reference clock frequency, PLL operating frequency
- More specifications can be added

# Machine-Learning Based AMS Circuit Generator



# Module Level: DCO Design Metrics

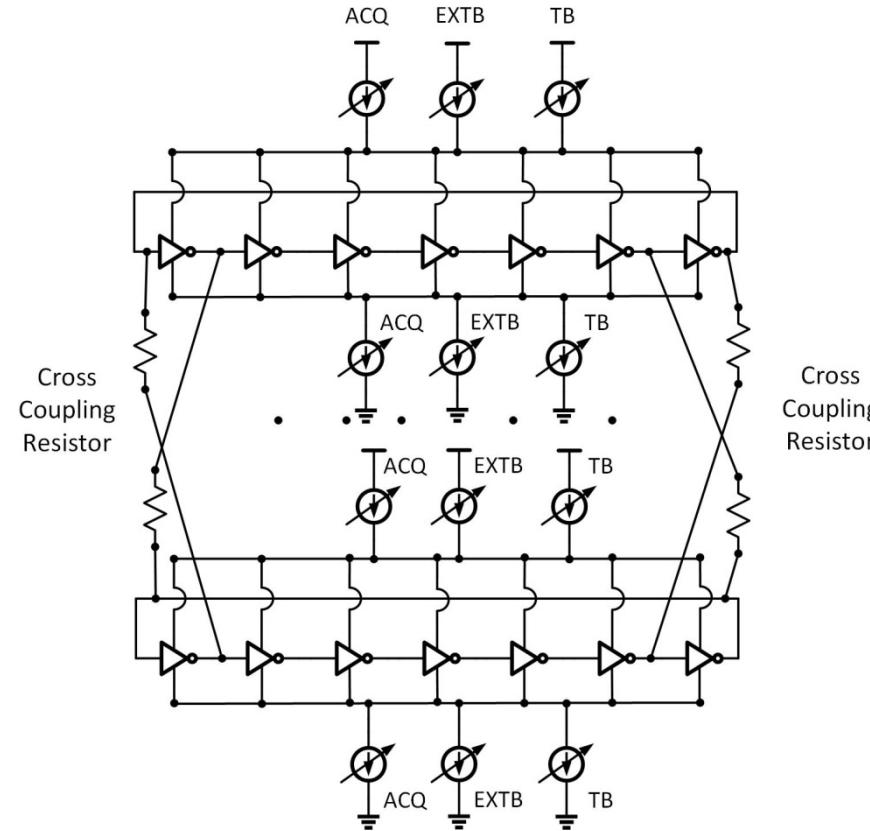


Design Metrics	
Metrics	Symbols
Max Frequency*	fmax
Min Frequency**	fmin
PVT DAC Resolution	pvt_reso
ACQ DAC Resolution	acq_reso
EXTB DAC Resolution	extb_reso
TB DAC Resolution	tb_reso
Max Voltage	vmax
Min Voltage	vmin
ACQ DAC INL	acq_inl

\* Max frequency is defined as the oscillating frequency when PVT control word is 11111 while all other control words are zero.

\*\*Min frequency is defined as the oscillating frequency when PVT control word is 00001 while all other control words are zero.

# Module Level: DCO Design Parameters



Design Parameters	
Parameters	Symbols
# of inverter stages	7
# of fingers of nmos	n_fing
# of fingers of pmos	p_fing
Width of ACQ	ACQ_w
Width of EXTB	EXTB_w
Width of TB	TB_w
Capacitive load	cap_load
Cross Coupling Resistor	cp_res
Bias current	i_bias

## Circuit Knowledge

- $n\_fing = p\_fing$  (nmos and pmos have almost the same strength)
- $EXTB\_w = TB\_w$  (we design EXTB DAC and TB DAC to be the same, from known good design)
- Bias current for DAC is  $250\mu A \sim 300\mu A$  (from known good design)

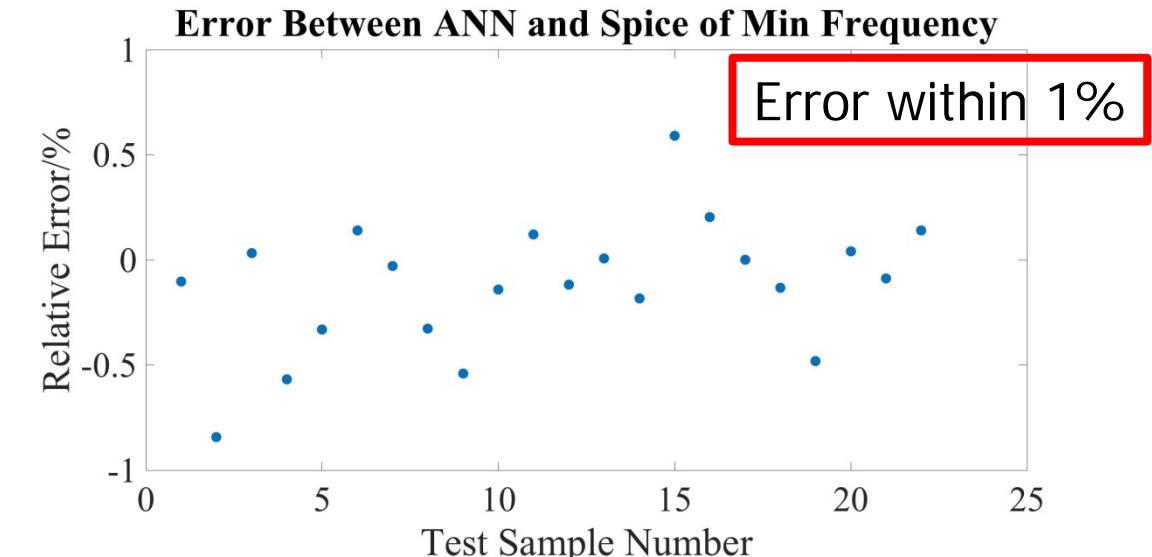
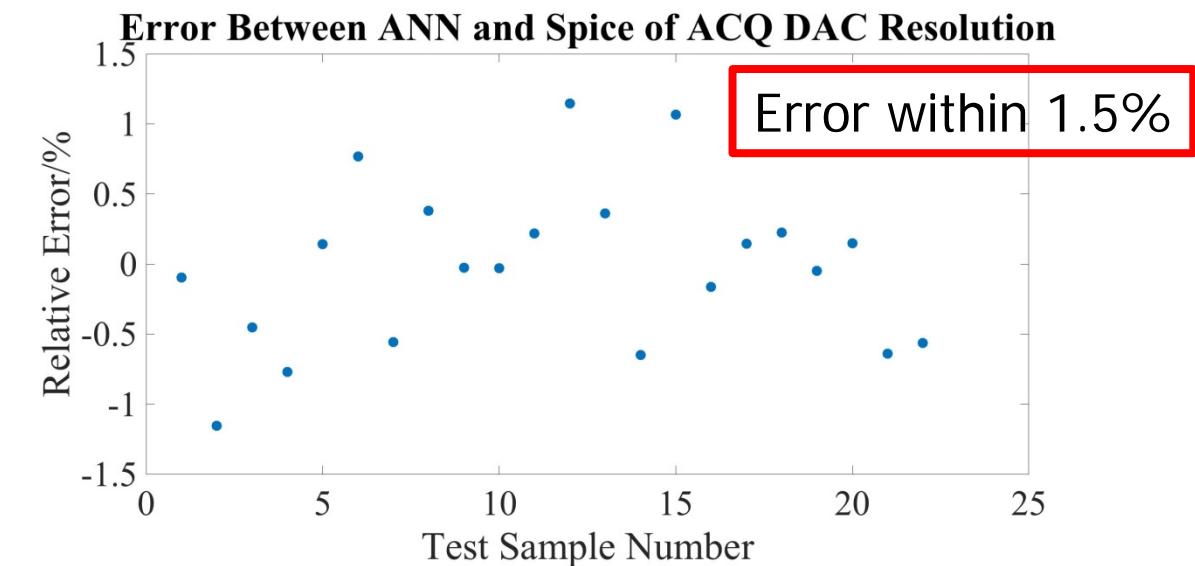
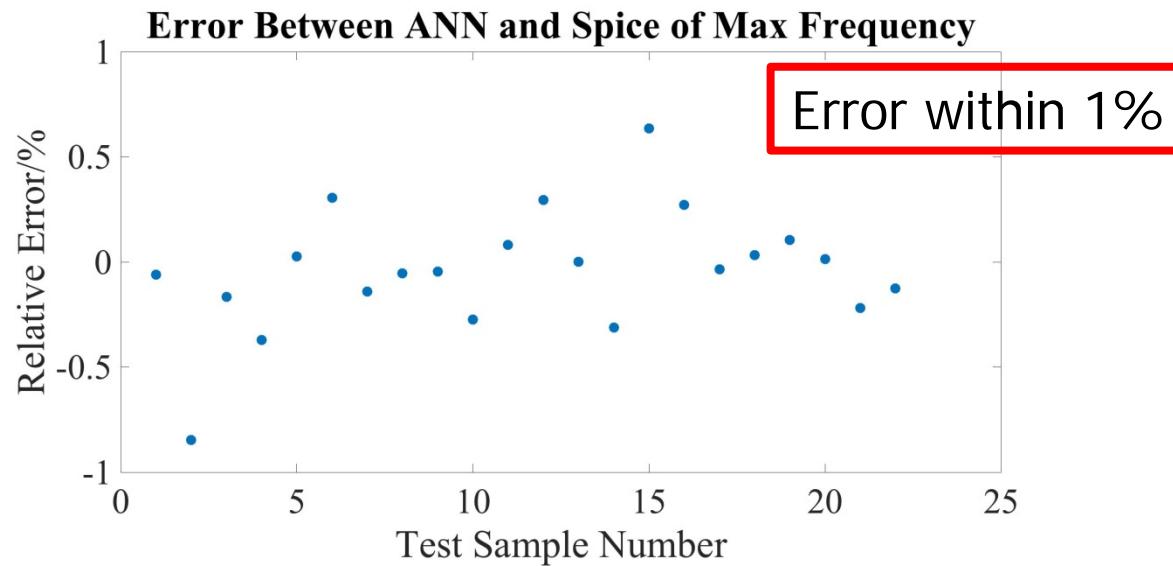
# Module Level: DCO Design Parameters and Metrics

Design Parameters		
Parameters	Symbols	Search Range
# of fingers of pmos	p_fing	[5,10], int
Width of ACQ	ACQ_w	[900nm,1000nm], 10nm discrete step
Width of EXTB	EXTB_w	[250nm,350nm], 10nm discrete step
Capacitive load	cap_load	[25fF,35fF], real
Cross Coupling Resistor	cp_res	[300ohm,400ohm], real
Bias current	i_bias	[200u,300u], real

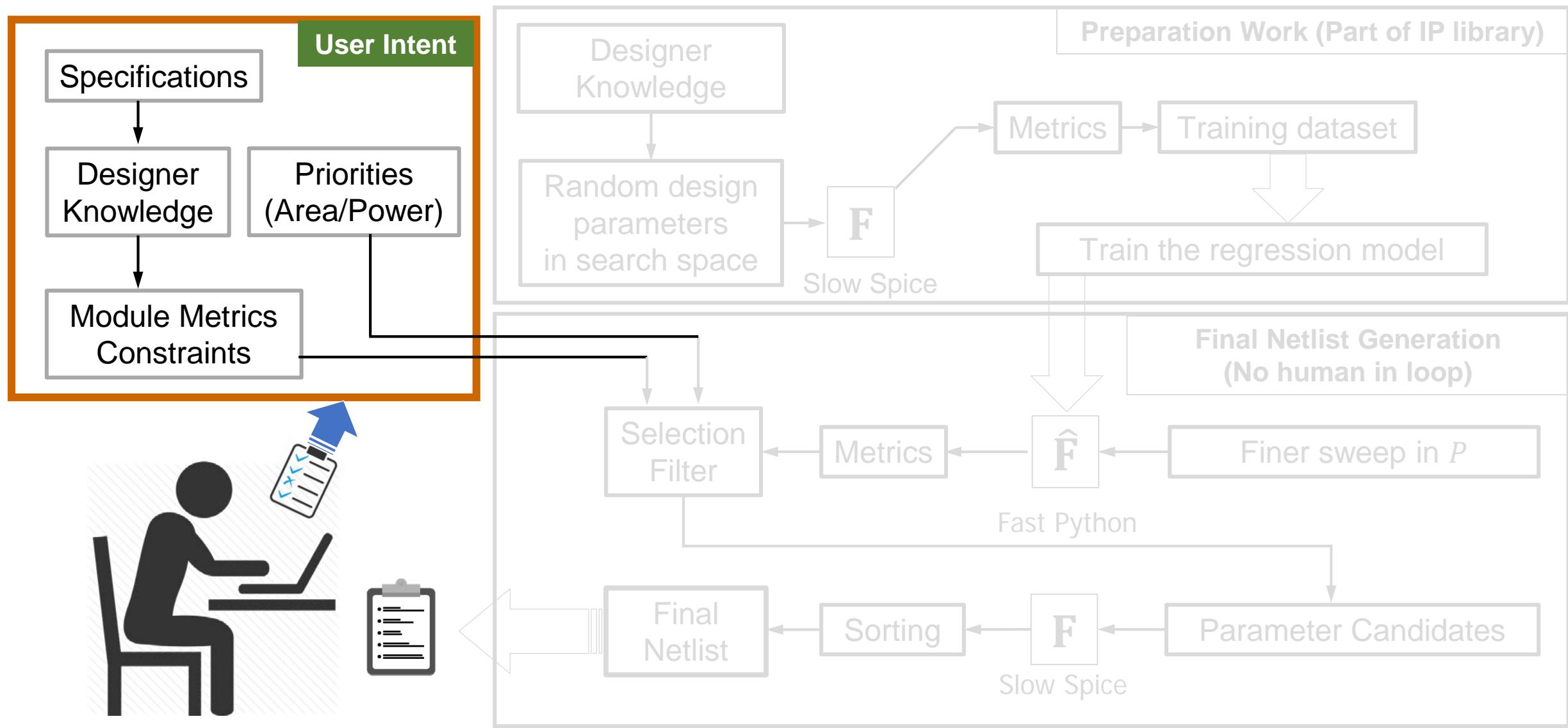
Design Metrics	
Metrics	Symbols
Max Frequency	fmax
Min Frequency	fmin
PVT DAC Resolution	pvt_reso
ACQ DAC Resolution	acq_reso
EXTB DAC Resolution	extb_reso
Max Voltage	vmax
Min Voltage	vmin
ACQ DAC INL	acq_inl

