

**UCAS-6 > Stanford > Imperial > Verify 2011**

---

# **Marching Memory**

**マーチングメモリ**

**Tadao Nakamura**

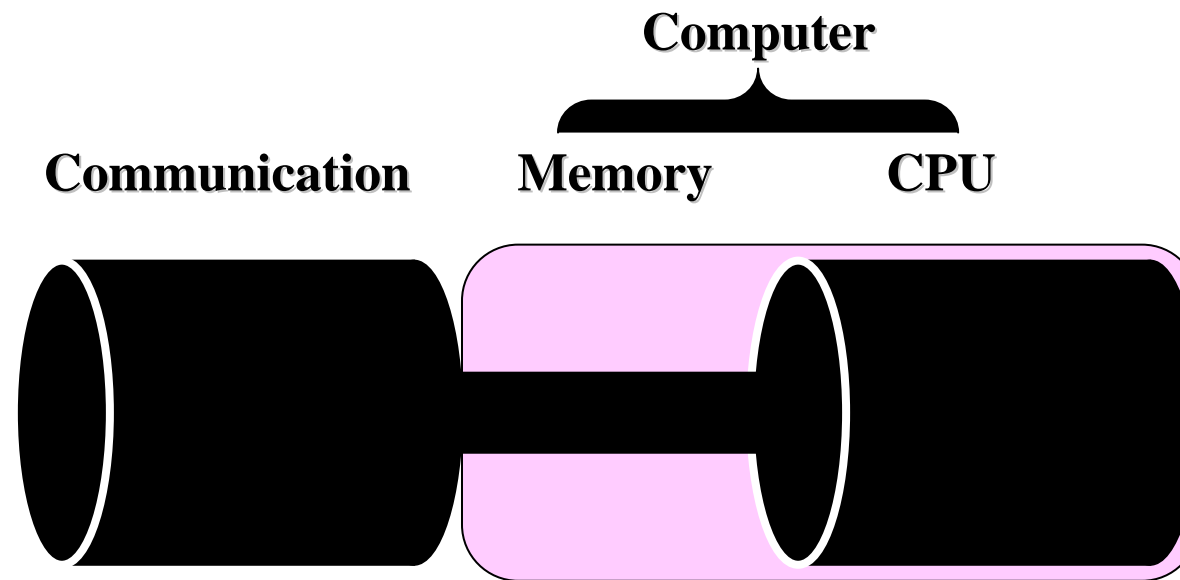
**中村維男**

**Based on Patent Application by**

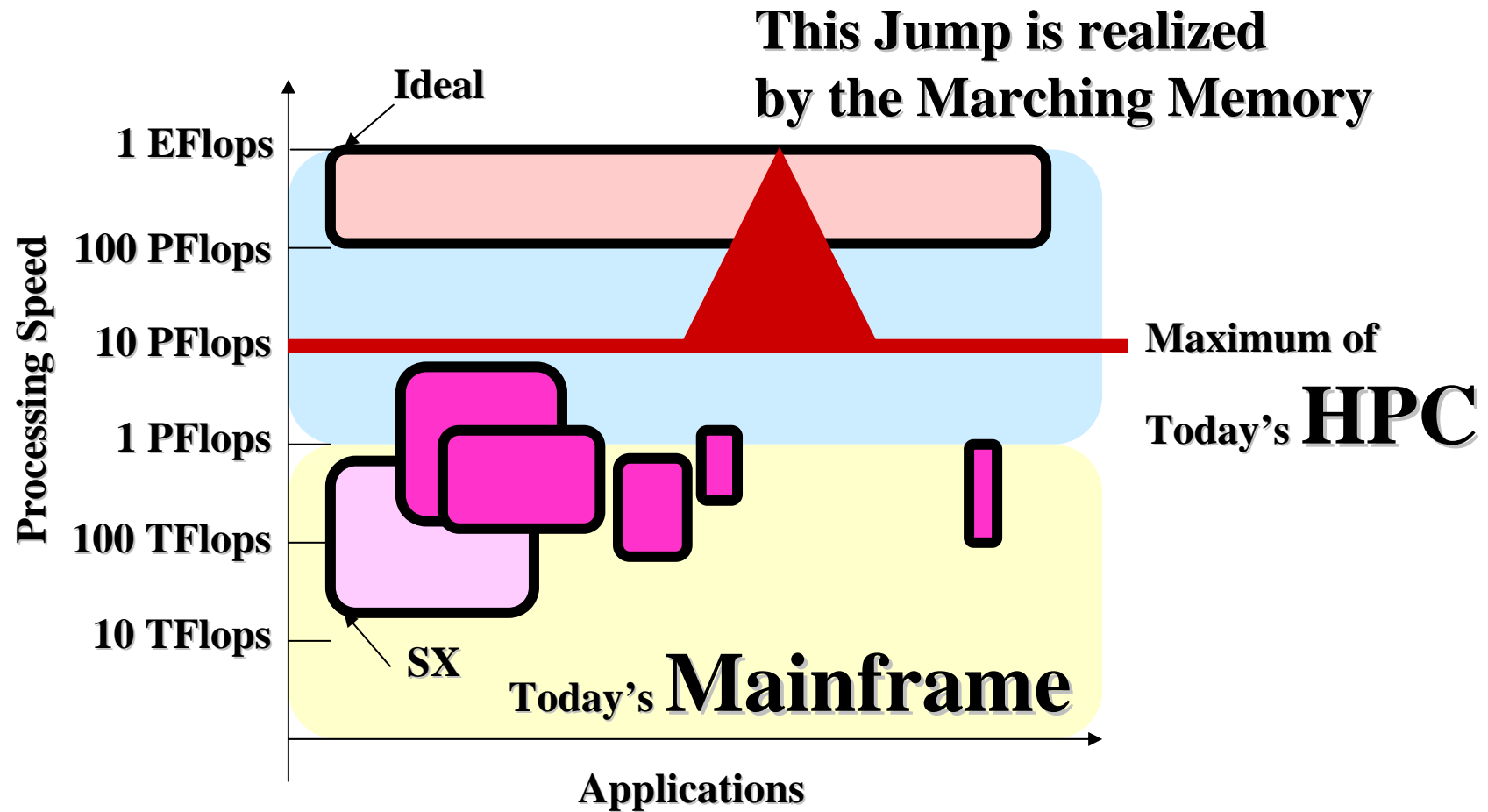
**Tadao Nakamura and Michael J. Flynn**

# C-M-C

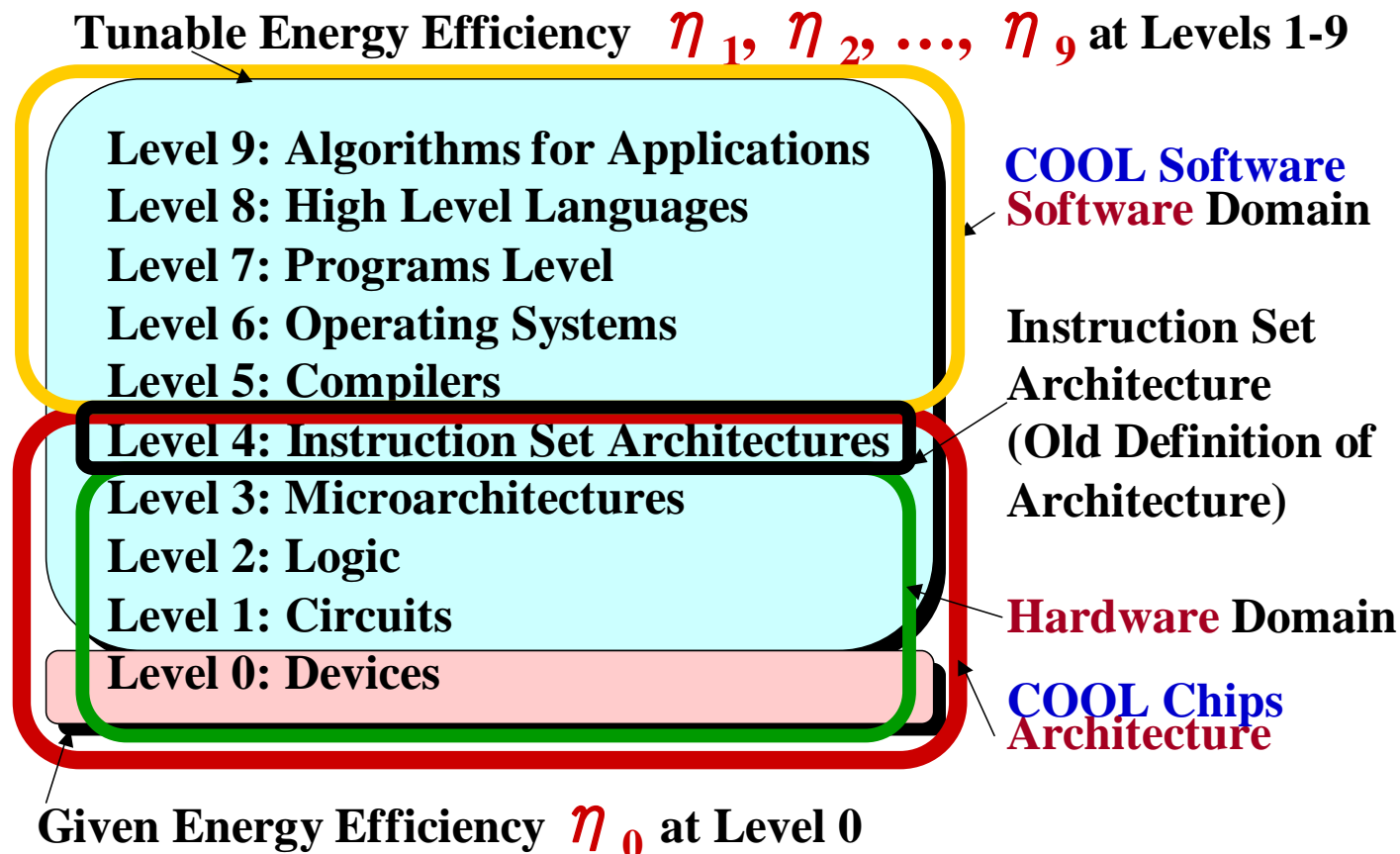
---



# HPC Today and beyond!

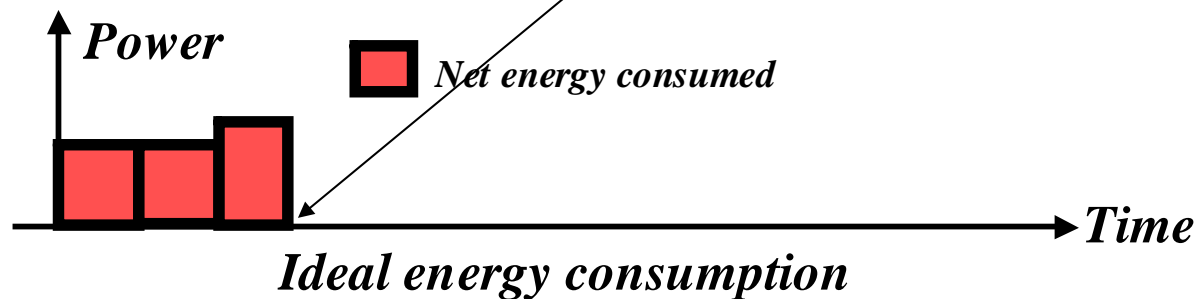
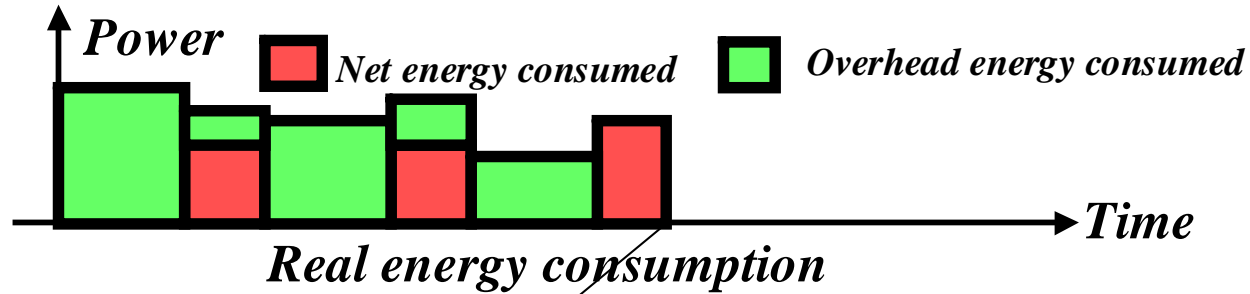
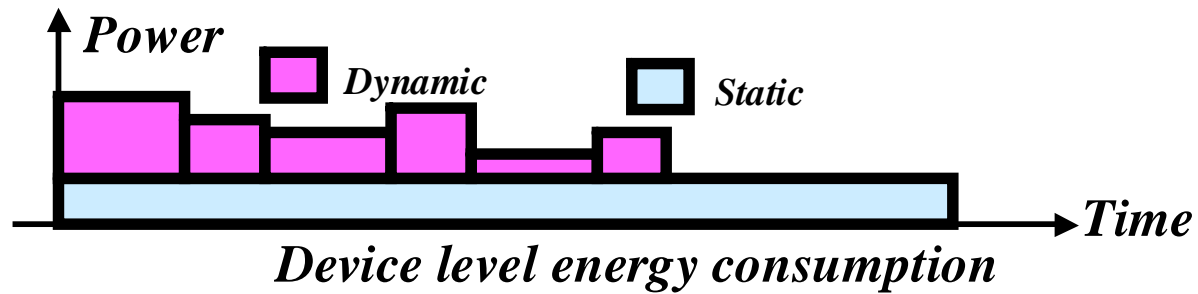


# Computer System



Energy Efficiency at All the Levels for *COOL Systems*

# Energy Consumption



# Difference Among Shift Registers, Delay Line Memory, CCD and Marching Memory

---

*Examples of streaming memory are shift registers, delay line memory and CCDs. However, these differ from marching memory in the content, and scale of streaming memory.*

***Shift registers:*** *Bit manipulation in a least unit, word of memory, performing multiplication and division in the binary system and serial –parallel and parallel-serial conversions*

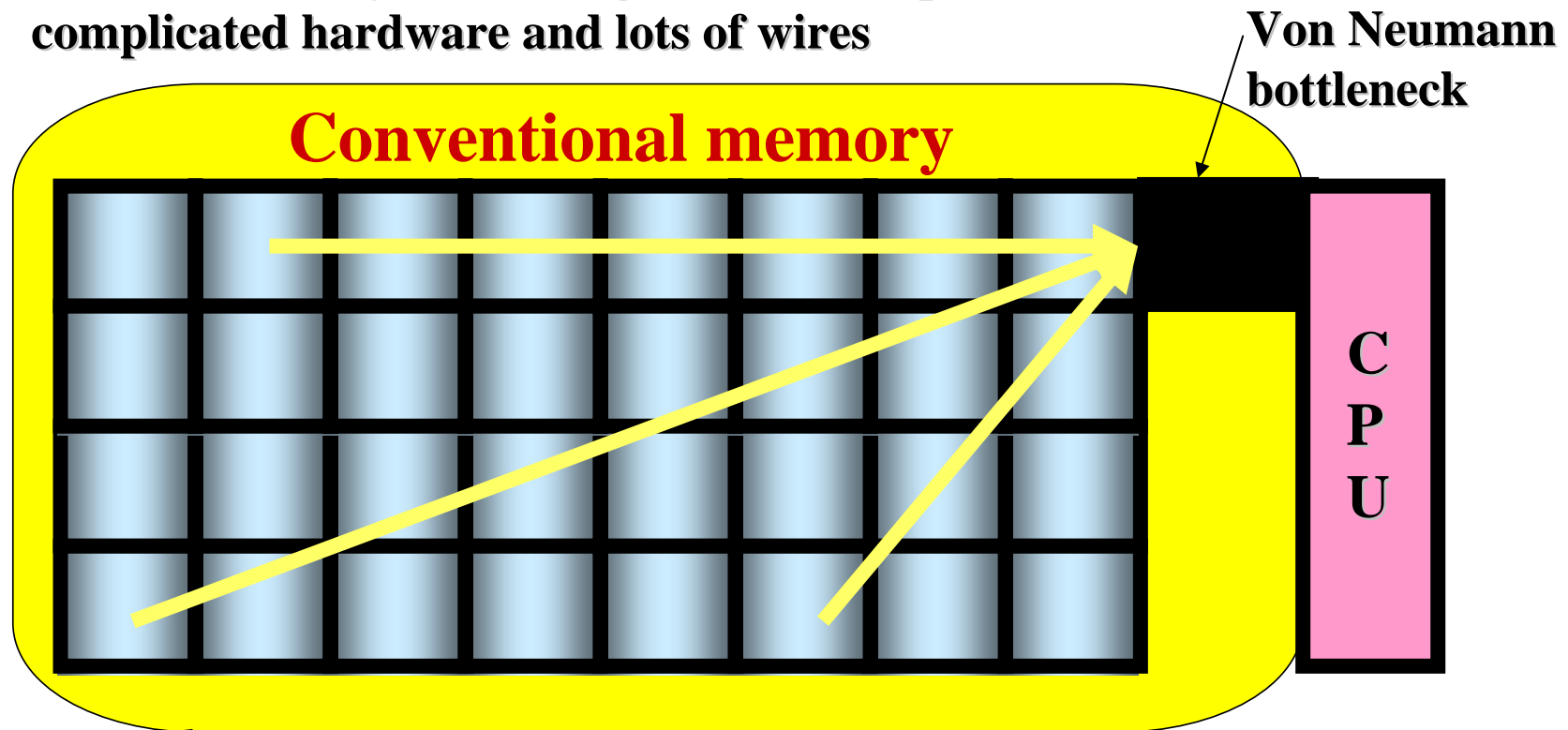
