

How is the Hierarchy Managed?

- **Registers ↔ Cache**
 - 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)**

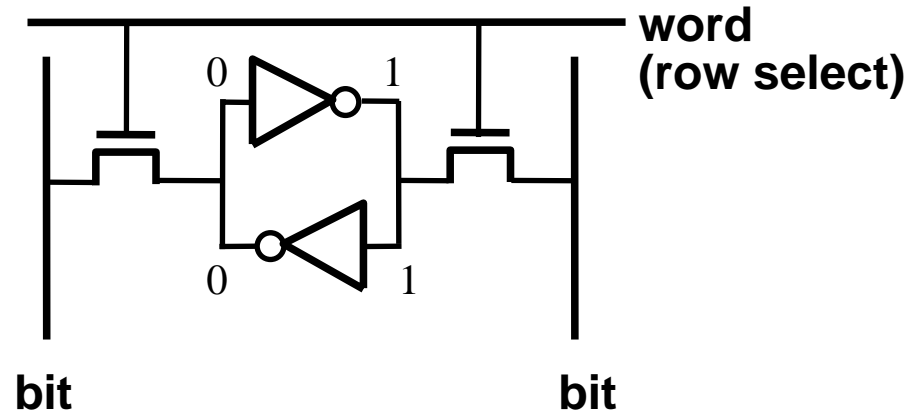
Main Memory Background

- **Performance of Main Memory:**
 - **Latency:** Cache Miss Penalty
 - *Access Time:* time between request and 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)
 - *Size:* DRAM/SRAM - *4-8*
 - *Cost/Cycle time:* SRAM/DRAM - *8-16*

