香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

OPTICS Lab

FIONA: Full-stack Infrastructure for Optical Neural Accelerator

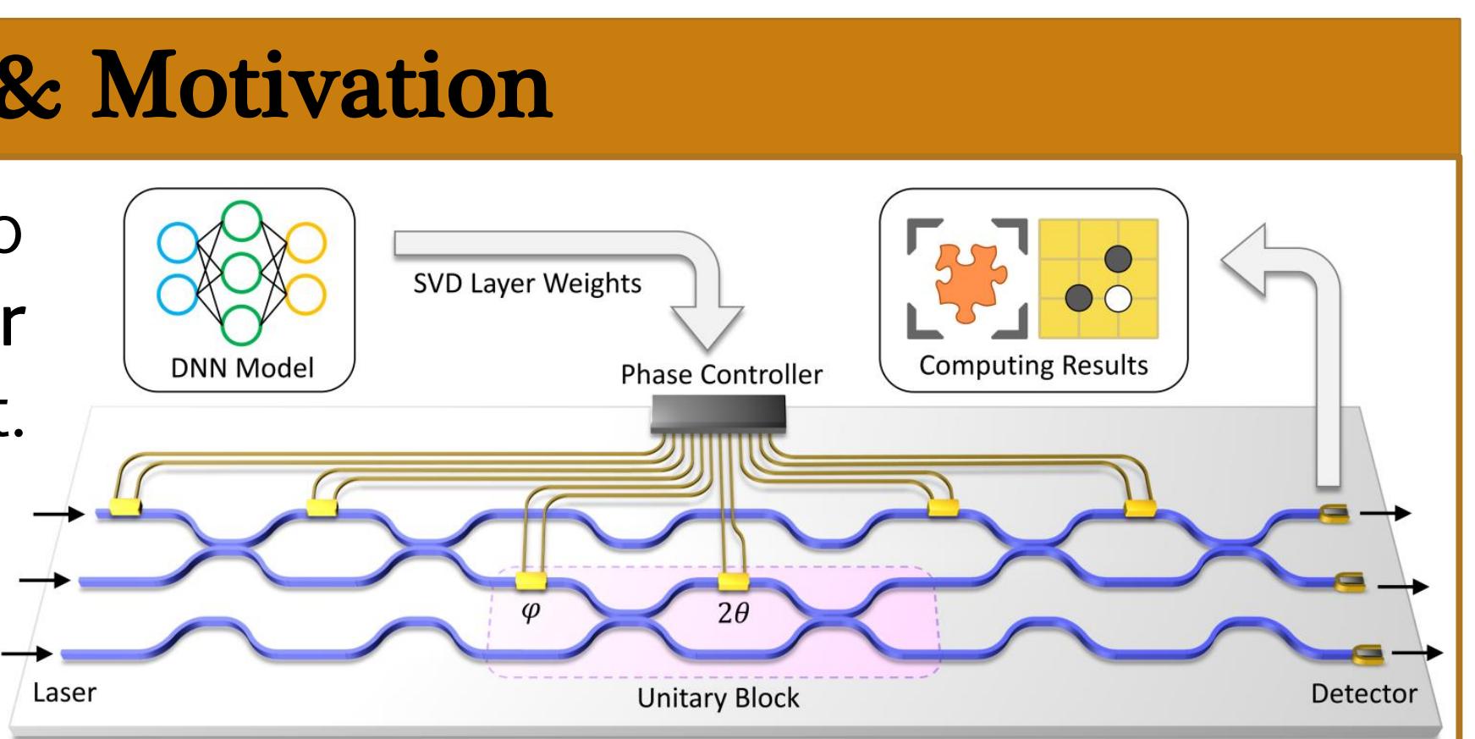
Yinyi Liu†, Bohan Hu†, Zhenguo Liu, Peiyu Chen, Linfeng Du, Jiaqi Liu, Xianbin Li,
Wei Zhang, Jiang Xu*

香港科技大學（廣州） THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

BIG DATA SYSTEM LAB

Background & Motivation

Photonic-based computing is promising to perform vector operations **faster** with **higher energy efficiency** than electronic counterpart.



- Existing Works can NOT fulfill:
  - ➢ Multi-Task Reconfigurable
  - ➢ Support Simulation for DSE
  - ➢ Physically Executable for Verification
- Proposed FIONA Framework
  - ✓ Support Functional/RTL level Simulation
  - ✓ Contain Hardware and Software Stacks
  - ✓ Design for Rapid End-to-end Prototyping

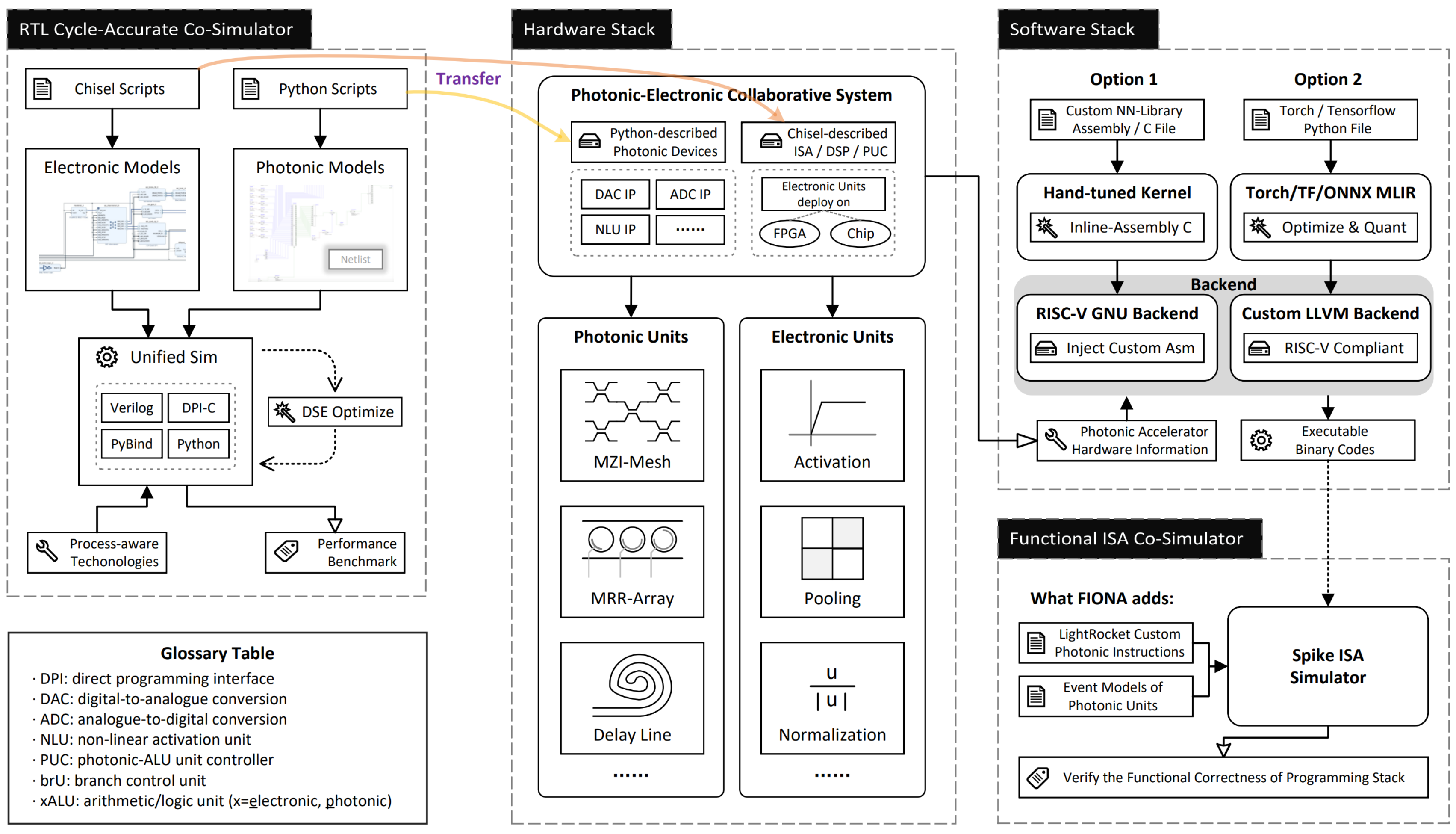
Table 1. Comparison: Progress in Photonic Computing

