EDA-Schema: A Graph Datamodel Schema and Open Dataset for Digital Design Automation

Pratik Shrestha, Alec Aversa, Saran Phatharodom, and Ioannis Savidis
Drexel University
Philadelphia, Pennsylvania, USA
{ps937,aja367,sp694,is338}@drexel.com

ABSTRACT

The growing complexity of very large-scale integrated (VLSI) circuits due to CMOS technology scaling has led to an increased interest in utilizing machine learning (ML) techniques for design automation. However, the lack of available datasets and established standards for the representation of datasets presents substantial challenges within the research community. Of particular concern is the lack of interoperability and comparability of ML-driven research in the design of digital circuits, which effectively limits collaboration. In this paper, EDA-schema, an open and comprehensive graph schema, is introduced to address such challenges by providing a structured framework for representing datasets for digital design automation. The schema represents the physical attributes and quality-of-results (QoR) metrics of a circuit across various stages of the physical design flow, including logical synthesis, floorplanning, placement, clock network synthesis, and global and local routing. Utilizing the Skywater 130 nm process design kit (PDK) and the OpenROAD toolset, a dataset of physical designs is generated and analyzed based on the circuits from the IWLS'05 benchmark suite. The dataset is made publicly available, anticipating contributions that further advance the field of ML-driven digital design.

CCS CONCEPTS

• **Hardware** → **Physical design (EDA)**; • **Computing methodologies** → **Machine learning**; • **Information systems** → *Graph-based database models.*

KEYWORDS

physical design, machine learning, open dataset

ACM Reference Format:
Pratik Shrestha, Alec Aversa, Saran Phatharodom, and Ioannis Savidis. 2024. EDA-Schema: A Graph Datamodel Schema and Open Dataset for Digital Design Automation. In *Great Lakes Symposium on VLSI 2024 (GLSVLSI '24), June 12–14, 2024, Clearwater, FL, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3649476.3658718

1 INTRODUCTION

With the continuous scaling of CMOS technology, the design complexity of integrated circuits (IC) has significantly increased, posing challenges for traditional design methodologies. To address such challenges, researchers have turned to machine learning (ML) techniques, leveraging the ability of ML algorithms to extract meaningful patterns and make informed predictions from large-scale datasets [1, 2]. However, the absence of easily accessible and curated circuit datasets as well as a universally accepted schema for organizing and sharing such datasets has resulted in a range of challenges. The variation in technology nodes and design tools, the lack of open tools and nodes, the time consuming scripting and parsing of design reports, the arbitrary pre-processing methods, the lack of meaningful benchmark comparisons, and, as previously mentioned, the absence of a standardized dataset limit the unified methodologies needed to formulate and evaluate machine learning problems. Addressing such challenges necessitates a comprehensive dataset and a precise schema to represent data from physical design. Therefore, this paper introduces EDA-schema, an open-source and comprehensive graph schema, as well as an open-source dataset generated from the IWLS'05 circuit benchmark suite.

While OpenABC-D [3] is a dataset curated by synthesizing hardware IP with the open-source EDA tool Yosys-abc, the primary focus of the tool is logic synthesis optimization. Similarly, in METRICS2.1 [4], standards and metrics for RTL-to-GDS design tools and flows are proposed. However, the tool only tracks a limited number of data points and does not consider the netlist and corresponding substructures, only capturing a subset of design metrics and ignoring many others considered for the ML driven design automation.

This paper presents a property graph schema developed to represent digital circuit data and the associated attributes. The contributions of the paper include:

- A standardized set of graph structures and feature sets representing a digital circuit and corresponding subcomponents,
- A dataset of physical designs generated from the IWLS'05 benchmark circuit suite [5] utilizing the open-source 130 nm Process Design Kit (PDK) provided by Skywater and the open-source toolset OpenROAD [6], and
- An analysis of the dataset of physical designs, demonstrating the completeness, reproducibility, diversity, and extensibility provided by the dataset and schema for machine learning based EDA applications.

The formatted dataset of physical designs and the source code of the datamodel schema are released on GitHub[1].

GLSVLSI '24, June 12–14, 2024, Clearwater, FL, USA

ACM ISBN 979-8-4007-0605-9/24/06
https://doi.org/10.1145/3649476.3658718

[1]A processed physical design dataset along with code to access and represent the data as EDA-schema data-model have been made available on https://github.com/drexel-ice/EDA-schema

The paper is organized as follows. Background on the EDA flow and the data generated at different stages of the automated process is provided in Section 2. The datamodel, property graph, and corresponding data model entities of EDA-schema are described in Section 3. A discussion on the open dataset, including the generation of the data, the utilized benchmark circuits, the tools employed, and the setup parameters that are used, as well as an analysis of the dataset, is provided in Section 4. The utility of the schema and open dataset for machine learning based digital circuit design is discussed in Section 5. Some concluding remarks are provided in Section 6.

2 ELECTRONIC DESIGN AUTOMATION FLOW

The automated design of a digital circuit is a comprehensive process that includes various stages that facilitate the efficient and functionally accurate implementation of an integrated circuit while meeting target power, performance, and area (PPA) specifications. PDKs, which provide technology-specific data and design guidelines, are fundamental to the design process. EDA tools utilize PDK information when executing 7design and optimization algorithms to generate circuit topologies and layouts, while calculating PPA metrics from the generated circuits to meet target specifications. The overall EDA design flow, as shown in Fig. 1, is divided into five critical stages: logical synthesis, floorplanning, placement, clock network synthesis, and routing. Each stage depends on design constraints and outputs generated from the preceding stage. Constraints are often stage-specific, further impacting the design of the circuit. Various data formats are utilized to represent the different aspects of the circuit, often adhering to standard data structures. Typically, the completion of each design phase results in the generation of the following files.

