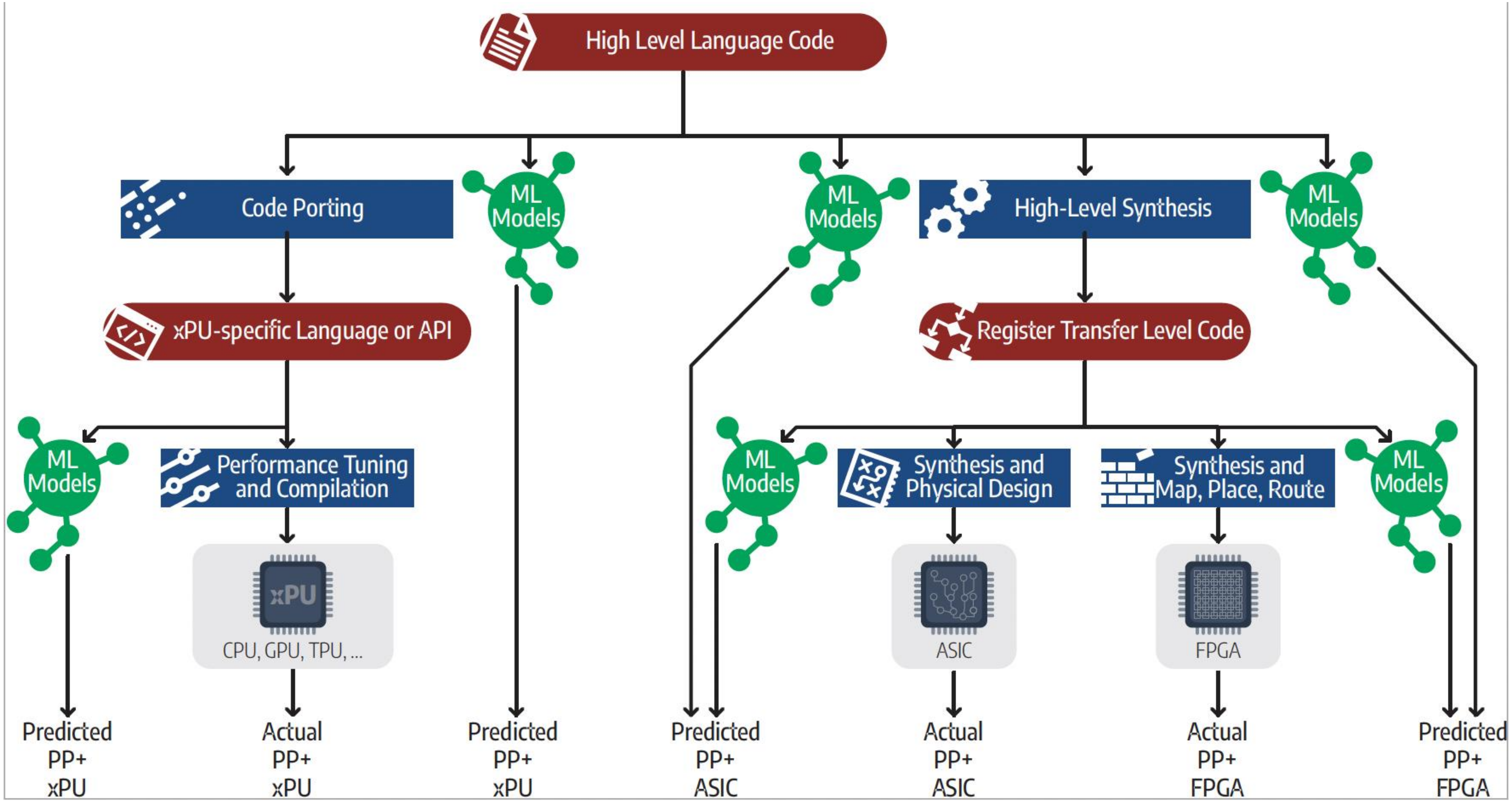
The University of Texas at Austin
Electrical and Computer Engineering
Cockrell School of Engineering

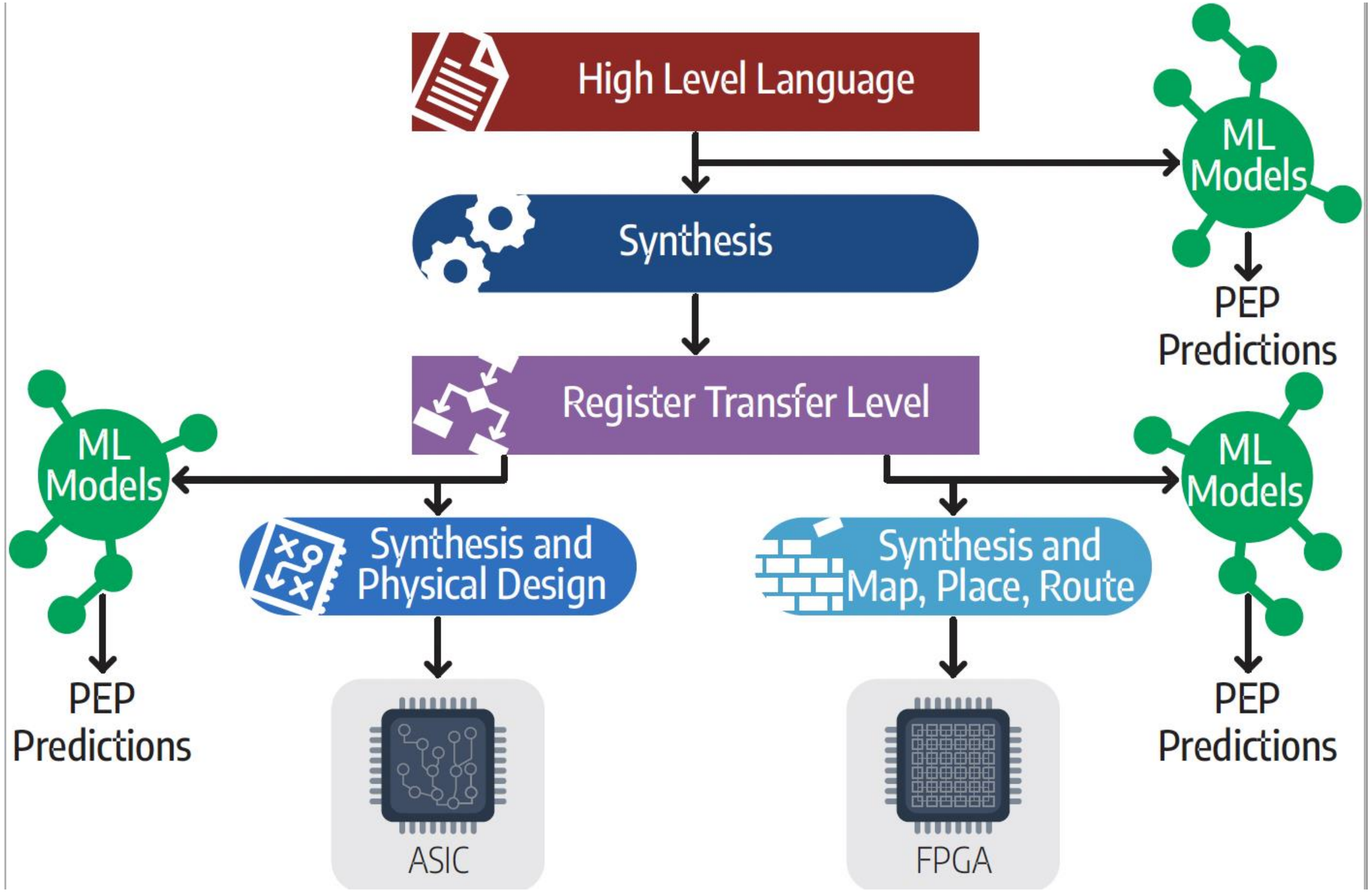
ML4ACCEL: AN OPEN-SOURCE DATASET FOR ML-GUIDED ACCELERATOR DESIGN

Zhigang Wei, Aman Arora, Ruihao Li, Lizy John
The Laboratory for Computer Architecture (LCA)
The University of Texas at Austin

