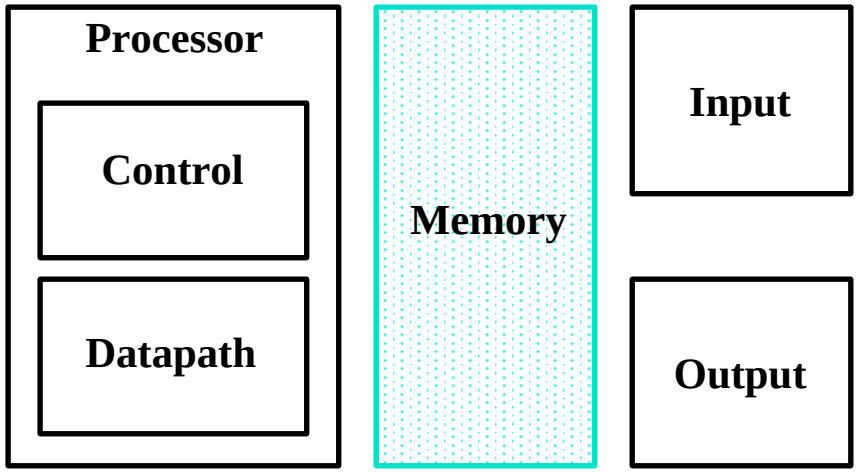
Memory System I

The Big Picture: Where are We Now?

- **The Five Classic Components of a Computer**



- **Today's Topics:**
  - Locality and Memory Hierarchy
  - SRAM Memory Technology
  - DRAM Memory Technology
  - Memory Organization

Technology Trends (from 1st lectures)

	Capacity	Speed (latency)
Logic:	2× in 3 years	2× in 3 years
DRAM:	4× in 3 years	2× in 10 years
Disk:	4× in 3 years	2× in 10 years

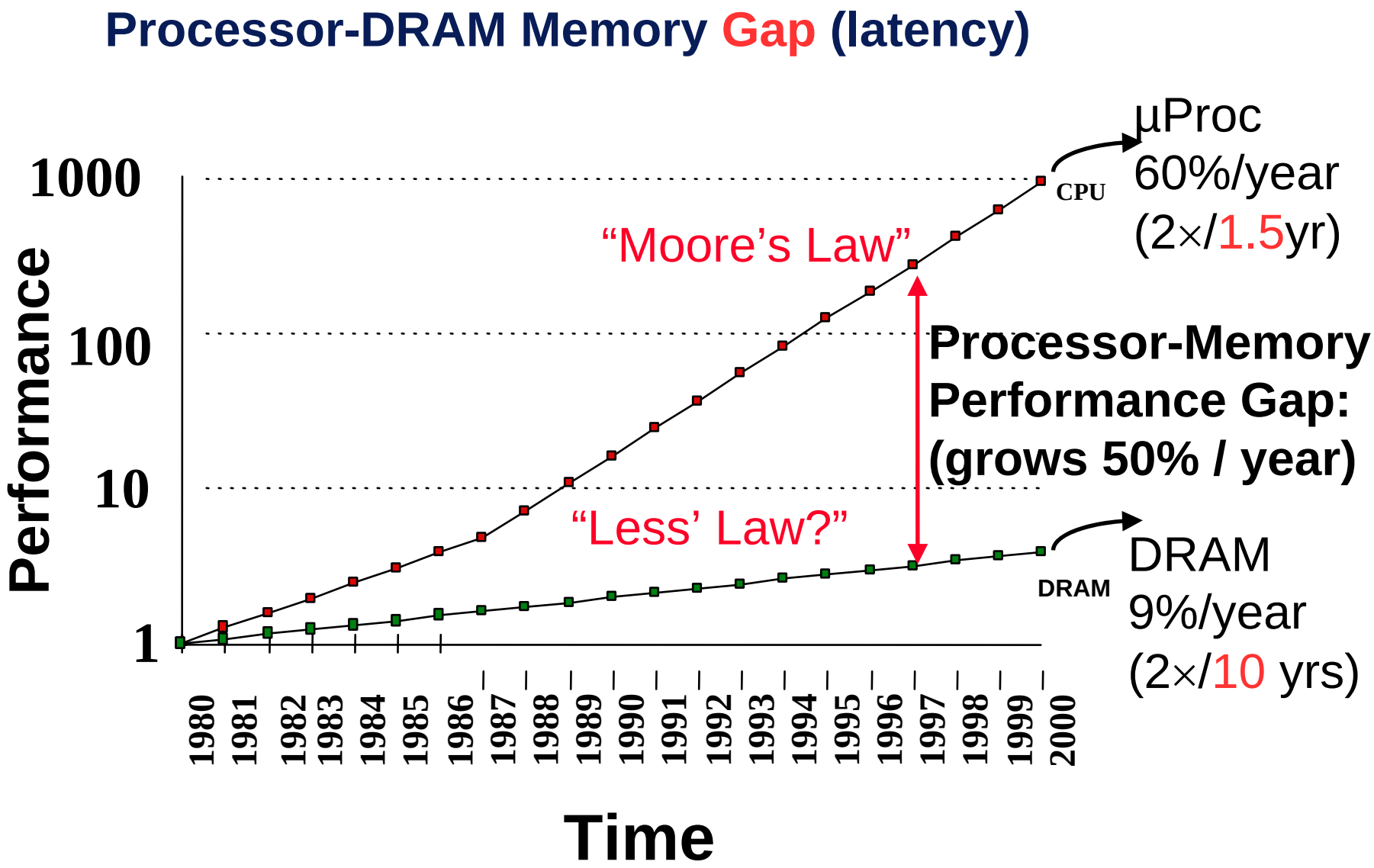
1000:1!