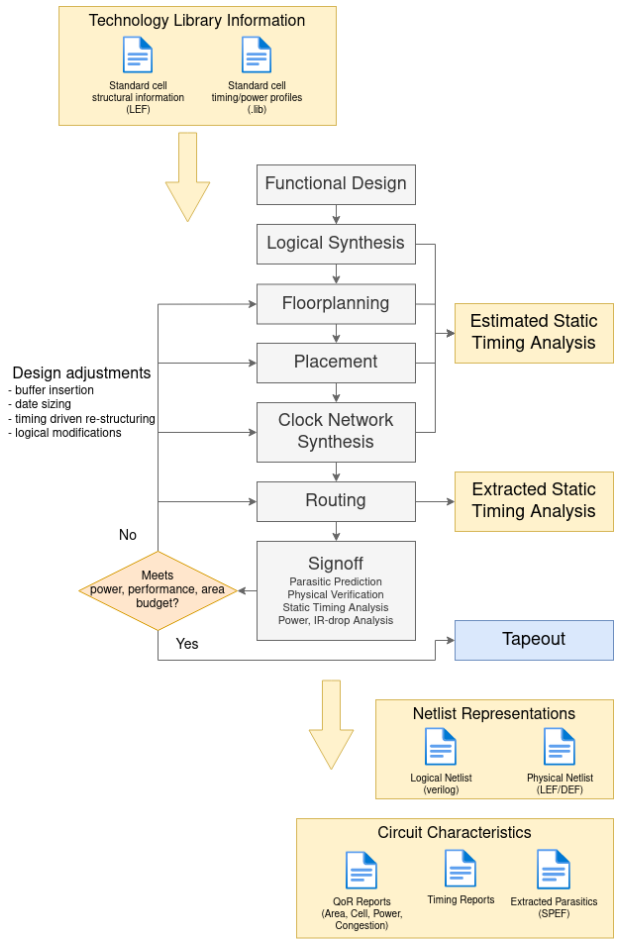


Figure 1: Block representation of the physical design automation flow.

(1) **Logical Netlist:** Technology-specific verilog or VHDL files that utilize standard cells defined in the liberty files (.lib) of a process technology node and are logical representations of the netlist. Liberty files provide detailed timing and electrical properties of digital standard cells for a specific PDK, which are utilized to optimize a circuit and verify that design constraints are met.
(2) **Physical Netlist:** Library Exchange Format (LEF) and Design Exchange Format (DEF) files are technology-specific netlist representations that include physical coordinates of input/output pins, standard cell placements, and interconnect routing.
(3) **Interconnect Parasitics:** Standard Parasitic Extraction Format (SPEF) files [7] contain information on the parasitic impedance of a circuit; specifically, the resistance (R), inductance (L), and capacitance (C) of the devices and interconnects of a circuit layout. Interconnects are segmented into sections and RLC values are reported for each section, which enables accurate analysis of signal timing and power consumption. While an estimation of the parasitic impedance can be obtained after placement, the true parasitic impedance of the interconnect is determined after completion of routing.
(4) **Quality-of-Results (QoR) reports:** QoR reports track design quality metrics at each stage of the physical design flow, with reports after logical synthesis providing initial, but inaccurate, estimates of power consumption, occupied area, and timing. More accurate metric scores are obtained in subsequent stages of the design flow.
(5) **Timing reports:** Timing reports are generated using a static timing analysis (STA) tool. The report includes information on timing paths, arrival time, and required time. Multiple timing paths are extracted from a single circuit.

3 EDA DATAMODEL SCHEMA

Two types of data results are returned by the executed algorithms of the physical design flow, which are utilized to assess metrics evaluated on each circuit: (1) structural data of the circuit (.lib, LEF, DEF, and SPEF files), and (2) QoR metrics (power, timing, and area reports). EDA-schema, a property graph data model, represents circuits and subcomponents as directed graphs. Nodes in the graph represent both the interconnects and electronic components that include gates and passive devices, while edges represent connections between components. The netlist, clock network, interconnect, and timing path are primary graph entities of the circuit and schema, with each described in Sections 3.1, 3.2, 3.3, and 3.4, respectively. Each node and edge in the graph contains structural features, QoR metrics, and derived features. The features of sub-components of a graph that are not graphs themselves are stored as tabular entities

associated with the primary graph entities. An entity relationship diagram for the schema is shown in Fig. 2, and entity properties are listed in Table 1.

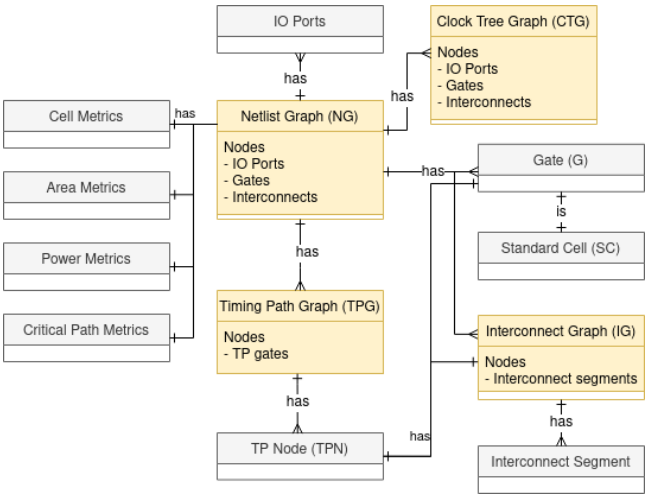


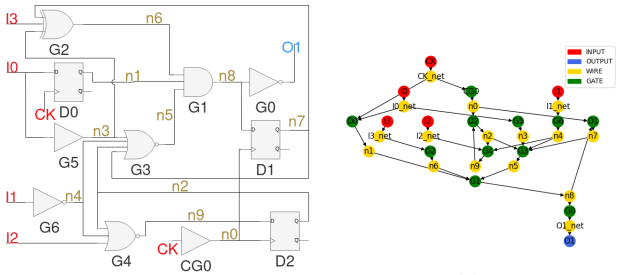
Figure 2: EDA-schema entity relationship diagram. Graph entities are highlighted in yellow, while the remaining are tabular entities.

3.1 Netlist

Verilog and DEF files provide structural details on logic gate functionality, input/output port properties, coordinates of the placed devices, and interconnect routing paths. The extracted information is mapped to a heterogeneous netlist graph, denoted as $NG = (V, E)$, where nodes $v \in V$ represent input/output ports (IO), gates (G), and interconnects (I), and edges $e \in E$ denote connections between devices and components of the circuit. In addition to the structural and topological details obtained from DEF files, the netlist entity includes density features as attributes, which are calculated as

- pin density = $\frac{number\ of\ pins\ in\ circuit}{area\ of\ circuit}$,
- cell density = $\frac{number\ of\ gates\ in\ circuit}{area\ of\ circuit}$, and
- net density = $\frac{number\ of\ interconnects\ in\ circuit}{area\ of\ circuit}$.