ML for Accelerator Design

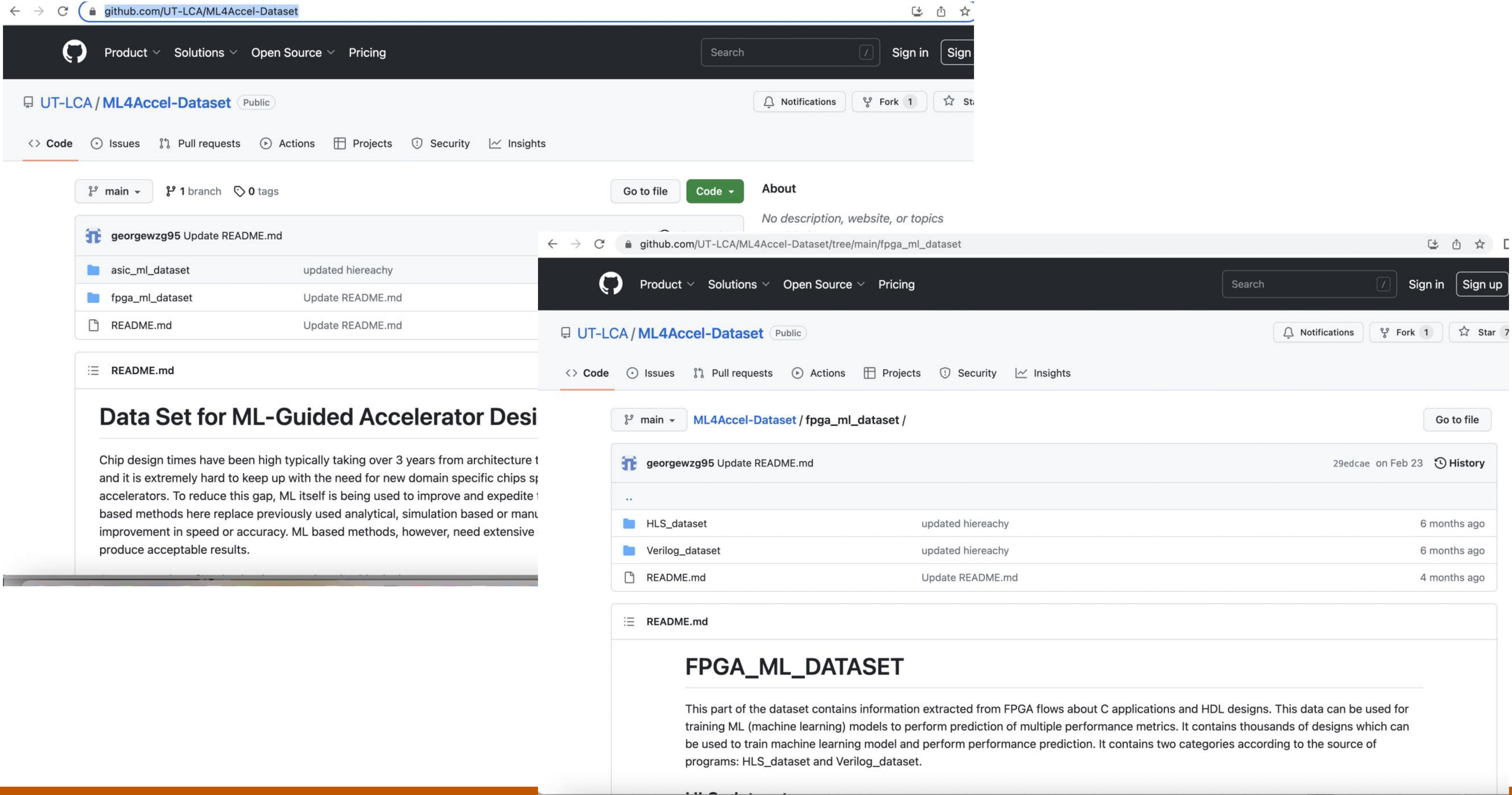


ML for Accelerator Design



ML for Design

- Performance Estimates
- Power Estimates
- Resource Usage and Timing Estimates
- Operation Delay Estimates
- Congestion Prediction
- DRC Violation Prediction
- IR Drop Prediction
- Estimation of Quality of Synthesis



No description, website, or topics

georgewzg95 Update README.md

asic_ml_dataset	updated hiereachy
fpga_ml_dataset	Update README.md
README.md	Update README.md

README.md

Data Set for ML-Guided Accelerator Desi

Chip design times have been high typically taking over 3 years from architecture t
and it is extremely hard to keep up with the need for new domain specific chips sp
accelerators. To reduce this gap, ML itself is being used to improve and expedite t
based methods here replace previously used analytical, simulation based or manu
improvement in speed or accuracy. ML based methods, however, need extensive
produce acceptable results.

georgewzg95 Update README.md 29edcae on Feb 23

..

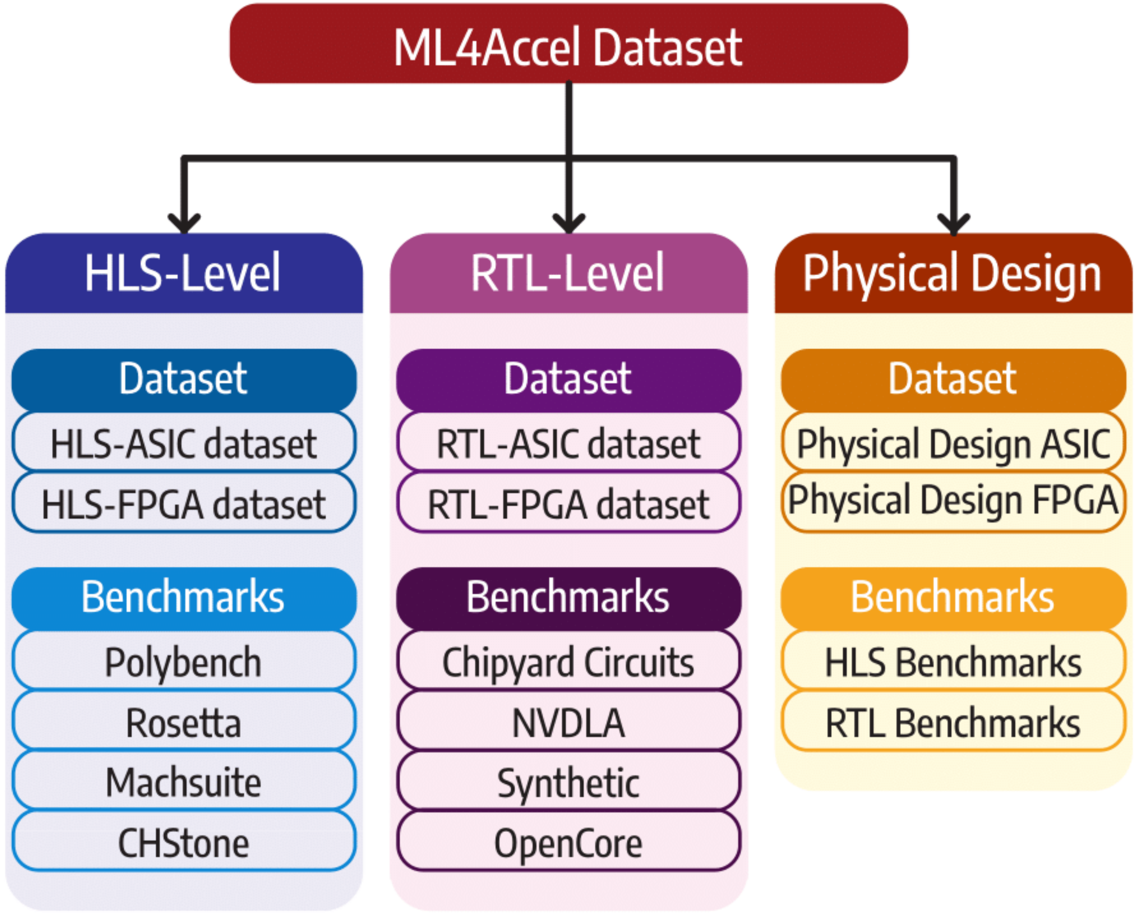
HLS_dataset	updated hiereachy	6 months ago
Verilog_dataset	updated hiereachy	6 months ago
README.md	Update README.md	4 months ago

README.md

FPGA_ML_DATASET

This part of the dataset contains information extracted from FPGA flows about C applications and HDL designs. This data can be used for training ML (machine learning) models to perform prediction of multiple performance metrics. It contains thousands of designs which can be used to train machine learning model and perform performance prediction. It contains two categories according to the source of programs: HLS_dataset and Verilog_dataset.