Related Works	Training Support	Multi-Task Reconfigurable	Simulator for DSE	Physically Executable
[1, 2]	✔		✔	
[3, 4]	✔	✔		
[5, 6]	✔	✔	✔	
[7–9]	✔	❍		✔
Ours	✔	✔	✔	✔

*Note: ❍ denotes it only supports specific DNN or no software stack.

FIONA Framework

Overview of FIONA Toolchain: from Simulation to Implementation



Glossary Table

- DPI: direct programming interface
- DAC: digital-to-analogue conversion
- ADC: analogue-to-digital conversion
- NLU: non-linear activation unit
- PUC: photonic-ALU unit controller
- brU: branch control unit
- xALU: arithmetic/logic unit (x=electronic, photonic)

*Simulation: Cross-Domain Simulation Workflow of Unified-Sim

dut.sv (module that inserts photonic unit caller)

```
import "DPI-C" function void array_handler(
    string filename, string funcname,
    input bit[W:0] array_in[A:0][B:0][C:0],
    output bit[W:0] array_out[P:0][Q:0]
);

initial begin
    bit[W:0] dac_val[A:0][B:0][C:0];
    bit[W:0] adc_val[P:0][Q:0];
    ......
    array_handler("mrr", "dotprod",
        dac_val, adc_val);
    ......
end
```

register.hh (FIONA exposed python func registry)

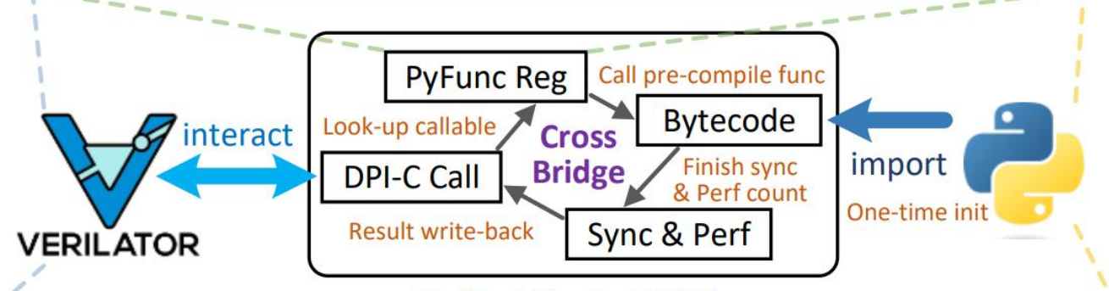
```
typedef std::pair<std::string, std::string> PyFileFunc;
typedef std::vector<PyFileFunc> PyFileFuncVec;

const PyFileFuncVec pyfilefunc_reg {
  {"mzi", "mvmul"},  // matrix-vector multiplication
  {"mrr", "dotprod"}  // dot product
};
```

mzi.py (photonic MZI model function)

```
def mvmul(arr_in, bit_width):
    np_arr = Parser(arr_in, bit_width)
    ...... # photonic model codes
    return arr_out
```

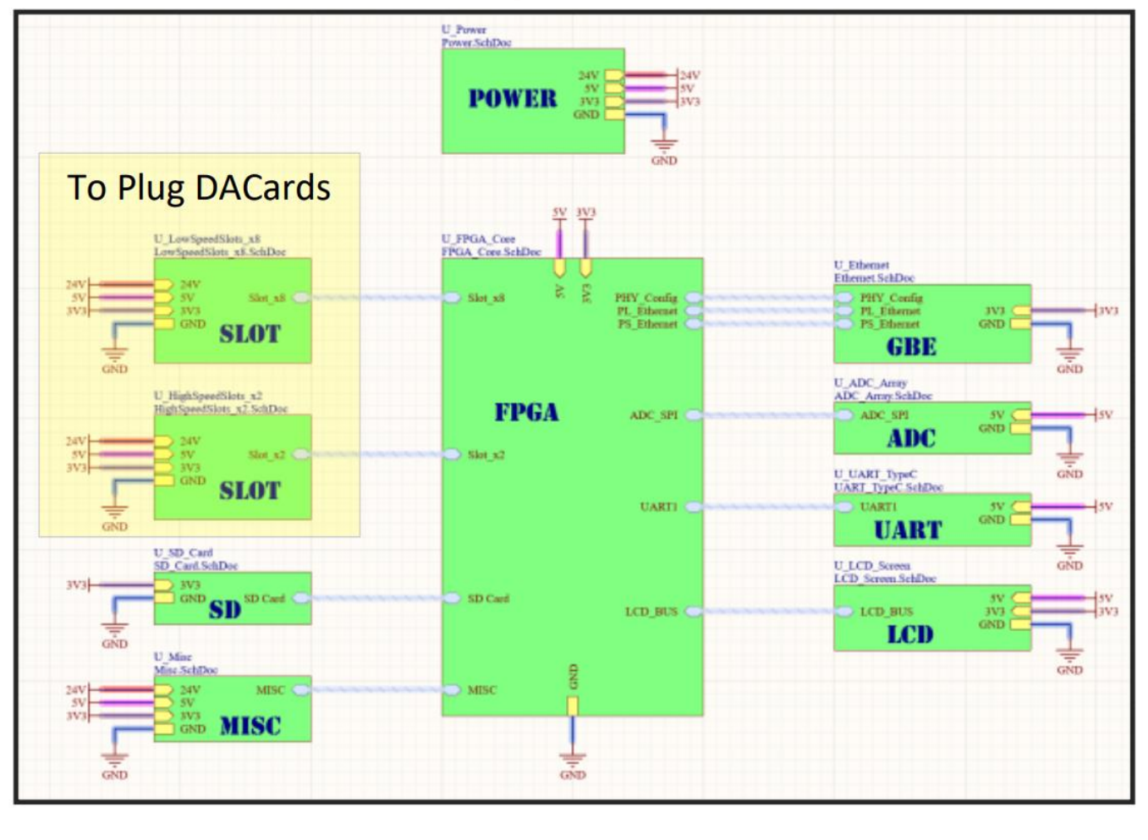
mrr.py (photonic MRR model function)

