Scaling of Memory Performance and Capacity with CXL Memory Expander

August, 2022 | Samsung Electronics Co., Ltd.

Agenda

- Industry Trends and Challenges
- Introduction of CXL (Compute Express Link)
- CXL Memory Expander Features
- SMDK: Unified Software Solution for CXL
- Application Benchmark Test Results
- Summary and Future Plan
Industry Trends and Challenges

Massive demand for data-centric technologies and applications

Memory bandwidth and density not keeping up with increasing CPU core count

Need a next gen interconnect for heterogeneous computing and server disaggregation
The Fast-Growing Computing Workloads

- Large-scale adoption of AI and ML
- Smarter devices
- Hyper-connected networks
- Super-intelligent services
- Digital transformation

First Era
- Perceptron
- NETtalk
- ALVINN
- LeNet-5
- RNN for Speech

Modern Era
- BiLSTM for Speech
- TD-Gammon v2.1
- Deep Belief Nets & Layer-wise pretraining
- AlexNet
- ResNets
- GPT-3
- LaMBDA

Pandemic

Large-scale adoption of AI and ML
- Smarter devices
- Hyper-connected networks
- Super-intelligent services
- Digital transformation

1960 - 2020
Evolution of Hyperscale Computing Environment

- From Converged to Composable Architecture

Converged Architecture
- TOR based Rack Scalable Architecture
- Network Challenge

Hyper-Converged Architecture
- Server & Storage Combined Architecture
- Divergence Challenge
- SmartNIC, MS Catapult, AWS Nitro

Disaggregated / Composable Architecture
- Pooled Arch.: Memory, Compute, Storage
- Interconnect Challenge
The Rising Need for Better Connectivity

- Can be tailored and optimized for various AI applications

**SoC Interconnect**
DIE / PACKAGE

**Processor Interconnect**
NODE

**Data Center Interconnect**
DATA CENTER

**Customer Interconnect**
MOBILE / BROADBAND

A new class of interconnect for device connectivity in the era of AI
CXL: Solution for the Era of HPC

- CXL as the core of composable computing infrastructure

**Key Features of CXL Interface**

- Cache Coherence
- Connectivity
- Byte Addressable
- Low Latency
CXL is a high-performance, low-latency protocol that leverages PCIe physical layer

- High-speed and low-latency interconnect
- Leverages PCIe Physical layer (PCIe 5.0, PCIe 6.0)
- Supports various types of memories (volatile, non-volatile)
- CPU and CXL device memory coherency
- Supports switching and memory pooling
- Supports link level integrity and data encryption
- Open standard (non-proprietary)
- Broad industry support in CXL consortium
- Regular specification updates (CXL 1.1, CXL 2.0, CXL 3.0)
CXL Use Cases (1/2)

- **Capacity and Bandwidth Expansion**

**Capacity Expansion**
- IMDB Server
  - CPU 0
  - xTB DRAM
  - CPU 0
  - xTB DRAM
- IMDB Server
  - CPU 0
  - yTB DRAM
  - CPU 1
  - yTB DRAM

**IMDB Server**
- CPU 0
- yTB DRAM
- CPU 1
- yTB DRAM
- CXL zTB
- CXL zTB

**Bandwidth Expansion**
- IMC Server
  - CPU 0
  - xGB DRAM
  - CPU 0
  - xGB DRAM
  - CPU 0
  - xGB DRAM
  - CPU 0
  - xGB DRAM
- IMC Server
  - CPU 0
  - yGB DRAM
  - CPU 0
  - yGB DRAM
  - CPU 0
  - yGB DRAM
  - CPU 1
  - yGB DRAM

**IMC Server**
- CPU 0
- yGB DRAM
- CPU 0
- yGB DRAM
- CPU 0
- yGB DRAM
- CPU 0
- yGB DRAM
- CPU 1
- yGB DRAM
- CPU 1
- yGB DRAM
- CPU 1
- yGB DRAM
- CPU 1
- yGB DRAM
- CXL zGB
- CXL zGB
- CXL zGB
- CXL zGB
## CXL Use Cases (2/2)

### Tiering and Pooling

<table>
<thead>
<tr>
<th>Memory Tiering</th>
<th>IMC Server</th>
<th>CPU 0</th>
<th>CPU 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IMC Server</td>
<td>CPU 0</td>
<td>CPU 1</td>
</tr>
<tr>
<td></td>
<td>CPU 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPU 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xTB DRAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xTB DRAM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory Pooling</th>
<th>IMC Server</th>
<th>CPU 0</th>
<th>CPU 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IMC Server</td>
<td>CPU 0</td>
<td>CPU 1</td>
</tr>
<tr>
<td></td>
<td>CPU 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPU 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xTB DRAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xTB DRAM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory Pooling</th>
<th>IMC Server</th>
<th>CPU 0</th>
<th>CPU 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IMC Server</td>
<td>CPU 0</td>
<td>CPU 1</td>
</tr>
<tr>
<td></td>
<td>CPU 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPU 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xTB DRAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xTB DRAM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory Pooling</th>
<th>MEMORY BOX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CXL yTB</td>
</tr>
<tr>
<td></td>
<td>CXL yTB</td>
</tr>
</tbody>
</table>
CXL Memory Expansion Solution

