NVDIMM solution for embedded systems

Presenter: Igor Sharovar, CTO, Truememorytechnology

igor.Sharovar@truememorytechnology.com

Keiji Kimura, Director, Green Computing System Research Organization at Waseda University, Tokyo, Japan

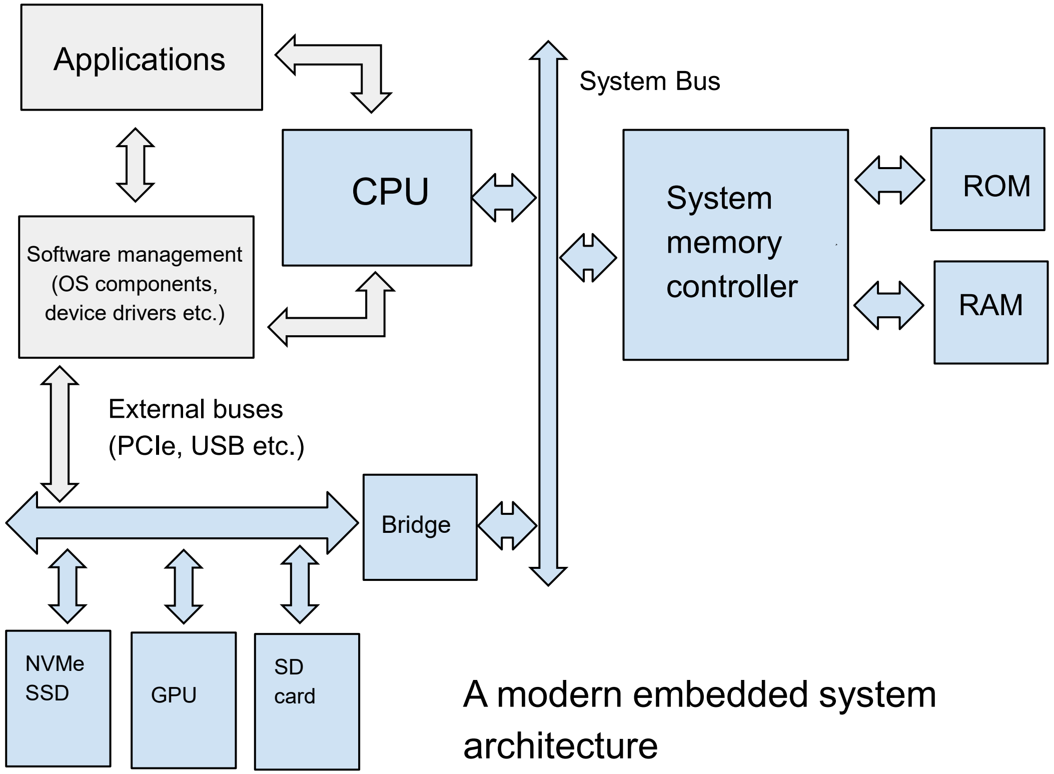
keiji@waseda.jp <mailto:Keiji@waseda.jp>

The current challenges for high-performance aplications

- **Embedded systems** today face significant constraints with high-performance applications, primarily due to power consumption and CPU utilization.

- **Traditional SSDs** based on NVMe PCIe gen3 x4 interfaces, while fast (up to 4 GB/s bandwidth), demand substantial CPU resources.