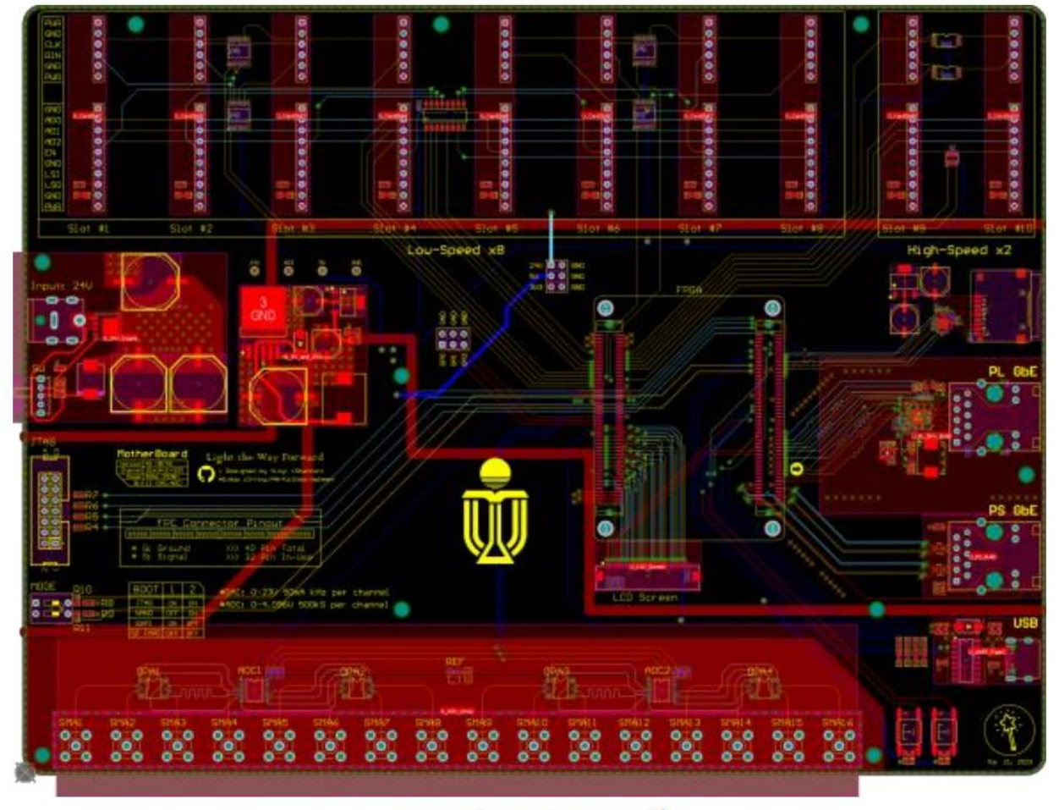
```
def dotprod(arr_in, bit_width):
    np_arr = Parser(arr_in, bit_width)
    ...... # photonic model codes
    return arr_out
```



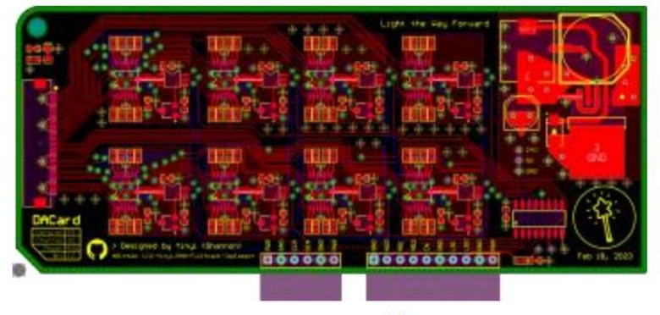
Unified Sim in FIONA

*Implementation: Transferable Rapid Prototyping Template

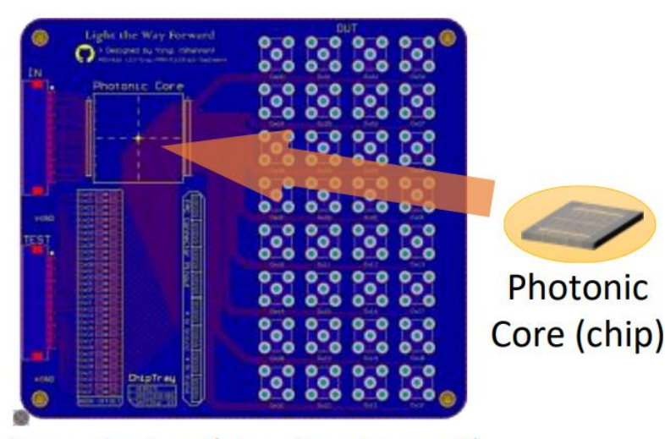




Mother Board[1]



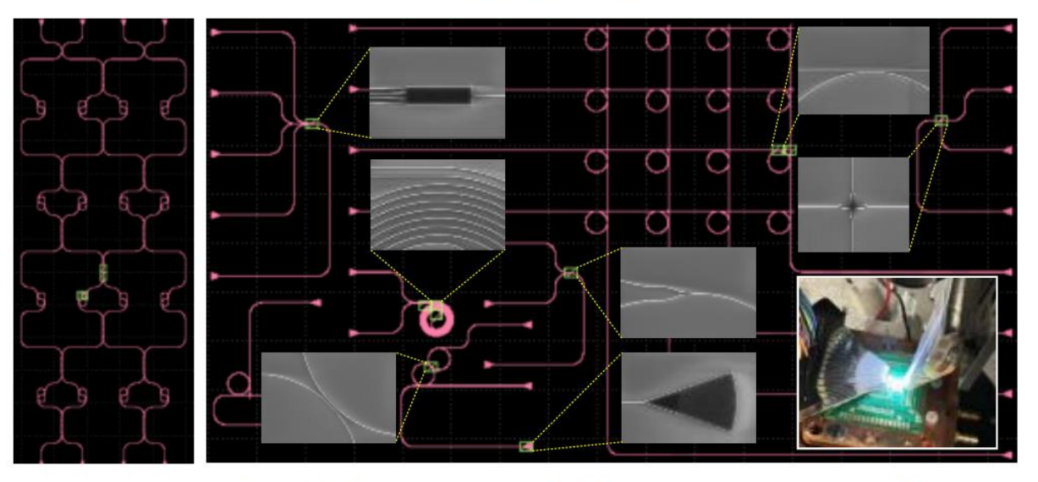
DACard[1]



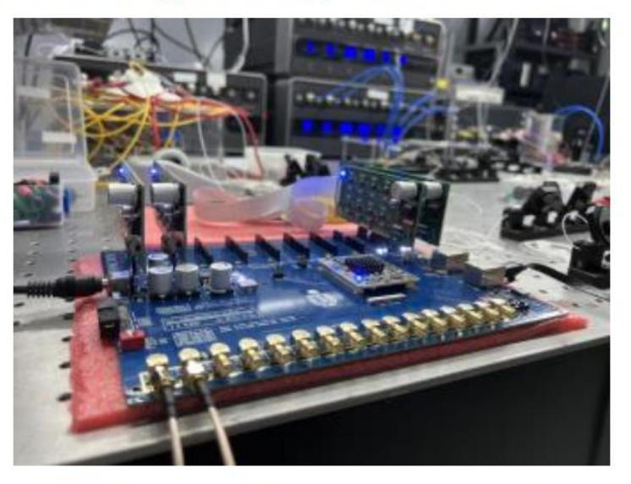
Interfacing/Testing Board[1]

