GLOBAL EDGE

Intelligence Of Things

ARM Memory Management

By,
CM Naveen Reddy.

Overview

Page

Memory Management Unit

Cache

Page, Page fault, Paging and Demand paging

Paging: divide Memory into fixed sized **Pages** for both Virtual and physical memory

•Another terminology
–A Virtual Page: ***page***
– A physical page: ***frame***

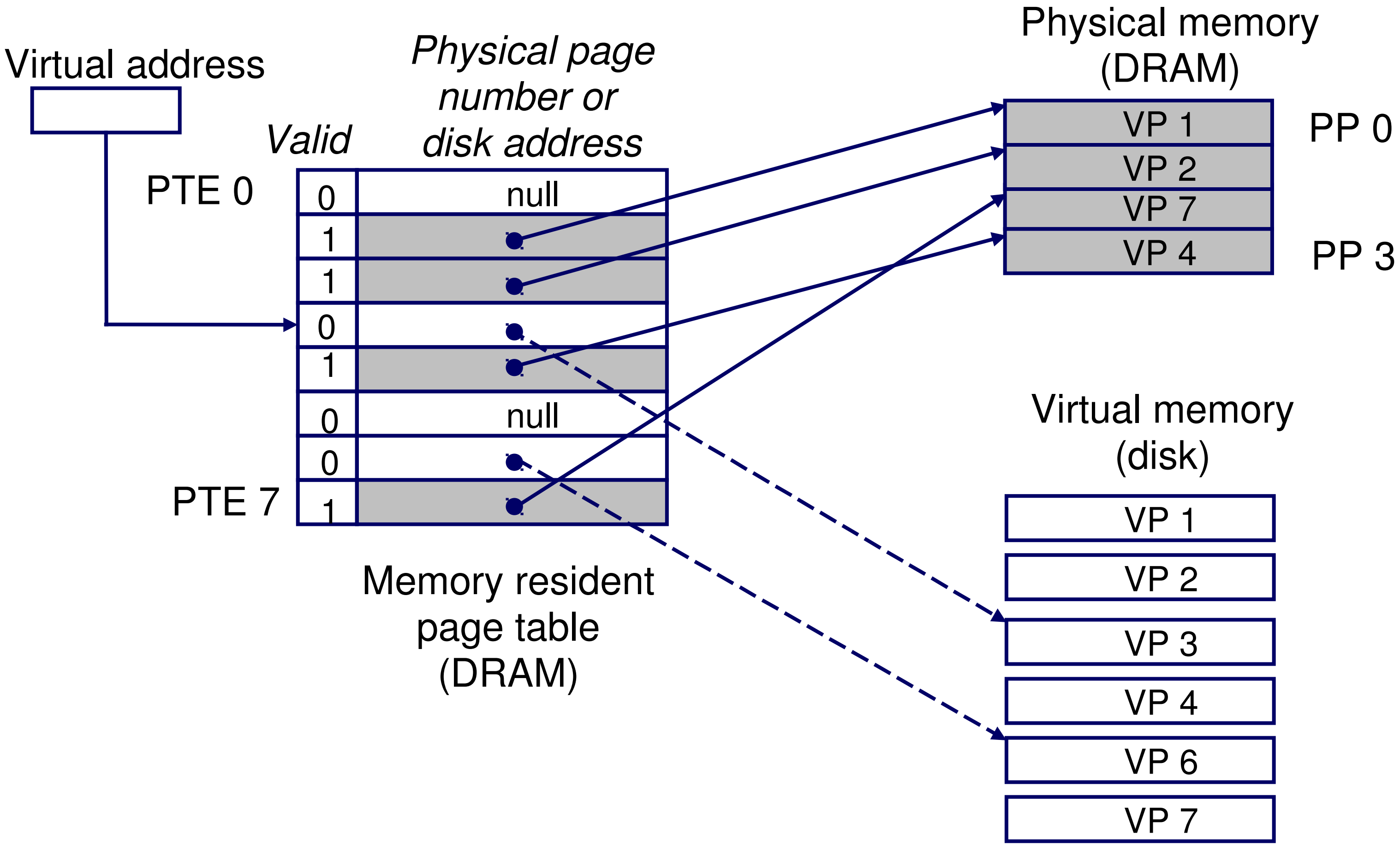
The ARM MMU supports four page sizes. The largest sizes are called sections and the smaller sizes are called pages:

Supersections: 16 MB memory blocks (24-bit offsets)
Sections: 1 MB memory blocks (20-bit offsets)
Large pages: 64 KB pages (16-bit offsets)
Small pages: 4 KB pages (12-bit offsets)

Page Faults

page table entry(PTE) tells the memory management unit whether a specific page is mapped into physical memory ("valid") or is not ("invalid"). An attempt to access an invalid page will cause the MMU to generate a ***page fault.***

- Example: An instruction references a word contained in VP 3, a miss that triggers a page fault exception



The low bits of the address form the offset within the page (or page frame).

For example, with a 64-bit address, if a page size is 1 MB,then the lowest 20 bits (which address 1 MB) form the offset and the top 44 bits form the page number.

Page translation

- Address bits = page number + page offset

- Translate virtual page number (vpn) to physical page (frame) number (ppn/pfn) using page table.

Demand paging

Demand paging is an approach to memory management where we load pages into memory only as they are needed by the process.