# Formation of Regression Model

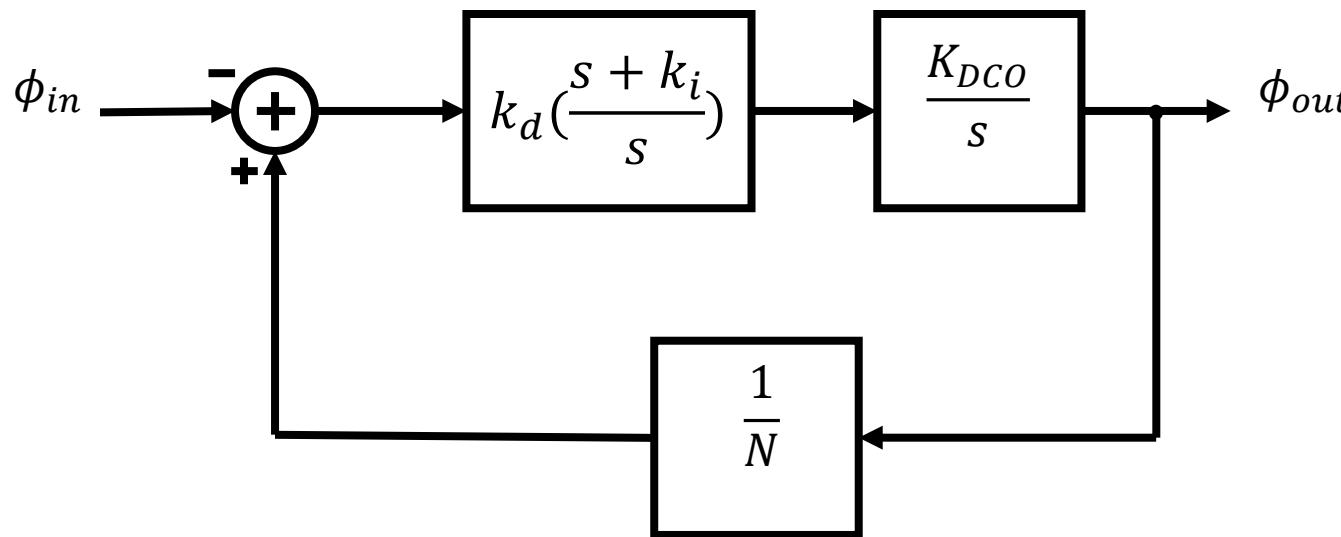
- Training set: 220 randomly generated points (198 for training, 22 for verification)
- Regressor: Artificial Neural Network (ANN)
  - Activation function: elu
  - 6 neurons for input layer, 20 neurons for hidden layer, and 8 neurons for output layer



# Machine-Learning Based AMS Circuit Generator



## Block Level: Loop Analysis



Transfer Function of PLL

$$H(s) = \frac{\phi_{out}}{\phi_{in}} = \frac{k_d K_{DCO} (s + k_i)}{s^2 + \frac{k_d K_{DCO}}{N} s + \frac{k_i k_d K_{DCO}}{N}} = \frac{2N\zeta\omega_n(s + \frac{\omega_n}{2\zeta})}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Damping ratio

$$\zeta = \frac{1}{2} \sqrt{\frac{k_d K_{DCO}}{N k_i}}$$

Natural Frequency

$$\omega_n = \sqrt{\frac{k_d k_i K_{DCO}}{N}}$$

- Continuous-time approximation: PLL bandwidth much smaller than input frequency  $\omega_{-3dB} \ll \omega_{in}$

$$\Rightarrow \frac{k_d k_i K_{DCO}}{N} \ll \frac{1}{4} \omega_{in}^2$$

- Usually we choose damping ratio  $\zeta > 1$  in order to have a relatively short settling time. That is:

$$\frac{k_d K_{DCO}}{N k_i} > 4$$

*derived value*

*known value*

### Design equations

$$\omega_n = 0.1 \times \frac{1}{2} \omega_{in}$$

$$k_i = \frac{\omega_n}{2\zeta}$$

$$K_{DCO} = \frac{N \omega_n^2}{k_d k_i}$$

## DPLL Target Specs

**Design 1**

<b>Design Specifications</b>	
Metrics	Value
Reference clock frequency	40 MHz
Multiplication factor	<b>90</b>
output frequency	<b>3.6 GHz</b>

**Design 2**

<b>Design Specifications</b>	
Metrics	Value
Reference clock frequency	40 MHz
Multiplication factor	<b>95</b>
output frequency	<b>3.8 GHz</b>

**Chosen/Known Variables**

Parameters	Value
$\zeta$	1.5
$k_d$	255

**Chosen/Known Variables**

Parameters	Value
$\zeta$	1.5
$k_d$	255

## Derived Module Parameters

Design 1

Calculated Variables	
Parameters	Value
$\omega_{-3dB}$	$\approx 4 \text{ MHz}$
$\omega_n$	$\approx 2 \text{ MHz}$
$k_i$	$\frac{1}{2^6} F_{ref} = 625000 \text{ Hz}$

Design 2

Calculated Variables	
Parameters	Value
$\omega_{-3dB}$	$\approx 4 \text{ MHz}$
$\omega_n$	$\approx 2 \text{ MHz}$
$k_i$	$\frac{1}{2^6} F_{ref} = 625000 \text{ Hz}$

# Derived DCO Metric Constraints

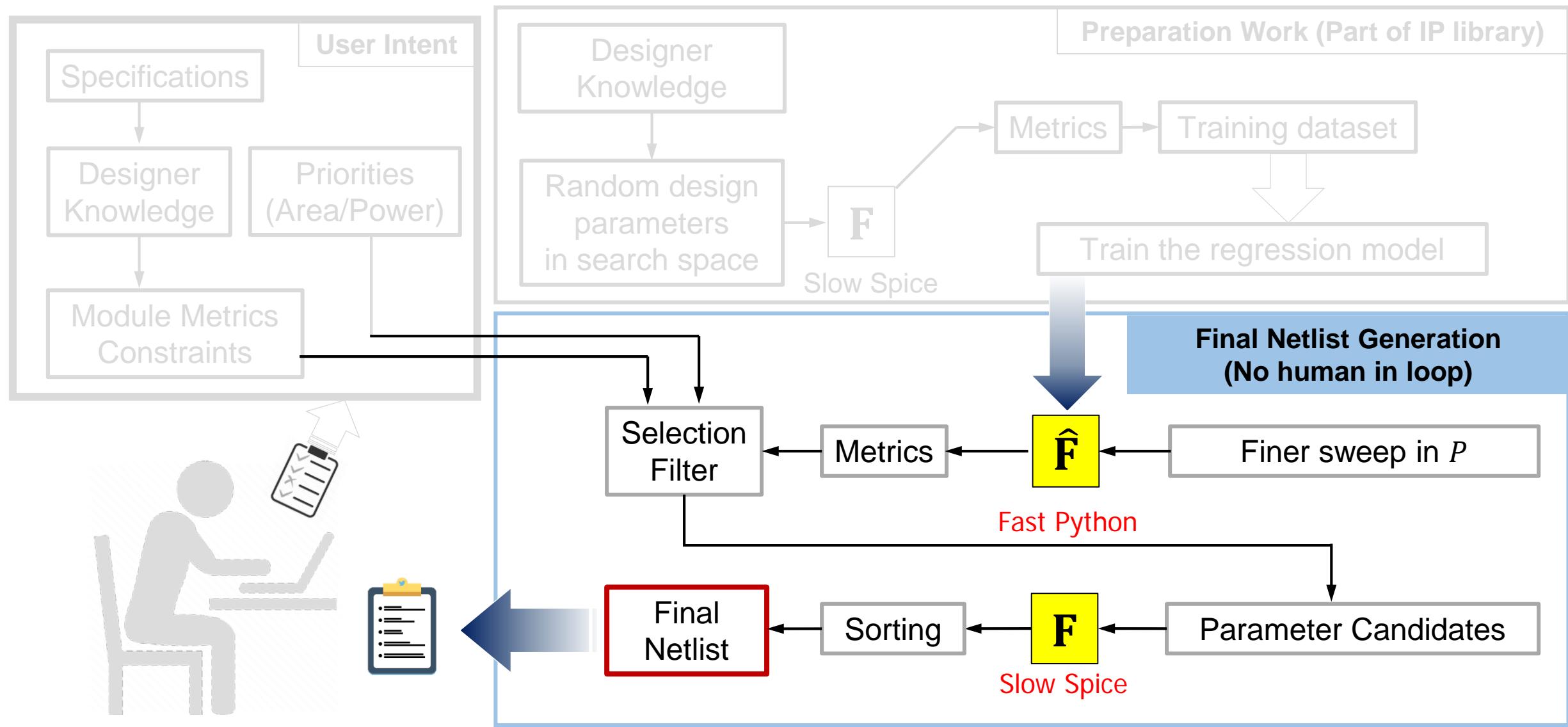
**Design 1**

Design Metrics		
Metrics	Symbols	Requirement
Max Frequency	fmax	> 3.7 GHz
Min Frequency	Fmin	< 3.5 GHz
PVT DAC Resolution	pvt_reso	-----
ACQ DAC Resolution	acq_reso	$\approx 1.9\text{MHz}/\text{code}$
EXTB DAC Resolution	extb_reso	-----
Max Voltage	vmax	$> V_{IH} = 0.5\text{ V}$
Min Voltage	vmin	$< V_{IL} = 0.3\text{ V}$
ACQ DAC INL	acq_inl	-----

**Design 2**

Design Metrics		
Metrics	Symbols	Requirement
Max Frequency	fmax	> 3.9 GHz
Min Frequency	Fmin	< 3.7 GHz
PVT DAC Resolution	pvt_reso	-----
ACQ DAC Resolution	acq_reso	$\approx 2\text{ MHz}/\text{code}$
EXTB DAC Resolution	extb_reso	-----
Max Voltage	vmax	$> V_{IH} = 0.5\text{ V}$
Min Voltage	vmin	$< V_{IL} = 0.3\text{ V}$
ACQ DAC INL	acq_inl	-----

# Machine-Learning Based AMS Circuit Generator



**Design 1**

- $f_{max} > 3.7 \text{ GHz}$
- $f_{min} < 3.5 \text{ GHz}$
- $acq\_reso \in [1.8 \text{ MHz}, 2.0 \text{ MHz}]$
- $extb\_reso \in [490 \text{ kHz}, 510 \text{ kHz}]$
- $\max\{f_{max} - f_{min}\}$  (max tuning range)

**Design 2**

- $f_{max} > 3.9 \text{ GHz}$
- $f_{min} < 3.6 \text{ GHz}$
- $acq\_reso \in [1.9 \text{ MHz}, 2.1 \text{ MHz}]$
- $extb\_reso \in [490 \text{ kHz}, 510 \text{ kHz}]$
- $\max\{f_{max} - f_{min}\}$  (max tuning range)

## Final Design Parameters and Metrics

## Design 1

Design Parameters	
Parameters	Value
p_fing	8
ACQ_w	900 nm
EXTB_w	340 nm
cap_load	32 fF
cp_res	320 ohm
i_bias	200 uA