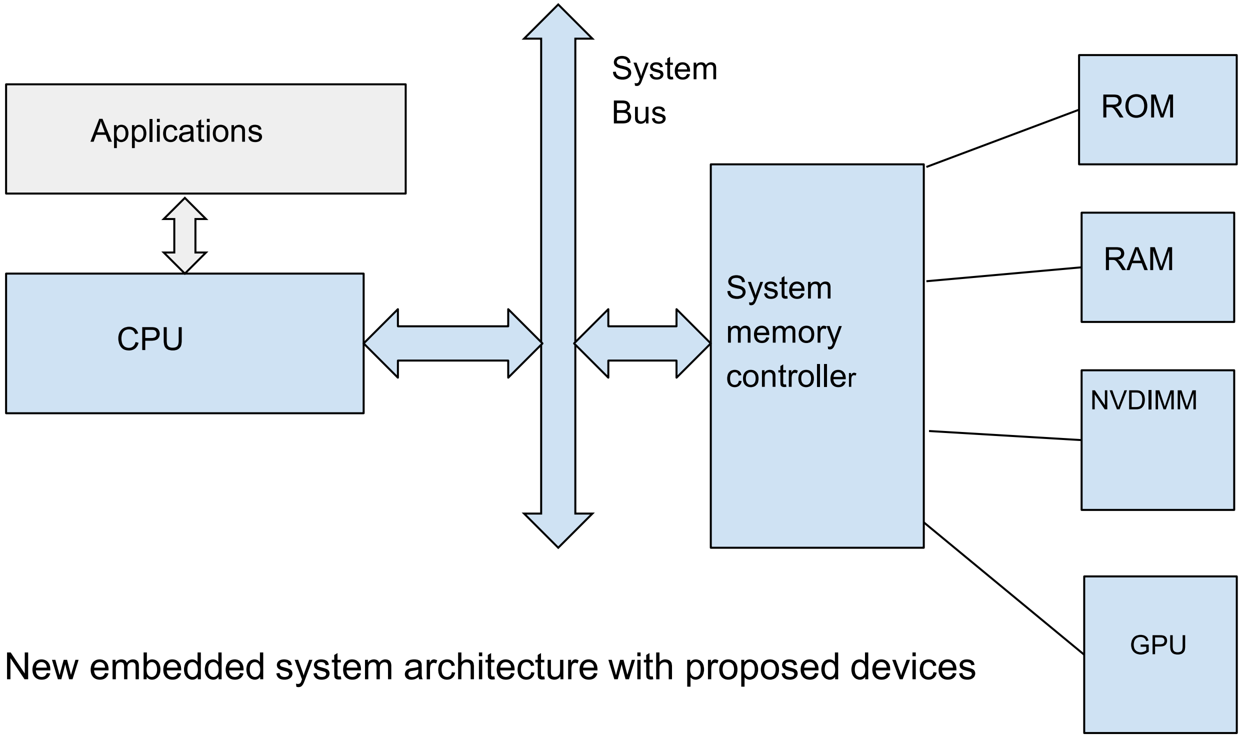
- **Results in higher power usage** and inefficiencies due to extra memory copies required from device-mapped memory (PCIe) to application memory.



A modern embedded system architecture

A new NVDIMM solution

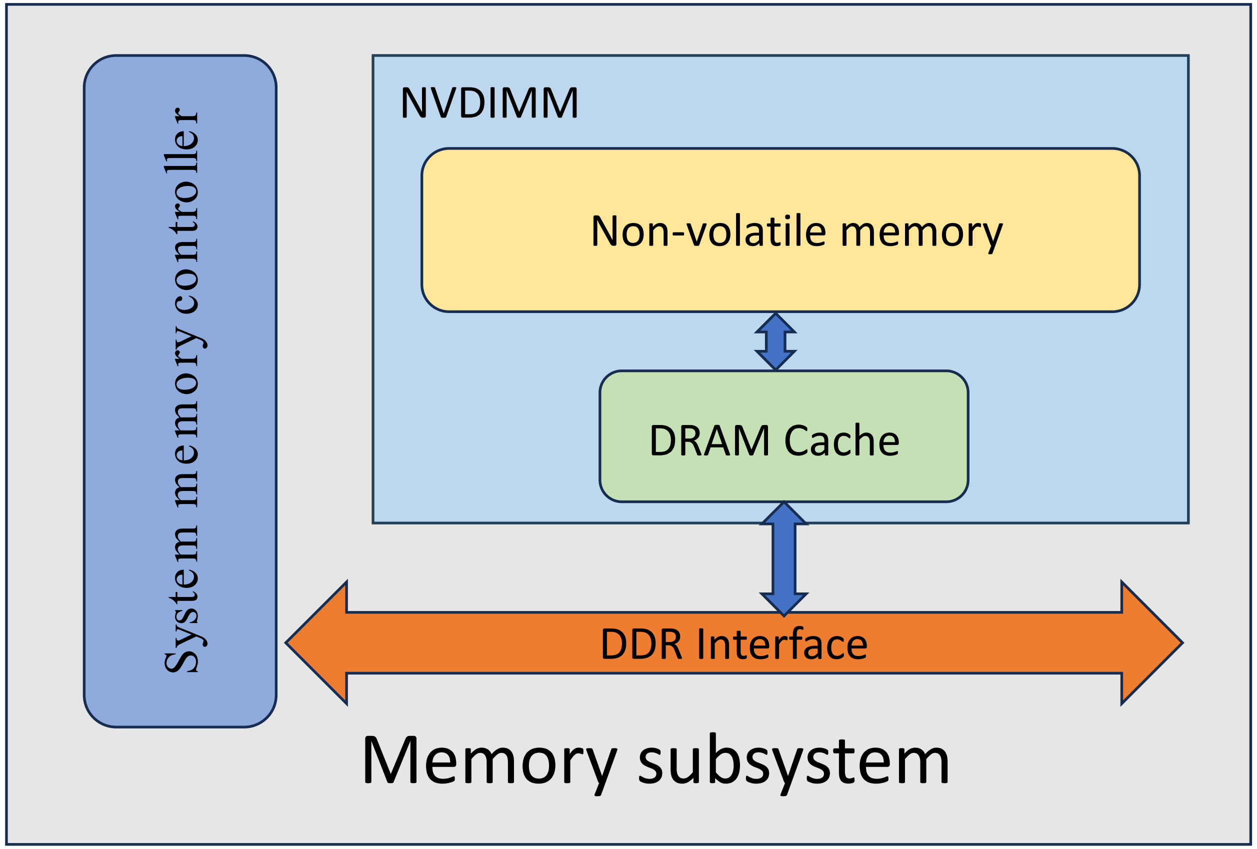
- **Unmatched Bandwidth:** Achieve up to 20 GB/s with DDR4 or 40 GB/s with DDR5.
- **Optimized CPU Utilization:** Eliminate the need for software management components, allowing applications to access data directly from non-volatile storage.
- **Direct Data Access:** Applications can retrieve data straight from I/O devices, reducing memory copy overheads.
- **Compact System Footprint:** Integrate system and non-volatile memory through a unified interface.