For example, when a process starts executing and tries to load its first instruction, the operating system will get a page fault because the required page has not been loaded and mapped onto a page frame.

Demand paging

Demand paging is an approach to memory management where we load pages into memory only as they are needed by the process.

For example, when a process starts executing and tries to load its first instruction, the operating system will get a page fault because the required page has not been loaded and mapped onto a page frame.

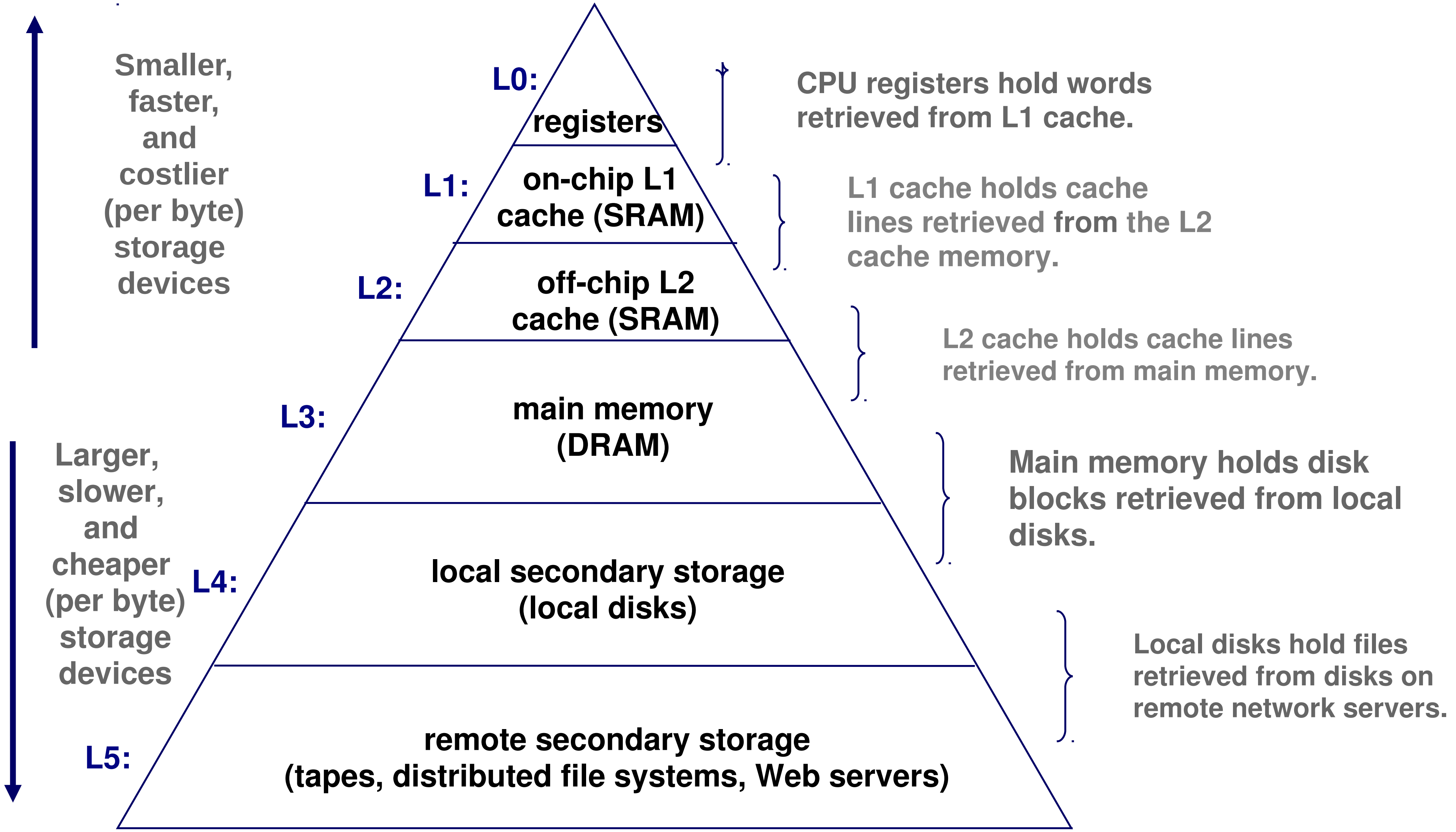
Bring a page into memory only when it is needed"

- Less I/O needed"
- Less memory needed "
- Faster response"
- More users"

Page is needed ⇒ reference to it"

- invalid reference ⇒ abort"
- not-in-memory ⇒ bring to memory

Memory Hierarchy



Smaller, faster, and costlier (per byte) storage devices

Larger, slower, and cheaper (per byte) storage devices

L0: registers

L1: on-chip L1 cache (SRAM)

L2: off-chip L2 cache (SRAM)

L3: main memory (DRAM)

L4: local secondary storage (local disks)

L5: remote secondary storage (tapes, distributed file systems, Web servers)

CPU registers hold words retrieved from L1 cache.

L1 cache holds cache lines retrieved from the L2 cache memory.

L2 cache holds cache lines retrieved from main memory.

Main memory holds disk blocks retrieved from local disks.

Local disks hold files retrieved from disks on remote network servers.

`From lecture-9.ppt`

DRAM/SRAM is “volatile”: contents lost if power lost

Disks are “non-volatile”: contents survive power outages

DRAM makes use of single transistor and capacitor for each memory cell, whereas each memory cell of SRAM makes use of an array of 6 transistors.

