ITMO

Computer Systems Design

Lesson 13

Memory subsystem.

Basic optimizations

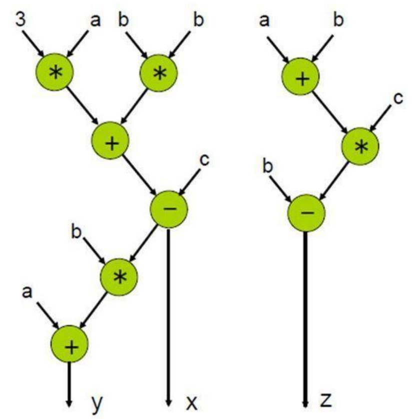
Alexander Antonov, Assoc. Prof., ITMO University

Hangzhou, 2025

Outline the lesson

- Communications and memory
- Memory idealisms and hierarchy
- Integration of computations and memory
- Basic strategies of memory performance increase
- Multi-bank memory
- Multi-port memory
- Special-purpose memories

Communications and memory



Common objective: ***transmission of data between operations***

Communications: in space

Memory: in time

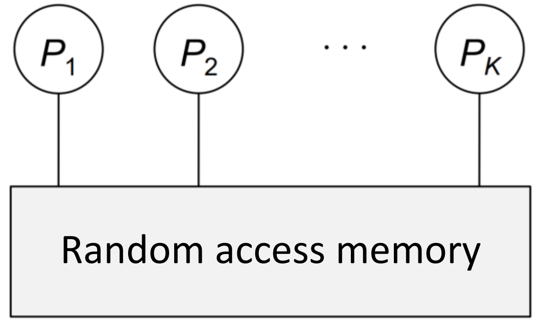
Basic communication models in parallel architectures

- Shared memory

  PRAM *–* ***P****arallel* ***R****andom* ***A****ccess* ***M****achine*

  Natural extension of traditional RAM machine

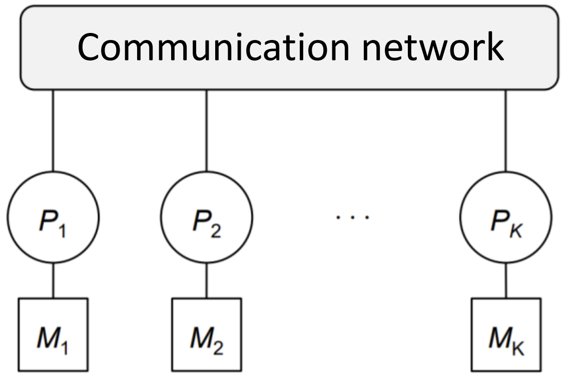
  New data is instantly (and implicitly) accessible for everyone, but memory bus can be a bottleneck



- Message passing

  Memory is distributed, computations are done locally and asynchronously, synchronization is done on demand

  Needs explicit data transfer, but more scalable

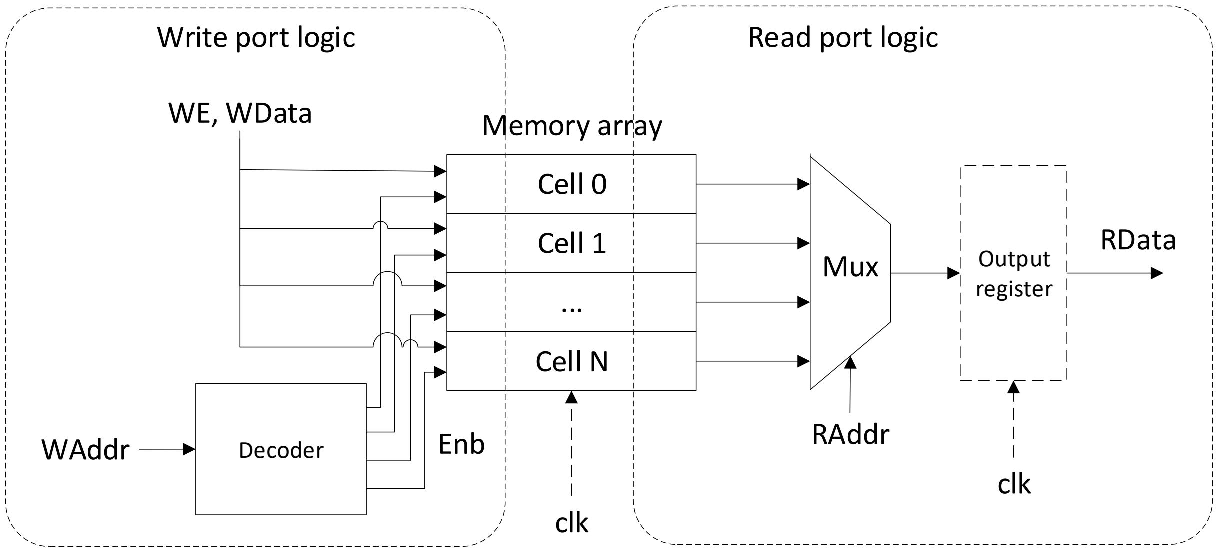


Basic Random Access Memory (RAM, register-based)

Memory array with read/write access to any cell ("random access")

Attributes:

- width (HW increase: ***linear***)
- depth (HW increase: ***O(nlogn)***)
- port configuration: *xRyW* (e.g. 2R1W, 4R2W, etc.)



Memory idealisms

- Infinite capacity.
- Infinite bandwidth.
- Instantaneous (zero) latency.
- Persistence (non volatility).
- Zero or (very low) implementation cost.