***Delay line Memory:*** *Feedbacked delay lines are used to keep one bit in it.*

***CCD:*** *transmitting analog signalized electric charge to the output after photo-electric conversion on the image receptor.*

***Marching memory:*** *DRAM type memory with storage and marching functions of information / data.*

# Concept of Conventional Memory

Data accessed by addressing with several procedures and complicated hardware and lots of wires



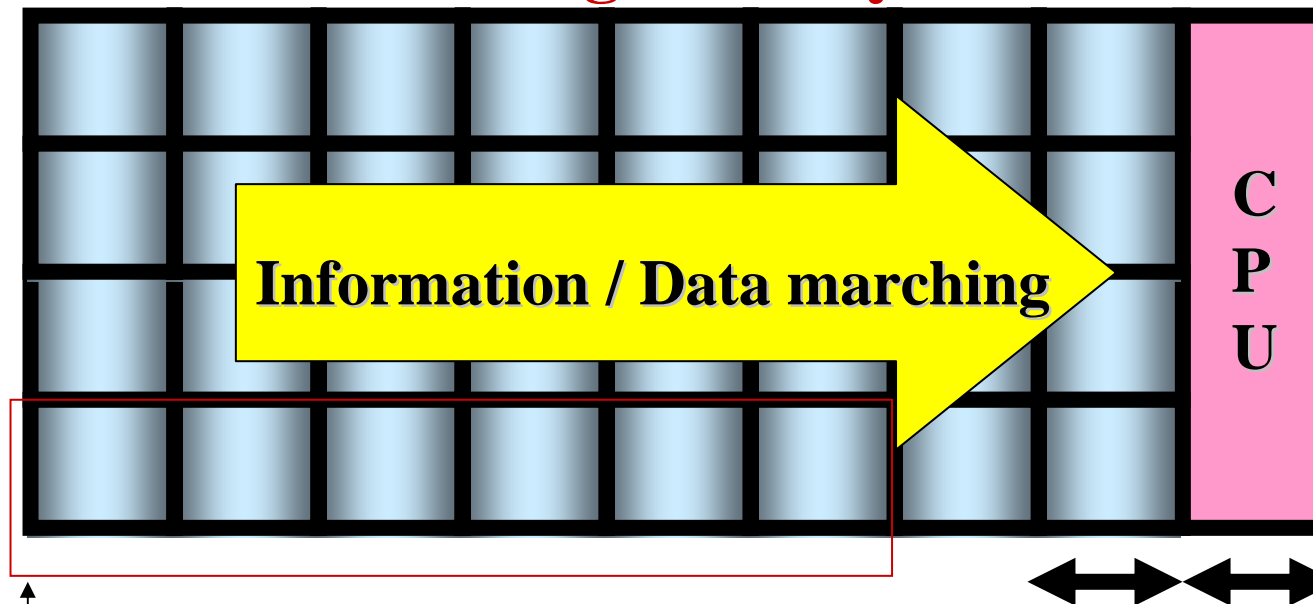
**Memory access time is 1000 times the clock cycle of the CPU**

The memory bottleneck as a memory wall

# Concept of Marching Memory

Data marches, column by column, to the “DRAM” pins / edge  
No long addressing / sense lines.

## Marching memory



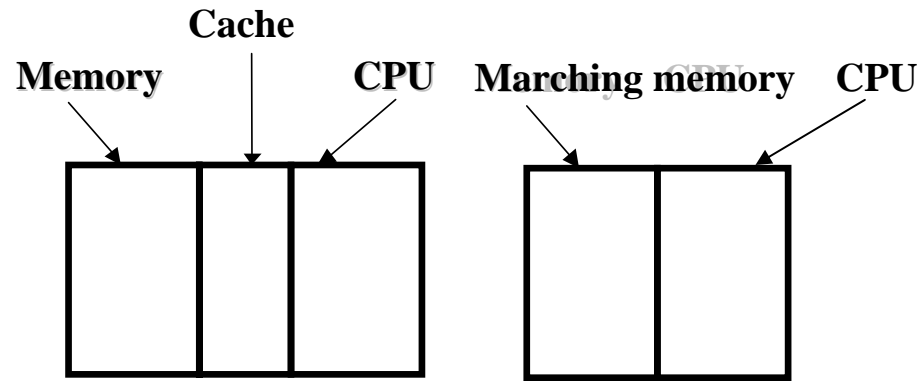
**1 memory unit marching time = CPU's clock cycle**

Any portion available to be off for energy saving, which means that  
all the cells / memory units are not always active through the control.