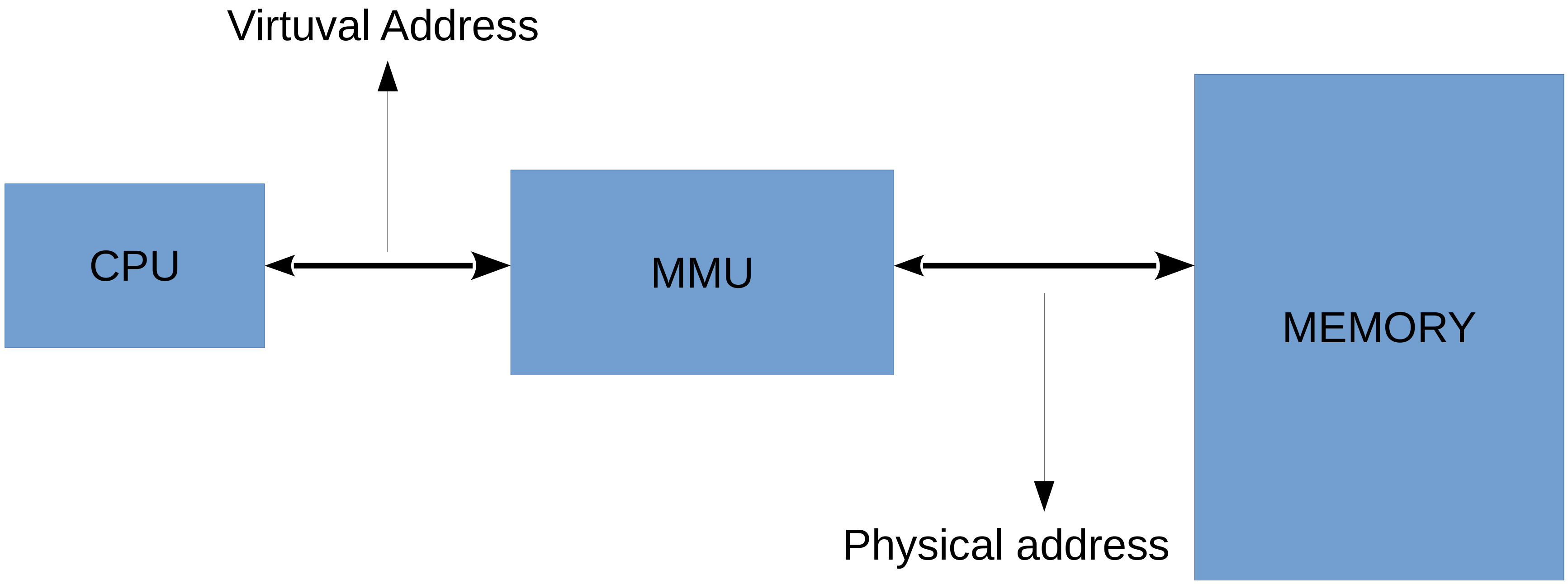
DRAM needs refreshing, whereas SRAM does not require refreshing of the memory cell.

Both retain data till the time they are supplied with power.

SRAMs are used in Caches because of higher speed.

DRAMs are used for main memory in a PC because of higher densities.

Memory Management Unit (MMU)



A memory management unit that supports paging causes every logical address (virtual address) to be translated to a physical address (real address) by translating the logical page number of the address to a physical page frame number.

