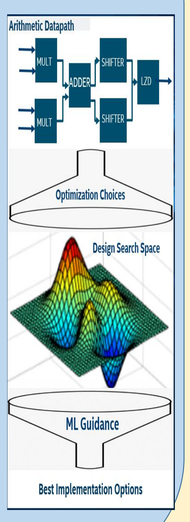
Arithmetic Datapath Design Exploration using Machine Learning

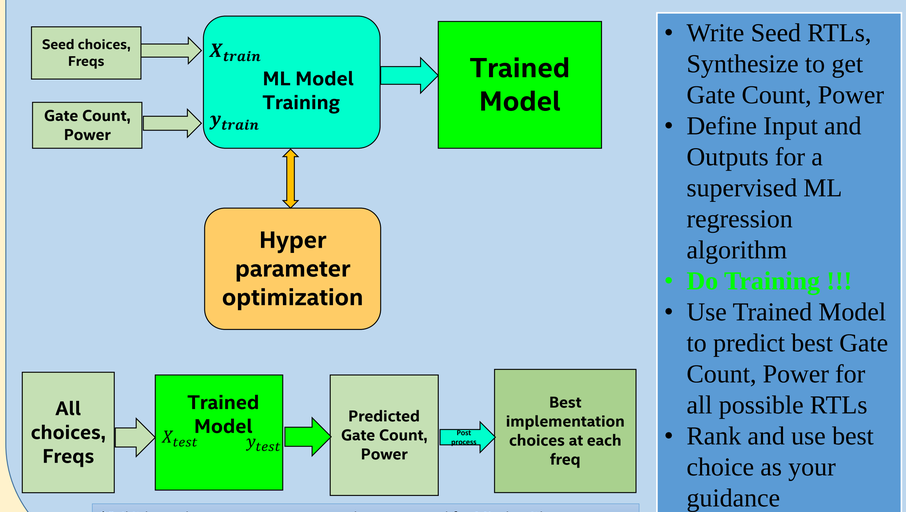
Mahesh G Vutukuri, Sreekanth Madgula, Ashutosh Garg
Intel Bangalore, Folsom
mahesh.g.Vutukuri@intel.com, sreekanth.madgula@intel.com,
ashutosh.garg@intel.com

Motivation

- All Hardware accelerators implement large arithmetic datapaths
- Large design search space to explore for optimization
  - 20 independent binary choices means 1 million options !!!
- Machine Learning (ML) techniques for intelligent guidance assist in selecting a few high-quality choices for implementation
- Proof of Concept on a small arithmetic datapath is explored
- Best Choice by Model agrees with Human choices, Shows up some new points as well



ML Based exploration overview



*Scikit learn, hyperopt open source packages are used for ML algorithms

- Write Seed RTLs, Synthesize to get Gate Count, Power
- Define Input and Outputs for a supervised ML regression algorithm
- **Do Training !!!**
- Use Trained Model to predict best Gate Count, Power for all possible RTLs
- Rank and use best choice as your guidance

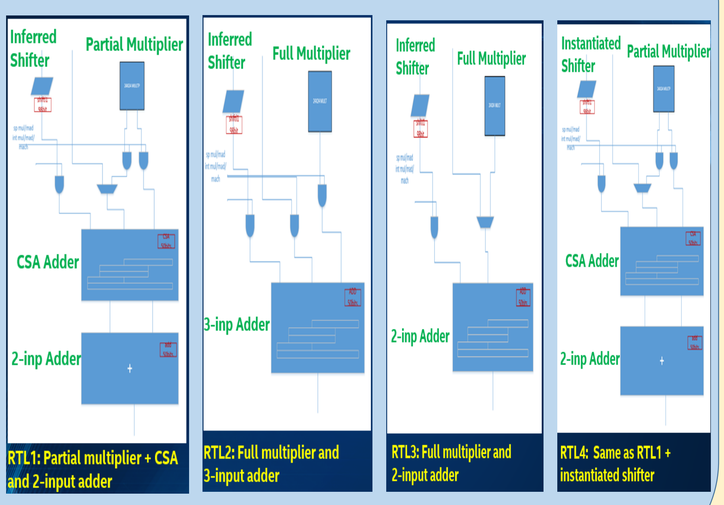
ML Training: Input Seed data Preparation

