COMP3211 03S2 Lecture 10.1

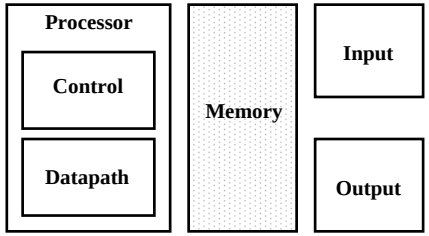
Locality and Memory Technology

Adapted from

CS152: Computer Architecture and Engineering
Dave Patterson (www.cs.berkeley.edu/~pattrsn)

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

	Capacity	Speed (latency)
Logic:	2x in 3 years	2x in 3 years
DRAM: was	4x in 3 years	2x in 10 years
now	2x in 2 years	5% per year
Disk:	4x in 3 years	2x in 10 years

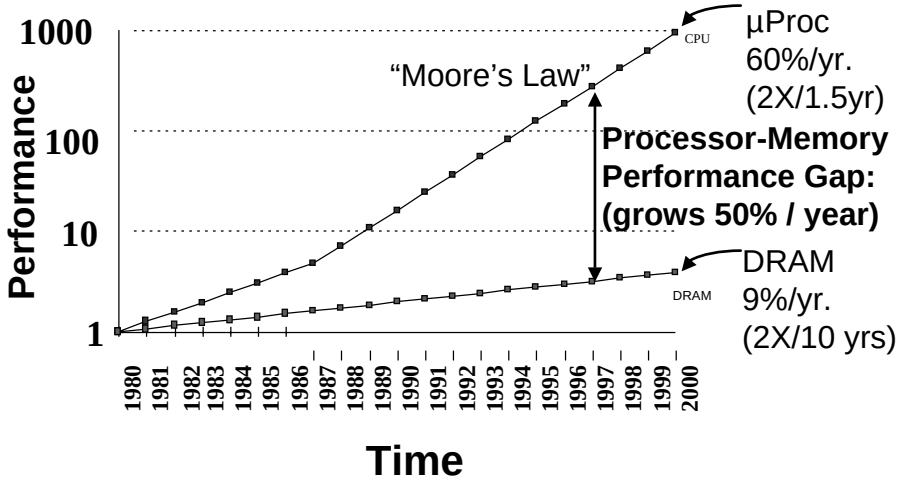
DRAM

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	120 ns
1996	64 Mb	110 ns
1998	128 Mb	100 ns
2000	256 Mb	90 ns
2002	512 Mb	80 ns

1000:1! 2:1!

Who Cares About the Memory Hierarchy?

Processor-DRAM Memory Gap (latency)



Today's Situation: Microprocessor

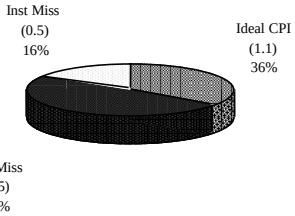
° Rely on caches to bridge gap

° Microprocessor-DRAM performance gap
  - time of a full cache miss in instructions executed

1st Alpha (7000):	340 ns/5.0 ns = 68 clks x 2 or	136 instructions
2nd Alpha (8400):	266 ns/3.3 ns = 80 clks x 4 or	320 instructions
3rd Alpha (HPDS10):	180 ns/1.7 ns =108 clks x 6 or	648 instructions

Impact on Performance

° Suppose a processor executes at
  - Clock Rate = 200 MHz (5 ns per cycle)
  - CPI = 1.1
  - 50% arith/logic, 30% ld/st, 20% control



° Suppose that 10% of memory operations get 50 cycle miss penalty

° CPI = ideal CPI + average stalls per instruction
= 1.1(cyc) +(0.30 (datamops/ins)
x 0.10 (miss/datamop) x 50 (cycle/miss))
= 1.1 cycle + 1.5 cycle
= 2. 6

° 58 % of the time the processor is stalled waiting for memory!