2:1!

DRAM

Year	Size	Cycle Time
1980	64 Kb	250 ns
1983	256 Kb	220 ns
1986	1 Mb	190 ns
1989	4 Mb	165 ns
1992	16 Mb	145 ns
1995	64 Mb	120 ns

Who Cares About the Memory Hierarchy?



The Goal: Illusion of Large, Fast, Cheap Memory

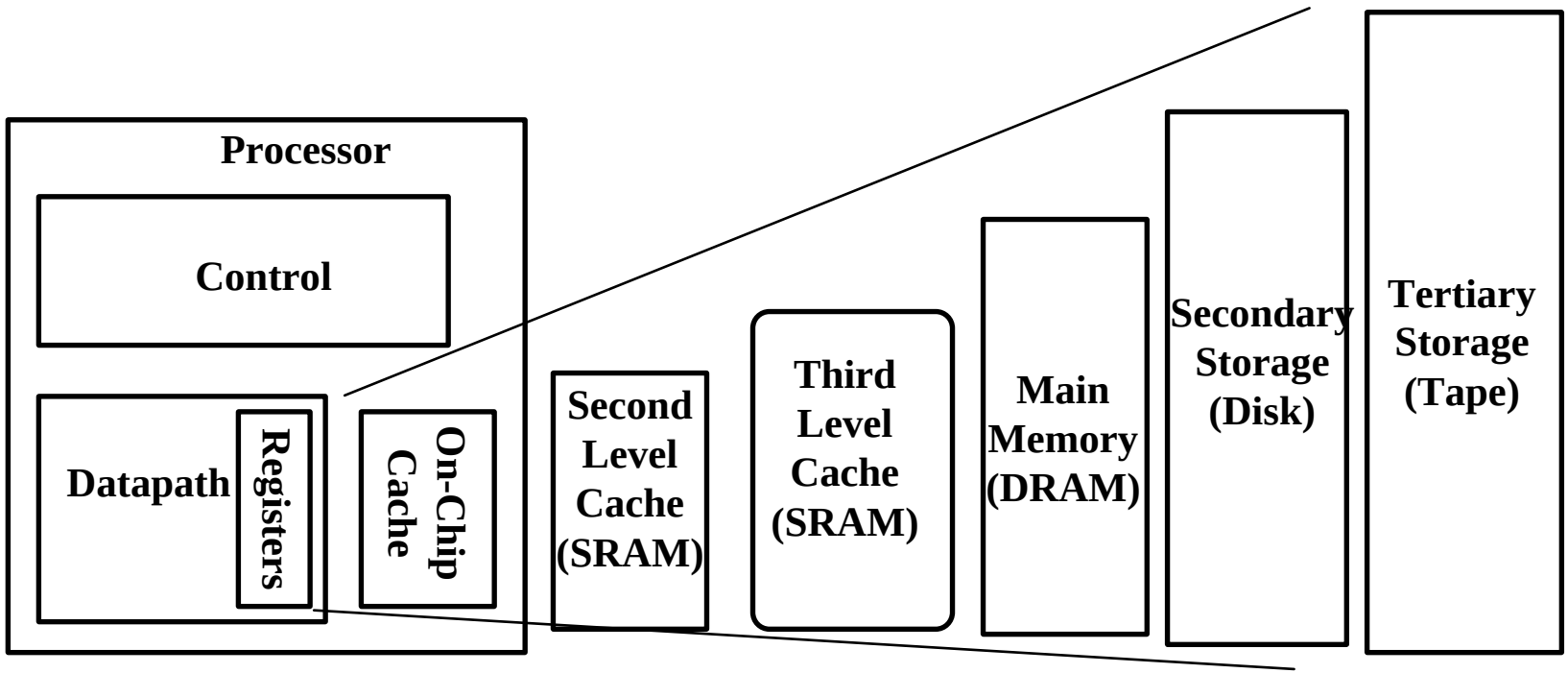
- **Fact:**

  Large memories are slow

  Fast memories are small
- **How do we create a memory that is large, cheap and fast (most of the time)?**
  - Hierarchy
  - Parallelism

Memory Hierarchy of a Modern Computer System

- **By taking advantage of the principle of locality:**
  - Present the user with as much memory as is available in the cheapest technology.
  - Provide access at the speed offered by the fastest technology.

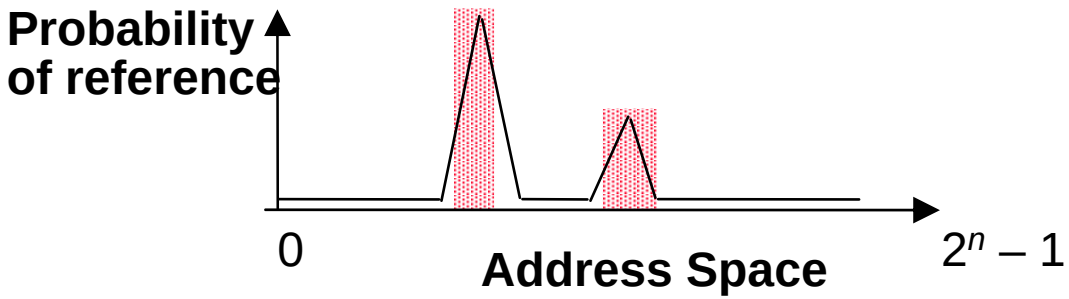


Memory Hierarchy of a Modern Computer System

Let's look at numbers for an Intel Pentium 4, 3.2 GHz Server.