New embedded system architecture with proposed devices

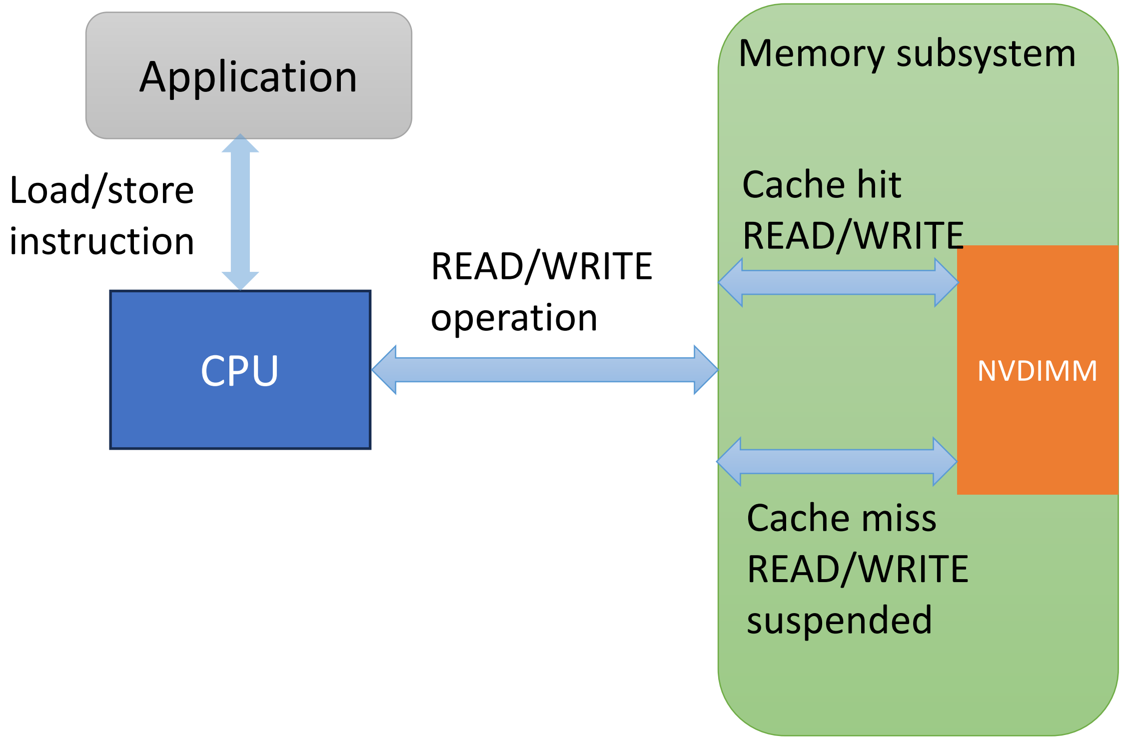
Technology

- **The size of NVDIMM module** equal the size of non-volatile memory. For example, a 512GB NVDIMM consists 512 GB Flash and 2 GB DRAM
- **All Read/Write operations** come to the DRAM cache. In a case of a cache miss a memory operation is suspended in the system memory controller and resumes after completion of swap data between flash and DRAM
- **The solution supports DDR4 and DDR5 protocols**. The system memory controller has a minor modification to work with the NVDIMM. We plan to reuse existing IP of the memory controller.
- **The core of the technology** is a patented method that synchronizes access to the cache with the device operations without violating standard DDR protocol