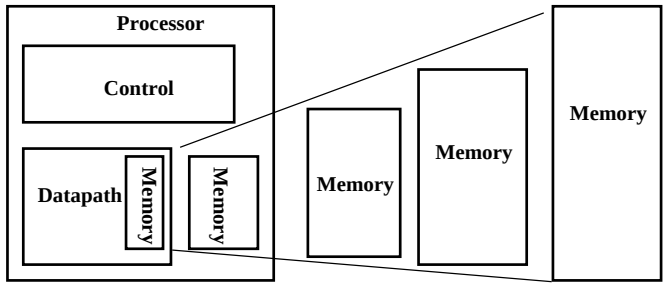
° a 1% instruction miss rate would add an additional 0.5 cycles to the CPI!

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

An Expanded View of the Memory System



Why hierarchy works

° **The Principle of Locality:**

- **Program access a relatively small portion of the address space at any instant of time.**

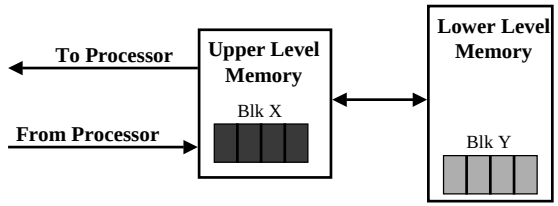
Memory Hierarchy: How Does it Work?

° **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



Memory Hierarchy: Terminology

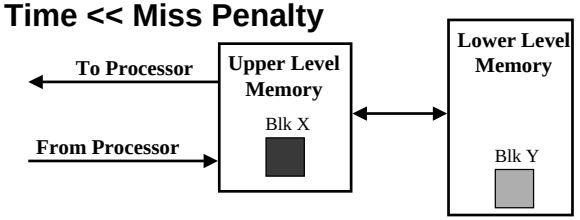
° **Hit: data appears in some block in the upper level (example: Block X)**

- **Hit Rate: the fraction of memory accesses 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 retrieved from a block in the lower level (Block Y)**

- **Miss Rate = 1 - (Hit Rate)**
- **Miss Penalty: Time to replace a block in the upper level +**
  Time to deliver the block the processor

