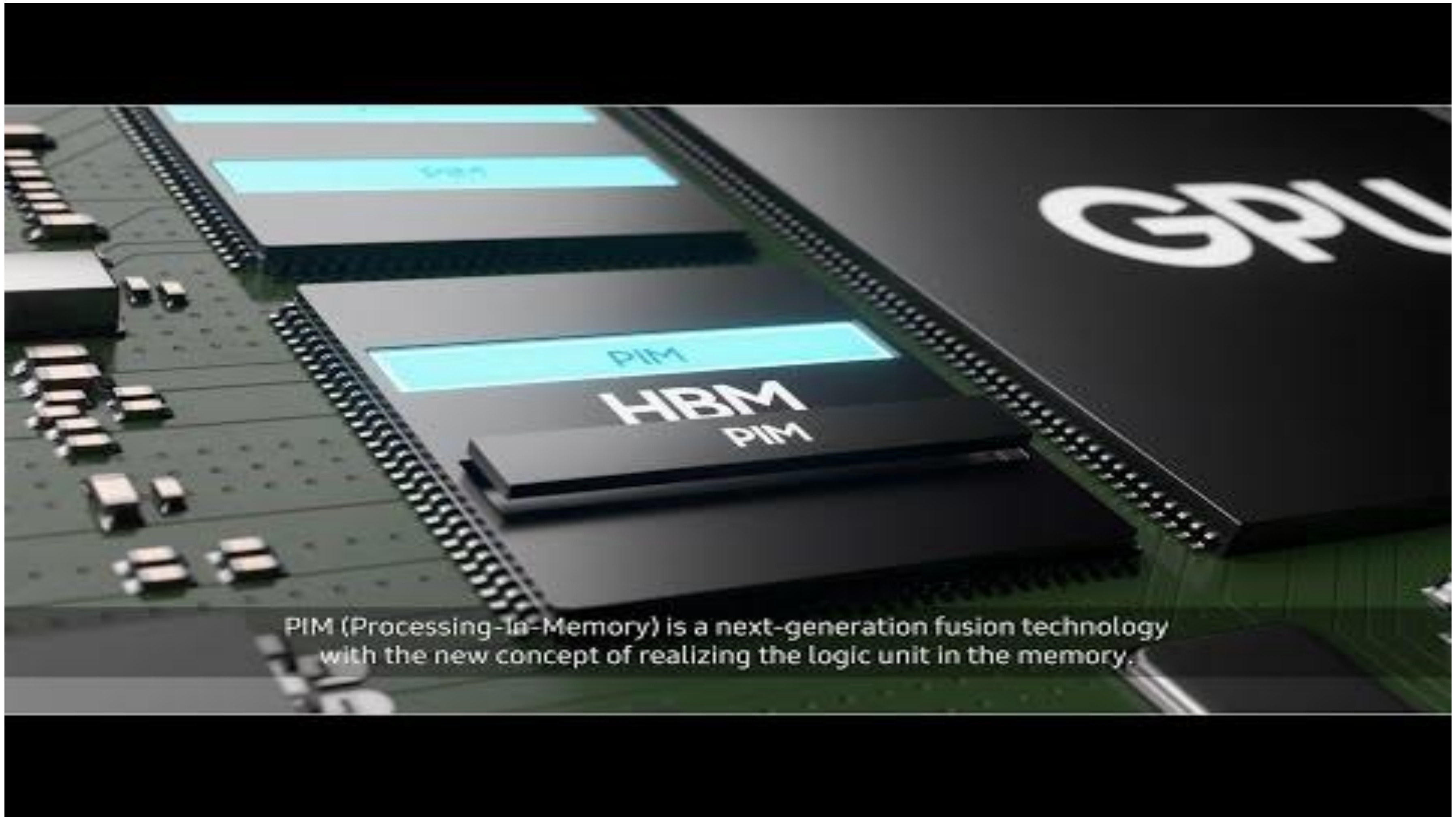
Introducing Real-world HBM-PIM Powered System for Memory-bound Applications

Samsung Electronics
DRAM Design Team
Sukhan Lee

Introducing HBM-PIM

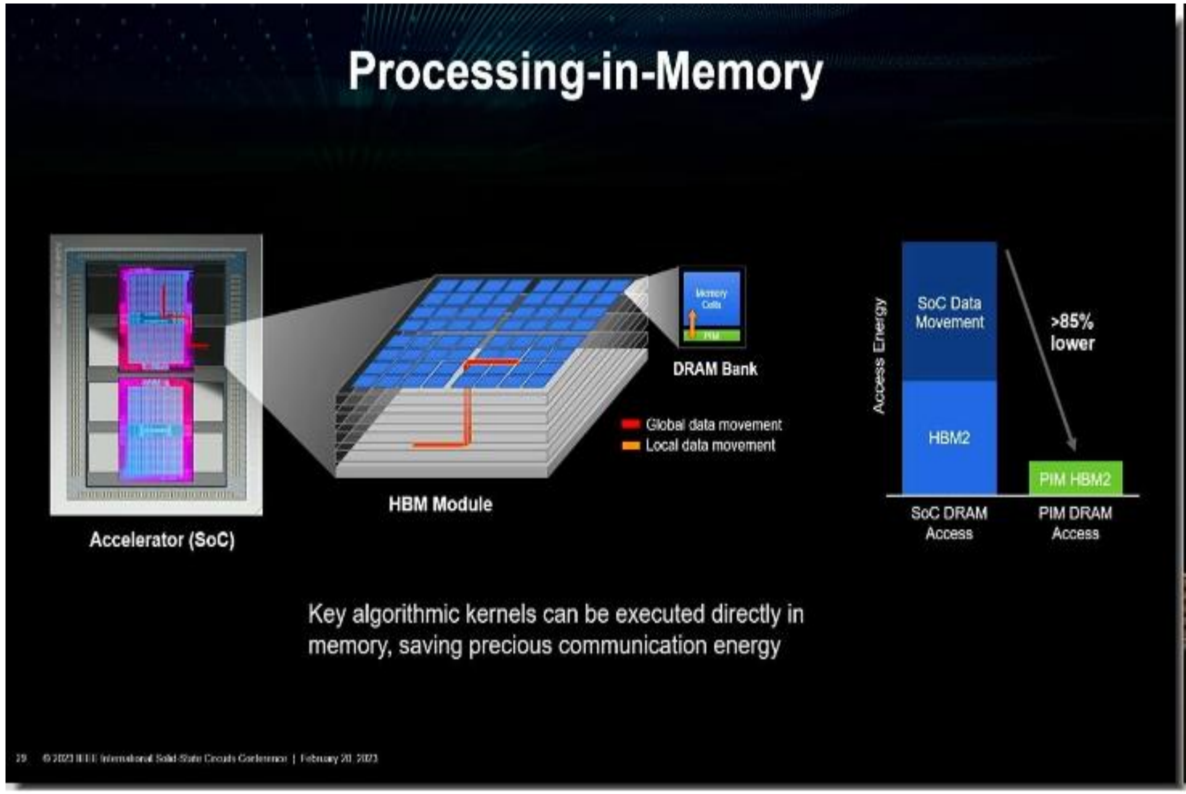
- Video: Samsung Electronics Semiconductor Unveils Cutting-edge Memory Technology to Accelerate Next-generation AI | Samsung Semiconductor Global (youtube)



PIM (Processing-In-Memory) is a next-generation fusion technology with the new concept of realizing the logic unit in the memory.

Introducing HBM-PIM

- AMD, access energy efficiency can be improved by executing the main algorithm kernel directly in memory
- By the presentation of Dr. Lisa Su from AMD, the energy reduction would be 85% by PIM compared to utilizing conventional HBM, and this reduction is more effective than our evaluation (~70%) because SEC did not include host-side energy reduction effect.

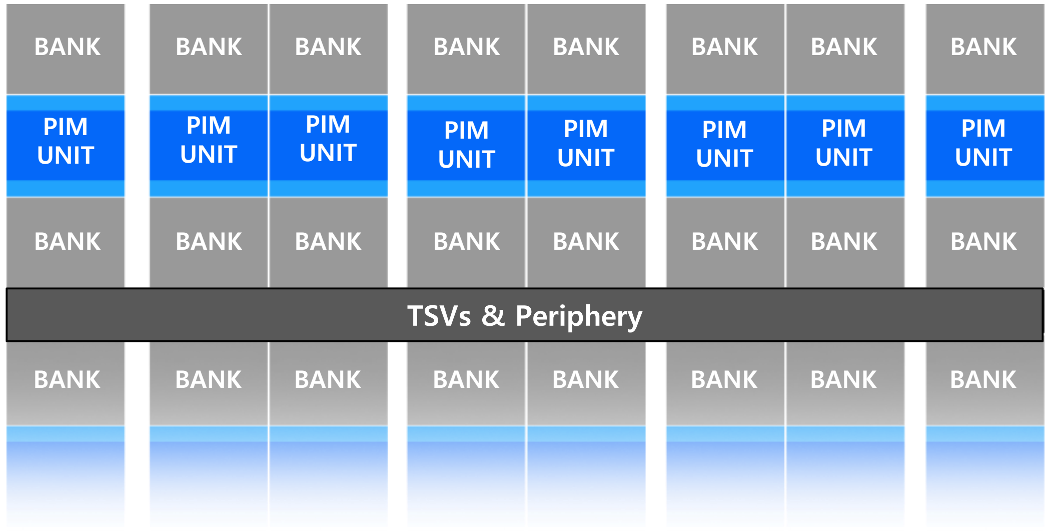


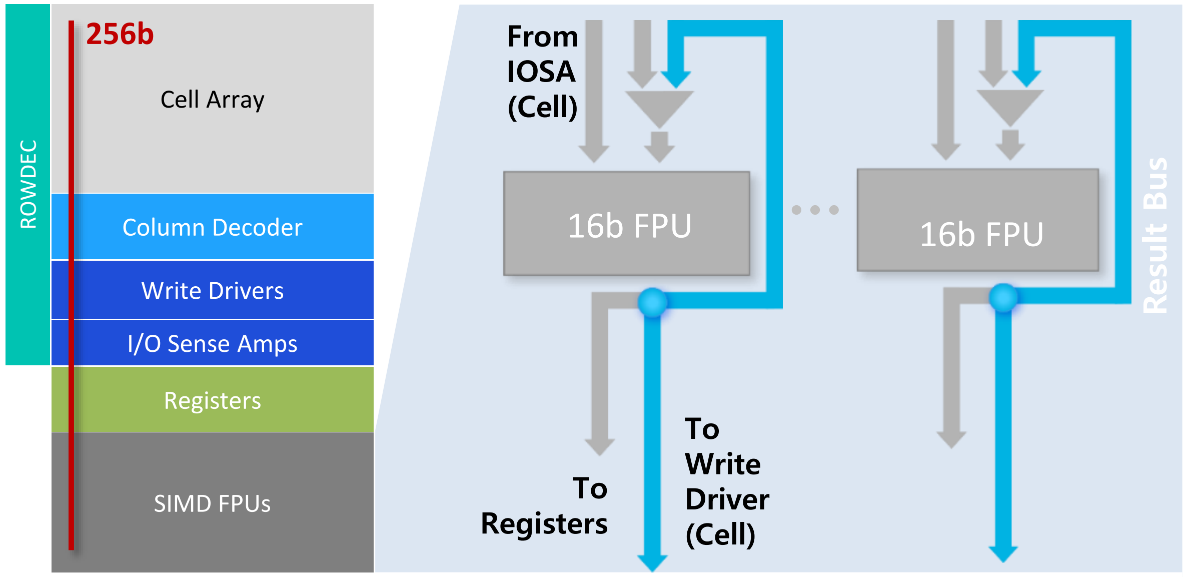
Contents

The Architecture of PIM DRAM-die and Aquabolt-XL [ISCA2021, ISSCC2021]

Aquabolt-XL HBM2-PIM Architecture

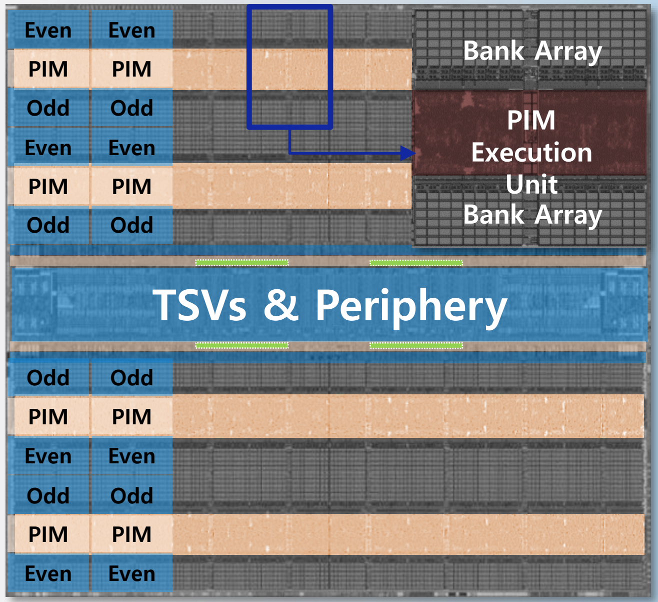
- PIM concept: Place a programmable PIM execution unit at the I/O boundary of a bank based on HBM2
  - Exploit bank-level parallelism: access multi banks/FPUs in a lockstep manner
  - Support both standard HBM and Aquabolt-XL modes for versatility
  - Minimize engineering cost of re-designing DRAM core to support PIM
- Drop-in Replacement: the same form-factor, timing parameters, and commands as Aquabolt product
  - Facilitate drop-in replacement of JEDEC-specification compliant Aquabolt HBM2 with Aquabolt-XL HBM-PIM for any system
- Passive device: DRAM RD/WR command triggers execution of a PIM instruction in PIM mode
  - Preserving determinism – Host knows the status of DRAM