Specs of Electronic Modules

- ADC: 12-bit, 0-4.096V, up to 1MS/s, x16 channels
- FPGA: XC7Z020CLG400-2, DDR3 1GB, EMMC 8GB
- GBE: giga-byte ethernet ports for data exchange
- LCD: touch screen for monitoring or interaction
- MISC: miscellaneous for LEDs, buttons, headers, etc.
- SLOT (DAC): 12-bit, 0-23V @50mA, x320 channels x10 slots, bus speed up to 40MHz



Template Collections of Photonic Modules[2]

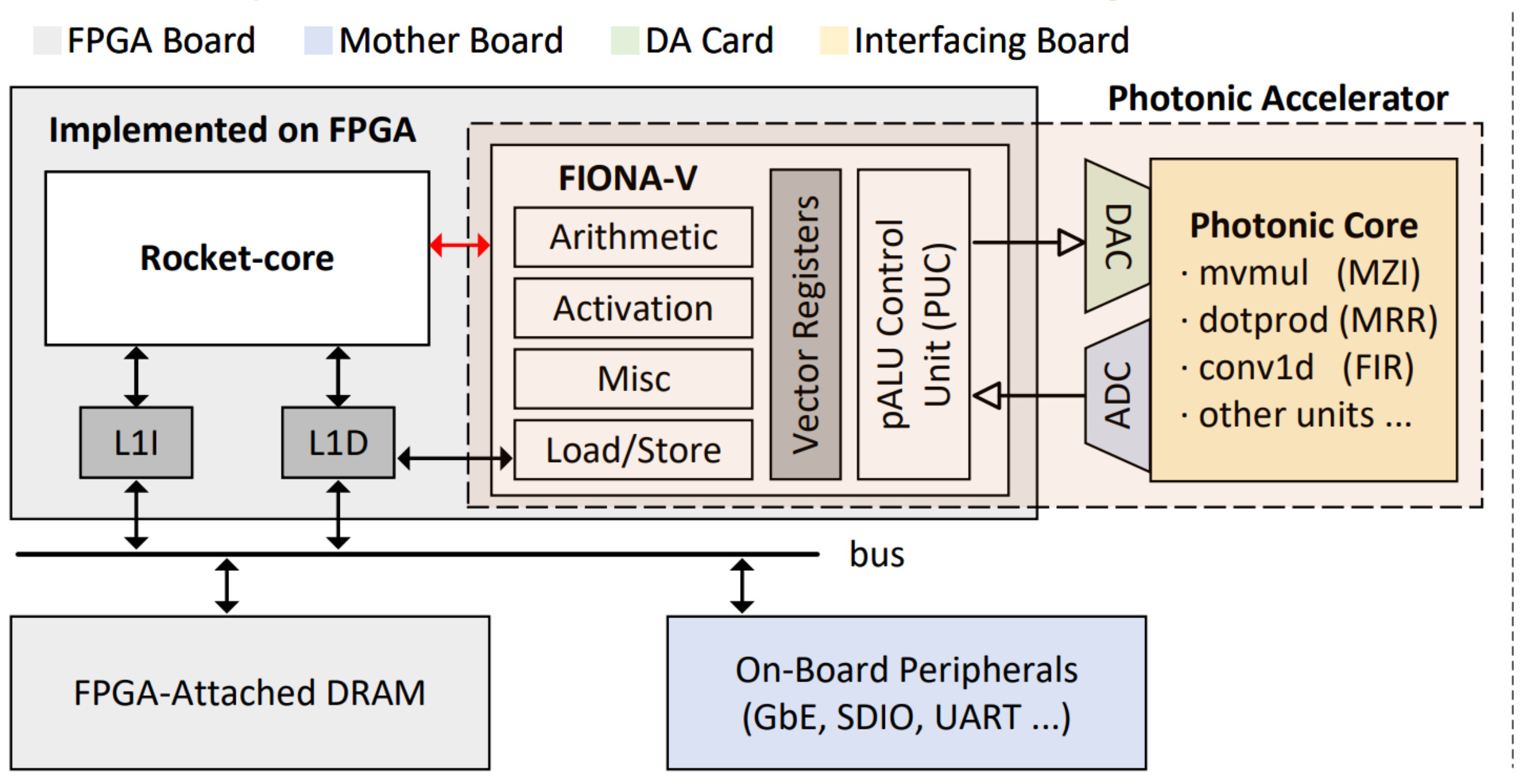


Experimental Setup using FIONA[3]

1. Schematics and PCB layouts of electronic modules [MotherBoard, DACard, Interfacing, etc.];
2. Chip layouts and scanning electron microscope (SEM) images of photonic modules;
3. Snapshot of experimental setup using FIONA rapid prototyping hardware template.

Case Study: LightRocket using FIONA Toolchain

*System Architecture & Board Assignment



*Register Map

Index	Scalar Reg
0	32 bits <zero>
1	32 bits
...	32 bits
31	32 bits

Index	Vector Reg
0	16 bits, <all zero>
1	16 bits, 32 elems
...	16 bits, 32 elems
31	16 bits, 32 elems

STRIDE	[31:0] x1 group
VLEN	[31:0] x1 group
VMASK	[31:0] x32 groups
MAT	[15:0][31:0][31:0]

LightRocket Registers

Note: FIONV-V is a baseline vector processor core based on RISC-V with custom instruction set, including photonic operations and nonlinear functions. The red line between Rocket-core and FIONA-V denotes the RoCC interfaces.