- Doubled Capacity than Conventional Memory

Max. 8TB for 1CPU

Max. 16TB for 1CPU

Note: Max capacity varies with system configurations
CXL Memory Expander

- New Solution for Memory Dominant Applications

Data Acceleration
High Capacity / Bandwidth
Enhanced Security / RAS

CXL interface

CXL Memory Expander
CXL Controller
DDR

GPU
CPU
Smart NIC

DDR
CXL Memory Expander Line-up

- Built with FPGA and ASIC Controller

<table>
<thead>
<tr>
<th>Year</th>
<th>FPGA</th>
<th>Host (Controller)</th>
<th>ASIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>'21</td>
<td>PCIe 3.0 (x16)</td>
<td>3200, 128GB</td>
<td></td>
</tr>
<tr>
<td>'22</td>
<td>PCIe 5.0 (x8)</td>
<td>Media (DRAM)</td>
<td>4800+, 512GB</td>
</tr>
</tbody>
</table>

As of August, 2022
CXL Memory Expander (1/3)

- Solution Overview
CXL Memory Expander (2/3)

Product Features

- Form Factor - EDSFF (E3.S)
- Media - DDR5 4800
- Module Capacity - Max 512 GB
- CXL Link Width - x8
- Maximum CXL Bandwidth - 32GB/s (PCIe 5.0)
- Other Features - RAS, Interleaving, Diagnostics etc.
- Availability - Q3’22 for evaluation/testing
CXL Memory Expander (3/3)

- **Supported Features**
  - CXL 2.0
  - Device Type: Type 3
  - Support viral and data poisoning
  - Memory error injection
  - Multi-symbol ECC
  - Media scrubbing
  - Post package repairs (hard/soft)

* Image Source: CXL Consortium
**SMDK*, Unified Interface for Memory**

* Scalable Memory Development Kit

SW development kit to enable **Software-Define Memory** system on heterogeneous memories

**Plugin**
- Two selectable paths, Compatible and Optimization Path, without or with modification of application SW
- Intelligent Tiering Engine supports memory tiering scenarios with priority, capacity, bandwidth, and so on
- Memory Pool Management supports scalability reflecting memory request status and system resource

**Kernel**
- Memory Partitioning allows logical memory views for heterogeneous physical DRAM and CXL memory
Benefits of SMDK

- **Unified SW Solution**
  Full-stack SW all about heterogenous memory system

- **Client Experience**
  Transparent as well as Optimized Memory uses

- **CXL Ecosystem**
  OSS for CXL Industry and Research field

SMDK is available as open source on GitHub

- https://github.com/OpenMPDK/SMDK/releases/tag/smdk_v1.1
Experimental Setup

Configuration of Test Bed

- SW
  - ML/AI
    - Bert
    - NAS NET
  - IMDB
    - Mem cached
    - Redis
  - BM
    - MLC
    - Stream
  - Container
  - SMDK
    - OpenMPDK / SMDK Public

- HW
  - CXL CRB
  - DDR5 4800
  - CXL DRAM (FPGA)
  - CXL DRAM (ASIC)
Memory Benchmark Test Results

- Comparable Performance with DDR Memory

![Diagram showing comparison between DDR and CXL-FPGA/ASIC in terms of normalized bandwidth and ML/AI process]

- Normalized Bandwidth
  - DDR
  - CXL-FPGA
  - CXL-ASIC

- MLC 1:1 R/W
  - DDR: 1.0
  - CXL-FPGA: 0.88 (4.6x)
  - CXL-ASIC: 0.92 (4.7x)

- STREAM Copy
  - DDR: 1.0
  - CXL-FPGA: 0.19
  - CXL-ASIC: 0.19

System Test Results (ML/AI)

- ML/AI Applications (BERT* & NASNet**)

![Bar chart showing Inferences per Minutes (Normalized)](See appendix for detail test condition)

* Bidirectional Encoder Representations from Transformers

** Neural Architecture Search Network
System Test Results (IMDB)

- IMDB Redis* Memory Usage (Scale-up vs Scale-out)

### Single Node (DDR+CXL FPGA)
- **Client**
  - Set 60GB
  - Get 60GB
- **Redis**
  - DDR5 (32GB)
- **CXLMem** (64GB)
- **System #1**