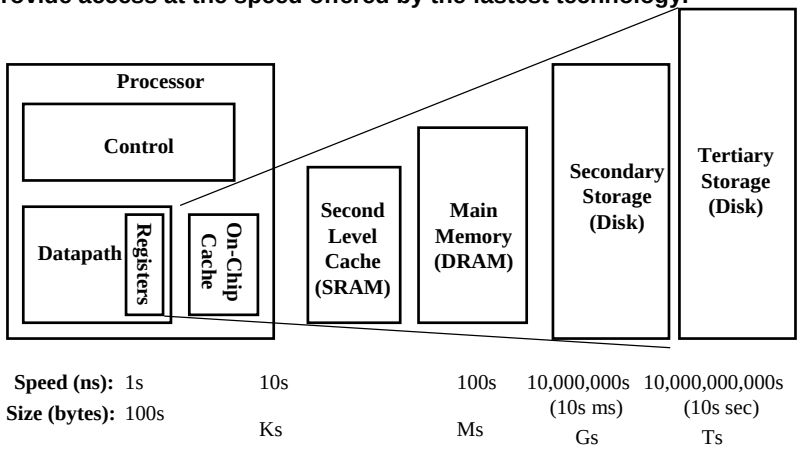
° **Hit Time << Miss Penalty**



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.**



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

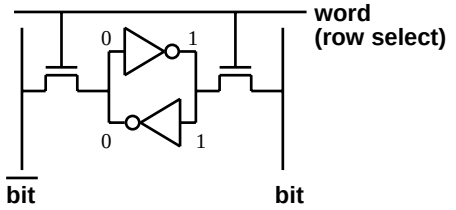
- Random Access:
  - "Random" is good: access time is the same for all locations
  - DRAM: Dynamic Random Access Memory 4 – 8x capacity
    - High density, low power, cheap, slow of SRAM
    - Dynamic: need to be "refreshed" regularly
  - SRAM: Static Random Access Memory access times 8 – 16x
    - Low density, high power, expensive, fast faster than DRAM
    - Static: content will last "forever"(until lose power)
- "Not-so-random" Access Technology:
  - Access time varies from location to location and from time to time
  - Examples: Disk, CDROM
- Sequential Access Technology: access time linear in location (e.g.,Tape)
- The next few lectures will concentrate on random access technology
  - The Main Memory: DRAMs + Caches: SRAMs
  - Disk technology

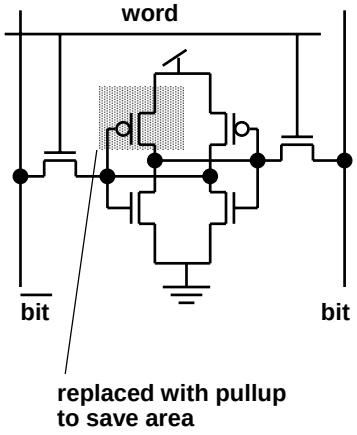
Random Access Memory (RAM) Technology

- 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 more denser

Static RAM Cell

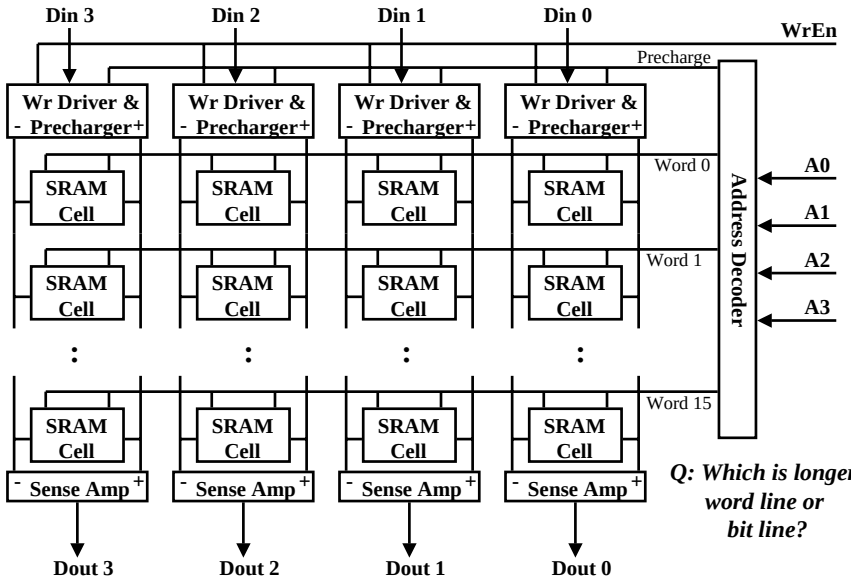
6-Transistor SRAM Cell





- Write:
  1. Drive bit lines (bit=1, $\overline{\text{bit}}$=0)
  2. Select row
- Read:
  1. Precharge bit and $\overline{\text{bit}}$ to Vdd
  2. Select row
  3. Cell pulls one line low
  4. Sense amp on column detects difference between bit and $\overline{\text{bit}}$

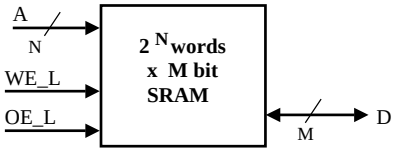
Typical SRAM Organization: 16-word x 4-bit



Q: Which is longer: word line or bit line?

