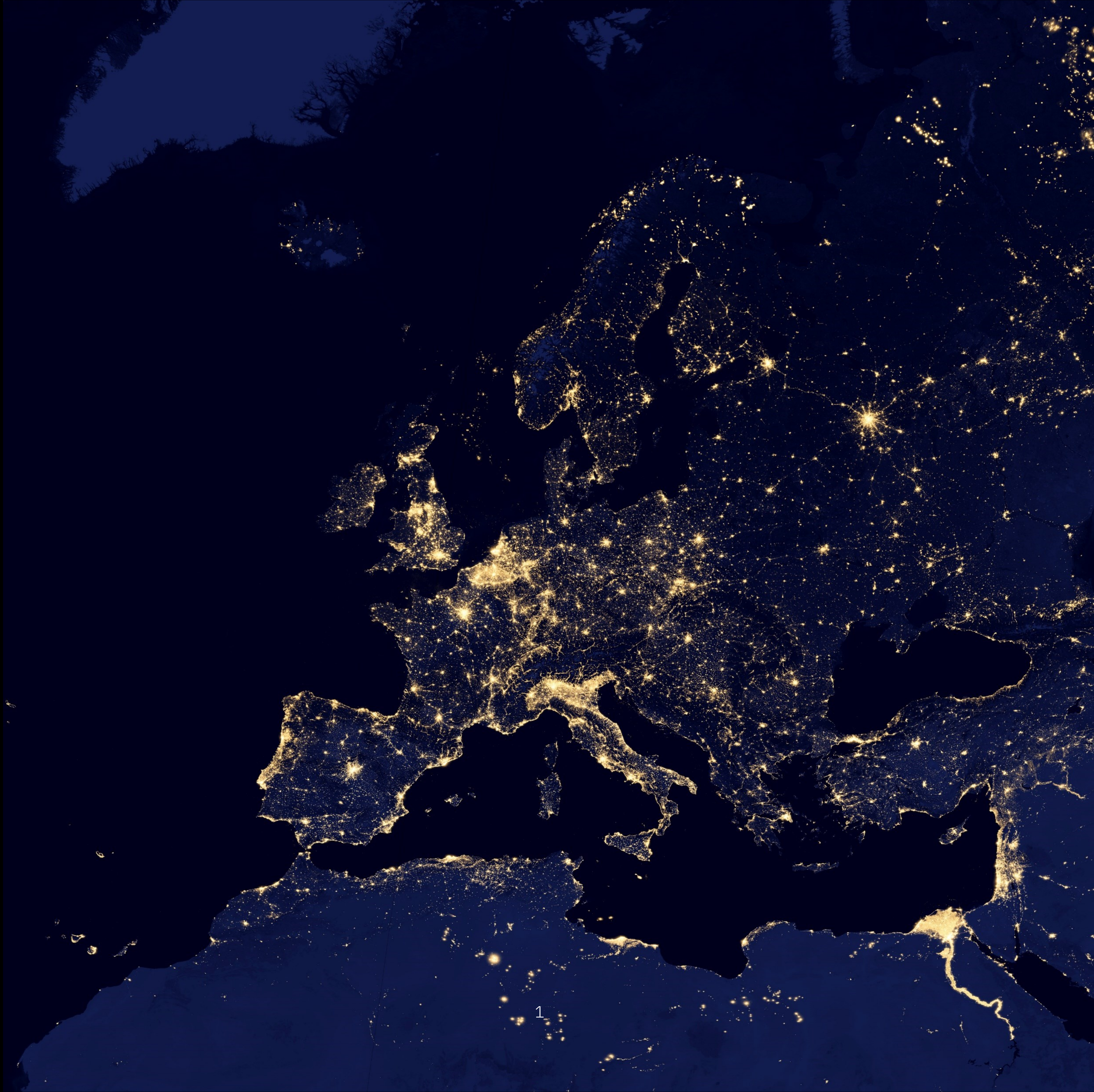
IBM Research Europe

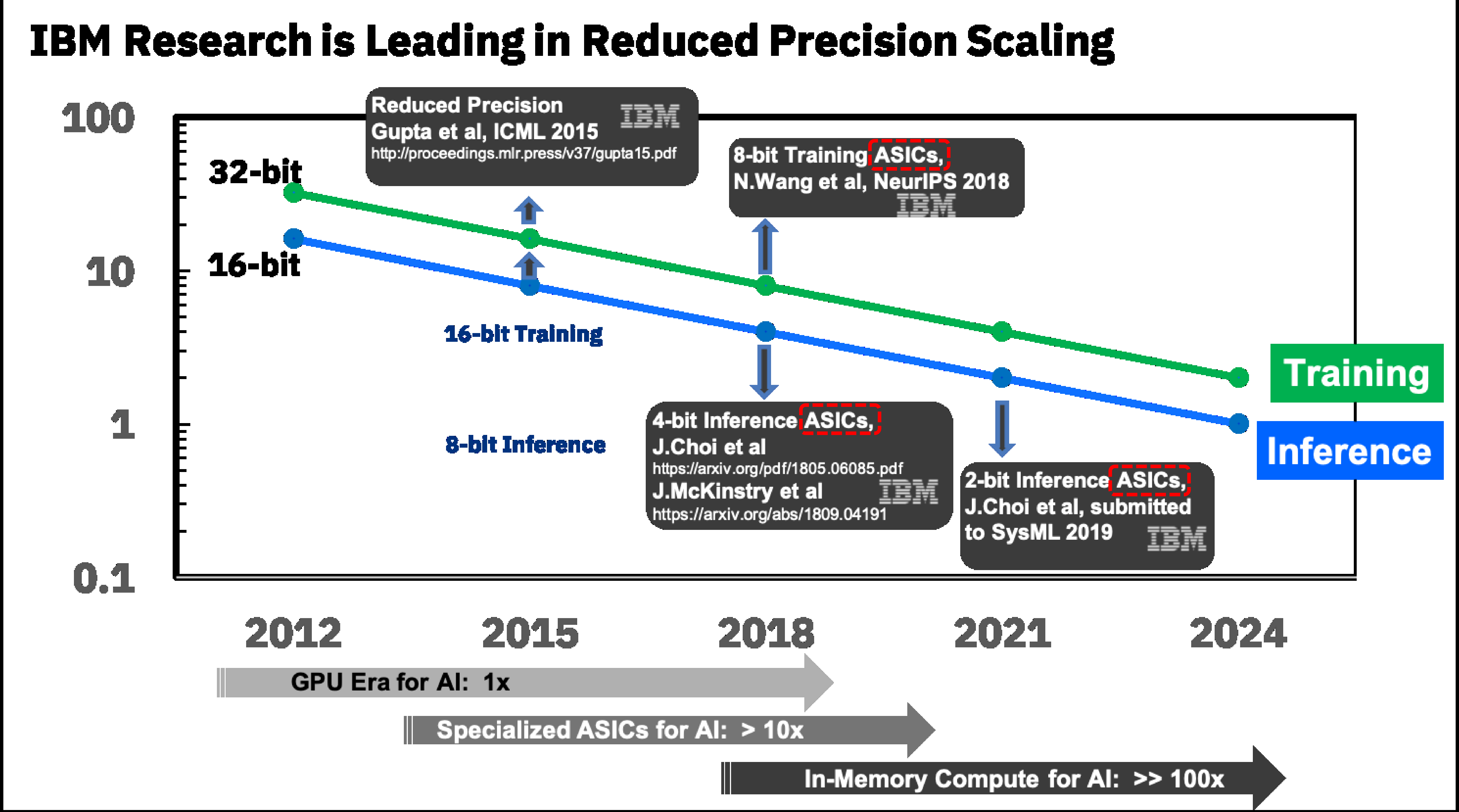
PHRYCTORIA: A Messaging System for Transprecision OpenCAPI-attached FPGA Accelerators

Dionysios Diamantopoulos, Mitra Purandare, Burkhard Ringlein and Christoph Hagleitner

Cloud FPGAs & Tape Group
Cloud & AI Systems Research Department
did@zurich.ibm.com



IBM Research is Leading in Reduced Precision Scaling



IBM Research is Leading in Reduced Precision Scaling

Reduced Precision
Gupta et al, ICML 2015
http://proceedings.mlr.press/v37/gupta15.pdf

8-bit Training ASICs,
N.Wang et al, NeurIPS 2018

4-bit Inference ASICs,
J.Choi et al
https://arxiv.org/pdf/1805.06085.pdf
J.McKinstry et al
https://arxiv.org/abs/1809.04191

2-bit Inference ASICs,
J.Choi et al, submitted to SysML 2019

GPU Era for AI: 1x

Specialized ASICs for AI: > 10x

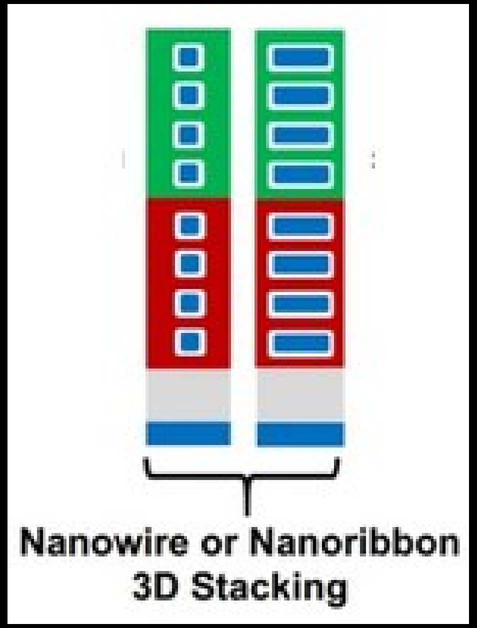
In-Memory Compute for AI: >> 100x

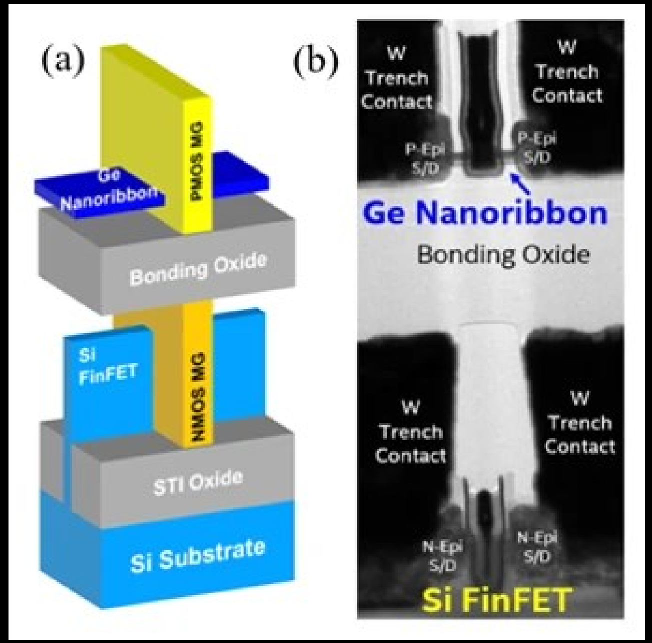
Source: https://www.ibm.com/blogs/research/2018/12/8-bit-precision-training/

Did anyone say Moore’s Law End ?

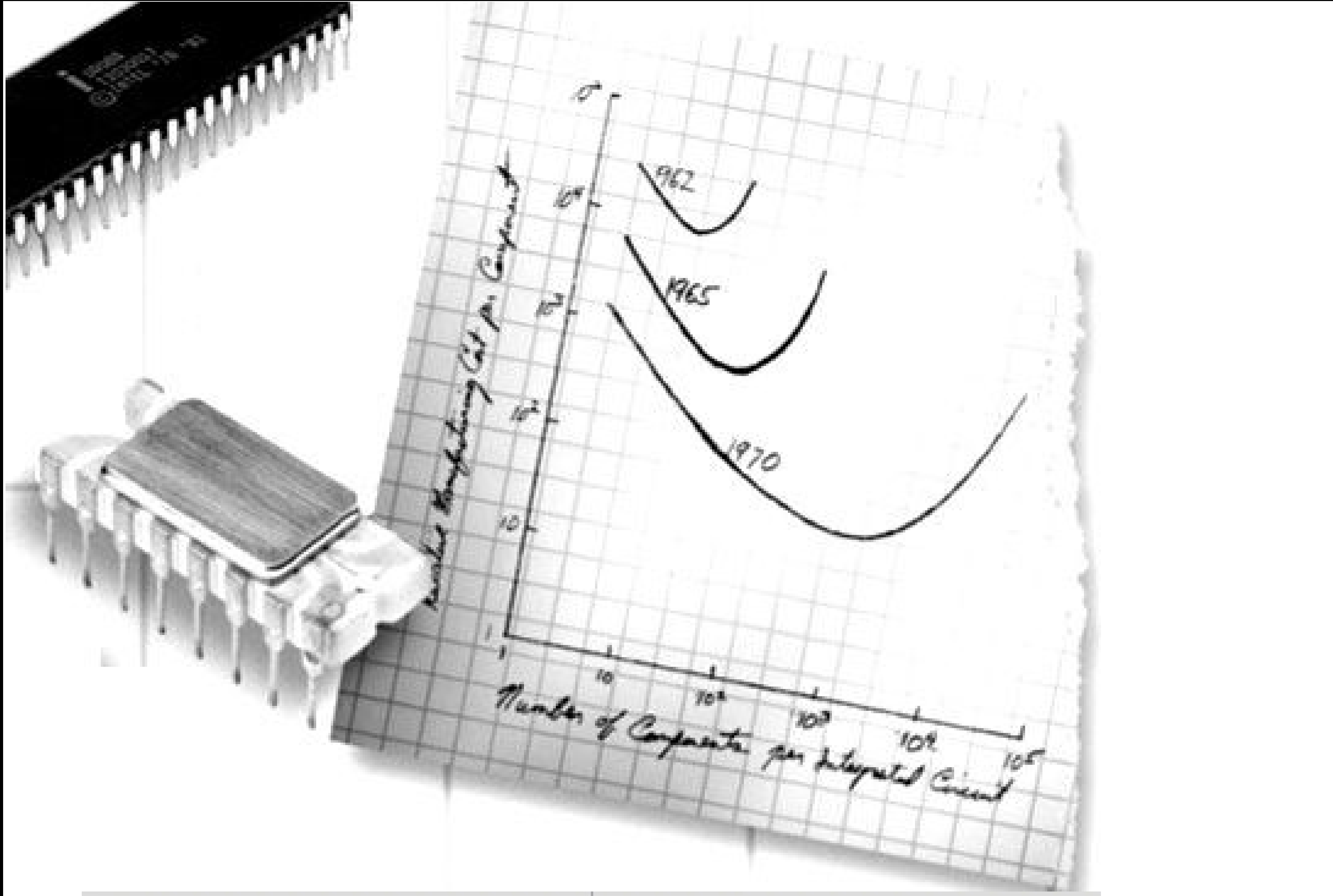
—

Transistor scaling

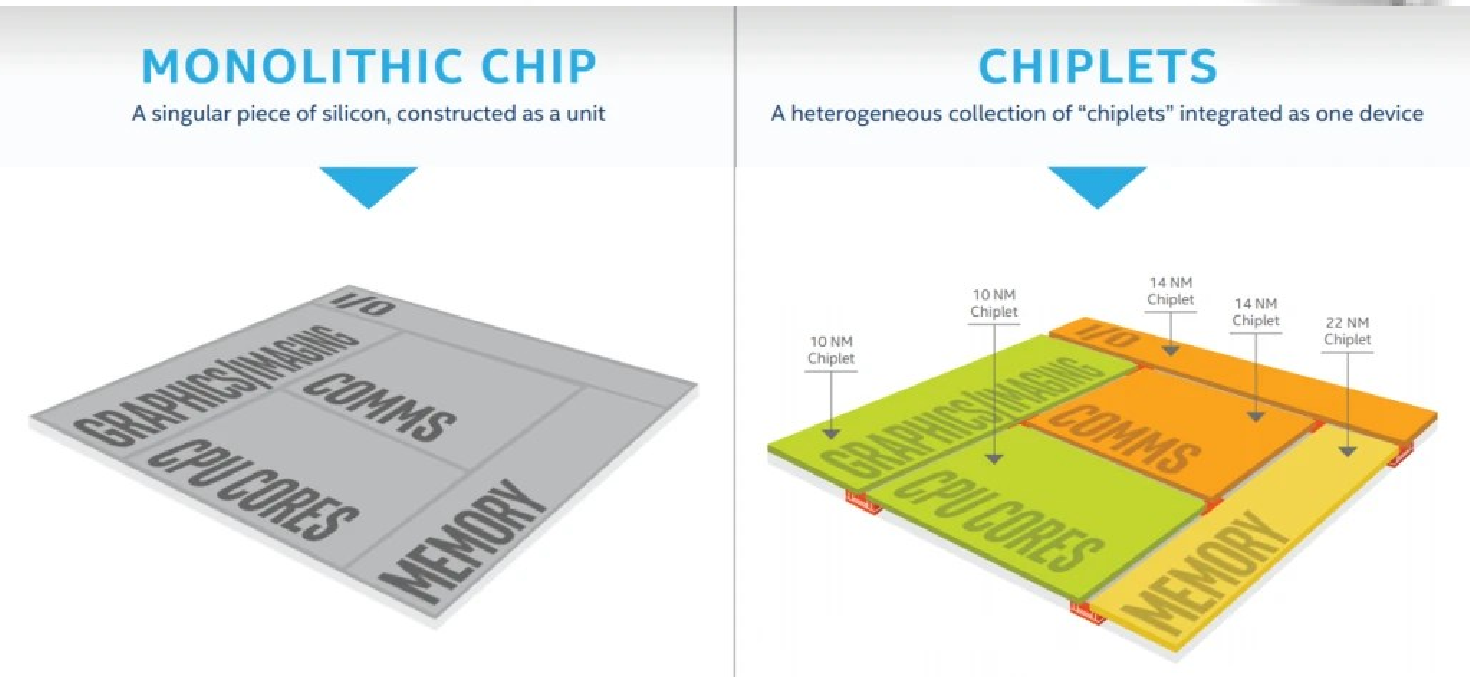




Intel, IEDM 2019, Germanium-based GAAFET PMOS device layer on top of a more traditional silicon FinFET NMOS



