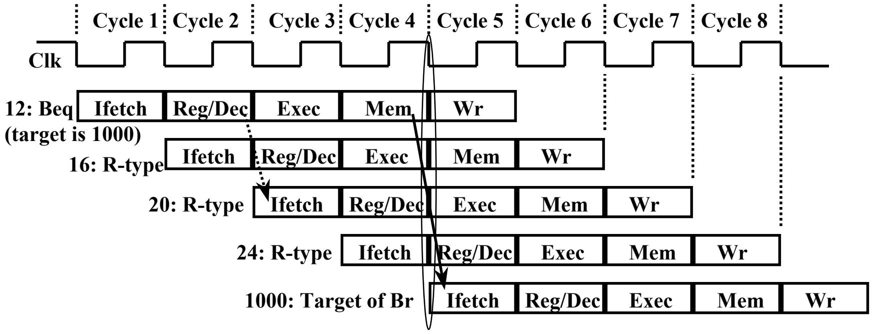
361
Computer Architecture
Lecture 16: Memory Systems

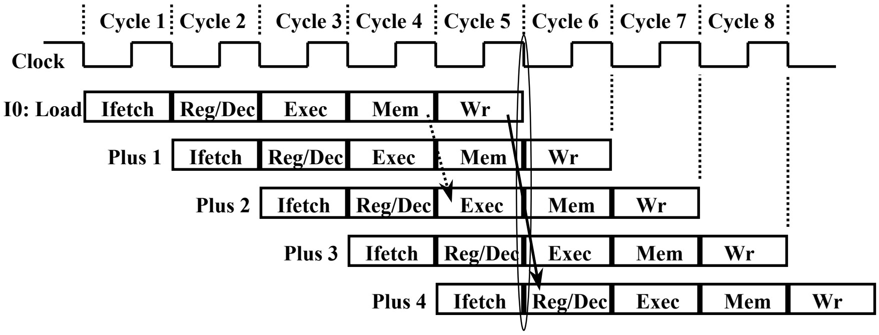
Recap: Solution to Branch Hazard



° In the Simple Pipeline Processor if a Beq is fetched during Cycle 1:
- Target address is NOT written into the PC until the end of Cycle 4
- Branch's target is NOT fetched until Cycle 5
- 3-instruction delay before the branch take effect

° This Branch Hazard can be reduced to 1 instruction if in Beq's Reg/Dec:
- Calculate the target address
- Compare the registers using some "quick compare" logic

Recap: Solution to Load Hazard



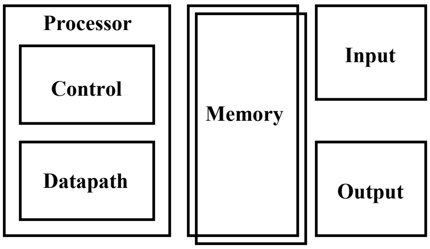
- ° In the Simple Pipeline Processor if a Load is fetched during Cycle 1:
  - The data is NOT written into the Reg File until the end of Cycle 5
  - We cannot read this value from the Reg File until Cycle 6
  - 3-instruction delay before the load take effect
- ° This Data Hazard can be reduced to 1 instruction if we:
  - Forward the data from the pipeline register to the next instruction

Outline of Today's Lecture

- ° Recap and Introduction
- ° Memory System: the BIG Picture?
- ° Questions and Administrative Matters
- ° Memory Technology: SRAM
- ° Memory Technology: DRAM
- ° A Real Life Example: SPARCstation 20's Memory System
- ° Summary

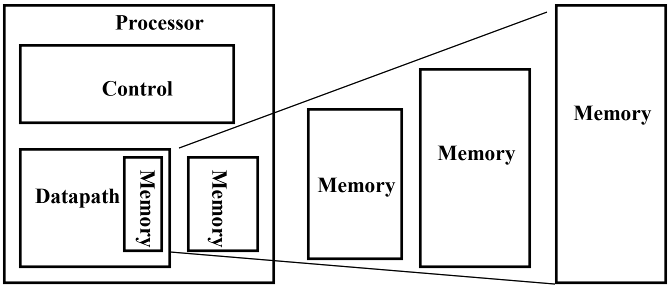
The Big Picture: Where are We Now?