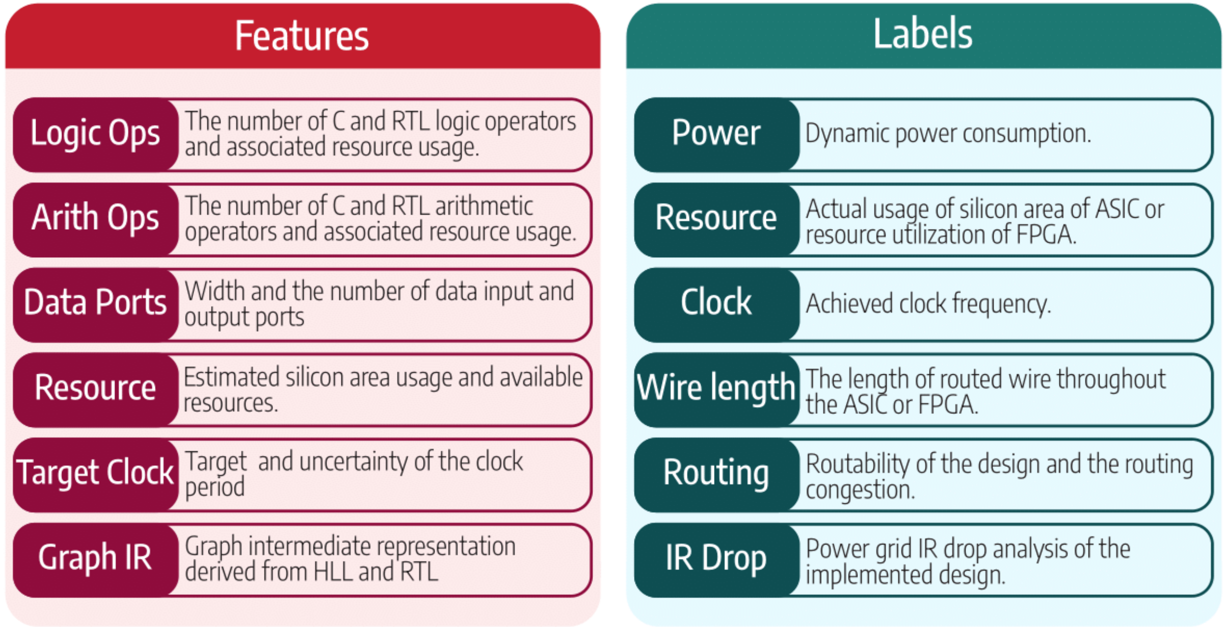
Overview of ML4Accel



ML4Accel Dataset

- HLS-Level
  - Dataset
    - HLS-ASIC dataset
    - HLS-FPGA dataset
  - Benchmarks
    - Polybench
    - Rosetta
    - Machsuite
    - CHStone
- RTL-Level
  - Dataset
    - RTL-ASIC dataset
    - RTL-FPGA dataset
  - Benchmarks
    - Chipyard Circuits
    - NVDLA
    - Synthetic
    - OpenCore
- Physical Design
  - Dataset
    - Physical Design ASIC
    - Physical Design FPGA
  - Benchmarks
    - HLS Benchmarks
    - RTL Benchmarks

ML4Accel dataset domains



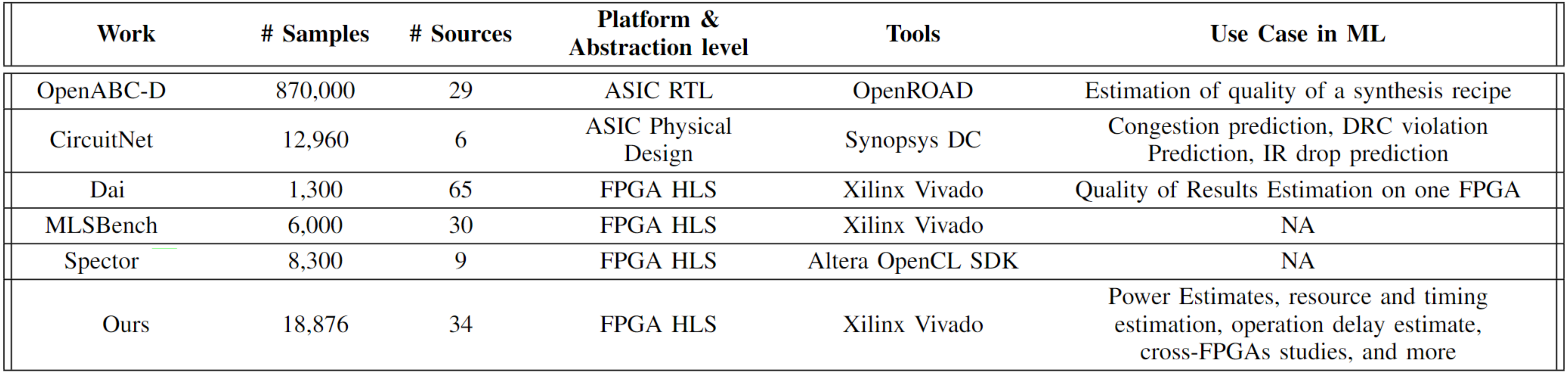
Features	
Logic Ops	The number of C and RTL logic operators and associated resource usage.
Arith Ops	The number of C and RTL arithmetic operators and associated resource usage.
Data Ports	Width and the number of data input and output ports
Resource	Estimated silicon area usage and available resources.
Target Clock	Target and uncertainty of the clock period
Graph IR	Graph intermediate representation derived from HLL and RTL

Labels	
Power	Dynamic power consumption.
Resource	Actual usage of silicon area of ASIC or resource utilization of FPGA.
Clock	Achieved clock frequency.
Wire length	The length of routed wire throughout the ASIC or FPGA.
Routing	Routability of the design and the routing congestion.
IR Drop	Power grid IR drop analysis of the implemented design.

ML4Accel dataset features and labels

HLS-FPGA dataset: HLSDataset

HLSDataset is a subset of ML4Accel targeting prediction tasks in FPGA via HLS



Work	# Samples	# Sources	Platform & Abstraction level	Tools	Use Case in ML
OpenABC-D	870,000	29	ASIC RTL	OpenROAD	Estimation of quality of a synthesis recipe
CircuitNet	12,960	6	ASIC Physical Design	Synopsys DC	Congestion prediction, DRC violation Prediction, IR drop prediction
Dai	1,300	65	FPGA HLS	Xilinx Vivado	Quality of Results Estimation on one FPGA
MLSBench	6,000	30	FPGA HLS	Xilinx Vivado	NA
Spector	8,300	9	FPGA HLS	Altera OpenCL SDK	NA
Ours	18,876	34	FPGA HLS	Xilinx Vivado	Power Estimates, resource and timing estimation, operation delay estimate, cross-FPGAs studies, and more