Component	Access Speed (Time for data to be returned)	Size of Component
Registers	1 cycle = 0.3 nanoseconds	8 registers
L1 Cache	3 cycles = 1 nanoseconds	Separate Data and Instruction Caches: 16 Kbytes each
L2 Cache	20 cycles = 7 nanoseconds	256 Kbytes, 8-way set associative
L3 Cache	40 cycles = 13 nanoseconds	4096 Kbytes, 8-way set associative
Memory	300 cycles = 100 nanoseconds	16 Gigabytes

Memory Hierarchy: Why Does It Work? Locality!

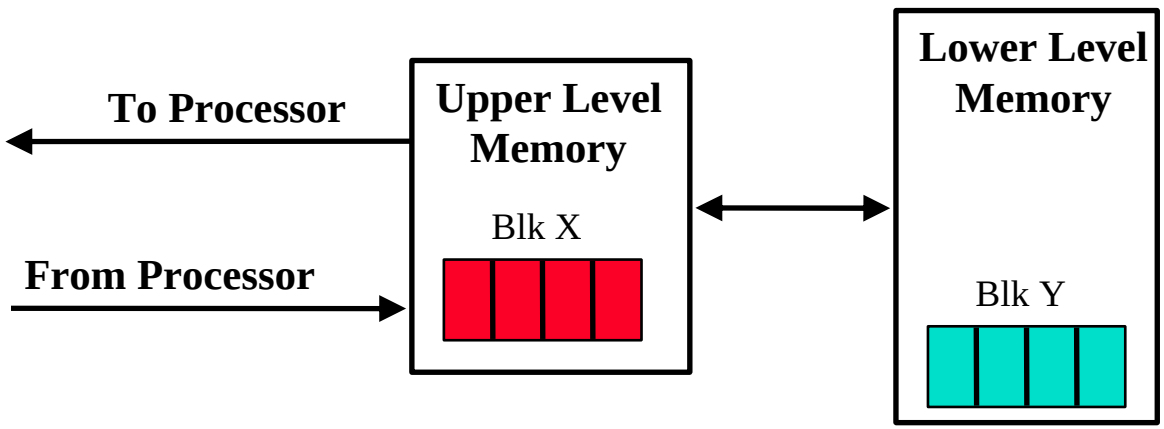


- **Temporal Locality (Locality in Time):**

⇒**Keep most recently accessed data items closer to the processor**

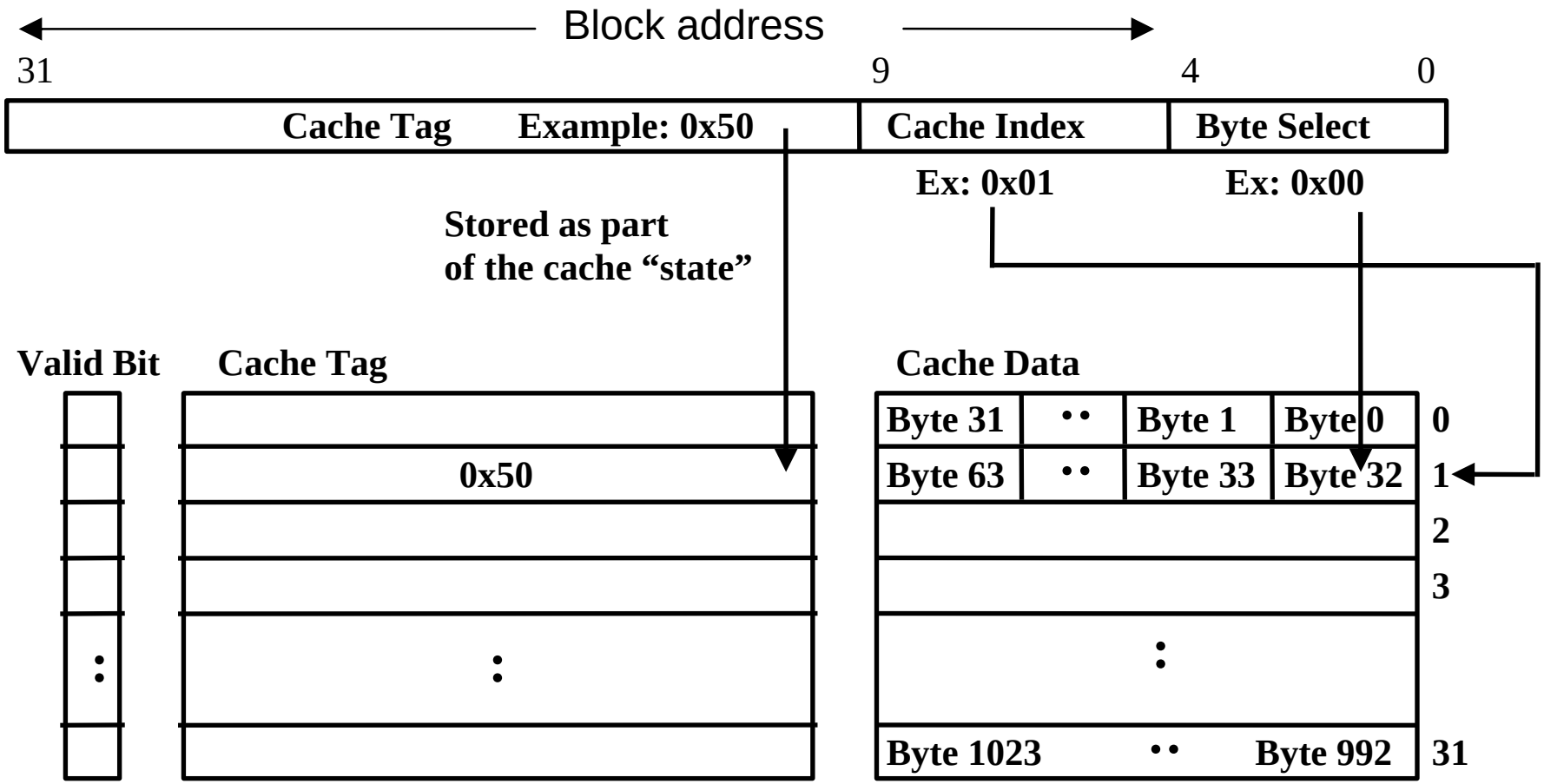
- **Spatial Locality (Locality in Space):**