° The Five Classic Components of a Computer



° Today’s Topic: Memory System

An Expanded View of the Memory System



Speed:	Fastest	Slowest
Size:	Smallest	Biggest
Cost:	Highest	Lowest

The Principle of Locality

° The Principle of Locality:

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

° 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.

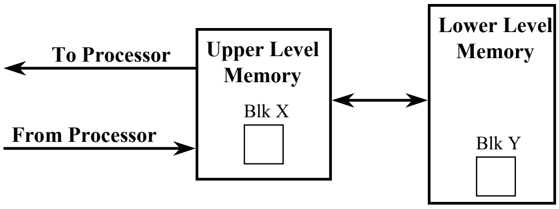
Memory Hierarchy: Principles of Operation

° At any given time, data is copied between only 2 adjacent levels:

- Upper Level: the one closer to the processor
  - Smaller, faster, and uses more expensive technology
- Lower Level: the one further away from the processor
  - Bigger, slower, and uses less expensive technology

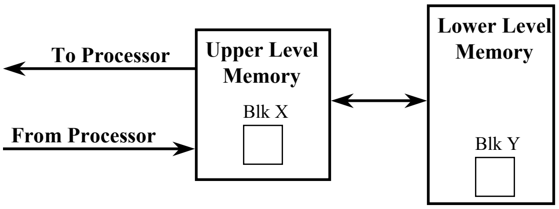
° Block:

- The minimum unit of information that can either be present or not present in the two level hierarchy



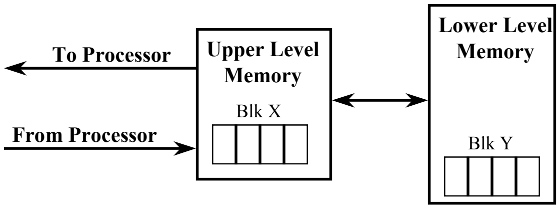
Memory Hierarchy: Terminology

- ° 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 to replace a block in the upper level +
    Time to deliver the block the processor
- ° Hit Time << Miss Penalty



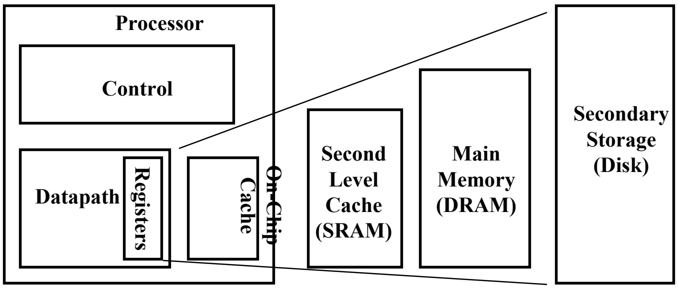
Memory Hierarchy: How Does it Work?

- ° Temporal Locality (Locality in Time): If an item is referenced, it will tend to be referenced again soon.
  - Keep more recently accessed data items closer to the processor
- ° Spatial Locality (Locality in Space): If an item is referenced, items whose addresses are close by tend to be referenced soon.
  - Move blocks consists of contiguous words to the upper levels



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.



Speed (ns):	1s	10s	100s	10,000,000s (10s ms)
Size (bytes):	100s	Ks	Ms	Gs

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"

° "Non-so-random" Access Technology:
  - Access time varies from location to location and from time to time
  - Examples: Disk, tape drive, CDROM

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

Technology Trends

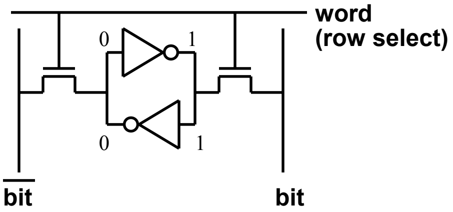
	Capacity	Speed
Logic:	2x in 3 years	2x in 3 years
DRAM:	4x in 3 years	1.4x in 10 years
Disk:	2x in 3 years	1.4x in 10 years