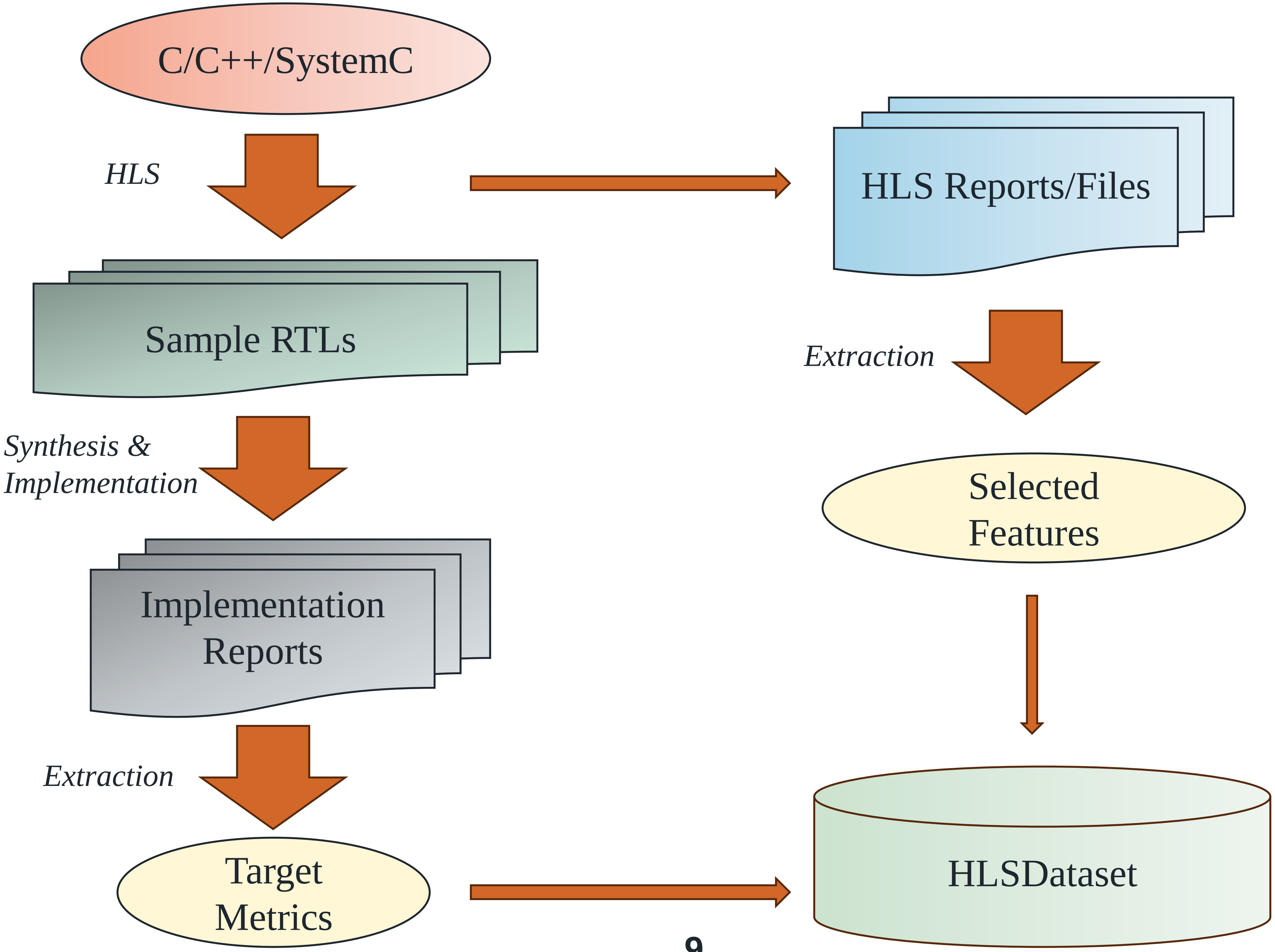
Comparison between HLSDataset and prior open-source datasets for training ML models for chip design

Generation of HLSDataset

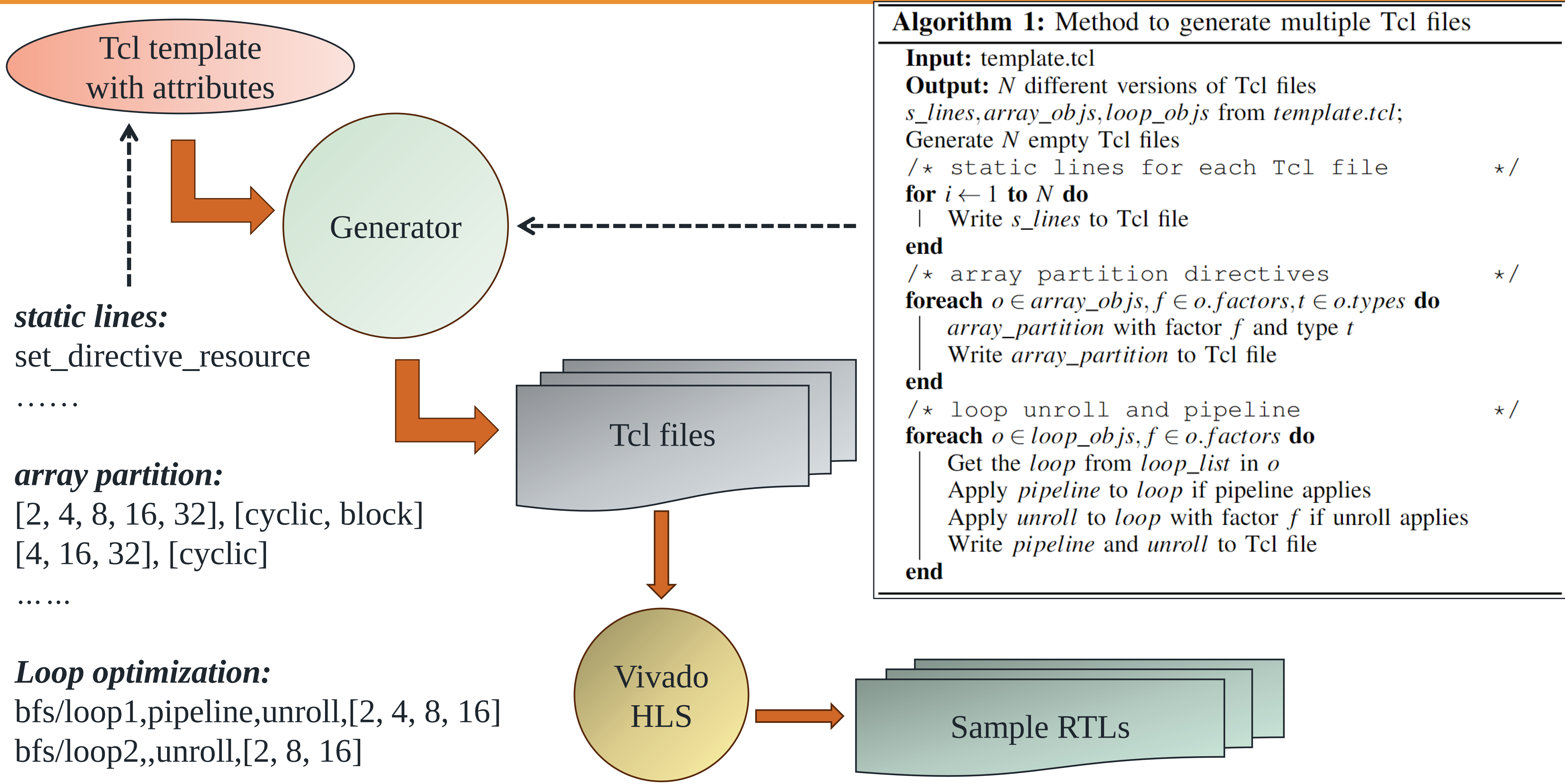
Category	Details
Num samples	18,876
Num applications	34
Application sources	Polybench, Machsuite, CHStone, Rosetta
FPGAs	Xilinx ZU9EG, XC7V585T
Clock frequency	100MHz
Domains	Multimedia, Arithmetic, Signal Processing, ML
Size	~50GB
Machines	9 16-coe Intel Xeon 5218 2.3GHz with 384GB RAM
Time	More than 1,500 hours
Tools	Xilinx Vivado/Vitis