- 5 chipmakers/foundries in the 16nm/14nm market—GlobalFoundries, Intel, Samsung, TSMC UMC, SMIC (14nm finFETs).
- GlobalFoundries and UMC last year halted their respective 7nm process efforts.
- Currently, TSMC's 7nm process is in its peak (orders from AMD for its Ryzen 3000-series CPUs and Navi graphics cards). Huge invest in 5nm.
- Compared to 7nm, Samsung’s 5nm finFET technology provides up to a 25% increase in logic area with 20% lower power or 10% higher performance.
- TSMC expects mass 3nm production in 2022.
- A nanosheet FET is a type of gate-all-around (GAA) architecture. That’s not the only possible scenario. “The industry is very conservative. They will try to extend the finFET as much as possible,” IMEC’s Naoto Horiguchi said. “At 3nm, we have a window to use a finFET. But we need several process innovations for finFET in terms of overall improvement.
- TSMC announced starting 2nm development (Apr. 2020)


MONOLITHIC CHIP

A singular piece of silicon, constructed as a unit

CHIPLETS

A heterogeneous collection of “chiplets” integrated as one device

System scaling: Beyond the transistor, e.g. Intel’s EMIB (Embedded Multi-die Interconnect Bridge) and Foveros to connect chiplets in both 2 and 3 dimensions (HBM in CPU-GPU)

https://semiengineering.com/5nm-vs-3nm/

Did anyone say Moore’s Law End ?

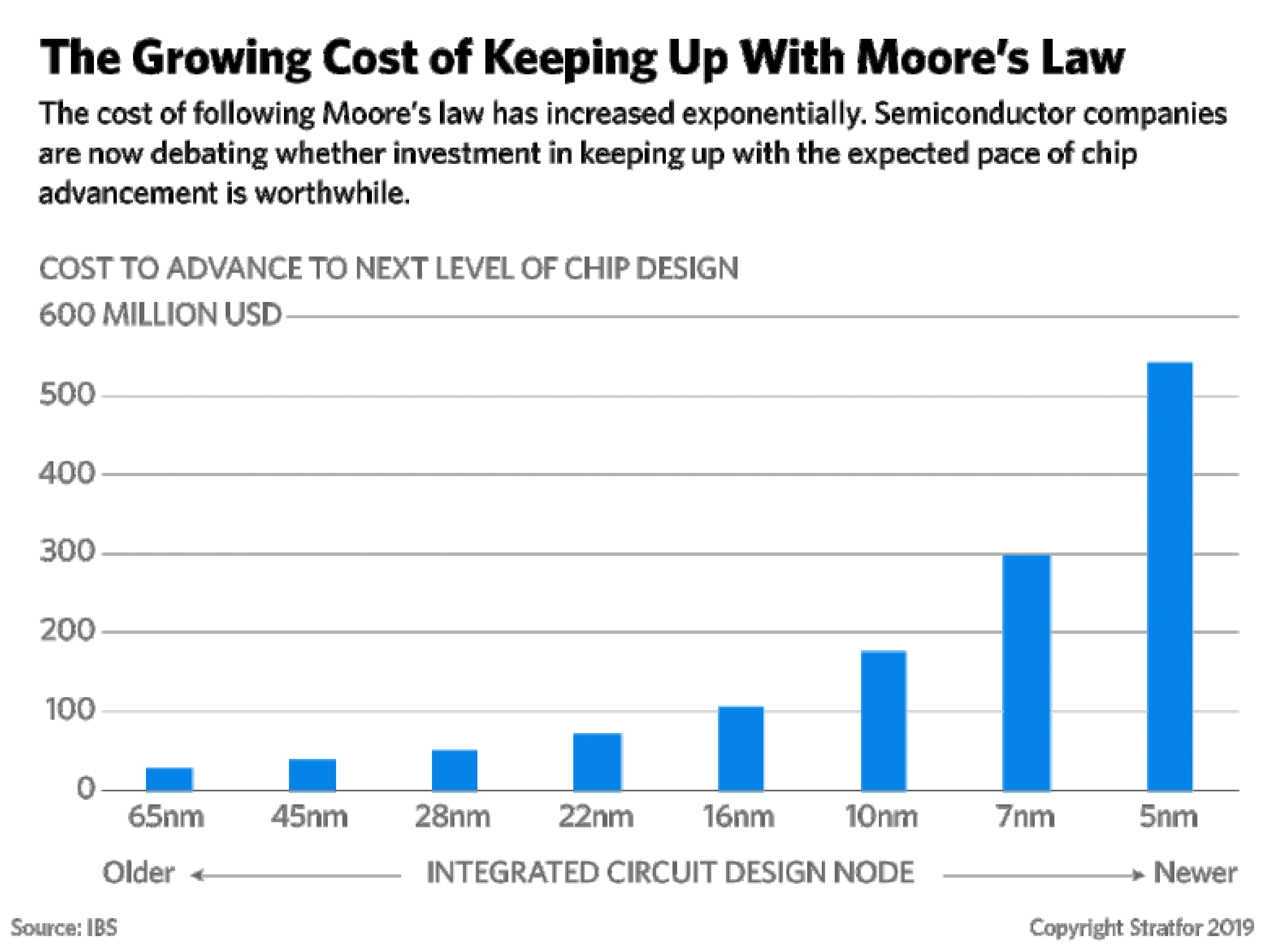
—

~~Transistor scaling~~ Cost scaling

$20B per fab run at 3nm (IBS)

- After 5nm, the next full node is 3nm. But 3nm is not for the faint of heart.
- The cost to design a 3nm device ranges from $500 million to $1.5 billion, according to IBS.
- Process development costs ranges from $4 billion to $5 billion, while a fab runs $15 billion to $20 billion, according to IBS.
- “Transistor costs at 3nm are expected to be 20% to 25% higher than at 5nm based on same level of maturity,” IBS’ Jones said. “Expect 15% more performance and with 25% less power consumption compared to 5nm finFETs.”

https://semiengineering.com/5nm-vs-3nm/



The Growing Cost of Keeping Up With Moore’s Law

The cost of following Moore’s law has increased exponentially. Semiconductor companies are now debating whether investment in keeping up with the expected pace of chip advancement is worthwhile.

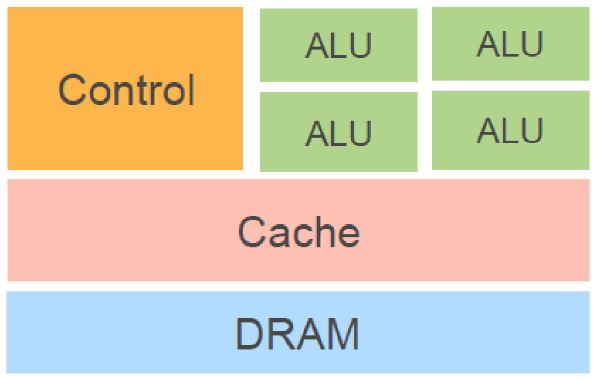
Source: IBS

Copyright Stratfor 2019

- The cost to design a 28nm planar device ranges from $10 million to $35 million (Gartner).
- The cost to design a 7nm system-on-a-chip (SoC) ranges from $120 million to $420 million (Gartner).
- 5nm is a completely new process with updated EDA tools and IP. The cost to design a 5nm device ranges from $210 million to $680 million (Gartner).

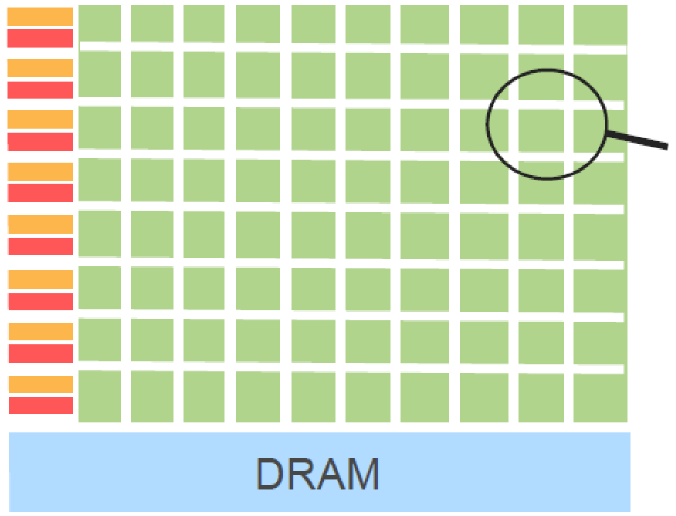
Silicon alternatives for rapid enterprise-ready specialization

A GPU is effective at processing the same set of operations in parallel – single instruction, multiple data (SIMD). A GPU has a well-defined instruction-set, and fixed word sizes – for example single, double, or half-precision integer and floating point values.

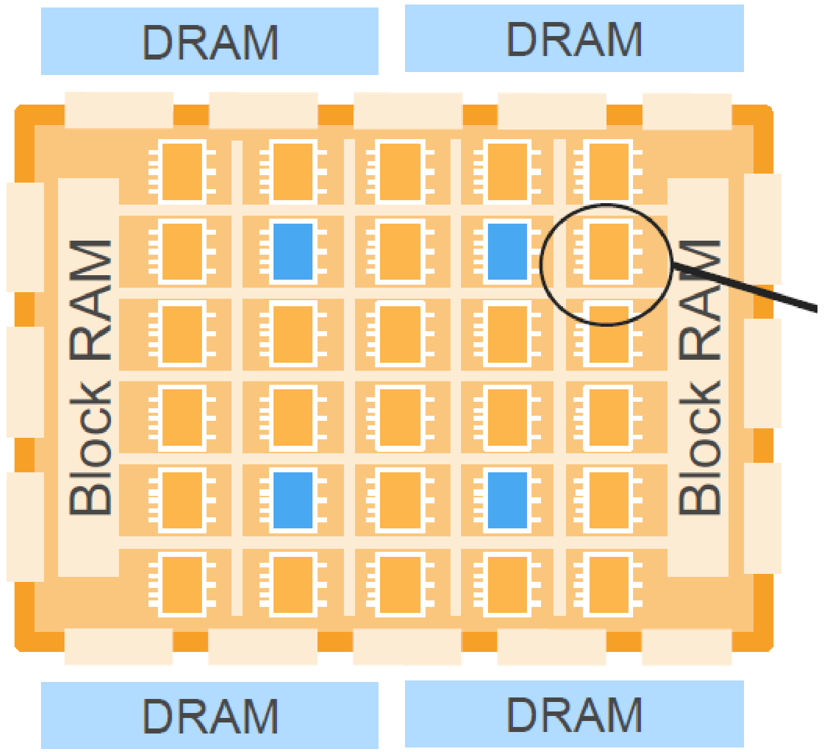


CPU

(one core)



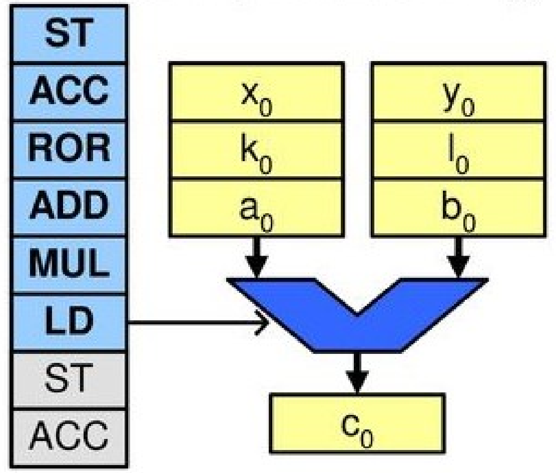
GPU



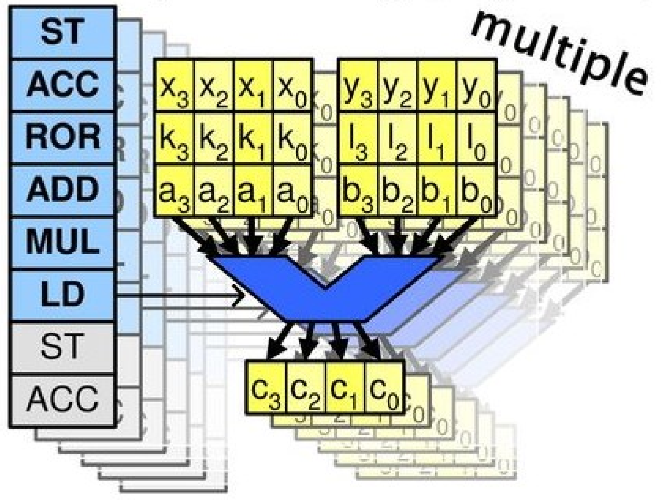
- ✓ Reconfigurable **logic**
- ✓ Reconfigurable **memory**
- ✓ Reconfigurable **interconnects**

FPGA

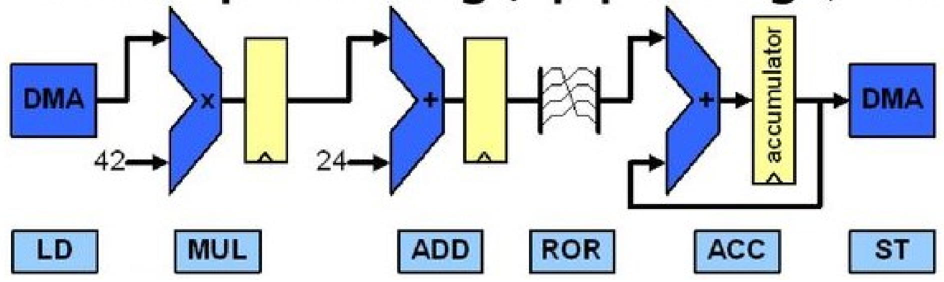
Scalar processing

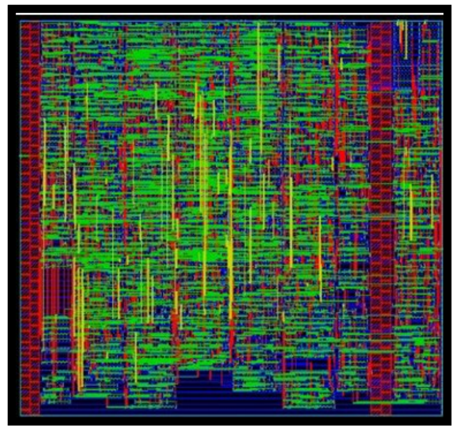


SIMD processing (e.g. GPU)



Dataflow processing / pipelining (FPGAs)