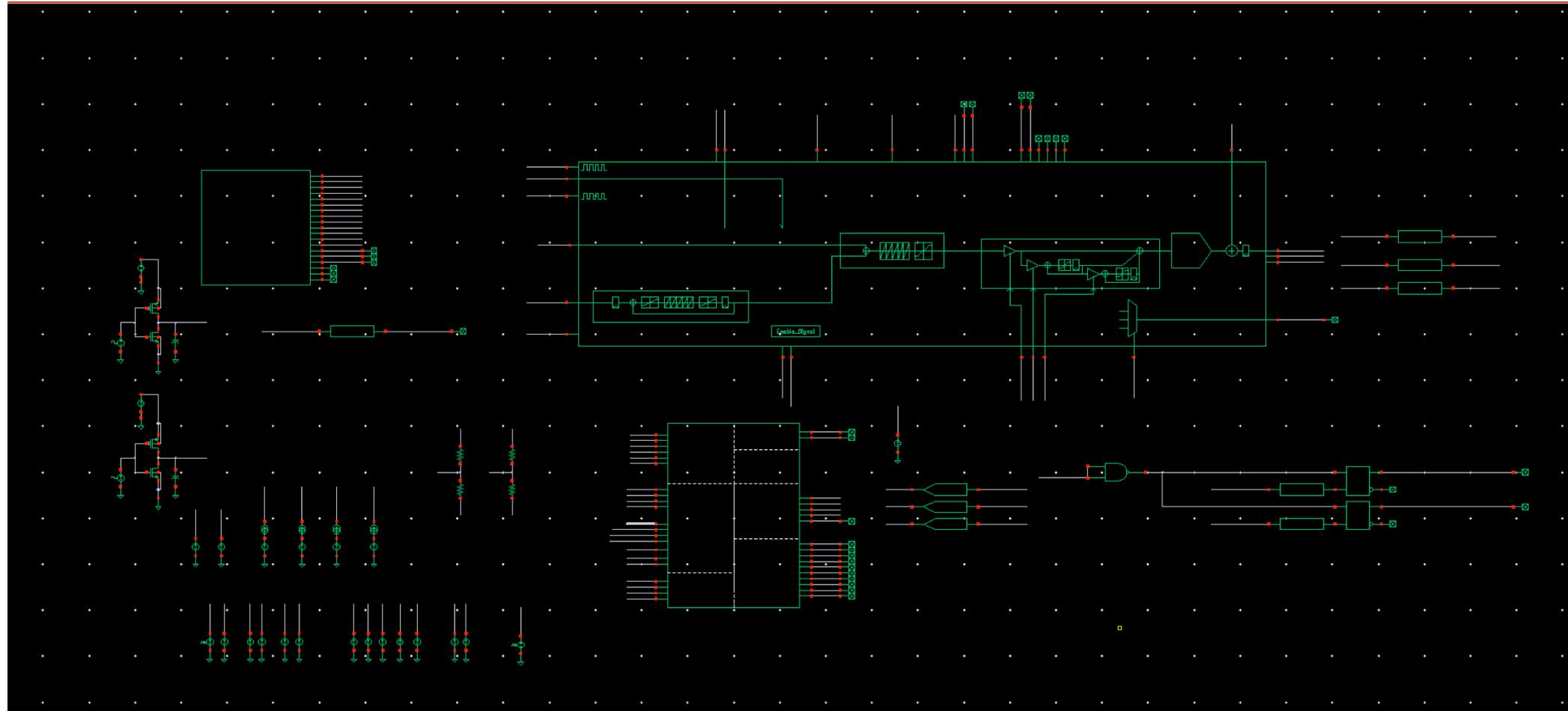
Design Metrics			
Metrics	ANN	SPICE	Error
fmax	3.931 GHz	3.949 GHz	0.5%
fmin	1.623 GHz	1.615 GHz	0.5%
pvt_reso	74.64 MHz	75.17 MHz	0.7%
acq_reso	1.9999 MHz	2.0123 MHz	0.6%
extb_reso	493.0 kHz	483.0 kHz	2%
vmax	0.78 V	0.78 V	0%
vmin	0.13 V	0.13 V	0%
acq_inl	0.03 LSB	0.01 LSB	---

# Final Design Parameters and Metrics

## Design 2

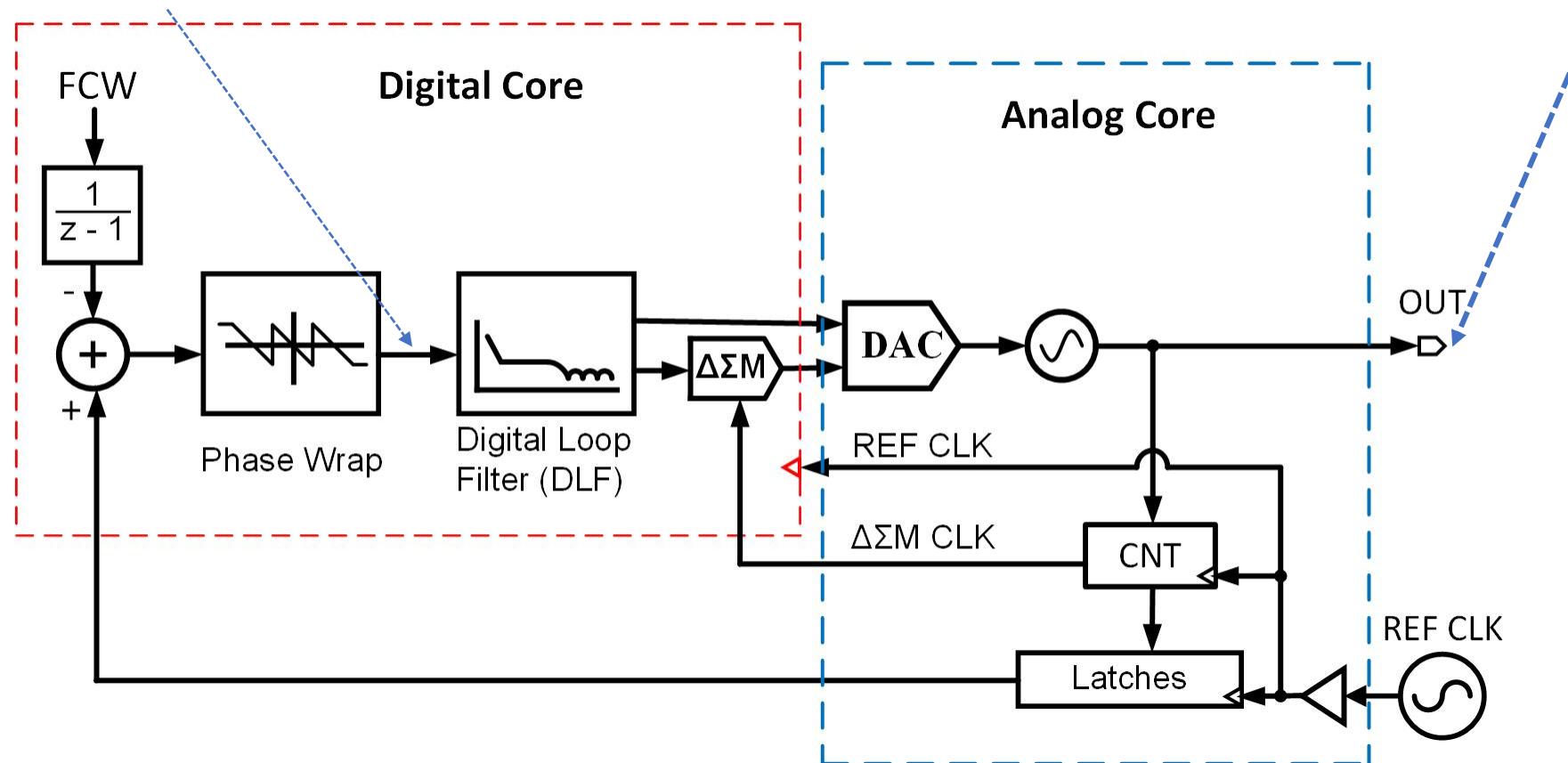
Design Parameters	
Parameters	Value
p_fing	9
ACQ_w	900 nm
EXTB_w	320 nm
cap_load	31 fF
cp_res	330 ohm
i_bias	200 uA

Design Metrics			
Metrics	ANN	SPICE	Error
fmax	4.043 GHz	4.036 GHz	0.2%
fmin	1.615 GHz	1.600 GHz	0.9%
pvt_reso	78.28 MHz	78.53 MHz	0.3%
acq_reso	2.0997 MHz	2.0841 MHz	0.6%
extb_reso	493.7 kHz	496.0 kHz	0.5%
vmax	0.77 V	0.77 V	0%
vmin	0.14 V	0.14 V	0%
acq_inl	0.03 LSB	0.02 LSB	---