Component	Implementation	Encoding	RTL1	RTL2	RTL3	RTL4	RTL5	RTL6
Mult+Add	full mult w/ 3 input addition	0	3	0	1	3	1	3
	full mult w/2 input addition	1						
	partial mult w/3 input addition	2						
	partial mult w/csa+2input addition	3						
Shifter	inferred	0	0	0	0	1	0	0
	Instantiated	1						
LZD	top_hierarchy	0	0	0	0	0	1	1
	current_hierarchy	1						
Pipelining	Manual	0	0	0	0	0	1	1
	Auto	1						

	Input Features (Xn)					Output Features (Y)	
	Freq	Mult+adder*	Shifter	LZD	Pipeline	GateCount(K)	Power (uW)
Training Set	Freq_1	3	0	0	0	322	67.3
	Freq_1	0	0	0	0	320	68.1
	Freq_1	1	0	0	0	314	67.1
	...	...	....	...	...	...	...
	Freq_11	3	1	0	0	403	108.9
	Freq_11	1	0	1	1	464	134.4
	Freq_11	3	0	1	1	428	120.4

- 6 Seed RTLs are written and encoded – form Input Features of ML Model
- Synthesized at 11 different frequencies , form Input Samples of ML Model , 11 Freqs x 6 RTLs = 66 samples
- Syn Gate Count, Cdyn: form Target features of ML Model

Seed RTL examples

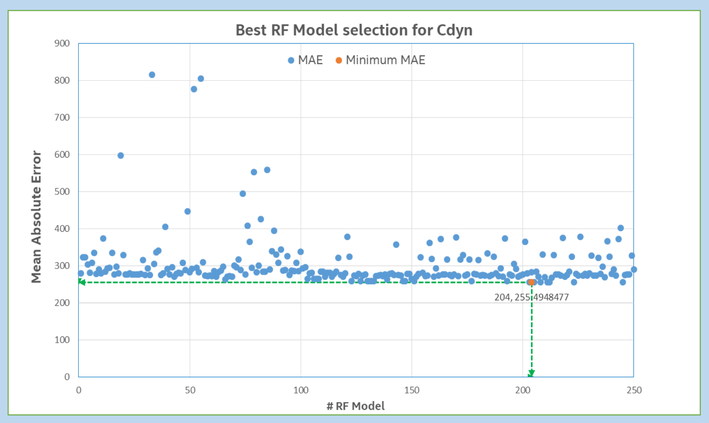


4 out of 6 seed RTLs are shown , remaining 2 are auto pipelined

ML Training and Cross Validation

- Posed ML Algorithm selection along with Hyper parameter tuning as Model search space problem for hyperopt package
- Hyper-parameter search is done using Bayesian algorithms (ex: TPE) to cover many possible values of learning_rate, no.of trees, tree_depth ..etc with fewer iterations
- Obtained Best Model using Group Kfold Cross-validation Mean absolute Percentage Error
- Random Forest Model is the winner
- Trained RF Model with all input training data

Best Model Selection of Cross-Validation



- 250 CV RF Models are built
- 204th RF Model has lowest MAE, MAPE – Selected 204th Hyper-parameters for final RF Model training