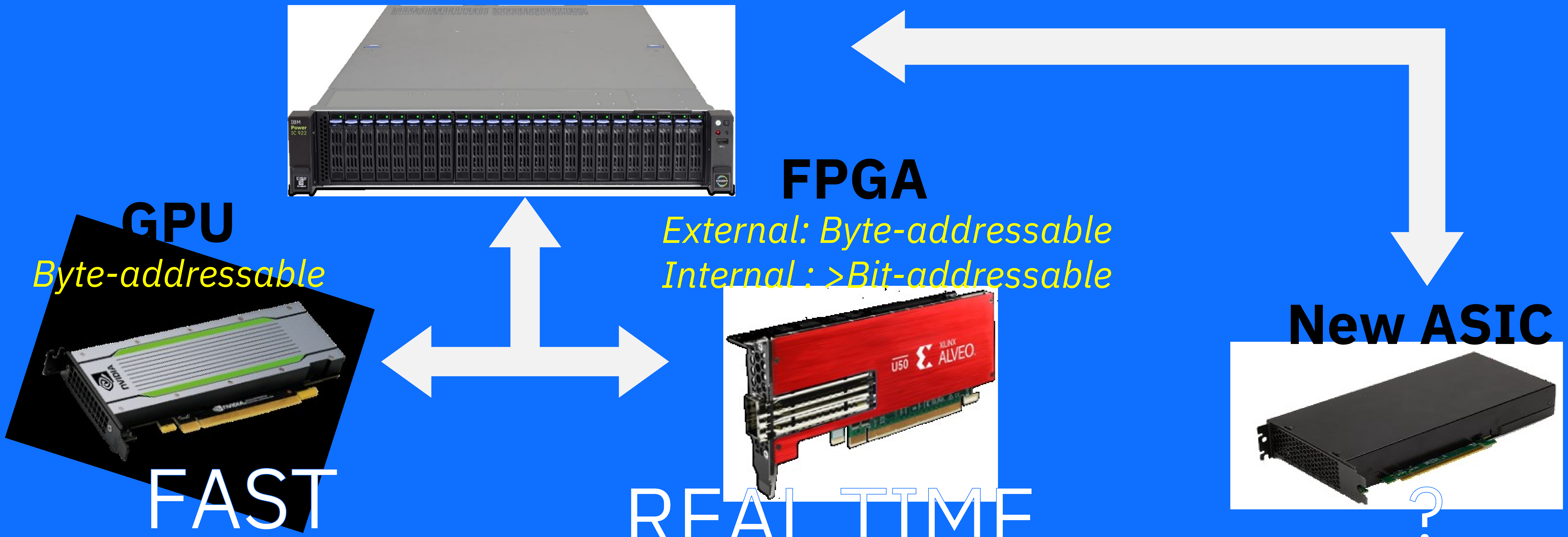


ASICs

- An **FPGA** is effective at processing the same or different operations in parallel – multiple instructions, multiple data (MIMD). An FPGA does not have a predefined instruction-set, or a fixed data width.

Figures source: AWS - Announcing Amazon EC2 F1 Instances with Custom FPGAs, Bringing Hardware Acceleration closer to the programmer, Ecoscale-ExaNest workshop, 2017

Silicon alternatives for rapid POWER^{TM} specialization



Byte-addressable

GPU

Byte-addressable

FAST

- *Energy efficient*
- *Optimized for scale-out servers*
- *Purpose built for inferencing*
- *40x better low-latency throughput than CPUs*
- *2x decoding performance over prior generation GPUs*

FPGA

External: Byte-addressable

Internal : >Bit-addressable

REAL TIME

- *Low batch, low latency inferencing solution for ML, video processing, genomics, analytics*
- *Reconfigurable to optimize for shifting workloads*
- *20x versus throughput over CPU's, 3x reduced latency over GPUs*

New ASIC

?

- *Custom accelerator designed for inferencing workloads*
- *Designed for power efficiency and cost savings*

Note: Used material from "IC922 Seller Deck (2020-Mar-24)"

A look on an Enterprise's Portfolio for the AI Era

From Mission-Critical workloads to AI and Cloud Computing leadership

PowerVM and high RAS

L922

- Industry leading reliability and computing capability
- PowerVM ecosystem focus for outstanding utilization
- Focus on memory capacity with up to 4TB of RAM

1st TOP500!
200,795 TFlop/s
...and Sierra 2nd

4,608x

Accelerated Compute

AC922

- Industry first and only in advanced IO with 2nd Generation CPU - GPU NVLink delivering ~5.6x higher data throughput
- Up to 4 integrated NVIDIA "Volta" GPUs air cooled (GTH) and up to 6 GPUs with water cooled (GTX) version
- OpenCAPI support (FPGA, ASIC)
- Memory coherence

Data, Inferencing, and Cloud

IC922

NEW!

- Storage dense, high bandwidth server – up to 24 NVMe or SAS/SATA in 2U[1]
- Advanced IO with PCIe Gen4
- Optimized inferencing server with up to 6 Nvidia T4 GPUs at GA and additional accelerators in roadmap[1]
- OpenCAPI support[1] (FPGA, ASIC)
- Price/performance server

Big Data

LC922 / LC921

- Big Data server with up to 120TB storage capacity, large form factor support, KVM support, leveraging P9 compute for a composable design
- 1U and 2U form factors
- Advanced IO with PCIe 4.0/CAPI 2.0 (FPGA, ASIC)
- Up to 44 cores (2U) or 40 cores (1U) at lower frequency

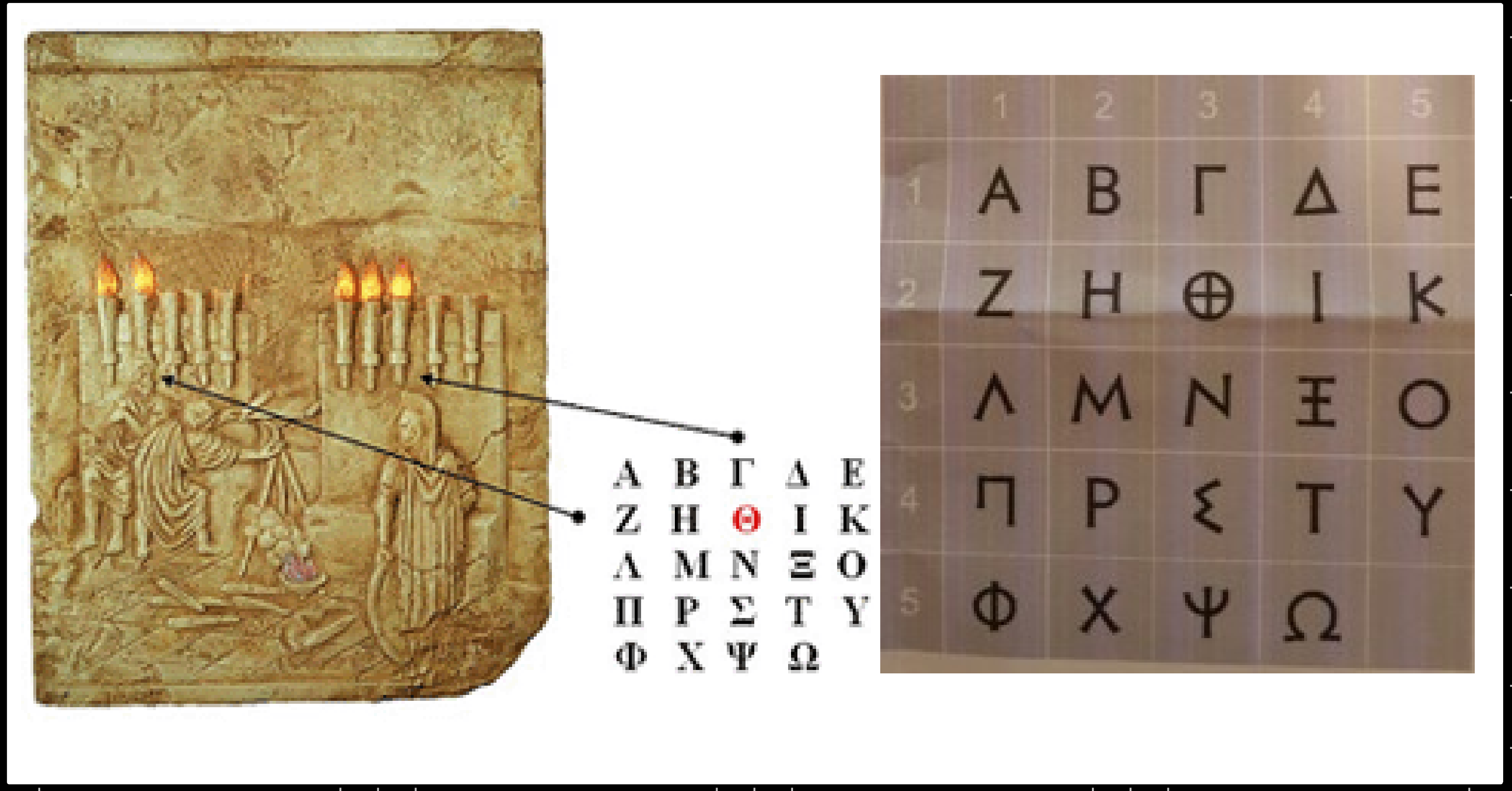
Note: Used material from "IC922 Seller Deck (2020-Mar-24)"

Proposed POWER Processor Technology and I/O Roadmap

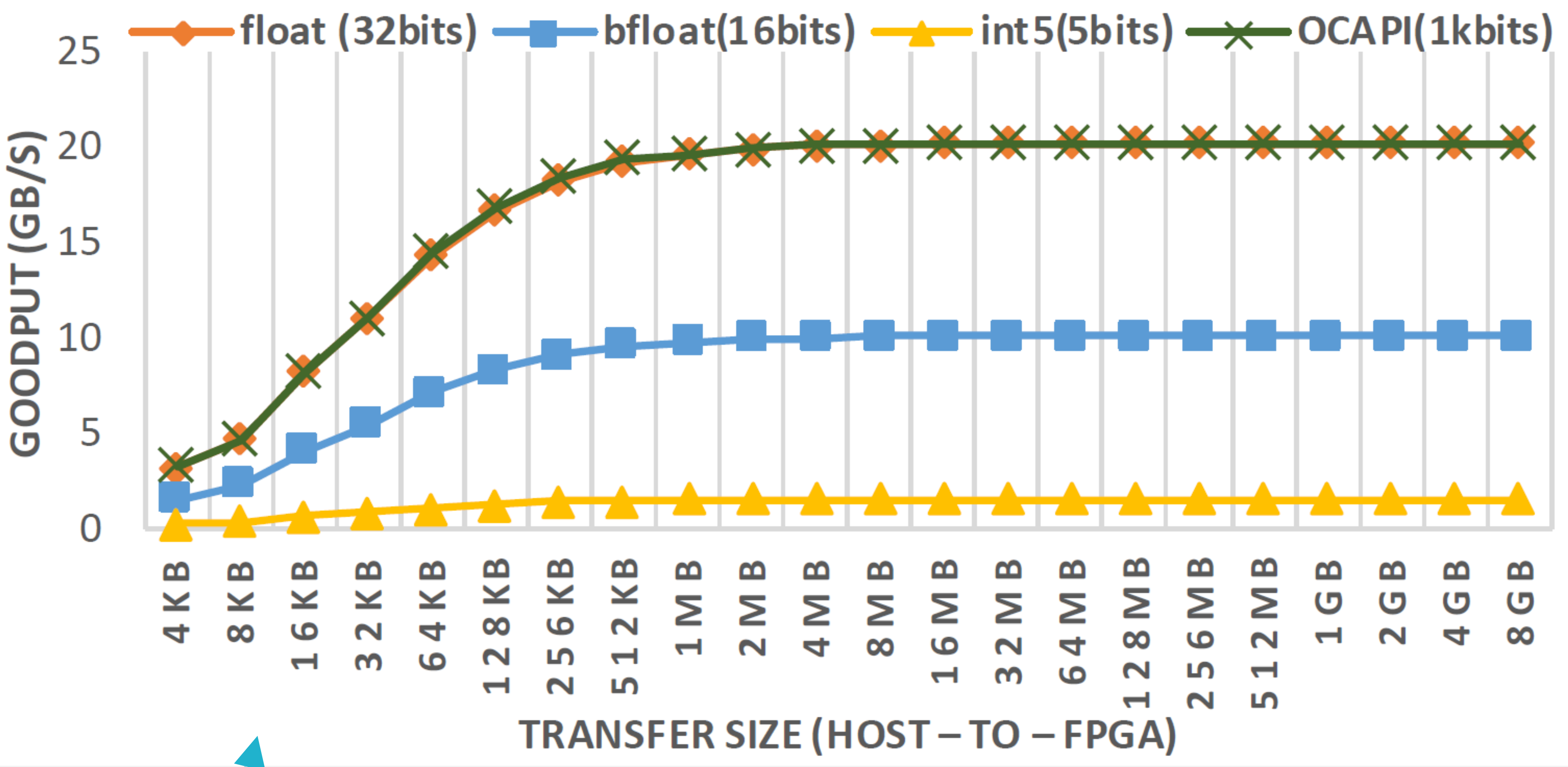
	POWER7 Architecture		POWER8 Architecture		POWER9 Architecture			POWER10
	2010 POWER7 8 cores 45nm New Micro-Architecture New Process Technology	2012 POWER7+ 8 cores 32nm Enhanced Micro-Architecture New Process Technology	2014 POWER8 12 cores 22nm New Micro-Architecture New Process Technology	2016 POWER8 w/ NVLink 12 cores 22nm Enhanced Micro-Architecture With NVLink	2017 P9 SO 12/24 cores 14nm New Micro-Architecture Direct attach memory New Process Technology	2018 P9 SU 12/24 cores 14nm Enhanced Micro-Architecture Buffered Memory	2019 P9 w/ Adv. I/O 12/24 cores 14nm Enhanced Micro-Architecture New Memory Subsystem	2020+ P10 TBA cores New Micro-Architecture New Technology
Sustained Memory Bandwidth	65 GB/s	65 GB/s	210 GB/s	210 GB/s	150 GB/s	210 GB/s	350+ GB/s	435+ GB/s
Standard I/O Interconnect	PCIe Gen2	PCIe Gen2	PCIe Gen3	PCIe Gen3	PCIe Gen4 x48	PCIe Gen4 x48	PCIe Gen4 x48	PCIe Gen5
Advanced I/O Signaling	N/A	N/A	N/A	20 GT/s 160GB/s	25 GT/s 300GB/s	25 GT/s 300GB/s	25 GT/s 300GB/s	32 & 50 GT/s
Advanced I/O Architecture	N/A	N/A	CAPI 1.0	CAPI 1.0 , NVLink	CAPI 2.0, OpenCAPI3.0, NVLink	CAPI 2.0, OpenCAPI3.0, NVLink	CAPI 2.0, OpenCAPI4.0, NVLink	TBA

PHRYCTORIA motivation:
I/O + low-bit = ?

Traditional communication mechanisms for modern low-precision data-types (e.g. brain-float16, int5) cannot exploit the bandwidth of emerging communication links for FPGA accelerators (e.g. OpenCAPI, PCIe4, etc).



PHRYCTORIA name inspired after the ancient Greek communication system “ΦΡΥΚΤΩΡΙΑ”, 1900 B.C.



Heterogeneous System: IBM IC922, 2POWER9 CPUs, AlphaData ADM-9H7 (Xilinx VU37P FPGA), OpenCAPI 3.0 25Gbps8.