⇒**Move blocks consisting of contiguous words to the upper levels**



Example: 1 KB Direct Mapped Cache With 32 Byte Blocks

- **For a 2^N byte cache:**
  - The uppermost (32 - *N*) bits are always the Cache Tag
  - The lowest M bits are the Byte Select (Block Size = 2^M)



Memory Hierarchy: Terminology

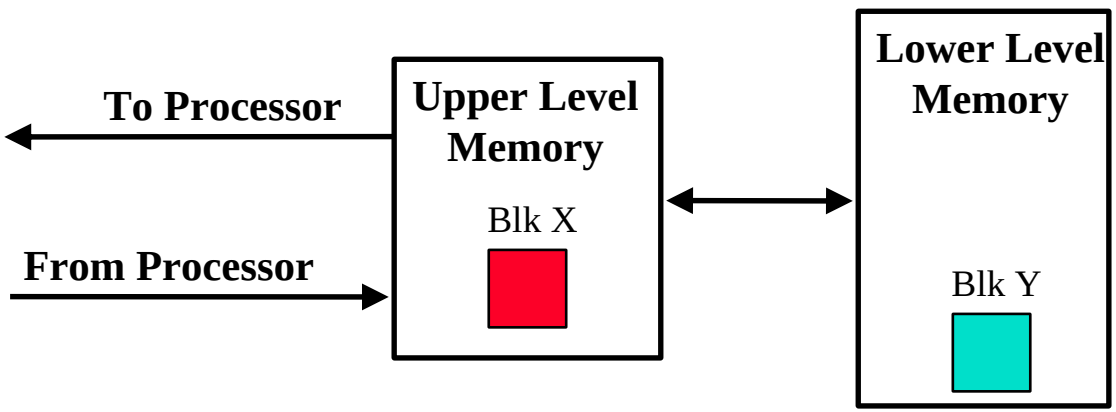
- **Hit: data appears in some block in the upper level (example: Block X)**
  - Hit Rate: the fraction of memory access found in the upper level
  - Hit Time: Time to access the upper level which consists of

    RAM access time + Time to determine hit/miss

- **Miss: data needs to be retrieve from a block in the lower level (Block Y)**
  - Miss Rate = 1 - (Hit Rate)
  - Miss Penalty(Time): **Time to replace a block in the upper level +**

    Time to deliver the block the processor

- **Hit Time << Miss Penalty**



How Is the Hierarchy Managed?

- **Registers ↔ Memory**
  - by compiler (programmer?)
- **cache ↔ memory**
  - by the hardware
- **memory ↔ disks**
  - by the hardware and operating system (virtual memory)
  - by the programmer (files)

Memory Hierarchy Technology

- **Random Access:**
  - "Random" is good: access time is the same for all locations
  - DRAM: Dynamic Random Access Memory
    - High density, low power, cheap, slow
    - Dynamic: need to be "refreshed" regularly
  - SRAM: Static Random Access Memory
    - Low density, high power, expensive, fast
    - Static: content will last "forever"(until lose power)
- **"Non-so-random" Access Technology:**
  - Access time varies from location to location and from time to time
  - Examples: Disk, CDROM, DRAM page-mode access
- **Sequential Access Technology: access time linear in location (*e.g.*,Tape)**
- **We will concentrate on random access technology**
  - The Main Memory: DRAMs + Caches: SRAMs

Main Memory Background

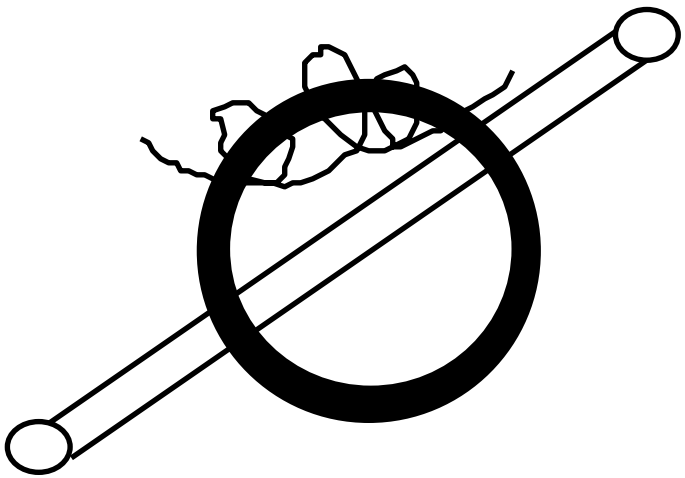
- **Performance of Main Memory:**
  - Latency: Cache Miss Penalty
    - *Access Time*: time between request and when word arrives
    - *Cycle Time*: time between requests
  - Bandwidth: I/O & Large Block Miss Penalty (L2)
- **Main Memory is *DRAM :* Dynamic Random Access Memory**
  - Dynamic, needs to be refreshed periodically (8 ms)
  - Addresses divided into 2 halves (Memory as a 2D matrix):
    - *RAS* or *Row Access Strobe*
    - *CAS* or *Column Access Strobe*
- **Cache uses *SRAM :* Static Random Access Memory**
  - No refresh (6 transistors/bit *vs*. 1 transistor)

