Near Data Processing in AI Era: Challenges and Opportunities

Memory System Research
Eui-cheol Lim
## Contents

1. SK Hynix & MSR introduction (3)
2. AI era Architecture trend (5)
3. Memory based Solution Projects (5)
4. Data Hierarchy – ultimate near data processing architecture (4)
## SK hynix Business

### SK hynix’s Product Portfolio

<table>
<thead>
<tr>
<th>Business Category</th>
<th>DRAM</th>
<th>NAND</th>
<th>CIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datacenter Solutions</td>
<td>DDRx, HBMx</td>
<td>SATA eSSD, PCIe eSSD</td>
<td></td>
</tr>
<tr>
<td>Mobile Solutions</td>
<td>LPDDRx</td>
<td>eMMC, UFS</td>
<td>eMCP, uMCP</td>
</tr>
<tr>
<td>Computing Solutions</td>
<td>DDRx, LPDDRx</td>
<td>PCIe cSSD, SATA cSSD</td>
<td>VGA, HD, FHD~8Mp</td>
</tr>
<tr>
<td>Graphics Solutions</td>
<td>GDDRx, DDRx, LPDDRx, HBMx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datacenter Solutions</td>
<td>DDRx, LPDDRx</td>
<td>SATA eSSD, PCIe eSSD</td>
<td></td>
</tr>
<tr>
<td>Mobile Solutions</td>
<td>LPDDRx</td>
<td>eMMC, UFS</td>
<td>eMCP, uMCP</td>
</tr>
<tr>
<td>Computing Solutions</td>
<td>DDRx, LPDDRx</td>
<td>PCIe cSSD, SATA cSSD</td>
<td>VGA, HD, FHD~8Mp</td>
</tr>
<tr>
<td>Graphics Solutions</td>
<td>GDDRx, DDRx, LPDDRx, HBMx</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Automotive Solutions</strong></td>
<td>DDRx, LPDDRx, GDDRx</td>
<td></td>
<td>eMMC</td>
</tr>
<tr>
<td><strong>Consumer Solutions</strong></td>
<td>DDRx, LPDDRx</td>
<td>eMMC, UFS</td>
<td></td>
</tr>
<tr>
<td><strong>Computing Solutions</strong></td>
<td>DDRx, LPDDRx</td>
<td>PCIe cSSD, SATA cSSD</td>
<td>VGA, HD, FHD~8Mp</td>
</tr>
<tr>
<td><strong>Graphics Solutions</strong></td>
<td>GDDRx, DDRx, LPDDRx, HBMx</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MSR (Memory System Research) in SK hynix

SK hynix

- Global Sales & Marketing
  - DRAM Product Planning
  - NAND Product Planning
  - eSSD Business TF
  - Marketing
  - Sales
  - GSM Strategy
  - New Product Enablement
    - Memory System Research
    - SK hynix Greater America
      - SK hynix Greater China
        - SK hynix Deutschland
        - SK hynix Japan
        - SK hynix Singapore
        - SK hynix India
        - SK hynix Hong Kong
  - CIS Business
- Chief Product & Production Officer
- Quality & Reliability Assurance
- R&D
- CIS Business
- Digital Transformation
- Communication & External Affairs
- Finance
- Corporate Strategy & Planning
- Corporate Culture
- Procurement
- Safety Health Environment
- Ethics Management
- Global Planning & Operation
- Happiness Culture Office
- Future Technology

1) Footnote: As of July, 2020
MSR – Research Area

**Memory System for High Capacity Memory Server**
- Advanced HW/SW solution for 2-tier memory (e.g. DRAM + MDS or PCM) to utilize high capacity and/or non-volatility
- 2nd-tier memory extension solution w/ new interconnect
- Management SW stack for new memory hierarchy
- Quantitative analysis of next-generation memory architecture

**Storage System for Next Cloud Storage**
- Core technology for next-generation storage devices (e.g. ZNS, KV SSD) suited for Software Defined Storage
- ZNS-based next generation SSD solutions
- Key-Value SSD
- Computational Storage:
  - In-line (Compression, Encryption) SSD
  - In-situ (Analytics) processing SSD

**Emerging Technology for AI/ML in Data Center**
- Memory-based deep-learning computing solutions for higher energy efficiency
- Near Data Processing for DLRM (Deep Learning Recommendation Model)
- AI Computing-in-Memory:
  - PNM (Custom HBM w/ an AI accelerator)
  - ANMAC (Analog MAC w/ resistive memory)
- AI SW stack for memory-based AI solutions
Contents