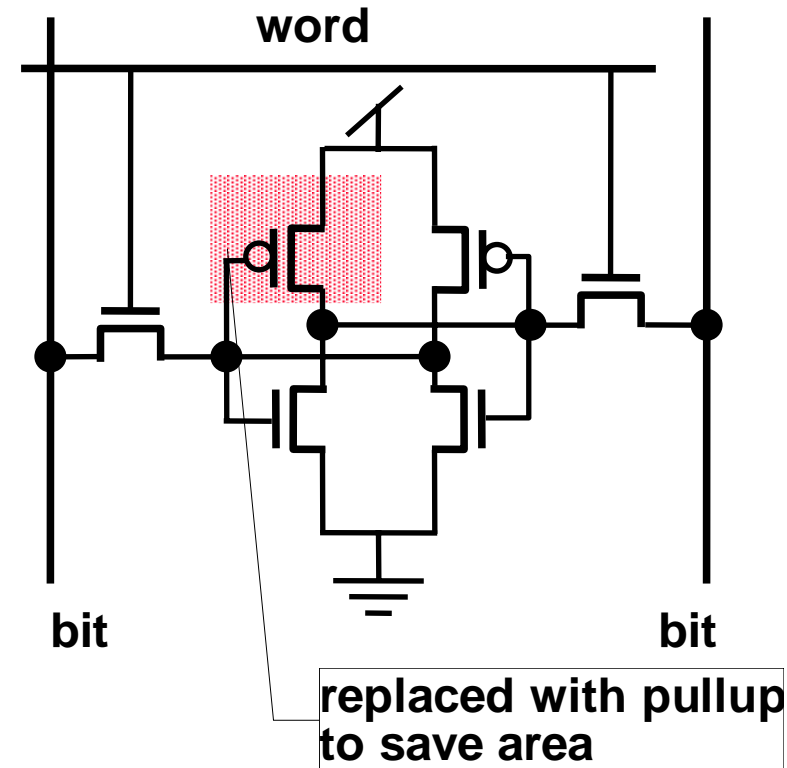
Random Access Memory (RAM) Technology

- **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

Static RAM Cell



6-Transistor SRAM Cell



- **Write:**

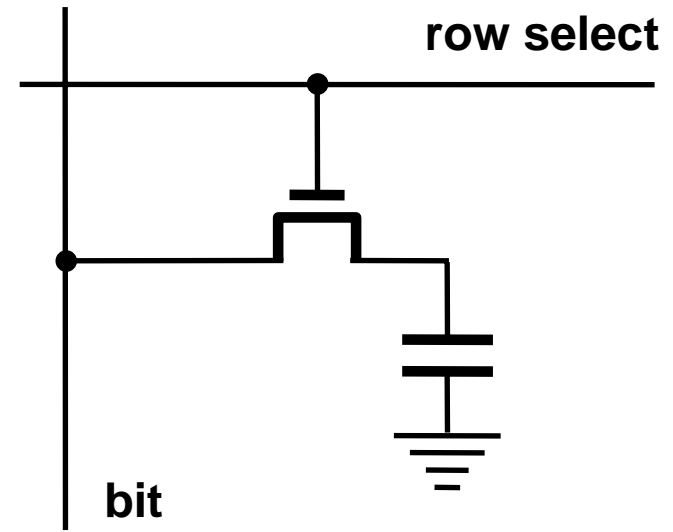
1. Drive bit lines (bit=1, bit=0)