As an example, the netlist graph for the circuit depicted in Fig. 3a is shown in Fig. 3b.



(a) (b)

Figure 3: Sample benchmark circuit represented as (a) schematic diagram and (b) a netlist graph NG.

Table 1: EDA-schema datamodel features.

	Feature	Data Type	Unit	Source
Netlist Graph (NG)	width	num	µm	DEF file
	height	num	µm	
	no. of inputs	num	#	
	no. of outputs	num	#	
	pin density	num	μm^{-2}	Calculated
	cell density	num	μm^{-2}	
	net density	num	μm^{-2}	
Cell Metrics (CM)	no. of combinational cell	num	#	QoR reports
	no. of sequential cell	num	#	
	no. of buffer cell	num	#	
	no. of inverter cell	num	#	
	no. of marcos	num	#	
	no. of total cell	num	#	
Area Metrics (AM)	combinational cell area	num	μm^2	
	sequential cell area	num	μm^2	
	buffer cell area	num	μm^2	
	inverter cell area	num	μm^2	
	marco area	num	μm^2	
	cell area	num	μm^2	
	net area	num	μm^2	
	total area	num	μm^2	
Power Metrics (PM)	combinational power	num	µW	
	sequential power	num	µW	
	macro power	num	µW	
	internal power	num	µW	
	switching power	num	µW	
	leakage power	num	µW	
	total power	num	µW	
Critical Path Metric (CPM)	start point	str		STA reports
	end point	str		
	path type	str		
	no. of slack violations	num	#	
	worst slack	num	ns	
	worst negative slack	num	ns	
	total negative slack	num	ns	
	no. of hold violations	num	#	
	worst hold violation	num	ns	
	total hold violation	num	ns	
Standard Cell (SC)	functionality name	str		LEF file
	width	num	µm	
	height	num	µm	
	no. of input pins	num	#	
	no. of output pins	num	#	
	input capacitance (min, max, avg)	num	pF	Liberty file
	output capacitance (min, max, avg)	num	pF	
	leakage power (min, max, provided)	num	nW	
	is inverter gate	boolean		Calculated
	is buffer gate	boolean		
	is sequential gate	boolean		
	drive strength	num		
Gate (G)	name	str		DEF file
	standard cell	str		
	no. of fan-ins	num	#	
	no. of fan-outs	num	#	
	gate co-ordinates	(num, num)		
IO Port (IO)	name	str		DEF file
	direction	str		
	co-ordinate	(num, num)		
	capacitance	num	fF	
Interconnect Graph (IG)	name	str		DEF file
	no. of intputs	num	#	
	no. of outputs	num	#	
	bounding box co-ordinates (xmin, ymin, xmax, ymax)	(num, num, num, num)	#	
	half perimeter width length	num	µm	Calculated
	RUDY	num		
	resistance	num	fF	SPEF file
	capacitance	num	fF	
Interconnect Segment (IS)	name	str		DEF file
	length	num	µm	
	start point co-ordinate	(num, num)		
	end point co-ordinate	(num, num)		
	midpoint	(num, num)		Calculated
	RUDY	num		
	resistance	num	fF	SPEF file
	capacitance	num	fF	
Clock Tree Graph (CG)	no. of buffers	num	#	DEF file
	no. of inverters	num	#	
	no. of sinks	num	#	
Timing Path Graph (TPG)	start point	str		STA reports
	end point	str		
	path type	str		
	arrival time	num	ns	
	required time	num	ns	
	slack	num	ns	
	no. of gates	num	#	
	is critical path	boolean		
Timing Path Node (TPN)	cell delay	num	ns	STA reports
	arrival time	num	ns	
	slew	num	ns	
	is rising transition	boolean		
	is falling transition	boolean		

Gate nodes extend standard cells (*SC*) with design-specific structural and positional attributes. LEF files provide structural features, while the liberty files describe the timing and power profiles of the standard cells. A range of capacitance and power numbers are provided in a .lib file for each gate, with the value contingent upon the logical conditions of the applied input and the pins of the standard cell. The information provided by the various files is mapped to a standard cell entity.

Quality metrics that extend the netlist graph, including the cell count, area, and power consumption, are extracted from QoR reports and are listed in Table 1 as Cell Metrics (*CM*), Area Metrics (*AM*), and Power Metrics (*PM*), respectively. Critical path information from the timing reports, provided as the Critical Path Metric (*CPM*), further extend the attributes of the netlist.

3.2 Clock Network

Clock networks are substructures derived from the netlist. Clock network graphs (*CNG*), similar to netlist graphs, are comprised of nodes $v \in V$ that represent input/output ports (*IO*), gates (*G*), and interconnects (*I*), and edges $e \in E$ that denote connections between components. Notably, the attributes of the clock network entity are distinct and specifically pertain to clock-related information. The clock network graph for the circuit depicted in Fig. 3a is shown in Fig. 4.

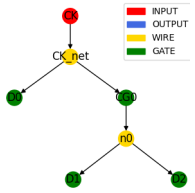
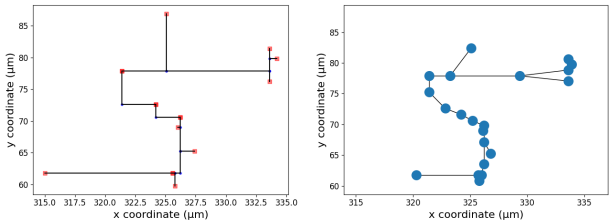


Figure 4: Clock network graph *CNG* extracted from the netlist graph *NG*.

3.3 Interconnect

The structural properties of an interconnect entity (*I*) are extracted from the DEF files, while the parasitic impedance is extracted from the SPEF files. Physical representations are also used to calculate derivative properties, including the interconnect wire length, half perimeter wire length (HPWL), and Rectangular Uniform wire DensitY (RUDY) [8], which is a metric used to evaluate routing resource utilization and congestion.

To comprehensively analyze the physical characteristics of the circuit interconnects, detailed properties of estimated interconnect segments and the corresponding spatial relationships are extracted from the netlist graph. While the true physical characteristics of the interconnect only become available after routing, the insertion point and the destination point of the interconnects are available after placement, which allows for an estimate of the routing paths of a circuit post placement. The physical representation of the estimated interconnect paths is extracted from the DEF file generated after placement, with an example of an interconnect path shown in Fig. 5a. The properties of the nets are parsed and mapped on to an interconnect graph $IG = (V, E)$, where nodes $v \in V$ represent interconnect segments and edges $e \in E$ represent connections between two or more interconnect segments. The interconnect graph captures the spatial layout and connections of the circuit. An example of an interconnect graph is shown in Fig. 5b, with each node of the graph populated with the feature set listed in Table 1. The features of the parasitic impedance and the impedance of the interconnect segments are obtained from the SPEF file.