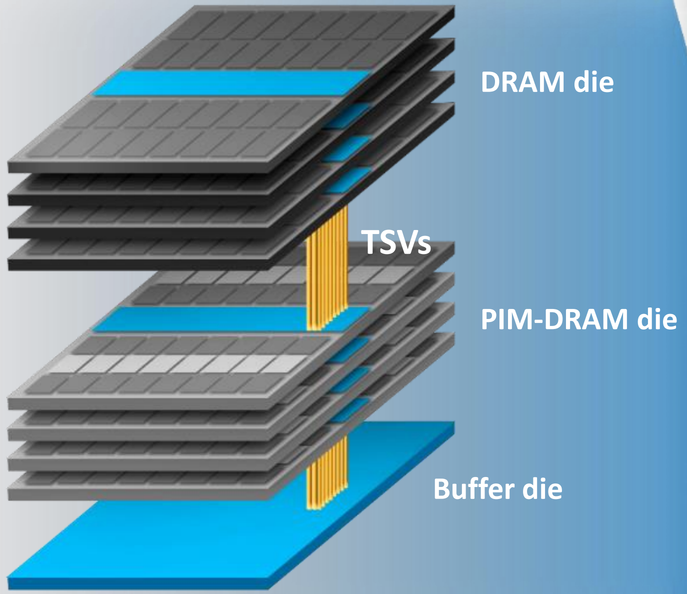


Chip Implementation and Integration with systems

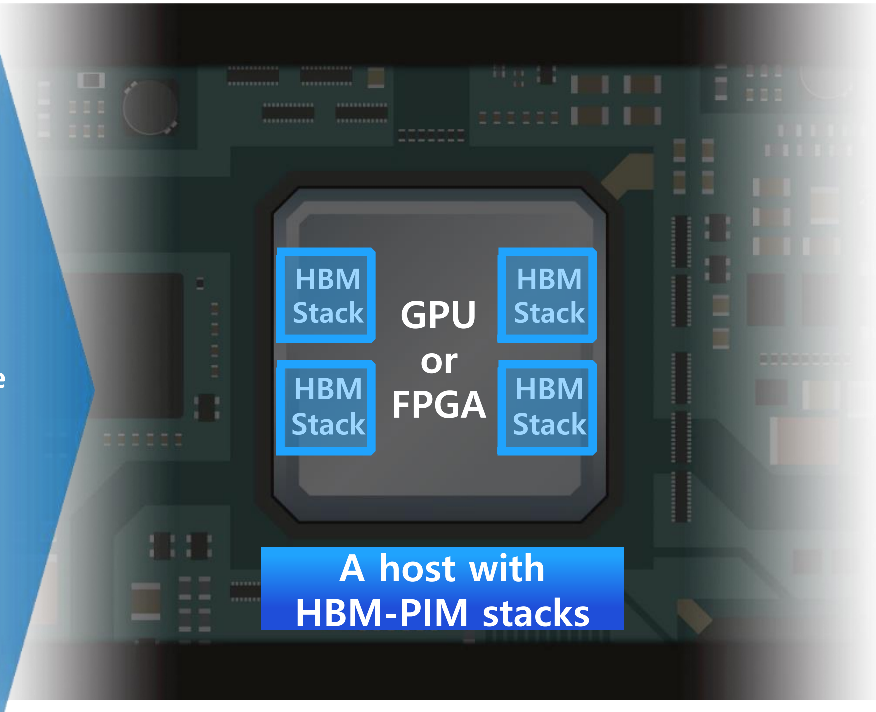
- Implemented PIM by modifying a commercial HBM2 design (Aquabolt). Resulting HBM-PIM device codenamed Aquabolt-XL
- Integrated the fabricated Aquabolt-XL with an unmodified GPU and Xilinx FPGA
  - Validated fabricated HBM-PIM in system with unmodified HBM controller
  - Off-chip and on-chip PIM compute bandwidth is 1.23 TB/s and 4.92 TB/s, respectively.



Aquabolt-XL silicon die photo

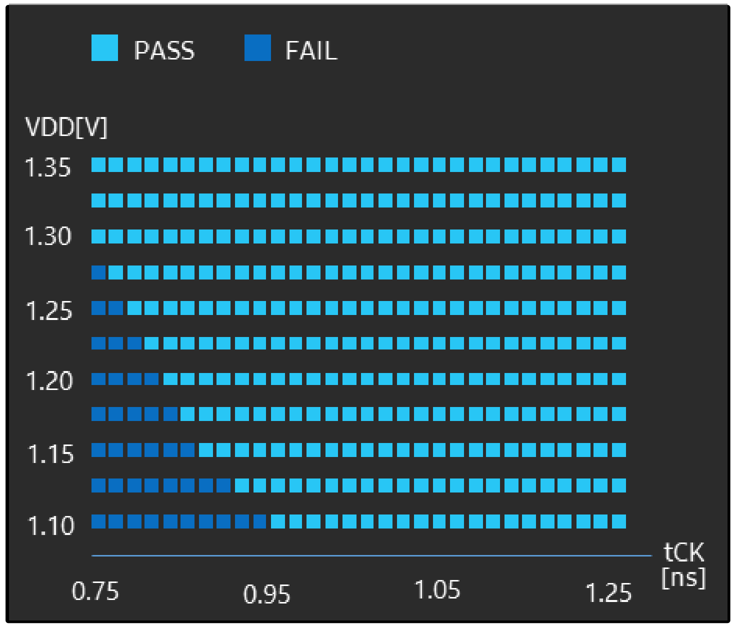


3D-stacked PIM-HBM

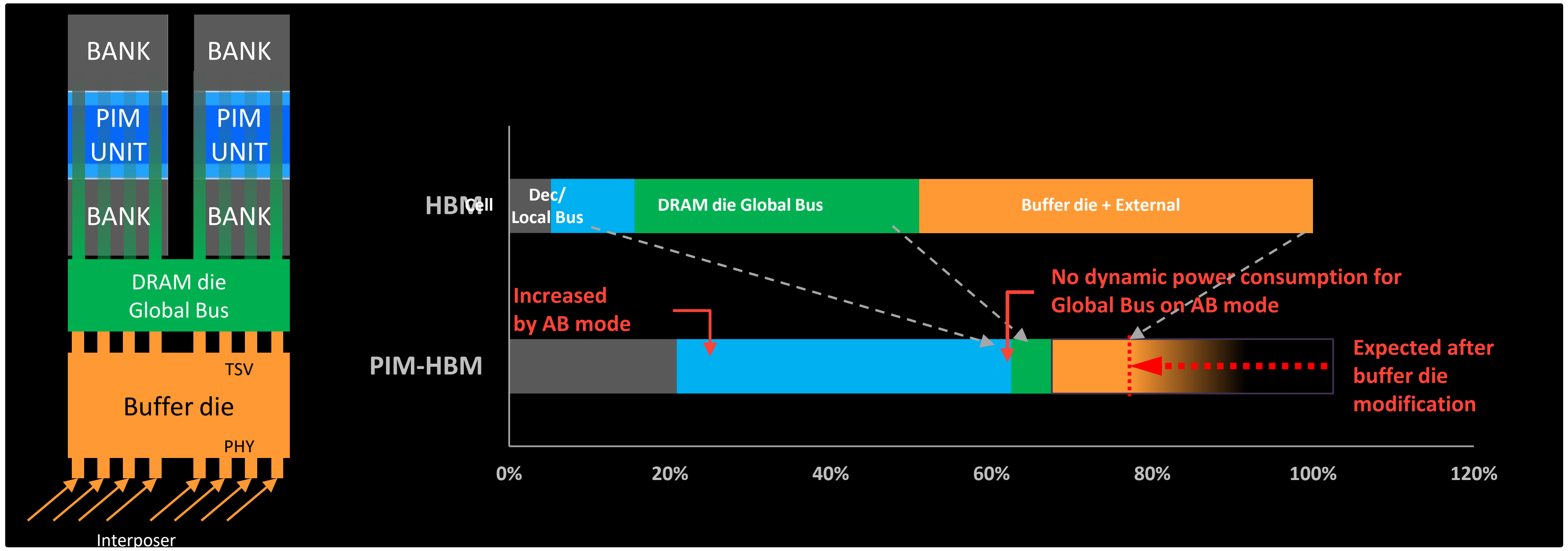


Chip Evaluation

- **Aquabolt-XL passes commercial product engineering steps**
  - Operate at 2.4Ghz, without any shmoo hole
- **Power analysis**
  - On the heavy all-channel/bank PCU operation workload, it consumes 5.4% higher current than Aquabolt



Shmoo plot of Aquabolt-XL



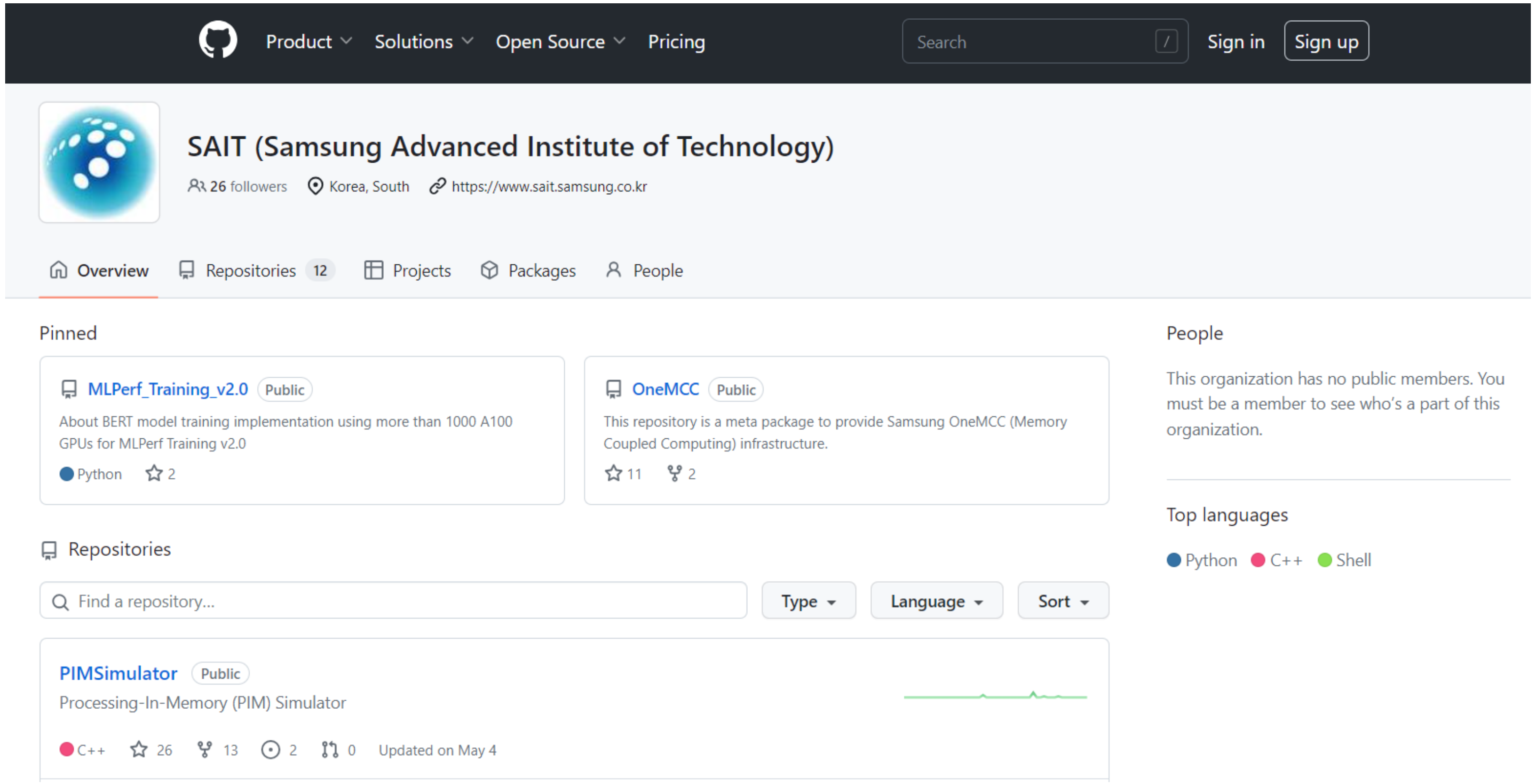
I/O Power breakdown comparison between HBM and PIM-HBM

Reference: Aquabolt-XL HBM2-PIM, LPDDR5-PIM with in-memory processing, and AXDIMM with acceleration buffer in *IEEE Micro*

HBM-PIM Simulator

HBM-PIM Simulator

- SEC opened HBM-PIM simulator to the public
- DRAMsim2-based "cycle-accurate" model for Aquabolt-XL
- It contains internal control register sets and procedures for PIM with HBM JEDEC specification
- GitHub URL: https://github.com/SAITPublic/PIMSimulator → Let me explain Readme.md



SAIT (Samsung Advanced Institute of Technology)

26 followers · Korea, South · https://www.sait.samsung.co.kr

Overview · Repositories 12 · Projects · Packages · People

Pinned

MLPerf_Training_v2.0 Public
About BERT model training implementation using more than 1000 A100 GPUs for MLPerf Training v2.0
● Python ☆ 2

OneMCC Public
This repository is a meta package to provide Samsung OneMCC (Memory Coupled Computing) infrastructure.
☆ 11 2

Repositories

Find a repository...
Type · Language · Sort

PIMSimulator Public
Processing-In-Memory (PIM) Simulator
● C++ ☆ 26 13 2 0 Updated on May 4

People

This organization has no public members. You must be a member to see who's a part of this organization.