All idealisms cannot be approached simultaneously

Memory hierarchy

Memory technologies

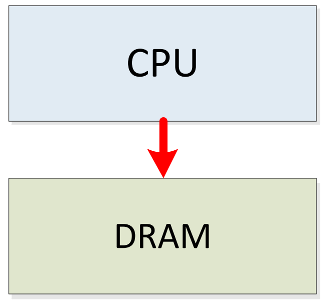
Memory tech can be various types ***with various characteristics***

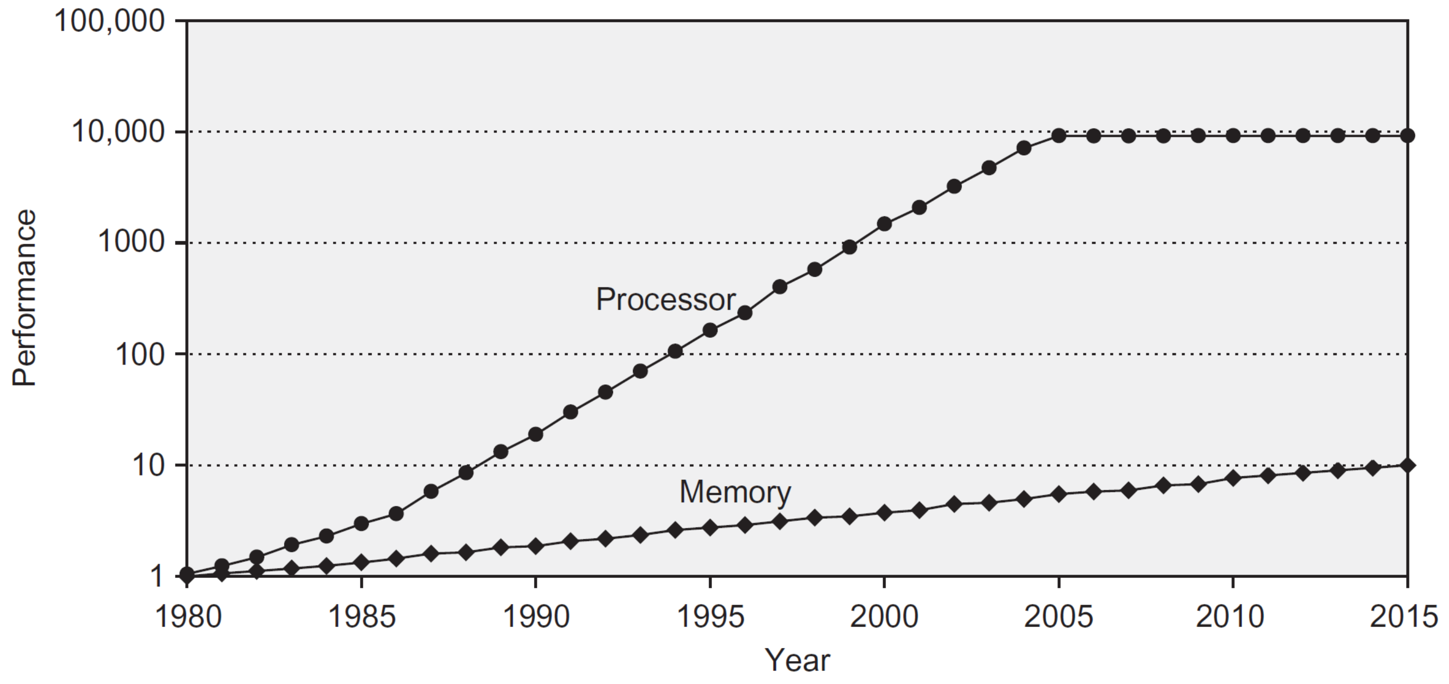
	Volume	Latency	Cost/GiB	Typical usage
Flip-flop Registers	1000 bit	20 ps	Large	Processor
SRAM	10 KiB – 10 MiB	1-10 ns	1000$	Caches
DRAM	10 GiB	80 ns	10$	Main memory
NV Flash	100 GiB	100 us	1$	I/O subsystem
NV Hard Disk	1 TB	10 ms	0.1$	I/O subsystem

Disadvantages of certain memory types should be compensated by advantages of other memory types

Processor/DRAM performance divergence

Traditionally, DRAM chips were used to build main memory





Tighter integration between processing and memory required

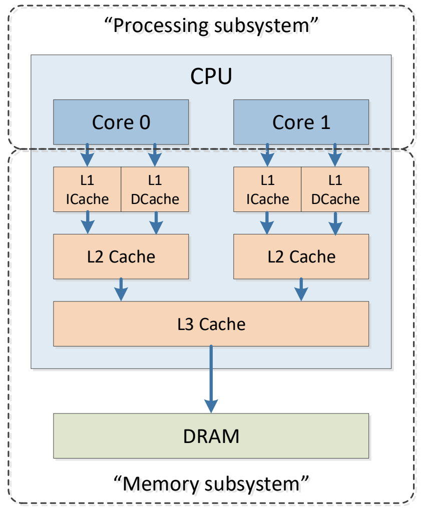
ITMO

Re-integration of computations and memory in vN architectures

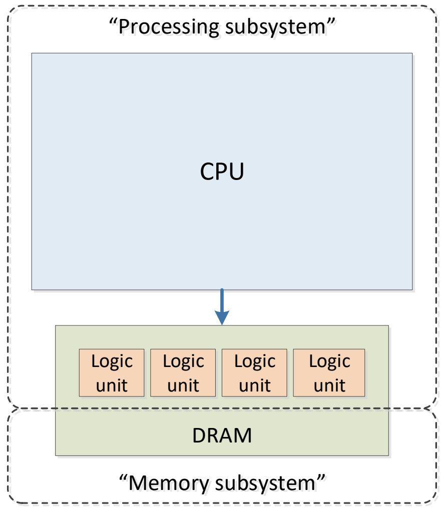
One of major vN architecture problems:

disintegration of computations and memory ("vN bottleneck")