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	145 ns
1995	64 Mb	120 ns

Static RAM Cell

6-Transistor SRAM Cell



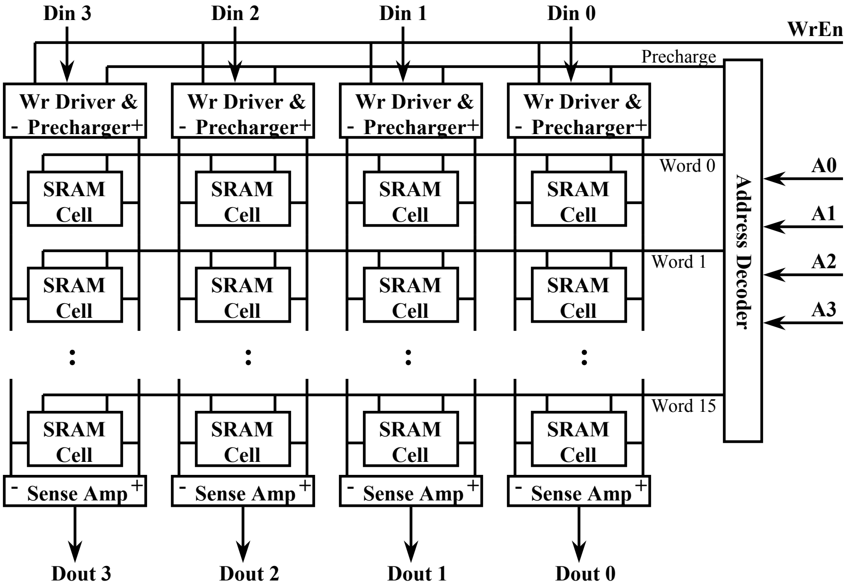
° Write:

1. Drive bit lines
2.. Select row

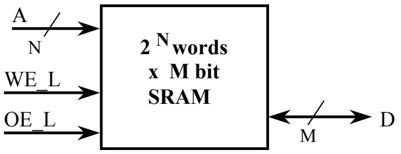
° Read:

1. Precharge bit and bit’ to Vdd
2.. Select row
3. Cell pulls one line low
4. Sense amp on column detects difference

Typical SRAM Organization: 16-word x 4-bit

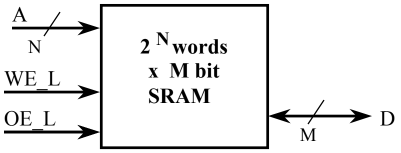


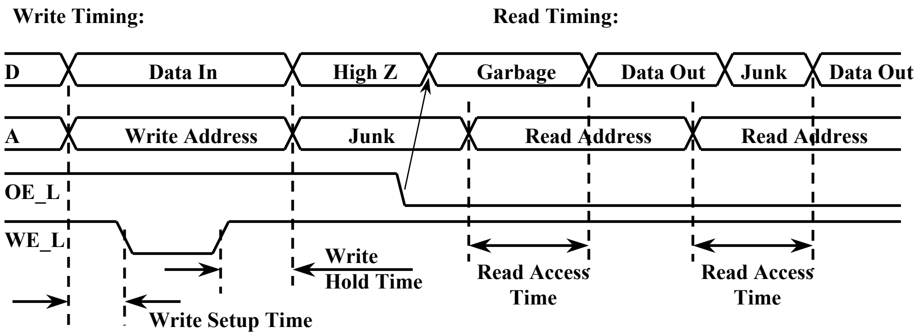
Logic Diagram of a Typical SRAM



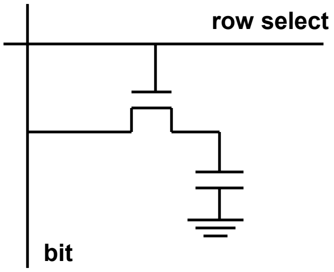
- ° Write Enable is usually active low (WE_L)
- ° Din and Dout are combined:
  - A new control signal, output enable (OE_L) is needed
  - WE_L is asserted (Low), OE_L is disasserted (High)
    - D serves as the data input pin
  - WE_L is disasserted (High), OE_L is asserted (Low)
    - D is the data output pin
  - Both WE_L and OE_L are asserted:
    - Result is unknown. Don't do that!!!