2. Select row

- **Read:**

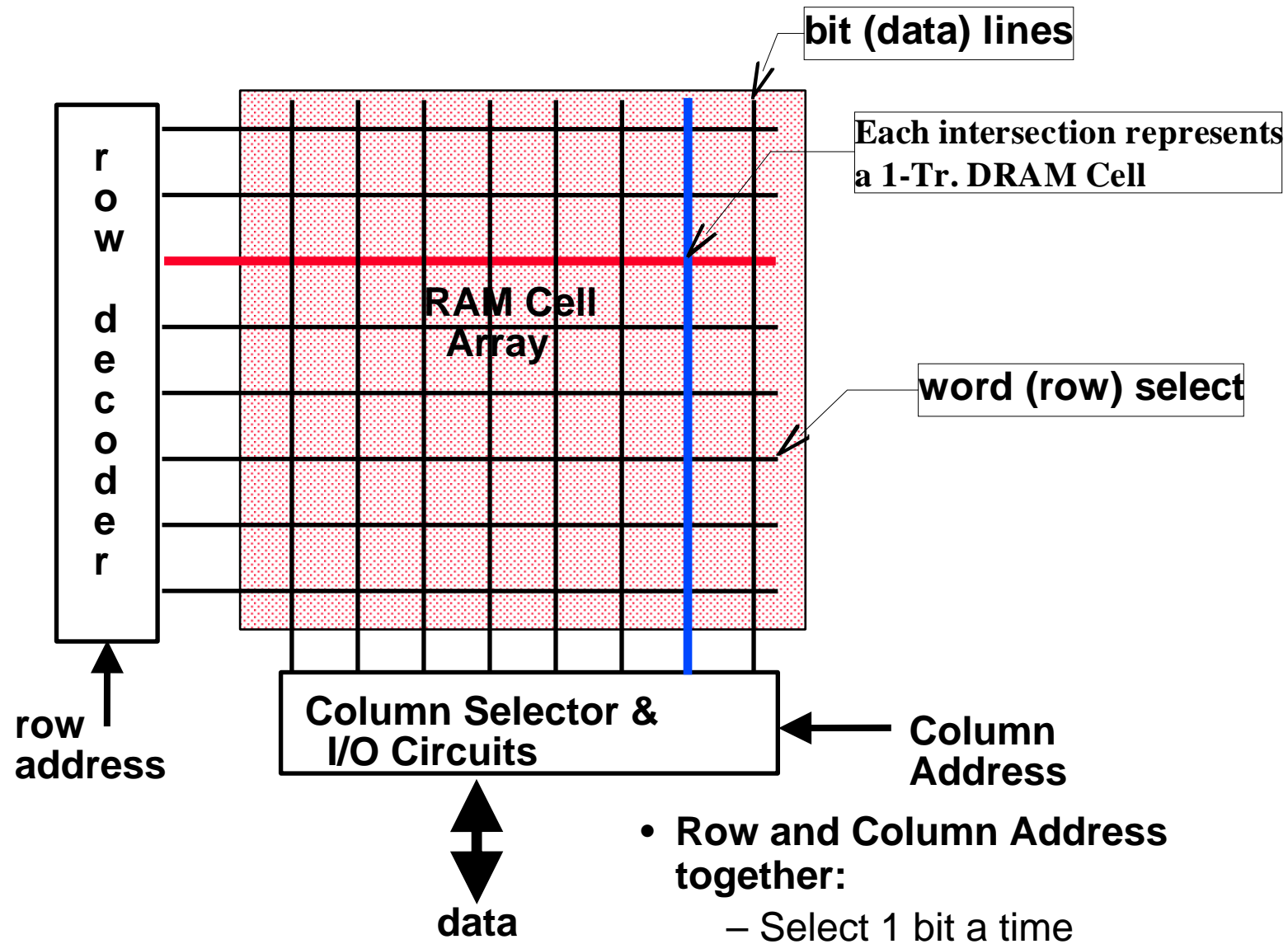
1. Precharge bit and bit to Vdd or Vdd/2 => make sure equal!
2. Select row
3. Cell pulls one line low
4. Sense amplifier on column detects difference between bit and bit