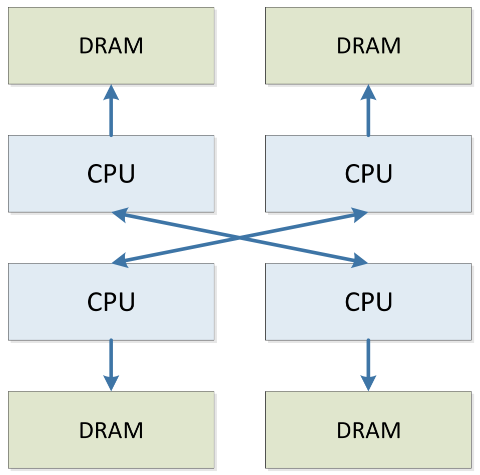
re-integration in different forms is being explored and implemented



Caching "closer" to processor



In-memory processing

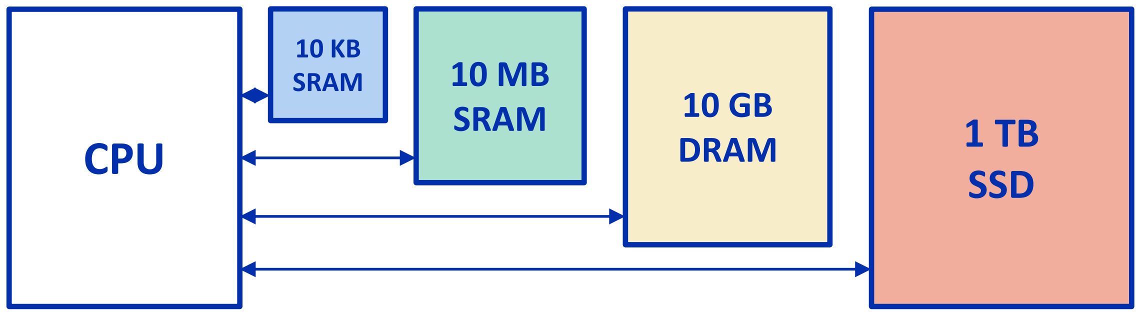


Distribution of computations clustered with memory (e.g. manycores, NUMA)

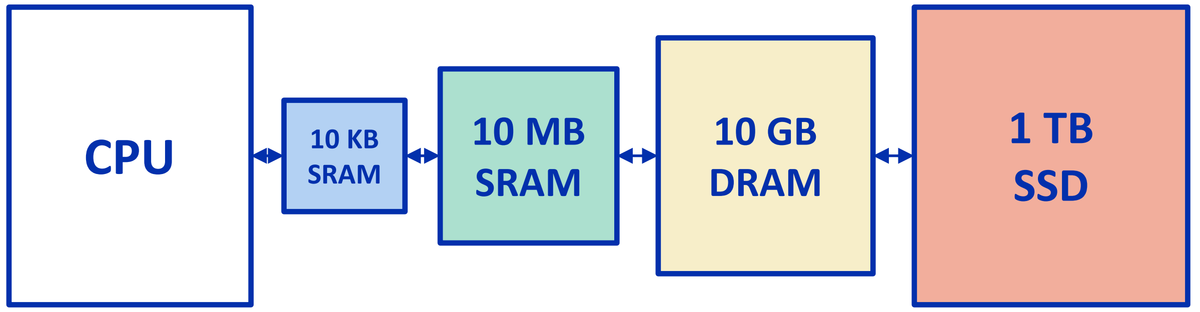
ITMO

Memory hierarchy management: SW vs. HW

Software management (open hierarchy)



Hardware management (closed hierarchy)



Basic strategies of memory performance increase

Strategy	Functional agility	Control logic complexity
Multiple ports	No access collisions, all data can be accessed simultaneously in arbitrary combinations	Control logic is replicated for each port, makes signal routing difficult, available tech library might have limitations
Multiple banks	Data residing in the same bank cannot be accessed simultaneously, data in different banks can be accessed simultaneously in arbitrary combinations	Control logic is replicated for each bank
Wide memory	Data residing only withing certain proximity and alignment can be accessed simultaneously	Control logic complexity remains mostly the same

Multibanking: bank mapping

Fixed data/bank mapping

Multiple SRAM banks (or DRAM chips) can be implemented

Bank size increase ***increases latency O(logn))***

Each bank/chip has ***specific (limited) number of ports***

Issue: data/bank mapping?