# Computer Organization Alternatives

---



(a) Computer organizations

Increasing bandwidth

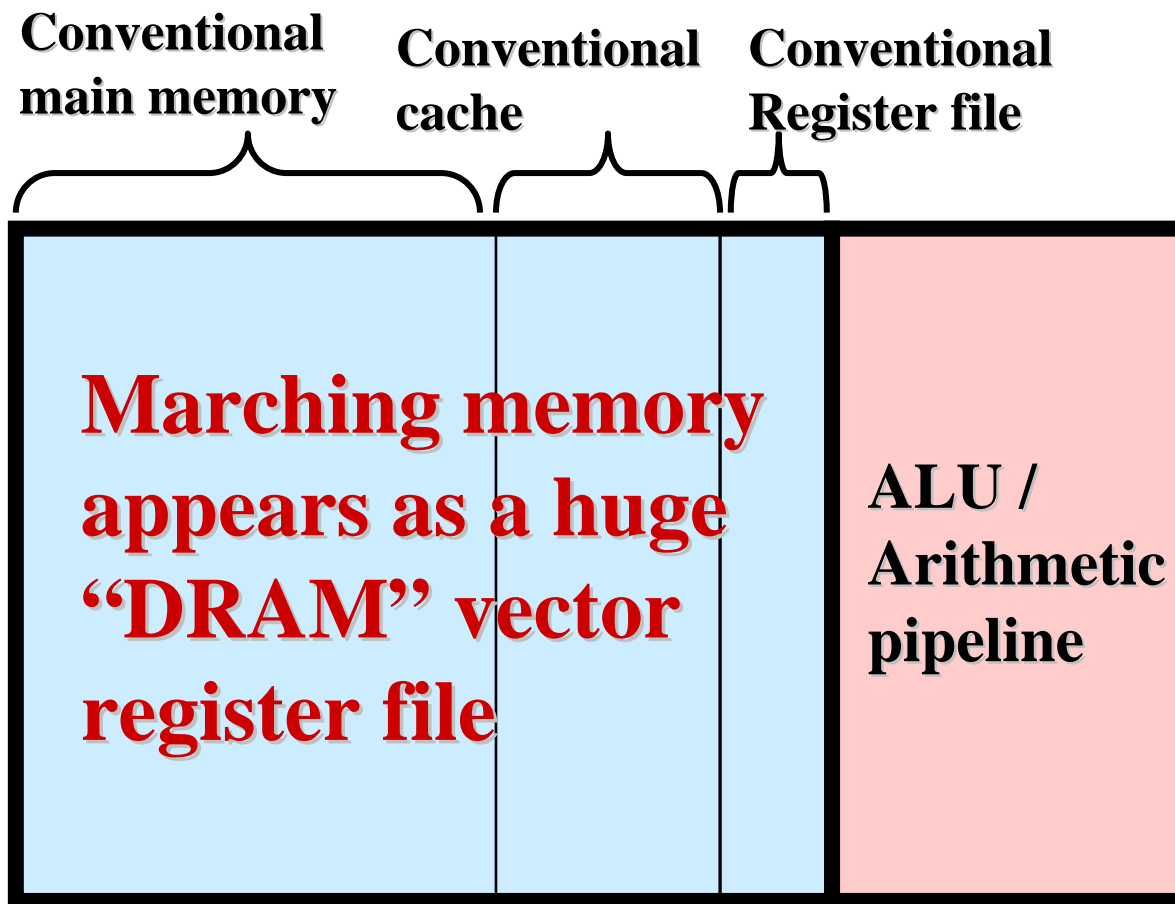


The memory bottleneck as a memory wall

(b) Bandwidths

# Organization Consisted of Marching Memory and ALUs / Arithmetic Pipelines.

---



# Kinds of Marching Memory

---

## **1. Simple Marching Memory** for vector data and streaming data

### **1-1. Extended Simple Marching Memory** for random access with high locality of data

## **2. Complex Marching Memory** for random access

# Classification of Marching Memory

---

**Definition:** The abbreviation of Marching Memory is MM

## 1) Simple MM

**Sequential Access Mode**

As a **core** of a complex MM <<<<<

As an independent MM **unit** for streaming/vector data only

As an independent MM **unit** for sequential programs only

## 2) Extended simple MM

**Random Access Mode with High Locality**

As a **core** of a complex MM <<<<<

As an independent MM **unit** for data with high locality

As an independent MM **unit** for programs without long distance branches

## 3) Complex MM

**Random Access Mode with Low Locality**

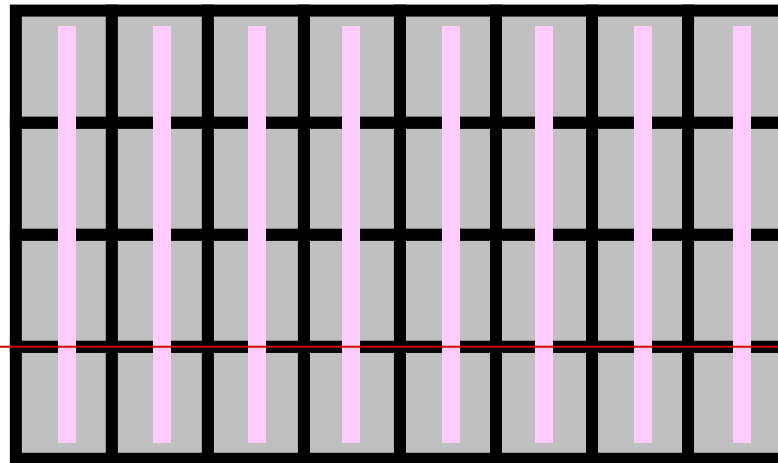
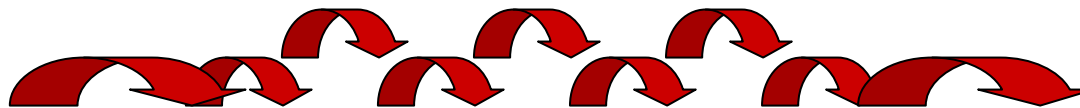
As an independent MM **unit** for huge programs with long distance branches

As an independent MM **unit** for general types of data

# Structure of a Simple Marching Memory

From the previous chip

To the arithmetic units

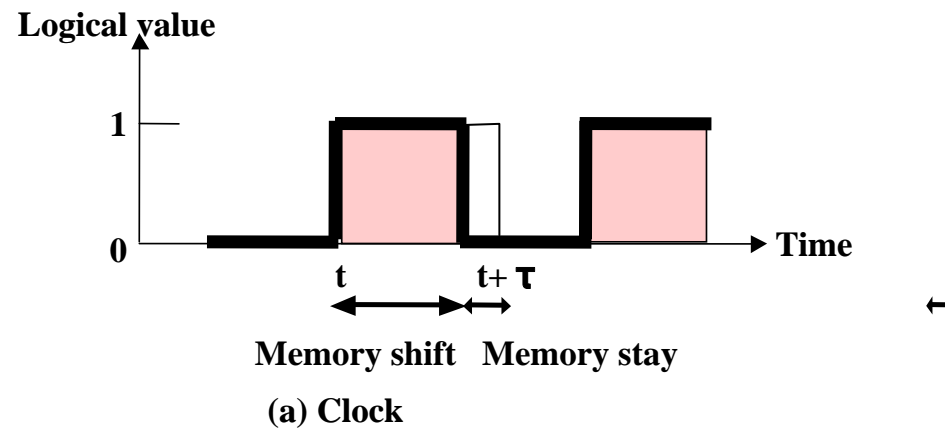
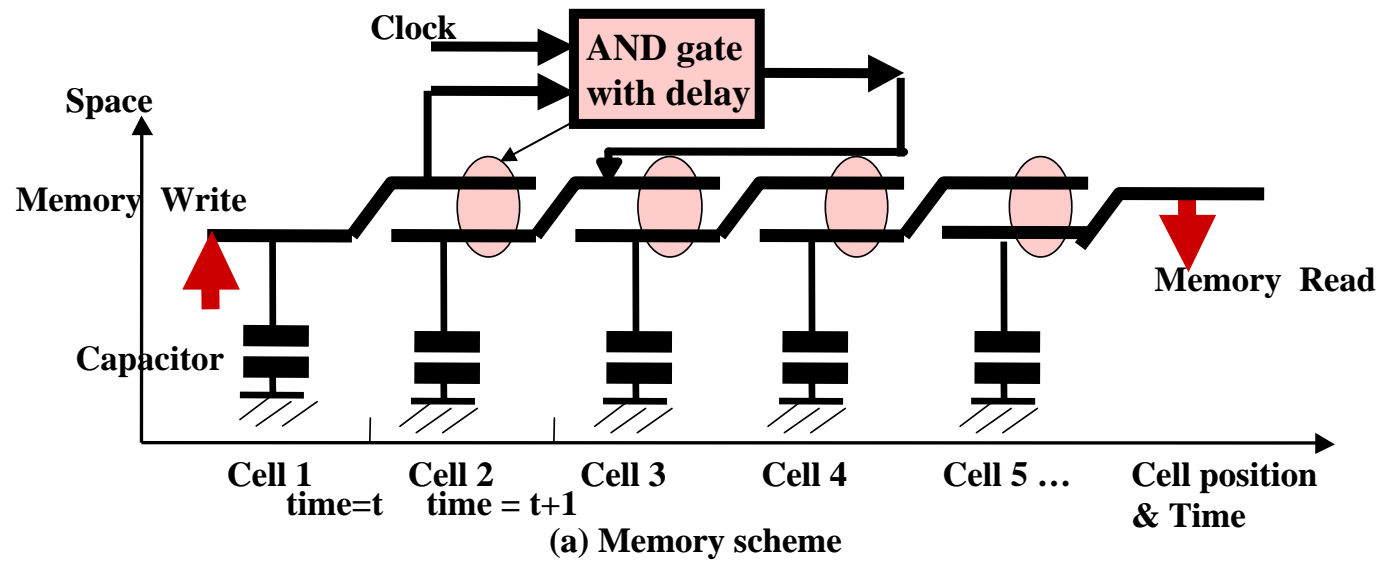


Almost no long wires on a chip because of minimum addressing overhead, no precharge, no activation

Any portion available to be off for energy saving, which means that all the cells / memory units are not always active through the control.