(a) (b)

Figure 5: Geometrical representation of the (a) estimated route based on data from the SPEF file and (b) interconnect spatial graph based on the estimated routing provided after placement.

3.4 Timing Path

Static timing analysis (STA) is a crucial step that ensures timing constraints are met by identifying critical signal paths (timing paths), and computing the arrival and required times of the identified paths. Assuring the identified critical paths meet timing constraints is essential for the overall optimization of the performance and reliability of a circuit. Subgraphs representing timing paths, described as timing path graphs $TPG = (V, E)$, are shown in Fig. 6. The *TPG*s are derived from the netlist graph *NG*. Nodes of a timing path subgraph are either gates *G* or interconnects *I* appended with timing features extracted from the STA generated timing reports and mapped to the circuit netlist.

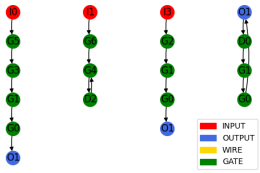


Figure 6: Timing path graphs *TPG* extracted from the netlist graph *NG* of the circuit shown in Fig. 3a.

4 OPEN DATASET

Publication or open profiling of datasets generated using commercial EDA tools and commercial PDKs is often restricted by term of use requirements. Therefore, generating open data by utilizing open-source tools and publicly available PDKs and benchmark circuits is of importance. Use of openly available resources allows for the unrestricted sharing of generated design data and reports from open-source tools, which enables broader research and collaboration. In this section, an open dataset of physical designs is described for the benchmark circuits listed in Table 2. The generation of the dataset is discussed in Section 4.1, and the analysis of the generated dataset is provided in Section 4.2.

Table 2: Summary of circuit characteristics, computational time, and memory usage of the generated dataset of IWLS'05 benchmark circuits. Large circuits with median gate count greater than 50,000 are highlighted in red.

							Time of Computation						Memory	
Circuits	**No. of inputs**	**No. of outputs**	**Median no. of gates**	**No. of flip-flops**	**Median no. of nets**	**Median no. of timing paths**	**Logical Synthesis**	**Floorplan**	**Placement**	**CTS**	**Routing**	**Total**	**Raw data**	**Processed data**
ac97_ctrl	84	48	13124	2211	6153	2827	11m	7m	23m	8m	8m	58m	12G	1.435G
aes_core	259	129	20884	530	13673	1195	33m	8m	27m	8m	8m	1h 25m	11G	1.166G
des3_area	240	64	3977	64	2347	116	5m	5m	17m	5m	5m	40m	2.9G	0.298G
des3_perf	234	64	100455	8808	56818	16287	1h 39m	20m	1h 53m	27m	25m	4h 46m	85G	8.221G
ethernet	96	115	69541	10544	35649	20220	35m	16m	1h 18m	29m	28m	3h 8m	87G	8.907G
fpu	70	40	23228	643	14370	1669	2h 59m	9m	33m	9m	8m	4h 0m	14G	9.215G
i2c	19	14	1053	129	485	196	3m	5m	15m	5m	5m	35m	1.9G	0.054G
mem_ctrl	115	152	9431	1051	5081	3422	13m	6m	21m	6m	6m	55m	7.6G	1.676G
pci	162	207	21612	3220	10769	3770	15m	8m	33m	8m	8m	1h 14m	15G	1.543G
sasc	16	12	792	118	340	180	2m	5m	15m	5m	5m	34m	1.8G	0.040G
simple_spi	16	12	1007	131	473	5311	2m	5m	15m	5m	5m	35m	1.9G	1.337G
spi	47	45	3091	229	1631	493	4m	5m	16m	5m	5m	38m	3G	0.187G
ss_pcm	19	9	640	87	294	109	2m	5m	16m	5m	5m	34m	1.6G	0.029G
systemcaes	260	129	9565	670	5630	1436	7m	6m	19m	6m	6m	47m	7G	0.702G
systemcdes	132	65	2722	190	1343	446	4m	5m	16m	5m	5m	37m	2.6G	0.150G
tv80	14	32	7497	364	4620	3008	17m	6m	19m	6m	6m	55m	5.1G	1.052G
usb_funct	128	121	14792	1740	7963	159	11m	6m	21m	6m	6m	53m	6.1G	0.487G
usb_phy	15	18	674	108	275	141	2m	5m	15m	5m	5m	34m	1.7G	0.033G
vga_lcd	89	109	106047	17059	52511	42493	2h 26m	24m	2h 29m	1h 3m	1h 2m	7h 25m	160G	17.744G
wb_dma	217	215	4241	521	1996	1136	21m	6m	16m	5m	6m	56m	4.1G	0.328G
Total							**10h 20m**	**2h 52m**	**11h 28m**	**3h 49m**	**3h 47m**	**32h 20m**	**428G**	**82.836G**

4.1 Dataset Generation

A comprehensive dataset for physical design is comprised of four key components: (1) the circuits used to create the dataset, (2) the PDK of the fabrication process, (3) the software tools used for physical design, and (4) a specific set of configuration parameters that govern the generation of the dataset. The open dataset is generated using the IWLS'05 sequential benchmark circuits [5]. A general characterization of the benchmark circuits is provided through attributes listed in Table 2. The PDK for the Skywater 130 nm technology node is utilized on the OpenROAD toolset, which is used for physical design and verification. The generated raw dataset is curated and formatted utilizing the EDA-schema described in Section 3. Multiple layouts are generated for each benchmark circuit using the parameterized design configurations listed in Table 3, which results in 960 physical designs, with 48 iterations of each of the 20 benchmark circuits. The 960 physical designs consist of a total of 20,272,194 gates (including filler cells), 11,086,480 interconnects, and 5,020,528 timing paths after routing. The dataset is generated on a server with 94 GB memory and 4 GHz CPU cores and requires a total of 32 hours and 20 minutes of compute time. The final raw dataset, which contains 428 GB of netlists and performance metric reports, is curated and compressed to 82.83 GB of .csv and .json files formatted using the EDA-schema.