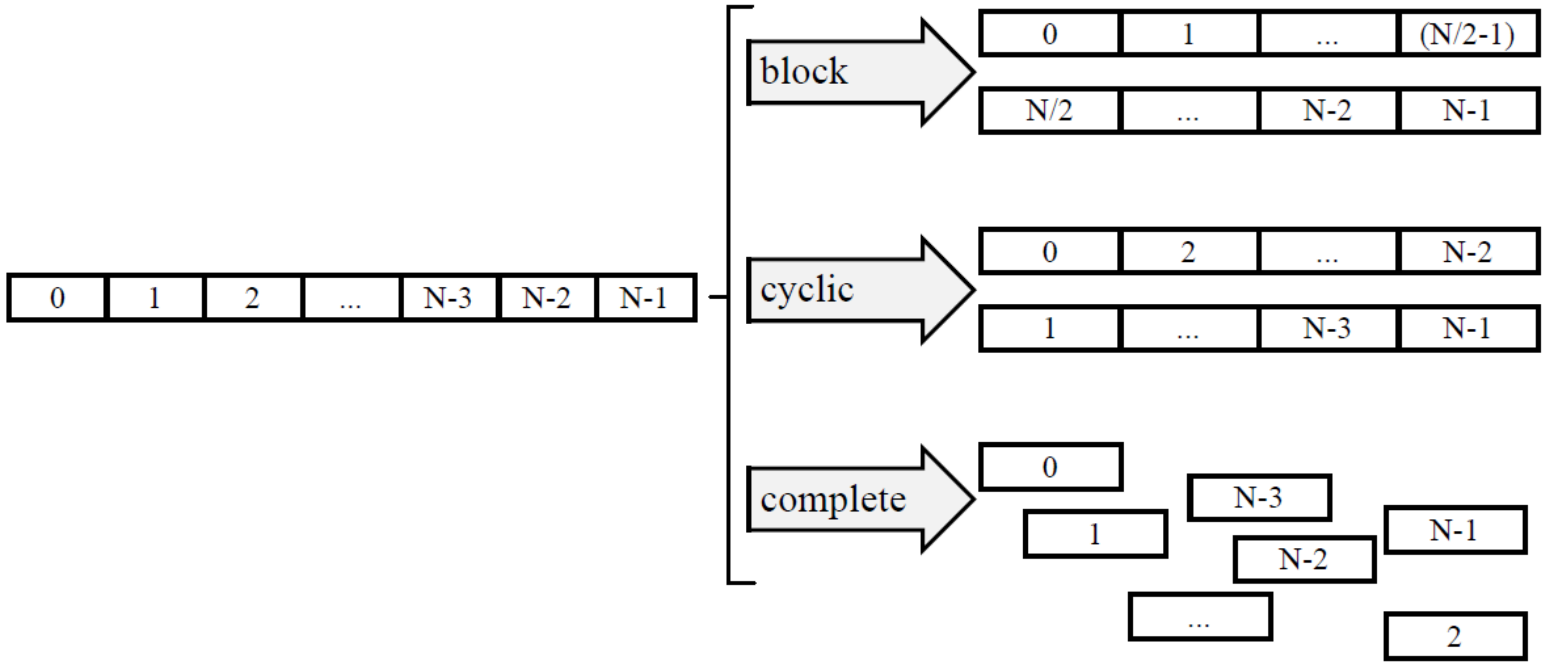
Design considerations:

- Simultaneously unused data can be put in the same bank (in various addresses) and accessed by the same port
- Simultaneously used data should be transmitted by various ports or put in different banks
- If access pattern is fixed, several banks can be assembled in a single wide bank
- RAW dependencies are sensitive to access latency: put "hot" data in smaller (and faster) banks and rare data in bigger banks (with pipelined access for throughput)

Actual for designing custom hardware architectures

Associative caches can do re-mapping dynamically (see Lecture 14)

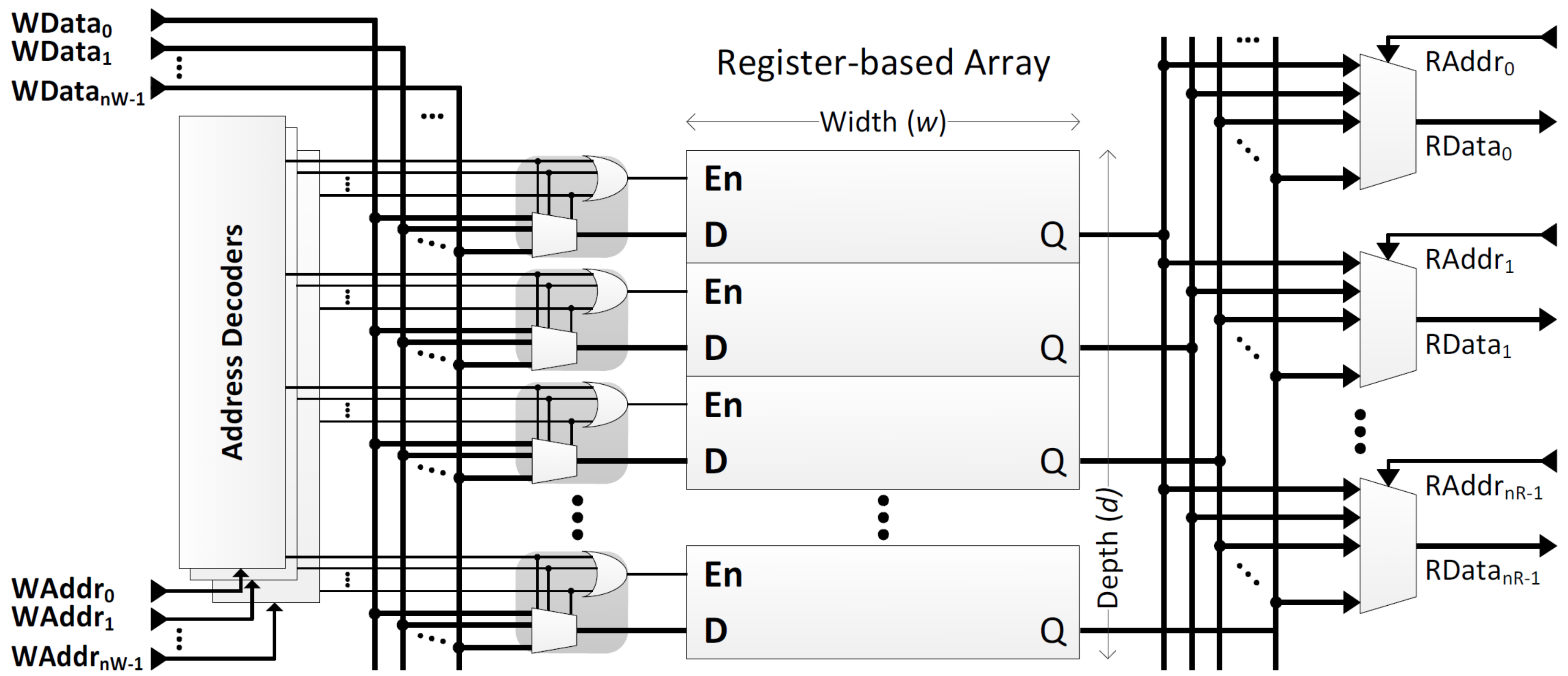
Examples of bank mappings in hardware



UG902. Vivado Design Suite User Guide. High-Level Synthesis. Xilinx

Multi-porting

Multi-ported RAM: register-based array



A. M. Abdelhadi and G. G. Lemieux, "Modular Multi-ported SRAM-based memories," in Proceedings of the 2014 ACM/SIGDA International Symposium on Field-programmable Gate Arrays, 2014.