Top languages

● Python ● C++ ● Shell

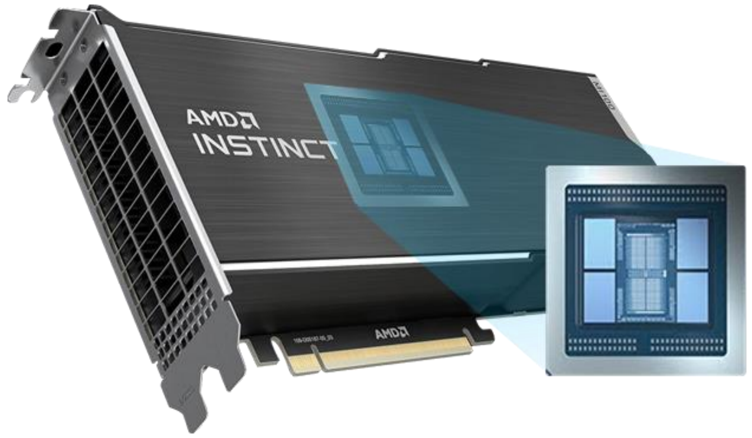
HBM-Powered Systems

Introducing HBM-PIM Powered Board Systems

SAMSUNG

- Need a host: HBM-PIM cannot perform any operation itself without a host
- Thus, SEC collaborated with two system-board companies

AMD MI50/100-PIM GPU



Xilinx Alveo U280 FPGA

HBM-PIM Powered Systems

SAMSUNG

- AMD MI50
  - Based on Normal MI50/100 GPU system
  - Consists of four HBM2 stacks
  - GPU FP16 performance: 26.5 TFLOPS
  - HBM-PIM bandwidth: 1024GB/s
  - HBM-PIM internal bandwidth/performance: 4096GB/s – (FP16) 4 TFLOPS
  - Capacity: 24GB
- AMD MI100
  - Consists of four HBM2 stacks
  - GPU FP16 performance: 184.6 TFLOPS
  - HBM-PIM bandwidth: 1228GB/s
  - HBM-PIM internal bandwidth/performance: 4912GB/s – (FP16) 4.9 TFLOPS
  - Capacity: 24GB

Pros: Use only software techniques
Cons: GPU architecture limits the gain

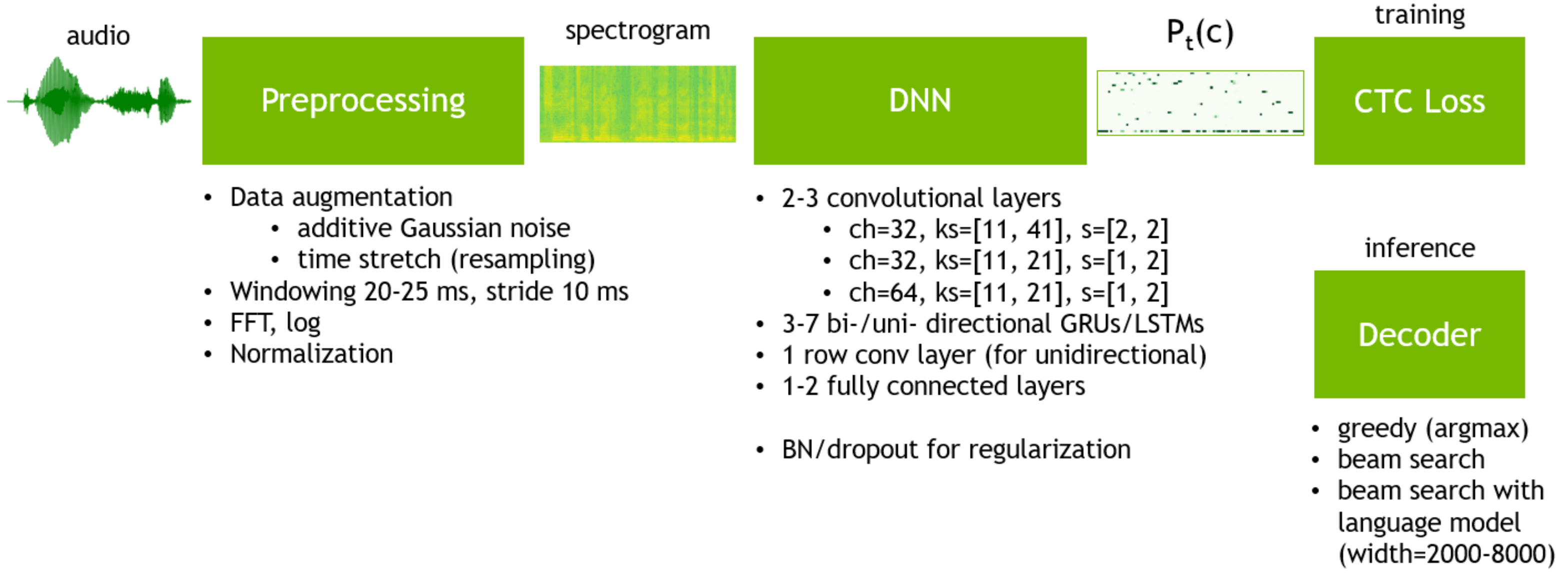
- Xilinx U280
  - Consists of two HBM2 stacks
  - HBM-PIM bandwidth: 460GB/s
  - HBM-PIM internal bandwidth/performance: 1840GB/s – (FP16) 1.8 TFLOPS
  - Capacity: 12GB

Pros: Can show maximum performance gain
Cons: Need to build (RTL) everything including the baseline

Application 1: Deepspeech on MI50-PIM [ISCA2021, Hotchips 2021]

DeepSpeech2

- Sequence to sequence speech recognition DNN algorithm from BAIDU
- From our observation, DS2 inference has matrix-vector multiplications for RNNs (GRU/LSTMs) on the single batch case and elementwise (BN: batch normalization) operations
- More than 80% of the layer consists of GEMV operation
- Because of these characteristics, DS2 is selected for the first accelerator target algorithm.

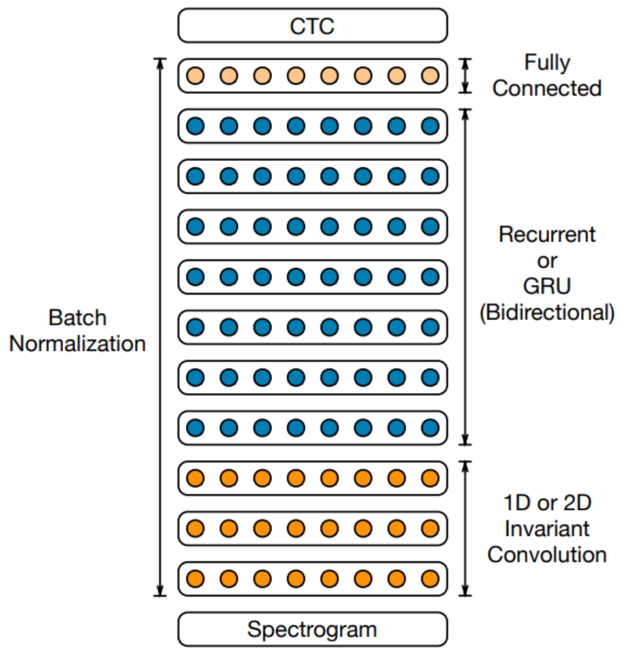


- Data augmentation
    - additive Gaussian noise
    - time stretch (resampling)
- Windowing 20-25 ms, stride 10 ms
- FFT, log
- Normalization

- 2-3 convolutional layers
    - ch=32, ks=[11, 41], s=[2, 2]
    - ch=32, ks=[11, 21], s=[1, 2]
    - ch=64, ks=[11, 21], s=[1, 2]
- 3-7 bi-/uni- directional GRUs/LSTMs
- 1 row conv layer (for unidirectional)
- 1-2 fully connected layers

- BN/dropout for regularization

- greedy (argmax)
- beam search
- beam search with language model (width=2000-8000)

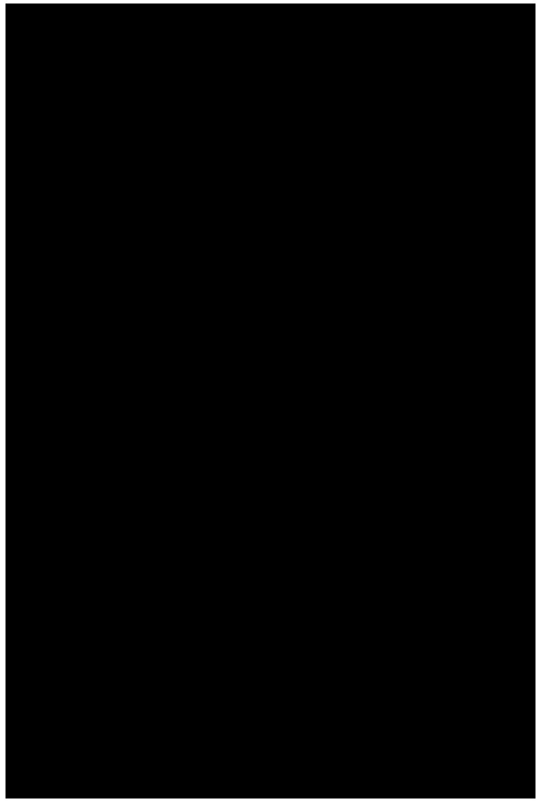


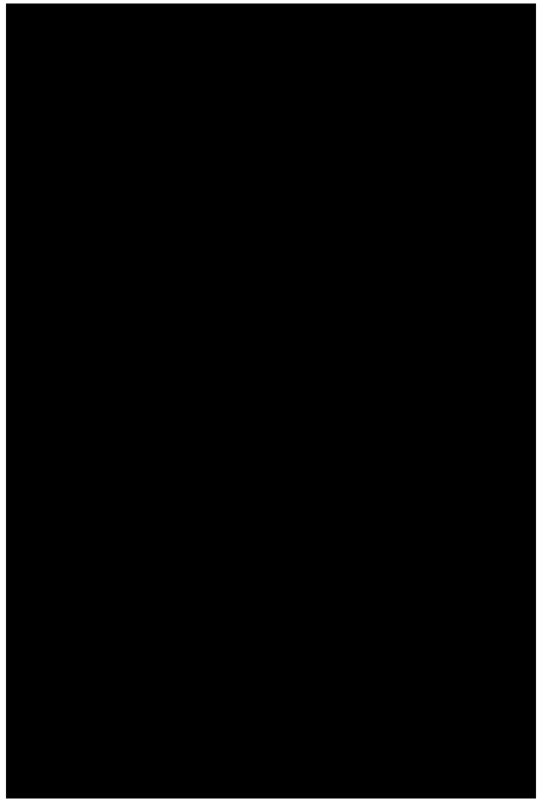
CTC
Fully Connected
Recurrent or GRU (Bidirectional)
Batch Normalization
1D or 2D Invariant Convolution
Spectrogram

Demo and System Evaluation

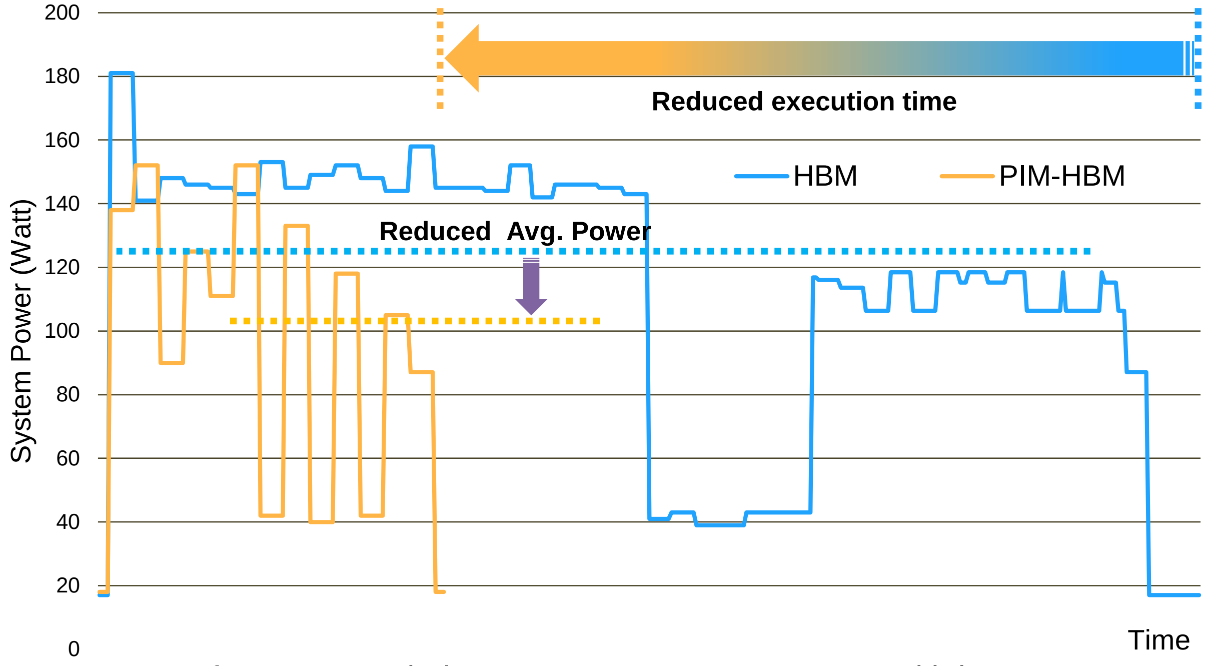
- MI50-PIM shows 3.2x faster and 3.5x higher energy efficiency than the baseline (normal HBM)
- Working demo and real-time avg. system power analysis