Random Access Memory (RAM) Technology

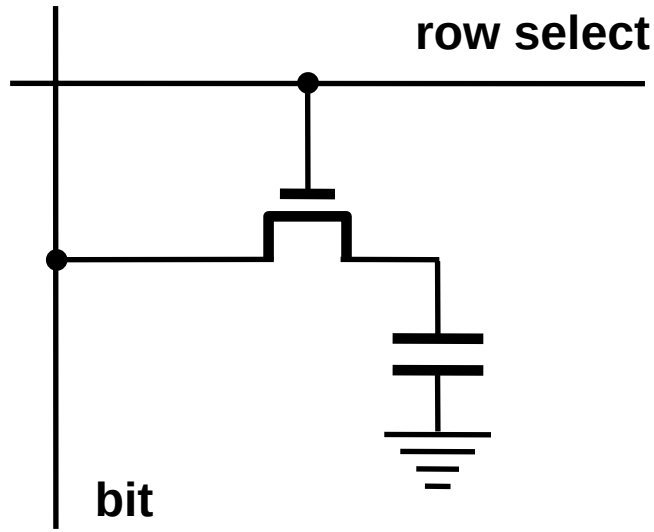
- **Why do computer designers need to know about RAM technology?**
  - Processor performance is usually limited by memory bandwidth
  - As IC densities increase, lots of memory will fit on processor chip
    - Tailor on-chip memory to specific needs
      - **Instruction cache**
      - **Data cache**
      - **Write buffer**
- **What makes RAM different from a bunch of flip-flops?**
  - Density: RAM is much denser

Main Memory Deep Background

- **“Out-of-Core”, “In-Core,” “Core Dump”?**
- **“Core memory”?**
- **Non-volatile, magnetic**
- **Lost to 4 Kbit DRAM (today using 512 Mbit DRAM)**
- **Access time 750 ns, cycle time 1500-3000 ns**

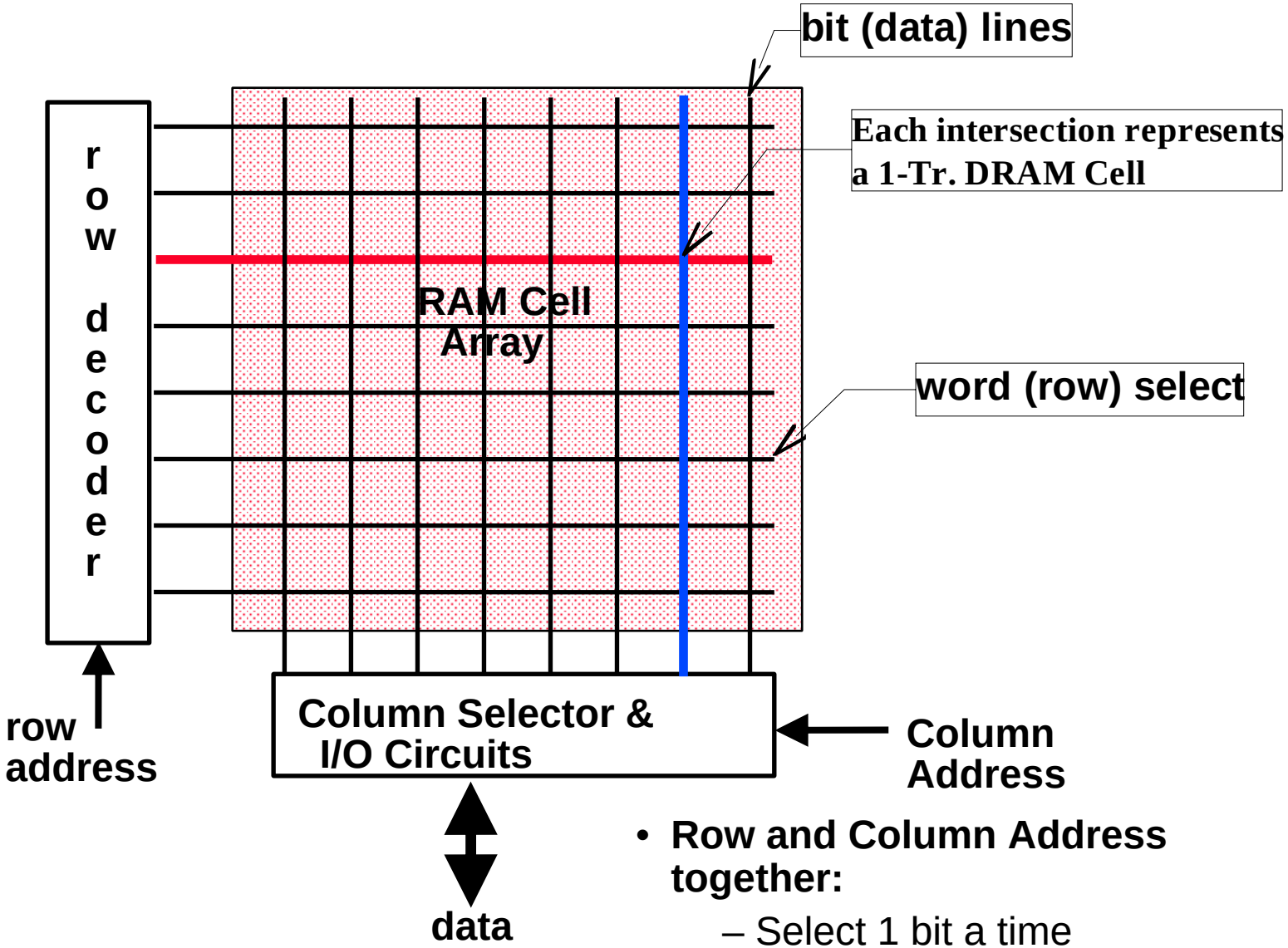


1-transistor Memory Cell (DRAM)