**Information / Data marching**

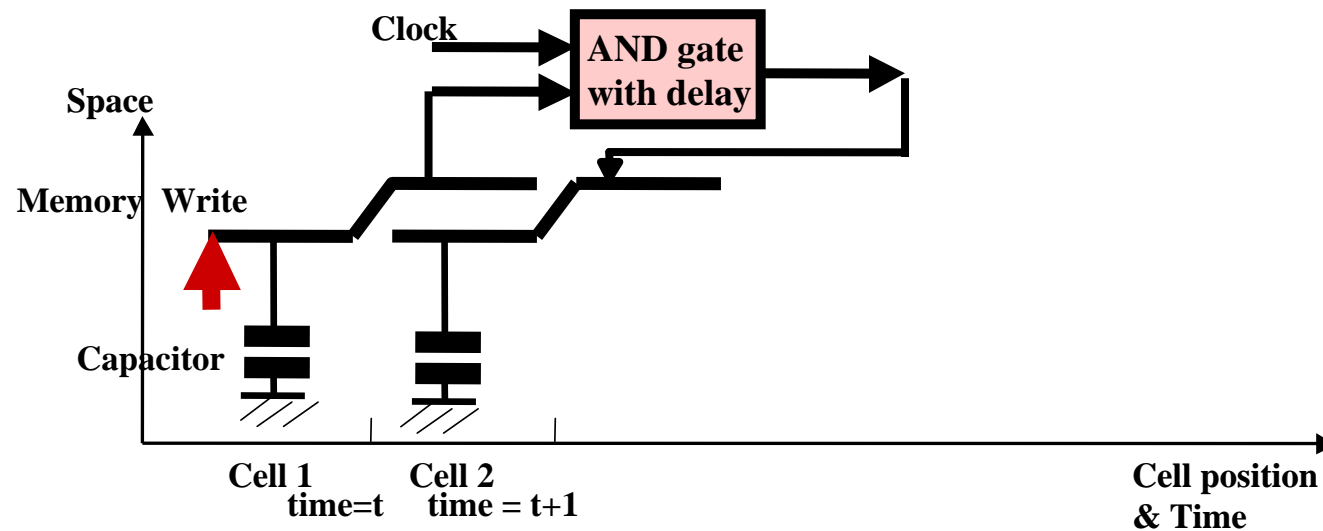
# Circuit for a simple marching memory for vector and streaming data



# One stage of Marching Memory consisting of an AND gate and one capacitor

---

**Vague part!**



# How to implement Marching Memories

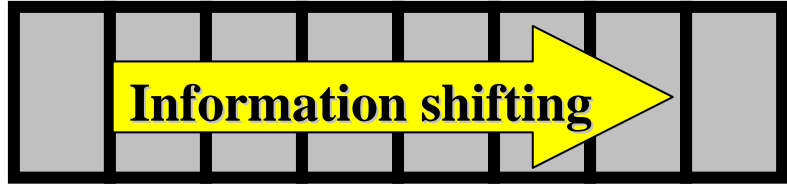
---

- 1. Reliable implementation by 2-dimensional arrays of flip-flops**
- 2. A definite way of using memory cells that have one normal AND gate consisting of several CMOS FETs, and one capacitor**
- 3. An expected way of using memory cells that have one (special) AND gate function MOS FET and one capacitor**

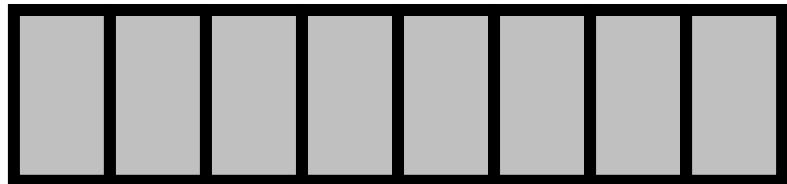


# Extended simple marching memory

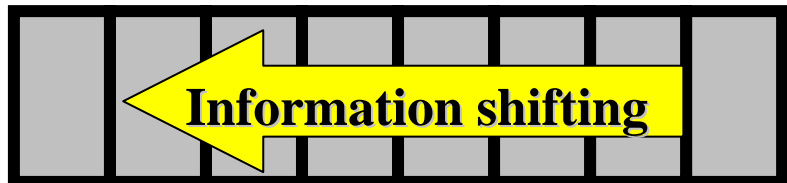
---



(a) Shifting (marching) right

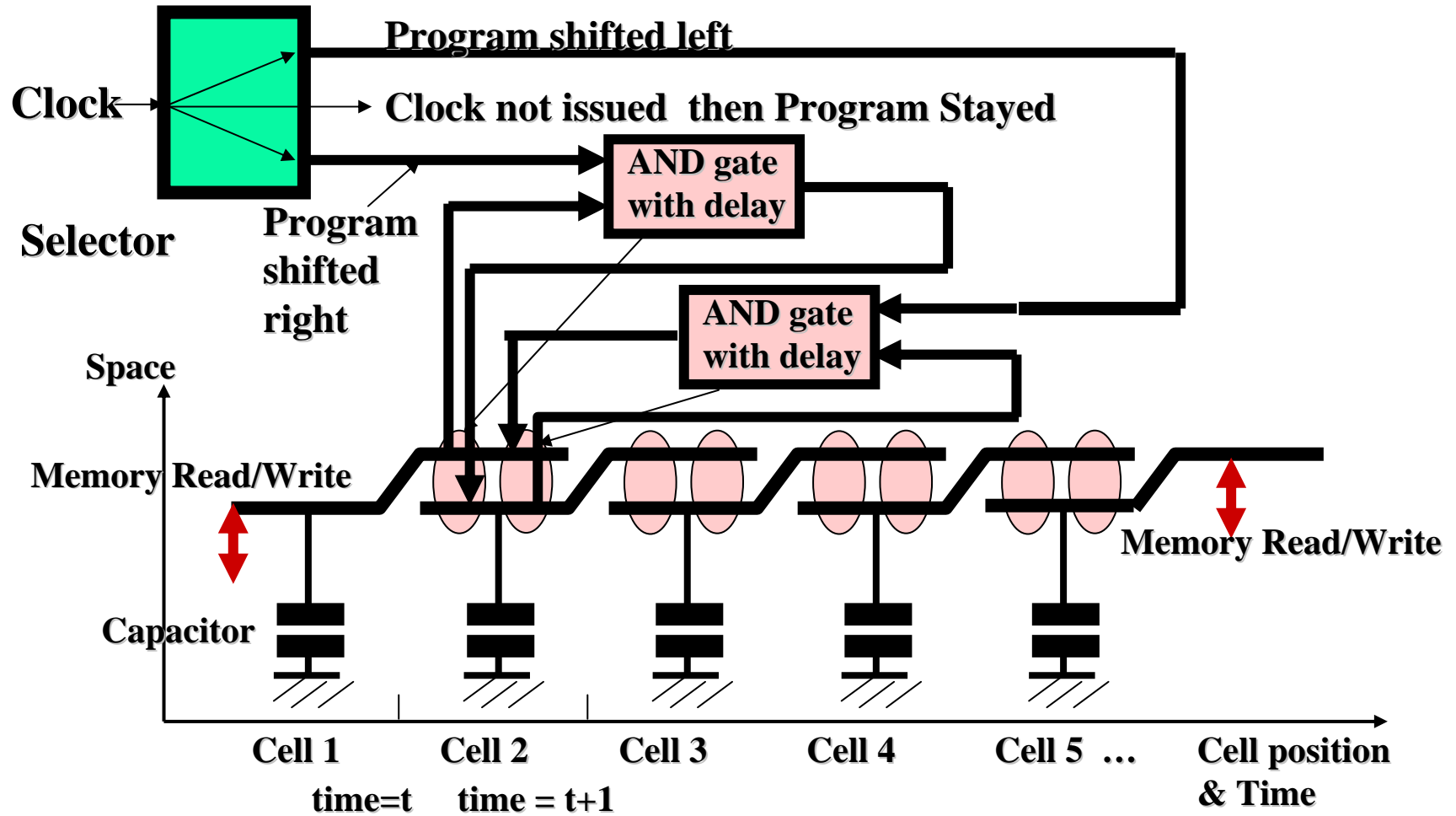


(b) Staying



(c) Shifting (marching) left

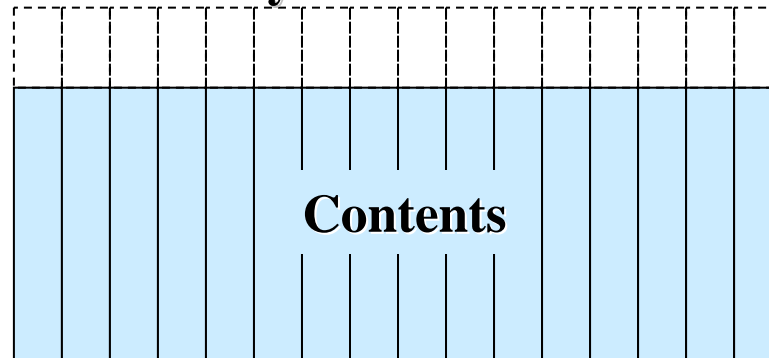
# Circuit for extended simple marching memory



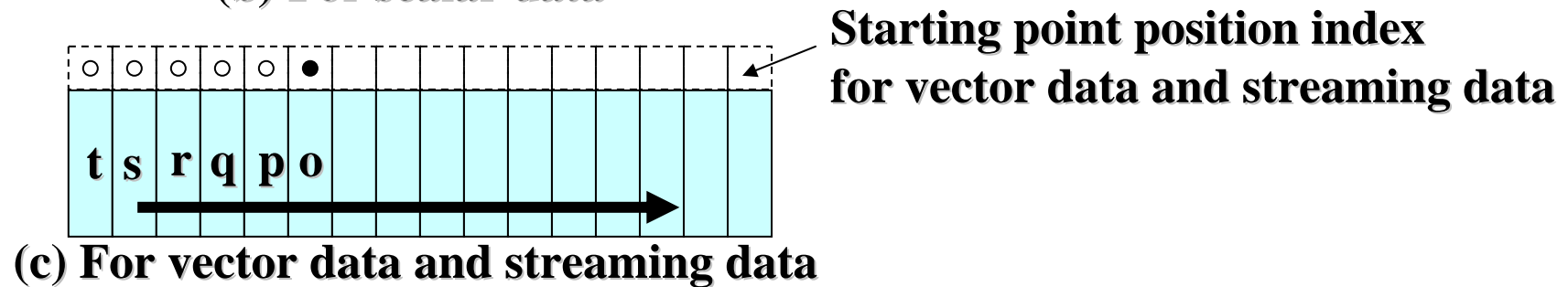
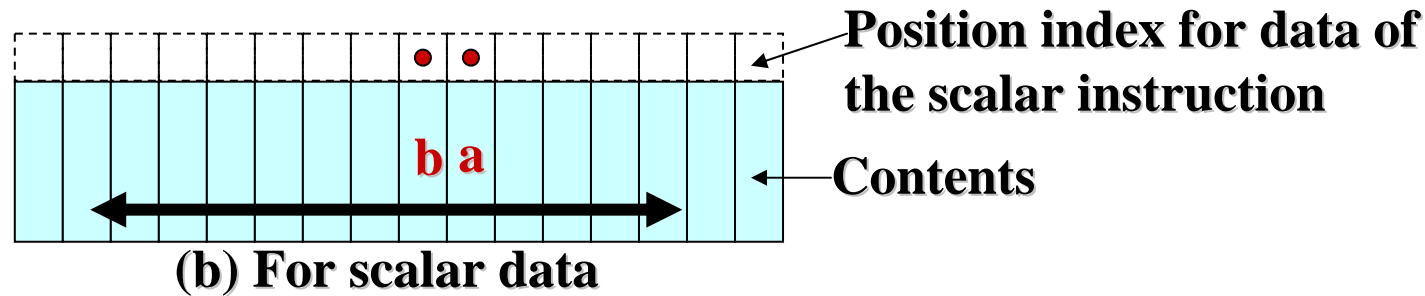
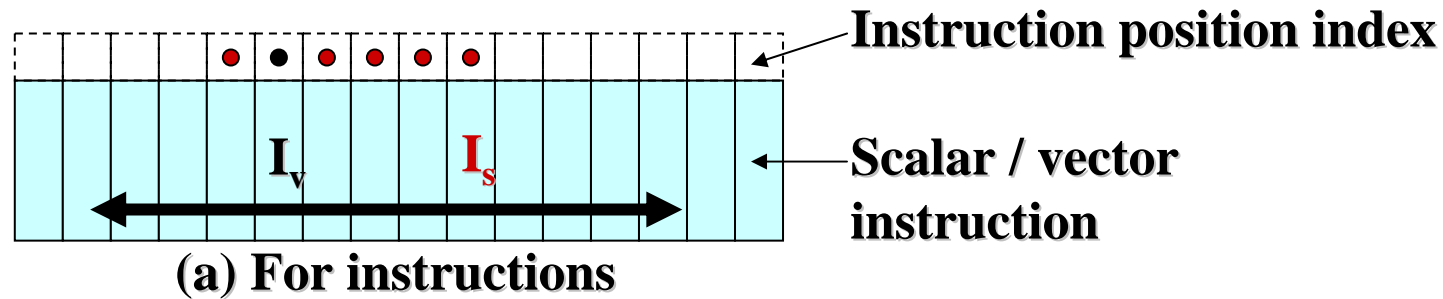
# Accessing data in marching memory requires marching to the correct column