-2x BW utilization for *brain-float16*

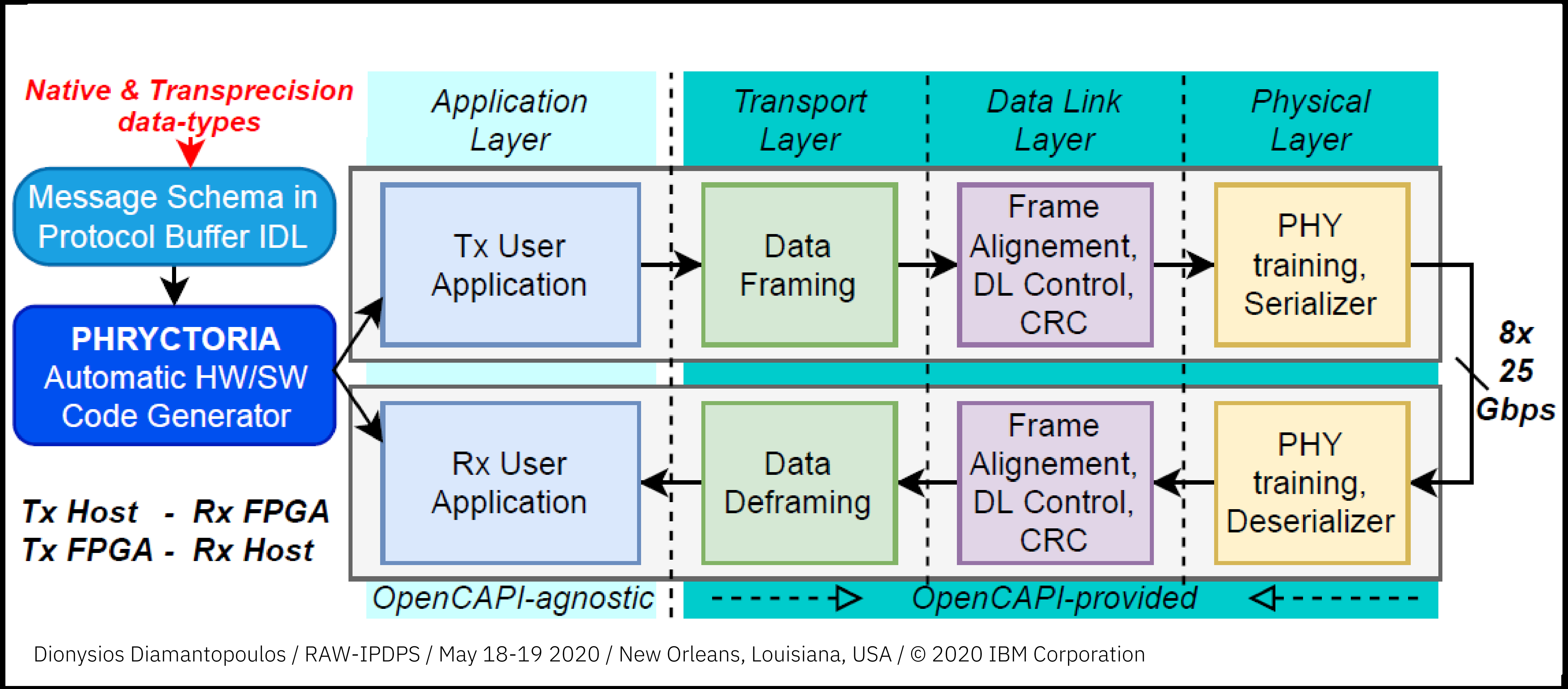
-6x BW utilization for *int5*

- The actual number of bits needed to reassemble a datum of a transprecision data type from the total number of bits transferred for that datum is very low.
- The number of bits transferred per second defines throughput.
- However, out of the transferred bits, the number of bits needed to reconstruct a datum of a certain data type at the accelerator end defines the goodput.

PHRYCTORIA messaging system

Contributions:

- ➢ supporting message schemas expressed in protobuf IDL for specifying to/from communication data structure between the POWER host and the FPGA accelerator
- ➢ automatically generating the serialization/deserialization functions for accelerator and software interfaces from the protobuf schemas
- ➢ leveraging varint encoding for transprecision data types to increase the goodput of the communication.



Dionysios Diamantopoulos / RAW-IPDPS / May 18-19 2020 / New Orleans, Louisiana, USA / © 2020 IBM Corporation

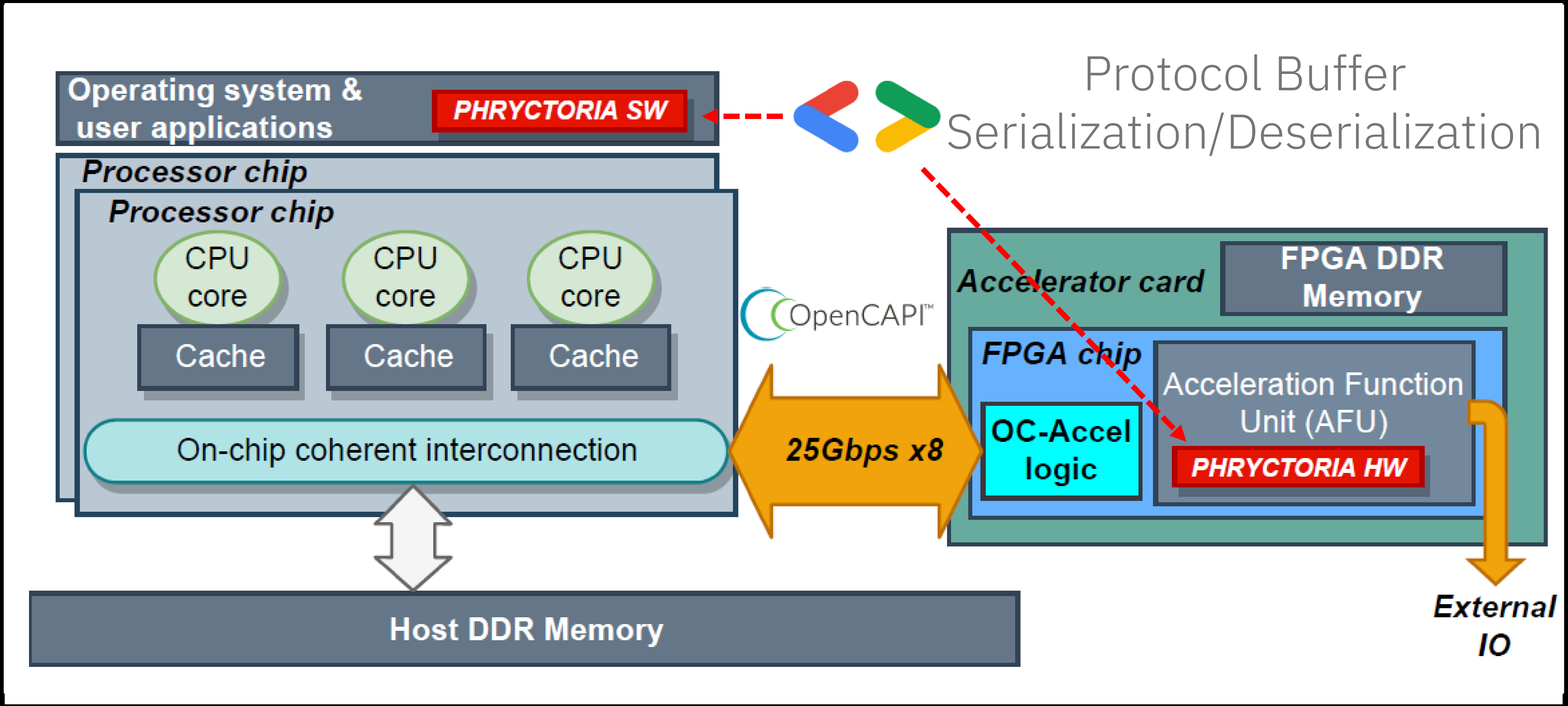
Where protobuf fits in?

Platform independent message representation formats, like Protocol buffers (protobufs), XML, or JSON allow applications running on different systems, to exchange data.

Practically what we propose?

We address the issue of low goodput for transprecision FPGA accelerators by leveraging the varint encoding of protobuf for serialization/deserialization of transprecision data types.

PHRYCTORIA conceptual system



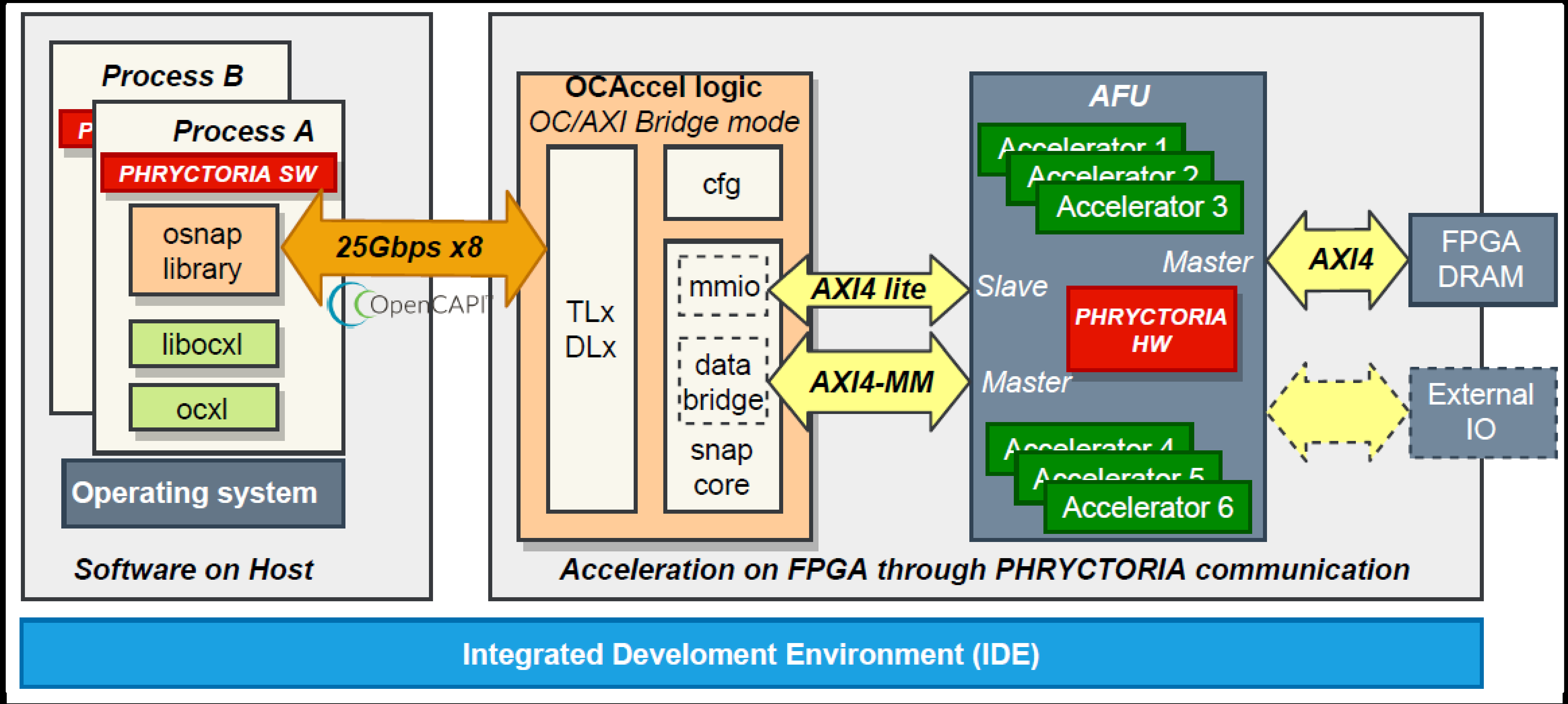
The PHRYCTORIA system assumes the co-existence of FPGA-based accelerator cards along with general-purpose processors in a shared memory symmetric multi-processor system (SMP)

The architecture assumes that the reconfigurable logic devices can directly access the same virtual address space of general purpose processors.

A specific OpenCAPI Accelerator logic (OC-Accel logic), overlaid on FPGA's resources enables the coherent access to the CPU's on-chip coherent interconnection bus.

The PHRYCTORIA system has two components, i.e. the PHRYCTORIA SW, layered as an application on the OS and the PHRYCTORIA HW, layered inside the AFU, as an interface between the OC-Accel logic and the accel/tors.

PHRYCTORIA integrated development environment



To facilitate a rapid development and deployment environment, the OpenCAPI consortium introduced the OC-Accel Integrated Development Environment (IDE).

This framework consists of HW components, SW components and automation tools that allow users to quickly develop, debug and evaluate FPGA accelerators coherently attached to CPUs using the OpenCAPI communication link.

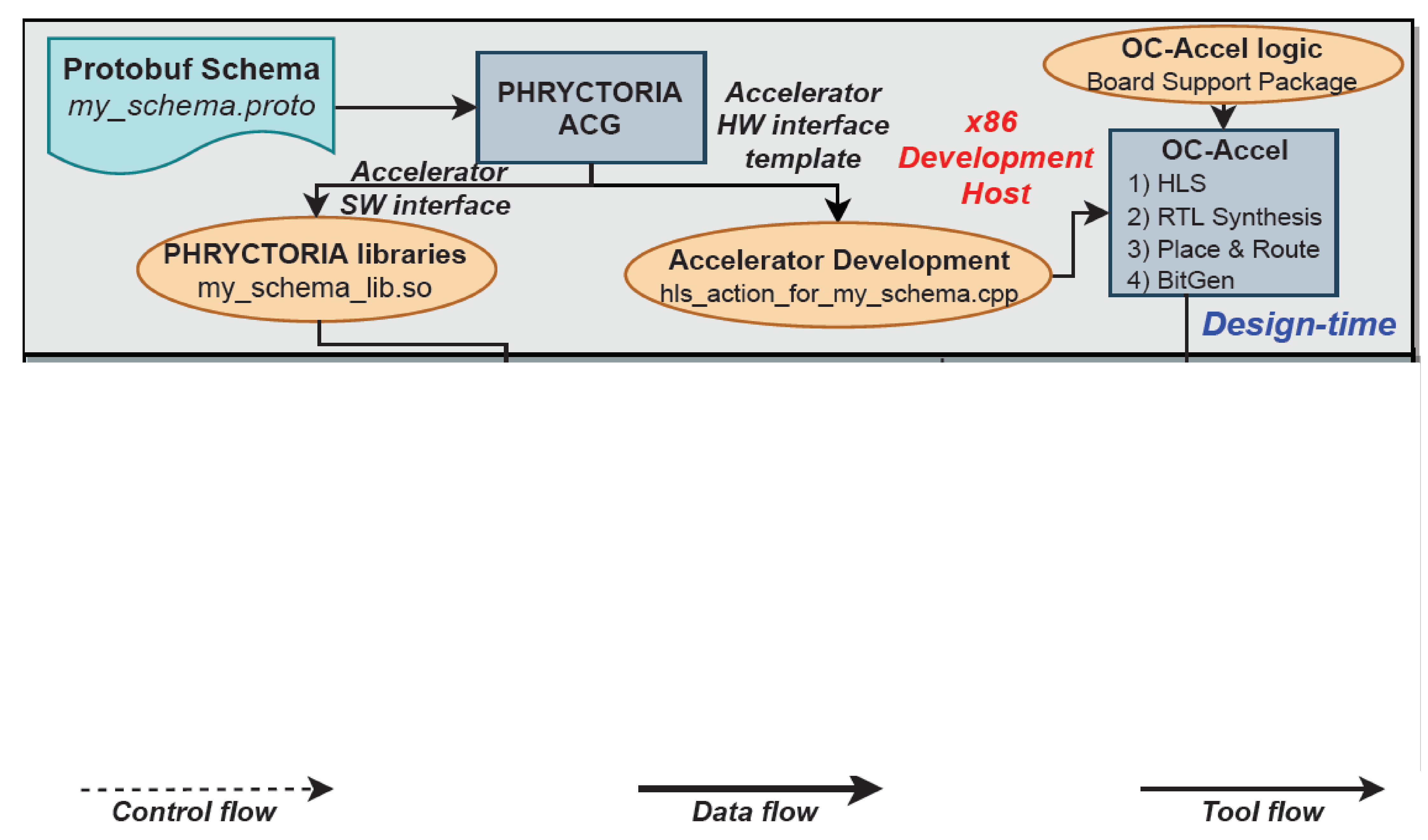
We introduce PHRYCTORIA IDE by extending OC-Accel IDE both in the software and hardware component list.