- **Write:**
  - 1. Drive bit line.
  - 2. Select row.
- **Read:**
  - 1. Precharge bit line to Vdd/2.
  - 2. Select row.
  - 3. Cell and bit line share charges.
    - Very small voltage changes on the bit line.
  - 4. Sense (fancy sense amp).
    - Can detect changes of ~10^6 electrons.
  - 5. Write: restore the value.
- **Refresh.**
  - 1. Just do a dummy read to every cell.

Classical DRAM Organization (Square)

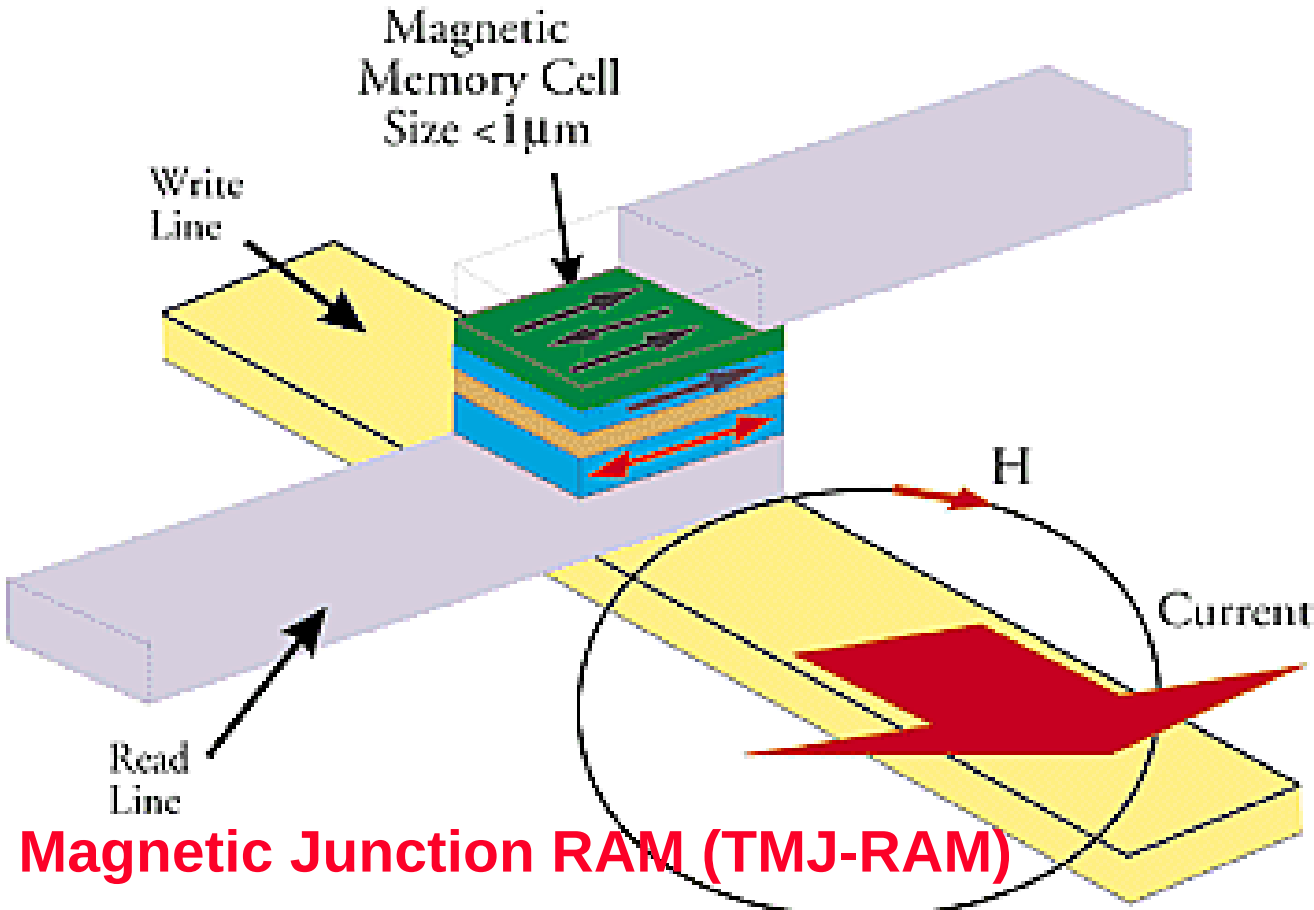


- **Row and Column Address together:**
  - Select 1 bit a time

DRAM Performance

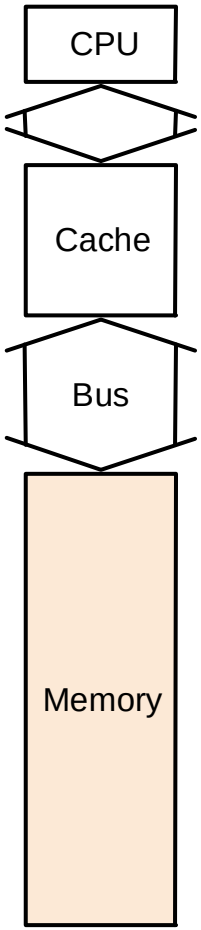
- **A 60 ns (t_{RAC}) DRAM can.**
  - perform a row access only every 110 ns (t_{RC}).
  - perform column access (t_{CAC}) in 15 ns, but time between column accesses is at least 35 ns (t_{PC}).
    - In practice, external address delays and turning around buses make it 40 to 50 ns.
- **These times do not include the time to drive the addresses off the microprocessor, nor the memory controller overhead.**
  - Drive parallel DRAMs, external memory controller, bus to turn around, SIMM module, pins…
  - 180 ns to 250 ns latency from processor to memory is good for a "60 ns" (t_{RAC}) DRAM.