DeepSpeech2 DEMO


Aquabolt


Aquabolt-XL

Average system power of DeepSpeech2 over time

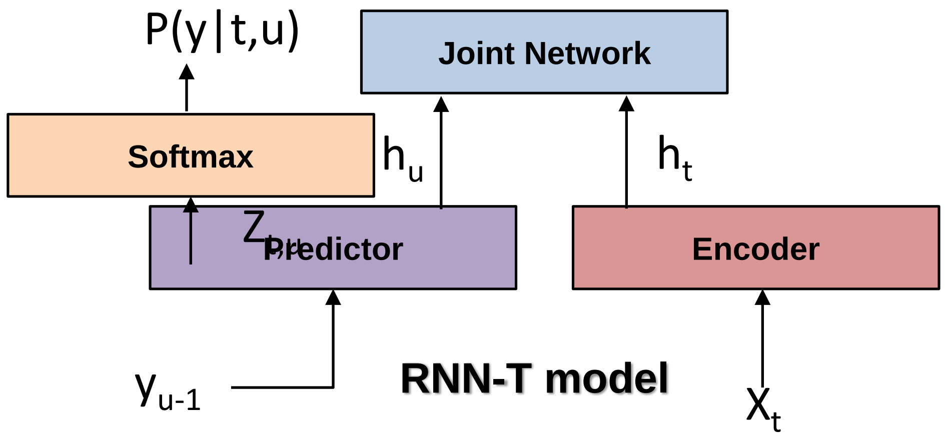


Reference: Aquabolt-XL HBM2-PIM, LPDDR5-PIM with in-memory processing, and AXDIMM with acceleration buffer in *IEEE Micro*

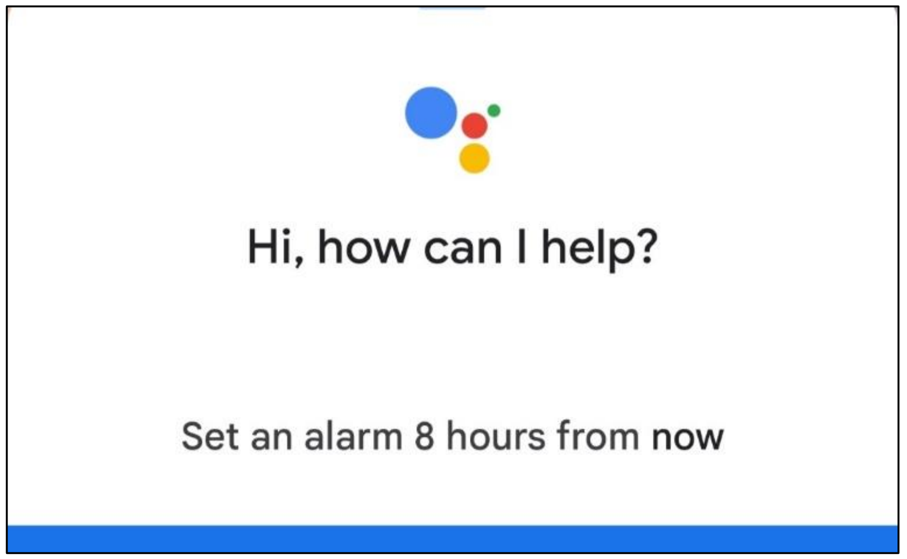
Application 2: RNN-T on U280-PIM [FPGA2022, CES2022, SC2022]

Demo Target Application: RNN-T

- Encoder : Five LSTM layers (hidden dimension: 1024) and one time stacking layer.
  - It receives speech audio data divided into ‘n’ time steps and calculates ‘n/2’ of encoder outputs.
- Predictor : Embedding and two LSTM layers (hidden dimension: 320).
  - It uses previous token (argmax character) as an input and generates the outputs.
- Joint network : Two FC and one ReLU layers.
  - It concatenates the encoder and predictor output and calculates the probability distribution of result characters.
- Main key layer : LSTM layer consumes 90.7% of the running time, and vector-matrix multiplication kernel uses 78.8%.



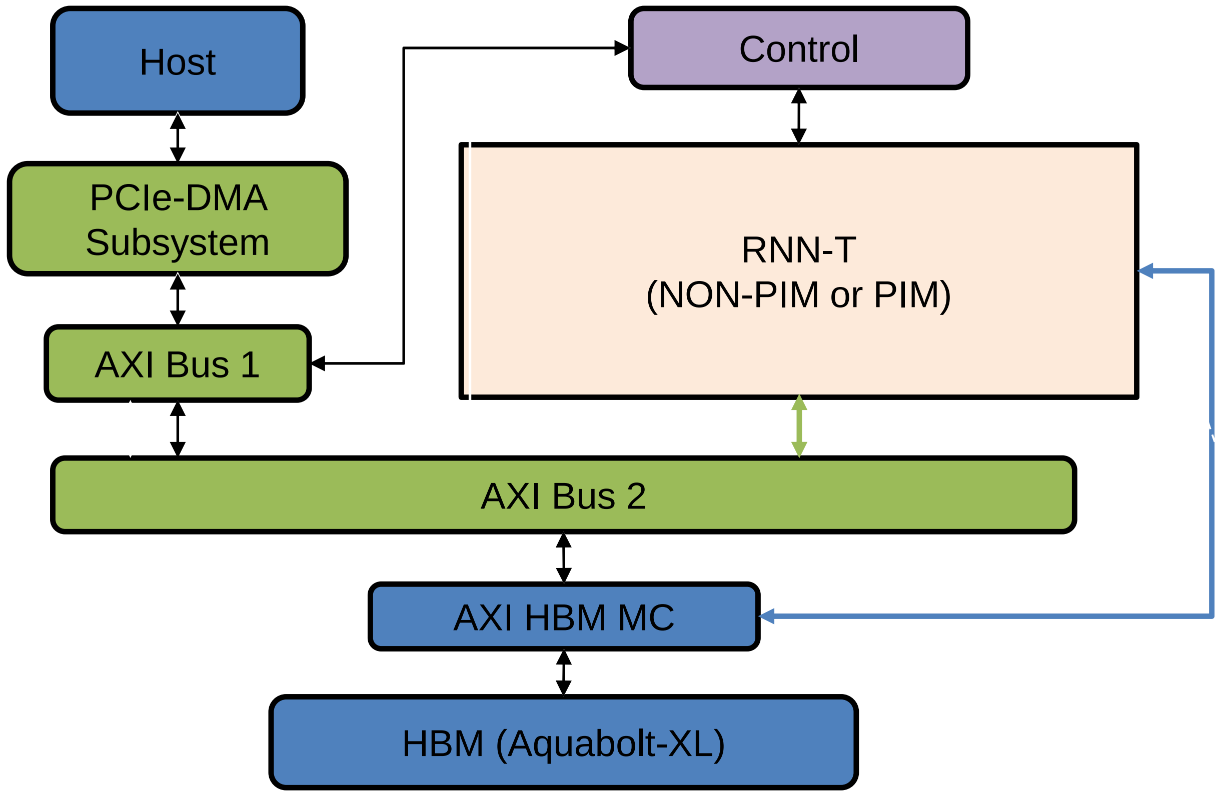
RNN-T model

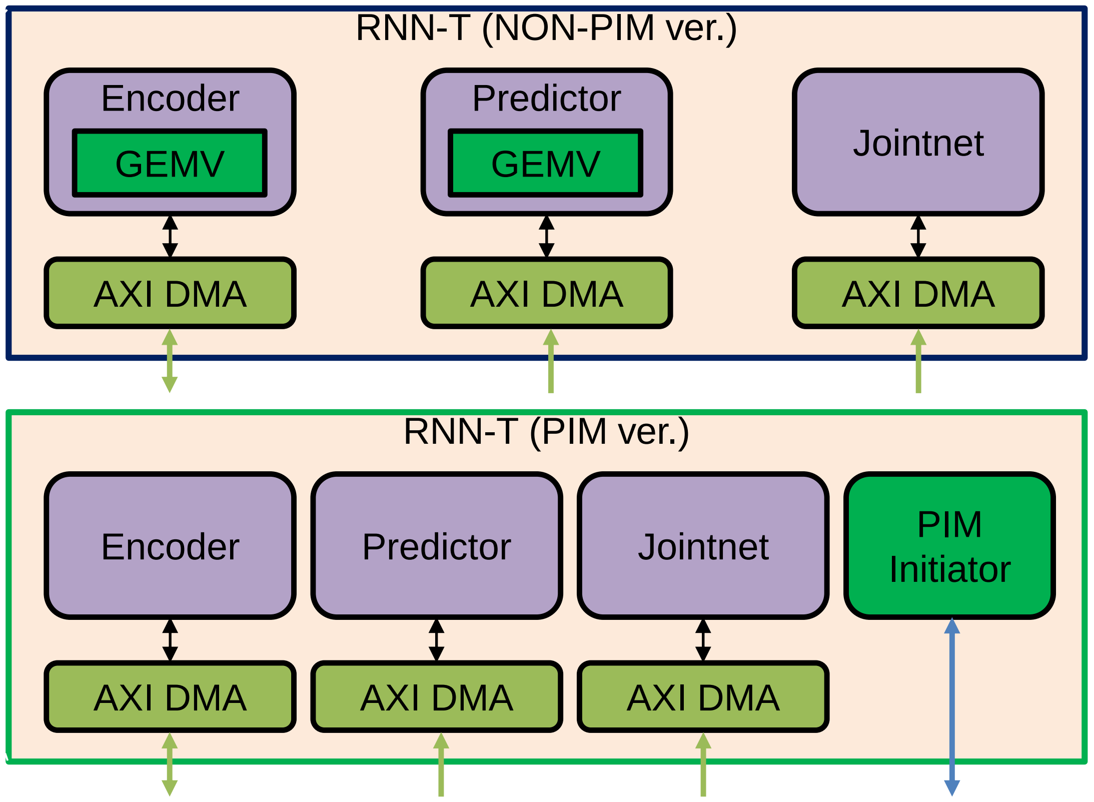


‘Hey Google’ Voice search

RNN-T Accelerator (FPGA) Top-level Design

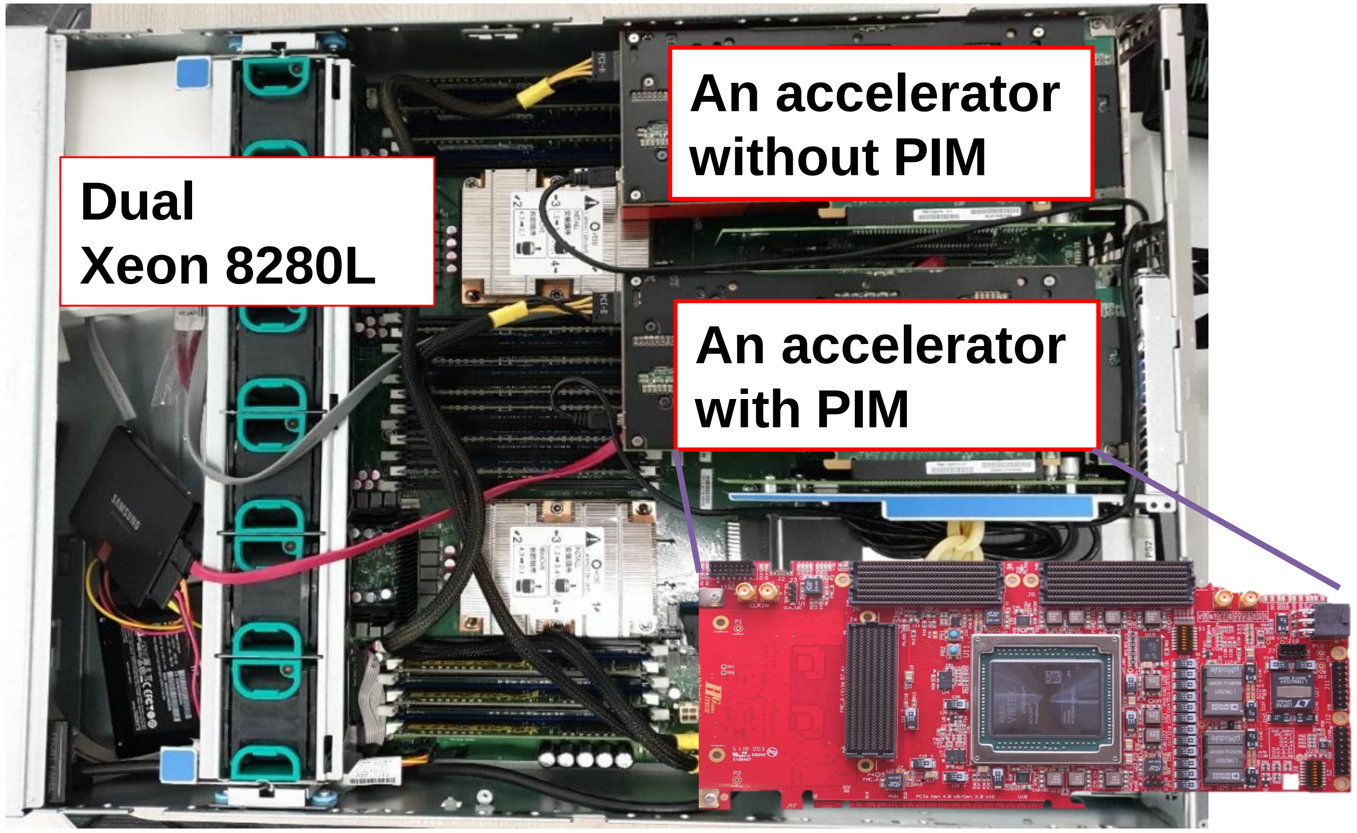
- We have designed two versions of RNN-T accelerator: PIM vs NONPIM
  - NONPIM: perform vector-matrix multiplication in LSTM in FPGA
  - PIM: perform vector-matrix multiplication in HBM-PIM
- PIM initiator (controller): DMA supporting in-order memory access + PIM instruction generator
- Activation function (Sigmoid, Tanh): LUT-based approximation