### 2-Node Cluster (DDR x 3)
- **Client**
  - Set 60GB
  - Get 60GB
- **Redis**
  - DDR5 (32GB)
  - DDR5 (32GB)
  - DDR5 (32GB)
  - **System #1**
- **Cluster**
  - **System #2**

### Scale-up vs Scale-out
- **Performance [MB/s]**
  - **128B SET**: Single-Node (DRAM + CXL) 30, 2-Node Cluster (DRAM) 49
  - **4KB**: Single-Node (DRAM + CXL) 455, 2-Node Cluster (DRAM) 172
  - **1MB**: Single-Node (DRAM + CXL) 699, 2-Node Cluster (DRAM) 186
  - **2.64x**
  - **128B GET**: Single-Node (DRAM + CXL) 27, 2-Node Cluster (DRAM) 66
  - **4KB**: Single-Node (DRAM + CXL) 496, 2-Node Cluster (DRAM) 173
  - **1MB**: Single-Node (DRAM + CXL) 659, 2-Node Cluster (DRAM) 189
  - **2.86x**

* Remote dictionary server

(See appendix for detail test condition)
A Proven Memory Expansion Solution

- Increasing System Memory Capacity
- Widening Memory Bandwidth
- Supporting RAS/Security based on Memory Controller

2X Increase
83% Increase
RAS Security
Summary and Future Plan

- AI and pandemic drive demand for memory bandwidth and capacity, and new interconnect standard CXL allows expansion of memory.

- Samsung developed the industry’s first ASIC-based 512GB CXL memory expander, which will be available for early evaluation this quarter.

- Memory intensive applications such as IMDB and AI/ML have been tested on CXL memory expander with an open-source software, SMDK.

- Samsung to cooperate further on CXL 3.0 and beyond, and provide next-gen memory solutions like memory disaggregation, SDM*, and more.

* Software-defined memory
Enhanced Data Service
AI/ML
NLP, Recommendation
Edge Computing

Industry First CXL™ Memory Expanders
Appendix

Test Condition (ML/AI and IMDB)

ML/AI

For BERT and Nasnet
TensorFlow (CPU) >= 1.11.0+, Python ~ 3.7, Numpy < 1.20.0

For BERT
Multi-process, 3 cores/process, batch-size:128,
max_seq_num:256, num-test-data/process: 512
dataset=CoLA
do_train=true, do_eval=true, data_dir=$GLUE_DIR/CoLA
vocab_file=vocab.txt
init_checkpoint=$BERT_BASE_DIR/bert_model.ckpt
max_seq_length=128, train_batch_size=32
learning_rate=2e-5, num_train_epochs=3.0

For NASNet
Multi-process, 3 cores/process, batch-size: 100,
eval_image_size:236, num-test-data/process: 200
dataset_name=imagenet, num_preprocessing_threads=4
labels_offset=0, model_name=inception_v3
preprocessing_name=inception_v3
moving_average_decay=None, quantize=False, use_grayscale=False

IMDB

For scale-up vs scale-out

Redis-server :master
cluster-enabled yes
cluster-node-timeout 300000
save ""
stop-writes-on-bgsave-error yes
rdbcompression yes
rdbchecksum yes
rdb-del-sync-files no
repl-diskless-sync no
rdb-del-sync-files no
replica-serve-stale-data yes
replica-read-only yes
repl-diskless-sync-delay 5
repl-diskless-load disabled
repl-disable-tcp-nodelay no
replica-priority 100
client-output-buffer-limit replica 0 0 0
maxclients 1000000
maxmemory-policy noeviction
maxmemory-samples 10
maxmemory-eviction-tenacity 100
replica-lazy-flush no
lazyfree-lazy-user-del no
lazyfree-lazy-user-flush no
oom-score-adj no
oom-score-adj-values 0 200 800
disable-thp yes

Redis-server :replica
save ""
port 6380
replicaof 127.0.0.1 6379
replica-read-only yes
stop-writes-on-bgsave-error yes
rdbcompression yes
rdbchecksum yes
rdb-del-sync-files no
repl-diskless-sync no
rdb-del-sync-files no
replica-serve-stale-data yes
replica-read-only yes
repl-diskless-sync-delay 5
repl-diskless-load disabled
repl-disable-tcp-nodelay no
replica-priority 100
client-output-buffer-limit replica 0 0 0
maxclients 1000000
maxmemory-policy noeviction
maxmemory-samples 10
maxmemory-eviction-tenacity 100
replica-lazy-flush no
lazyfree-lazy-user-del no
lazyfree-lazy-user-flush no
oom-score-adj no
oom-score-adj-values 0 200 800
disable-thp yes