4.2 Dataset Analysis

In this section, an in-depth analysis of the generated dataset is provided. The objective is to characterize key aspects of the dataset while also providing insights into the generated circuits.

4.2.1 Power, Performance, and Area Metrics. The distribution of the total area, total power, and worst case arrival time for each circuit of the dataset after completion of routing is shown in Fig. 7a. The circuits exhibit a large range in the area occupied, which spans from 5,748 μm^2 to 1,141,293 μm^2, and the total power consumed, which spans from 0.761 nW to 251 µW. *Ethernet*, *des3_perf*, and *vga_lcd* occupy a larger area as compared to the rest of the circuits, with an average area of 999,889 μm^2 amongst the three as compared to an average area of 81,421 μm^2 for the entire dataset. In addition, the interquartile range of the total power consumption of the 48 instances of the *fpu* is greater than that of the other circuits, reflecting a higher degree of variation in the power consumed. Although comparable to the rest of the circuits of the dataset based on the area occupied, the *fpu* on average consumes a significant amount of power (approximately 68x times greater than the next largest power consuming circuit *des3_perf*) and provides significantly worse performance (approximately 5x times longer worst case arrival time than the next worst circuit *vga_lcd*). The larger power consumption and the longer worse case arrival times of the *fpu* are a consequence of the functional complexity of the circuit, as evidenced by the large logical depth of the timing paths of the combinational logic, as shown in Fig. 8.

The distribution of the post-routing total area, total power, and worst case arrival time is shown in Fig. 7b for each circuit excluding the *fpu* and the comparatively larger circuits of *ethernet*, *des3_perf*,

Table 3: Constraints and parameters used to generate the dataset.

Parameters	**Values or Ranges**	**# of Samples**
Clock periods (ns)	{0.5, 1, 2, 5}	4
Aspect ratio	{0.5, 0.75, 1}	3
Max utilization	{0.3, 0.5}	2
Max skew (ns)	[0.01 - 0.2]	2
Max fanout	[50 - 250]	2
Max clock network capacitance (pF)	[0.05 - 0.3]	2
Max latency (ns)	[0 - 1]	2
Total circuits per design		**48**
Overall circuits in dataset		**48 * 20 = 960**
No. of gates in dataset		**20,373,194**
No. of nets in dataset		**11,086,480**
No. of timing paths in dataset		**5,020,528**

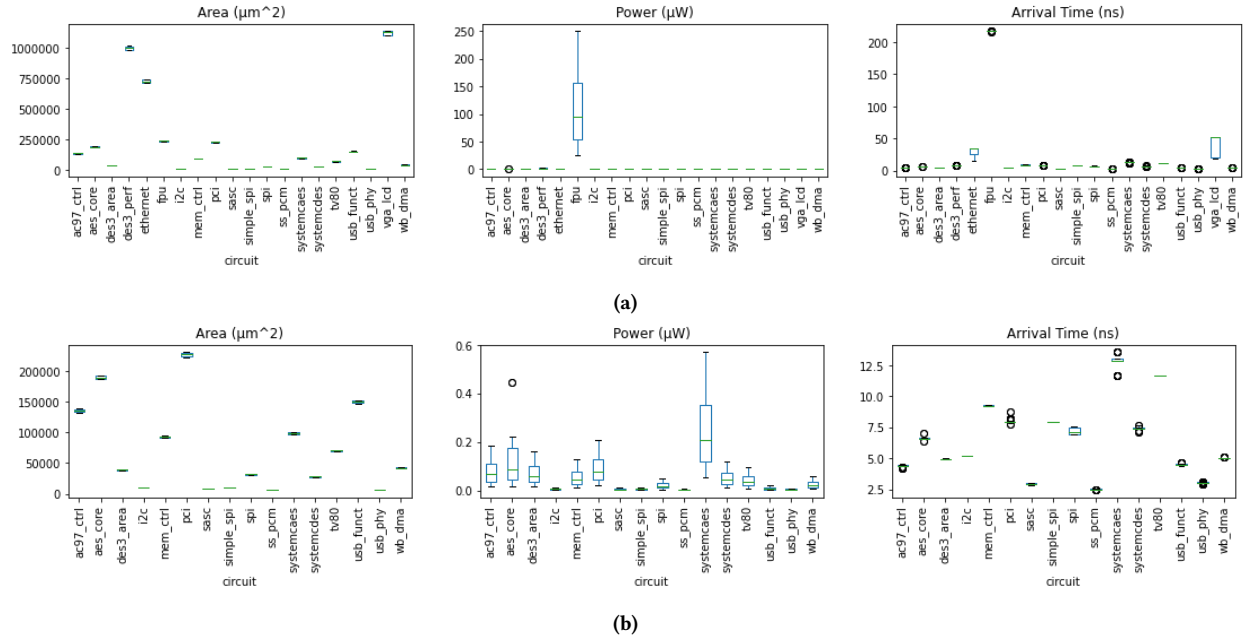


Figure 7: Distribution of post-routing total area, total power, and worst case arrival time of (a) all circuits across the dataset and (b) all circuits across the dataset excluding the *fpu*, *ethernet*, *des3_perf*, and *vga_lcd*.

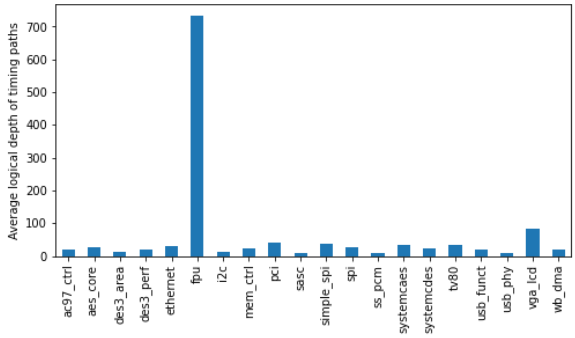


Figure 8: Average logical depth of the timing paths of the combinational logic of each circuit in the dataset.

and *vga_lcd*. While the excluded circuits offer valuable insights into 'corner cases' of generated circuit topologies, the unique characteristics each brings skews the general properties of the dataset. The filtered dataset provides a more targeted set of circuit characteristics. The distribution of the total area, total power consumption, and worst case arrival time post-completion of routing for the complete dataset with and without the *fpu*, *ethernet*, *des3_perf*, and *vga_lcd* is shown in Fig. 9a and Fig. 9b, respectively. The results indicate that the entire dataset includes considerable skew in area, power, and arrival time. By excluding the selected circuits, the prevalence of outliers is noticeably reduced, which allows for a detailed analysis of the fundamental characteristics of the dataset. The comparison between the entire dataset and the dataset with excluded circuits highlights the effect of 'outlier' circuits on the power, performance, and area characteristics of the dataset and underscores the significance of accounting for each factor when designing and analyzing a circuit with ML algorithms.