OC-Accel’s SW is extended by the **PHRYCTORIA Automatic Code Generator**

AND

OC-Accel’s HW is extended by the PHRYCTORIA AFU

PHRYCTORIA Design-time & Run-time System



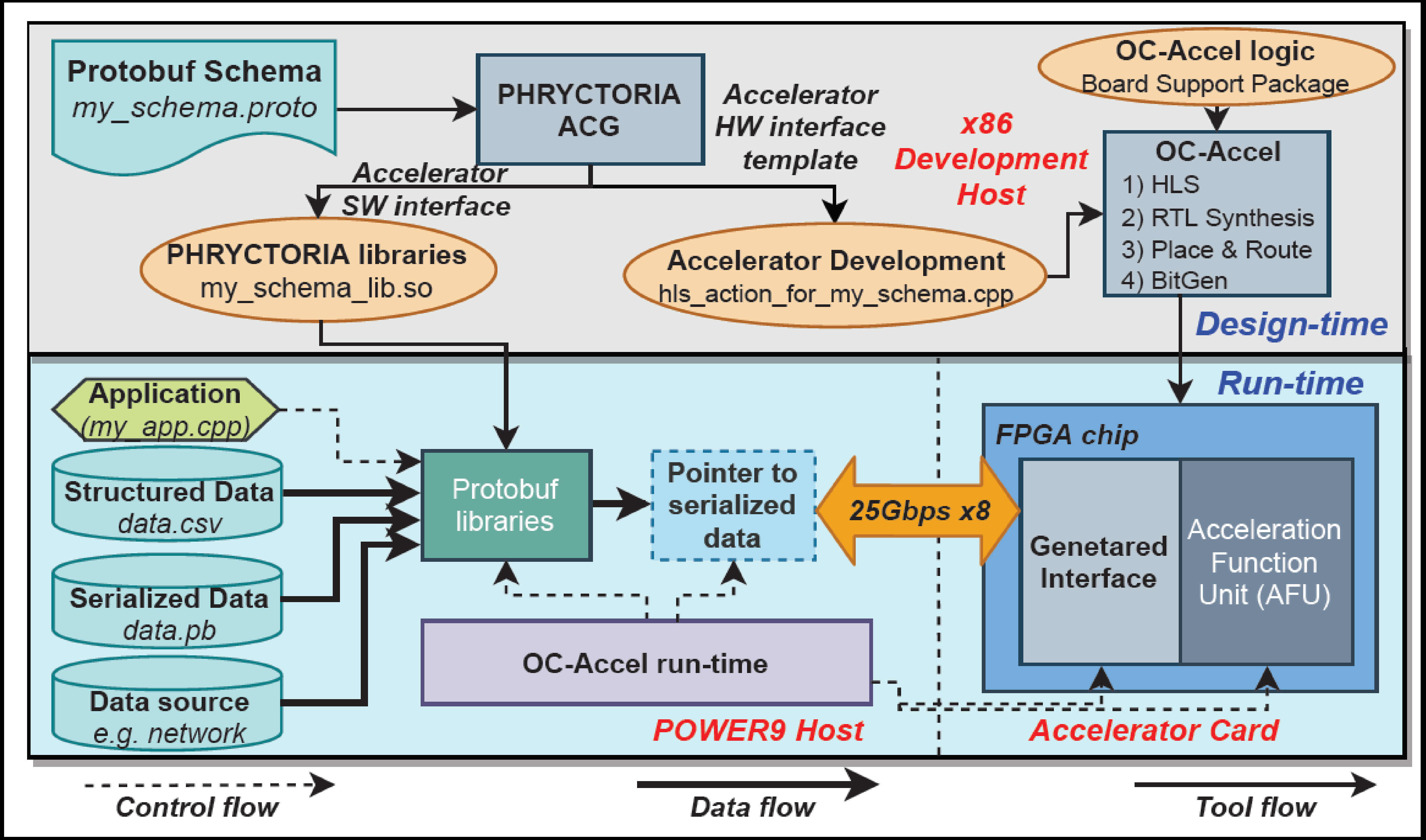
Given a protobuf schema by the user, ACG will generate a customized HW interface template and an Acc. SW interface.

The SW I/F that will provide the functions to send/receive streams of serialized data to/from the FPGA.

The generated HW interface with the OC-Accel logic are synthesized, placed and routed to provide the FPGA bitstream.

The Accelerator SW interface is compiled to a dynamic loadable library for the run-time system.

PHRYCTORIA Design-time & Run-time System



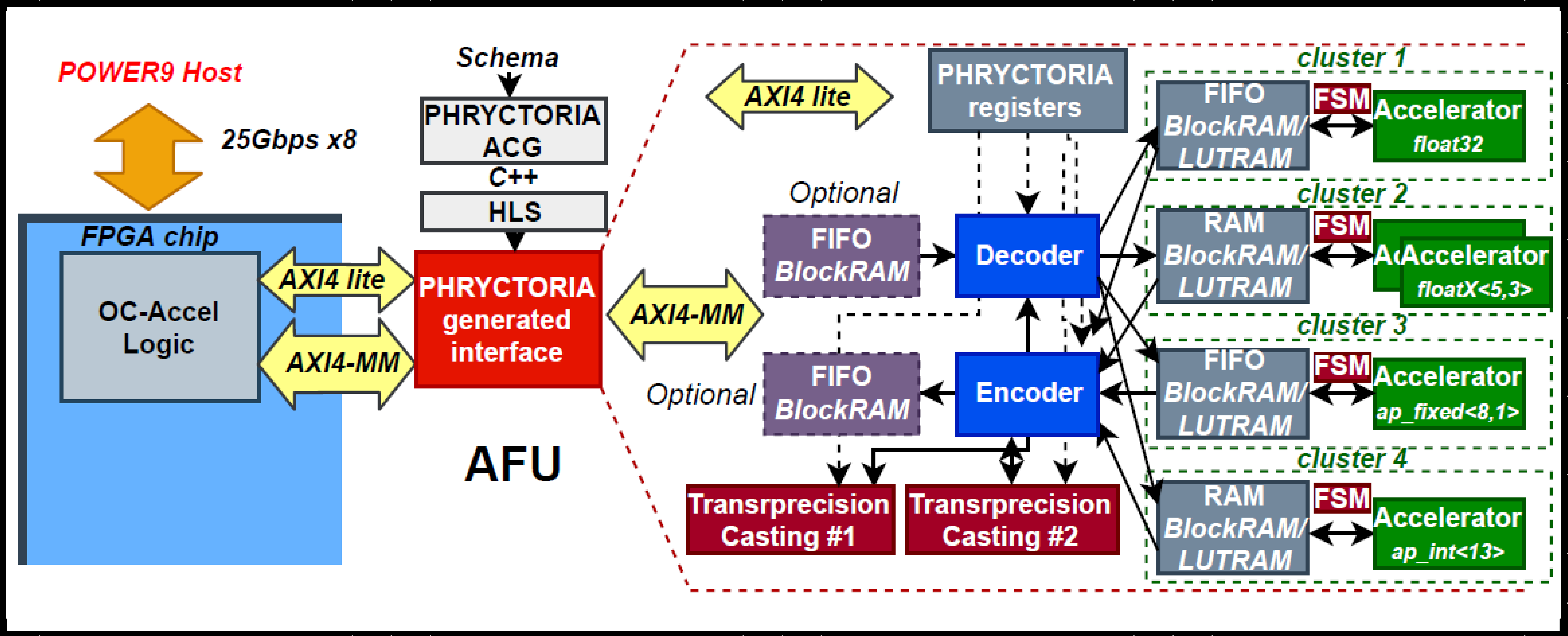
During run-time, an app can send/receive a volume of data to/from the FPGA using the generated SW libs and the protobuf libraries.

After serializing the data with protobuf encoding, a user-space virtual address pointer, pointing to these data, is passed to the OC-Accel runtime system.

This system performs basic discovering, availability and readiness status of the FPGA device, prior to scheduling a “job” descriptor to this device

Runtime can interact with the generated HW interface, via registers, to control the SERDES. All transactions are served over the 25GBps OpenCAPI link.

PHRYCTORIA Internal AFU structure



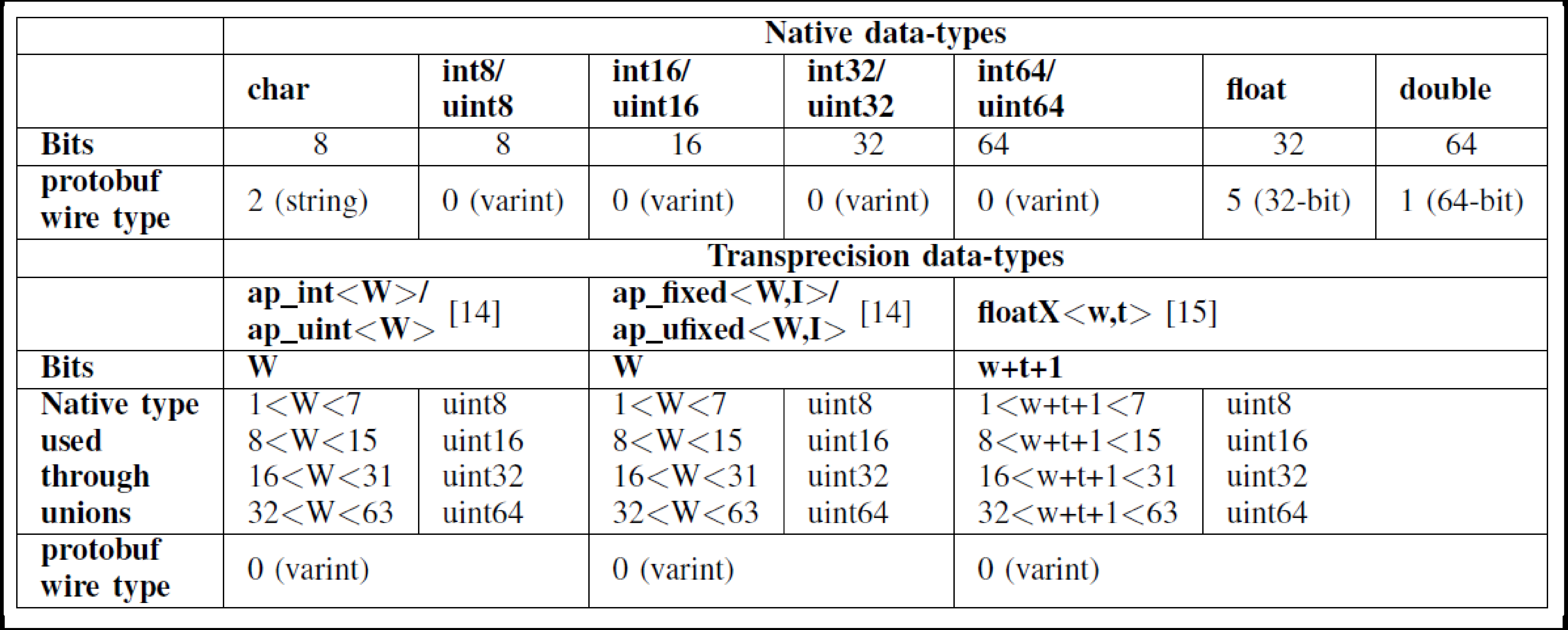
The micro-architecture depicted is designed in mind to scale up to the instances of acc/tors. This is enabled by formulating clusters of accelerators and embedded SRAMs (BRAMs/URAMs).

Each cluster is managed by an attached controller which is an FSM (C++/HLS) responsible for executing the program that run within the cluster. SRAMs can be configured either as RAMs or FIFOs, depending on access.

Decode/Encoder units break large data transfers from AXI4-MM into words and steers them to SRAMs. They are interfaced to the AXI4-MM bus using optional FIFOs in order to absorb the latency of (AXI4 in burst Mode).

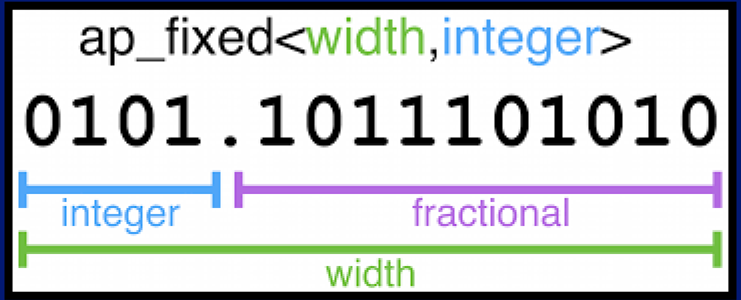
A Transprecision Casting unit may be involved in order to cast the data to the right representation. It is mandatory for all data-types transferred as uniform unsigned integer types (i.e. only native float and double do not need casting).

PHRYCTORIA Supported Transprecision Data-types



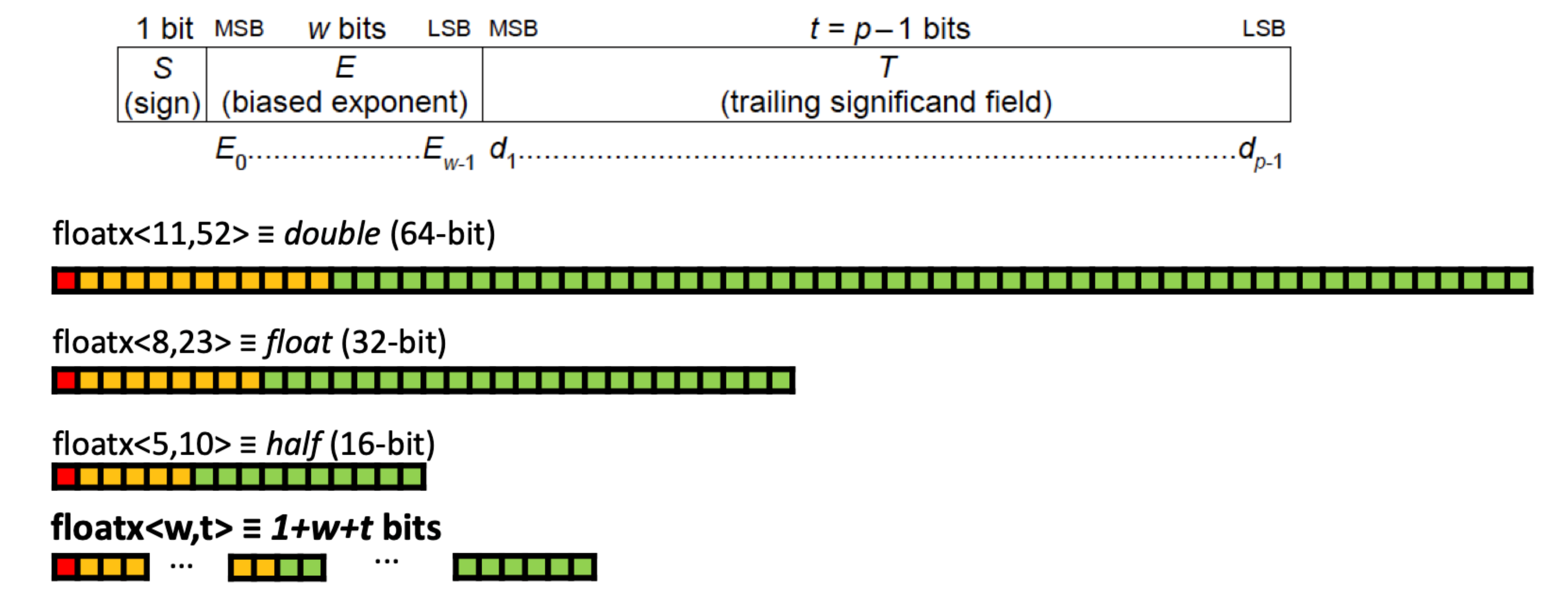
	Native data-types						
	char	int8/ uint8	int16/ uint16	int32/ uint32	int64/ uint64	float	double
Bits	8	8	16	32	64	32	64
protobuf wire type	2 (string)	0 (varint)	0 (varint)	0 (varint)	0 (varint)	5 (32-bit)	1 (64-bit)
	Transprecision data-types						
	ap_int<W>/ ap_uint<W> [14]		**ap_fixed<W,I>/ ap_ufixed<W,I>** [14]		**floatX<w,t>** [15]		
Bits	**W**		**W**		**w+t+1**		
Native type used through unions	1<W<7 8<W<15 16<W<31 32<W<63	uint8 uint16 uint32 uint64	1<W<7 8<W<15 16<W<31 32<W<63	uint8 uint16 uint32 uint64	1<w+t+1<7 8<w+t+1<15 16<w+t+1<31 32<w+t+1<63	uint8 uint16 uint32 uint64	
protobuf wire type	0 (varint)		0 (varint)		0 (varint)		

Arbitrary signed/unsigned integer types (ap int<W>/ap uint<W>) and arbitrary signed/unsigned fixed-point types (ap fixed<W>/ap ufixed<W>) using the HLS arbitrary Precision Types lib.