Read occurs by default whenever a change in address is sensed

Logic Diagram of a Typical SRAM

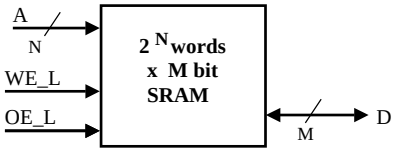


° **Write Enable is usually active low (WE_L)**

° **Din and Dout are combined to save pins:**

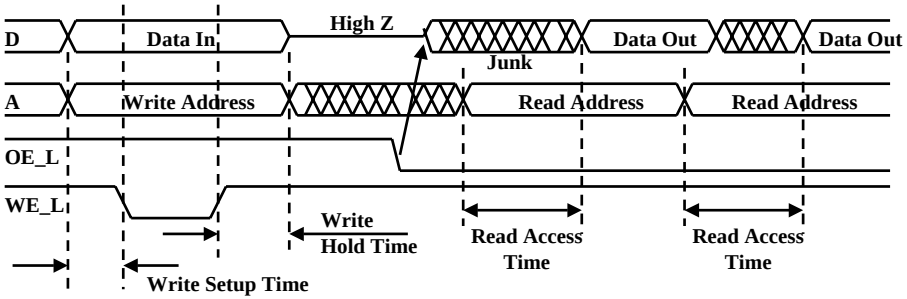
- **A new control signal, output enable (OE_L) is needed**
- **WE_L is asserted (Low), OE_L is deasserted (High)**
  - **D serves as the data input pin**
- **WE_L is deasserted (High), OE_L is asserted (Low)**
  - **D is the data output pin**
- **Both WE_L and OE_L are asserted:**
  - **Result is unknown. Don't do that!!!**

Typical SRAM Timing

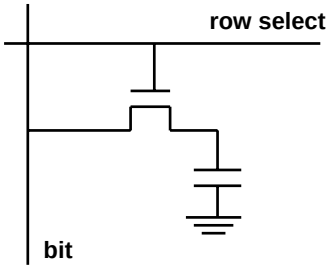


Write Timing:

Read Timing:



1-Transistor Memory Cell (DRAM)



° **Write:**

- **1. Drive bit line**
- **2.. Select row**

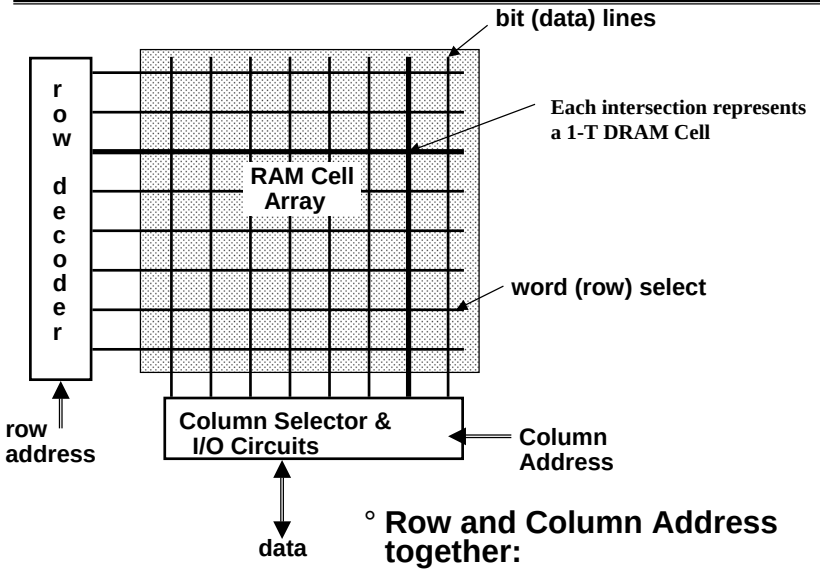
° **Read:**

- **1. Precharge bit line to Vdd**
- **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 ~1 million electrons**
- **5. Write: restore the value**