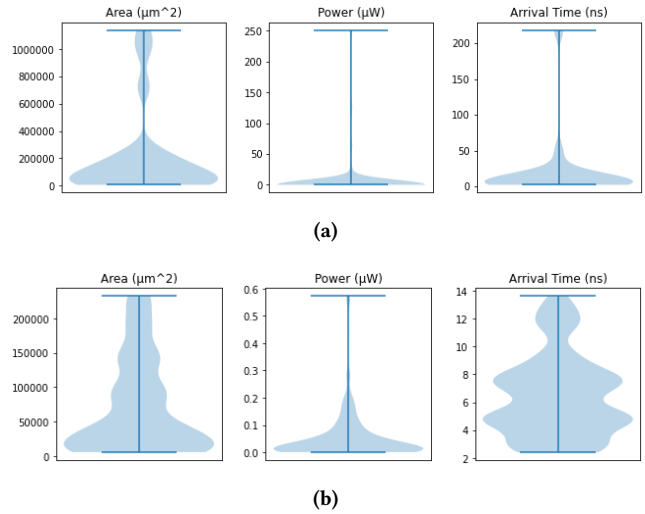


Figure 9: Distribution of the post-routing total area, total power, and worst case arrival time of (a) the complete dataset and (b) the complete dataset with the exclusion of the *fpu*, *ethernet*, *des3_perf*, and *vga_lcd* circuits.

The variation in circuit characteristics highlights the design trade-offs and optimization strategies needed across the different circuits. Circuit variability is essential for the creation of a comprehensive and generalizable dataset and to assure that a wide range of real-world scenarios are addressed. Providing a comprehensive coverage ensures that the dataset effectively represents the broad range of conditions encountered in circuit design and performance analysis.

4.2.2 Timing Analysis. The dataset includes circuits that both comply and violate the targeted timing budget, with 1,418,143 timing paths that meet the timing requirements and 3,602,385 timing paths that do not. The number of timing paths that meet and violate the slack requirements across the different circuits is listed in Table 4. Of the 960 circuits generated using the design constraints and parameters listed in Table 3, 73 circuits include timing paths with all slack requirements met. The comparison between circuits with at least one timing path slack violation and circuits with all timing path slack requirements met is shown in Fig. 10. Note that, 71.76% of timing paths violate the slack requirements, while only 7.81% of the circuits include timing paths with no slack violations. However, slack is a comparative metric relative to a target budget. A slack violation does not necessarily indicate a non-functional circuit. The violation only indicates that the circuit does not meet the predefined timing constraints or the targeted clock frequency. Consequently, the circuit potentially functions correctly, but at a reduced clock frequency. The inclusion of timing paths that both meet and violate the timing budget provides a comprehensive range of timing path behaviors and enhances the adaptability and usefulness of the dataset for various analytical and design-oriented use cases.

Table 4: Analysis of timing paths that meet and violate the slack requirements of the circuit.

Circuit	Slack met	Slack violated	Total	Percentage of slack met
ac97_ctrl	61979	75801	137780	44.98%
aes_core	15286	41248	56534	27.04%
des3_area	1696	3830	5526	30.69%
des3_perf	391064	384824	775888	50.40%
ethernet	200925	777957	978882	20.53%
fpu	13746	64858	78604	17.49%
i2c	3734	5650	9384	39.79%
mem_ctrl	21878	151978	173856	12.58%
pci	43691	141497	185188	23.59%
sasc	4285	3993	8278	51.76%
simple_spi	214728	40146	254874	84.25%
spi	12978	10680	23658	54.86%
ss_pcm	3064	2342	5406	56.68%
systemcaes	9682	62554	72236	13.40%
systemcdes	9865	10553	20418	48.32%
tv80	47996	97116	145112	33.08%
usb_funct	4725	2861	7586	62.29%
usb_phy	3813	2941	6754	56.46%
vga_lcd	330108	1687538	2017646	16.36%
wb_dma	22900	34018	56918	40.23%
Total	**1418143**	**3602385**	**5020528**	**28.24%**

Figure 10: Analysis of circuits meeting and violating the timing slack requirement.

4.2.3 Parameter Analysis. An analysis of the effects of clock period, core aspect ratio, and core utilization on key quality metrics (total area, total power, and worst case arrival time) is performed. The characterization of the effects of the parameters on the quality metrics is utilized to justify the inclusion of a given parameter as a component of the dataset.

Distributions of the total area, total power, and worst case arrival time as a function of design parameters for the *ac97_ctrl* circuit are shown in Fig. 11. Results indicate that changes in clock period, core aspect ratio, and core utilization lead to variations in the total area, total power, and worst case arrival time. The trend lines mark the mean of the total area, total power, and worst case arrival time of the circuit in relation to variations in the design parameters. Modifications in design parameters lead to observable shifts in mean values, which indicates an influence on the quality metrics of the circuit. The variation in mean values confirm the effect of a given design parameter on evaluated metrics.

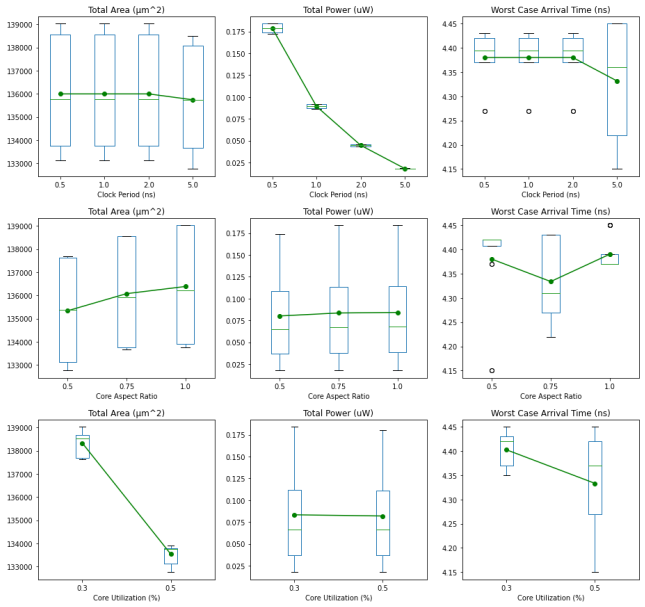


Figure 11: Distribution of post-routing quality of results metrics for total area, total power consumption, and worst case arrival time when considering design parameters of clock period, core aspect ratio, and core utilization.