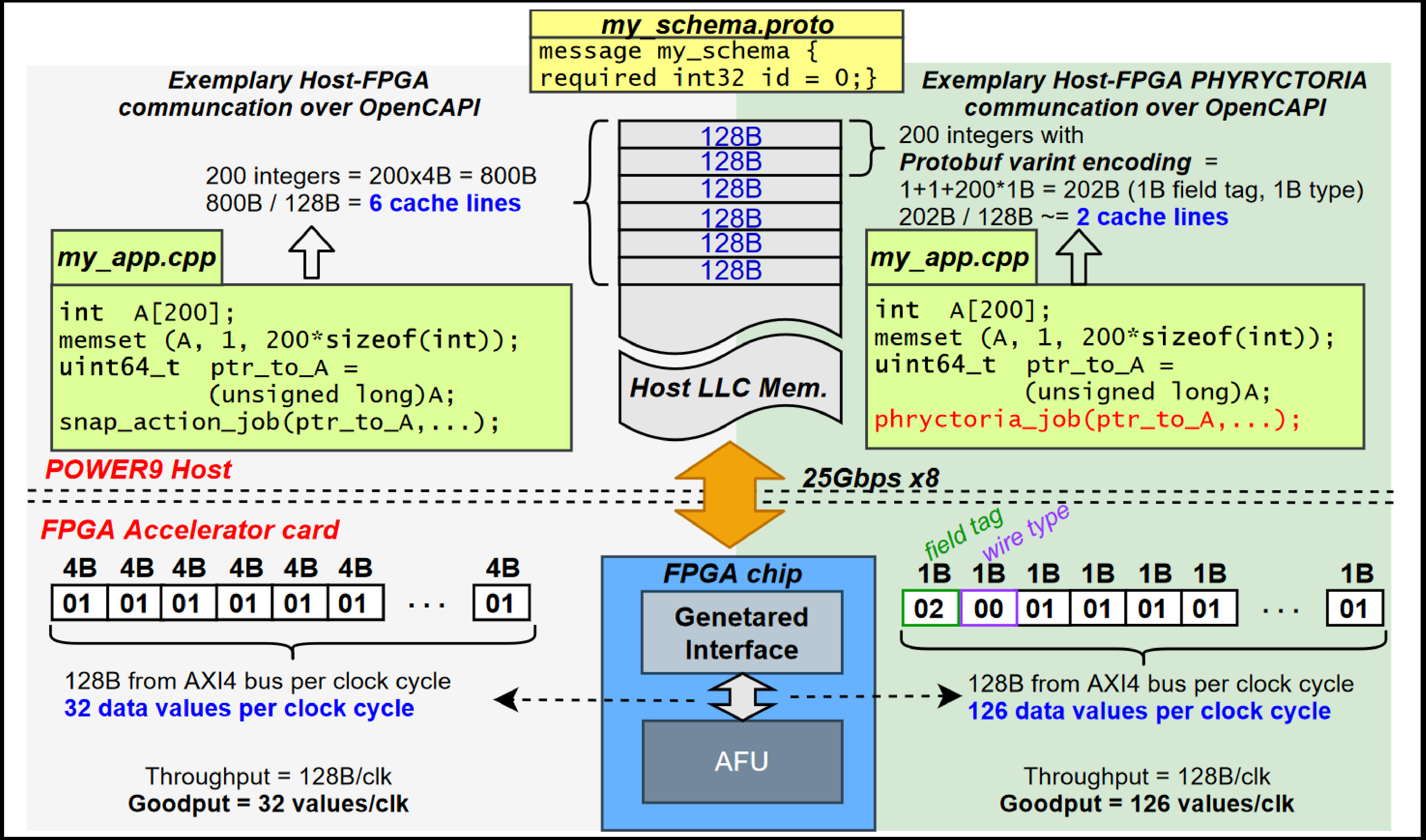


In addition, we support arbitrary floating point numbers using the FloatX library. FloatX emulates the types of custom precision and does not implement arbitrary floating point types.

IEEE 754 floatingpoint standard: $v = (-1)^s * 2^e * m$


1 bit
MSB
w bits
LSB
MSB
t = p−1 bits
LSB
S
(sign)
E
(biased exponent)
T
(trailing significand field)
E0.....................Ew-1
d1.........................dp-1
floatx<11,52> ≡ double (64-bit)
floatx<8,23> ≡ float (32-bit)
floatx<5,10> ≡ half (16-bit)
floatx<w,t> ≡ 1+w+t bits
...
...

How PHRYCTORIA manages to improve the communication goodput . . .



```
message my_schema {
required int32 id = 0;}
```

Exemplary Host-FPGA communcation over OpenCAPI
200 integers = 200x4B = 800B
800B / 128B = 6 cache lines
my_app.cpp

```
int  A[200];
memset (A, 1, 200*sizeof(int));
uint64_t  ptr_to_A =
          (unsigned long)A;
snap_action_job(ptr_to_A,...);
```

128B
128B
128B
128B
128B
128B
Host LLC Mem.
Exemplary Host-FPGA PHYRYCTORIA communcation over OpenCAPI
200 integers with
Protobuf varint encoding =
1+1+200*1B = 202B (1B field tag, 1B type)
202B / 128B ~= 2 cache lines
my_app.cpp

```
int  A[200];
memset (A, 1, 200*sizeof(int));
uint64_t  ptr_to_A =
          (unsigned long)A;
phryctoria_job(ptr_to_A,...);
```

POWER9 Host
25Gbps x8
FPGA Accelerator card
4B 4B 4B 4B 4B 4B 4B
01 01 01 01 01 01 . . . 01
128B from AXI4 bus per clock cycle
32 data values per clock cycle
Throughput = 128B/clk
Goodput = 32 values/clk
FPGA chip
Genetared Interface
AFU
field tag
wire type
1B 1B 1B 1B 1B 1B 1B
02 00 01 01 01 01 . . . 01
128B from AXI4 bus per clock cycle
126 data values per clock cycle
Throughput = 128B/clk
Goodput = 126 values/clk

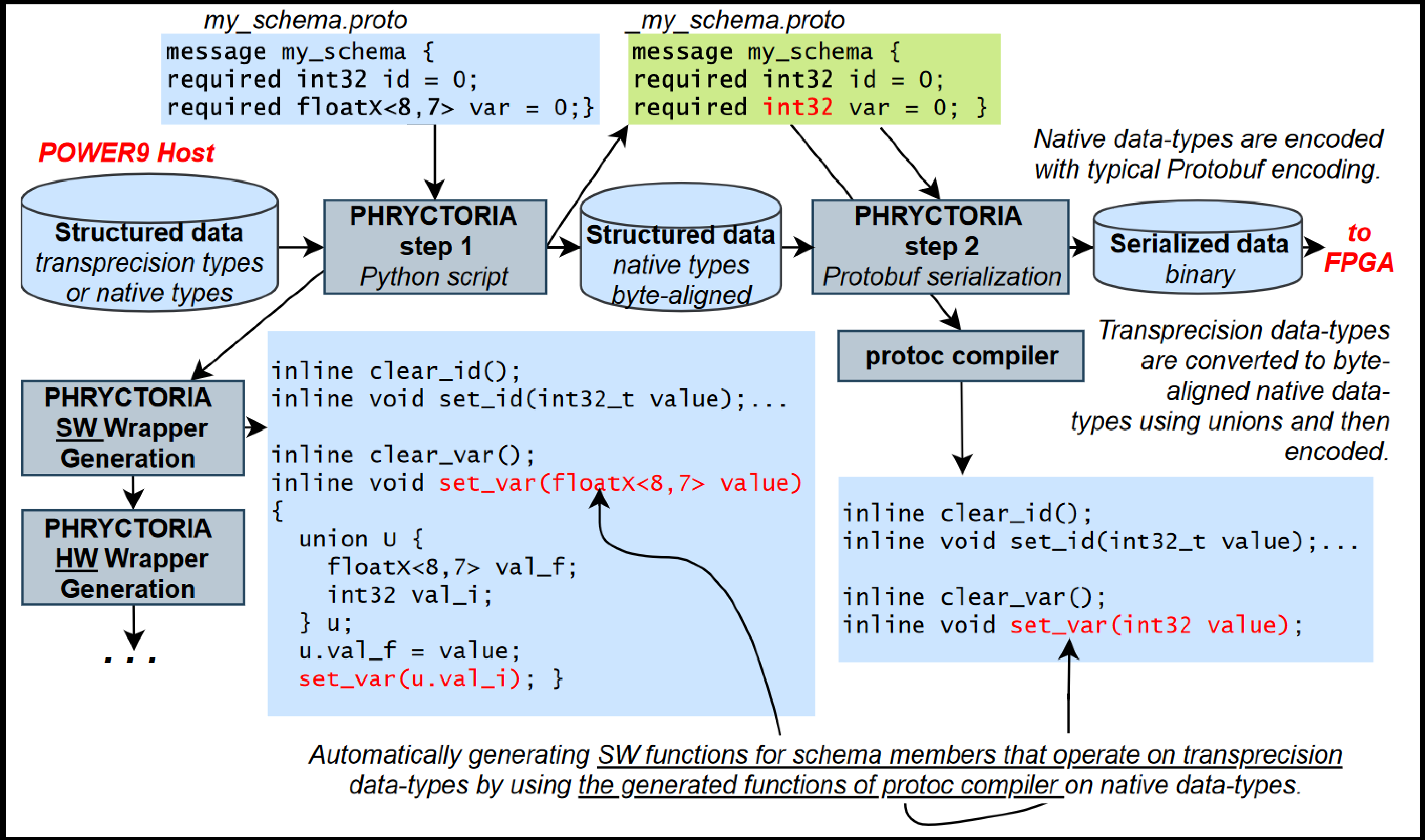
We depict a naive example of transmitting an array-A of 200 integers with value “01”, from the host to the FPGA.

Left: OpenCAPI’s DL interface operates on the LLC of the processor and as such the array-A is fetched in packed data of 128 Bytes, (POWER9’s cache line).

Right: Serialization function (automatically generated by ACG) will send 202 Bytes, i.e. 200 Bytes for the small integers and two extra bytes for pb headers.

The serialized data can fit in just 2 lines. On the FPGA side, cache line delivers 126 elements (128-2 for headers) i.e. 126/32=3.9x improvement / clk.

PHRYCTORIA Automatic Code Generator (ACG)



my_schema.proto

```
message my_schema {
required int32 id = 0;
required floatX<8,7> var = 0;}
```

_my_schema.proto

```
message my_schema {
required int32 id = 0;
required int32 var = 0; }
```

Native data-types are encoded with typical Protobuf encoding.

Transprecision data-types are converted to byte-aligned native data-types using unions and then encoded.

```
inline clear_id();
inline void set_id(int32_t value);...

inline clear_var();
inline void set_var(floatX<8,7> value)
{
  union U {
    floatX<8,7> val_f;
    int32 val_i;
  } u;
  u.val_f = value;
  set_var(u.val_i); }
```

```
inline clear_id();
inline void set_id(int32_t value);...

inline clear_var();
inline void set_var(int32 value);
```

Automatically generating SW functions for schema members that operate on transprecision data-types by using the generated functions of protoc compiler on native data-types.

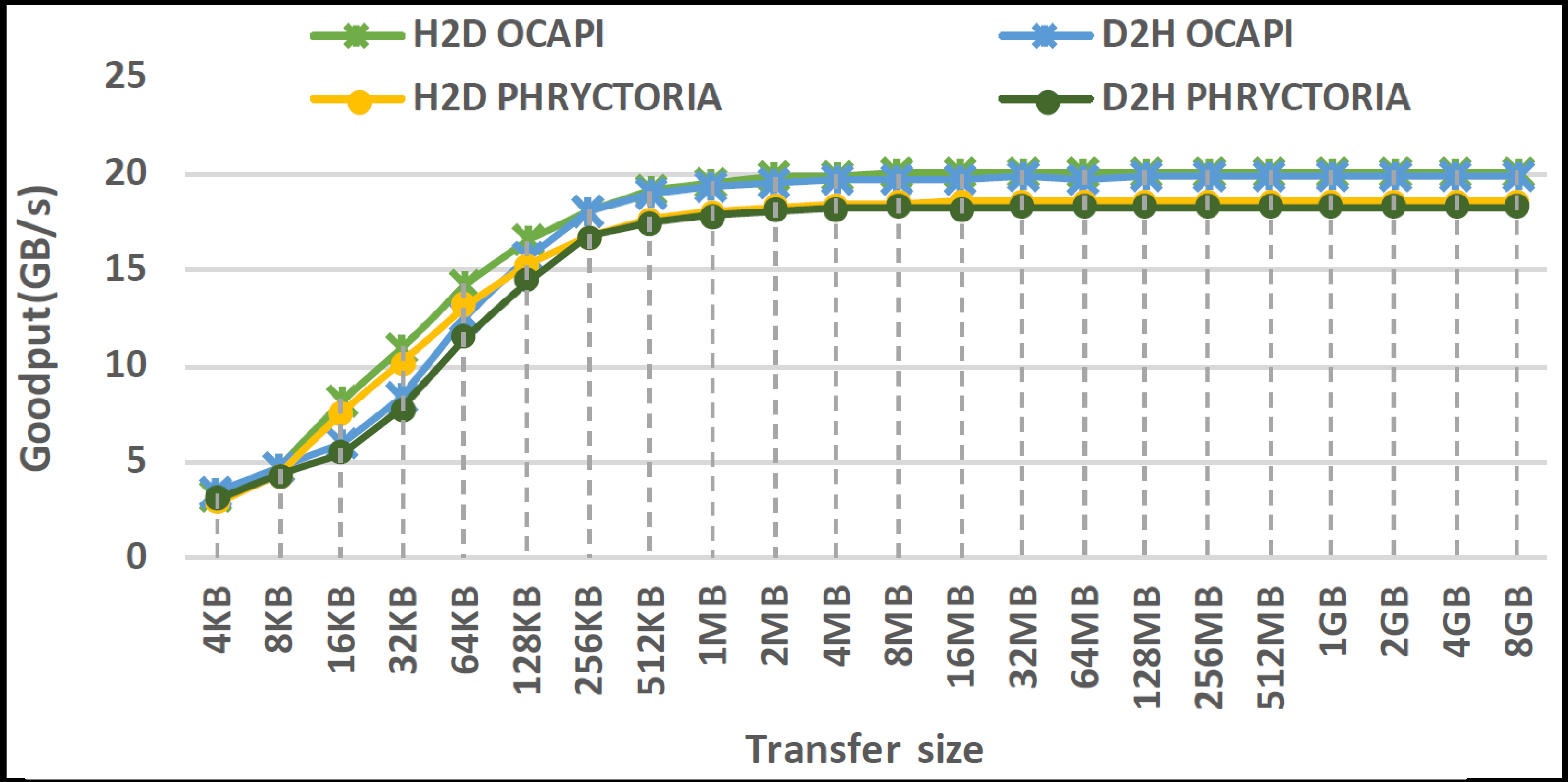
e.g. host sends to the FPGA messages with two data-types, a native supported int32 and a transprecision floatX<8,7>.

Protobuf supports the serialization of native data-types & the automatic generation of the SW functions that manipulate data in this schema through protoc.

A Python script parses the protobuf schema as a 1st step and converts the transprecision datatypes to byte-aligned native ones (to be processed by protoc)

For every generated function of protoc and only the converted data-types, a wrapper is generated to facilitate the conversion using unions.

PHRYCTORIA Evaluation: Synthetic Data Set



Allocate different sizes of native data-types on the host and either send as structured byte-aligned data to FPGA over OCAPI, or firstly serializes and then sends as pb stream.

The evaluation was realized for both host and FPGA acting as Rx and Tx, H2D and D2H. The intention was to show that the OpenCAPI throughput is not degraded for native data-types.

The figure depicts the average for all native data-types, as the deviation for the seven data-types was marginally 2.3% for a sequence of 1k repeated tests.

The bandwidth is not severely degraded for native datatypes (0.92%). We intentionally loaded data that could not benefit from the *varint,* by setting at least one bit on all bytes of every native type.