Something New: Structure of Tunneling Magnetic Junction

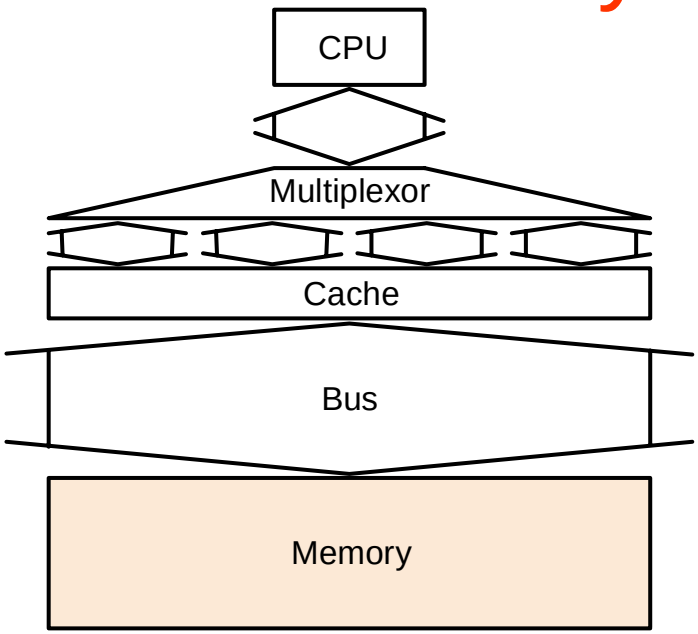


- **Tunneling Magnetic Junction RAM (TMJ-RAM)**
  - **Speed of SRAM, density of DRAM, non-volatile (no refresh)**
  - **"Spintronics": combination quantum spin and electronics**
  - **Same technology used in high-density disk-drives**

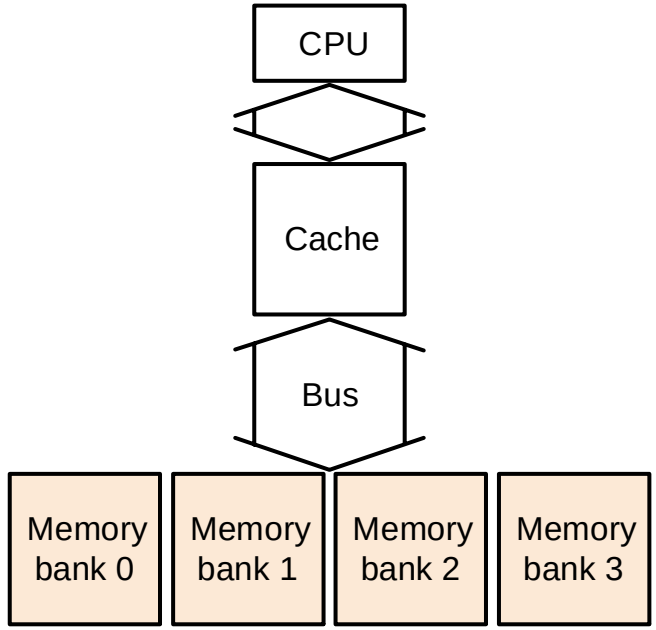
Main Memory Performance



a. One-word-wide memory organization



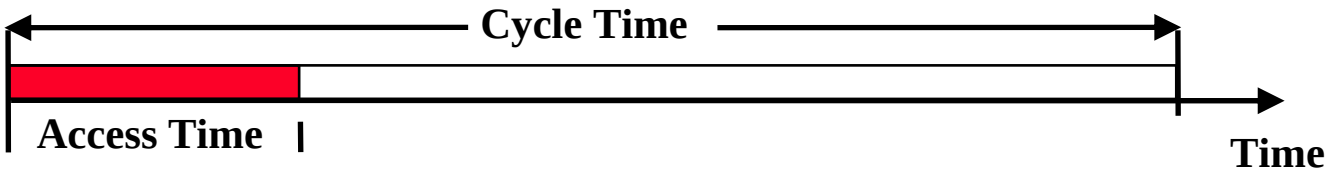
b. Wide memory organization



c. Interleaved memory organization

- ***Simple***:
  - CPU, Cache, Bus, Memory same width (32 bits)
- ***Wide***:
  - **CPU/Mux 1 word; Mux/Cache, Bus, Memory N words (Alpha: 64 bits & 256 bits)**
- ***Interleaved***:
  - **CPU, Cache, Bus 1 word: Memory N Modules (4 Modules); example is *word interleaved***