Typical SRAM Timing





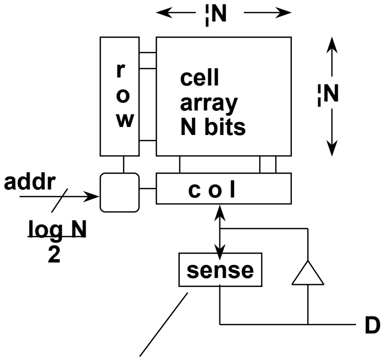
1-Transistor Cell



- Write:
  - 1. Drive bit line
  - 2.. Select row
- Read:
  - 1. Precharge bit line to Vdd
  - 2.. Select row
  - 3. Sense (fancy sense amp)
    - Can detect changes of ~1 million electrons
  - 4. Write: restore the value
- Refresh
  - 1. Just do a dummy read to every cell.

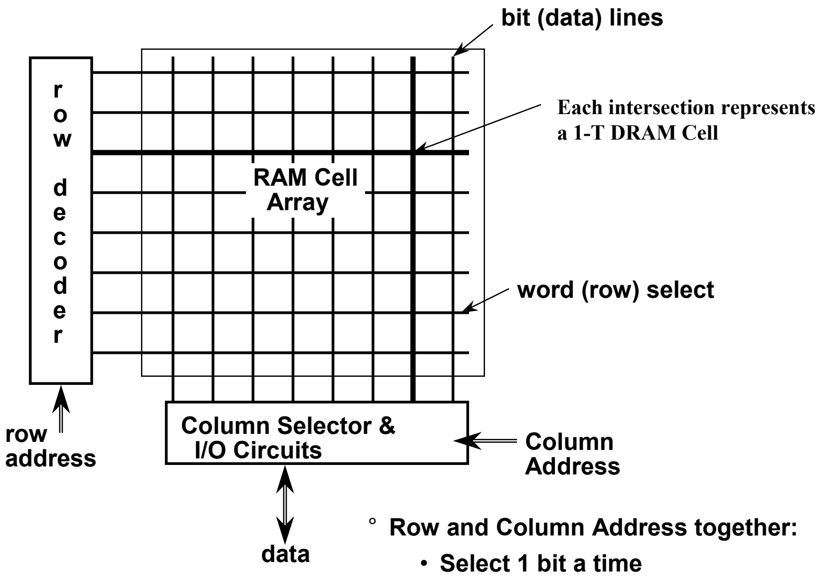
Introduction to DRAM

- Dynamic RAM (DRAM):
  - Refresh required
  - Very high density
  - Low power (.1 - .5 W active, .25 - 10 mW standby)
  - Low cost per bit
  - Pin sensitive:
    - Output Enable (OE_L)
    - Write Enable (WE_L)
    - Row address strobe (ras)
    - Col address strobe (cas)
  - Page mode operation



one sense amp
less pwr,
less area

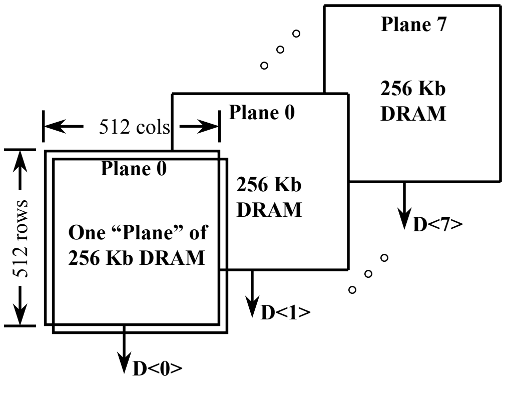
Classical DRAM Organization



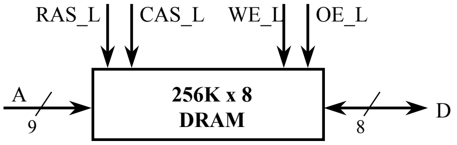
- Row and Column Address together:
  - Select 1 bit a time

Typical DRAM Organization

- Typical DRAMs: access multiple bits in parallel
  - Example: 2 Mb DRAM = 256K x 8 = 512 rows x 512 cols x 8 bits
  - Row and column addresses are applied to all 8 planes in parallel