1-Transistor Memory Cell (DRAM)

- **Write:**
 - 1. Drive bit line
 - 2. Select row
- **Read:**
 - 1. Precharge bit line to $V_{dd}/2$
 - 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 $\sim 10^6$ electrons
 - 5. Write: restore the value
- **Refresh**
 - 1. Just do a dummy read to every cell.



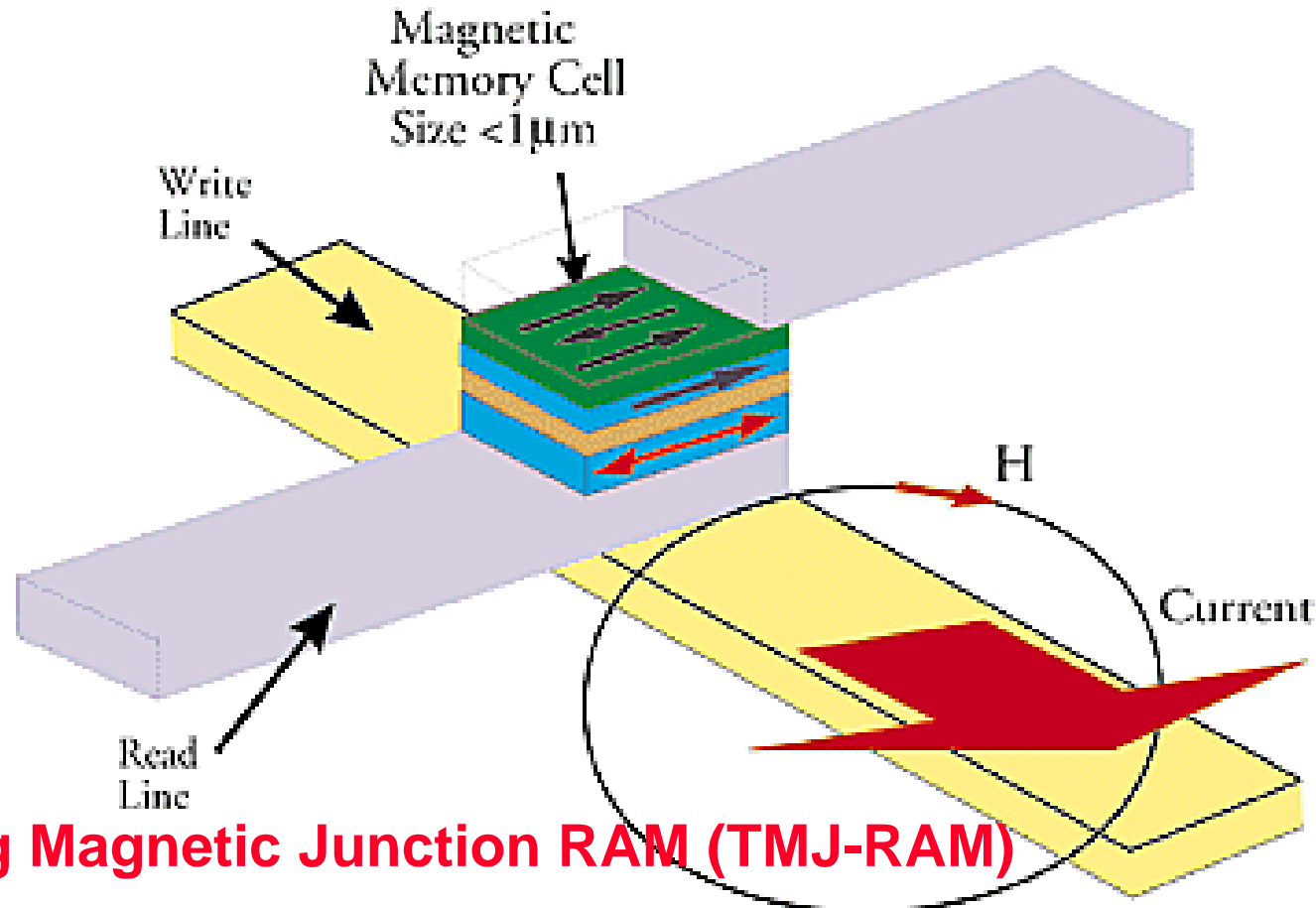
Classical DRAM Organization (square)