Evaluation Setup

- Host CPU transfers data and control signals to the FPGA through PCIe 3.0 interface, and FPGA runs the entire network and notifies the host of the end of the execution.
- PIM FPGA uses 27.4% less power from its smaller logic and lower peak performance (FLOPS) as much as 4.9 W, but it consumes only 0.6 W more power on HBM2 due to PIM accelerations.



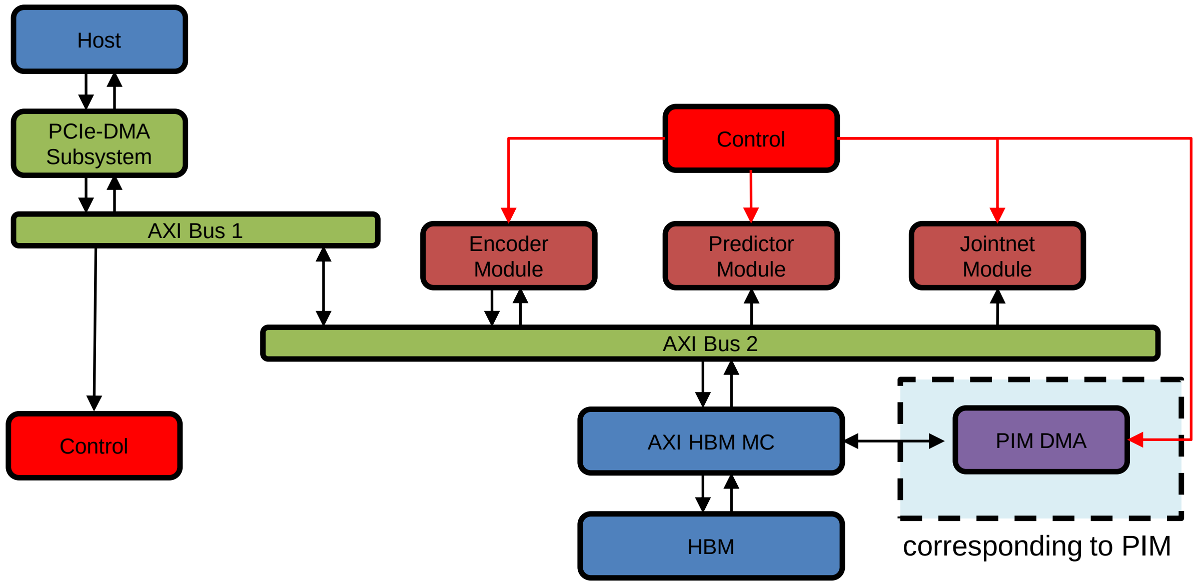
Xilinx Alveo U280 (SiP: VU47P + Two HBM-PIM)

Description	NON-PIM	PIM
Core Frequency	**300 MHz**	**300 MHz**
HBM Datarate	**1.8 Gbps**	**1.8 Gbps**
Memory	two HBM2s	two PIM-HBMs
CLB LUTs	32,511 (12.20%)	29,266 (10.88%)
CLB Registers	162,586 (6.24%)	157,772 (6.05%)
DSPs	221 (2.45%)	188 (2.08%)
Avg. FPGA power	23.223 W	18.224 W
Avg. HBM power	4.9 W	5.5 W
Peak DRAM BW	9.6 Gbps	4.8 Gbps
Peak Perf.(FPGA)	256.5 GFLOPS	47.4 GFLOPS

Demo Environment & Target application

SAMSUNG

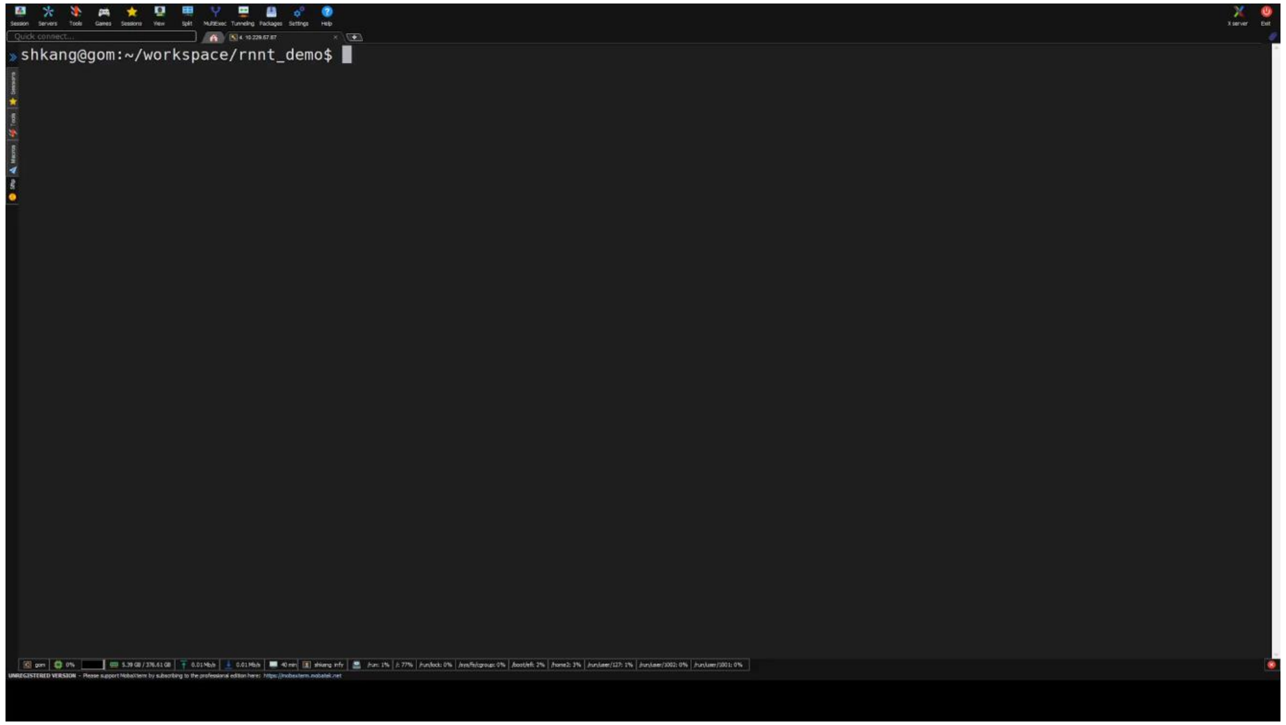
- Target Application : RNN-Transducer (RNN-T)
  - Main Module : Encoder, Predictor, Jointnet
    - Key layer : LSTM layer consumes 90.7% of the running time, and vector-matrix multiplication kernel uses 78.8%
- RNN-T PIM on U280
  - Demo Environment : Intel Xeon, Ubuntu 18.04, 2 * Xilinx Alveo U280 FPGA board (VU47, 2 HBM-PIM stacks)
  - Baseline system: RNN-T module @ 300Mhz without HBM-PIM
  - Target system : RNN-T module @ 300Mhz with HBM-PIM



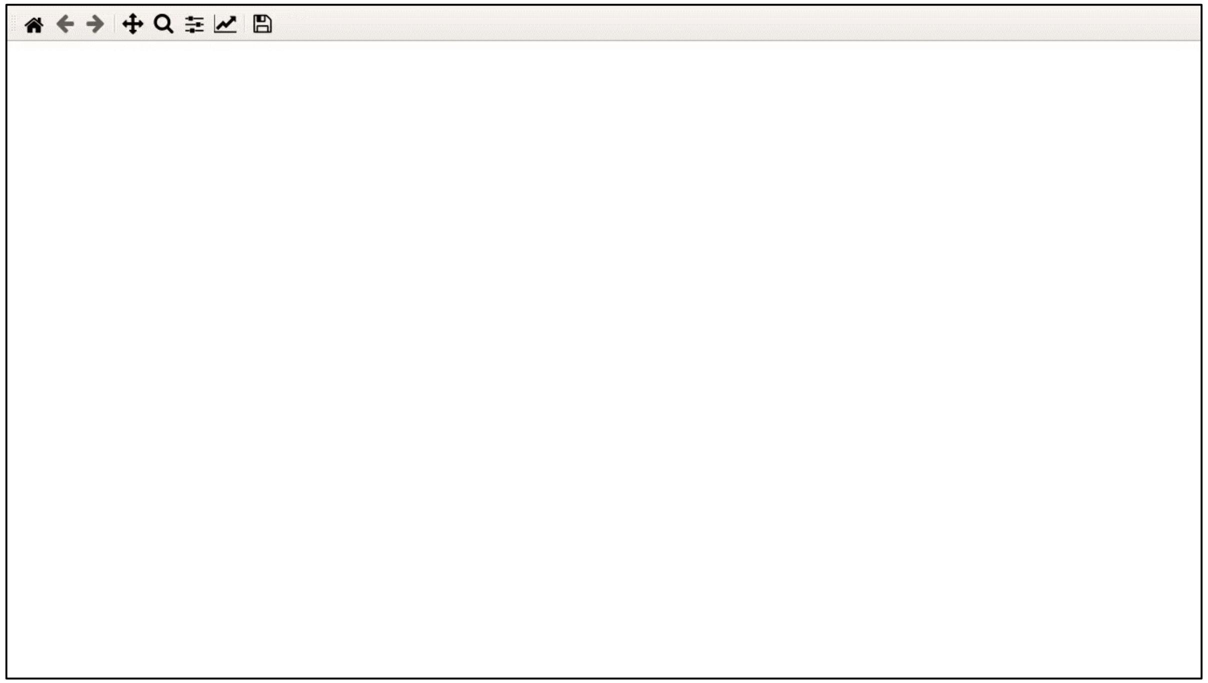
RNN-T FPGA System (rnnt_tiger) Structure

PIM Live Demo Demonstration

- RNN-T PIM paper is accepted to FPGA 22
  - "An FPGA-based RNN-T Inference Accelerator with PIM-HBM"



```
shkang@gom:~/workspace/rnnt_demo$ 
```