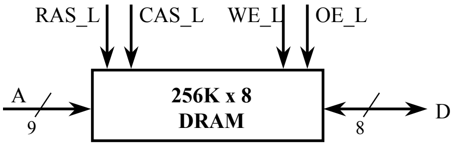
Logic Diagram of a Typical DRAM

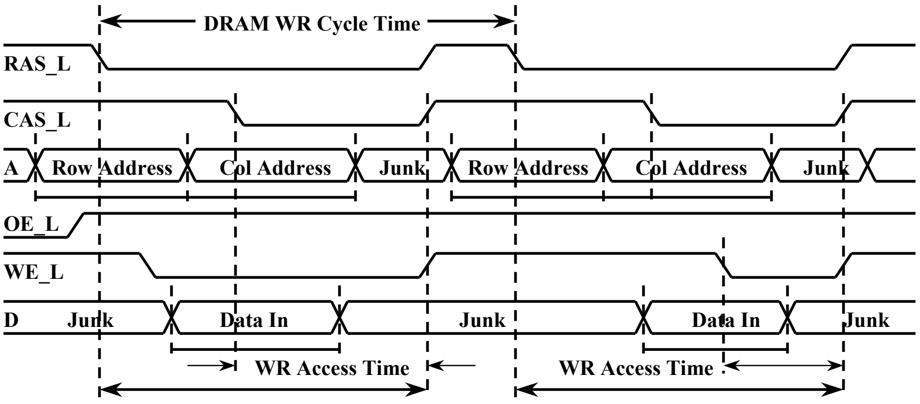


- Control Signals (RAS_L, CAS_L, WE_L, OE_L) are all active low
- Din and Dout are combined (D):
  - WE_L is asserted (Low), OE_L is disasserted (High)
    - D serves as the data input pin
  - WE_L is disasserted (High), OE_L is asserted (Low)
    - D is the data output pin
- Row and column addresses 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

DRAM Write Timing

- Every DRAM access begins at:
  - The assertion of the RAS_L



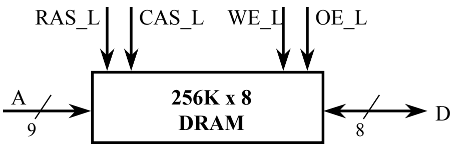


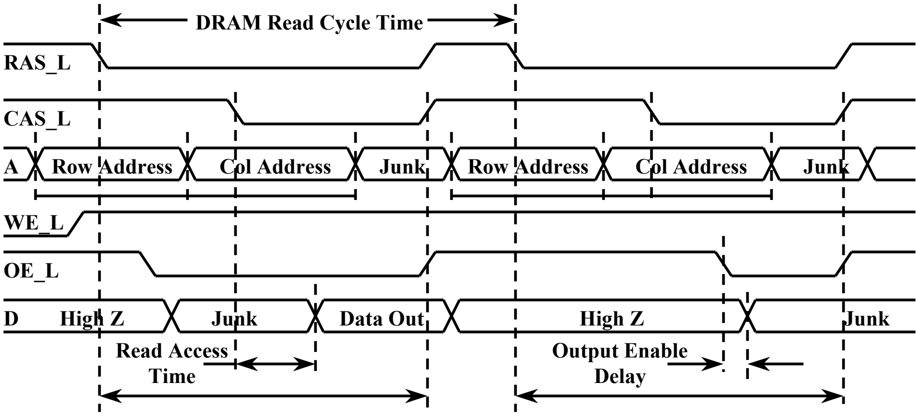
Early Wr Cycle: WE_L asserted before CAS_L