° **Refresh**

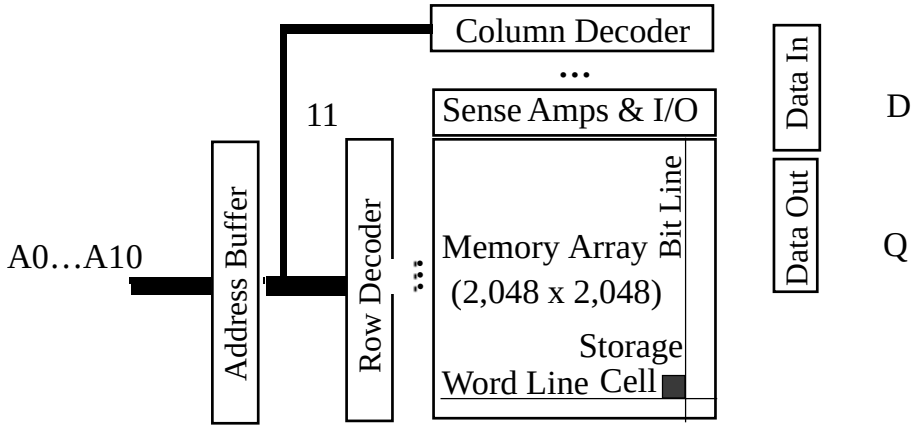
- **1. Just do a dummy read to every cell.**

Classical DRAM Organization (square)



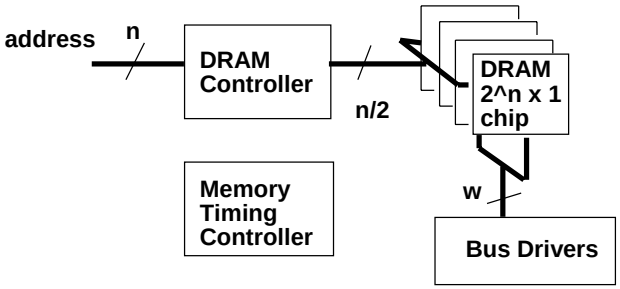
° **Row and Column Address together:**
- **Select 1 bit a time**

DRAM logical organization (4 Mbit)



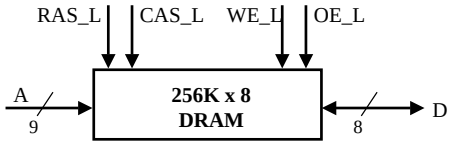
° **Square root of bits per RAS/CAS**

Memory Systems



Tc = Tcycle + Tcontroller + Tdriver

Logic Diagram of a Typical DRAM



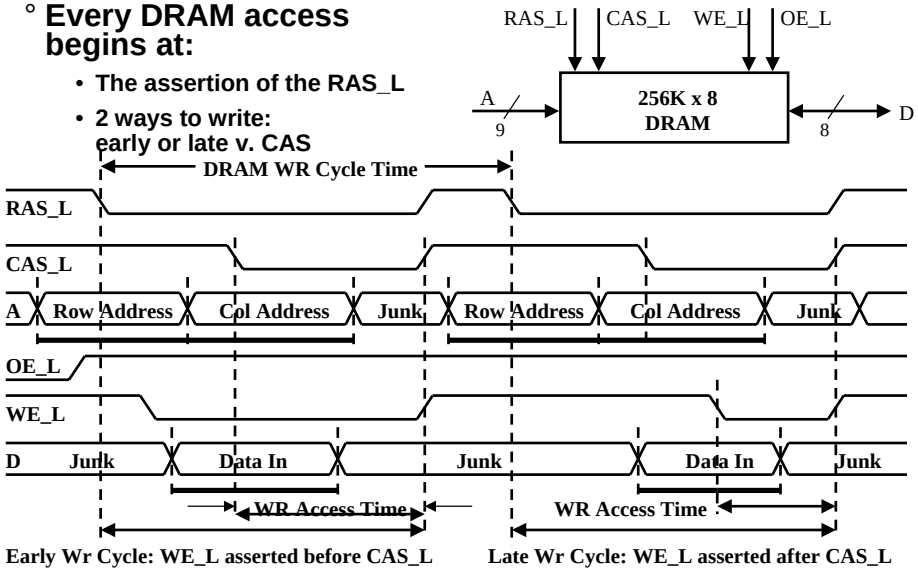
° **Control Signals (RAS_L, CAS_L, WE_L, OE_L) are all active low**