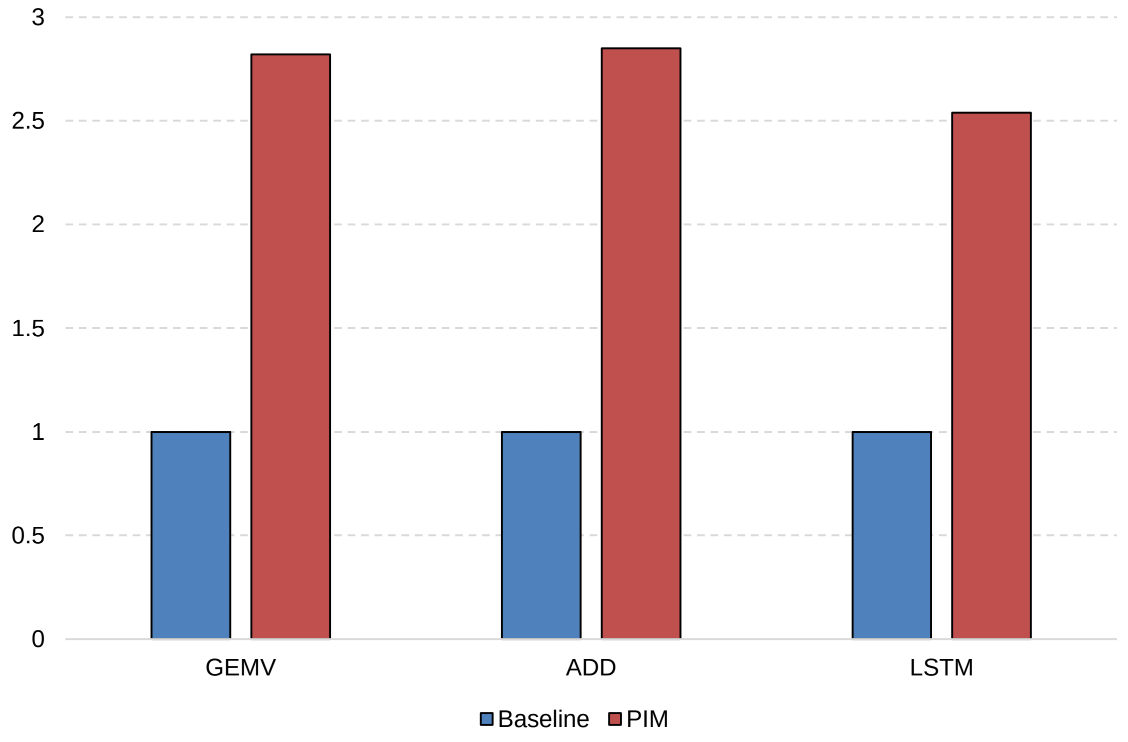


Performance Comparison

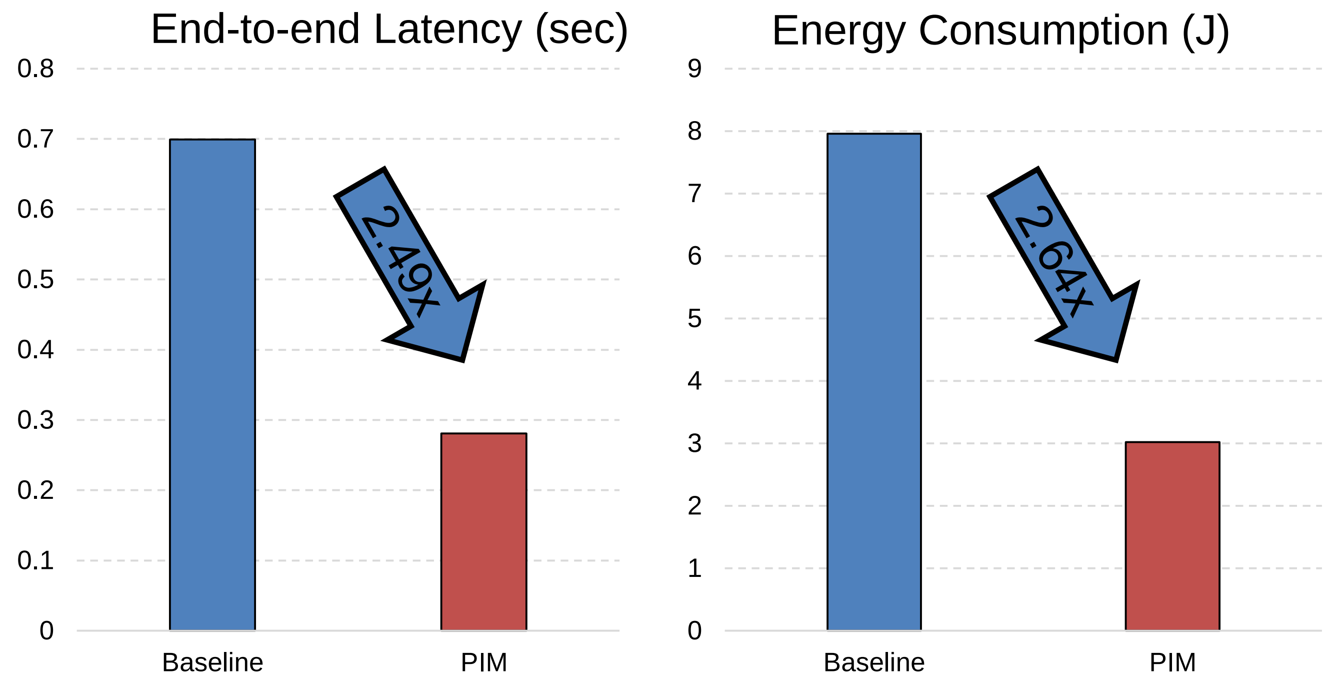
SAMSUNG

- PIM for Micro-benchmarks have performance gain up to 2.5 ~ 2.8x.
- PIM reduce the End-to-end RNN-T Execution Latency by up to 2.49x and the Energy Consumption by up to x2.64.

Micro-benchmark Speed-up

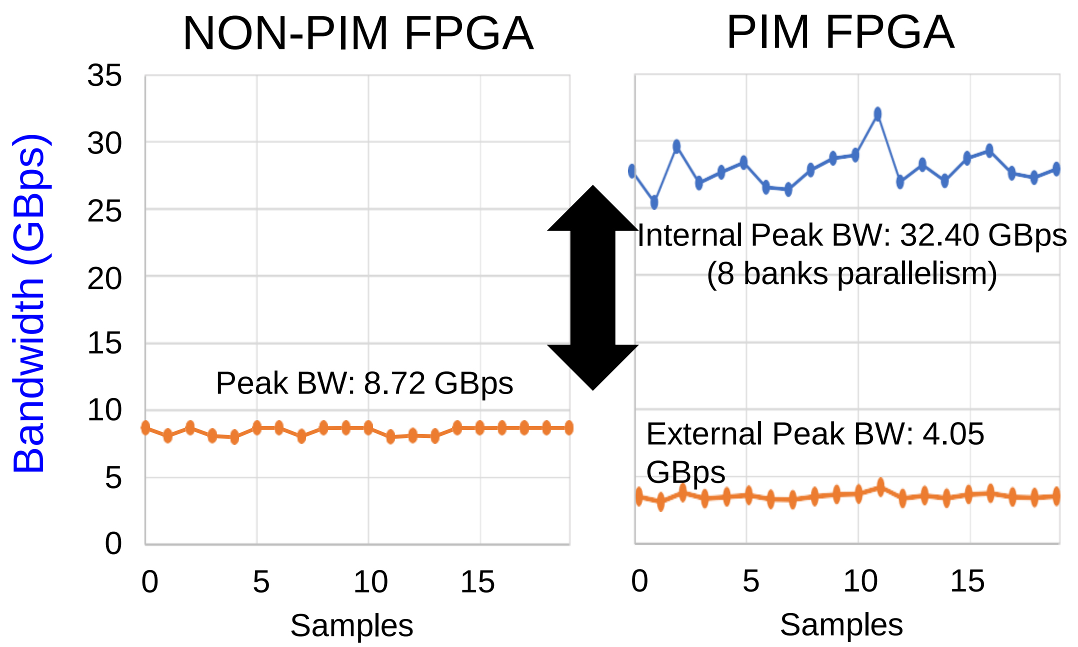


End-to-end RNN-T Execution



Performance Analysis & Comparison

- Performance
  - PIM FPGA runs 2.5 x faster than NONPIM FPGA
  - PIM FPGA (projected to A100's BW) runs 1.98x faster than the world record (A100 with tensor-RT)
- Accuracy: MLPerf-qualified
  - PIM accelerator presents an accuracy of 99.5% (8.09 WER) to the baseline (7.66 WER)



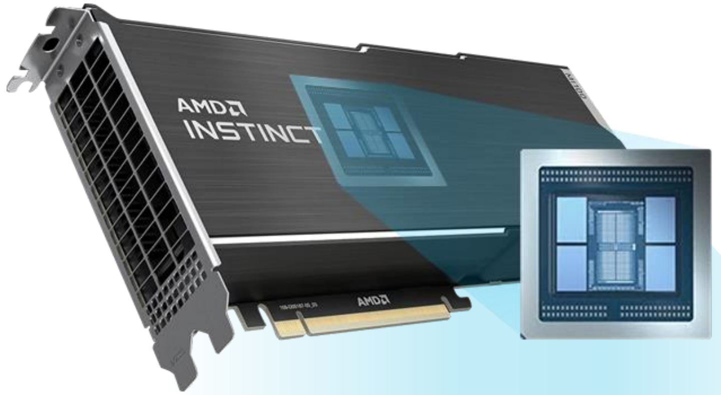
	Latency(ms)
PIM FPGA (1 pCH)	574.13
NON-PIM FPGA (1 pCH)	1422.38
PIM FPGA (32 pCHs, projected)	24.81
PIM FPGA (A100 BW, projected)	9.76
Intel Xeon Platinum 8280L	173.02
Nvidia A100 80GB w/o tensor-RT	82.63
Nvidia A100 80GB w/ tensor-RT	19.3

Application 3: T5-MoE on MI100-PIM Cluster [CES2022, SC2022, Memcon2023]

Architecture of HBM-PIM Cluster

- Installed 96 AMD MI100 GPUs fabricated with HBM-PIM
- Accelerate large-scale workloads with low latency and high energy efficiency