Caches

- Caches

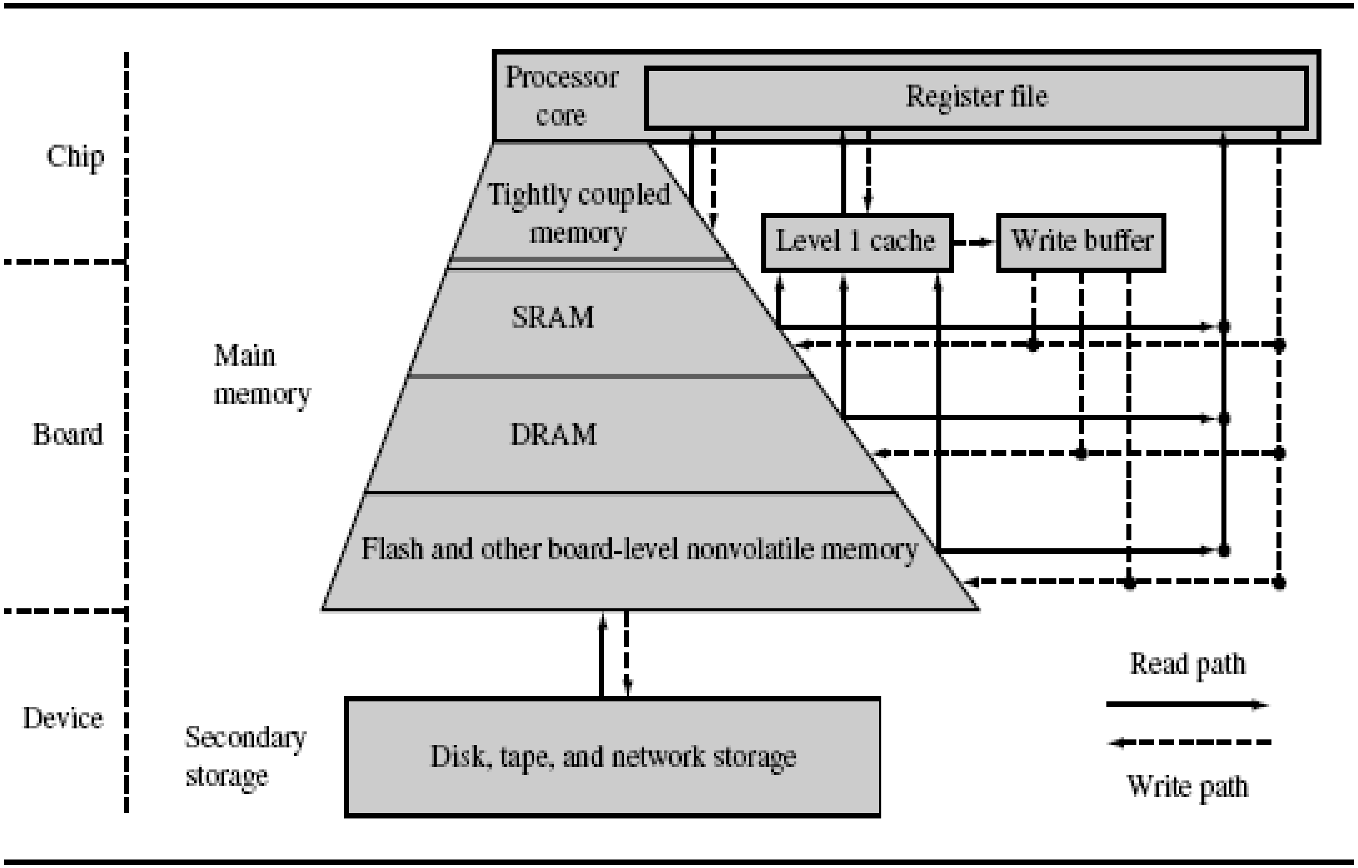
  - Non-cached system

  - cached system

  - write buffer

Caches

- Cache is the array of high speed, very small memory sitting between processor(ARM) core and the main memory



Why Caches?

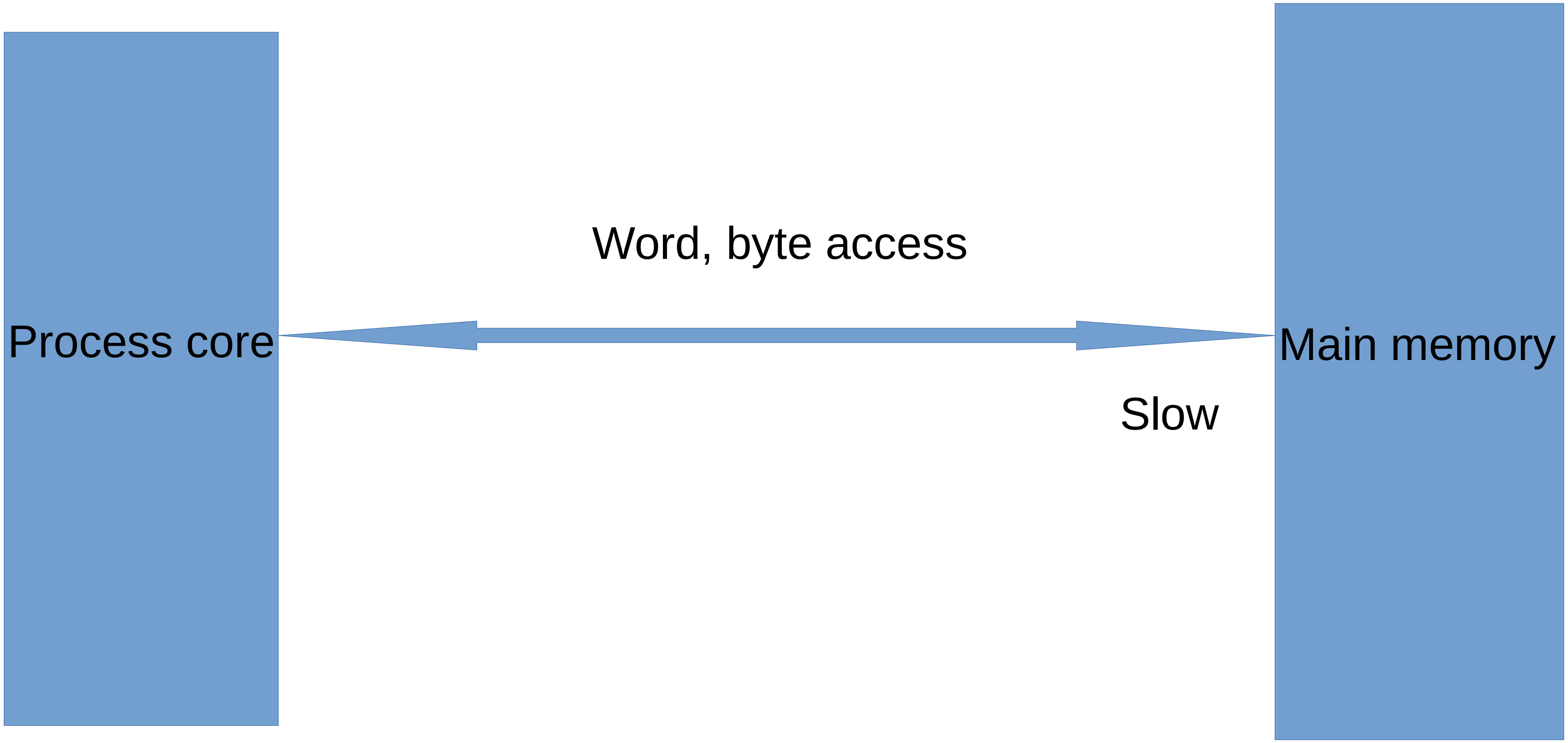
- To increase system performance

- To hide the slower main memory.