Software–NVDIMM communication

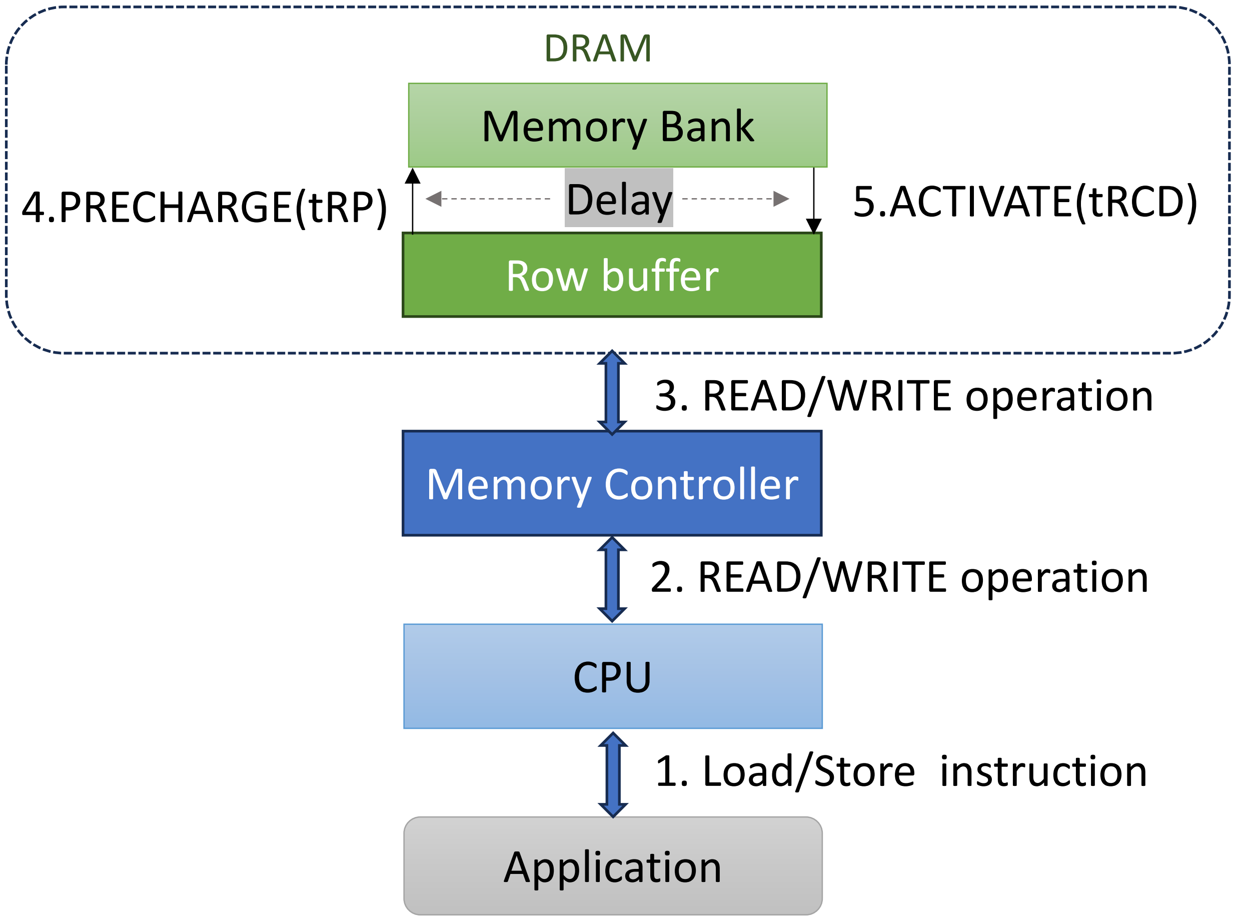
- **Byte access** for both RAM and NVRAM.
- **Load/store** instructions to access NVDIMM. There is no additional software management.
- **Cache hit**. Execute a normal DDR command.
- **Cache miss.** A memory command has been suspended but does not block the following memory commands. The CPU will remain idle until cache miss is resolved.