AMD MI100-PIM GPU



· Capacity : 24GB (4 cubes)
· PIM performance : 4.9 TFLOPS

Server node

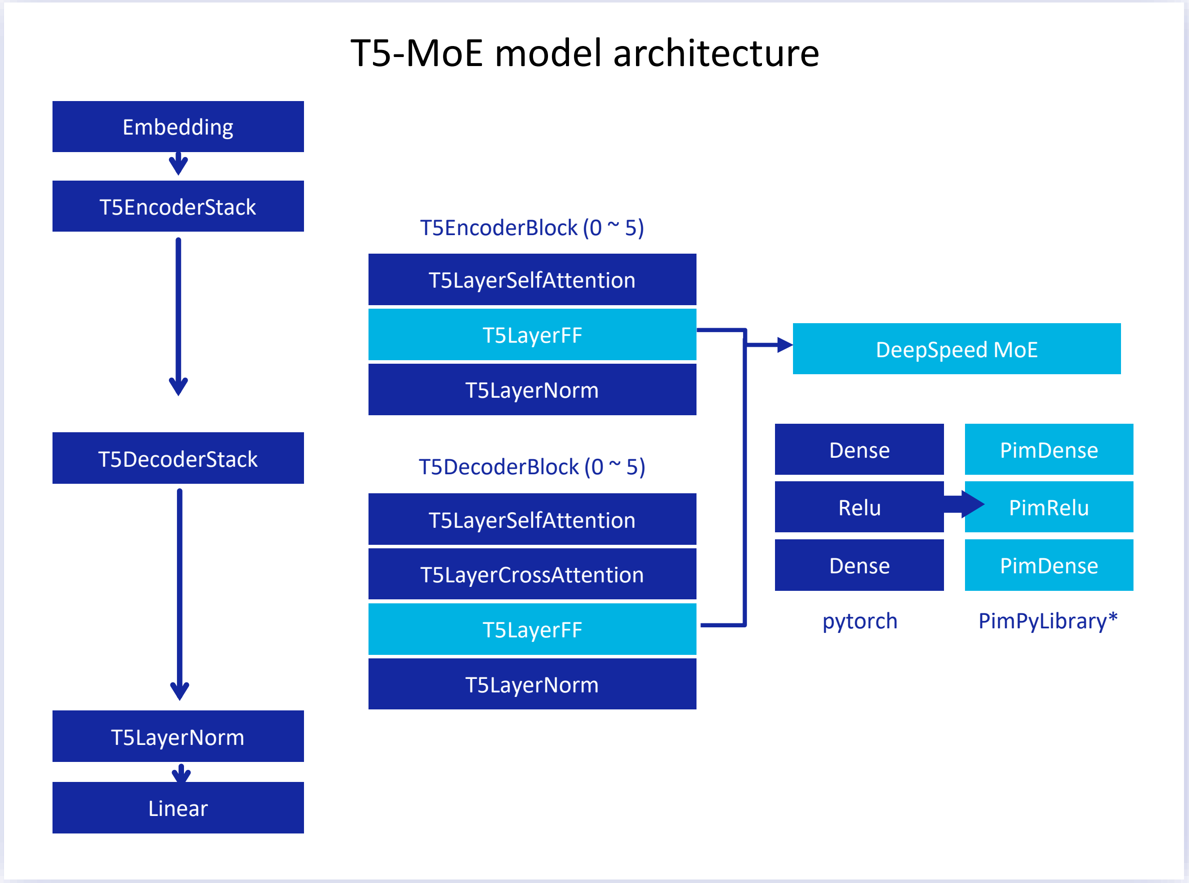
· 8 MI100-PIM GPUs per node

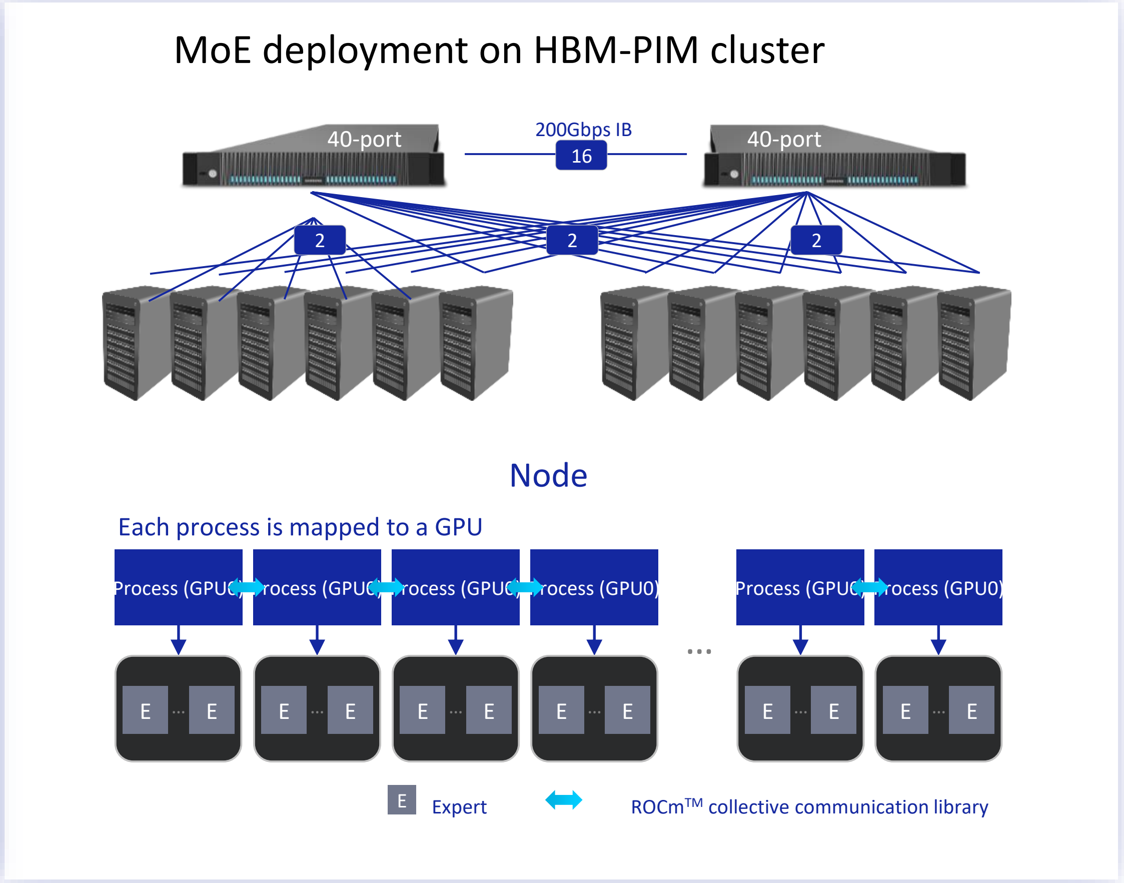
HBM-PIM cluster

· 12 nodes interconnected through 200G InfiniBand network
· Total 96 MI100-PIM GPUs in a cluster

Workload : T5-Based MoE (Mixture of Expert) Model

- MoE model mostly uses GEMV functions
- DeepSpeed MoE replaces T5LayerFF layer to accelerate T5-large model with PIM
- The MoE layers are updated to use our PimPyLibrary* APIs

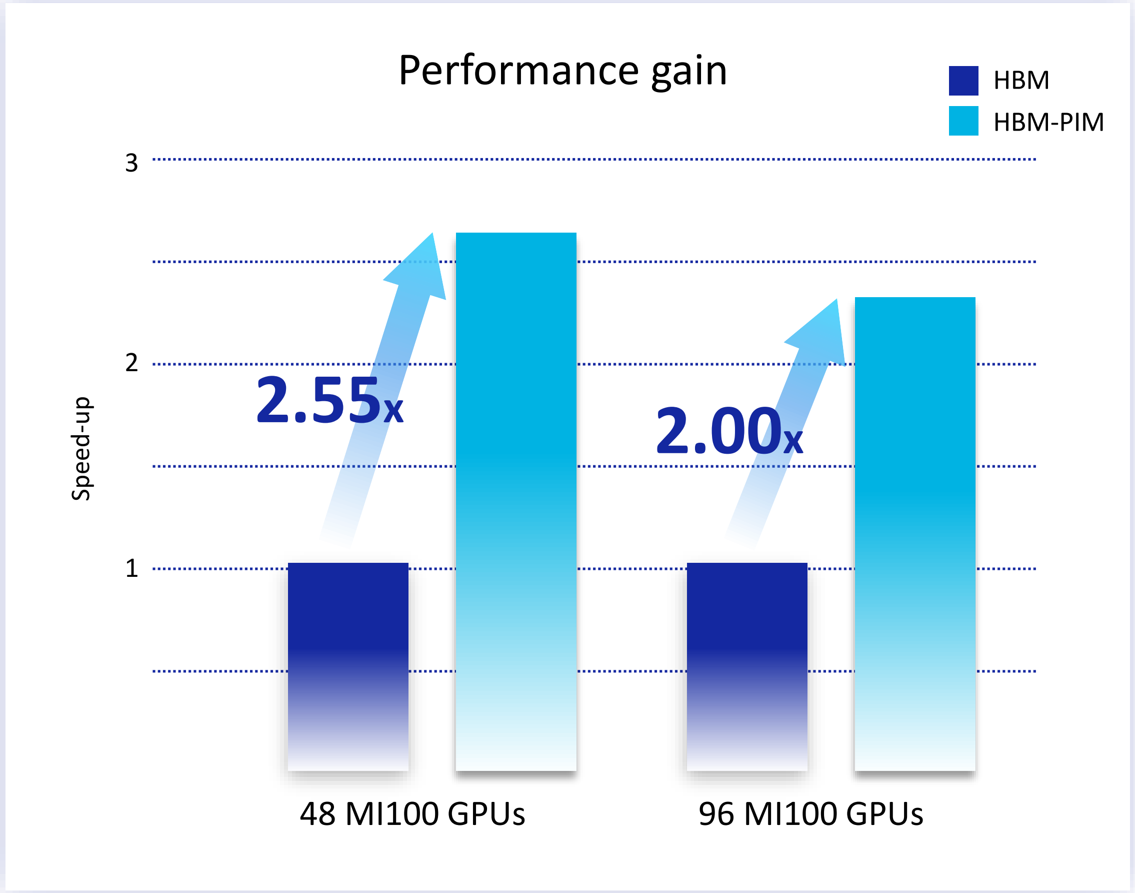


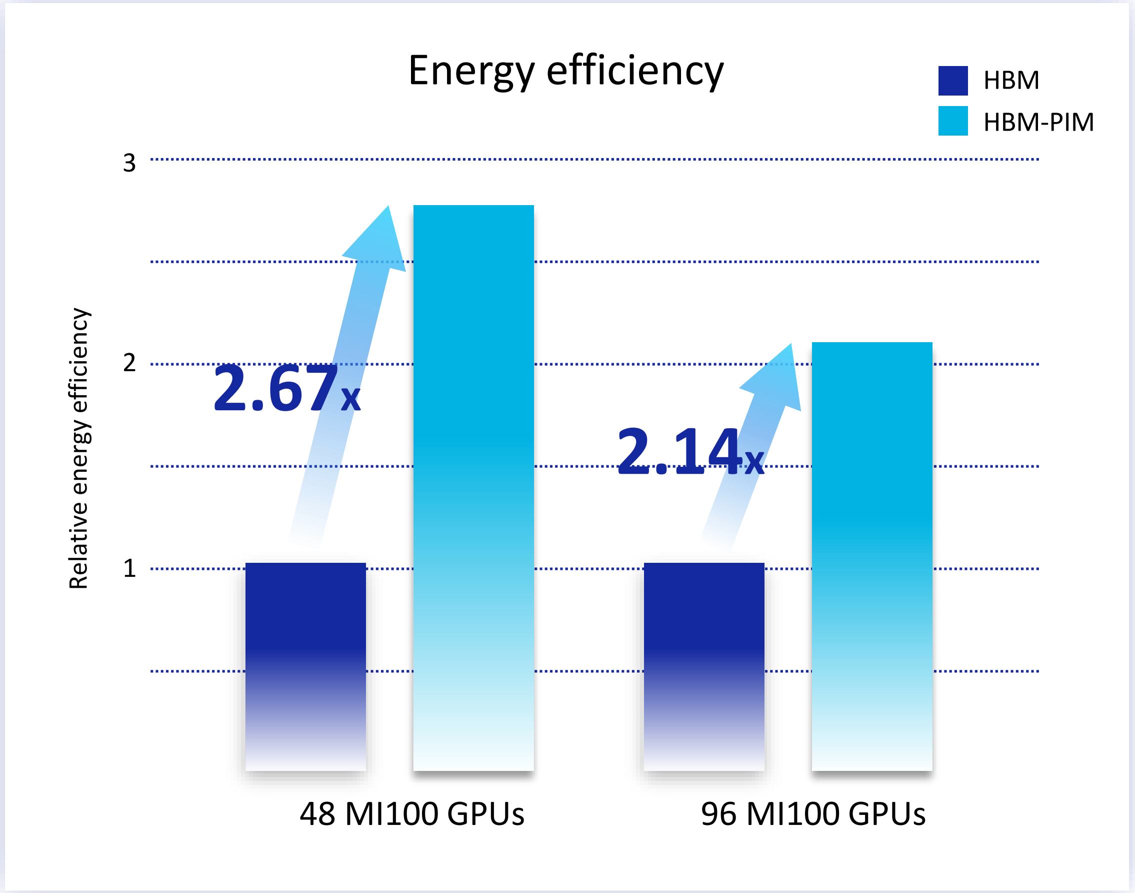

MoE deployment on HBM-PIM cluster
40-port
200Gbps IB
16
40-port
2
2
2
Node
Each process is mapped to a GPU
Process (GPU0)
E ... E
...
E Expert
ROCm™ collective communication library

* PimPyLibrary is a python library for providing PIM-enabled AI operators. PIM SDK provides not only PimPyLibrary but also full SW stack for utilizing PIM

Performance and Energy Efficiency of T5-MoE Model

- More than 2x greater performance compared to normal GPU clusters
- Increases system energy efficiency by more than 2x over baseline





* Acknowledgement: Jaeyoung Heo and professor Sungjoo Yoo (Seoul National University) provided the idea of PIM acceleration for this workload

Application 4: GPT on U280-PIM

GPT-3 Generative Model Workload

- GPT model consists summarization and generation stage
- By increasing the size of the output sequences, the total portion of the generation gets higher

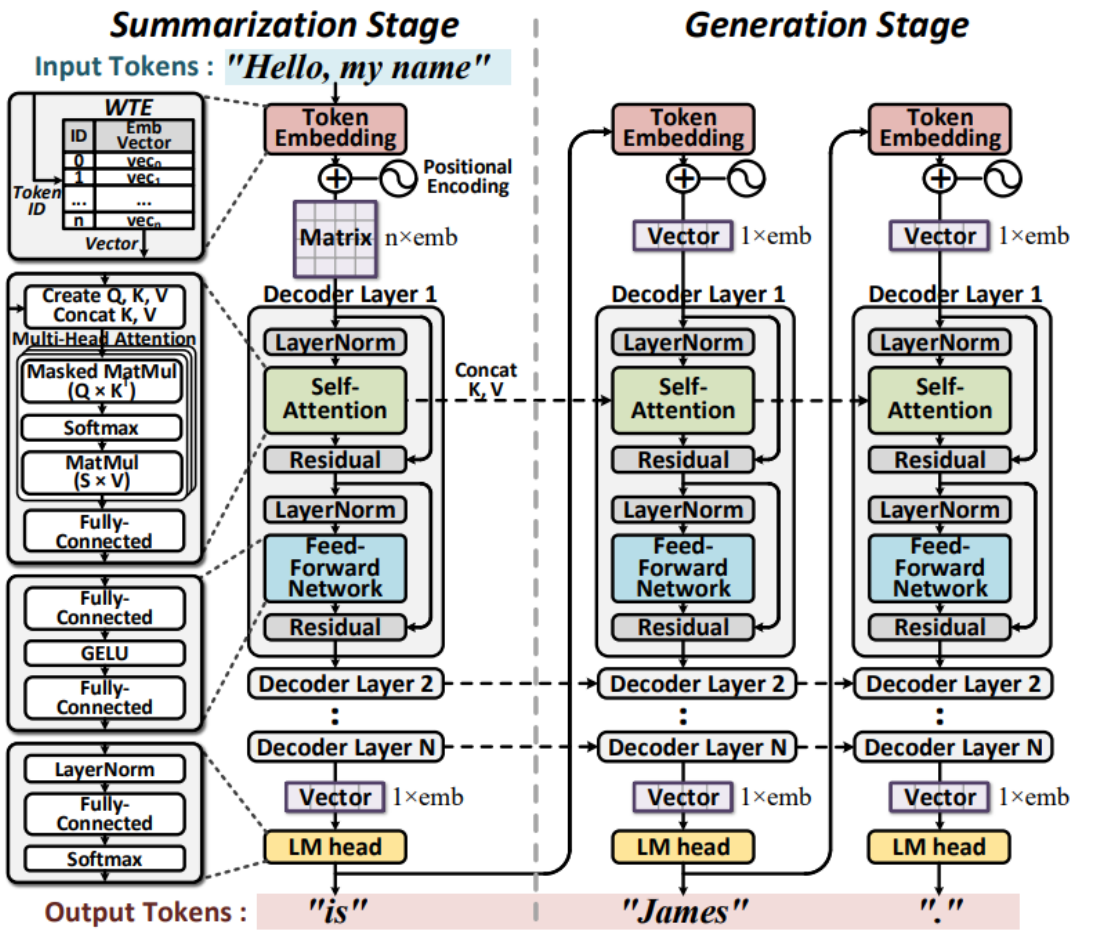
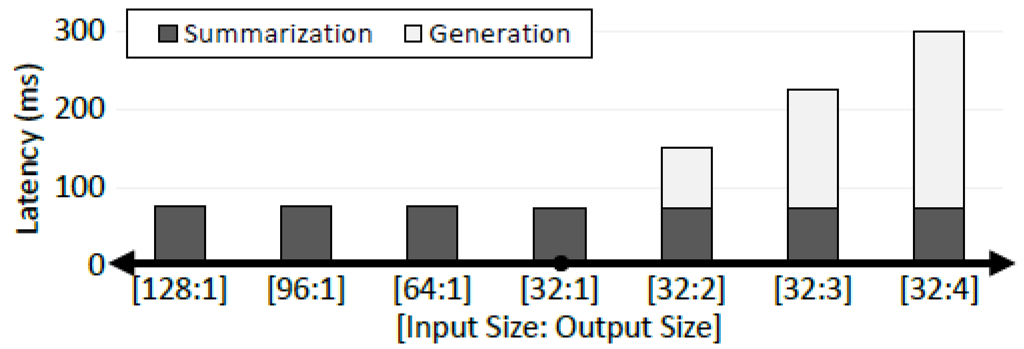