LightRocket FIONA-V Custom Instruction Set Architecture

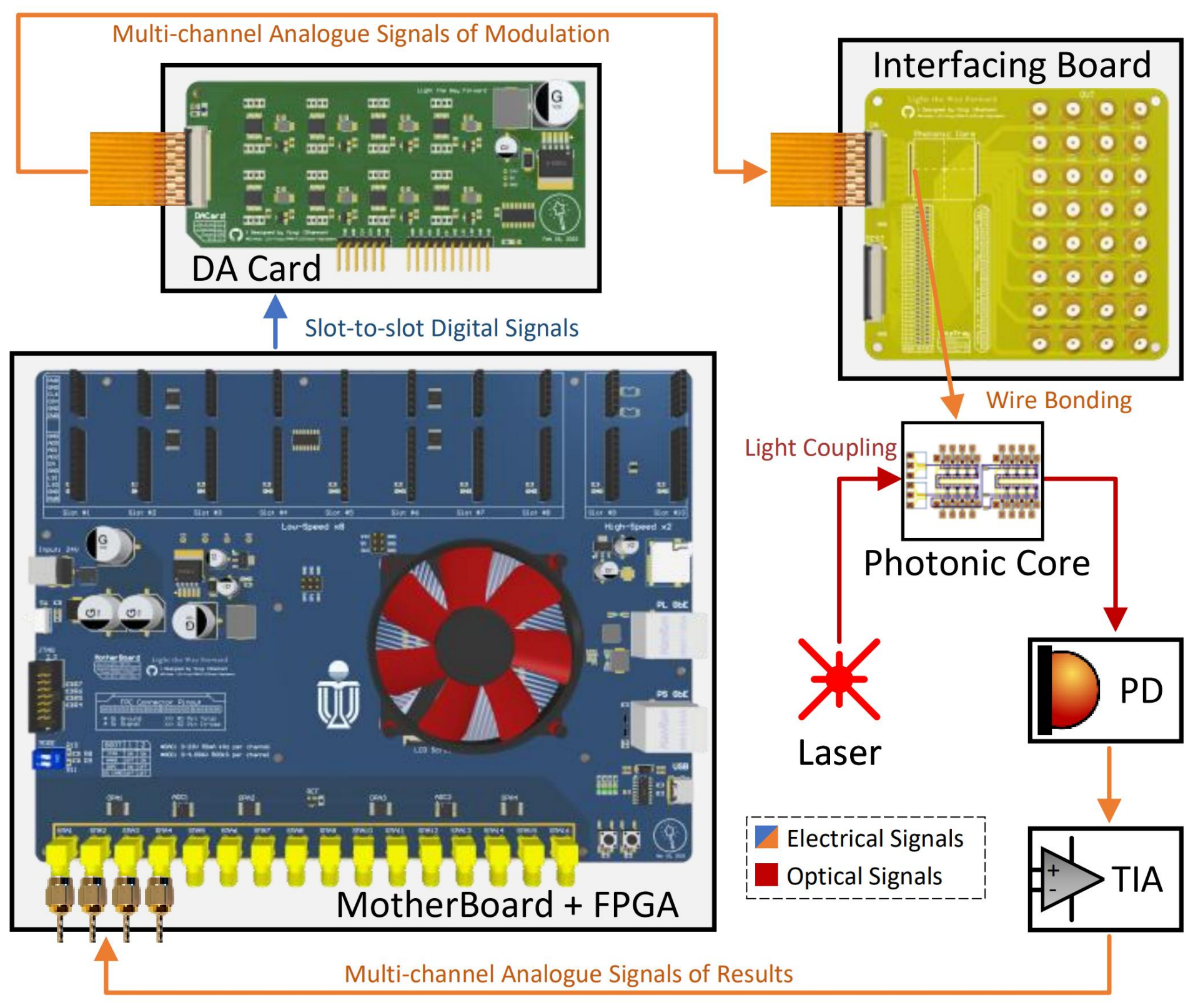
Table 3. Overview of LightRocket Custom Instruction Set Architecture

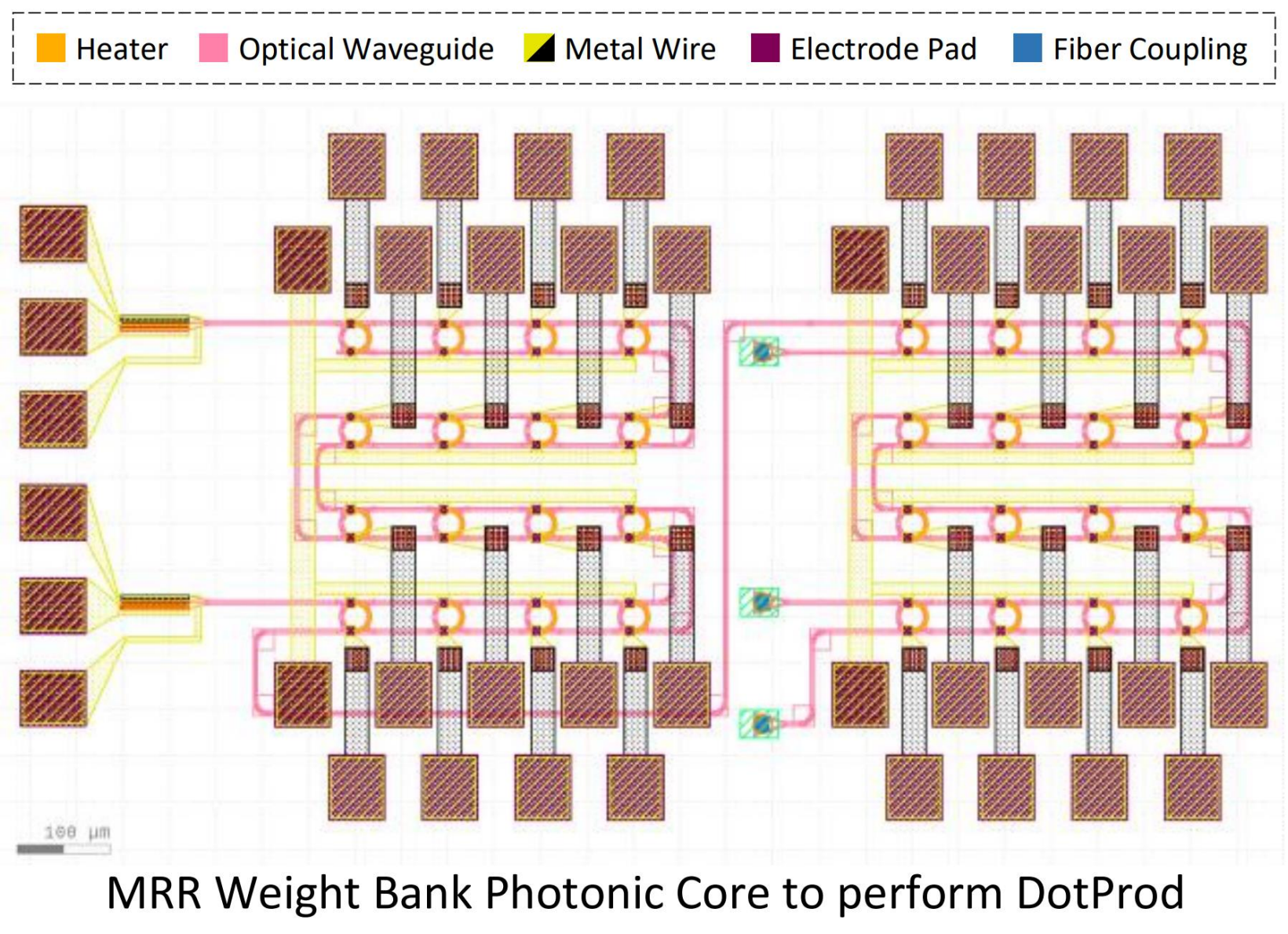
Category	Instruction	Operands								Description
		Funct7 [31:25]	VS2/RS2 [24:20]	VS1/RS1 [19:15]	Funct3 [14:12]			VD/RD [11:7]	OpCode [6:0]	
					xd	xs1	xs2			
pALU	DotProd	41H	V	V	1	0	0	S	0x0B	MRR: RD = VS1[i] · VS2[i]
	MVMul	42H	U	V	0	0	0	V		MZI: VD[i] = MAT @ VS1[i]
	Conv1D	43H	V	V	0	0	0	V		FIR: VD[i] = VS1 ⊛ VS2
eALU	ADD.V	01H	V	V	0	0	0	V		VD[i] = VS1[i] + VS2[i]
	SUB.V	02H	V	V	0	0	0	V		VD[i] = VS1[i] - VS2[i]
	ADD.VS	03H	S	V	0	0	1	V		VD[i] = VS1[i] + RS2
	SUB.VS	04H	S	V	0	0	1	V		VD[i] = VS1[i] - RS2
	MUL.VS	05H	S	V	0	0	1	V		VD[i] = VS1[i] * RS2
	DIV.VS	06H	S	V	0	0	1	V		VD[i] = VS1[i] / RS2
MISC	SHUFFLE.V	0AH	V	V	0	0	0	V		VD[i] = VS1[VS2[i]]
	MAX.V	0BH	0H	V	1	0	0	S		RD = Max(VS1[i])
	MIN.V		1H	V	1	0	0	S		RD = Min(VS1[i])
NLU	PRELU.V	0FH	S	V	0	0	1	V		Leaky param: α = RS2
	TANH.V		1H	V	0	0	0	V		VD[i] = f(VS1[i])
	SIGMOID.V		2H	V	0	0	0	V		where f is nonlinear function
MEM	LOAD.V	10H	U	S	0	1	0	V		VD[i] = Mem[RS1+i*STRIDE]
	STORE.V	11H	V	S	0	1	0	U		MEM[RS1+i*STRIDE] = VS2[i]
CFG	SET.R	18H	U	S	0	1	0	STRIDE		Reg: STRIDE = RS1
			U	S	0	1	0	VLEN		Reg: VLEN = RS1 (i = 0 to VLEN-1)
			S	S	0	1	1	VMASK		Reg: VMASK[RS2] = RS1
			S	S	0	1	1	MAT		Reg: MAT[RS2+i] = Mem[RS1+i]