° **Din and Dout are combined (D):**
- **WE_L is asserted (Low), OE_L is deasserted (High)**
  - **D serves as the data input pin**
- **WE_L is deasserted (High), OE_L is asserted (Low)**
  - **D is the data output pin**

° **Row and column addresses share the same pins (A)**
- **RAS_L goes low: Pins A are latched in as row address**
- **CAS_L goes low: Pins A are latched in as column address**
- **RAS/CAS edge-sensitive**

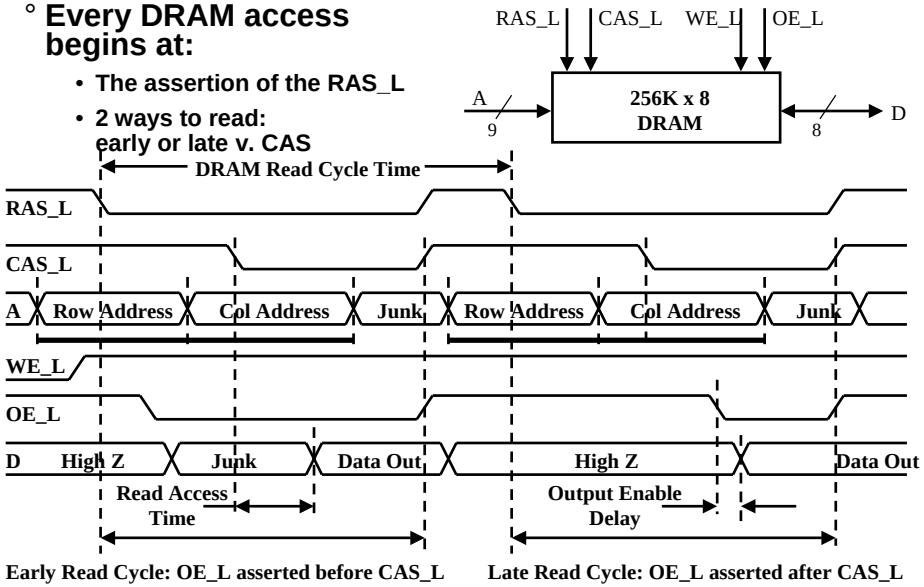
DRAM Write Timing

° **Every DRAM access begins at:**
- **The assertion of the RAS_L**
- **2 ways to write: early or late v. CAS**

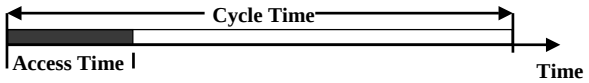


DRAM Read Timing

° **Every DRAM access begins at:**
- **The assertion of the RAS_L**
- **2 ways to read: early or late v. CAS**



Cycle Time versus Access Time



° **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**

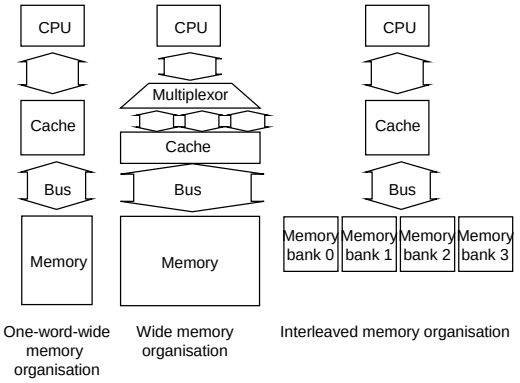
° **DRAM Bandwidth Limitation analogy:**
- **What happens if he runs out of money on Wednesday?**