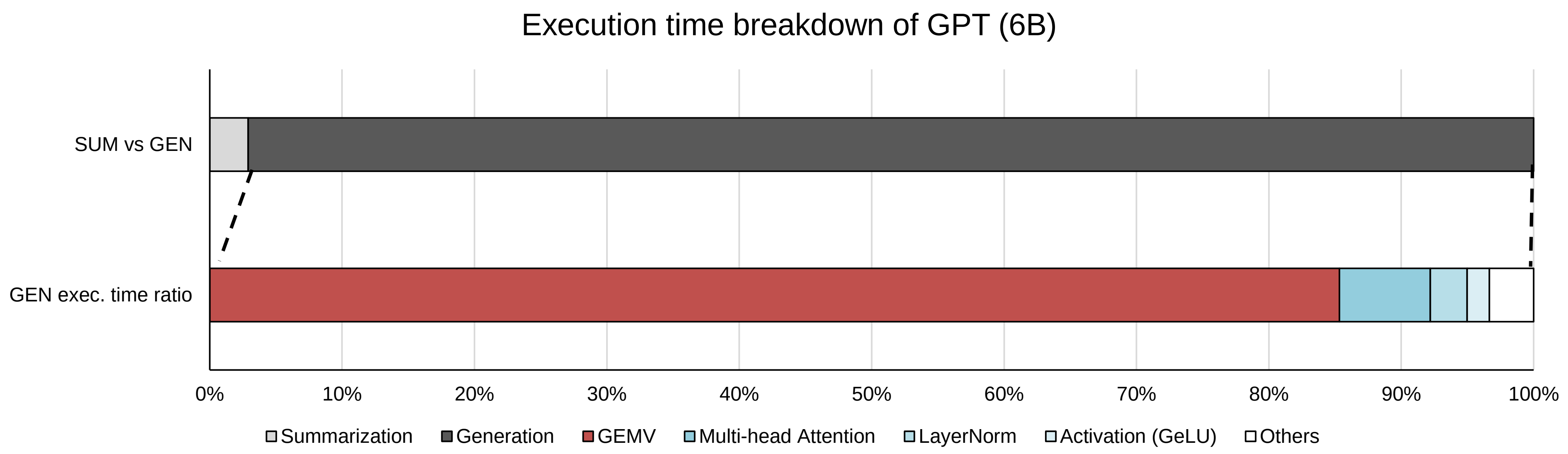
Figure 2. GPT-2 structure and illustration of summarization and generation stages in text generation.



Source : DFX: A Low-latency Multi-FPGA Appliance for Accelerating Transformer-based Text Generation, MICRO 2022

GPT-3 Profiling Result on GPU

- We found two key observations from GPT-3 workload analysis with profiling tool
  - Unlike BERT, GPT-3's summarization portion is smaller than generation phase
  - As the weight size increases, the portion of GEMV in the total execution time also increases.
- Currently, multi-batch inference is difficult to use in GPT-3 due to size and service level agreement (SLA) latency requirements.
- From this, it seems that GPT-3 is a memory bound application (capacity and bandwidth).



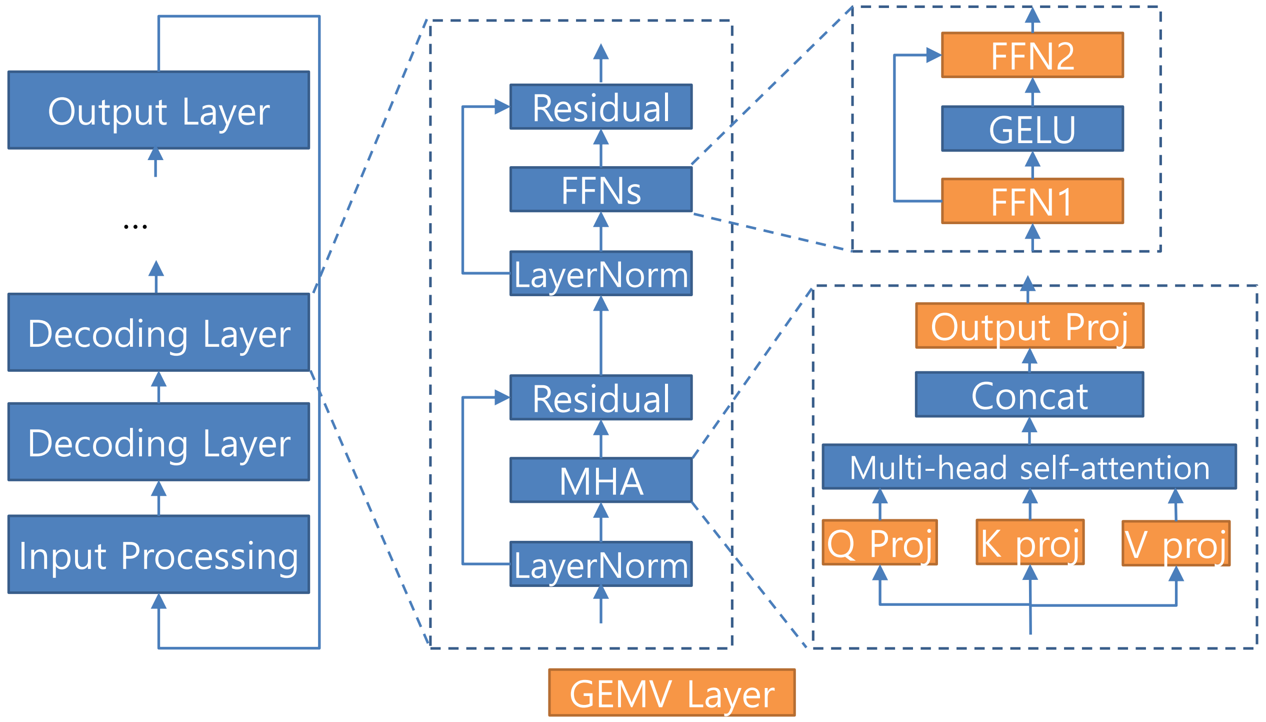
Nvidia A100, Fastertransformer, Input:Output = (64,64), Single batch

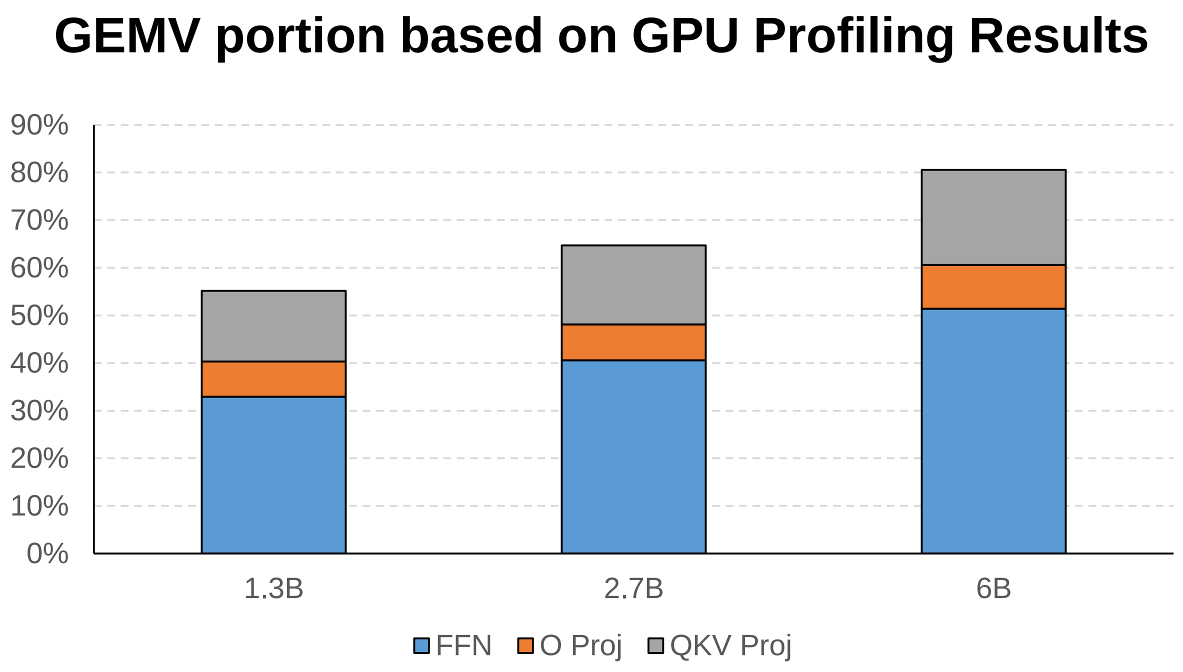
GPT-3 Latency for GEMV

- Memory-bound: vector-matrix multiplication takes most of execution times.
  - As the model size increases, the linear layer O(H^2) overwhelms attention layer O(HL).

  *H is hidden dimension, L is sequence length

- It's well-known that vector-matrix multiplication is a perfect fit for PIM technology
  - Accelerating vector-matrix multiplication of projections and feed-forward networks.





GPT-3 Acceleration with PIM Technology

- SEC is working on the implementation of GPT-3 Accelerator by GPU and FPGA, concurrently
- PIM is the key for small latency to keep SLA, with single batch GPT-3 inference
- The accelerator will be released this autumn.

Wrap-up

Lessons from HBM-PIM

- DRAM technology is <u>really</u> large and slow

- Custom design + EDA tool flow: best solution for limited resource but,
  - Handling EDA design flow in DRAM technology process is much complex than logic technology process
  - True hell: physical interface and timing analysis between legacy DRAM logic and EDA designed logic

- Energy efficiency with PIM is much better than our expectation
  - Most energy consumption comes from external channel

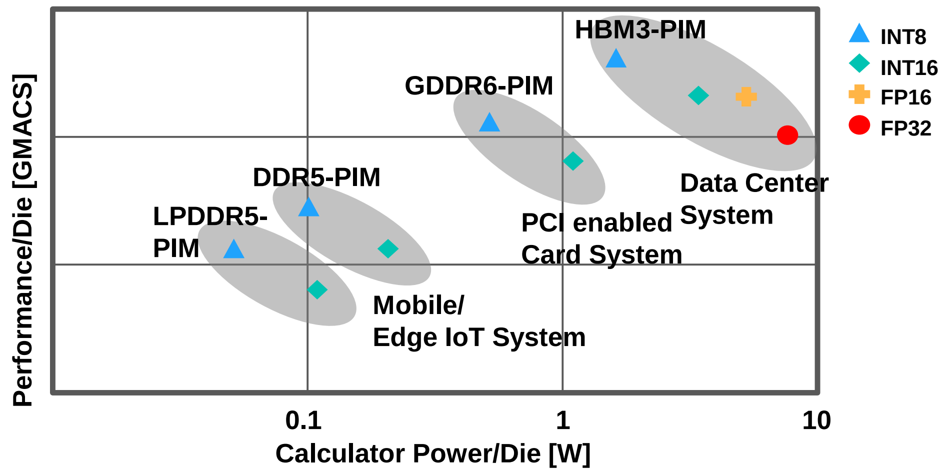
- Need specific protocol and interfaces for PIM
  - Deterministic interface has many pros for performance, but many engineering issues to use it for near-DRAM cell computing

- Software is key to success
  - "All that glitters is not gold" without software
  - Near-DRAM cell Computing is meaningless if there is no "end to end performance gain" analysis

Our Efforts

- Wider target applications
  - PIM supporting multiple functions
    - GEMV from MAC, MAD
    - Activation functions
    - INT8/INT16/BF16/FP16/FP32/FP64 support

Prospective PCU supported data format



- Various DRAM types: Applying different trade-off bias for the other DRAM
  - LPDDR: energy efficient first (reduce off-chip data transfer)
  - HBM: boosting bandwidth first (bank, rank parallelism)

Our Efforts

- New standards
  - Command truth table for near-DRAM cell computing
    - Add multiple bank activation/precharge command, PIM read/write/load/store commands, PIM execution commands
  - AC timing parameters
    - For practical implementation, several timing parameters are needed to redefined (such as tRCD at multibank activation)

- Collaborate with academia and industry
  - Brilliant ideas for PIM architecture
  - On the next-generation memory devices
  - Also targeting wider functions

Moreover..

- SEC is also developing CXL based near-memory processing called CXL-PNM, after AXDIMM (youtube)

PNM
Processing near Memory
Samsung developed CXL interface-based
PNM technology for the first time in the world.

END