FMS

Non-volatile memory emulator for embedded systems

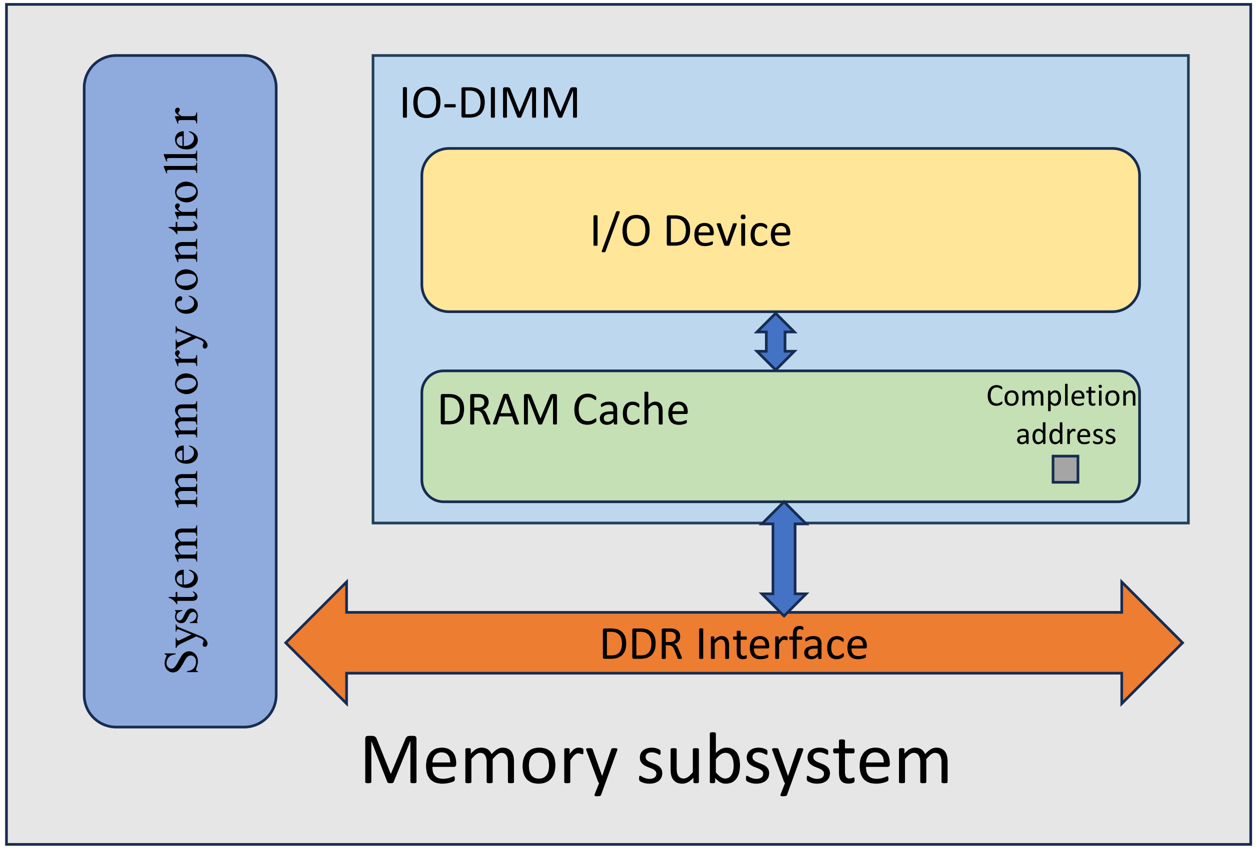
- **Emulation environment** on a base of a Zilinx FPGA board running Linux
- **Modified the RTL Memory Interface Generator** (MIG) of the FPGA to inject additional read/write delays for ACT and PRE commands. The injected delays emulate non-volatile operations of NVDIMM that require bigger latency than read/write DDR commands.
- **Software support** to work with NVDIMM devices. The software support includes C language functions to allocate and deallocate the NVDIMM region and kernel modifications to explicitly flush CPU cache to NVDIMM.



Emulation of a non-volatile operation in DRAM based main memory

IO-DIMM

- **An I/O device uses the DRAM cache** to communicate with the CPU. Examples of I/O devices are sensors, network cards, GPUs, etc.
- **An application allocates system memory** that is a part of the DRAM cache of IO-DIMM
- **There are no polling methods** to wait to complete the I/O device operations.
- **An application reads the completion address** to get the status of an I/O operation. If the operation is incomplete, the IO-DIMM module and the system memory controller suspend the Read memory command. CPU is in an idle state. Otherwise, if the I/O operation is completed, the Read command data is returned to the CPU, and the application gets the result.



References

[1] www.truememorytechnology.com
[2]https://www.jstage.jst.go.jp/article/transinf/E104.D/5/E104.D_2020EDP7092/_pdf/-char/en

Q & A