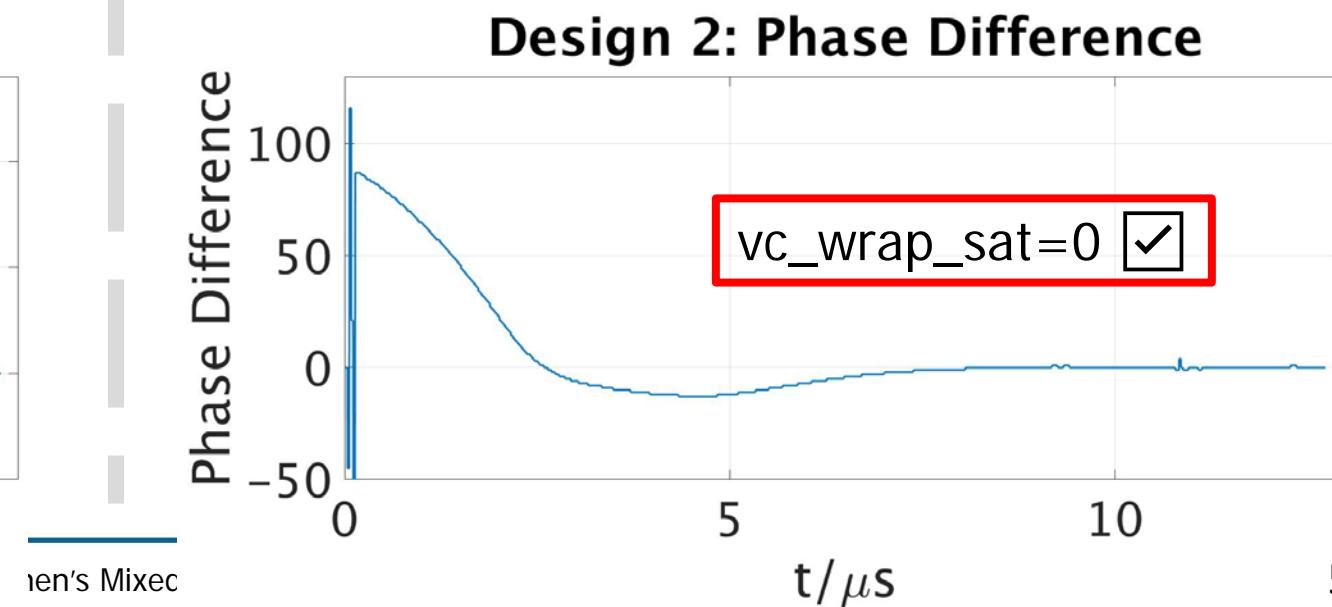
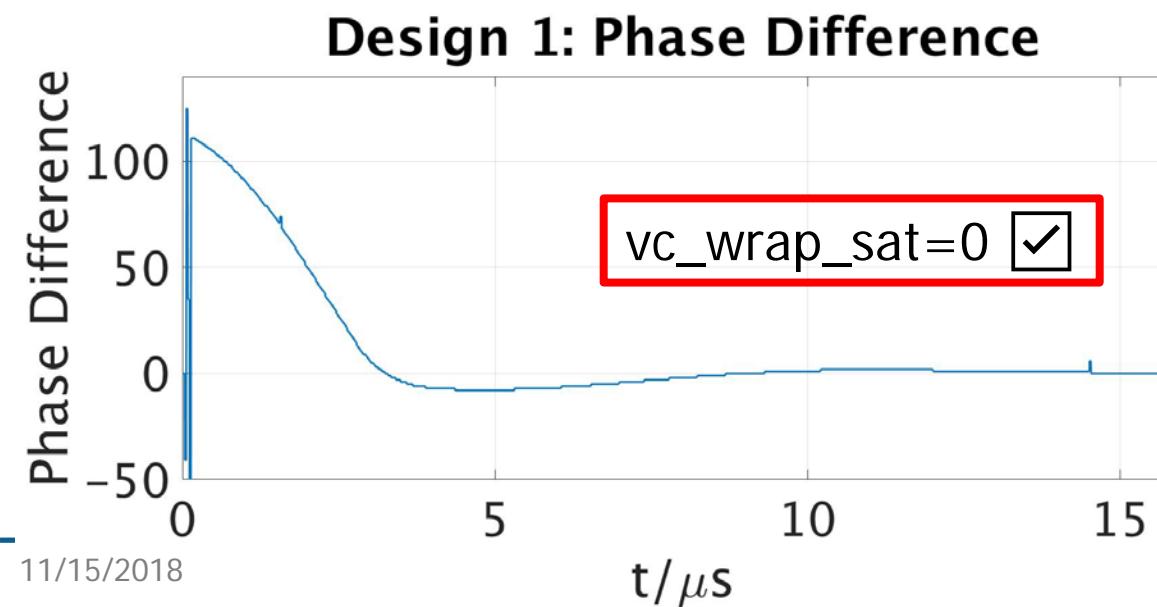
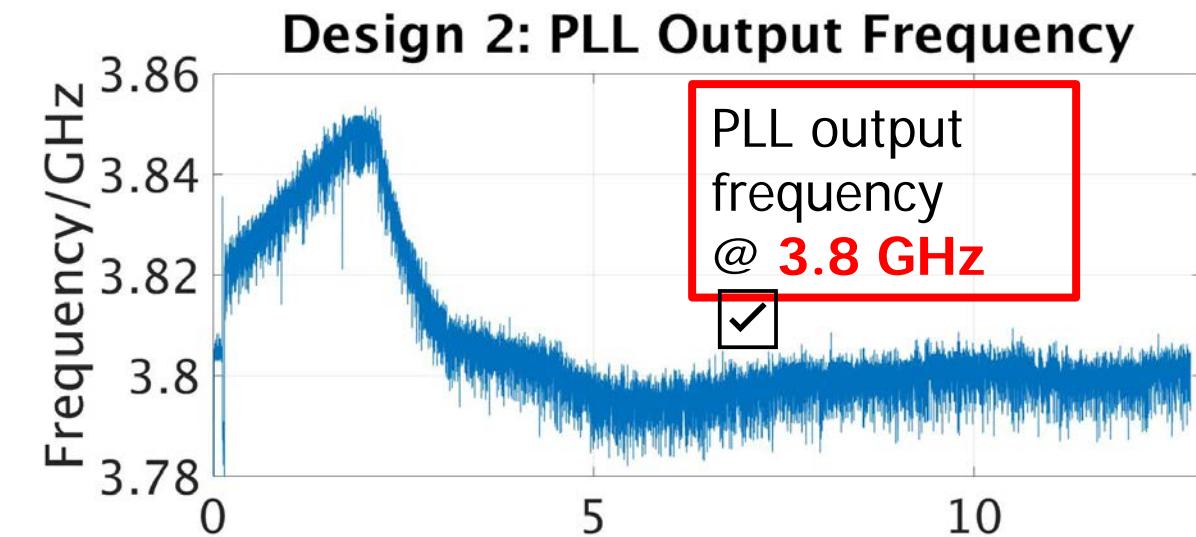
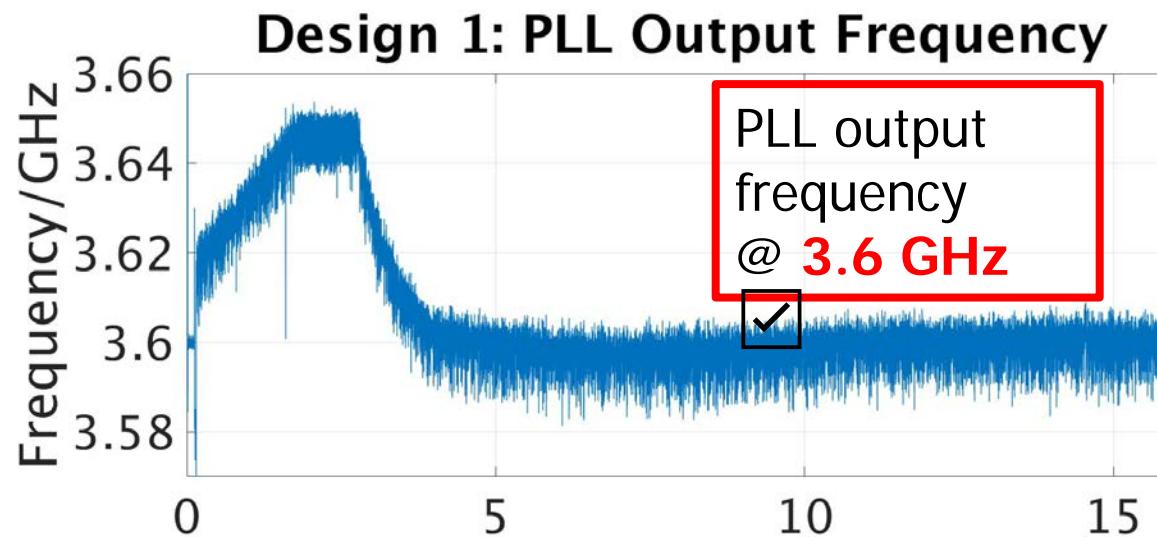


Is the phase difference zero?  
(vc\_wrap\_sat=0)

Is the output frequency correct?  
(pll\_freq)



## Final Verification Simulation Results

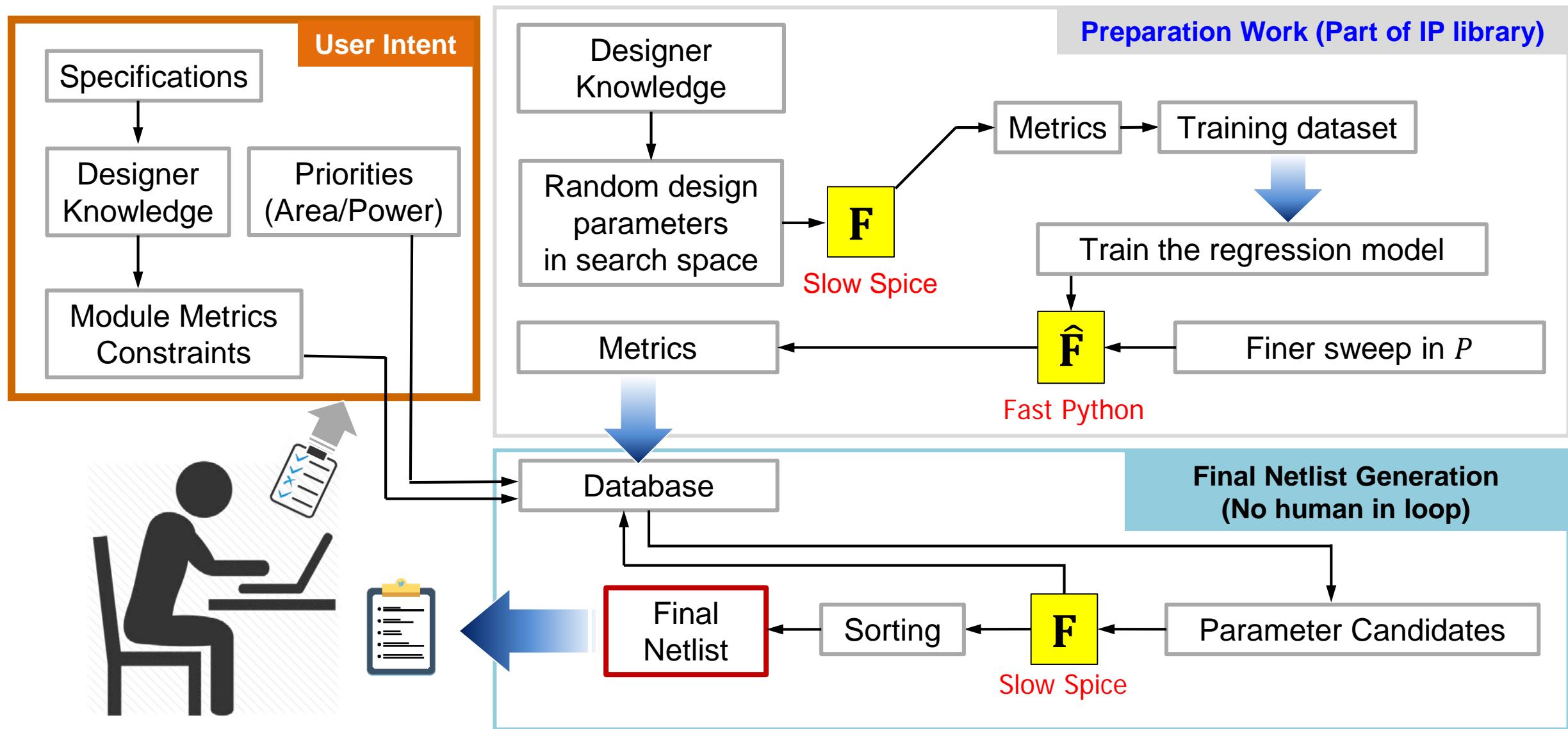


## Design Time Comparison

- Fine sweep in parameter space:  $5 \times 10^5$  points
  - Time consumption for conventional SPICE sweep:  
$$5 \times 10^5 \text{ points} \times 2 \text{ min/point} = 10^6 \text{ min} \approx 2 \text{ yrs}$$
  - Time consumption for the ML-based design flow:  
$$10 \text{ s (fine sweep in python)} + 2 \text{ points} \times 2 \text{ min/point} = 4 \text{ min}$$
  - Time consumption to train ML regression models:  
$$\begin{aligned} & 220 \text{ points} \times 2 \text{ min/point (generating training set)} \\ & + 0.5 \text{ hr (regression model training)} = 7.9 \text{ hrs} \end{aligned}$$
- ~100X time saving (after the ML model is built in this example)**

## Conclusion and Future Work

- Achieved milestones:
  - Built basic parameterized library for DPLL
  - Proved that the proposed ML-based design flow can lead to a successful design for basic DPLL architecture
  - Initial test benches are built for final verification
  - Significantly reduced design time after ML model is established
- Future works:
  - Add more specs, such as phase noise, jitter, etc.
  - Wider search space for larger spec range
  - Build more modes, such as Fractional-N modes, etc.



# Problem Statement

- Given
  - Parameter values  $P = \{p_1, p_2, p_3, \dots\}$
  - Alternative structures  $S = \{s_1, s_2, s_3, \dots\}$
  - Performance metrics  $M = \{m_1, m_2, m_3, \dots\}$
- Each KGD is a point in a space defined as a mapping of values from  $S \times P$  to values of  $M$
- Start with a few KGDs, i.e., specific points, and systematically explore a set of structures and parameter values to obtain a set of designs that efficiently meet wide range of design specifications



# Key Challenge

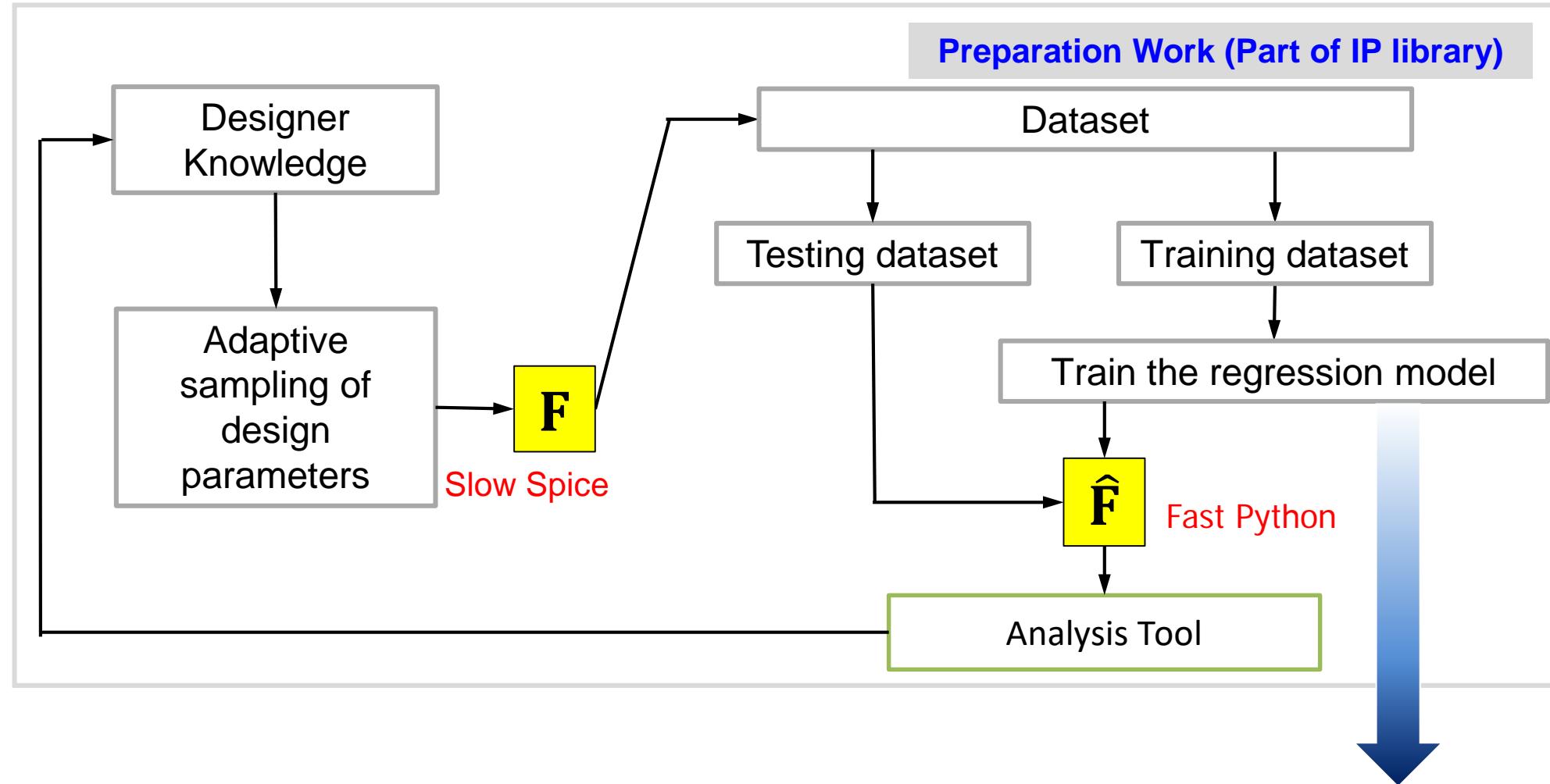
- Impossible to exhaustively enumerate all possible combinations of values of parameters
  - Too many parameters: sizes of many transistors, control voltages
  - Several parameters take continuous values
- Straightforward approaches for sampling may miss regions of uniquely desirable or undesirable performance