ITMO

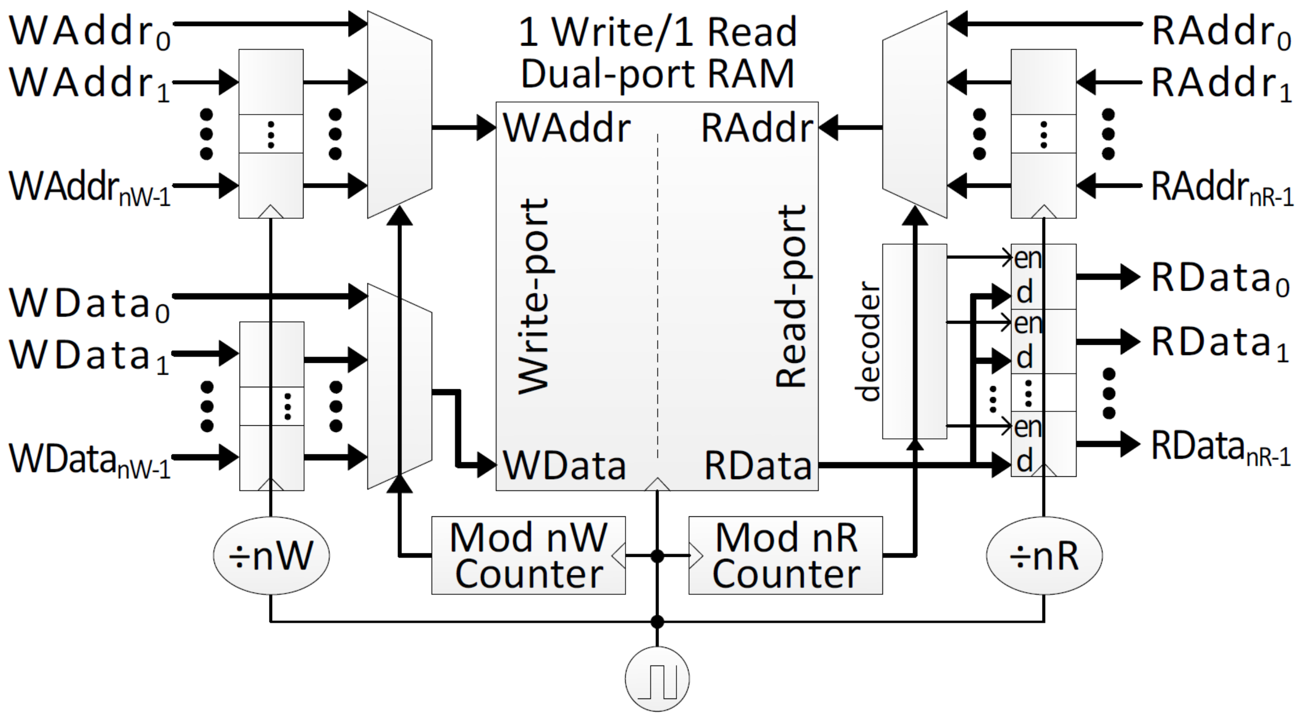
Multiple read/write ports: multi-pumping