1. SK Hynix & MSR introduction
2. AI era Architecture trend
3. Memory based Solution Projects
4. Data Hierarchy – ultimate near data processing architecture
Computer vs. Human Brain

- AI has already proven that it is more capable than human in certain applications
- However in energy perspective, human brain is much more efficient than current computer system
Human Brain vs. Neuromorphic vs. von Neumann

- Neuromorphic computing looks far better than von Neumann architecture, but it is in premature stage yet
Energy Cost of Data Movement

- Data movement consumes more than 60% of power in modern processors
- AI algorithm is more memory bounded than conventional app.

Power Portion by Data Movement

- More than 60% of power is consumed by data movement
  - Analyze power consumption in popular google apps

Low Operational Intensity

- In most cases, NN performances are bounded by BW
  - Recent NN Accelerators use internal memory to reduce bottleneck, but it is not sufficient

Norman P. Jouppi, “In-Datencenter Performance Analysis of a Tensor Processing Unit,” ISCA’2017

Amirali Boroumand, “GoogleWorkloads for Consumer Devices: Mitigating Data Movement Bottlenecks,” ASPLOS’18
Operational Intensity of AI algorithm decides the appropriate computing architecture (Host vs. PIM)

- High Operational Intensity: Host processing is better in performance & efficiency
- Low Operational Intensity: PIM is better in performance & efficiency
Interconnect: good news for NDP

- Interconnect will be the core of the Server system which has heterogeneous architecture
- Intel introduces the CXL interconnect which enables Intel CPU will be the center of heterogeneous Computing System.
- Opportunity: Value added Memory solution
  - Unlike conventional memory system such as DIMM, CXL supports hand shaking protocol which enables us to build additional value on it. Ex) DRAM cache, Data processing engine...

Interconnect will be the core of the Server system which has heterogeneous architecture

Intel introduces the CXL interconnect which enables Intel CPU will be the center of heterogeneous Computing System.

Opportunity: Value added Memory solution

- Unlike conventional memory system such as DIMM, CXL supports hand shaking protocol which enables us to build additional value on it. Ex) DRAM cache, Data processing engine...
Contents

1. SK Hynix & MSR introduction
2. AI era Architecture trend
3. Memory based Solution Projects
4. Data Hierarchy – ultimate near data processing architecture
Recommendation Model

- Recommendation is one of the key AI services that provides majority of revenue to datacenter companies

- DLRM is an advanced, open source Deep Learning Recommendation Model (from Facebook)
  - Composed of 3 layers (Embedding, Feature Interaction, Top/Bottom Multilayer Perceptron) and each layer has different computing characteristics
  - As the model size and # of data becomes large, the memory requirements for embedding goes higher
RPM (Recommendation Pooled Memory) Project Overview

- **Limitations in Conventional System**
  - Due to different compute characteristics of each layer, conventional system is using two different compute environment (Zion) – 8 socket CPU Server for Embedding and OAM (OCP Accelerator Module) for Feature Interaction, Top/Bottom MLP
  - However, (1) CPU has a limited memory capacity and bandwidth to fulfill the ever increasing memory requirements for Embedding; (2) CPU is not adequate for small IO applications

- **RPM: Applying Near Data Processing concept for Embedding layer to resolve the limitations in CPU**

![Diagram showing the comparison between conventional system and RPM](image)
RPM (Recommendation Pooled Memory) Project Challenges

- Several challenges to overcome when we adopt Near Data Processing concept to Embedding layer:
  - Bandwidth limitation between the x86 CPU server (or OAM) and RPM
  - Load balancing through the RPM modules
  - Utilization for each RPM module
  - RPM architecture for advanced RM algorithm

  **Load Balancing?**
  - How we could fully utilize all the RPM?

  **Bandwidth Limitation?**
  - How we could overcome the bandwidth limitation for interconnect?

  **RPM Utilization?**
  - How we could fully utilize the memory capacity and bandwidth of RPM?
  - What is the best compute engine for NDP unit to support RM inference as well as training?

  **RPM Architecture for Advanced RM?**
  - What is the best RPM architecture for advanced RM algorithm?
ACiM (Analog Computing in Memory) Project Overview

- Meaningful synergy between AI computing technologies and Memory technologies
  - PNM & PiM - Computing digital logic located near or in memory.
  - ACiM - Computing logic is made of memory cell (ReRAM, PCRAM and so on.)