# Approach

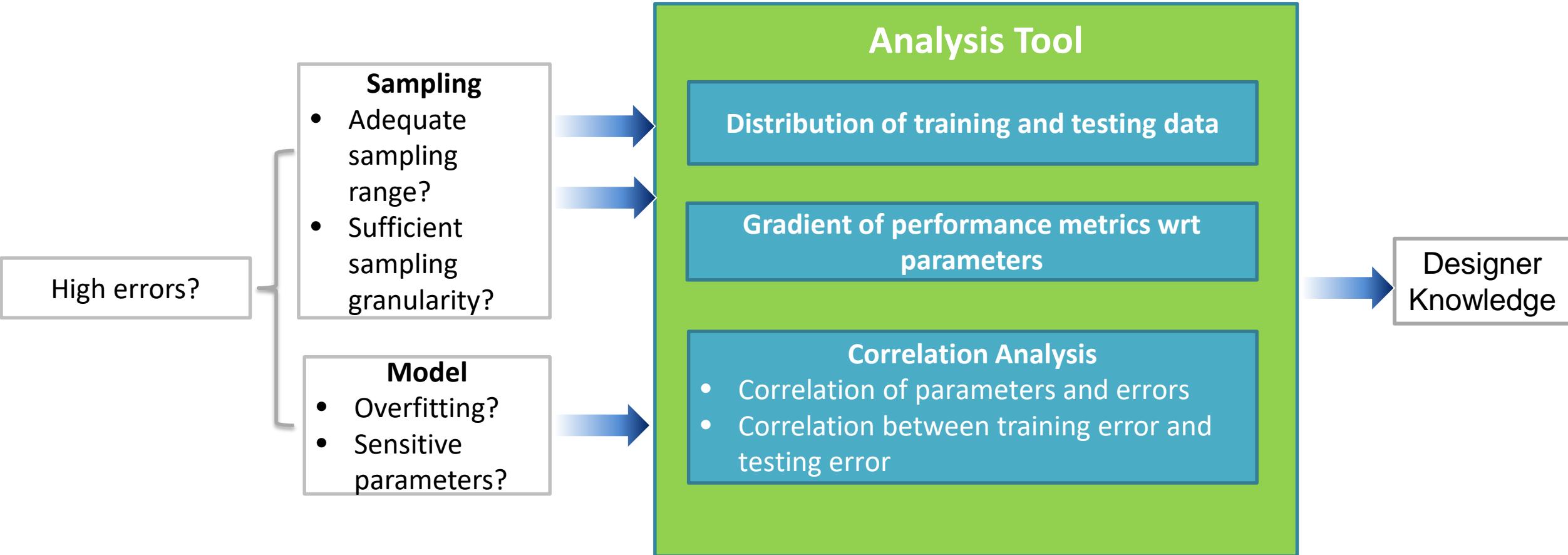
1. Capture all designer knowledge about  $S \times P \rightarrow M$  mapping from KGD
  - a) Captured: Ranges of values near KGDs; behavior near ends of ranges
  - b) Not captured yet: Regions of extreme behavior (values, slopes, ...); correlations/independence across subsets of parameters; ...
2. Methods and data-driven tools for human-expert-in-loop to derive additional knowledge by analyzing simulation results
3. Adaptive sampling of the parameter space that harnesses designer knowledge and simulations
4. Extend the data structures and methods to harness new types of knowledge derived

# More on Preparation



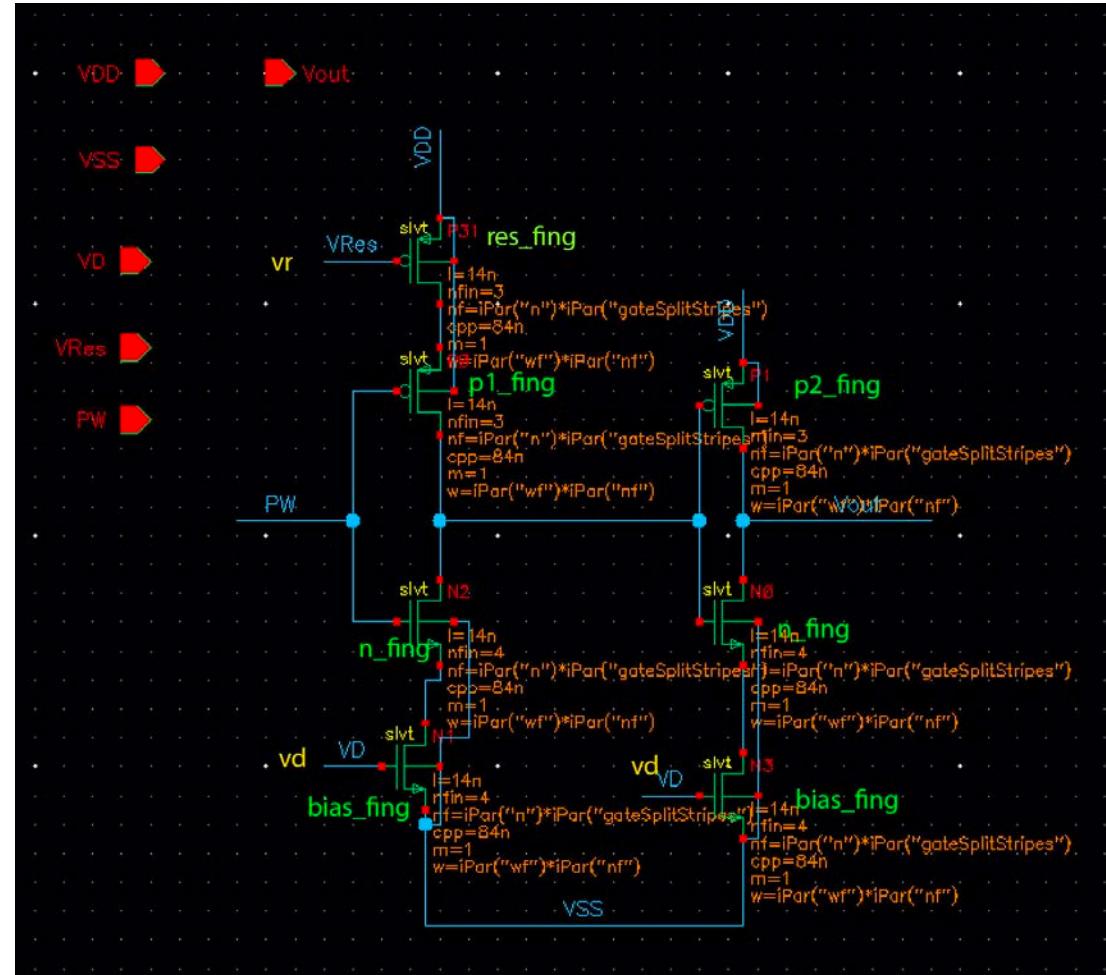
- Dataset

	<b>Input parameters</b> $P = \{p_1, p_2, p_3, \dots\}$	<b>Performance metrics</b> $M = \{m_1, m_2, m_3, \dots\}$	<b>Estimated performance metrics</b> $\hat{M} = \{\hat{m}_1, \hat{m}_2, \hat{m}_3, \dots\}$	<b>Error</b> $ M - \hat{M} $
Training dataset	Input to $F$	Generated by $F$ to train $\hat{F}$	Input $P$ into $\hat{F}$ after training	Training Error
Testing dataset	Input to $F$ and $\hat{F}$	Generated by $F$	Generated by $\hat{F}$	Testing Error



## Example 1: Pulse Shrinking Cell

Parameters
bias_fing
n_fing
p1_fing
p2_fing
res_fing
vd
vr



Metrics
Delay
Power
Resolution



# Correlation Analysis

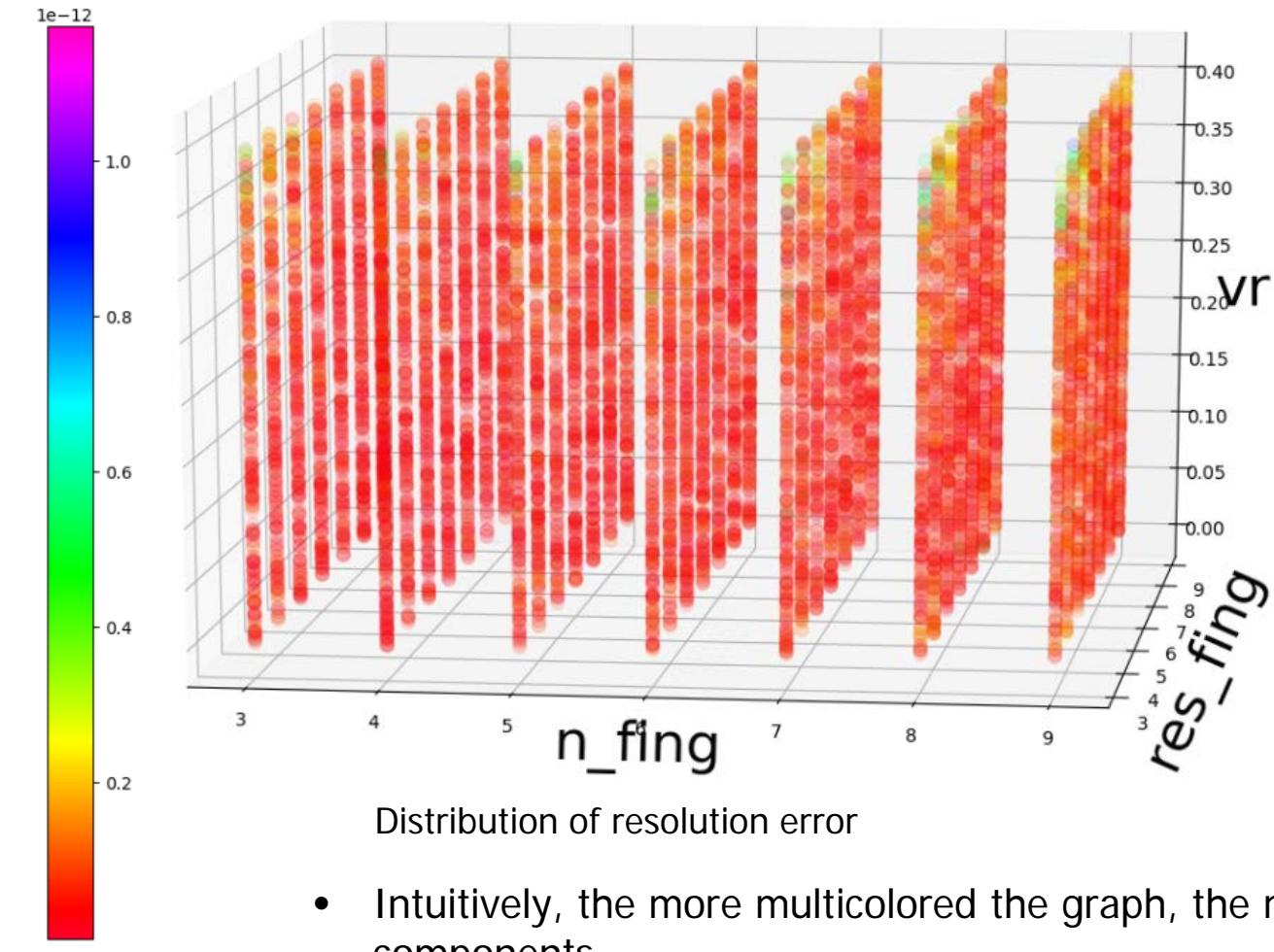
	bias_fing	n_fing	p1_fing	p2_fing	res_fing	vd	vr	Res error
bias_fing	1	0.015305	0.003947	-0.0108	-0.00194	0.003654	-0.00037	-0.01388
n_fing	0.015305	1	0.007958	0.001669	0.010479	-0.01148	-0.00162	0.151967
p1_fing	0.003947	0.007958	1	-0.00016	-0.00253	5.60E-05	0.008811	0.012607
p2_fing	-0.0108	0.001669	-0.00016	1	0.012534	-0.00584	-0.00533	0.082552
res_fing	-0.00194	0.010479	-0.00253	0.012534	1	-0.00316	0.010571	-0.16388
vd	0.003654	-0.01148	5.60E-05	-0.00584	-0.00316	1	0.008516	-0.13306
vr	-0.00037	-0.00162	0.008811	-0.00533	0.010571	0.008516	1	0.271166
error	-0.01388	0.151967	0.012607	0.082552	-0.16388	-0.13306	0.271166	1