HLSDataset generation efforts and used tools

Generation of HLSDataset



Generation of HLSDataset



static lines:
set_directive_resource
……

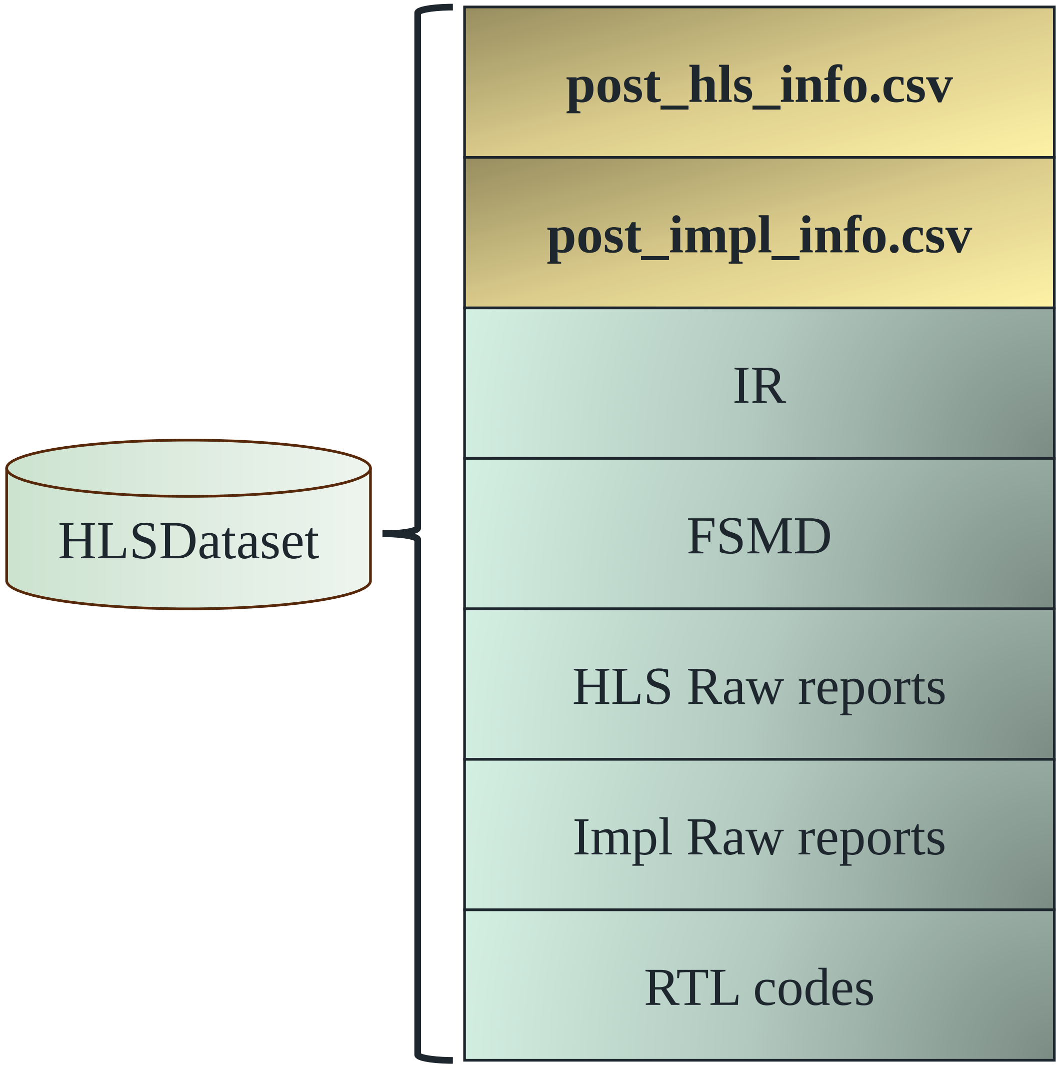
array partition:
[2, 4, 8, 16, 32], [cyclic, block]
[4, 16, 32], [cyclic]
… …

Loop optimization:
bfs/loop1,pipeline,unroll,[2, 4, 8, 16]
bfs/loop2,,unroll,[2, 8, 16]
… …

Algorithm 1: Method to generate multiple Tcl files

```
Input: template.tcl
Output: $N$ different versions of Tcl files
$s\_lines, array\_objs, loop\_objs$ from $template.tcl$;
Generate $N$ empty Tcl files
/* static lines for each Tcl file */
for $i \leftarrow 1$ to $N$ do
    Write $s\_lines$ to Tcl file
end
/* array partition directives */
foreach $o \in array\_objs, f \in o.factors, t \in o.types$ do
    $array\_partition$ with factor $f$ and type $t$
    Write $array\_partition$ to Tcl file
end
/* loop unroll and pipeline */
foreach $o \in loop\_objs, f \in o.factors$ do
    Get the $loop$ from $loop\_list$ in $o$
    Apply $pipeline$ to $loop$ if pipeline applies
    Apply $unroll$ to $loop$ with factor $f$ if unroll applies
    Write $pipeline$ and $unroll$ to Tcl file
end
```

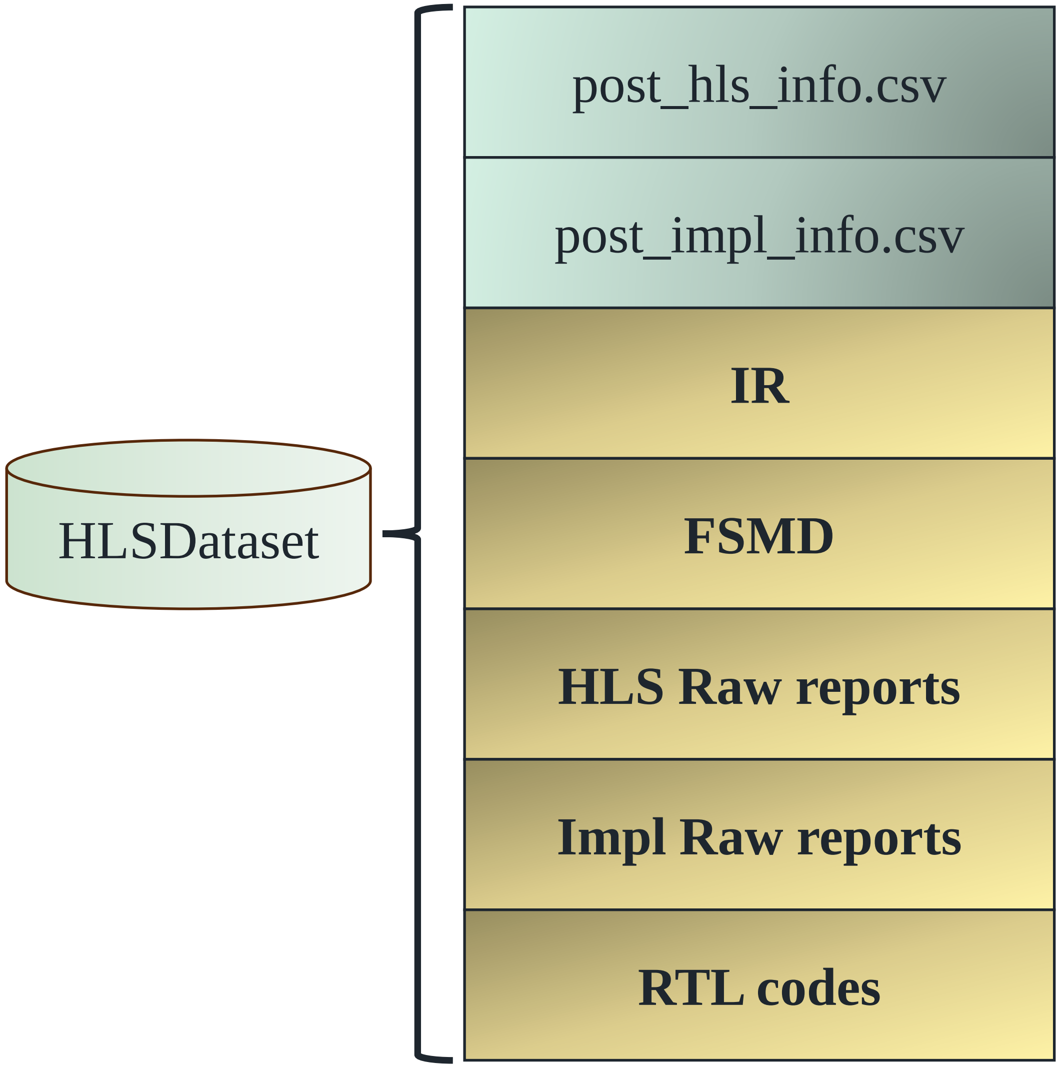
HLSDataset Contents



Category	***post_hls_info.csv*** **description**
Resource #	Estimated usage and available number of BRAM, DSP, FF and LUT
Clock	Target, estimation and uncertainty of the clock period
Logic ops	The number of C and RTL logic operators and associated resource usage
Arith ops	The number of C and RTL arithmetic operators and associated resource usage
Data ports	Width and the number of data input and output ports

Category	***post_implementation_info.csv*** **description**
power	Simulation-based dynamic power consumption
Resource #	Actual usage of BRAM, DSP, FF and LUT
Clock	Achieved clock frequency