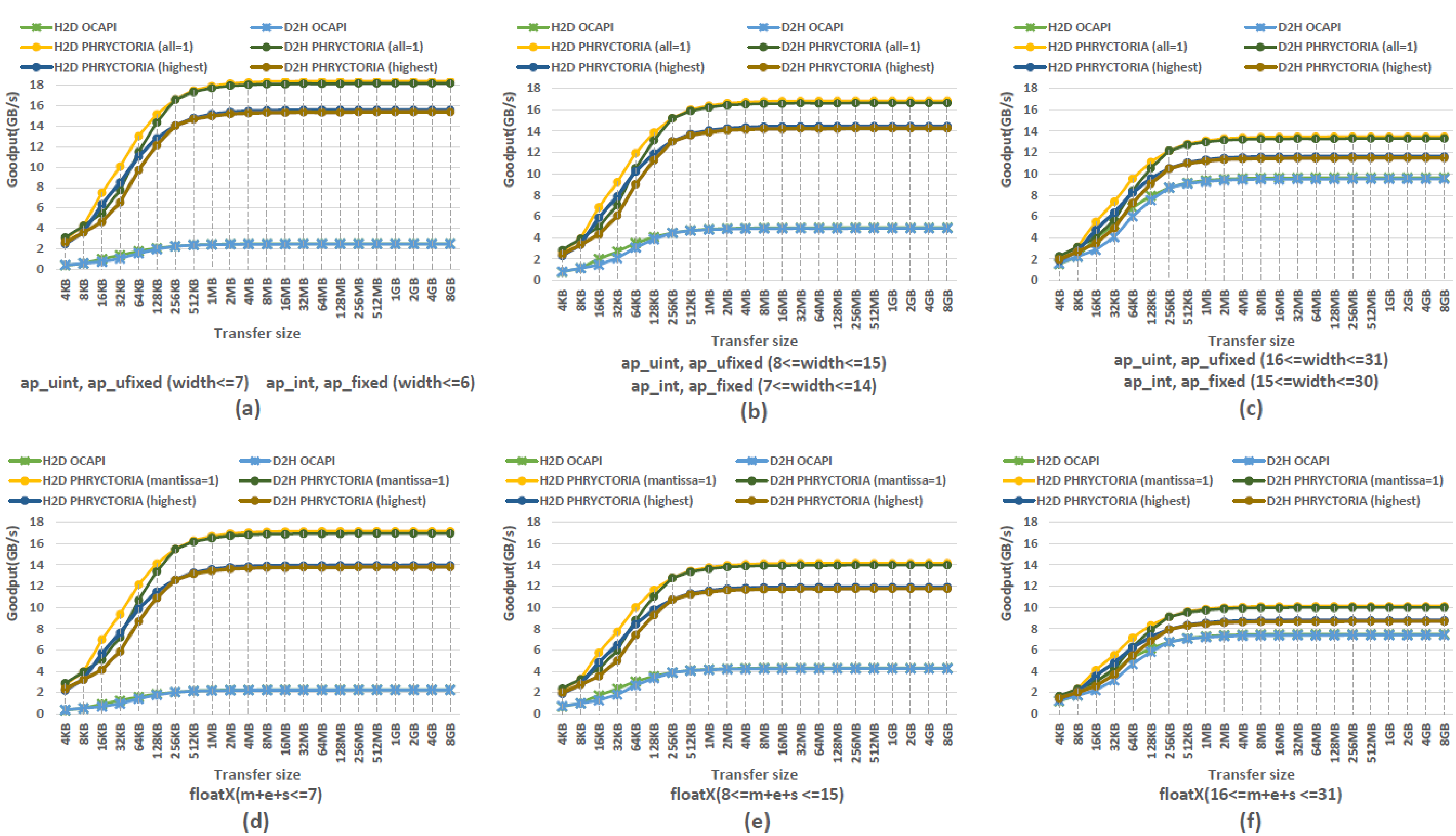
PHRYCTORIA Evaluation: Transprecision Data Set

6.3-7.4x

for < 8 bits

2.9-3.4x

for 8-15 bits



ap_uint, ap_ufixed (width<=7) ap_int, ap_fixed (width<=6)

(a)

ap_uint, ap_ufixed (8<=width<=15)
ap_int, ap_fixed (7<=width<=14)

(b)

ap_uint, ap_ufixed (16<=width<=31)
ap_int, ap_fixed (15<=width<=30)

(c)

floatX(m+e+s<=7)

(d)

floatX(8<=m+e+s <=15)

(e)

floatX(16<=m+e+s <=31)

(f)

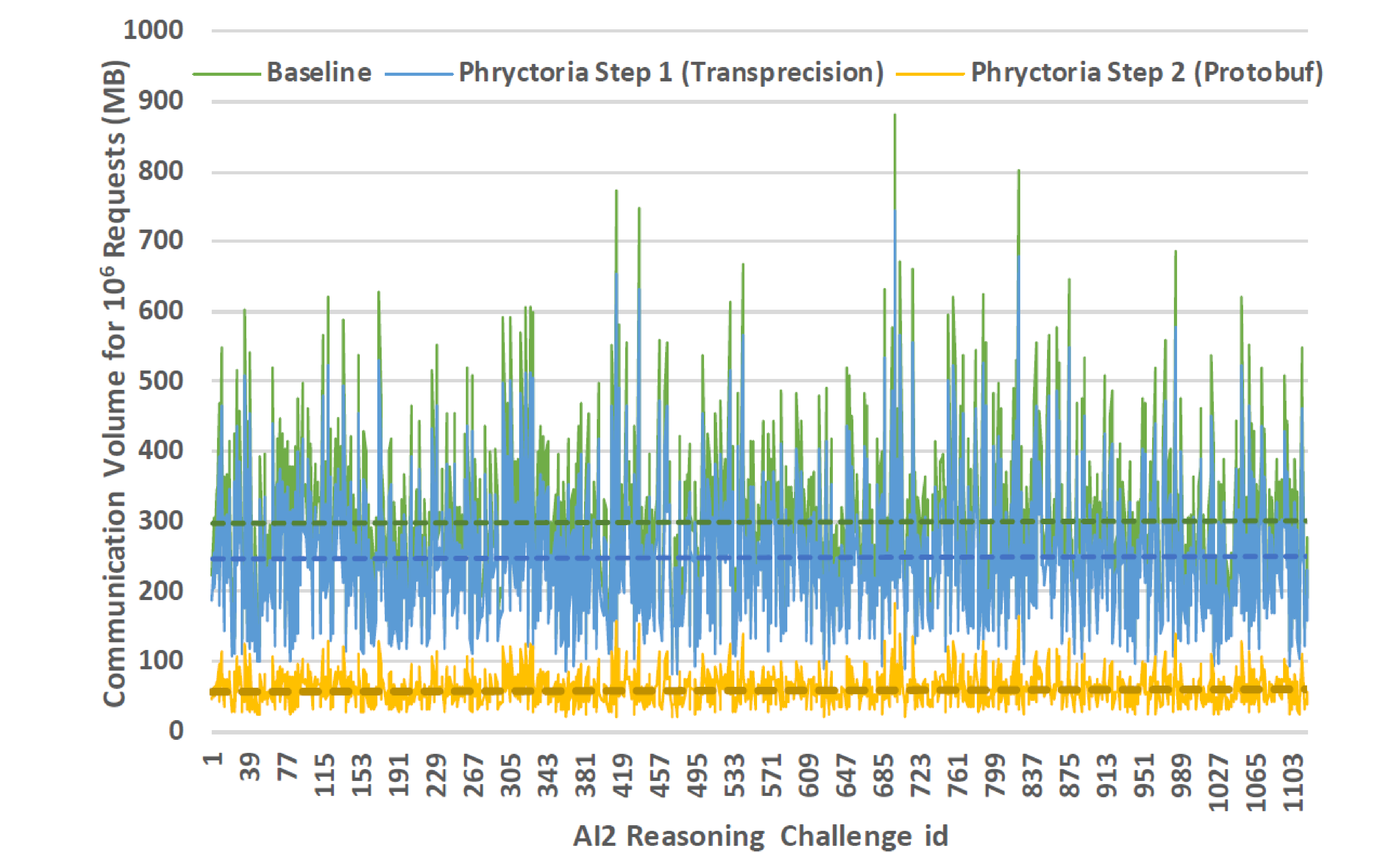
1.2-1.4x

for 16-32 bits

! -9.4x - -13.4x

for 64 bits

PHRYCTORIA Evaluation: Realistic NLP Data Set - AI2 Reasoning



Accelerators for an NLP domain that need to access the respective dataset over the OpenCAPI link.

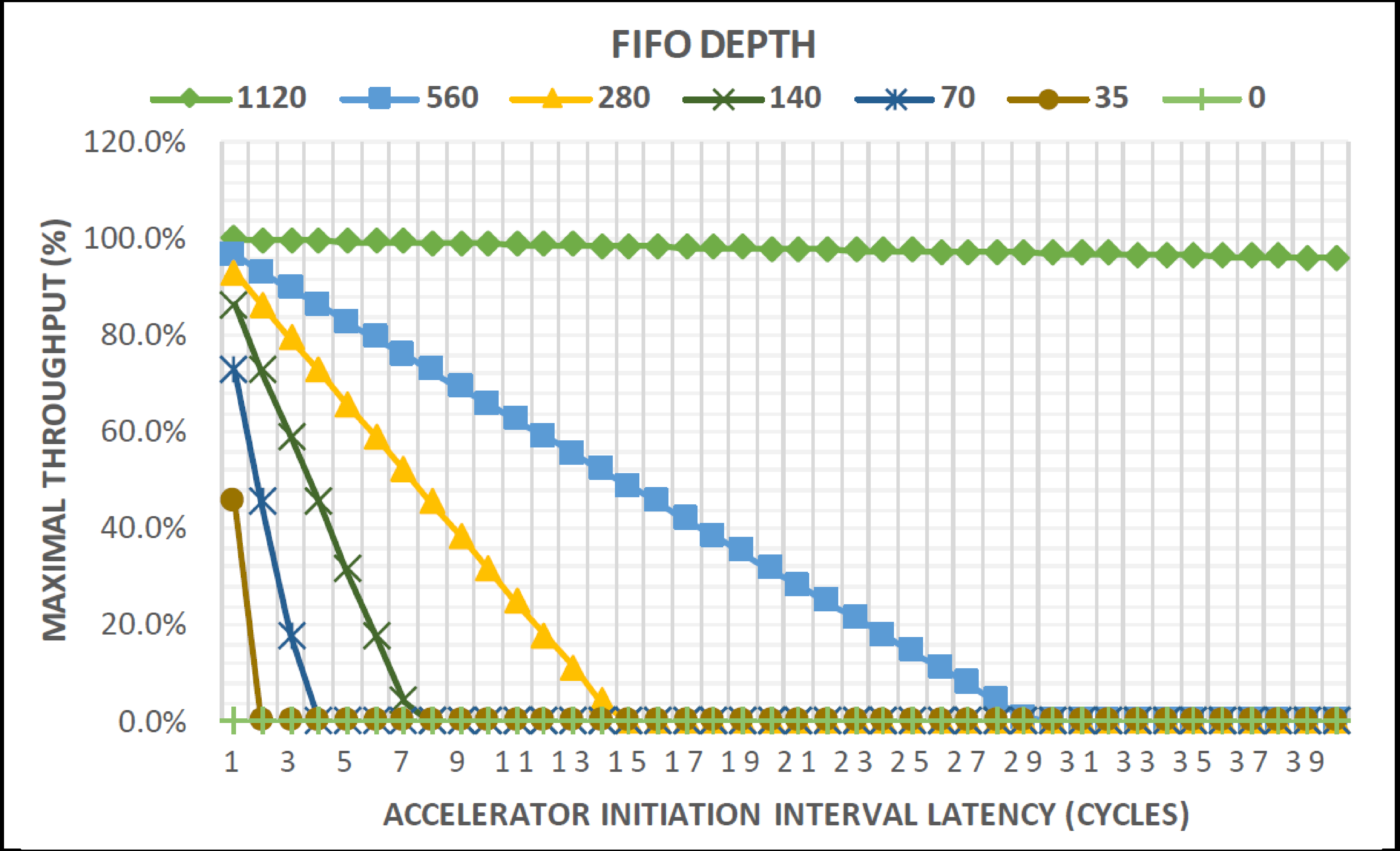
Allen Institute for AI Reasoning Challenge “data-set, which contains 7,787 genuine school-grade level, multiple-choice science questions, assembled to encourage research in adv. question answering.

From the 12 columns of data, 7 can be represented with unsigned integers of 8-bits, 2 with strings and 3 with 16 bits. The second gain comes directly from the varint encoding.

The data-set was reduced from 298.8 MB to 250.3 MB, from the first step, and 61.1 MB from serialization.

6.9x Goodput gain

PHRYCTORIA adaptation to accelerator's data-flow.



An important parameter of PHRYCTORIA HW design is the depth of the optional 128B-wide FIFOs between the AXI4-MM AFU interface and decoder/encoder.

These FIFOs can absorb the latency of the accelerator and keep the AXI4 bus operating in burst mode.

An accelerator with II=2 consumes 1 AXI words per 2 cycles. But VHLS scheduler is not able to keep a burst of AXI4 and 2 full AXI4 TX are issued.

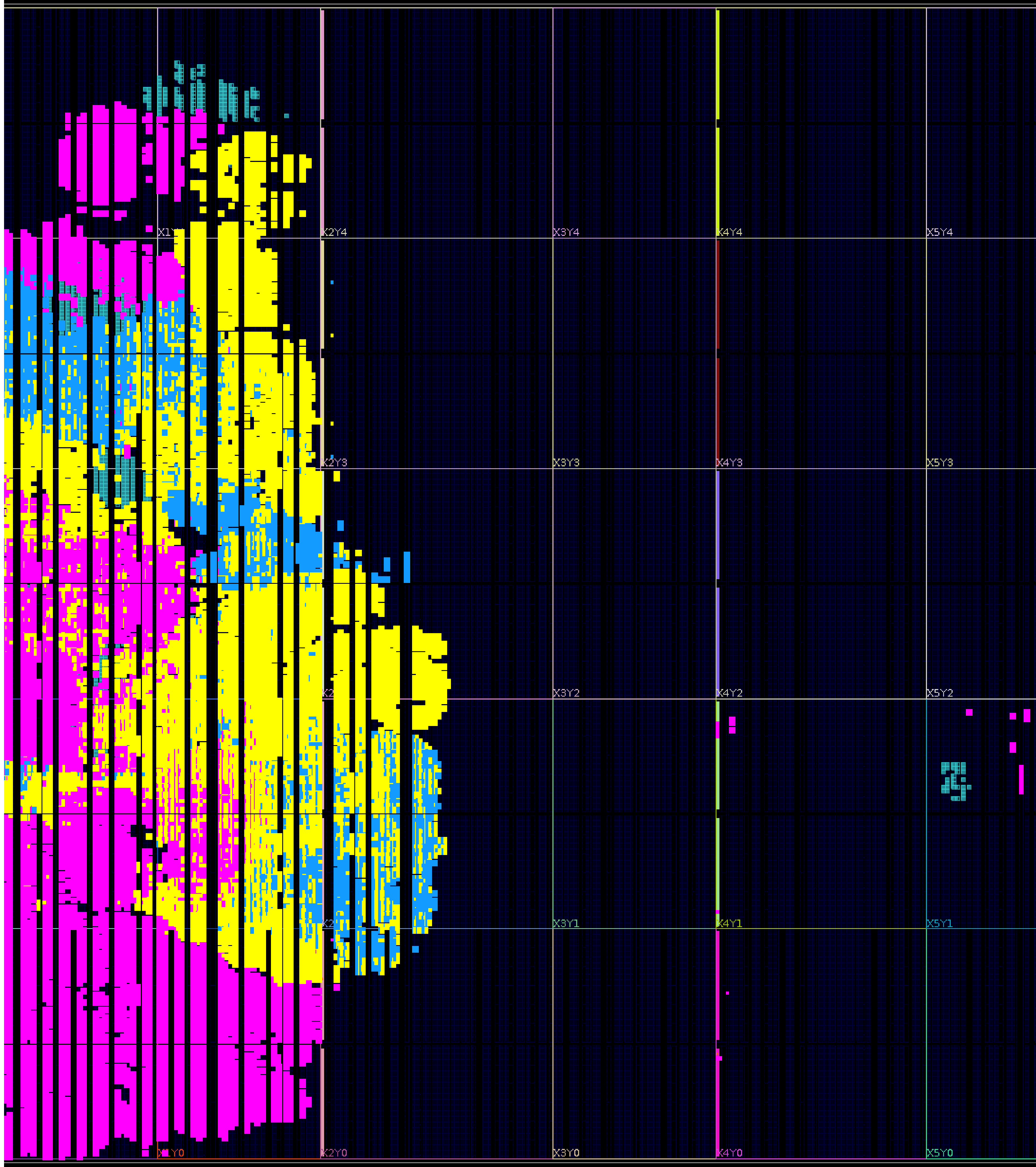
The FIFOs save temporally consequent AXI4 words per cycle in order to saturate the OpenCAPI-AXI4 BW.