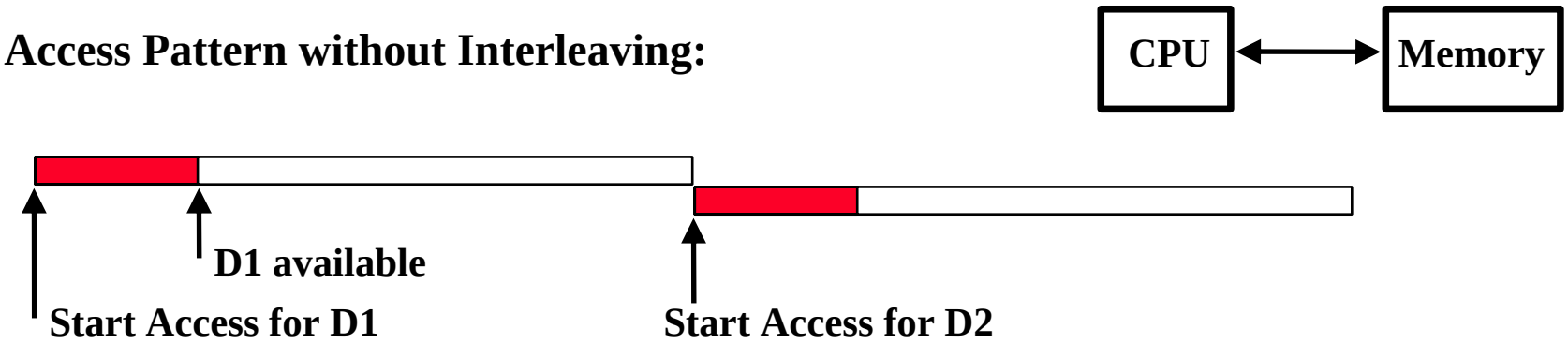
Main Memory Performance



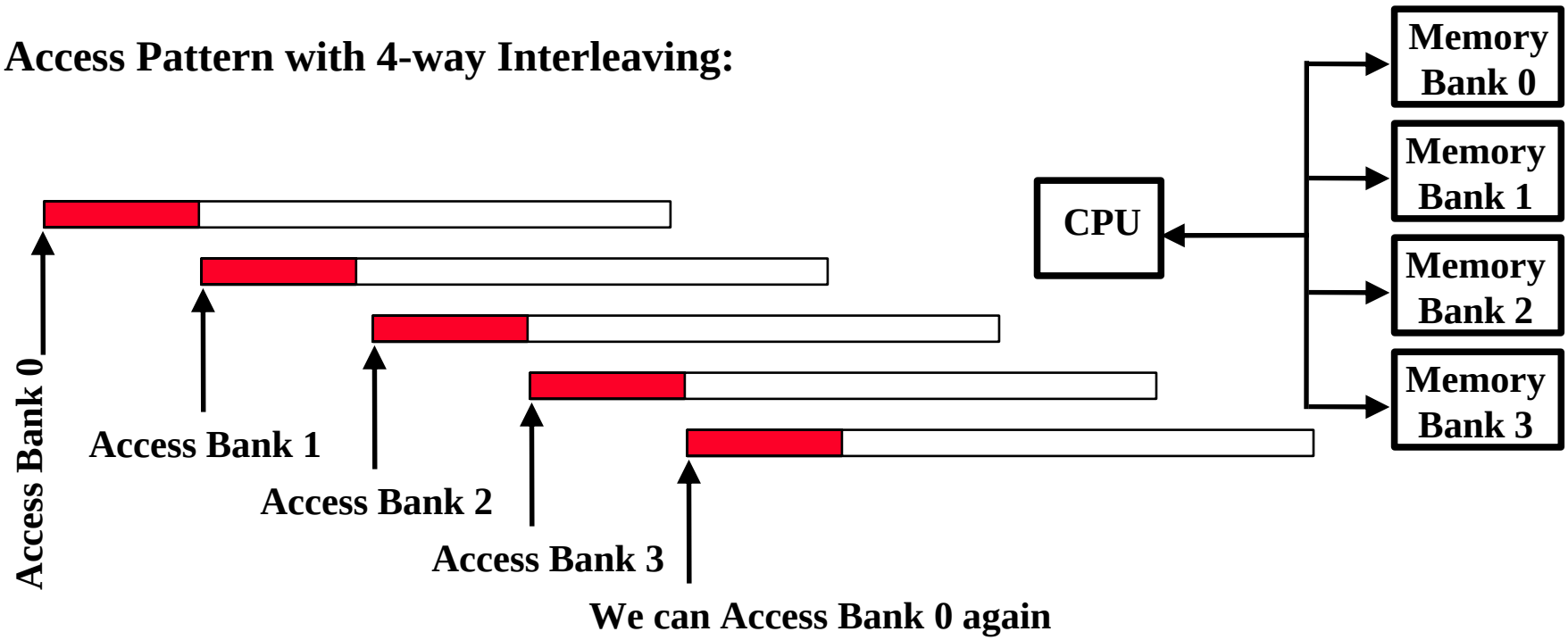
- **DRAM (Read/Write) Cycle Time >> DRAM (Read/Write) Access Time**
  - – - 2:1; why?
- **DRAM (Read/Write) Cycle Time :**
  - – How frequent can you initiate an access?
  - – Analogy: A little kid can only ask his father for money on Saturday
- **DRAM (Read/Write) Access Time:**
  - – How quickly will you get what you want once you initiate an access?
  - – Analogy: As soon as he asks, his father will give him the money
- **DRAM Bandwidth Limitation analogy:**
  - – What happens if he runs out of money on Wednesday?

Increasing Bandwidth – Interleaving

Access Pattern without Interleaving:



Access Pattern with 4-way Interleaving:



Main Memory Performance

- **Timing model**
  - 1 to send address,
  - 4 for access time, 10 cycle time, 1 to send data
  - Cache Block is 4 words
- ***Simple M.P.*** **= 4 x (1 + 10 + 1) = 48**
- ***Wide M.P.*** **= 1 + 10 + 1 = 12**
- ***Interleaved M.P.*** **= 1 + 10 + 1 + 3 = 15**