Sample Covariance Matrix for Training Dataset

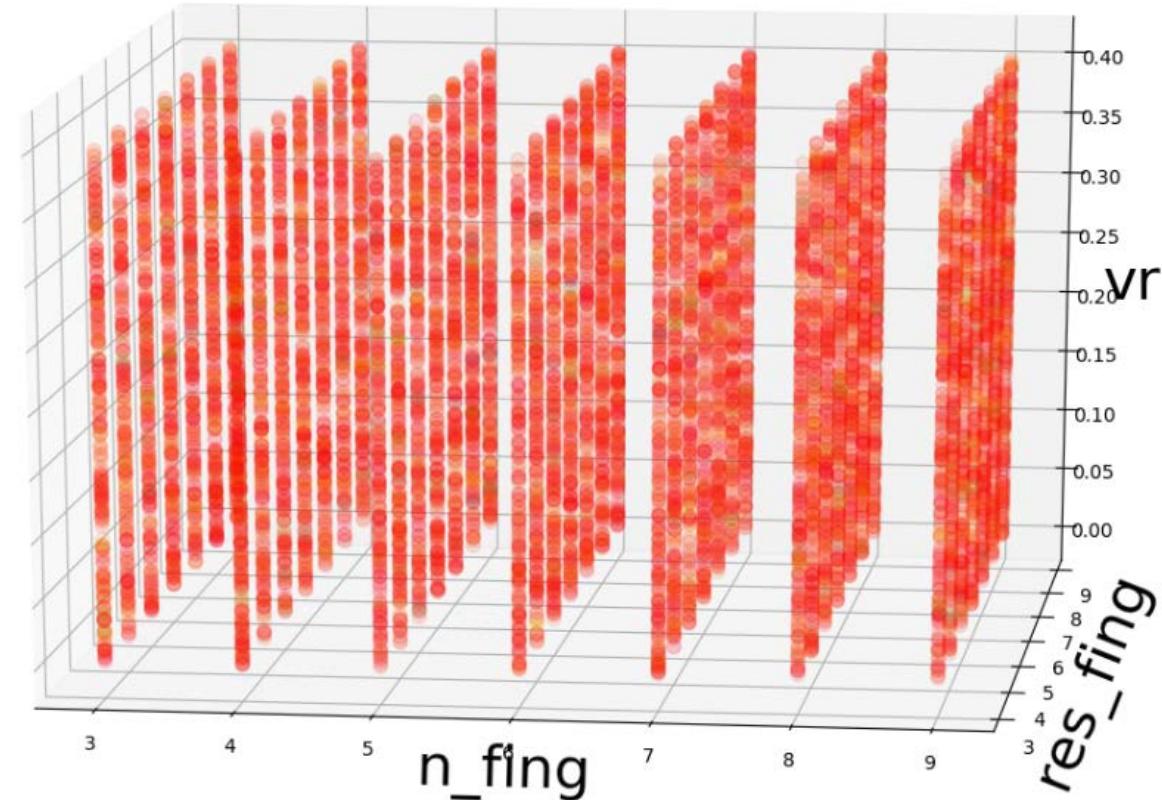
	bias_fing	n_fing	p1_fing	p2_fing	res_fing	vd	vr	Res error
bias_fing	1	-0.02135	-0.02353	0.020246	-0.01765	-0.00295	-0.00985	-0.02866
n_fing	-0.02135	1	0.033914	-0.02126	0.025804	-0.00241	0.001977	0.144278
p1_fing	-0.02353	0.033914	1	-0.03308	-0.00847	0.011496	0.006695	0.015276
p2_fing	0.020246	-0.02126	-0.03308	1	-0.0005	-0.00961	-0.02839	0.066326
res_fing	-0.01765	0.025804	-0.00847	-0.0005	1	0.002479	0.02136	-0.1529
vd	-0.00295	-0.00241	0.011496	-0.00961	0.002479	1	0.034609	-0.12732
vr	-0.00985	0.001977	0.006695	-0.02839	0.02136	0.034609	1	0.273941
error	-0.02866	0.144278	0.015276	0.066326	-0.1529	-0.12732	0.273941	1

Sample Covariance Matrix for Testing Dataset

- We analyze resolution error separately due to high prediction error.
- No overfitting problem since the covariance matrices of training data and testing data are similar.
- High resolution error for higher values of n\_fing and vr, and lower value of res\_fing.



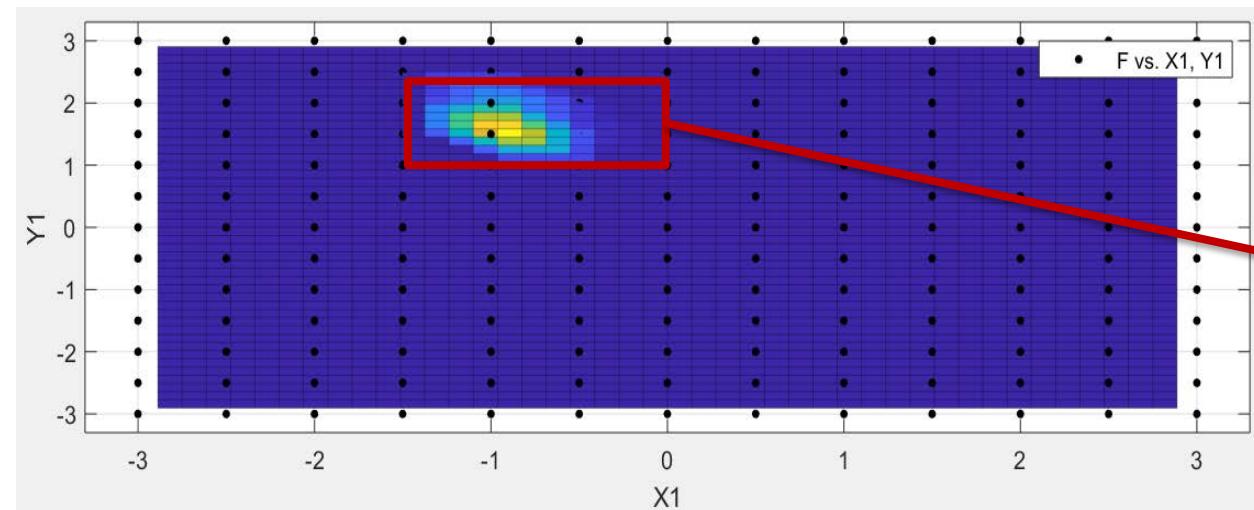
Distribution of resolution error



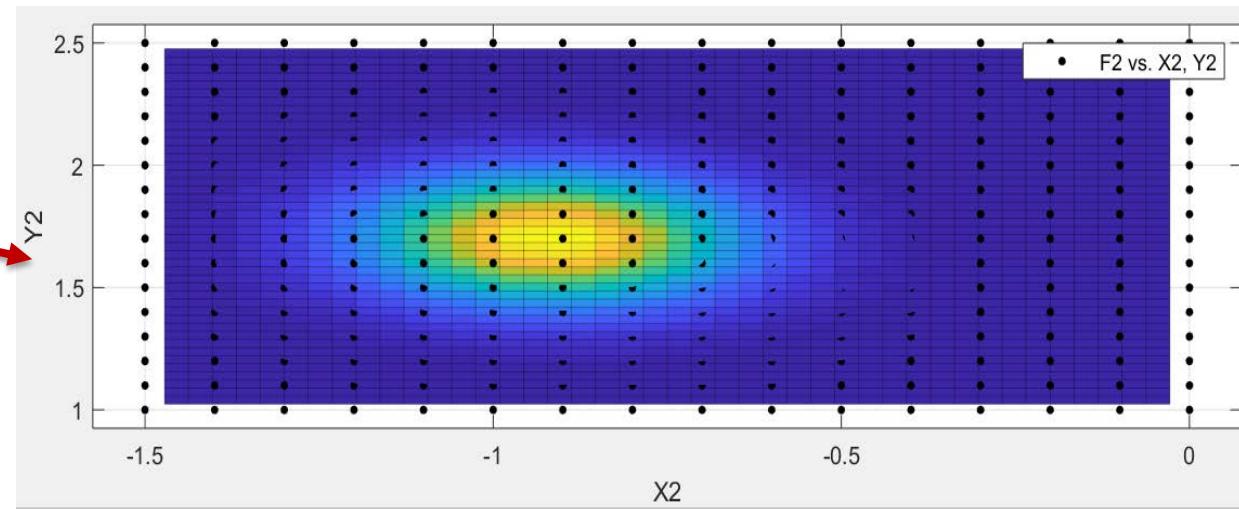
Distribution of resolution error after randomly shuffled

- Intuitively, the more multicolored the graph, the more information about the error in the selected components.
- We are developing a quantitative measure for this information.
- We select the components that provide maximum information about error for human expert.