- To handle large amounts of code and data at a faster and more efficient way.

Non-Cached system

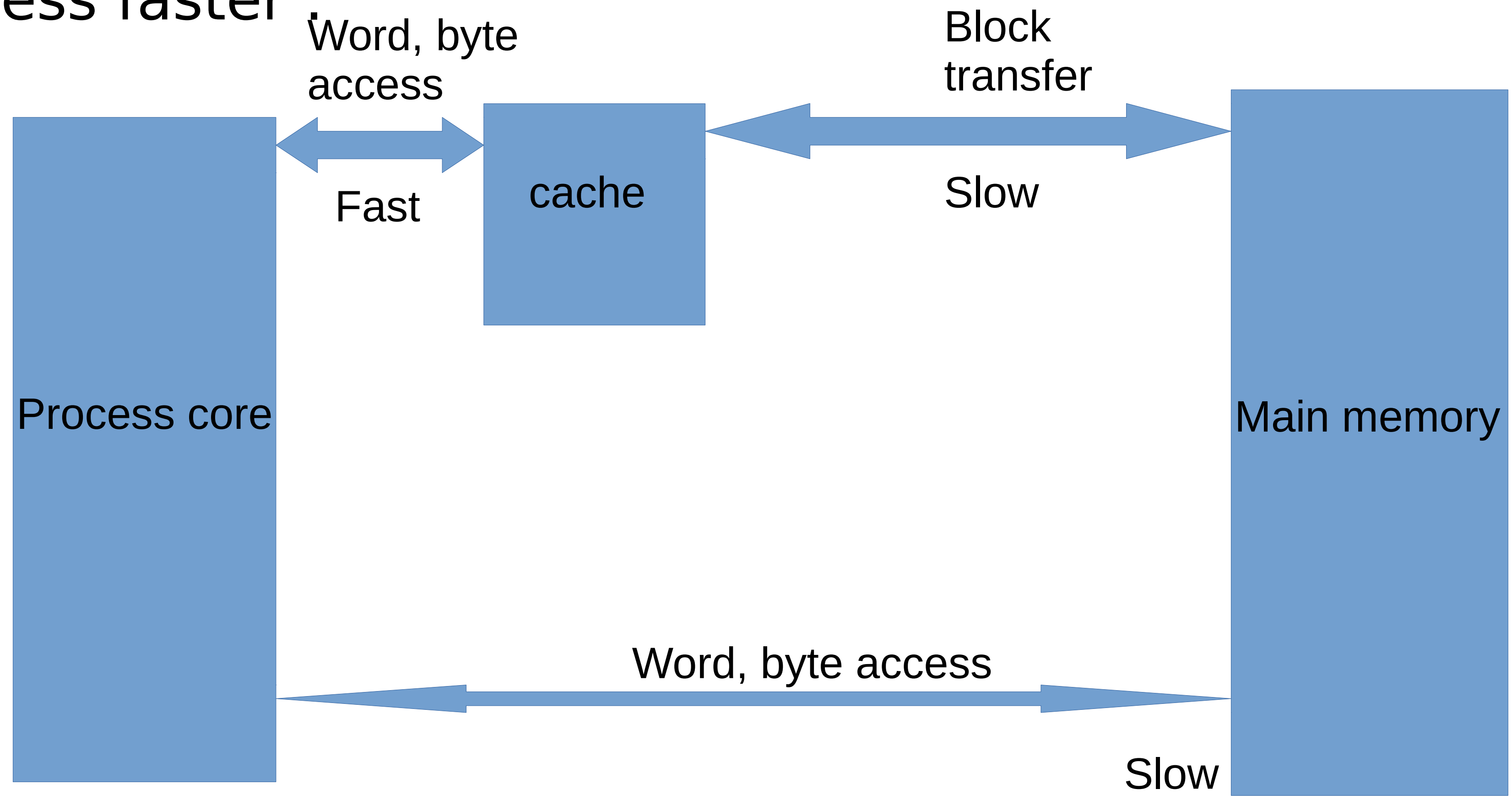
- Main memory is accessed directly by the processor core using the datatypes supported by the processor core.