ML Inference

	X_test					Y_test
RTL_ID	Freq	mult_adder	shifter	lzd	pipeline	Power(uW)
RTL1	freq1	3	0	0	0	60.0
RTL1	freq2	3	0	0	0	60.1
RTL1	freq3	3	0	0	0	60.1
...	...	...	...	...	...	
RTL1	freq_11	3	0	0	0	84.7
...	...	...	...	...	...	
RTL4	freq1	3	1	0	0	60.8
RTL4	freq2	3	1	0	0	61.2
...	...	...	...	...	...	
RTL4	freq_11	3	1	0	0	78.9
...	...	...	...	...	...	
RTL7	freq1	0	0	0	1	64.2
RTL7	freq2	0	0	0	1	64.6
...	...	...	...	...	...	
RTL7	freq11	0	0	0	1	95.3
...	...	...	...	...	...	
RTL20	freq1	2	0	0	0	59.9
RTL20	freq2	2	0	0	0	59.8
...	...	...	...	...	...	
RTL20	freq11	2	0	0	0	87.3
...	...	...	...	...	...	
RTL32	freq1	3	1	1	1	64.4
RTL32	freq2	3	1	1	1	63.2
...	...	...	...	...	...	
RTL32	freq11	3	1	1	1	87.9

Trained Model → Post process

- Enumerate all possible test set inputs
- New test input data is sent to Model.
- Model predicts Gate Count, Cdyn

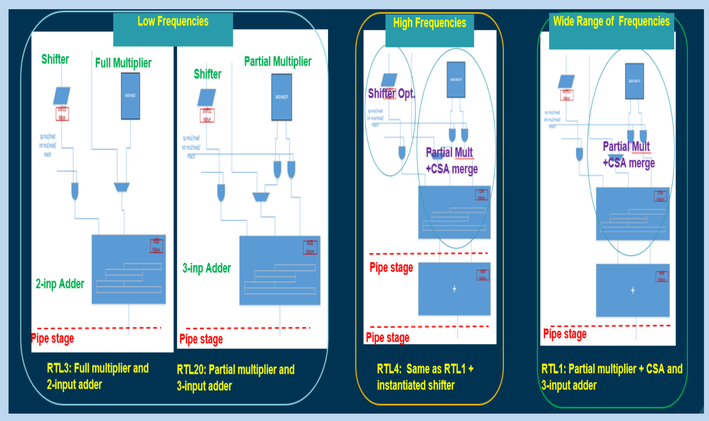
ML Inference Results Analysis

Frequency	RTL_Power_GC_Sorted
Low freq range	[[3, 20], [1]]
	[[3, 20], [1]]
	[[3, 20], [1]]
	[[3, 20], [1]]
	[[3, 20], [1]]
	[[1], [3, 20]]
	[[1], [3, 20]]
High freq range	[[4], [16, 24]]
	[[4], [16, 24]]
	[[1], [4]]
	[[1], [4]]
	[[1], [4]]
	[[31], [4]]
	[[4], [16, 24]]

RTL[3,20]@ low freq, RTL4 at high freq, RTL1 for wide range of frequencies

New RTL20 is predicted to be better at Low Frequencies.

ML Inference Results Analysis ..contd



Testing through Synthesis

Proof of pudding is in eating

- Wrote RTL20 and Synthesized
- Compared Gate Count, Power Predicted Vs Realized
- Gate-Count Error : **2.1%**
- Power Error : **5.8%**

Summary & Next steps

ML method is a "quick prototyping flow" for datapath design which accelerated design space exploration during early design cycle.

- ML was leveraged to guide component selection for optimization.
  - Evaluated multiple u-arch choices at a fraction of the effort.
  - Saved valuable designer time focusing on a few best choices.
- Many new choice predictions were too close to seed data.
  - Indicated limited set of input samples in training data. Bigger scope problem will not have this problem.
  - Component level data to provide additional granularity
- PoC work is presented for datapath design exploration.
  - Use case of arithmetic datapath lends well to ML problem formulation.
  - Further work is needed in applying to other datapaths and control logic.

Bibliography

1. Hung-Yi Liu and L. P. Carloni, "On learning-based methods for design-space exploration with High-Level Synthesis," 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 2013, pp. 1-7.
2. [2] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011
3. [3] https://github.com/hyperopt/hyperopt