*Note: V, S, U denote Vector Register, Scalar Register, and Unused, respectively. STRIDE, VLEN, VMASK, MAT are the FIONA custom registers.

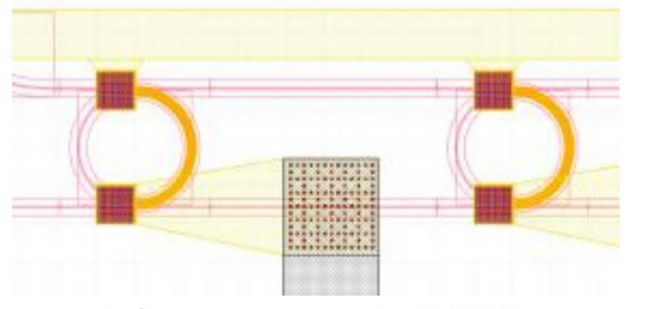
❑ Since FIONA-V vector core is designed to accelerate deep neural network (DNN) workloads, it does not implement the complete RISC-V instruction set. To support execution flow control & scalar operations, it should work with Rocket core through the RoCC interface.

LightRocket Hardware Setup using FIONA Rapid Prototyping Template

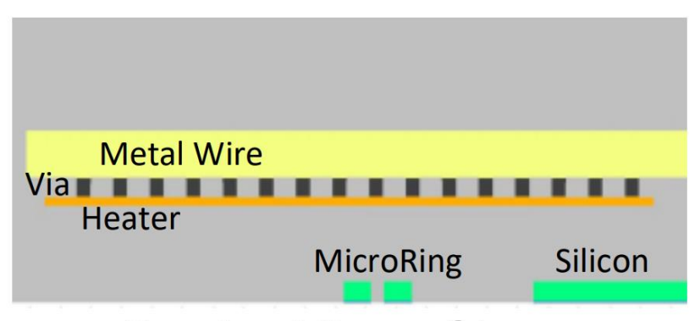




MRR Weight Bank Photonic Core to perform DotProd



Thermo-optic MRR

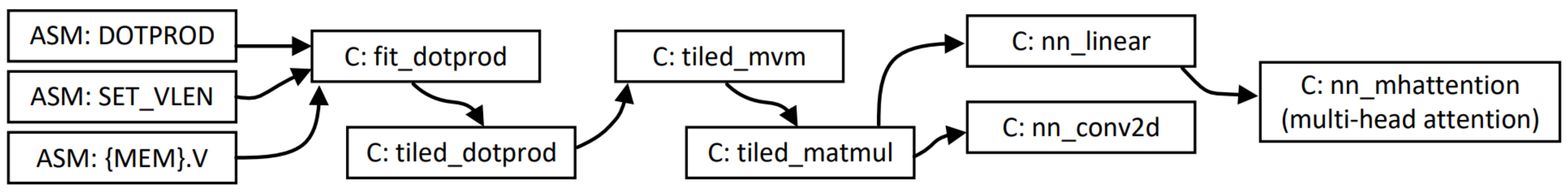


Section View of Layers

➢ In this case study, we decompose and assign matrix-multiplication operations to microring-based (MR) photonic dot-product core. The flow control with partitioning is done by the FIONA software stack.

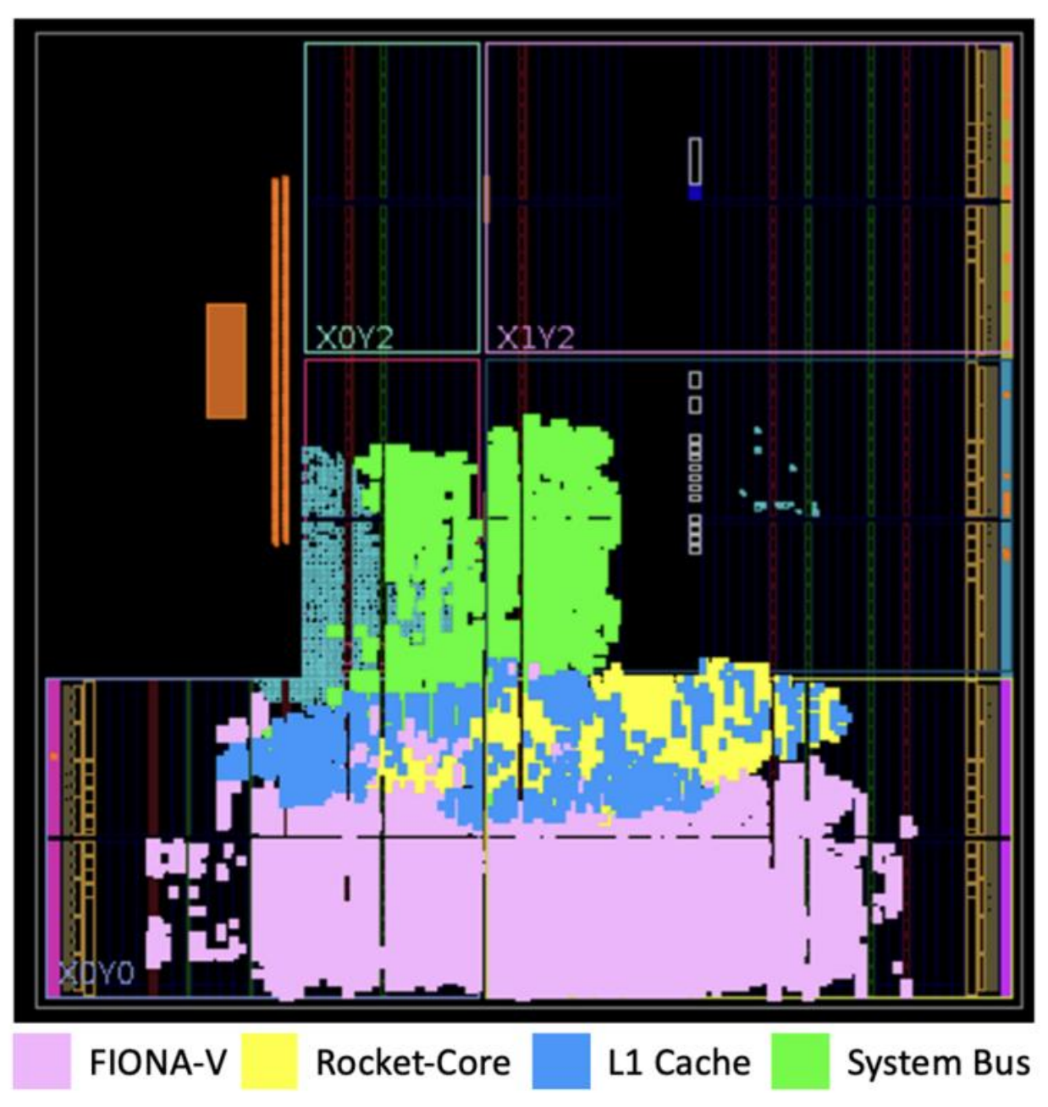
Profile DNN Workloads on LightRocket using FIONA Toolchain