NonCached system

Cached system

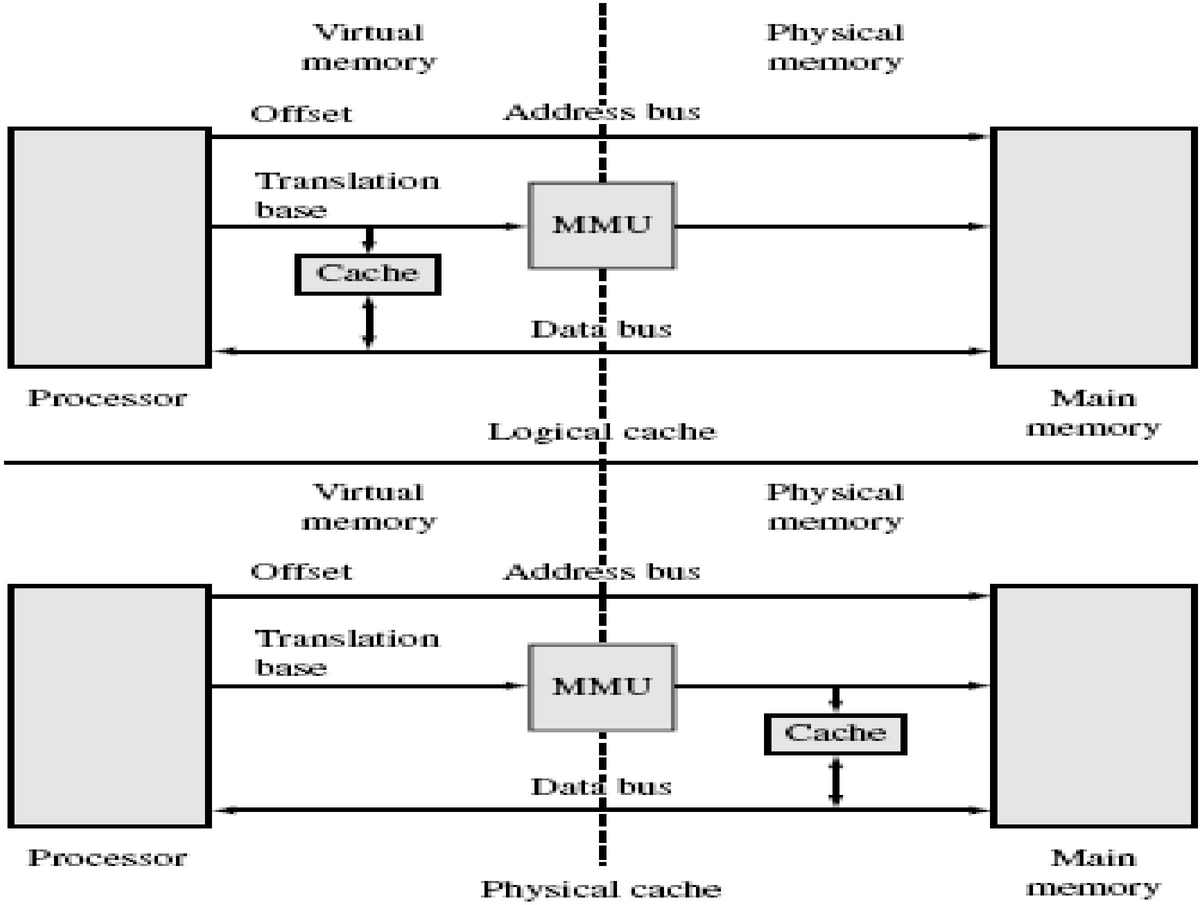
- Small memory placed between processor and main memory to store recently accessed transactions and make memory access faster .