---

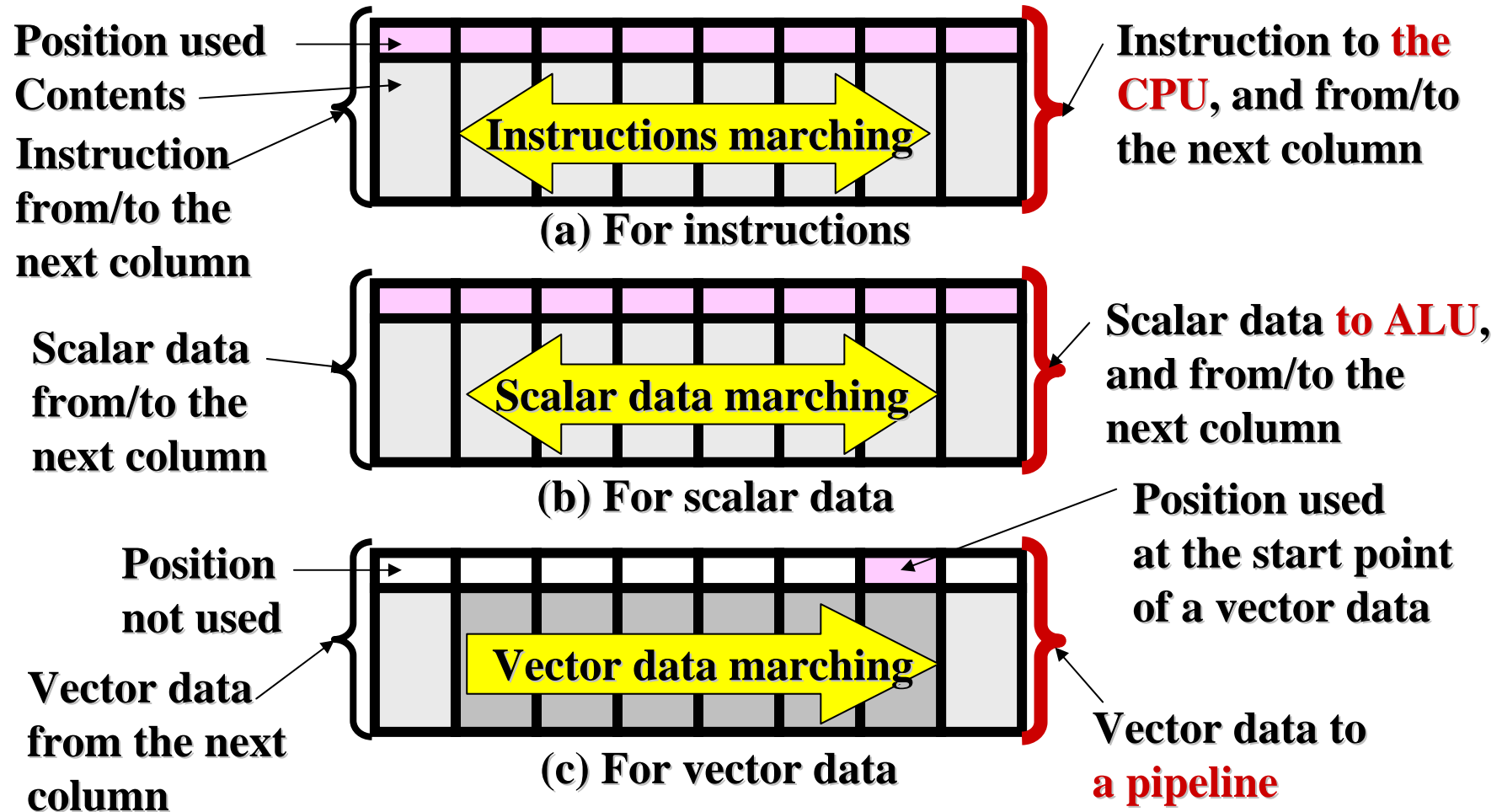
**Position indexes (tags) relative in  
marching information / data, that  
are created dynamically with  
variables by a counter**



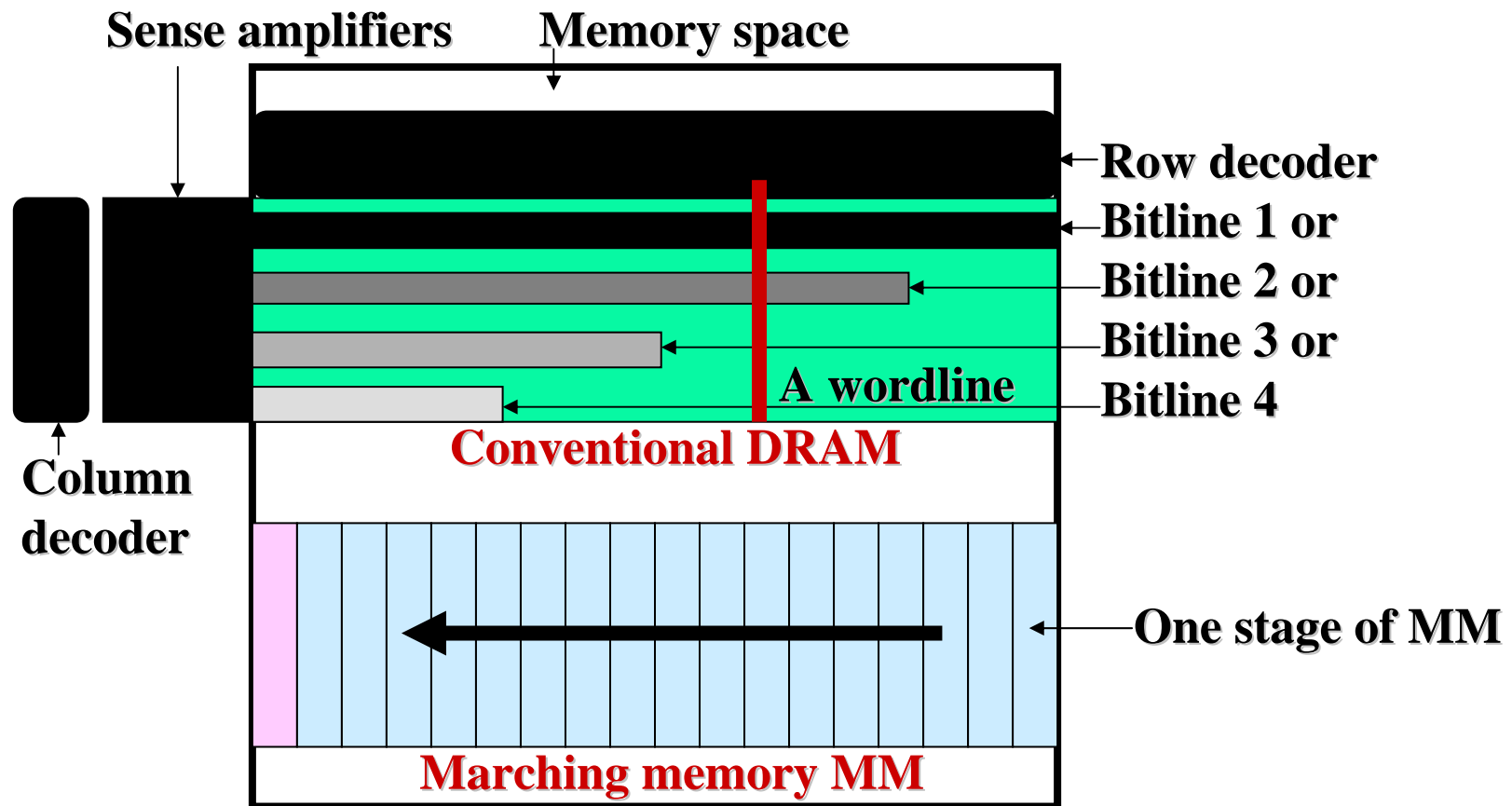
# Operation of marching memories in action