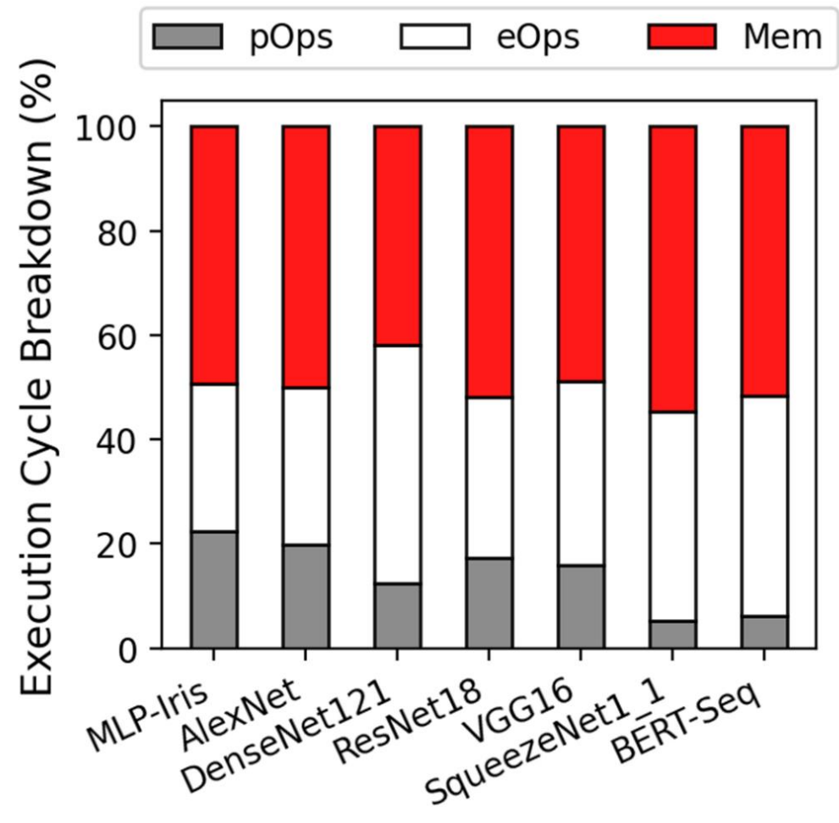
Hand-tuned Kernel Design & Progressively Wrap for Targeted Photonic Backend



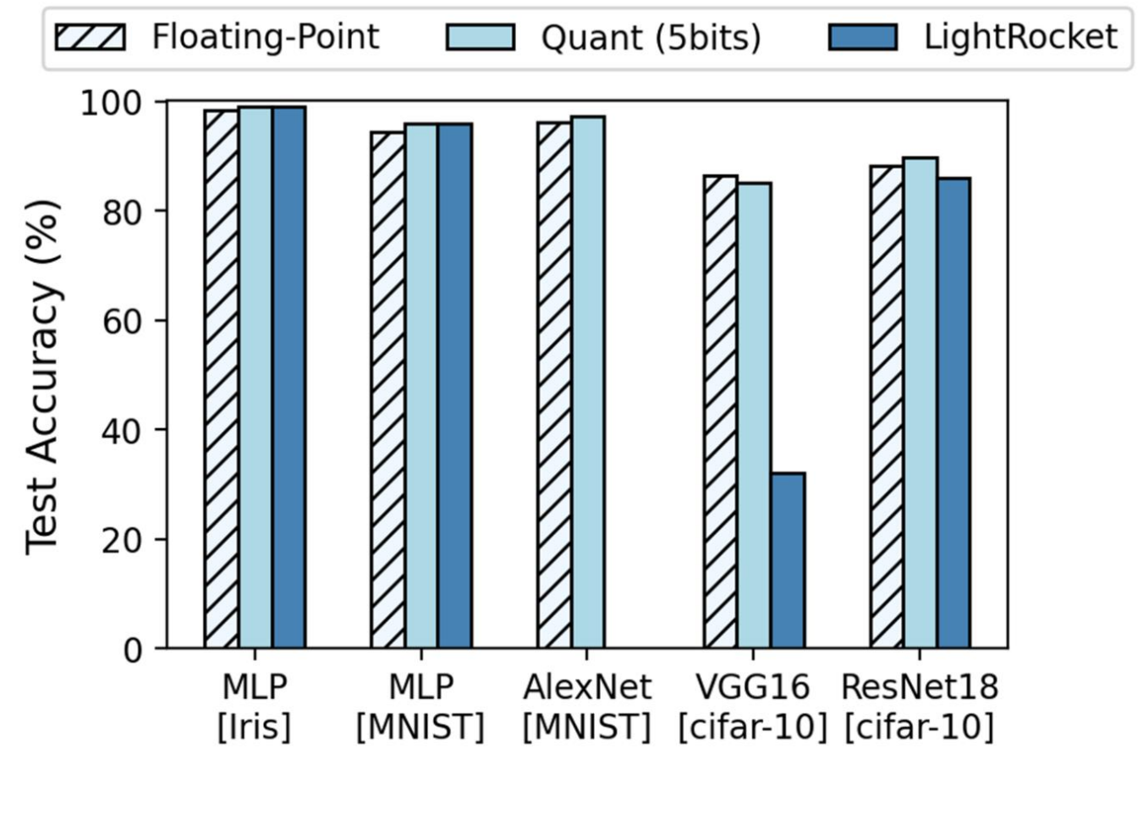
❑ Assembly (ASM) of FIONA-V vector instruction (including photonic ops) wrapped in C/C++ macros.
❑ Rocket-core serves to decompose and schedule arbitrary matrices to fit into targeted photonic units.