Cached system

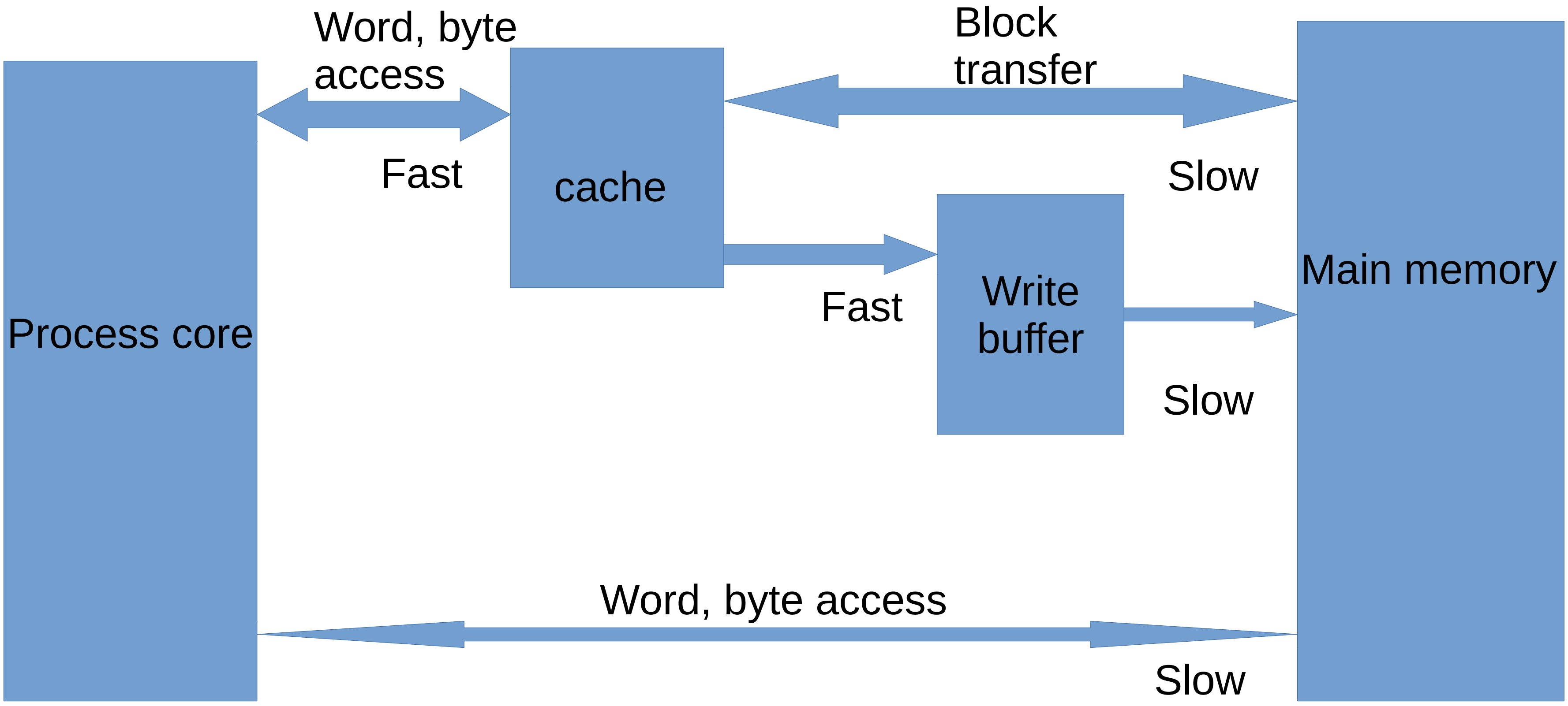
Logical Cache and Physical cache

Logical cache examples - ARM7, ARM9, ARM10
Physical cache examples - ARM11



Write Buffer

- The write buffer acts as a temporary buffer that frees available space in the cache memory.
- The cache transfers a cache line to the write buffer at high speed and then the write buffer drains it to main memory at slow speed.



Cached system

Cache Architecture

- ARM uses two bus architectures in its cached cores, the **Von Neumann** and the **Harvard**.
- In processor cores using the Von Neumann architecture, there is a ***single cache*** *used for* *instruction and data*. This type of cache is known as a **unified cache**. A unified cache memory contains both instruction and data values.
- The Harvard architecture has separate *instruction and data* buses to improve overall system performance, but supporting the *two buses requires two caches* - instruction cache (I-cache) and data cache (D-cache). This type of cache is known as a **split cache.**

Cache operation

- Many main memory locations are mapped onto one cache entry.
- May have caches for:
  - instructions;
  - data;
  - data + instructions (unified).
- Memory access time is no longer deterministic.

- Instruction cache to speed up executable instruction fetch.

- Data cache to speed up data fetch and store.

- TLB(Translation Look-aside Buffer) used to speed up virtual-to-physical address translation for both executable instructions and data.

Cache Elements

Cache memory

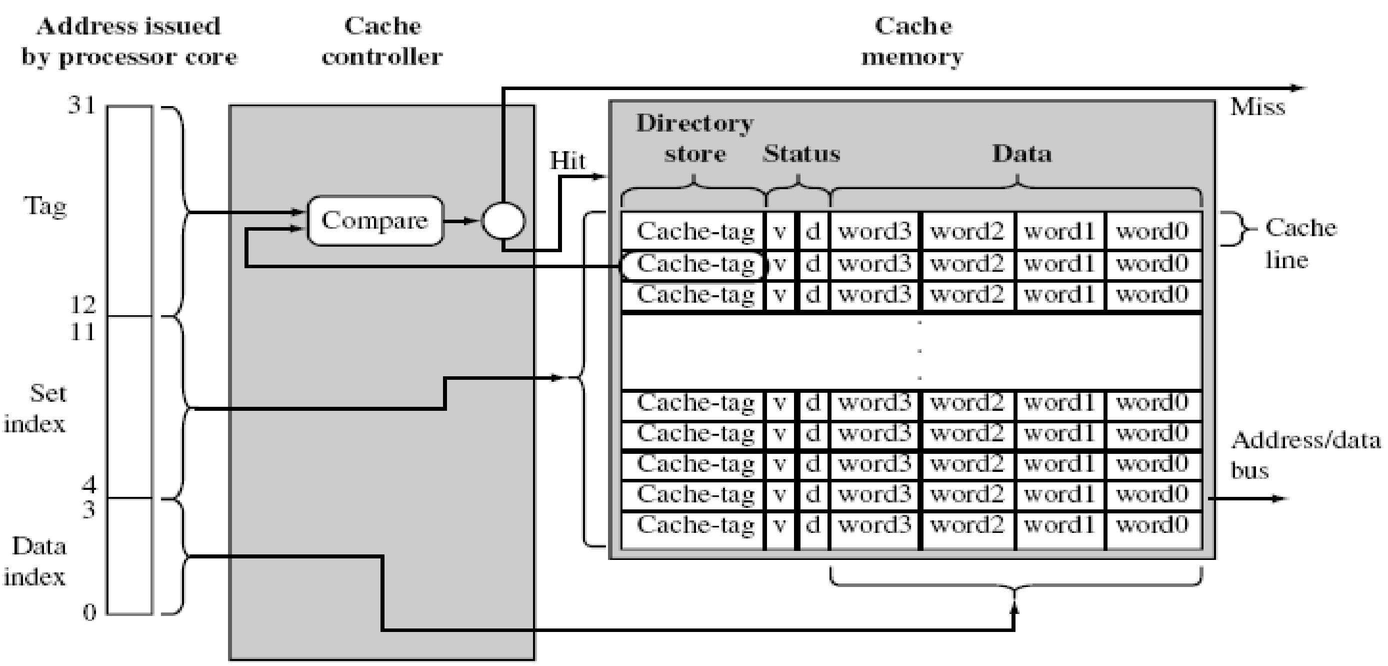
- Cache memory holds the data frequently requested by CPU

Cache controller

- The cache controller is hardware that copies code or data from main memory to cache memory *automatically*.