RAM is driven by fast clock

Ports are driven by slow clock

Ports are activated in round-robin fashion

Limitation:
RAM frequency

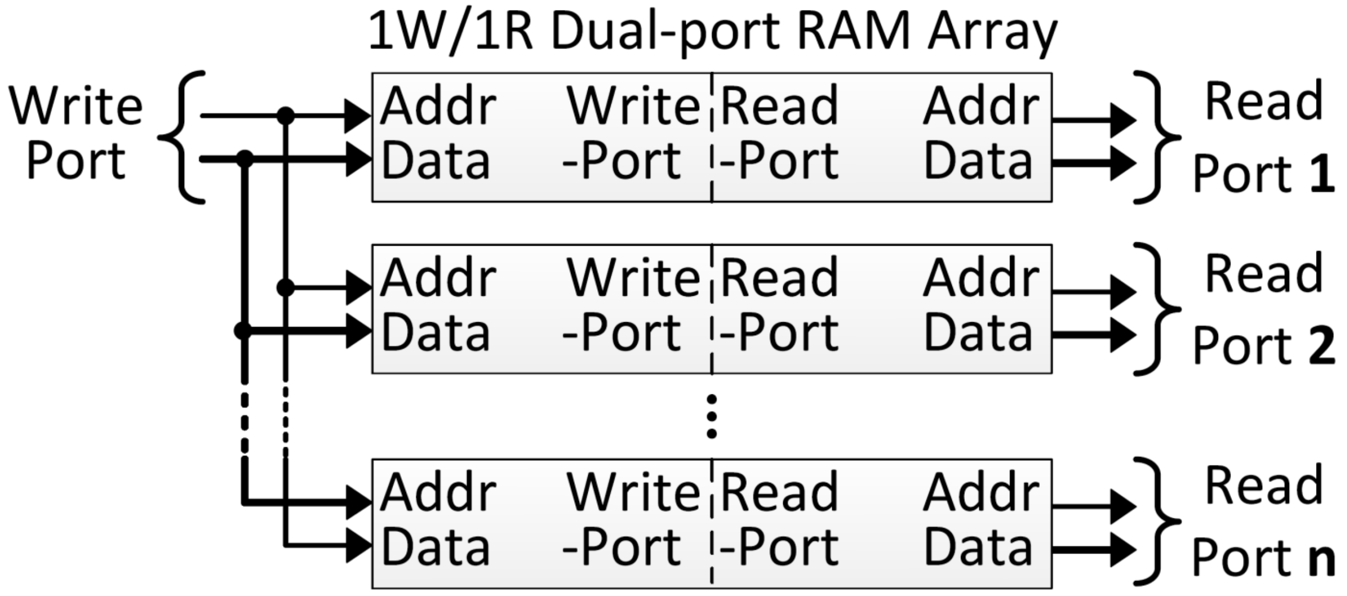


Multiporting via multibanking: read ports

RAM banks contain ***copies of the same data***

Write data (single channel) is ***broadcasted*** to all banks

Data is independently read from various banks



ITMO

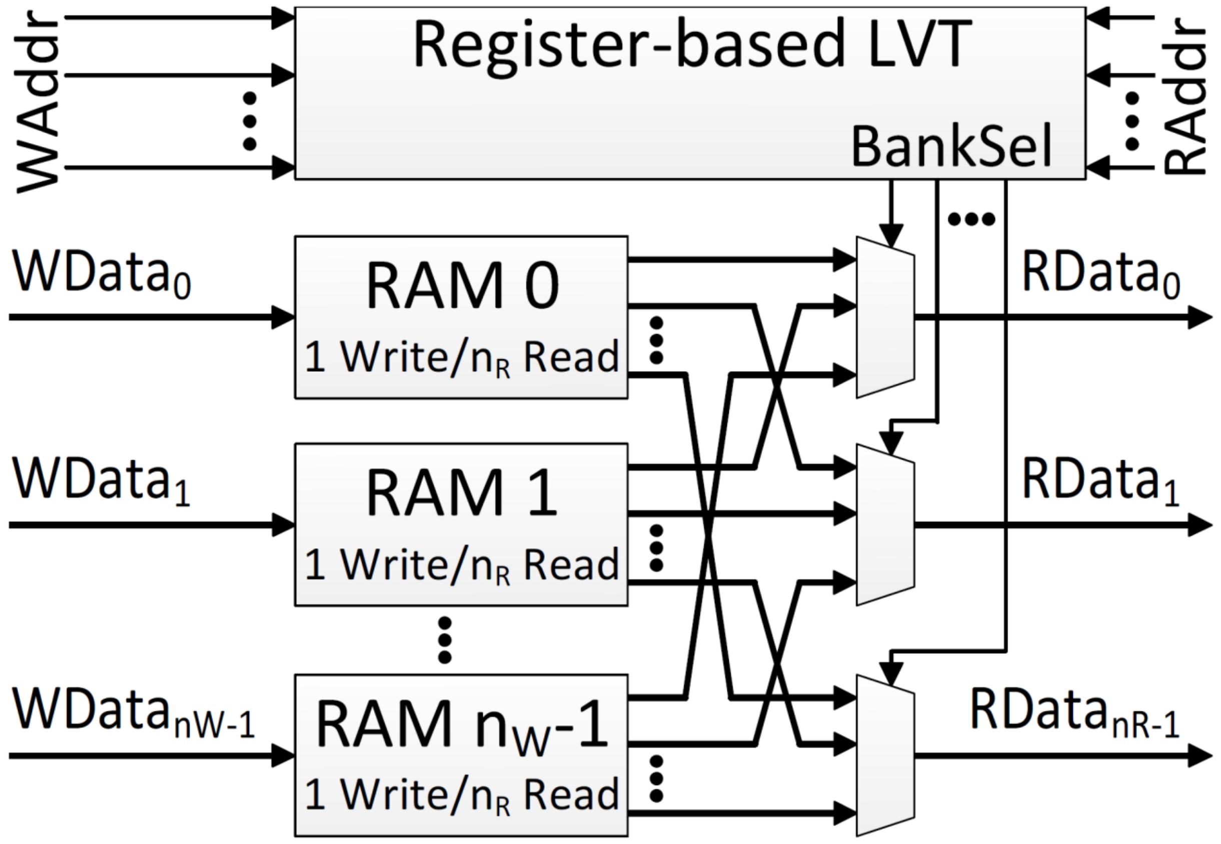
Multiporting via multibanking: write ports

RAM banks contain ***copies of the same data***

actual and outdated

New data is written to a ***single*** bank

LVT (Live Value Table) records actual bank number for every address, ***is a register-based memory***