To quantitatively characterize the relationship between the design parameters and the quality of results metrics for each circuit, a correlation analysis is performed between the design parameters used to generate each circuit and the mean score of each evaluated quality metric for the generated designs. The Pearson's correlation coefficient r is computed for each parameter-metric pair to determine the strength and direction of the corresponding linear relationships. The Pearson correlation coefficient is given by

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \quad (1)$$

where x_i and y_i are the values of the two variables being compared, $\bar{x}$ and $\bar{y}$ are the mean values of each variable, and n is the number of observations.

The correlation between the design parameters and the quality metrics for all circuits of the dataset is shown as violin plots in Fig. 12. The analysis of the spread and the shape of the plot for each parameter-metric pair allows for an assessment of the impact of each design parameter on each metric. The skew of the correlation coefficient towards either 1 or -1 indicates, respectively, a significant positive or negative effect on a metric by a given design parameter. The large skew towards the extreme ends of a given metric, instead of around zero, underscores the critical effect each parameter has on the resulting circuit. For example, the total area of a circuit is positively correlated with the aspect ratio and negatively correlated with the target clock period, indicating that both parameters substantially affect the resulting circuit topologies. The core utilization parameter results in a broad range of correlations, with significant skew towards both -1 and 1, while also resulting in values around zero, which suggests a varying impact on circuit PPA. Therefore, some design instances are sensitive to changes in core utilization, while others are not.

The dataset also includes circuits that are unaffected by changes in design parameters. For example, variations in design parameters do not change the critical timing paths of the *simple_spi* circuit, which indicates that no effect is observed on the worst case arrival time. Circuits that exhibit insensitivity to changes in design parameters are listed in Table 5.

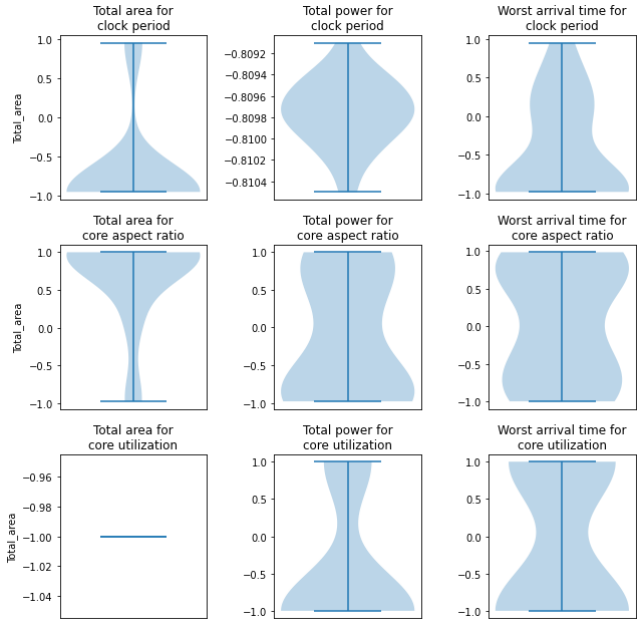


Figure 12: Distribution of post-routing Pearson's coefficient between the quality of results metrics for total area, total power, and worst case arrival time and the design parameters of clock period, core aspect ratio, and core utilization.

Table 5: Circuits that exhibit no change in quality of results metrics with changes in a design parameter.

Parameter	Metric	Circuit
Clock Period	Total Area	i2c, simple_spi, usb_phy
	Worst Arrival Time	i2c, simple_spi
Core Aspect Ratio	Total Power	fpu
	Worst Arrival Time	i2c, simple_spi
Core Utilization	Total Power	fpu
	Worst Arrival Time	i2c, simple_spi

5 EDA SCHEMA FOR MACHINE LEARNING

The primary aim of this work is to provide a foundational dataset for machine learning applications. In Section 5.1, the datasets and feature sets utilized in prior research are examined, assessing the effectiveness of the proposed data model in achieving interoperability and comparability. A baseline analysis is described in Section 5.2 of three prediction tasks that utilize metrics provided by EDA-schema and an open dataset.

5.1 Feature Set Analysis

In this section, an analysis of datasets and feature sets utilized in prior research is provided, which allows for the evaluation of the proposed data model and schema for interoperability and compatibility. An analysis of eight EDA tasks with associated features is performed as listed in Table 6, with bold features directly available within EDA-schema for ML tasks. In addition, the italicized features listed in Table 6 represent attributes that are not directly included in EDA-schema, but are derived using the provided structures and features, which demonstrates the ability to expand EDA-schema. For example, the number of nets with a fanout of ≤ 10 described in [9] is not explicitly listed in EDA-schema. However, a value is calculated by using the interconnect fanout feature. Similarly, metrics including the distance between a driver cell and target cells as well as pin and net density are computed using the structural and spatial features of the netlist and the interconnect graphs in EDA-schema.

5.2 Baseline Analysis

The quality of results metrics (PPA) evaluated on the initial physical design stage (floorplan, placement, and CTS) by the design tool are considered as the baselines used for evaluation of the developed models that predict design parameters after completion of the final design stage (routing). The models predict final stage design parameters based on data available in the currently executing physical design stage. The baseline and prediction errors are evaluated using Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), which are calculated as, respectively,

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|M_i - M_{b,i}| \text{ and} \tag{2}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}|1 - \frac{M_{b,i}}{M_i}|, \tag{3}$$

for metric M with baseline M_b, where n is the total number of samples in the overall dataset.

Table 6: Features used for machine learning driven circuit prediction problems in prior work.