Inference & On-chip Training of DNN Workloads on MRR Dot-Product Backend





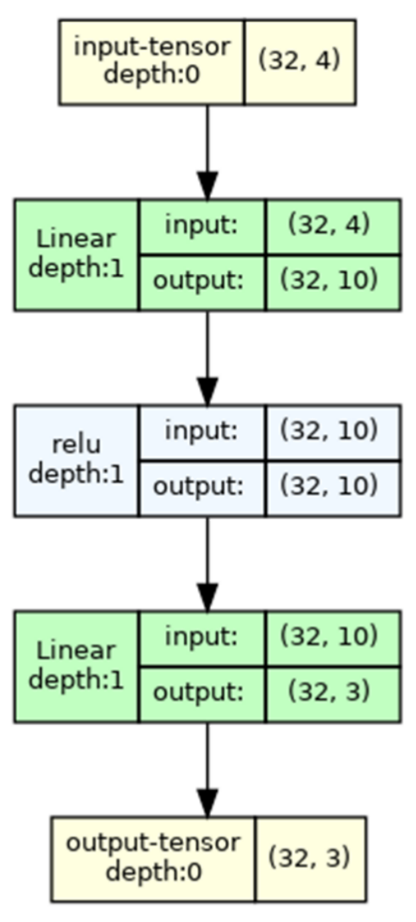
(a) Cycle Breakdown



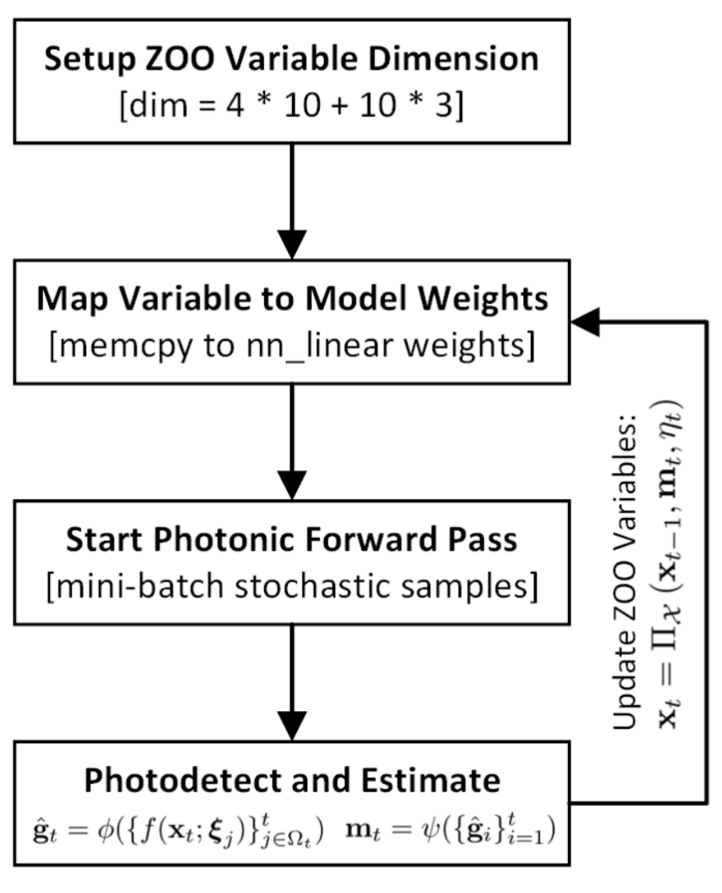
(b) Test Accuracy

Function Unit	LUTs (%)	Regs (%)	Block RAM	DSP
VRF	4%	4%	**15**	0
Activation	8%	10%	1	**4**
ALU	**38%**	22%	0	0
Config	10%	**31%**	0	0
LSU	3%	11%	0	0
MDU	13%	12%	0	2
PUC	23%	11%	7.5	0
Total	**100%**	**100%**	**23.5**	**6**

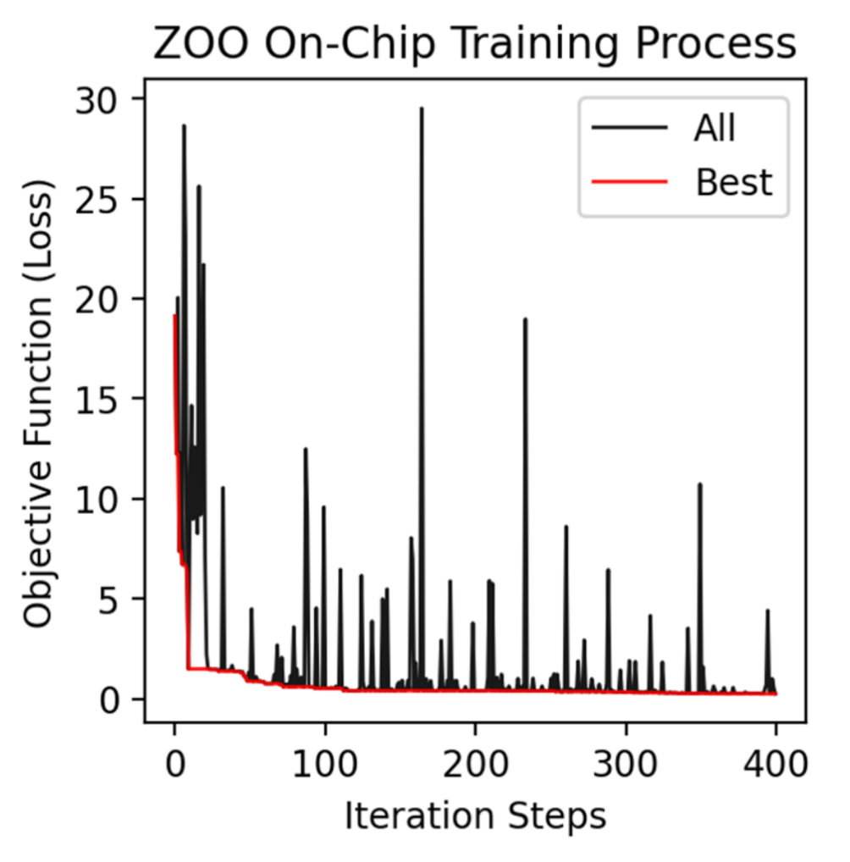
FIONA-V Resource Utilization Breakdown



(a) MLP-Iris Model


Setup ZOO Variable Dimension [dim = 4 * 10 + 10 * 3]
Map Variable to Model Weights [memcpy to nn_linear weights]
Start Photonic Forward Pass [mini-batch stochastic samples]
Photodetect and Estimate
$\hat{\mathbf{g}}_t = \phi(\{f(\mathbf{x}_t;\boldsymbol{\xi}_j)\}_{j\in\Omega_t}^t) \quad \mathbf{m}_t = \psi(\{\hat{\mathbf{g}}_i\}_{i=1}^t)$
Update ZOO Variables:
$\mathbf{x}_t = \Pi_{\mathcal{X}}(\mathbf{x}_{t-1}, \mathbf{m}_t, \eta_t)$

(b) On-Chip ZOO Training Flow



(c) Loss versus ZOO Iteration Steps

✓ Zeroth-order optimization (ZOO) is adopted for on-chip fine-tuning. ZOO can train without gradient.

Conclusion

FIONA toolchain provides a complete end-to-end solution for photonic accelerator systems:
- ➢ Software stack: to investigate the relationship among ISA, simulator and compiler.
- ➢ Hardware stack: to rapidly prototype and verify the designed photonic computing system.

✓ For researchers in the field of micro-architecture/compiler/system:
◆ No need to painfully master the hardware details as a prerequisite.
✓ For researchers in the field of device design/testing/characterization:
◆ An out-of-the-box toolkit to handle all the high-level details.

For more details, please refer to "**FIONA: Photonic-Electronic Co-Simulation Framework and Transferable Prototyping for Photonic Accelerator**", *International Conference on Computer-Aided Design (ICCAD), 2023.*