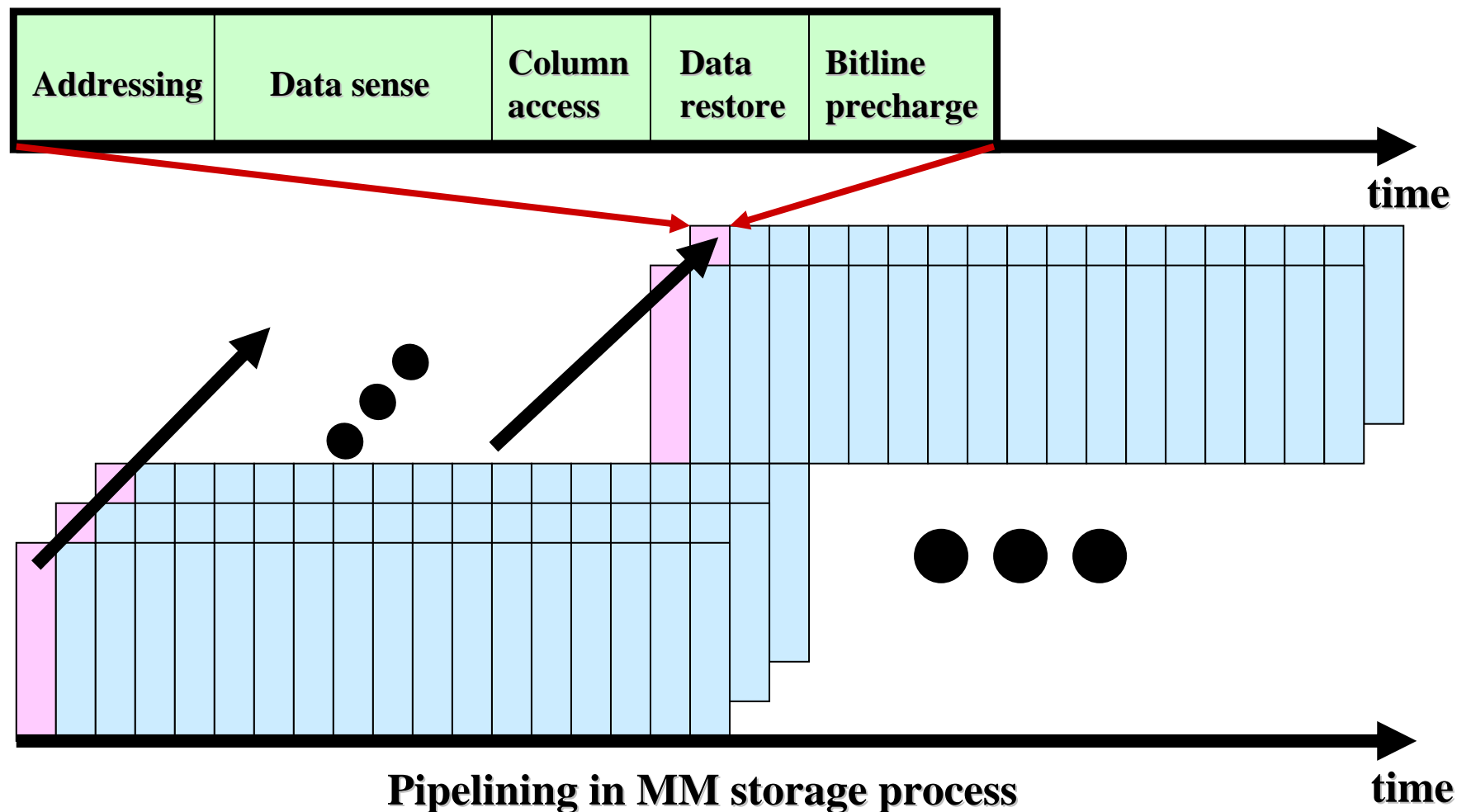
# Pragmatics of marching memories in general uses



# Comparison of MM with conventional DRAM in a memory cycle in terms of hardware quantity

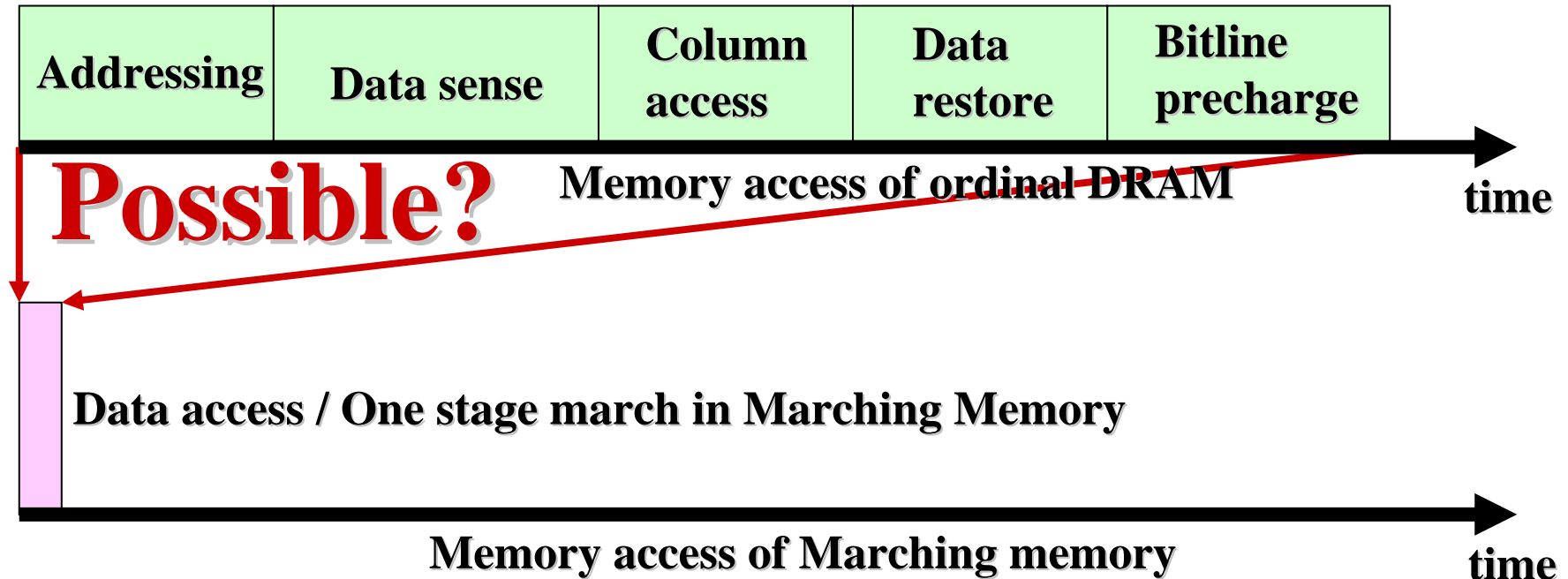


**Pipelining in MM but not so serious in energy consumption due to shorter processing time and if any, lower clock frequency even still higher**



# Comparison of MM with conventional memory in a typical read cycle

**And besides refresh!**



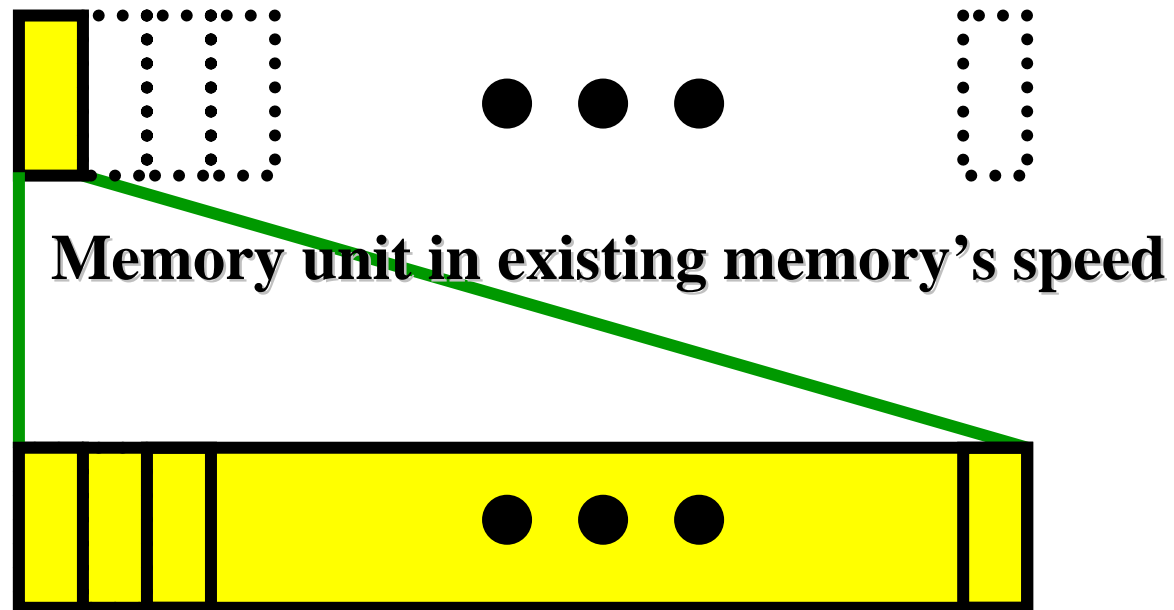


# (Extended) Simple marching memory speed

1 memory  
unit time  
in existing  
memory



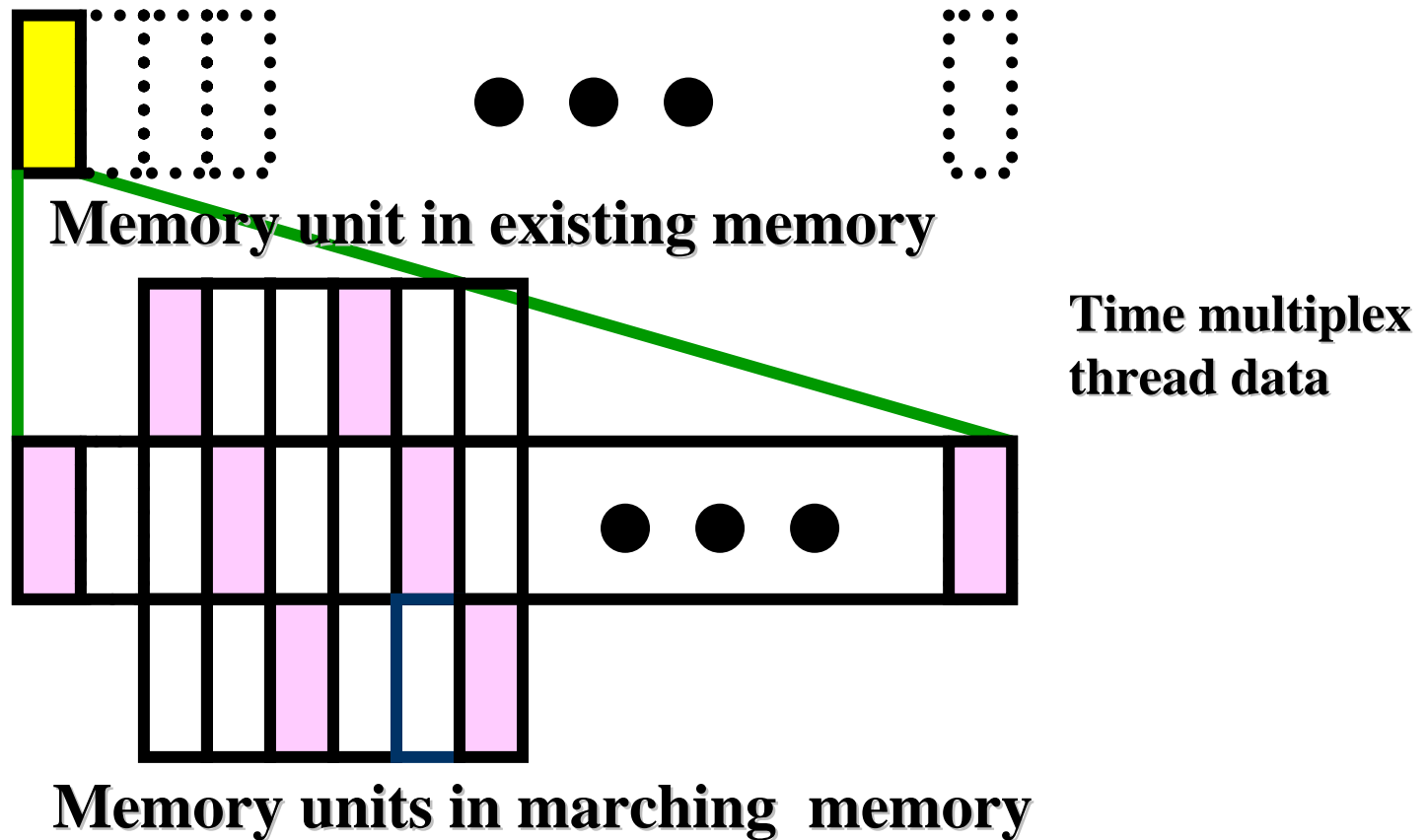
100 memory  
unit time in  
marching  
memory



The other 99 memory units of marching memory could be available in 100 memory units.