Main Memory Performance

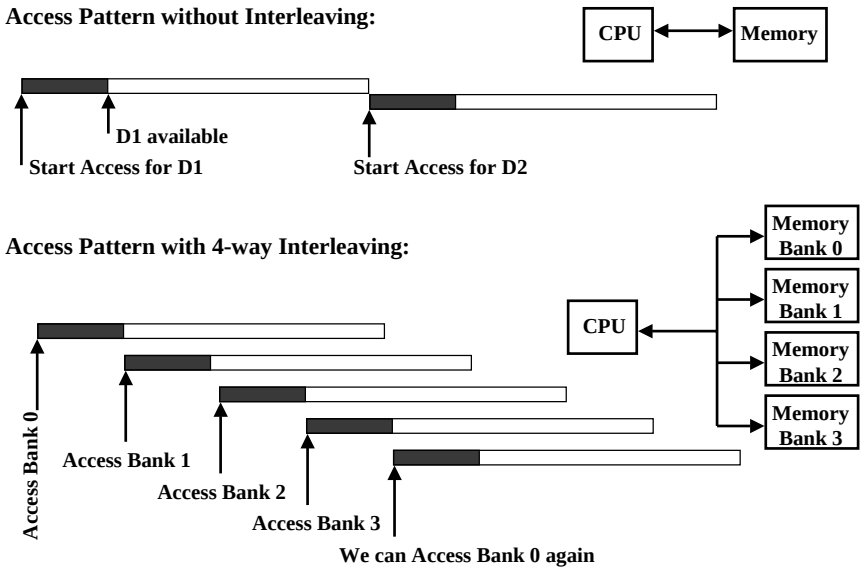
° ***Simple*:**
- **CPU, Cache, Bus, Memory same width (32 bits)**

° ***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***



Increasing Bandwidth - Interleaving



Main Memory Performance

° **Timing model**
- 1 to send address,
- 6 access time, 1 to send data
- Cache Block is 4 words

° ***Simple M.P.*** **= 4 x (1+6+1) = 32**

° ***Wide M.P.*** **= 1 + 6 + 1 = 8**

° ***Interleaved M.P.*** **= 1 + 6 + 4x1 = 11**

Address	Bank 0	Address	Bank 1	Address	Bank 2	Address	Bank 3
0		1		2		3	
4		5		6		7	
8		9		10		11	
12		13		14		15	

Independent Memory Banks

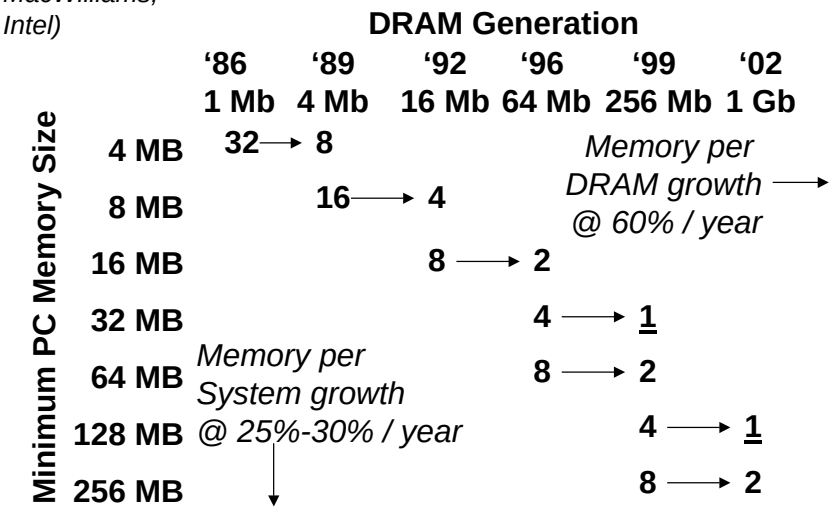
° **How many banks?**

number banks = number clocks to access word in bank
- **For sequential accesses, otherwise will return to original bank before it has next word ready**

° **Increasing DRAM size => fewer chips => harder to have banks**
- **Growth bits/chip DRAM : 50%-60%/yr**