Late Wr Cycle: WE_L asserted after CAS_L

DRAM Read Timing

° Every DRAM access begins at:
- The assertion of the RAS_L

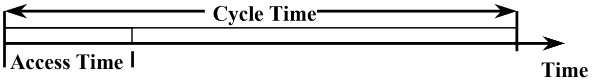




Early Read Cycle: OE_L asserted before CAS_L

Late Read Cycle: OE_L asserted after CAS_L

Cycle Time versus Access Time



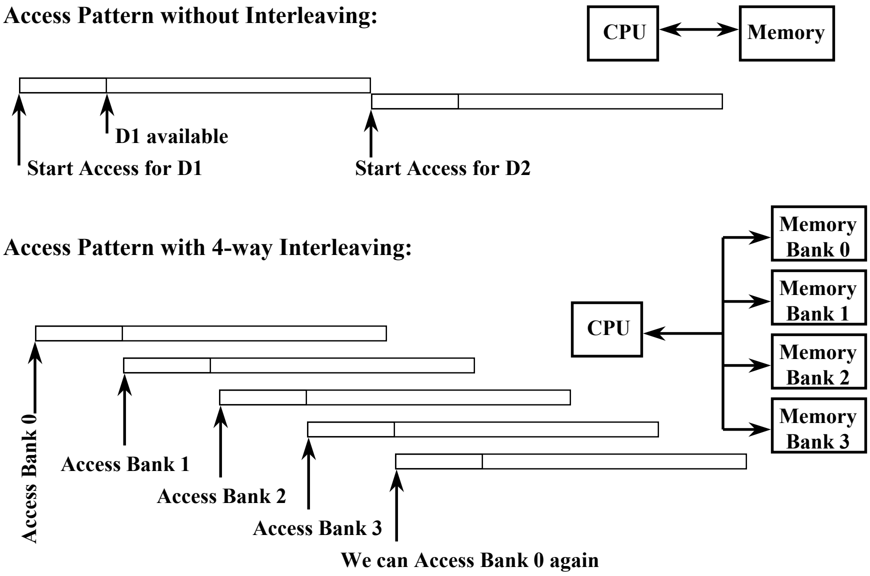
° DRAM (Read/Write) Cycle Time >> DRAM (Read/Write) Access Time

° 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:

CPU ↔ Memory

D1 available

Start Access for D1

Start Access for D2

Access Pattern with 4-way Interleaving:

CPU

Memory Bank 0

Memory Bank 1

Memory Bank 2

Memory Bank 3

Access Bank 0

Access Bank 1

Access Bank 2

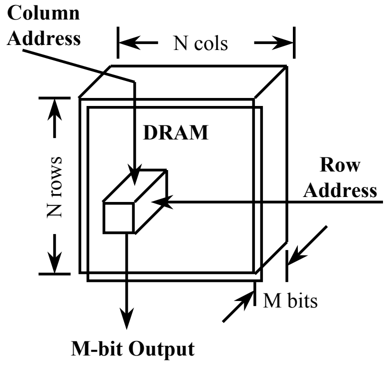
Access Bank 3

We can Access Bank 0 again

Fast Page Mode DRAM

° 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



Column Address

N cols

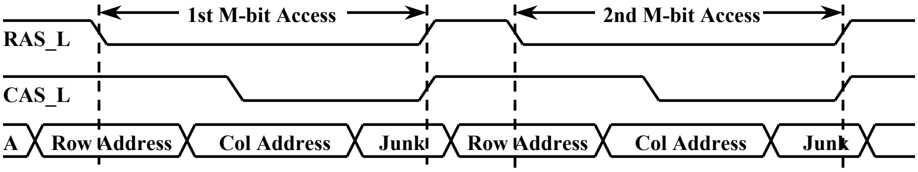
DRAM

N rows

Row Address

M bits

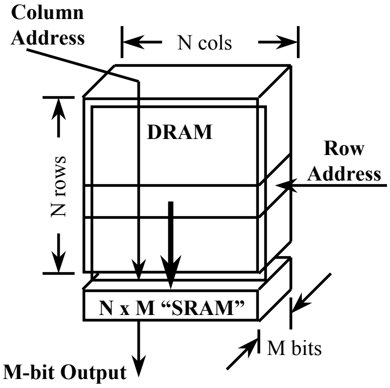
M-bit Output

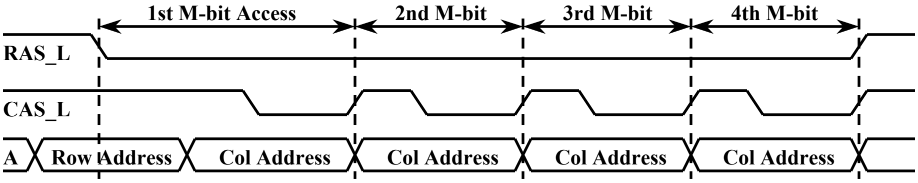


Fast Page Mode Operation

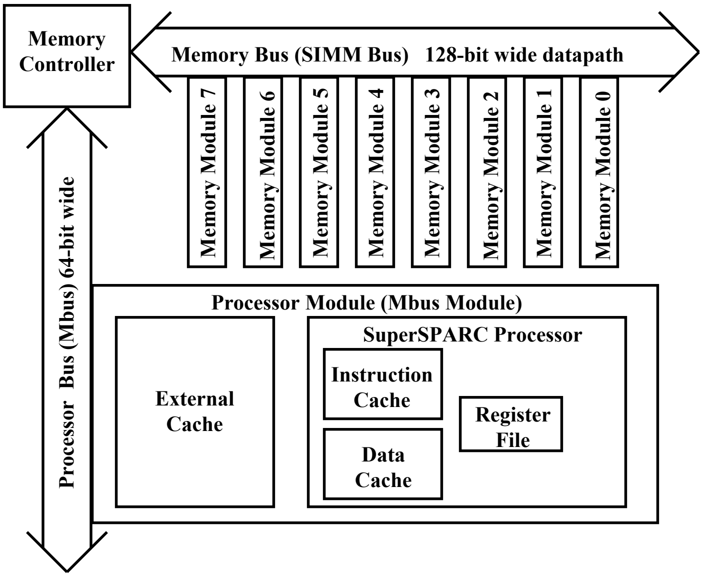
° Fast Page Mode DRAM
  - N x M “SRAM” to save a row

° 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



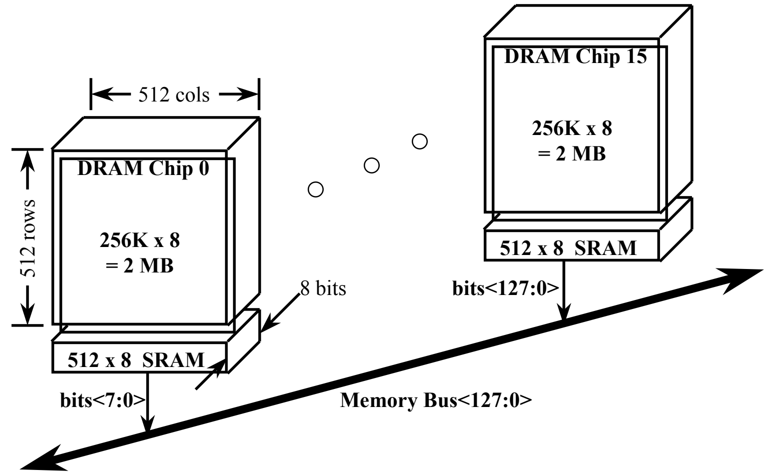


SPARCstation 20’s Memory System Overview



SPARCstation 20's Memory Module

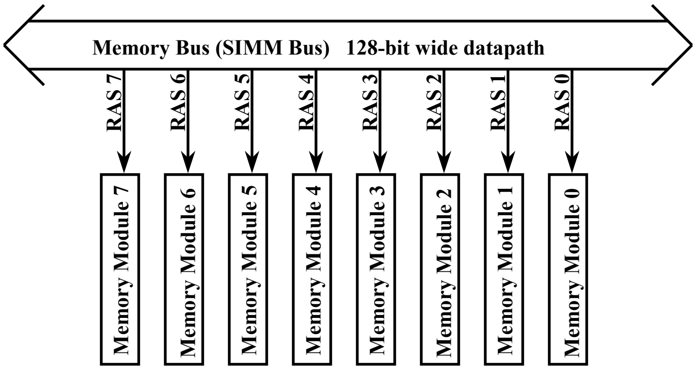
° Supports a wide range of sizes:
  - Smallest 4 MB: 16 2Mb DRAM chips, 8 KB of Page Mode SRAM
  - Biggest: 64 MB: 32 16Mb chips, 16 KB of Page Mode SRAM



SPARCstation 20's Main Memory

° Biggest Possible Main Memory :
  - 8 64MB Modules: 8 x 64 MB DRAM   8 x 16 KB of Page Mode SRAM

° How do we select 1 out of the 8 memory modules?
Remember: every DRAM operation start with the assertion of RAS
  - SS20's Memory Bus has 8 separate RAS lines



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.

Where to get more information?

- To be continued ...