# Scalar data for threads in a process, needs compiler support

---



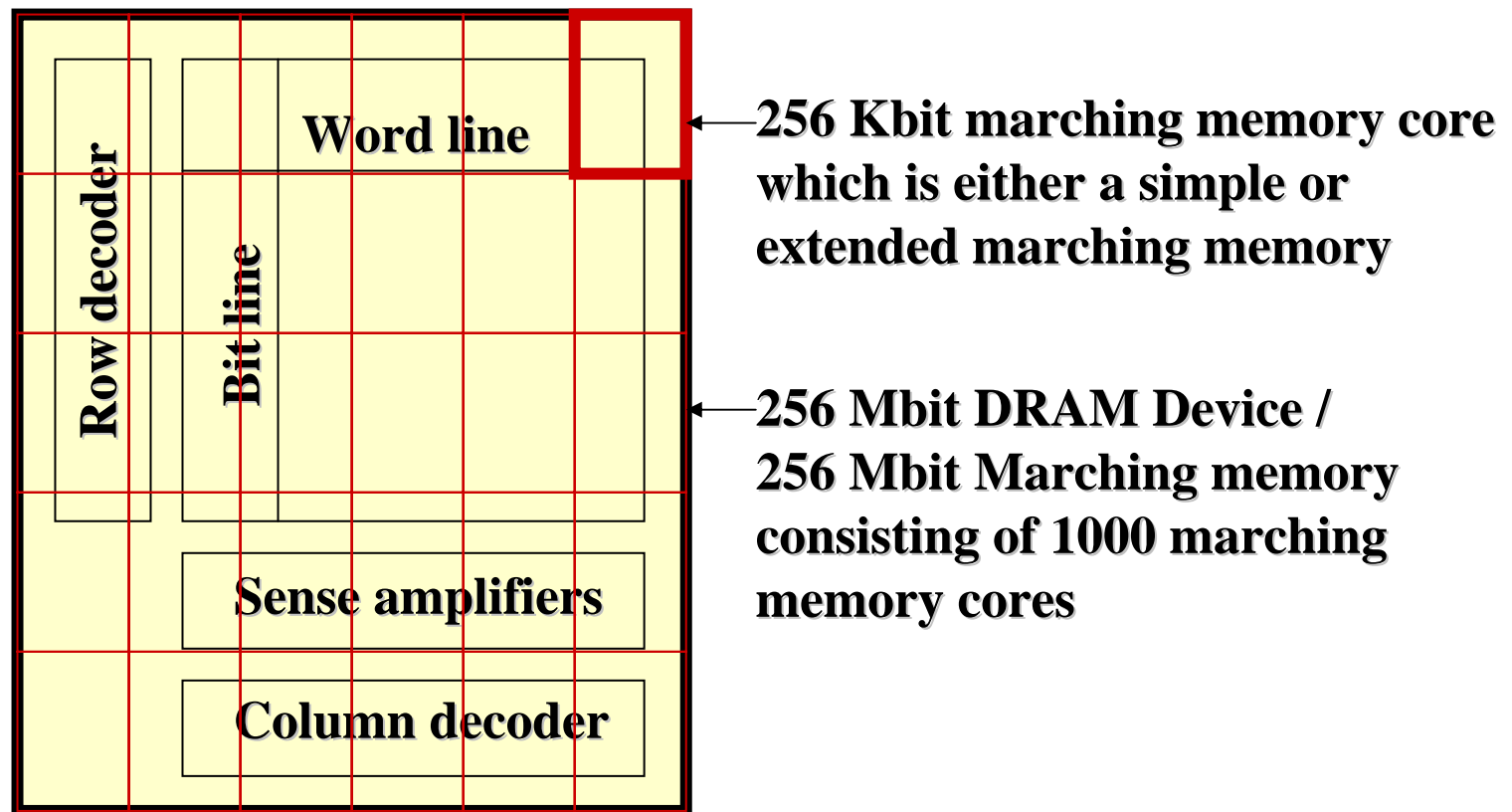
# Complex marching memory

---

1. Sometimes it is required to access a remote column
2. Use multiple (extended) simple marching memory cores with interconnection to a single simple marching memory to interface to arithmetic
3. Provides random access facility among multiple (extended) simple marching memory cores

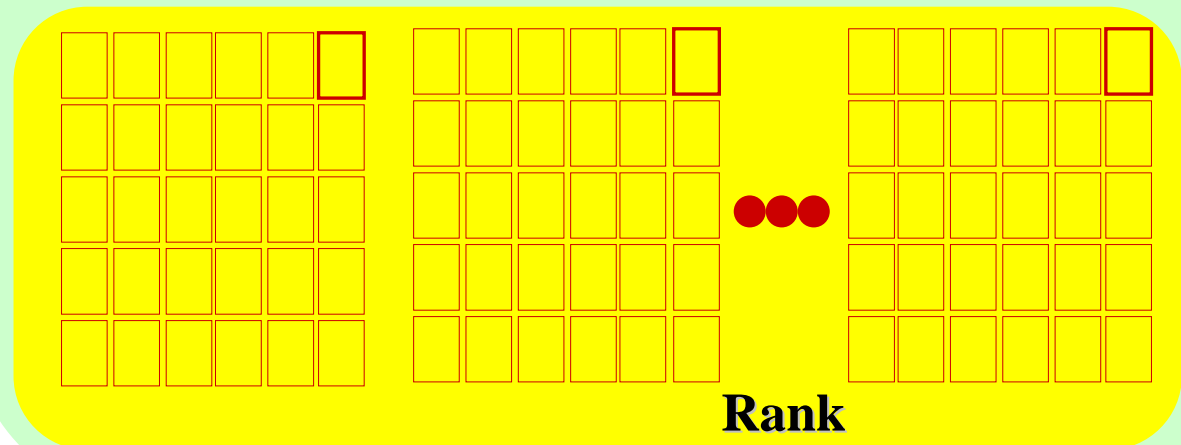
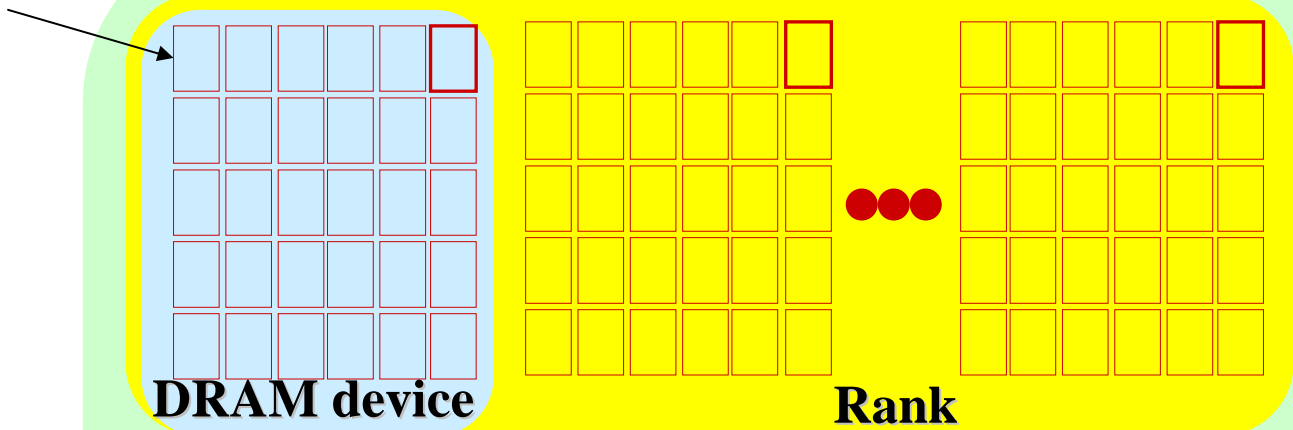
# Comparing conventional DRAM Device to complex (multi-core) marching memory