HLSDataset Contents



IR: IR codes and operator information

FSMD model: Finite State Machine Datapath model

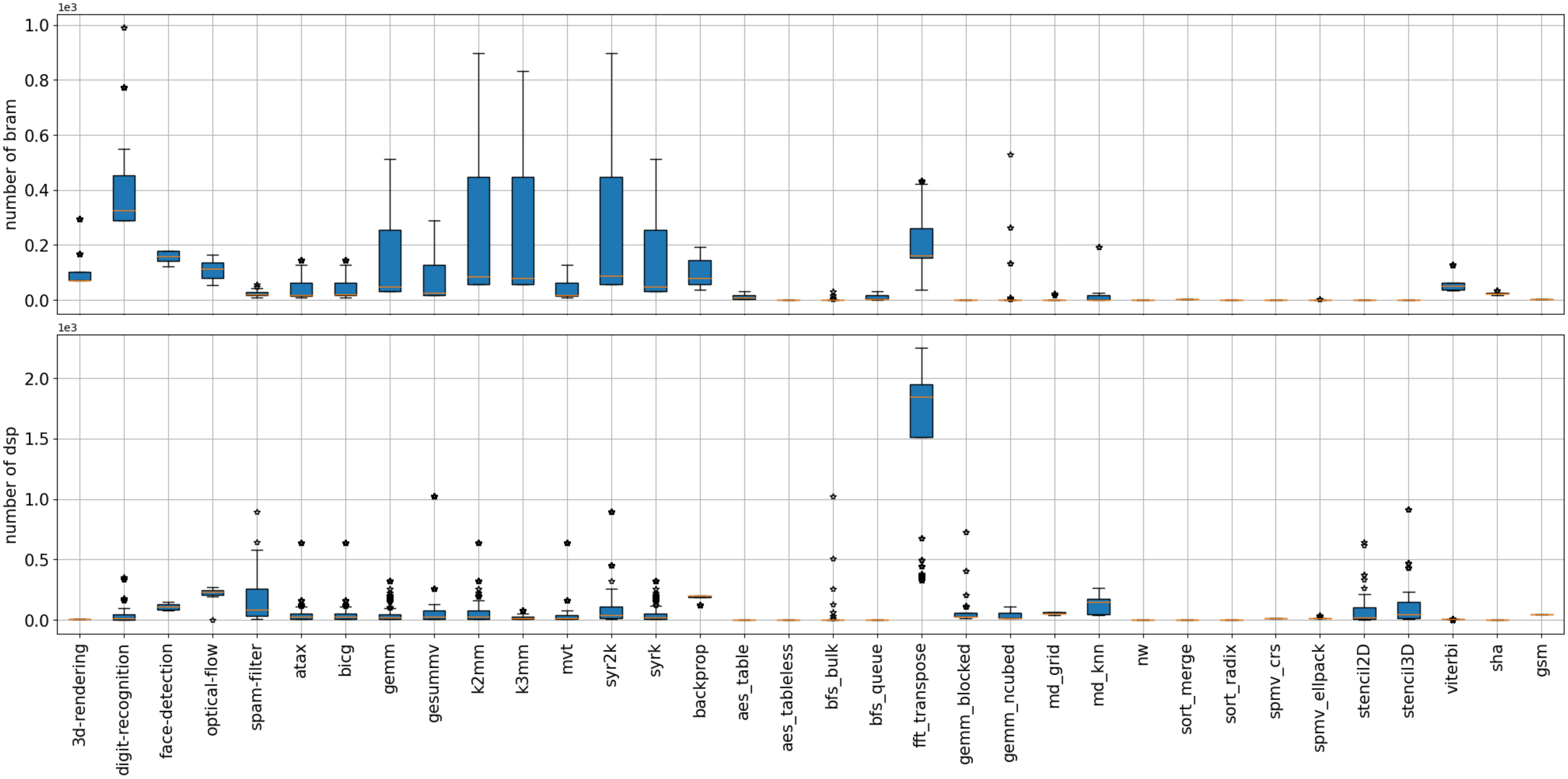
HLS Raw reports: resource usage estimation from HLS

Impl Raw reports: Resource utilization and timing reports generated after implementation

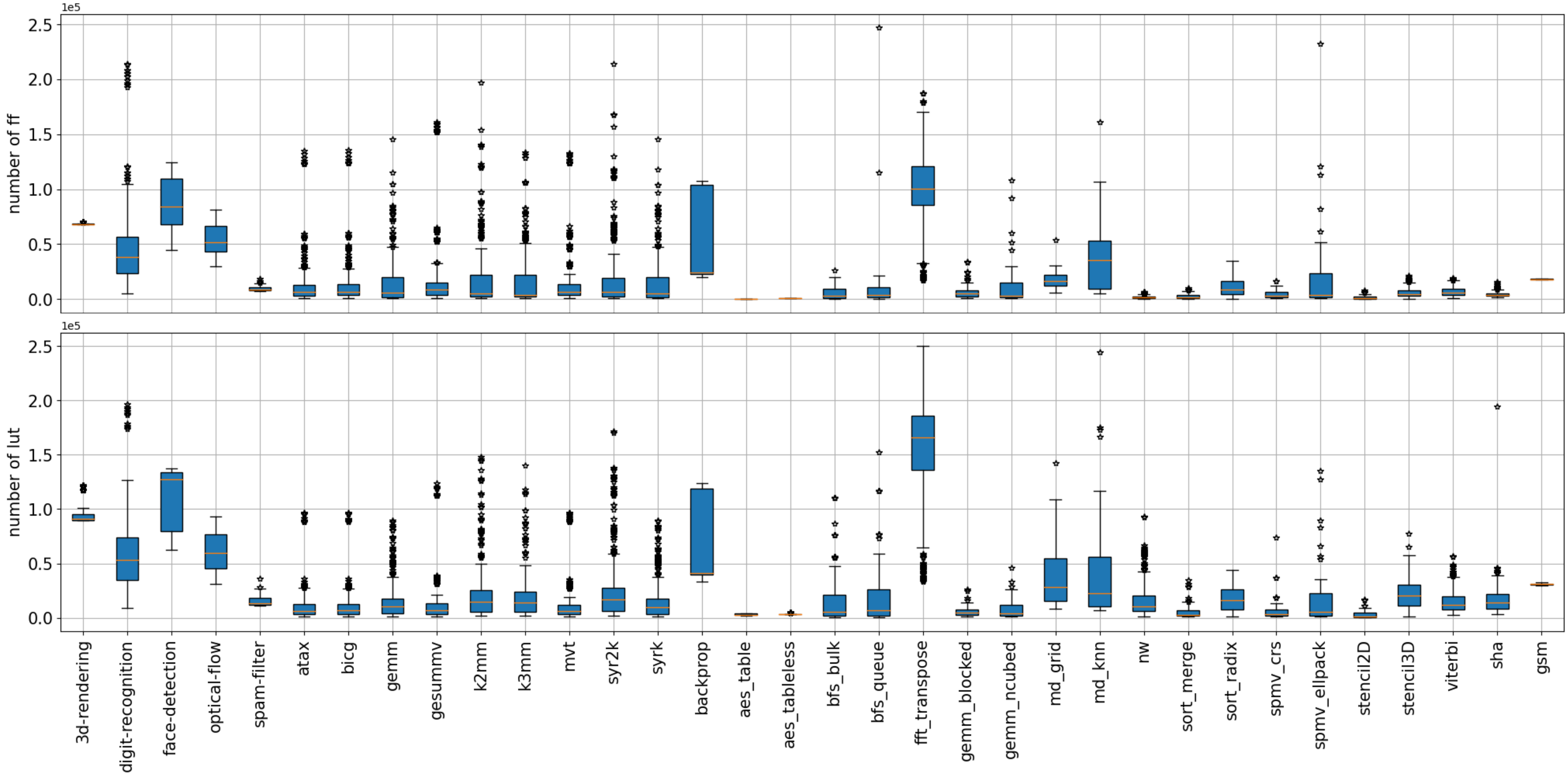
RTL codes

The University of Texas at Austin
Electrical and Computer Engineering
Cockrell School of Engineering