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

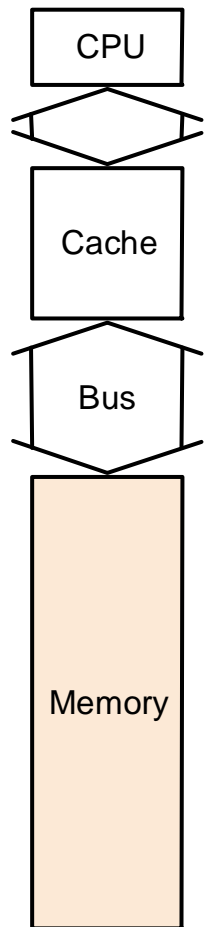
Something new: Structure of Tunneling Magnetic Junction



◦ 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

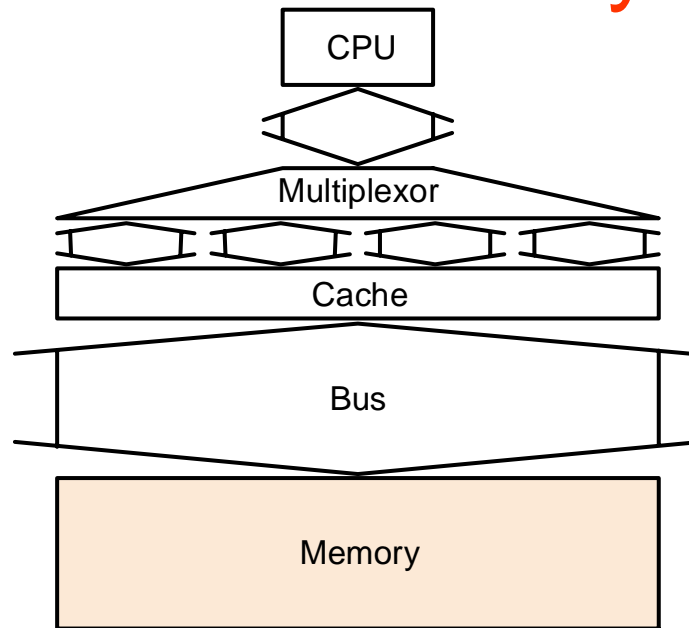
Main Memory Performance



a. One-word-wide memory organization

- **Simple:**

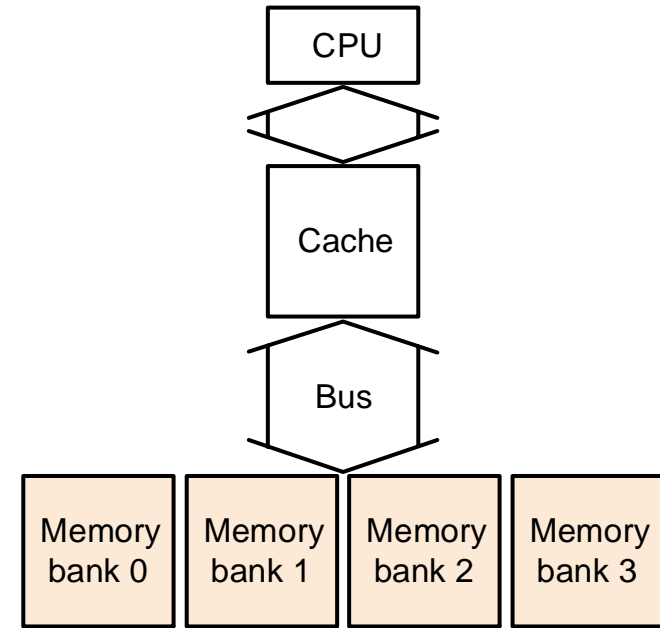
- CPU, Cache, Bus, Memory same width (32 bits)



b. Wide memory organization

- **Wide:**

- CPU/Mux 1 word; Mux/Cache, Bus, Memory N words (Alpha: 64 bits & 256 bits)

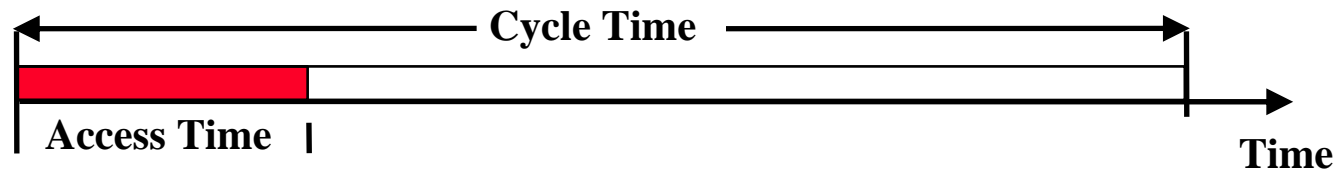


c. Interleaved memory organization

- **Interleaved:**

- CPU, Cache, Bus 1 word; Memory N Modules (4 Modules); example is *word interleaved*