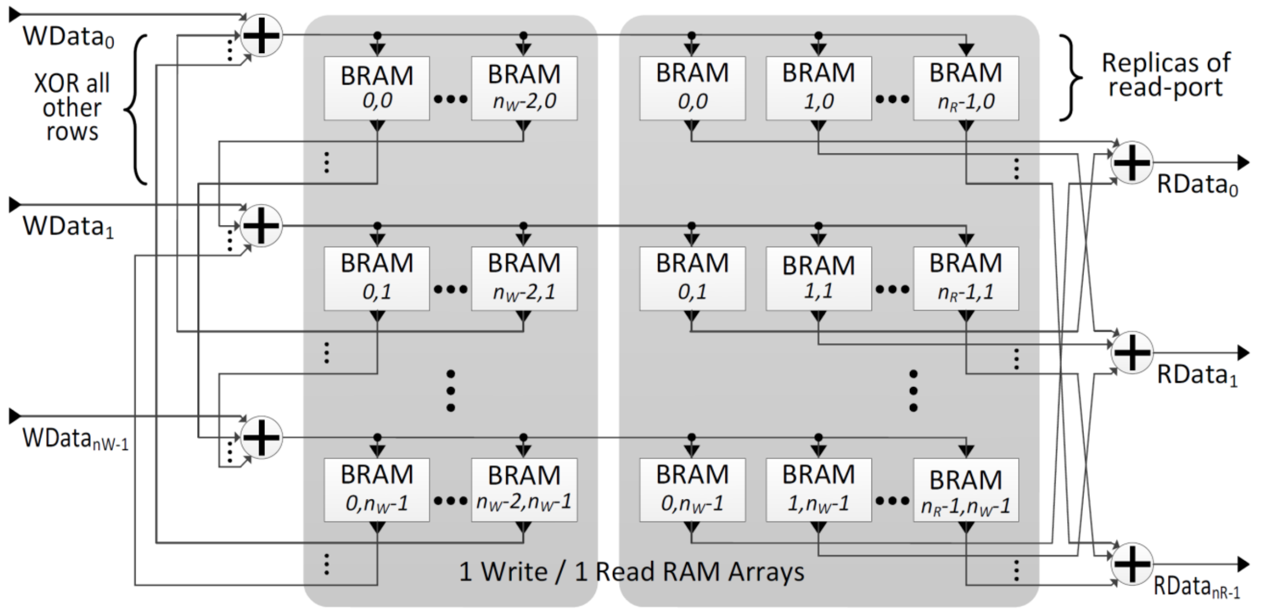


Multiporting via multibanking: XOR-based multiporting

Purpose: multiple write ports ***without register-based LVT***

Principle: ***XOR-ing new data with old*** when writing and reading

Double XOR-ing ***abbreviates***



Write: MEMDATA = OLD ⊕ NEW

Read: RDATA = MEMDATA ⊕ OLD = OLD ⊕ NEW ⊕ OLD = NEW

Multiporting via multibanking: ILVT

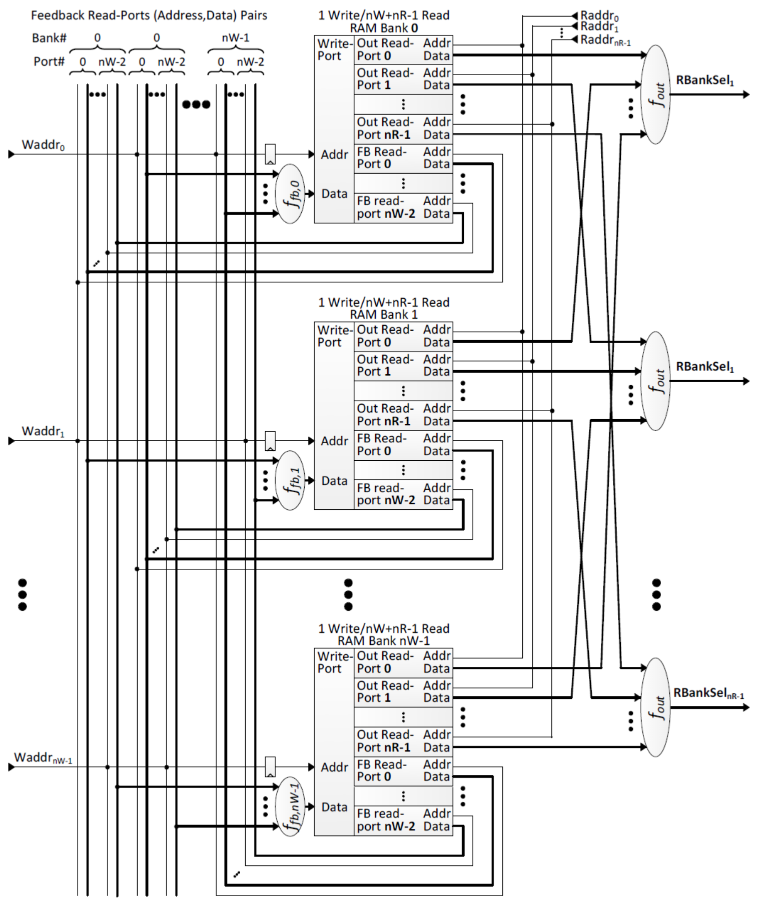
ILVT: invalidation-based live-value-table

Implements multiple 1WnR banks

LVT function is distributed across banks

f_{fb} and f_{out} binary functions proposed:

- newly generated f_{fb} contradicts all other banks
- f_{out} allows to identify uncontradicted bank (and get last actual value)



Considered state-of-the-art (2014)

Special-purpose memories

ITMO

Content-Addressable Memory (CAM)

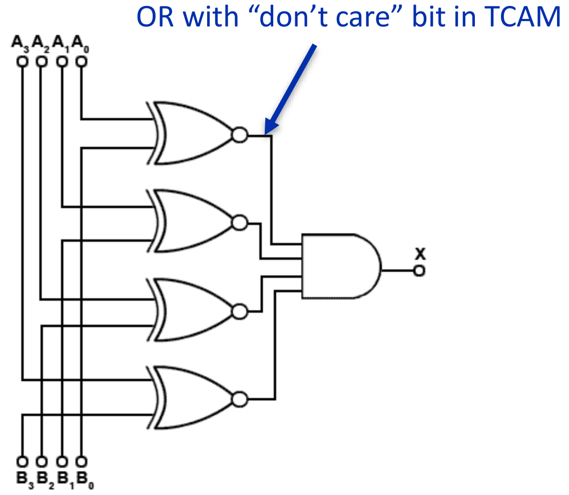
Memory that:

- ***contains key (tag) / value*** pairs
- ***returns*** the ***index*** of key matching the input key (then e.g. used to fetch data)