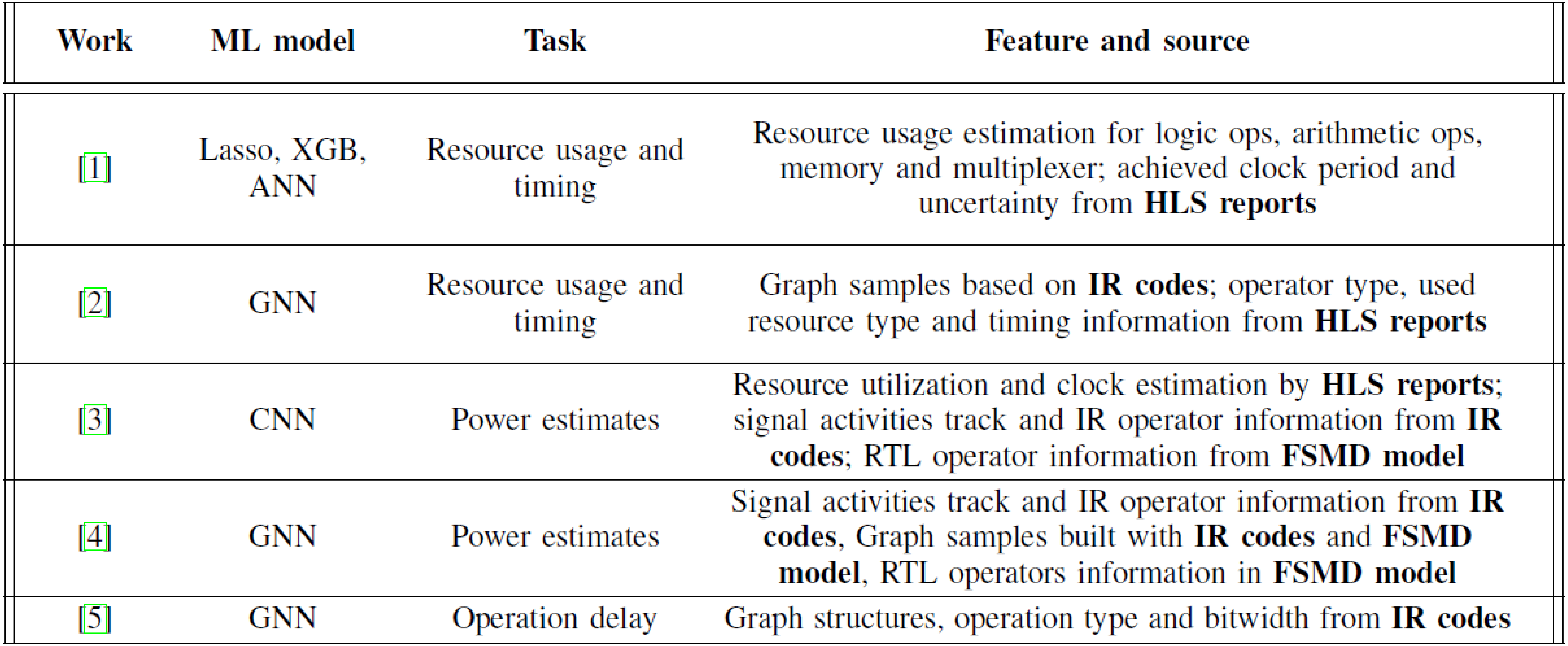
HLSDataset Resource Distribution for ZU9EG



HLSDataset Resource Distribution for ZU9EG



Potential Use Cases with HLSDataset

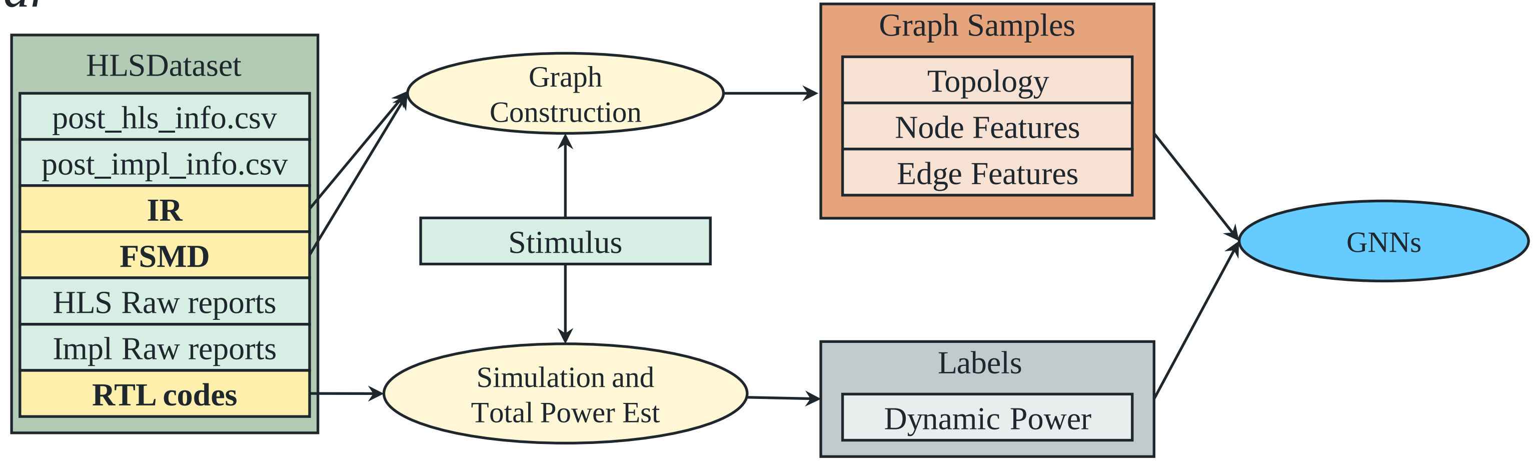


Work	ML model	Task	Feature and source
[1]	Lasso, XGB, ANN	Resource usage and timing	Resource usage estimation for logic ops, arithmetic ops, memory and multiplexer; achieved clock period and uncertainty from **HLS reports**
[2]	GNN	Resource usage and timing	Graph samples based on **IR codes**; operator type, used resource type and timing information from **HLS reports**
[3]	CNN	Power estimates	Resource utilization and clock estimation by **HLS reports**; signal activities track and IR operator information from **IR codes**; RTL operator information from **FSMD model**
[4]	GNN	Power estimates	Signal activities track and IR operator information from **IR codes**, Graph samples built with **IR codes** and **FSMD model**, RTL operators information in **FSMD model**
[5]	GNN	Operation delay	Graph structures, operation type and bitwidth from **IR codes**

- ***Power Estimates***
- ***Resource Utilization Estimates***
- ***Timing and Operation Delay Estimates***

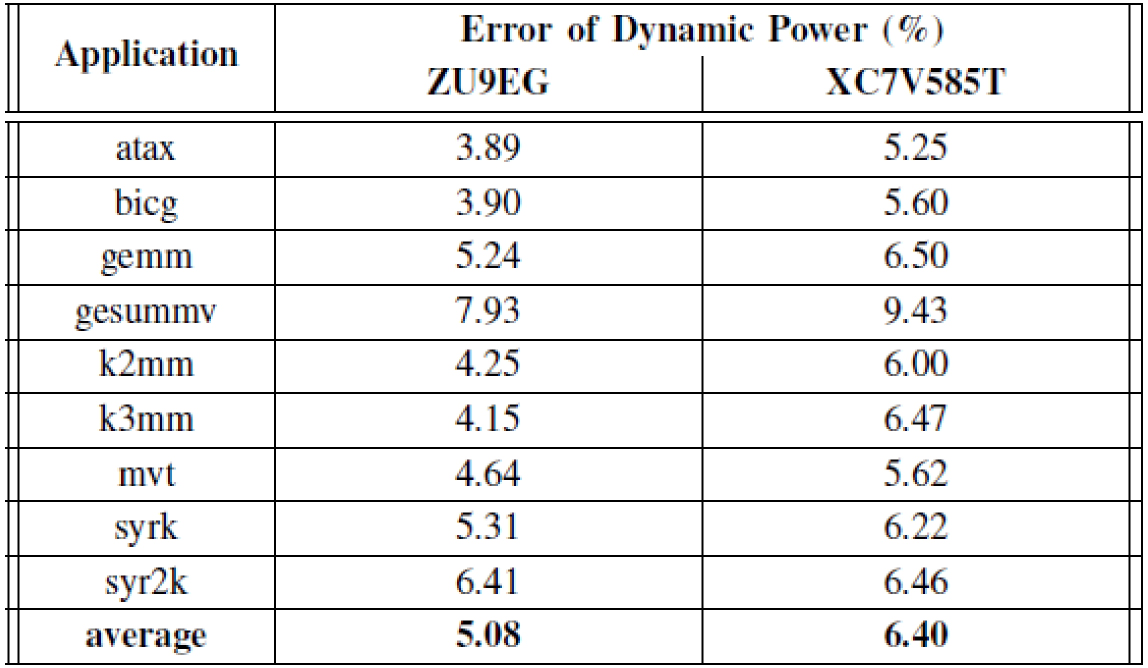
Case Study 1 with HLSDataset

Dynamic Power Estimation using PowerGear



Training setup:

- GNN model from PowerGear [4]
- Polybench subsets from HLSDataset
- 10-fold cross validation
- Trained on Ampere A100



Application	Error of Dynamic Power (%)	
	ZU9EG	XC7V585T
atax	3.89	5.25
bicg	3.90	5.60
gemm	5.24	6.50
gesummv	7.93	9.43
k2mm	4.25	6.00
k3mm	4.15	6.47
mvt	4.64	5.62
syrk	5.31	6.22
syr2k	6.41	6.46
average	**5.08**	**6.40**

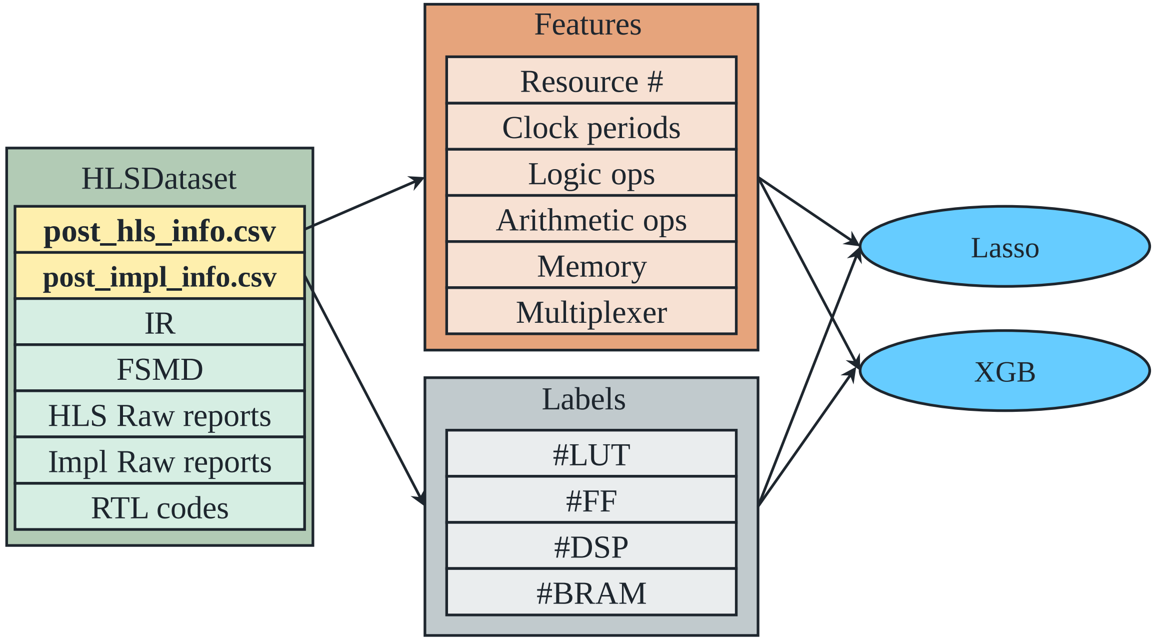
Dynamic power estimation errors - Training dataset and testing dataset are from Polybench subset of HLSDataset. Results for ZU9EG and XC7V585T.

Case Study 2 with HLSDataset

Resource Usage estimation

Training setup:

- XGB and Lasso model from S. Dai et al. [1]
- Polybench, Machsuite subsets from HLSDataset
- 10-fold cross validation
- Trained on Ampere A100



HLSDataset

- post_hls_info.csv
- post_impl_info.csv
- IR
- FSMD
- HLS Raw reports
- Impl Raw reports
- RTL codes

Features

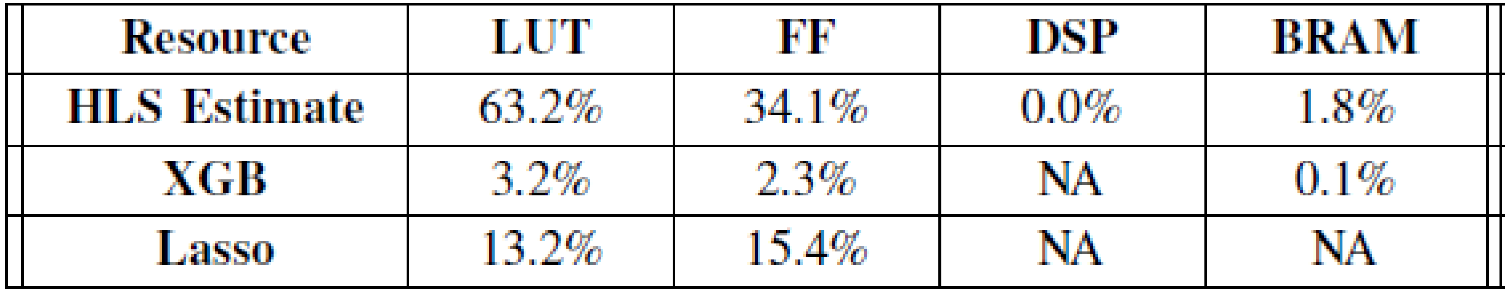
- Resource #
- Clock periods
- Logic ops
- Arithmetic ops
- Memory
- Multiplexer

Labels

- #LUT
- #FF
- #DSP
- #BRAM

Lasso

XGB



Resource	LUT	FF	DSP	BRAM
HLS Estimate	63.2%	34.1%	0.0%	1.8%
XGB	3.2%	2.3%	NA	0.1%
Lasso	13.2%	15.4%	NA	NA

***Resource estimation errors* -** Training dataset and testing dataset are from Machsuite and Polybench subsets of HLSDataset. Results for ZU9EG.

Thanks!

https://github.com/UT-LCA/ML4Accel-Dataset

Project is ongoing at

Laboratory for Computer Architecture

lca.ece.utexas.edu

We acknowledge a Meta’s support for our research via AI4AI Research gift

The University of Texas at Austin
Electrical and Computer Engineering
Cockrell School of Engineering

References

[1] S. Dai, Y. Zhou, H. Zhang, E. Ustun, E. F. Y. Young and Z. Zhang,”Fast and Accurate Estimation of Quality of Results in High-Level Synthesis with Machine Learning”, International Symposium on Field-Programmable Custom Computing Machines (FCCM), 2018.
[2] N. Wu, H. Yang, Y. Xie, P. Li, and C. Hao, ”High-level synthesis performance prediction using GNNs: benchmarking, modeling, and advancing”, Design Automation Conference (DAC), 2022.
[3] Z. Lin, J. Zhao, S. Sinha and W. Zhang, ”HL-Pow: A Learning-Based Power Modeling Framework for High-Level Synthesis”, Asia and South Pacific Design Automation Conference (ASP-DAC), 2020.
[4] Z. Lin, Z. Yuan, J. Zhao, W. Zhang, H. Wang, and Y. Tian, ”PowerGear: early-stage power estimation in FPGA HLS via heterogeneous edgecentric GNNs”, Conference and Exhibition on Design, Automation and Test in Europe (DATE), 2022.
[5] E. Ustun, C. Deng, D. Pal, Z. Li, and Z. Zhang, ”Accurate operation delay prediction for FPGA HLS using graph neural networks”, International Conference on Computer-Aided Design (ICCAD), 2020.
[6] L.-N. Pouchet. (2012) Polybench: The polyhedral benchmark suite. [Online]. Available: http://www.cs.ucla.edu/%7Epouchet/software/polybench
[7] B. Reagen, R. Adolf, Y. S. Shao, G.-Y. Wei, and D. Brooks, “Machsuite: Benchmarks for Accelerator Design and Customized Architectures,” Int’l Symp. on Workload Characterization (IISWC), 2014.
[8] Y. Hara, H. Tomiyama, S. Honda, H. Takada, and K. Ishii, “CHStone: A Benchmark Program Suite for Practical C-Based High-Level Synthesis,” Int’l Symp. on Circuits and Systems (ISCAS), 2008.
[9] Y. Zhou et al., ”Rosetta: A Realistic High-Level Synthesis Benchmark Suite for Software Programmable FPGAs”, Int’l Symp. on FieldProgrammable Gate Arrays (FPGA), 2018.
[10] B. C. Schafer and A. Mahapatra, “S2CBench: Synthesizable SystemC Benchmark Suite for High-Level Synthesis,” IEEE Embedded Systems Letters (ESL), 2014.
[11] A. B. Chowdhury et al., “OpenABC-D: A Large-Scale Dataset For Machine Learning Guided Integrated Circuit Synthesis”, arxiv, 2021.