PHRYCTORIA Logic Utilization

AD9V3, Xilinx VU3P FPGA - 251.775MHz

73%

empty space to fit your accelerators



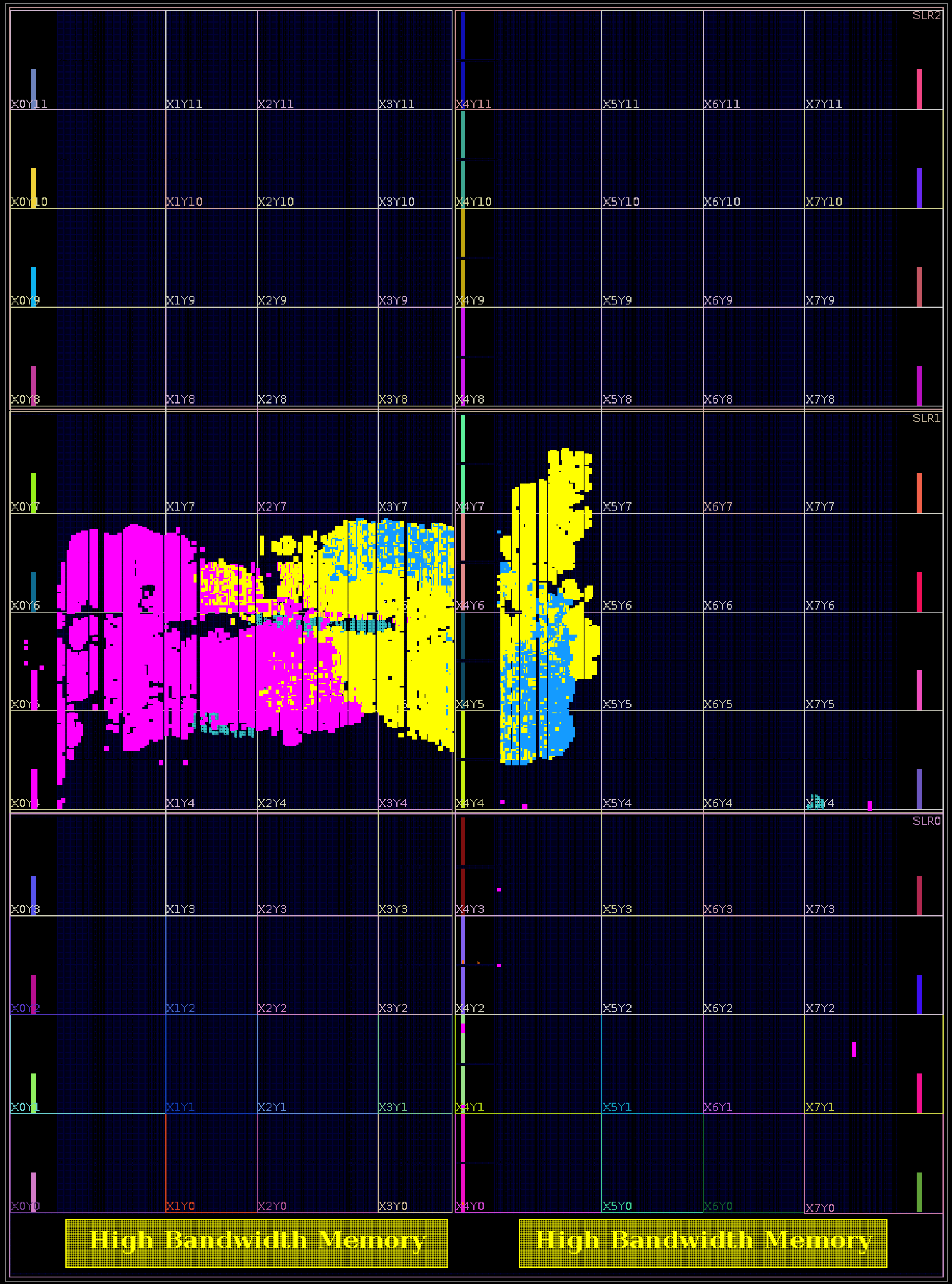
PHRYCTORIA Logic Utilization

AD9H7, Xilinx VU37P FPGA- 251.775MHz

87%

empty space to fit your accelerators

AD9H7 impressively met the timing constraint of 2.482375ns-402.840MHz



Take-away

- ❑ We explored the possibility to sustain high useful throughput on systems with OpenCAPI-enabled FPGAs for emerging transprecision data-types.
- ❑ By adopting the widely used protobuf IDL description and the OC-Accel framework we built a system, named PHRYCTORIA, that automatically generates the appropriate interfaces on a host SW and on FPGA HW.
- ❑ We showed that the inherent varint encoding of protobuf used in PHRYCTORIA sustains an effective throughout similar to the practical best of OpenCAPI for
  - ○ synthetic datasets with uniform transprecision data-types and
  - ○ a real NLP data-set with combined transprecision data-types

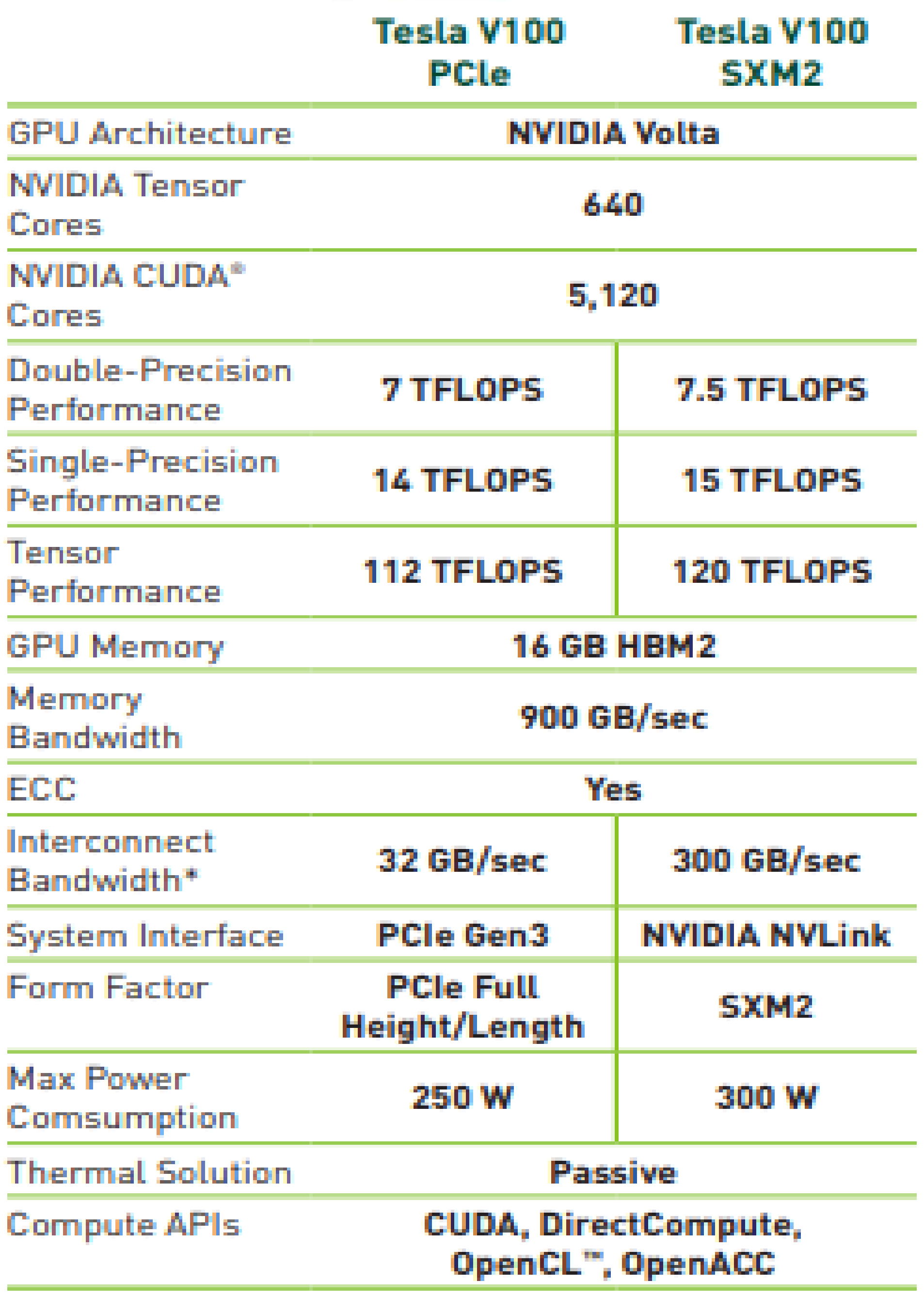
– Future work

- Integrate with accelerators from different domains (linear algebra, quantitative finance, computer vision, DSP.)
- Support synthesizable versions of arbitrary floating point.
- Support posits arithmetic system.
- Support FPGA on-board DRAM and FPGA in-package HBM.

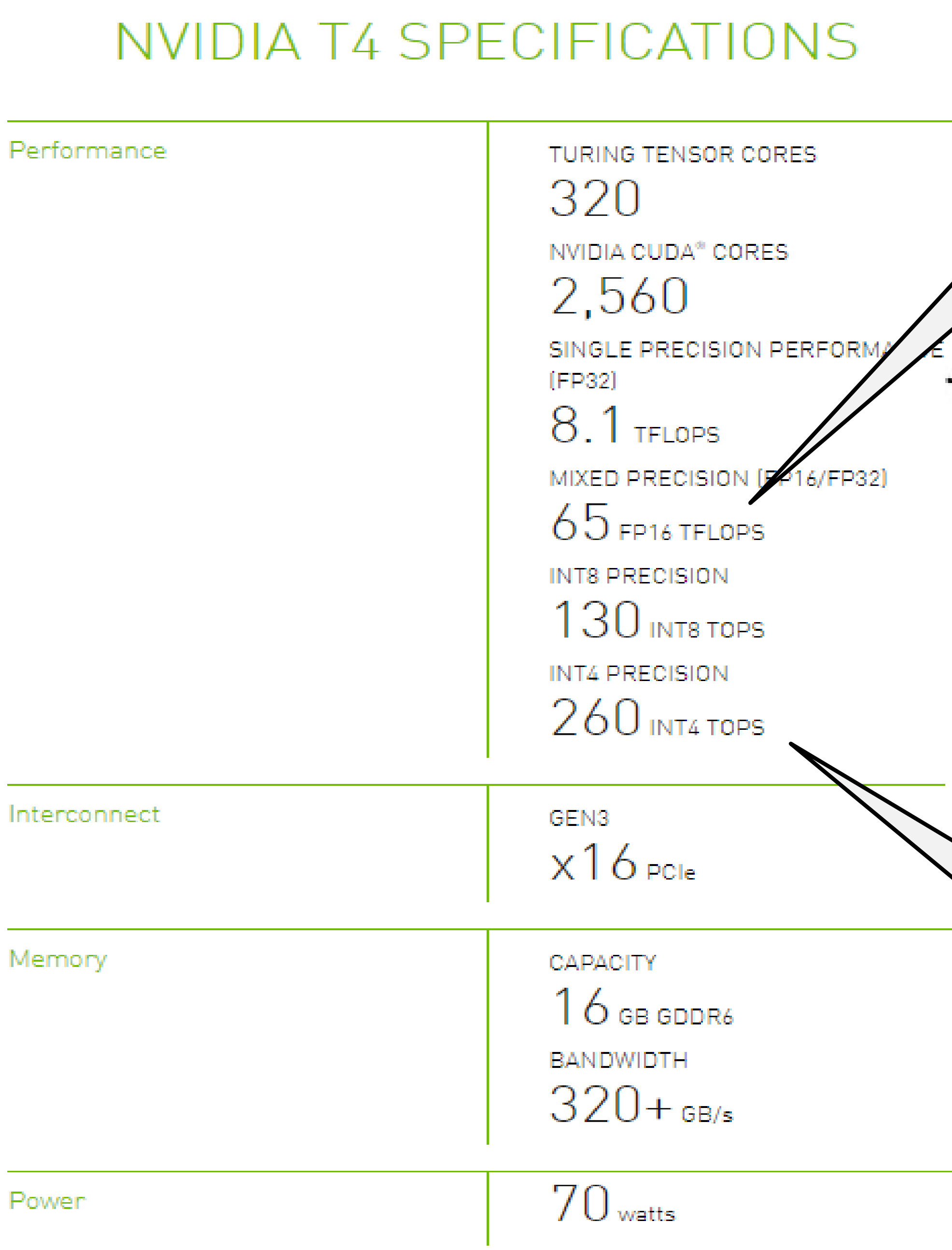
Thank you

Dionysios Diamantopoulos
—
did@zurich.ibm.com
https://researcher.watson.ibm.com/researcher/view.php?person=zurich-DID

AC922 Nvidia V100 vs. IC922 Nvidia T4



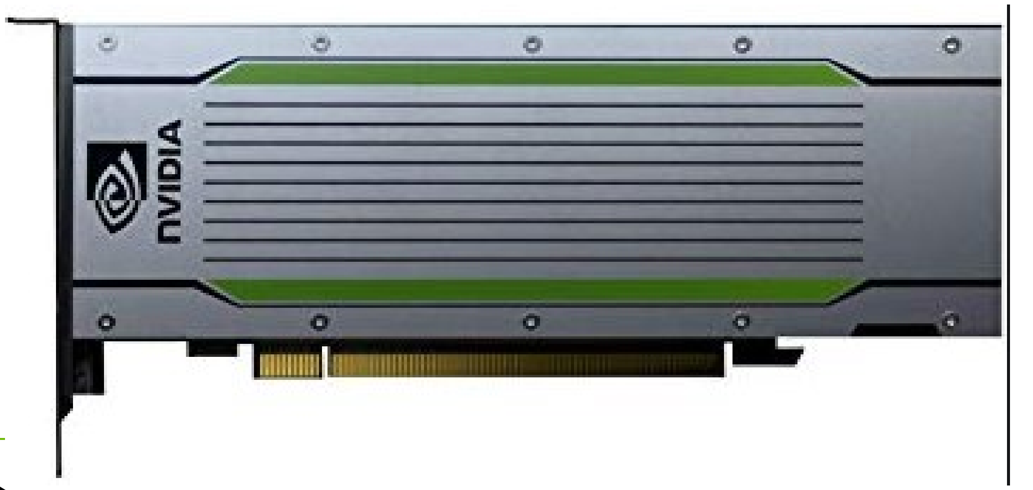
	Tesla V100 PCle	Tesla V100 SXM2
GPU Architecture	NVIDIA Volta	
NVIDIA Tensor Cores	640	
NVIDIA CUDA® Cores	5,120	
Double-Precision Performance	7 TFLOPS	7.5 TFLOPS
Single-Precision Performance	14 TFLOPS	15 TFLOPS
Tensor Performance	112 TFLOPS	120 TFLOPS
GPU Memory	16 GB HBM2	
Memory Bandwidth	900 GB/sec	
ECC	Yes	
Interconnect Bandwidth*	32 GB/sec	300 GB/sec
System Interface	PCIe Gen3	NVIDIA NVLink
Form Factor	PCIe Full Height/Length	SXM2
Max Power Comsumption	250 W	300 W
Thermal Solution	Passive	
Compute APIs	CUDA, DirectCompute, OpenCL™, OpenACC	



NVIDIA T4 SPECIFICATIONS

Performance	TURING TENSOR CORES 320 NVIDIA CUDA® CORES 2,560 SINGLE PRECISION PERFORMANCE (FP32) 8.1 TFLOPS MIXED PRECISION (FP16/FP32) 65 FP16 TFLOPS INT8 PRECISION 130 INT8 TOPS INT4 PRECISION 260 INT4 TOPS
Interconnect	GEN3 x16 PCIe
Memory	CAPACITY 16 GB GDDR6 BANDWIDTH 320+ GB/s
Power	70 watts

V100 FP16 performance is 31 TFLOPS – ½ of T4's



T4's INT4 performance is 260 TOPS – 2x of T4's INT8 TOPS

- T4 Specs: https://www.nvidia.com/en-us/data-center/tesla-t4/
- V100 Specs: https://www.nvidia.com/en-us/data-center/v100/