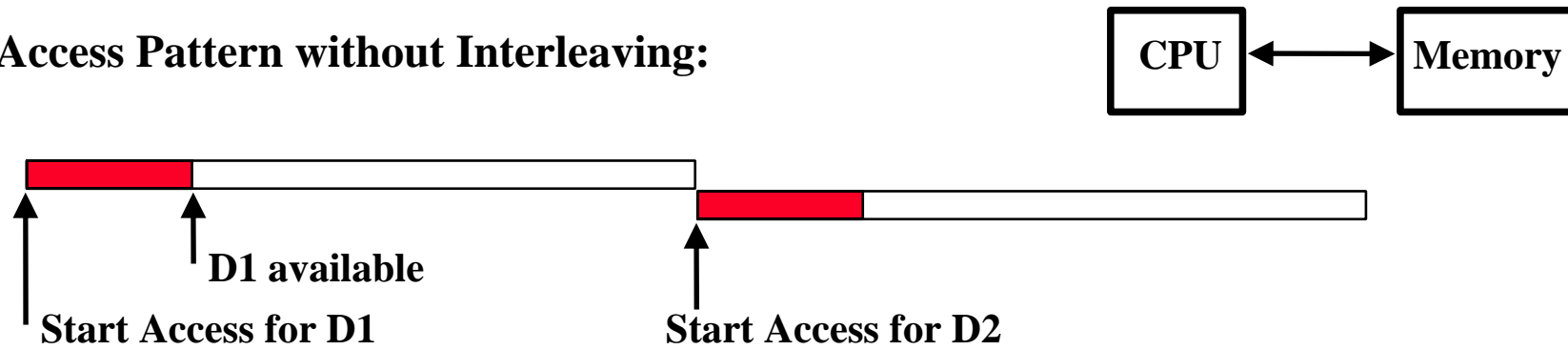
Main Memory Performance



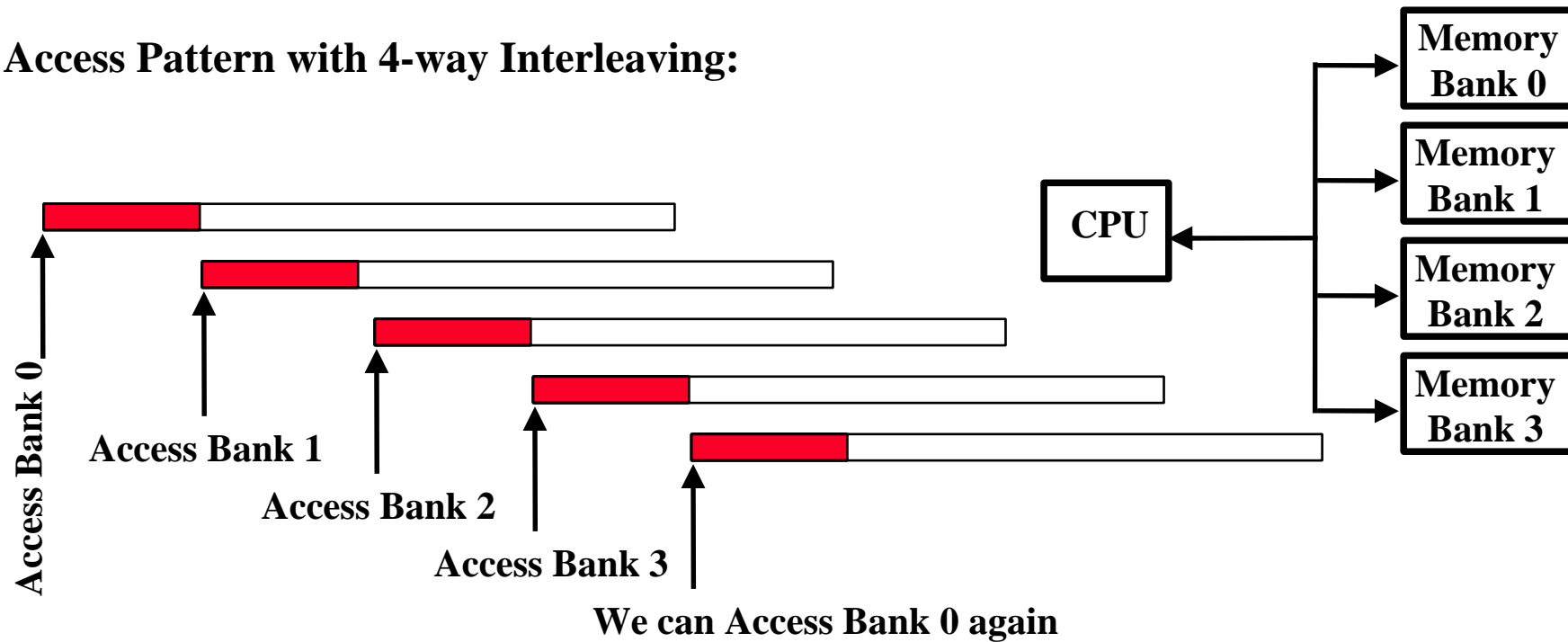
- **DRAM (Read/Write) Cycle Time \gg DRAM (Read/Write) Access Time**
 - - 2:1; why?
- **DRAM (Read/Write) Cycle Time :**
 - How frequent can you initiate an access?
 - Analogy: A little kid can only ask his father for money on Saturday
- **DRAM (Read/Write) Access Time:**
 - How quickly will you get 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.** $= 4 \times (1 + 10 + 1)$ $= 48$
- **Wide M.P.** $= 1 + 10 + 1$ $= 12$
- **Interleaved M.P.** $= 1 + 10 + 1 + 3 = 15$