- Target to achieve higher ‘Perf./Power’ and lower ‘Energy consumption’ than those of digital accelerators
  - Perf./Power - Digital MAC: < 2TOPs/Watt vs. Analog MAC array: > 200 TOPs/Watt
  - Power consumption: < 1 Watt

![Resistive memory cell array for MAC computations]

[AI Accelerator using ACiM tech (Analog MAC)]
ACiM (Analog Computing in Memory) Project Challenges

- Need to consider different things in all levels from a device level to an application level
- Need a holistic approach and tight collaboration among various R&D participants

**Device**

- Multi-level cell
- Retention & Reliability
- Read speed
- I-V linearity

**AMAC (Analog MAC)**

- ADC/DAC resolution
- Optimum sub-array size
- Ultra high Perf./Power in AMAC
- Analog aware inference

**Accelerator**

<table>
<thead>
<tr>
<th>SRAM</th>
<th>DRAM Cont.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital non-MAC</td>
<td>Analog MAC</td>
</tr>
<tr>
<td>Activation, Pooling, Vector Proc.</td>
<td></td>
</tr>
<tr>
<td>Tile 0</td>
<td>Tile 1</td>
</tr>
<tr>
<td>Tile 2</td>
<td>Tile 3</td>
</tr>
<tr>
<td>Tile 4</td>
<td>Tile n</td>
</tr>
<tr>
<td>Input / Control</td>
<td></td>
</tr>
<tr>
<td>Host I/F</td>
<td></td>
</tr>
</tbody>
</table>

- Energy efficiency in digital part
- SRAM minimization
- AMAC controller

**Application & SW**

- Low precision algorithm
- Application specific
- ACiM specific SW toolkits (e.g.) TFlite, Edge TPU compiler
Contents

1. SK Hynix & MSR introduction
2. AI era Architecture trend
3. Memory based Solution Projects
4. Data Hierarchy – ultimate near data processing architecture
Data Hierarchy – NDP in All Memory Layers

- **Architecture Concept** – Every Memory layers have own processing element
  - Minimizing data movement for the entire memory hierarchy
  - Data placed based on the data & workload’s characteristics
  - Compute data where data reside

**Memory Hierarchy**

- CPU
  - Memory only
- Cache
- Main Memory
- Storage Class Memory
- Storage (SSD/HDD)

**Data Hierarchy**

- CPU
  - Memory + Processing Element
- Cache
- Main Memory
- Storage Class Memory
- Storage (SSD/HDD)

**Data Hierarchy Concept**

- Data placement with respect to data and workload characteristics

- Job distribution
  - Data
  - Cache
  - DRAM
  - SCM
  - SSD

---

19
Research Topics – Data placement

- **Data placement**
  - How to *classify* the characteristics of data?
  - How to *place* the data at the appropriate layer
  - Whether the characteristics of data will be *determined only by the data itself*, or by the algorithms that utilize the data

---

**Data Hierarchy System Architecture**

- **Data**
  - **CPU**
  - **Data Layer 0**
  - **Data Layer 1**
  - **Data Layer 2**
  - **Data Layer 3**

---

**Legend**

- Processing Elements (CPU/GPU/FPGA/HWA/PE...)
- Memory/Storage Media (HBM/DRAM/SCM/SSD)
- **Data Placement**
- **Job distribution**
- **Communication**
Research Topics – Processor Architecture

- Processor architecture per layer
  - What is the feasible structure of processing elements for each layer?
  - Domain specific architecture is the key

**Exploration of Processor Architecture per Layer**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Processor Architecture?</th>
<th>What’s the Goal of the System?</th>
<th>What’s the main role of each layer?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Layer 0</td>
<td>?</td>
<td></td>
<td>Matrix Multiplication</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Activation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pre Processing</td>
</tr>
<tr>
<td>Data Layer 3/4</td>
<td>?</td>
<td>Deep Learning Domain Specific Acceleration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AI</td>
<td></td>
</tr>
</tbody>
</table>

Processor Architecture Candidates

- Processing Element
- Network
Research Topics – Job Distribution & Comm.

- Job distribution & Communication
  - In terms of job distribution, Data Hierarchy is similar to heterogeneous computing
  - Framework is required to assign a job to each layer and aggregate the results
THANK YOU

euicheol.lim@sk.com