Fewer DRAMs/System over Time

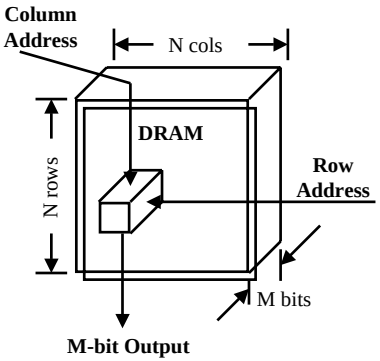
(from Pete MacWilliams, Intel)

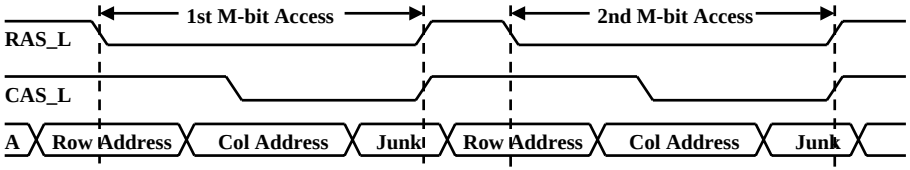


Page Mode DRAM: Motivation

° **Regular DRAM Organization:**
- N rows x N column x M-bit
- Read & Write M-bit at a time
- Each M-bit access requires a RAS / CAS cycle

° **Fast Page Mode DRAM**
- N x M "register" to save a row

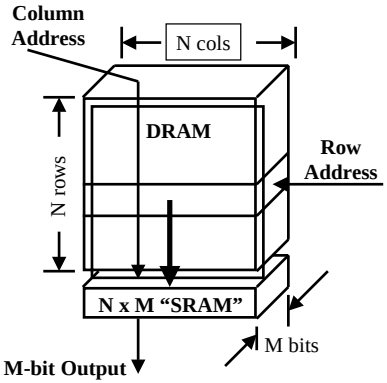


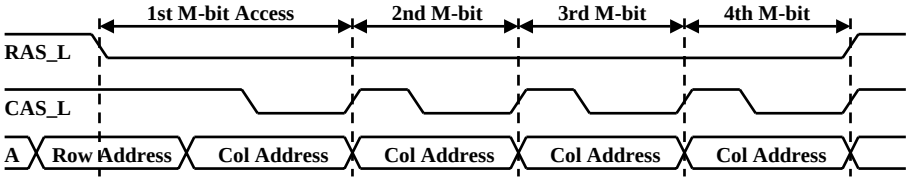


Fast Page Mode Operation

° **Fast Page Mode DRAM**
- N x M "SRAM" to save a row

° **After a row is read into the register**
- Only CAS is needed to access other M-bit blocks on that row
- RAS_L remains asserted while CAS_L is toggled





SDRAM & DDR SDRAM

° **These enhancements are intended to improve the bandwidth to DRAM**

° **S(ynchronous)DRAM replaces the asynchronous handshaked interface between memory controller & DRAM – repeated transfers thus avoid synchronisation overheads**
- Use of a byte counter allows multiple bytes to be transferred without further requests

° **DDR = Double Data Rate allows transfers to occur on both the rising and the falling edge of the clock thus effectively doubling bandwidth**

° **Memory is packaged in so-called Dual In-line Memory Modules, which provide 8-byte data transfers – a 150 MHz DDR SDRAM DIMM (denoted PC2400) thus has a bandwidth of 2400MB/s**

Summary

° **Two Different Types of Locality:**
- Temporal Locality (Locality in Time): If an item is referenced, it will tend to be referenced again soon.
- Spatial Locality (Locality in Space): If an item is referenced, items whose addresses are close by tend to be referenced soon.

° **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.

° **DRAM is slow but cheap and dense:**
- Good choice for presenting the user with a BIG memory system

° **SRAM is fast but expensive and not very dense:**
- Good choice for providing the user FAST access time.