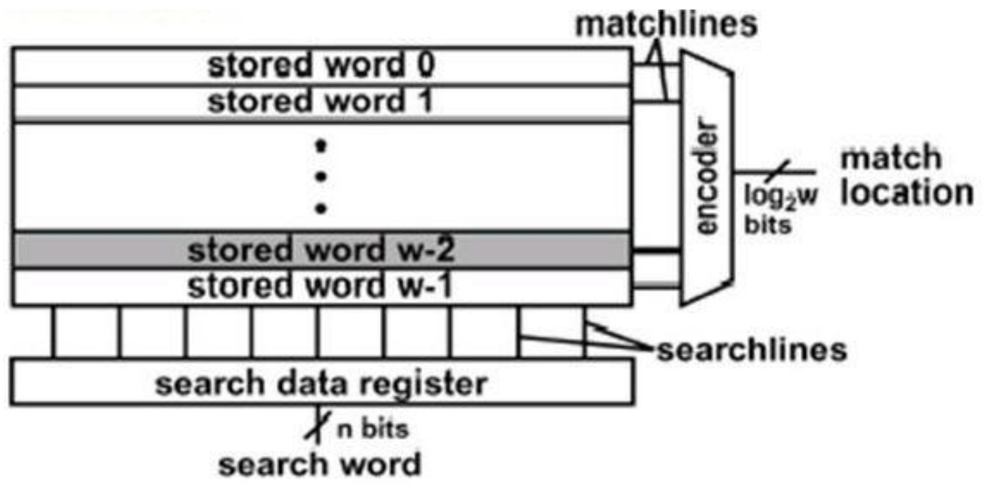
Expensive: ***linear growth of comparators (+ encoder)***

Ternary CAM (TCAM) adds "don't care" mask to input and/or stored keys

Commonly used in caches and search engines



Comparator



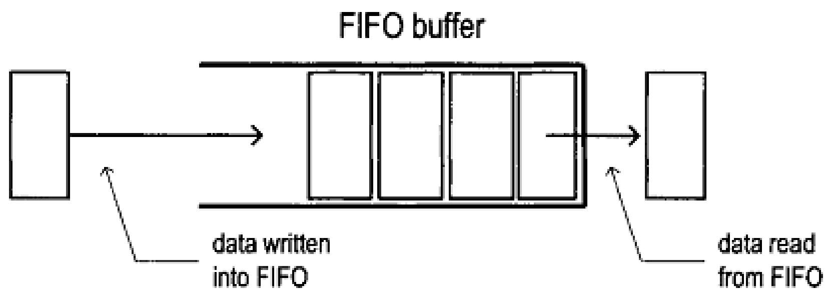
Content-addressable memory

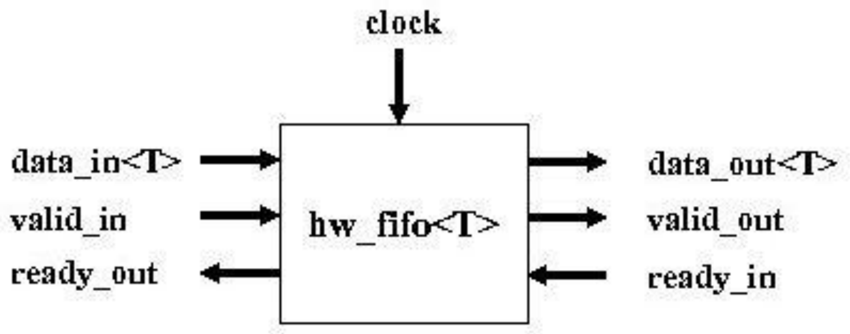
FIFO (First In, First Out)

Elastic buffer smoothening traffic spikes and (optionally) breaking combinational flow control paths (ready/valid signals)

Attributes:

- width, depth
- single/dual clock
- number of simultaneous pushes/pops
- push/pop schedule
  (e.g. FWFT or not? Can data pass through in a single cycle?)
- internal organization
  (e.g. circular (with data pointers) or with data copying)



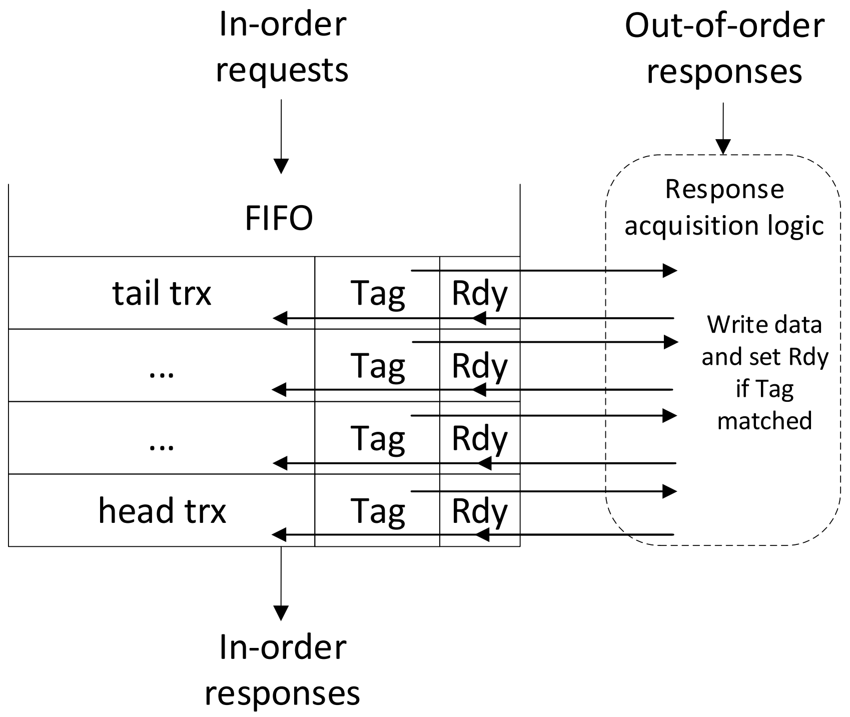


Reorder buffer (ROB)

Purpose: ***restoration of ordering*** for out-of-order responses

Is a FIFO buffer that:

- contains pending transactions ***in return order***
- when out-of-order response arrives: ***finds*** pending transaction by Tag, ***writes*** data, and ***marks*** as ready
- ***sends*** ready transaction(s) from its head (in return order)



Commonly implemented in processors and interconnects

ITMO

Thank you for the lesson!

Alexander Antonov, Assoc. Prof., antonov@itmo.ru

Hangzhou, 2025