Example 2: Adaptive sampling based on gradient to find the desirable region

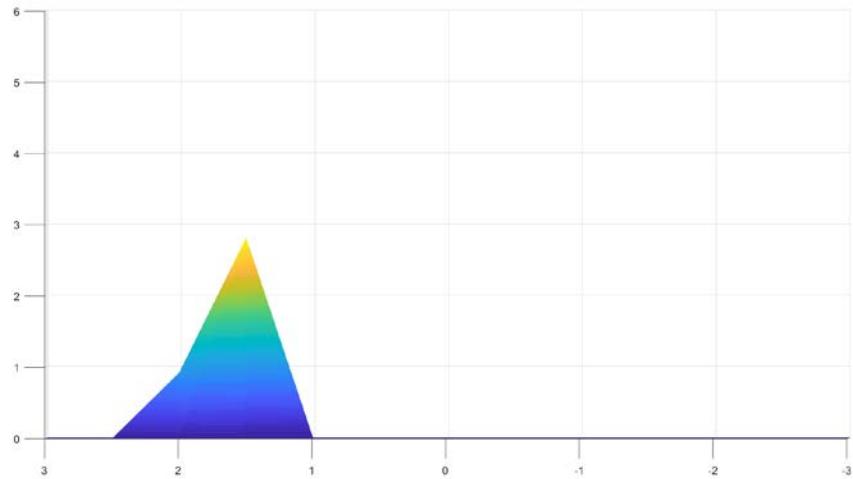


Sampling rate  $r_1$

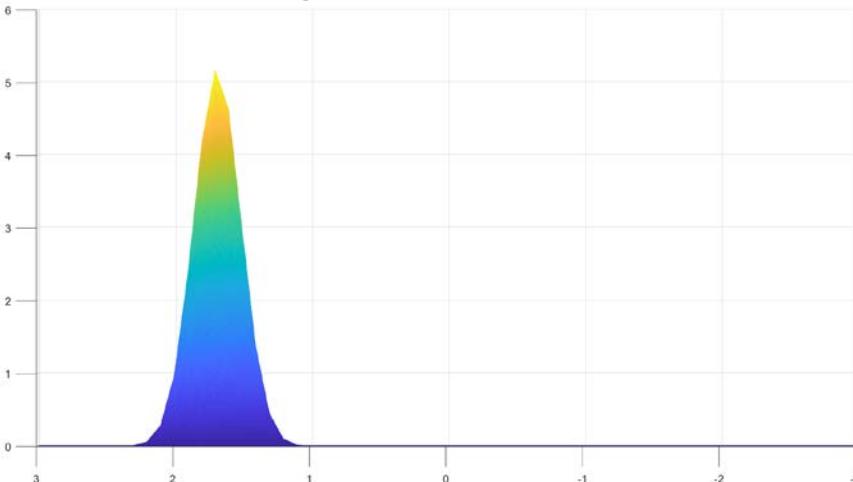


Sampling rate  $r_2$

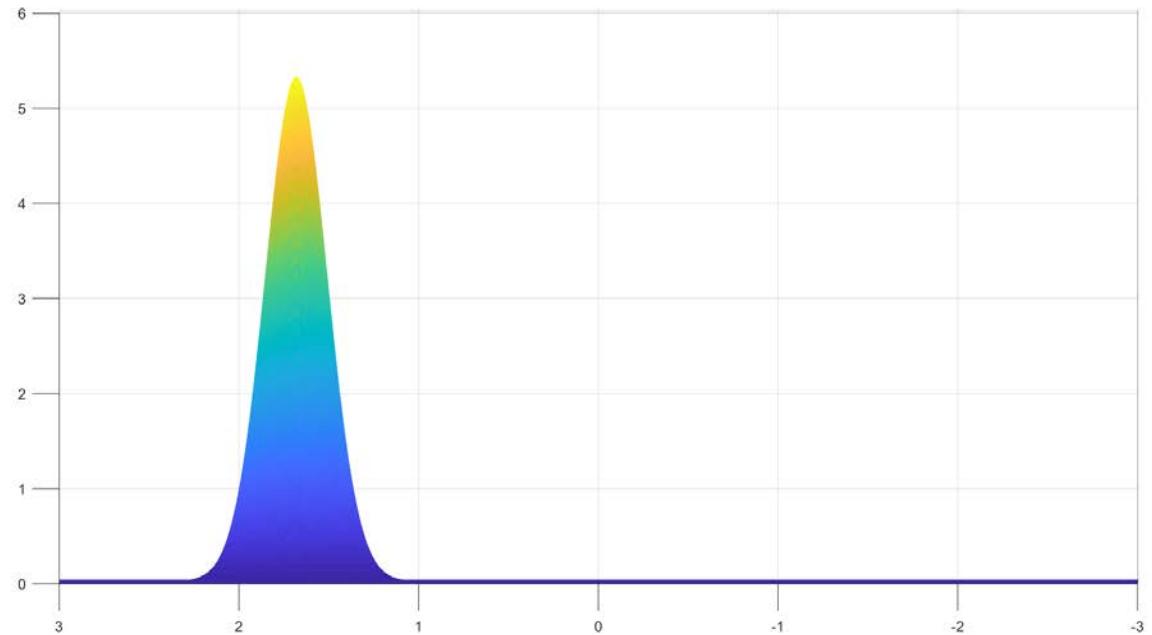
- There is a region in this area that has highly sub-optimal design.(Modeled by multivariate Gaussian)
- First sample with low rate  $r_1$ .
- Then in the high gradient area, sample with higher rate  $r_2$ .



Curve fitting result of samples with rate  $r_1$



Curve fitting result of additional samples with rate  $r_2$



Target Function

---

# Voltage Controlled Oscillator (VCO) Optimization

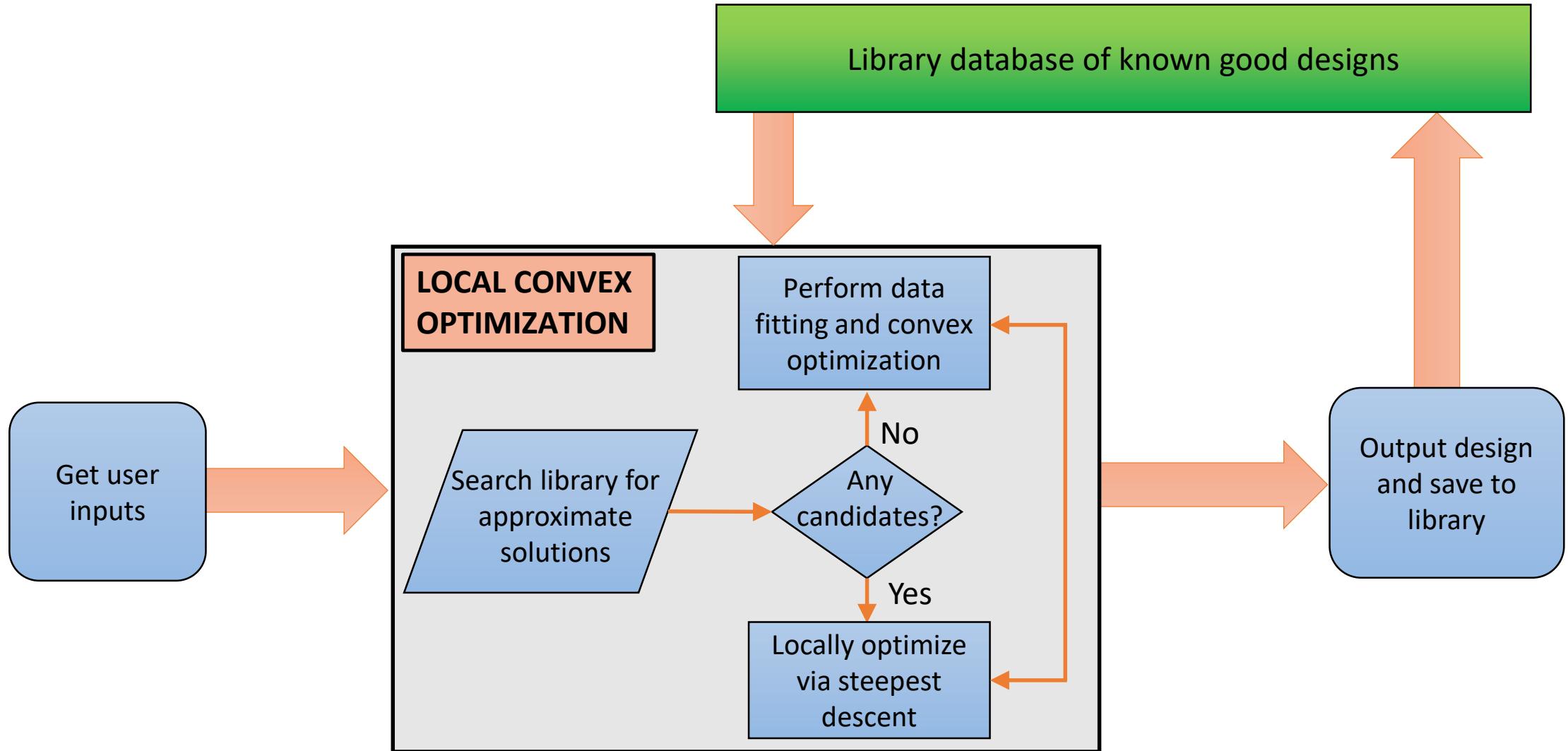
---

Jagannathan Arjun Sathyamoorthy  
Walter Unglaub

09/19/2018



# Local optimization around Known Good Designs



Library database updated each time the pipeline is executed, incrementally leading to more designs and faster optimization

# Desired VCO Operational Requirements and User Priorities for a *fixed* architecture

Specifications	Descriptions	User Priorities
$f_0$	Nominal frequency : The minimum running frequency $f_0$ of VCO when $V_{in} \geq V_{Th}$ .	$w_{f_0}$
$\Delta f = f_{max} - f_0$	Tuning range : Operational frequency range of the VCO.	$w_{\Delta f}$
(1) $J(\sigma_T)$ (2) $L(\Delta f)$	Spectral Purity : Can be specified by providing either (1) Timing Jitter (2) Phase Noise	$w_{SP}$
$P_{max}$	Power Consumed : Usually user specifies maximum permissible power consumption.	$w_P$
$SL = \frac{df_o}{dZ_o}$	Load Pulling : Sensitivity or rather insensitivity of output $f_o$ with change in output impedance $Z_o$ .	$w_{SL}$
$K_{VCO} = \frac{df_o}{dV_{in}}$	Sensitivity of VCO : For the current design we aim to have a linear VCO and hence $K_{VCO}$ is a constant determined by $\Delta f$ .	—

Aim: Design a VCO satisfying User defined operational requirements keeping hierarchy of priorities if

## Cost function for Optimization

Our variable parameters for design are the widths of the PMOS and NMOS transistors involved namely

$$W_P^i \text{ -- PMOS Width, } W_N^j \text{ -- NMOS Width where } 1 \leq i \leq N_{PMOS} \text{ & } 1 \leq j \leq N_{NMOS}$$

where  $N_{PMOS}$  and  $N_{NMOS}$  are the number of design effective PMOS and NMOS transistors in our circuit.

We view each of the design specifications as a function on our variable parameter space:

$$f_D : \mathbb{R}^{N_{PMOS}} \times \mathbb{R}^{N_{NMOS}} \rightarrow \mathbb{R} \text{ where } f_D(W_P^i, W_N^j) = y_D$$

Given a user specified list of design specification values i.e.  $(y_{D1}, y_{D2}, \dots, y_{Dk})$  along with their respective priorities  $(w_{D1}, w_{D2}, \dots, w_{Dk})$  we define our cost function using a Weighted Euclidean norm as

$$l = w_{D1}(f_{D1} - y_{D1})^2 + \dots + w_{Dk}(f_{Dk} - y_{Dk})^2$$

where we necessitate  $w_{D1} + \dots + w_{Dk} = 1$  and all weights  $w_{Di} \geq 0$ .

We then attempt to solve the Constrained Optimization problem using Convexification techniques and Simulation based Steepest Descent find the values  $(W_{PSOL}^i, W_{NSOL}^j)$  such that  $l(W_{PSOL}^i, W_{NSOL}^j) < \varepsilon$  or return that the problem has no solutions under the given constraints.

# Optimization Algorithm

Our Optimization Approach in Stages proceeds in two stages :

## Stage One : Library Exploitation and Steepest Descent

- Search Library of KGDs to find a finite set of solutions  $S$  closer than  $\varepsilon_{Stage1}$  and rank them in decreasing order of proximity.
- Perform Steepest Descent aided by Simulation to converge to a solution on the ordered set  $S$ .
- If **NO** solutions are found in  $S$  then proceed to Stage 2.

## Stage Two: Geometric Program Data fitting and Geometric Programming Unit

- Uniform Grid of Data is provided to Geometric Program Data Fit (GPDF) unit.
- GPDF fits data evaluated cost function to log – convex functions.
- Geometric Programming Unit (GP) is invoked to find the minimum.
- The parameters attaining the minimum is Simulated and validated.
- If the solution is not within tolerance of design, Steepest Descent aided by Simulation is performed using the minimum.
- If **NO** solution can be attained after Stage 2, inform the user that optimizer has no solutions under the constraint.