Paper	Objective	Available Feature Set
[9]	placement parameter optimization	**no. of cells, no. of nets, no. of IOs,** *no. of nets with fanout ≤ 10,* **no. of flip-flops, total cell area, no. of macros, macro area**
[10]	pre-placement net length estimation	**net's driver's area, fan-in and fan-out size**
[11]	total design power prediction	**minimum slack, maximum slack, worst input transition, worst output transition, switching power of driving net, cell switching power, cell internal power, cell leakage power**
[12]	pre-routing timing prediction	**driver and sink capacitance, sink locations,** *distance between driver and the target sink,* **max driver input transition**
[13]	arrival time prediction	**standard cell functionality, logic level, no. of fan-out, gate delay, interconnect capacitance, arrival time**
[14]	interconnect parasitic prediction	**interconnect capacitance, interconnect length, interconnect position,** *pin density, net density*
[15]	wire parasitics and timing prediction	**half perimiter wire length (HPWL), no. of sinks, congestion estimates, rise transition, fall transitions, interconnect length, interconnect RC**

Table 7: Averaged quality of results metrics evaluated across all design phases. Baseline errors estimating post-routing performance using post-floorplan, post-placement, and post-CTS metrics are also listed.

	Post Floorplan	Post Placement	Post CTS	Post Routing	Floorplan to Routing		Placement to Routing		CTS to Routing	
					MAE	MAPE	MAE	MAPE	MAE	MAPE
Total Area (μm^2)	205,063.05	209,734.92	211,510.75	211,510.75	6,447.70	2.32%	1,775.82	0.71%	0.0	0.00%
Total Power (μW)	5.9822	5.9790	6.1111	5.9989	0.0168	14.20%	0.0224	13.73%	0.112	2.72%
Worst Arrival Time (ns)	20.6551	19.6098	19.8900	19.8008	1.1544	10.72%	0.2332	4.43%	0.092	1.19%

Downstream prediction of (a) total area, (b) total power, and (c) worst case arrival time is performed utilizing EDA-schema, with results compared against baseline estimates provided by OpenROAD. Results are listed in Table 7 from the evaluation of the post-routing metrics when analyzed across all design phases and is compared to the corresponding baseline errors. Machine learning models that address prediction problems are expected to outperform the baseline estimates provided by EDA tools. The estimates provided by the EDA tools improve as the design flow progresses through each physical design stage. Completion of physical placement and routing provides more accurate data, which leads to improved estimates of the quality metrics of the design. In contrast, the total power consumption predicted by the EDA tool exhibits higher absolute error in later stages of the design flow due to modifications to the logic that optimize the power, performance, and area of the circuit.

6 CONCLUSION

In this paper, EDA-schema, a property graph data model that facilitates the use of machine learning in circuit design automation is described. An open-source dataset, generated from 20 benchmark circuits using the Skywater 130 nm PDK and the OpenROAD toolset offers a large initial set of physical designs for analysis. The dataset consists of 960 circuits (20 unique designs) that include 20,272,194 gates, 11,086,480 interconnects, and 5,020,528 timing paths. The generated raw dataset, which includes 428 GB of netlists and performance metric reports, is curated into an 82.83 GB dataset using the EDA-schema and is made publicly available. The diversity and completeness of the dataset provides a valuable resource for researchers. An analysis of the adaptability and extensibility of EDA-schema is performed, demonstrating the potential use of the schema for various research tasks, which ultimately advances the field of machine learning for digital design automation.

REFERENCES

[1] G. Huang, J. Hu, Y. He, J. Liu, M. Ma, Z. Shen, J. Wu, Y. Xu, H. Zhang, and K. Zhong, "Machine learning for electronic design automation: A survey," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, Vol. 26, No. 5, pp. 1–46, Jun. 2021.

[2] P. Shrestha and I. Savidis, "EDA-ML: Graph representation learning framework for digital IC design automation," *Proceedings of the International Symposium on Quality Electronic Design (ISQED)*, pp. 241–246, Apr. 2024.

[3] A. B. Chowdhury, B. Tan, R. Karri, and S. Garg, "OpenABC-D: A large-scale dataset for machine learning guided integrated circuit synthesis," *arXiv preprint arXiv:2110.11292*, pp. 1–18, Oct. 2021.

[4] J. Jung, A. B. Kahng, S. Kim, and R. Varadarajan, "METRICS2.1 and flow tuning in the IEEE CEDA robust design flow and OpenROAD," *Proceedings of the IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–9, Nov. 2021.

[5] C. Albrecht, "IWLS 2005 benchmarks," *In International Workshop for Logic Synthesis (IWLS): http://www.iwls.org*, Jun. 2005.

[6] T. Ajayi and D. Blaauw, "OpenROAD: Toward a self-driving, open-source digital layout implementation tool chain," *Proceedings of the Government Microcircuit Applications and Critical Technology Conference*, pp. 1–6, Mar. 2019.

[7] "IEEE standard for integrated circuit (IC) open library architecture (OLA)," *IEEE Std 1481-2019 (Revision of IEEE Std 1481-2009)*, pp. 1–641, Mar. 2020.

[8] P. Spindler and F. M. Johannes, "Fast and accurate routing demand estimation for efficient routability-driven placement," *Proceedings of the IEEE Design, Automation and Test in Europe Conference and Exhibition*, pp. 1–6, Apr. 2007.

[9] A. Agnesina, K. Chang, and S. K. Lim, "VLSI placement parameter optimization using deep reinforcement learning," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–9, Nov. 2020.

[10] Z. Xie, R. Liang, X. Xu, J. Hu, Y. Duan, and Y. Chen, "Net2: A graph attention network method customized for pre-placement net length estimation," *Proceedings of the Asia and South Pacific Design Automation Conference (ASPDAC)*, pp. 671–677, Jan. 2021.

[11] Y.-C. Lu, W.-T. Chan, V. Khandelwal, and S. K. Lim, "Driving early physical synthesis exploration through end-of-flow total power prediction," *Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD (MLCAD)*, pp. 97–102, Sept. 2022.

[12] E. C. Barboza, N. Shukla, Y. Chen, and J. Hu, "Machine learning-based pre-routing timing prediction with reduced pessimism," *Proceedings of the Design Automation Conference (DAC)*, pp. 1–6, Jun. 2019.

[13] P. Shrestha, S. Phatharodom, and I. Savidis, "Graph representation learning for gate arrival time prediction," *Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD (MLCAD)*, pp. 127–133, Sept. 2022.

[14] P. Shrestha and I. Savidis, "Graph representation learning for parasitic impedance prediction of the interconnect," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, May 2023.

[15] V. A. Chhabria, W. Jiang, A. B. Kahng, and S. S. Sapatnekar, "From global route to detailed route: ML for fast and accurate wire parasitics and timing prediction," *Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD (MLCAD)*, pp. 7–14, Sept. 2022.