About Cache Memory

- The cache memory is a dedicated memory array accessed in units called **cache lines**.
- Cache memory has three main parts *Directory store, Status information*, and *Data section*.
- All three parts of the cache memory are present for each cache line.



About Cache Line

- **Cache tag** is a directory store which holds the address, identifying where the cache line was copied from main memory. In this way it keeps track of information from main memory being stored in cache line of cache memory.
- **Data section** holds the data read from main memory.
- **Status bits** maintain state information. Two common status bits are the valid bit and dirty bit.
- A **valid bit** marks a cache line as active, meaning it contains live data originally taken from main memory and is currently available to the processor core on demand.
- A **dirty bit** defines whether or not a cache line contains data that is different from the value it represents in main memory.

Cache hit and cache miss

The cache controller divides the address of the request from processor core into three fields i.e . the ***tag field***, the ***set index field***, and the ***data index field***.

The controller then checks the *valid bit* to determine if the cache line is active, and compares the cache-tag to the tag field of the requested address.

If both the status comparison succeed, it is a ***cache hit***. If either the status comparison fails, it is a ***cache miss***.

The instruction and data cache

IDC operation:-

- The ARM720T contains an 8KB mixed Instruction and Data Cache(IDC).
- The cache comprises four segments of 64 lines each, each line containing eight words.
- The IDC is always reloaded a line at a time. The IDC is enabled or disabled using the ARM720T Control Register and is disabled on HRESETn.

  Note:- The MMU must never be disabled when the cache is on.
- However, you can enable the two devices simultaneously with a single write to the "*Control Register*".

Control Register

31	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNP/SBZ		V	UNP /SBZ			R	S	B	L	D	P	W	C	A	M

The C bit determines if data being read can be placed in the IDC and used for subsequent read operations. Typically, main memory is marked as cachable to improve system performance,and I/O space is marked as noncachable to stop the data being stored in the ARM720T cache.

Cachable reads (C=1)

A line fetch of eight words is performed when a cache miss occurs in a cachable area of memory, and it is randomly placed in a cache bank.

Note:- Memory aborts are not supported on cache line fetches and are ignored.

Uncachable reads (C=0)

An external memory access is performed and the cache is not written.

31	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNP/SBZ		V	UNP /SBZ			R	S	B	L	D	P	W	C	A	M

M Bit

MMU enable/disable:

0 = MMU disabled
1 = MMU enabled

31	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNP/SBZ		V	UNP /SBZ			R	S	B	L	D	P	W	C	A	M

C Bit

Cache enable/disable:

0 = Instruction and/or Data Cache (IDC) disabled

1 = Instruction and/or Data Cache (IDC) enabled

31	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNP/SBZ		V	UNP /SBZ			R	S	B	L	D	P	W	C	A	M

W Bit
Write buffer enable/disable:

0 = Write Buffer disabled
1 = Write Buffer enabled

DRAM/SRAM is “volatile”: contents lost if power lost

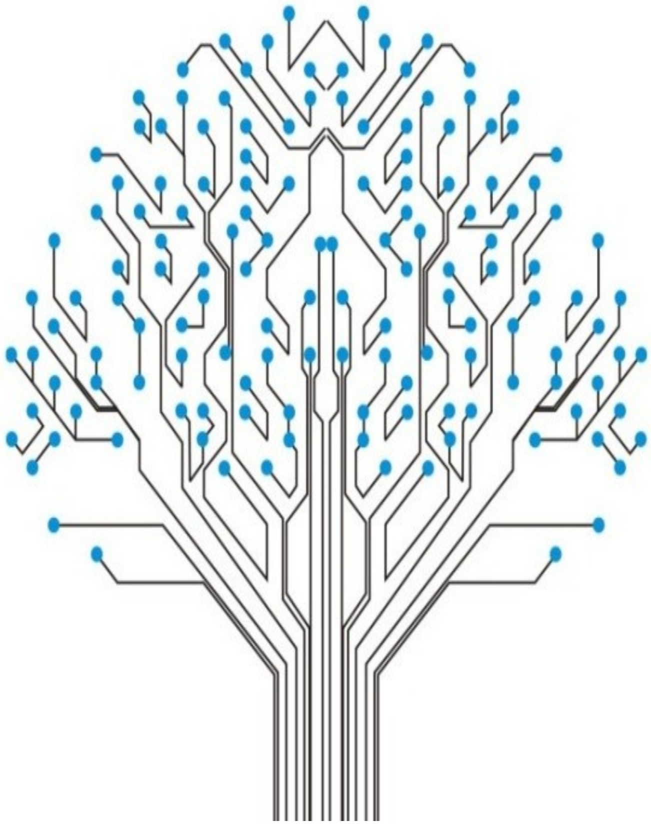
Disks are “non-volatile”: contents survive power outages

Any Questions ?

GLOBAL EDGE

Intelligence Of Things

Thank you



Fairness

Learning

Responsibility

Innovation

Respect