**The Final Solution is presented to user and stored in Database.**

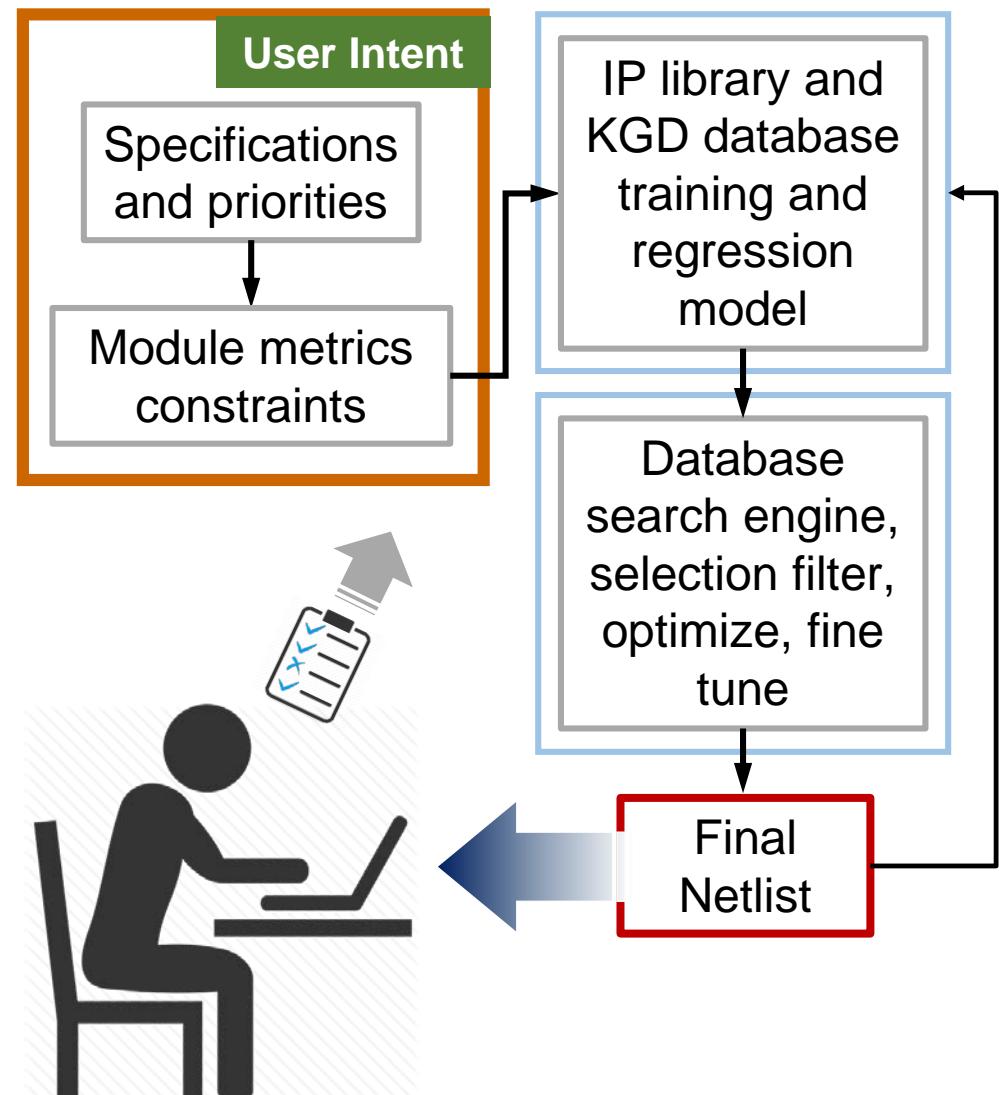
# Optimization Algorithm

---

Movie removed

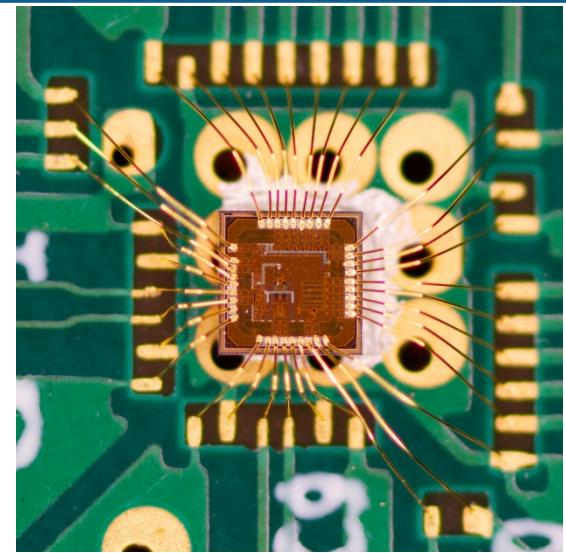
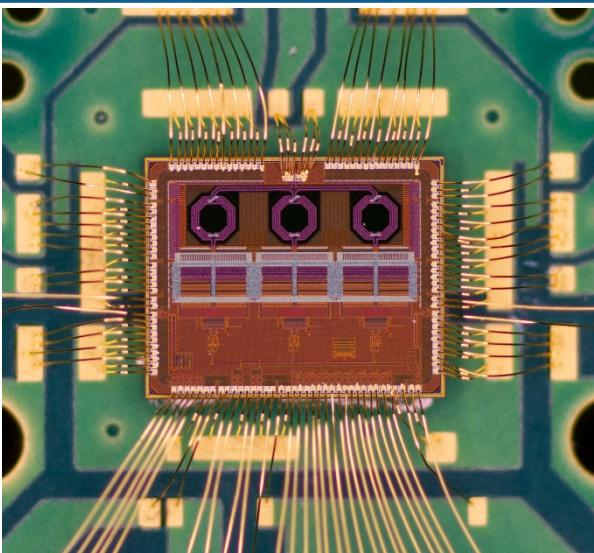
# Summary and discussion

- POSH TA-2: *Automated open-source mixed-signal IP schematic generation and distribution via MOSIS POSH repository*
- Leverage existing USC circuit designs in 180nm, 65nm and 14nm FinFET with access to advanced GF technology nodes
- Architectures for advanced technology IP blocks that meet or exceed performance metrics at month 48:
  - PLL Range: 10 MHz – 10 GHz
  - DLL Range: 10 MHz – 10 GHz
  - Analog to Digital Converters (ADC) range: 1 – 10,000 MS/s
  - Digital to Analog Converters (DAC) range: 1 – 10,000 MS/s
- Key aspect of technical approach is use of machine learning (e.g., ANN) and optimization (e.g., convex) around known-good design (KGD) points
- Library of *validated* open-source mixed-signal IP circuit modules with initial delivery to repository at month 6



# Analog and Mixed Signal Testing

- Fully Equipped Laboratory to test various Analog and Mixed Signal chips





## Additional slides

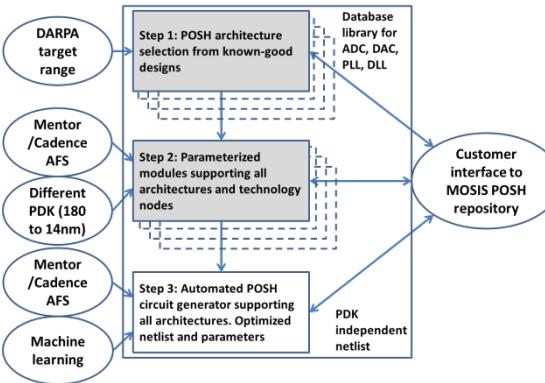
---

# Automated AMS IP generator for CMOS technologies

University of Southern California | Tony Levi | 48 months \$6M + \$1M option | MOSIS, Global Foundries  
Under contract starting June 25, 2018, FA8650-18-2-7853

## Innovative Claims

- Automated analog mixed signal (AMS) CMOS design has never been generally available.
- Today AMS is an expensive manual (human in the loop) task.
- We have a narrow focus on key AMS elements, PLL, DL, ADC, and DAC.
- Architecture choices are limited and mostly digital.
- Supported by extensive library of validated known good designs.

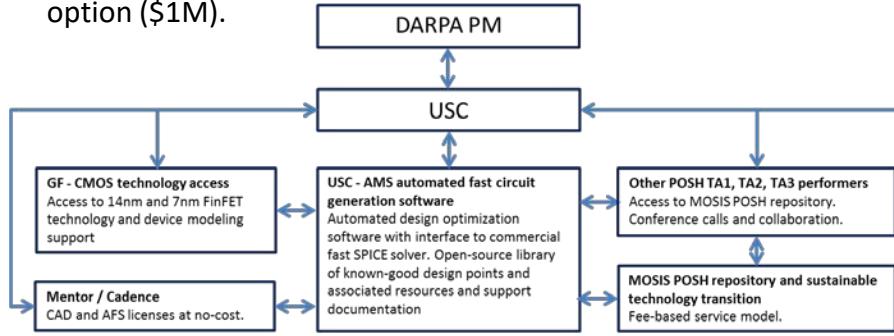


## Technical Rationale and Approach

- We are creating software that allows analog mixed signal circuits to be designed in an automated way. The objective is to dramatically increase the efficiency of analog circuit design with a goal of achieving a 24 hour design cycle.
- These aggressive technical goals will be achieved by:
  - Restricting architectural choices to limit optimization search space.
  - Restricting architectures to be as digital as possible and scale with mainstream digital CMOS technology.
  - Exploit a library of known-good designs via a template .
- Ensure a technology transition *and* design community acceptance by partnership with an established chip brokerage (MOSIS) with sustainability via fee-based added-value services.
- Initial performance tradeoff mitigated by continued machine learning and experience.

## Team organization, cost, milestones and schedule

- Phase 1 (24 months) delivery of open-source AMS design automation software and design repository (\$3.5M).
- Phase 2 (24 months following Phase 1) maintenance and preparing for technology transition (\$2.5M) and 7nm technology option (\$1M).



## Technical Area and Impact

- Delivery of efficient AMS CMOS design software capable of removing human-in-the-loop has the potential to dramatically reduce circuit development time and cost.
- Free, open-source, software in combination with known good design examples and a repository of support material enables a broad range of commercial and DoD to rapidly create mixed signal circuits.
- Within 24 months software will be available.
- Within 48 months *option* of advanced technology in 7nm with exceptional levels of performance will be available.
- Potential impact may include many applications of AMS such as sensing, wireless technologies, and IoT.
- Because this is an unrestricted open-source project, potential technology transition partners include all commercial and DoD circuit designers.



# Teaming and Personnel

Under contract from June 25, 2018, FA8650-18-2-7853

- Budget reallocation request submitted

Phase I	Affiliation and percentage time committed to project	Tasks and estimated percentage time	Roles and responsibilities	Phase I funds allocated 24 months)
Tony Levi	USC (23%), alevi@usc.edu, (213) 740-7318	Task 1 (20%), 2 (20%), 3 (20%), 4 (20%), 5 (20%)	PI and overall management. Optimization algorithms and AMS testing	
Bindu Madhavan	Consultant (20%)	Task 1 (40%), 2 (50%), 3 (10%)	AMS archive, AMS KGD, and architecture selection	
Edward Lee	Consultant (20%)	Task 1 (50%), 2 (50%)	AMS archive and AMS KGD	
Mike Chen	USC (23%), swchen@usc.edu, (213) 740-4691	Task 1 (20%), 2 (40%), 3 (40%)	AMS archive, AMS KGD, architecture selection and circuit generator	
Sandeep Gupta	USC (23%), sandeep@usc.edu, (213) 740-2251	Task 3 (100%)	Circuit generator	
Wes Hansford MOSIS	MOSIS (2%), hansford@usc.edu, (310) 448-9199		Interface to MOSIS and supervision of engineering resources	
Doan-Trang Tran MOSIS 0.3 FTE engineer	MOSIS (30%)	Task 4 (30%), 5 (70%)	Interface to MOSIS and supervision of engineering resources	
Bill Taylor	Global Foundries (1%)		Interface to Global Foundries and supervision of engineering resources	
TDB Engineer 0.5 FTE	Global Foundries (50%)	Task 2 (30%), 3 (70%)	Interface to Global Foundries and supervision of engineering resources	
			TOTAL	

## DIRECTIONS:

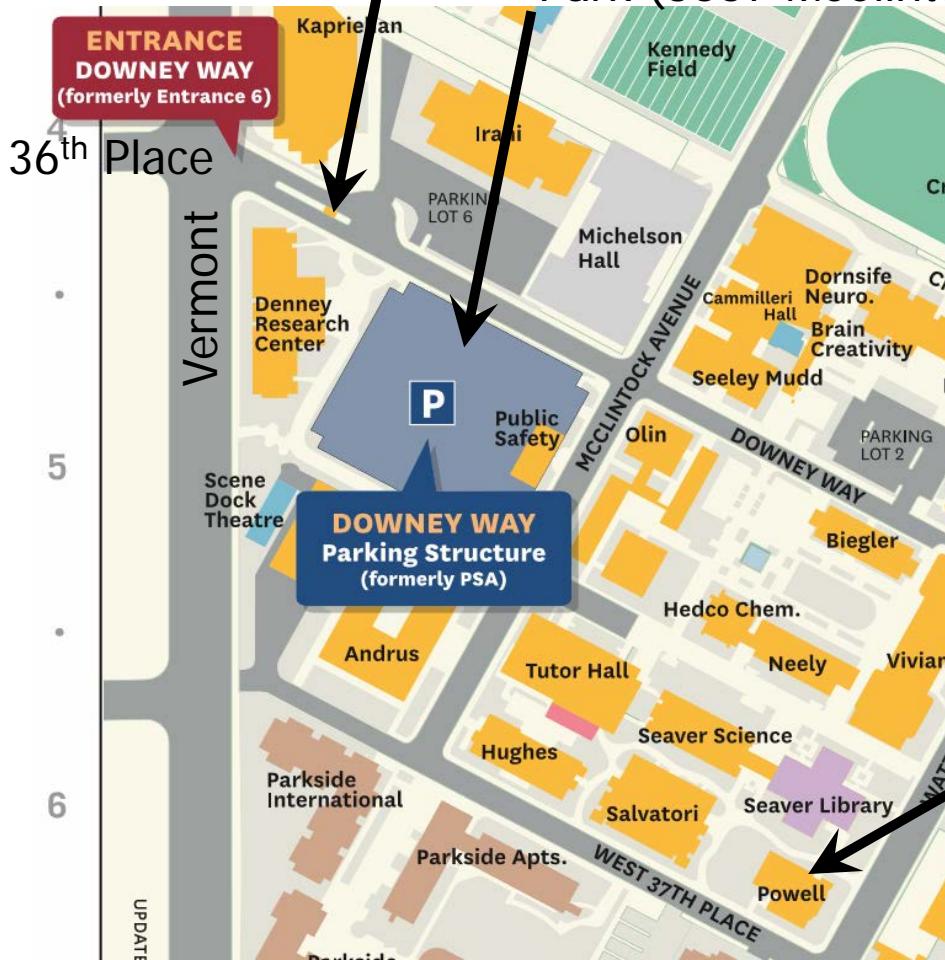
Wednesday September 19, 2018, 12.30pm-3.30pm PHE 223

Parking and directions to Powell Hall of Engineering (PHE room 223)

3737 Watt Way, Los Angeles, CA 90089

Pickup parking ticket (Abi Sivananthan, party of four)

Park (3667 McClintock Ave, Los Angeles, CA 90089)



Walk to PHE  
Room 223 is on 2<sup>nd</sup> floor