---



# Structure of conventional DRAM architecture

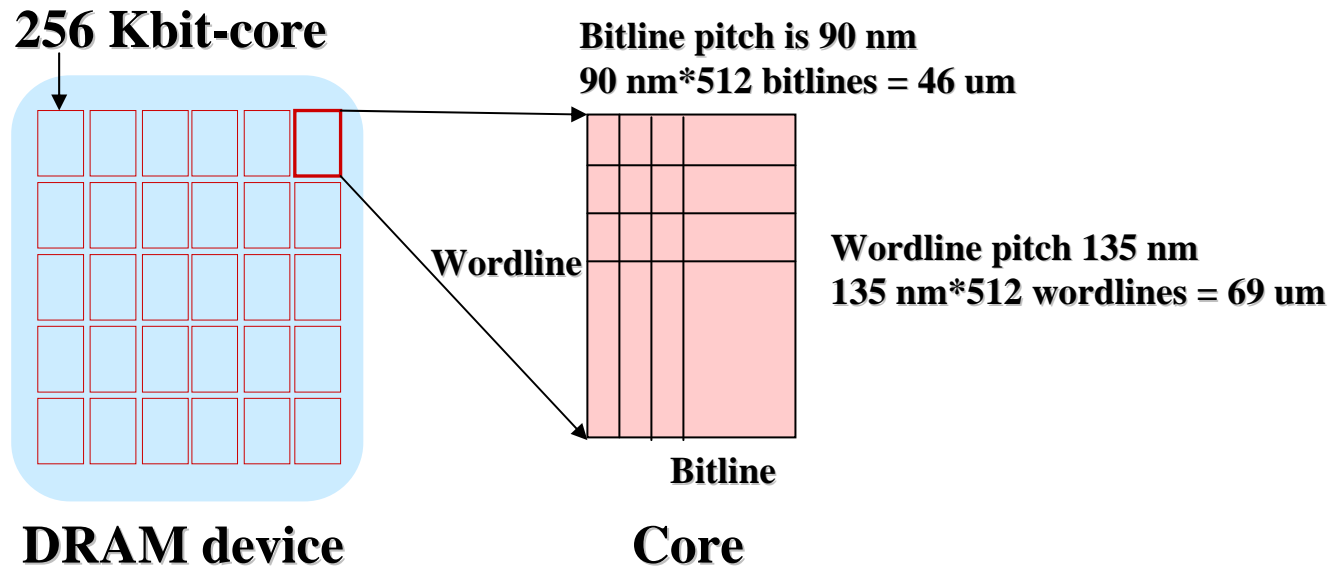
256 Kbit-core



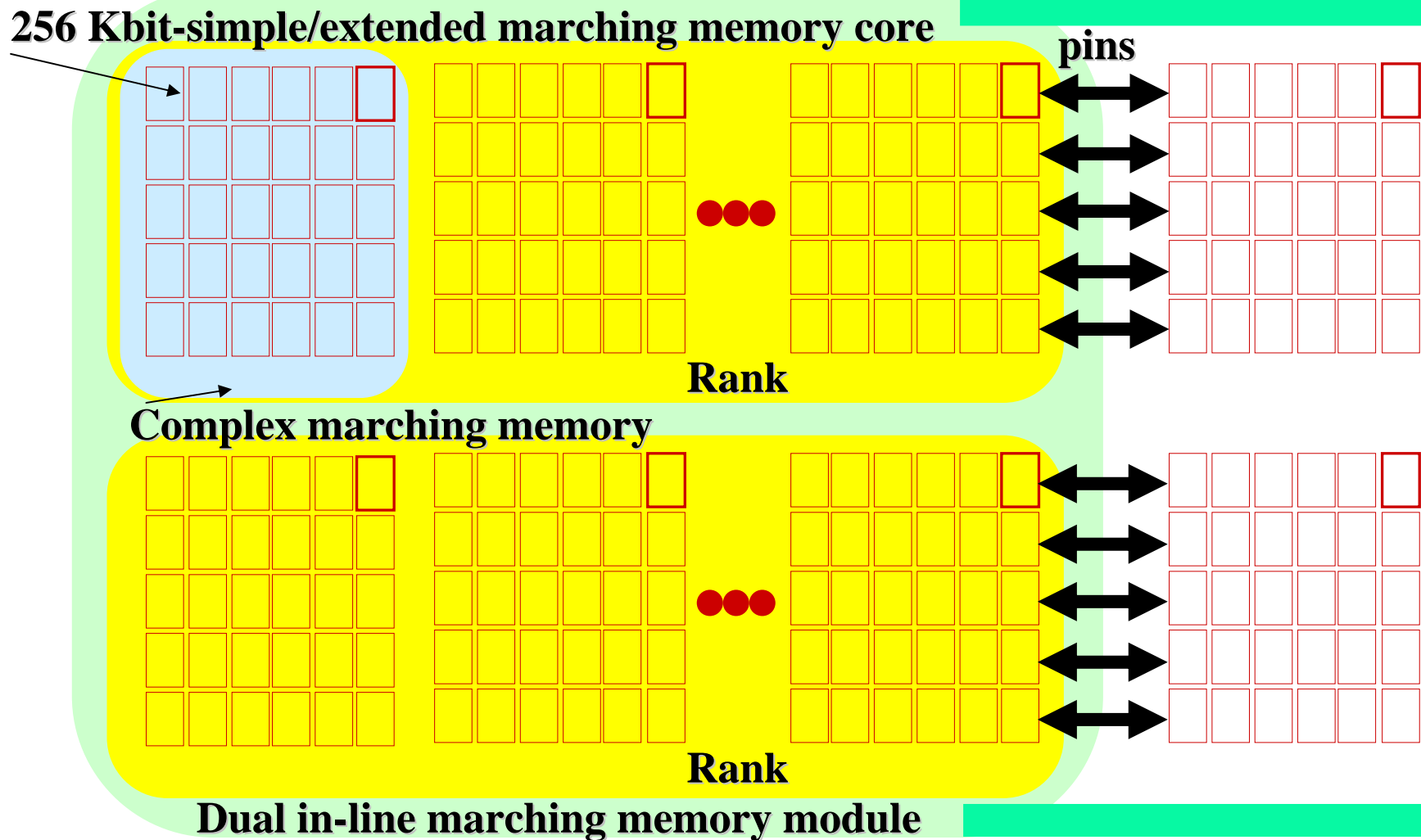
Dual in-line memory module

# Structure of conventional DRAM core

---



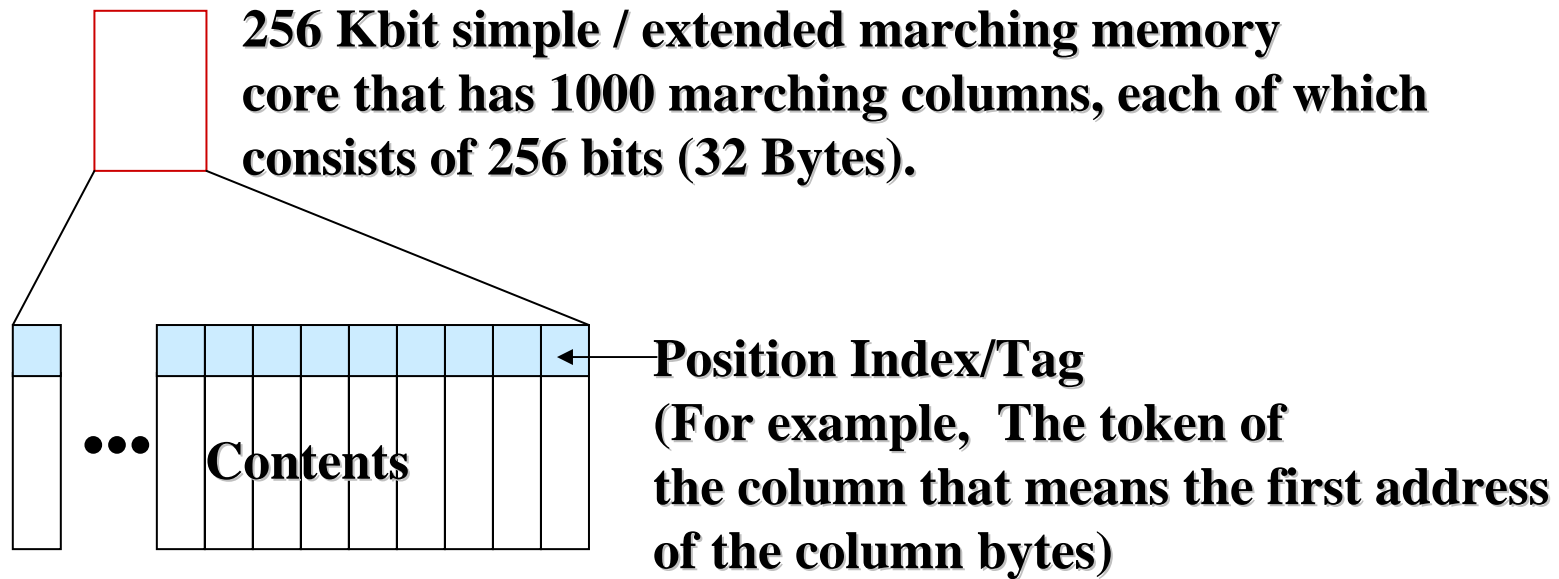
# Structure of complex (multi-core) marching memory



# Complex Marching Memory Core

## -Its Basics-

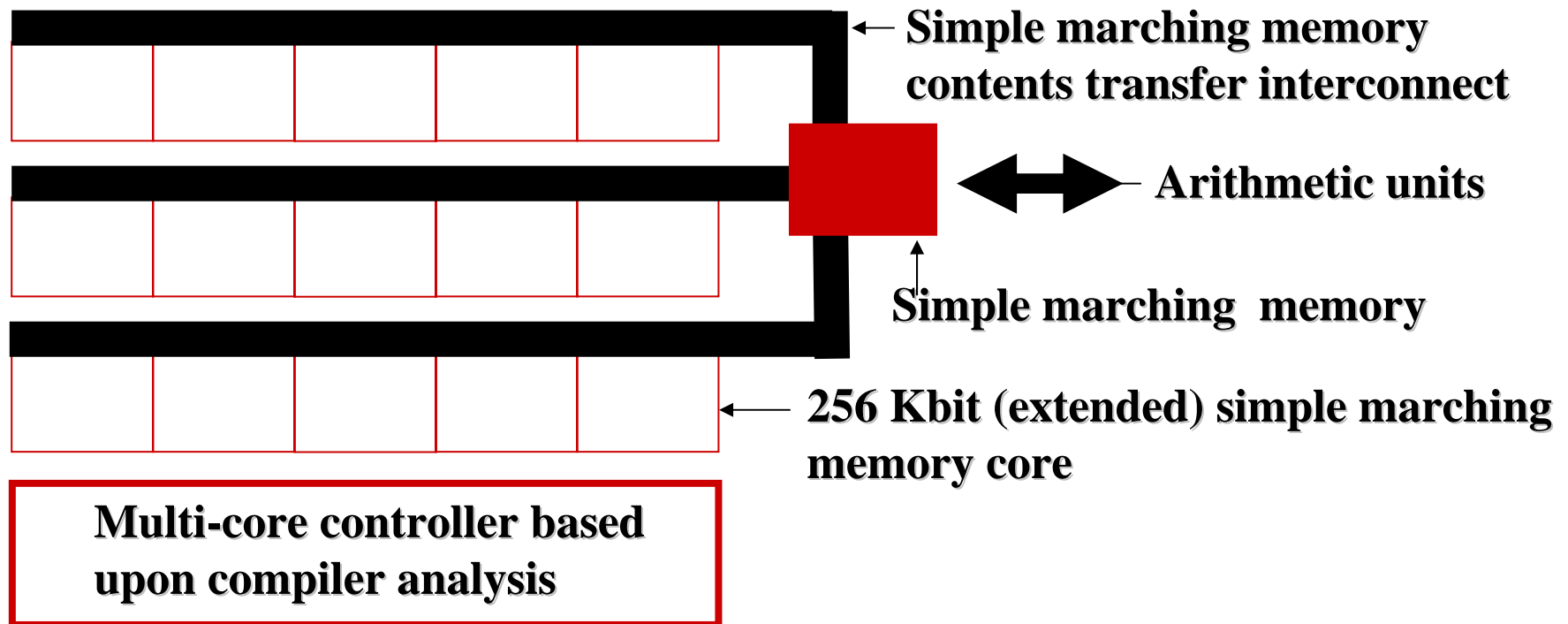
---





# Interconnecting 1000 simple MM cores in complex marching memory

One solution : Simple marching memory cores are interconnected to a single simple marching memory as the interface between the multiple cores and the arithmetic units. Data is transferred column by column between cores.



# Features of Marching Memory

---

**Row decoder**  
**Wordlines**  
**Column decoder**  
**Bitlines**  
**Sense Amplifiers**

**Disappeared!**

## **And then**

**None of addressing, sensing, restore, precharge and refresh exist**  
**MM's Downsizing with much less hardware**  
**No long wires even among complex MMs,**  
**Simple structure of MM systems total**

## **As a result**

**High speed**  
**Large capacity**  
**Low energy consumption**  
**Low cost**

# Conclusions

---

**We have presented two types of marching memory (MM) using DRAM type technology**

**1. Simple MM is fast and suitable for streaming applications**

**1-1. Extended Simple MM for random access with high locality of data**

**2. Complex MM for random access in a program uses multiple (extended) simple MM cores**

**This is slower and requires multi-core interconnection networks. More research is needed.**

**MM has potential for faster memory technology especially with good compiler support.**