

# CS354: Machine Organization and Programming

Lecture 18

Wednesday the October 14<sup>th</sup> 2015

Section 2

Instructor: Leo Arulraj

© 2015 Karen Smoler Miller

© Some examples, diagrams from the CSAPP text by Bryant and O'Hallaron

# Class Announcements

1. Programming Assignment 1 Grades and Feedback distributed in class. You can collect it from me during office hours too.
2. How many of you have started working on Programming Assignment 2? Solved 1 , 2, 3, 4 bombs?

# Lecture Overview

1. Stack Smashing Real Example
2. Ways to thwart Stack Smashing
3. Intro to Storage Technology: SRAM, DRAM, Disks

# SRAM

1. Six transistor circuit to store a single bit
2. **Bi-stable memory cell:** Can tolerate disturbances due to electrical noise, voltage perturbations etc.
3. **Persistent:** Retains value indefinitely
4. **Faster and More expensive than DRAM**
5. **Uses:** CPU Caches

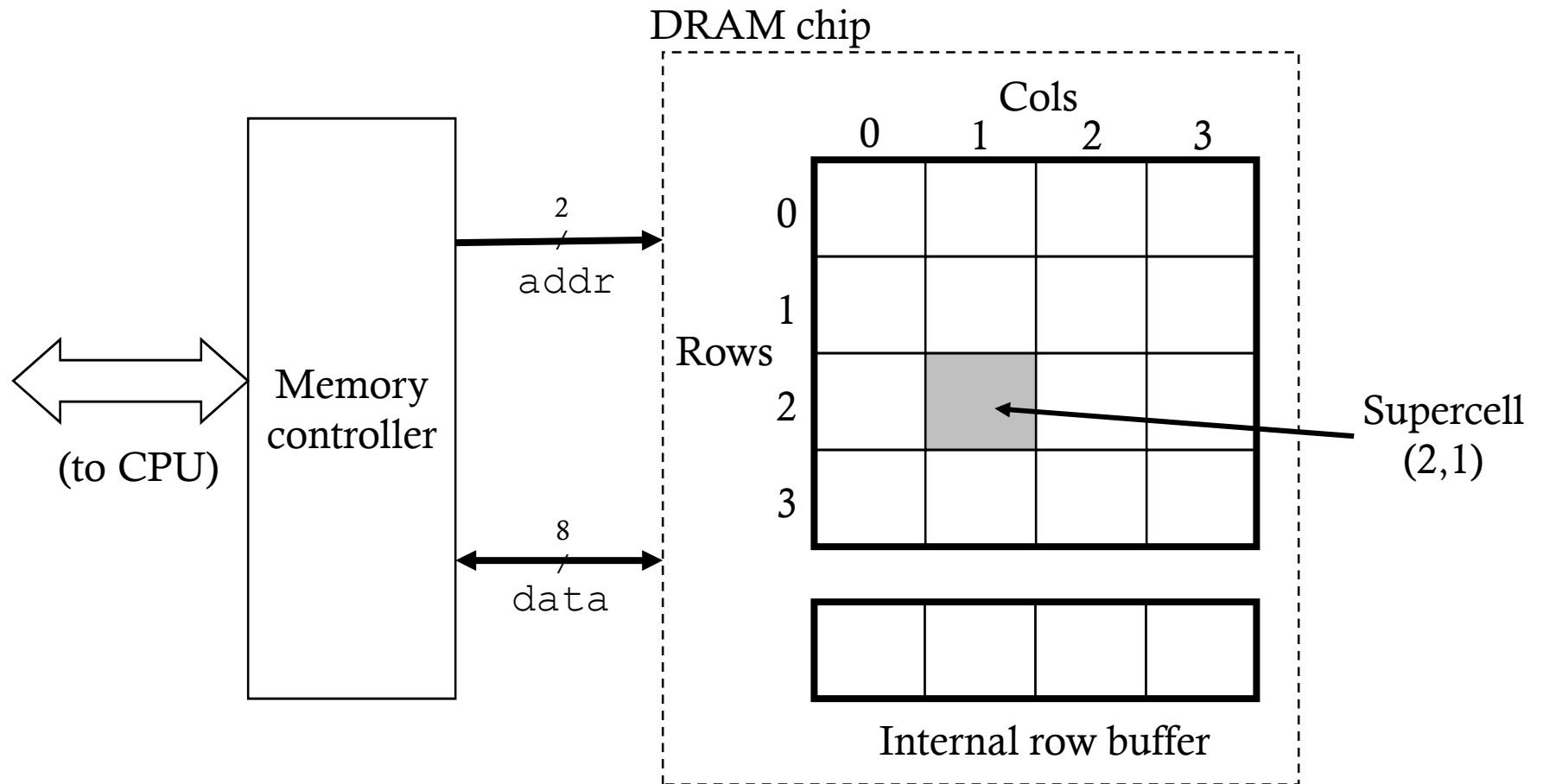
# DRAM

1. A capacitor to store a single bit and a transistor to access the bit
2. Sensitive to disturbances in voltage, electrical noise etc.
3. **Not persistent:** Value must be refreshed every 10-100ms
4. **Slower and less expensive than SRAM**
5. **Uses:** Main memory, frame buffers

# Conventional DRAM

1.  $d$  supercells each of  $w$  DRAM cells storing a total of  $dw$  bits of information.
2. Organized as a rectangular array of  $r$  rows and  $c$  columns.
3. Memory controller transfers  $w$  bits at a time to and from the DRAM chip.
4. Row address  $i$  is called a *RAS(Row Access Strobe) Request*.
5. Column address  $j$  is called a *CAS(Column Access Strobe) Request*.
6. Arranging DRAM in rows and columns allows lesser address pins but increases access time.

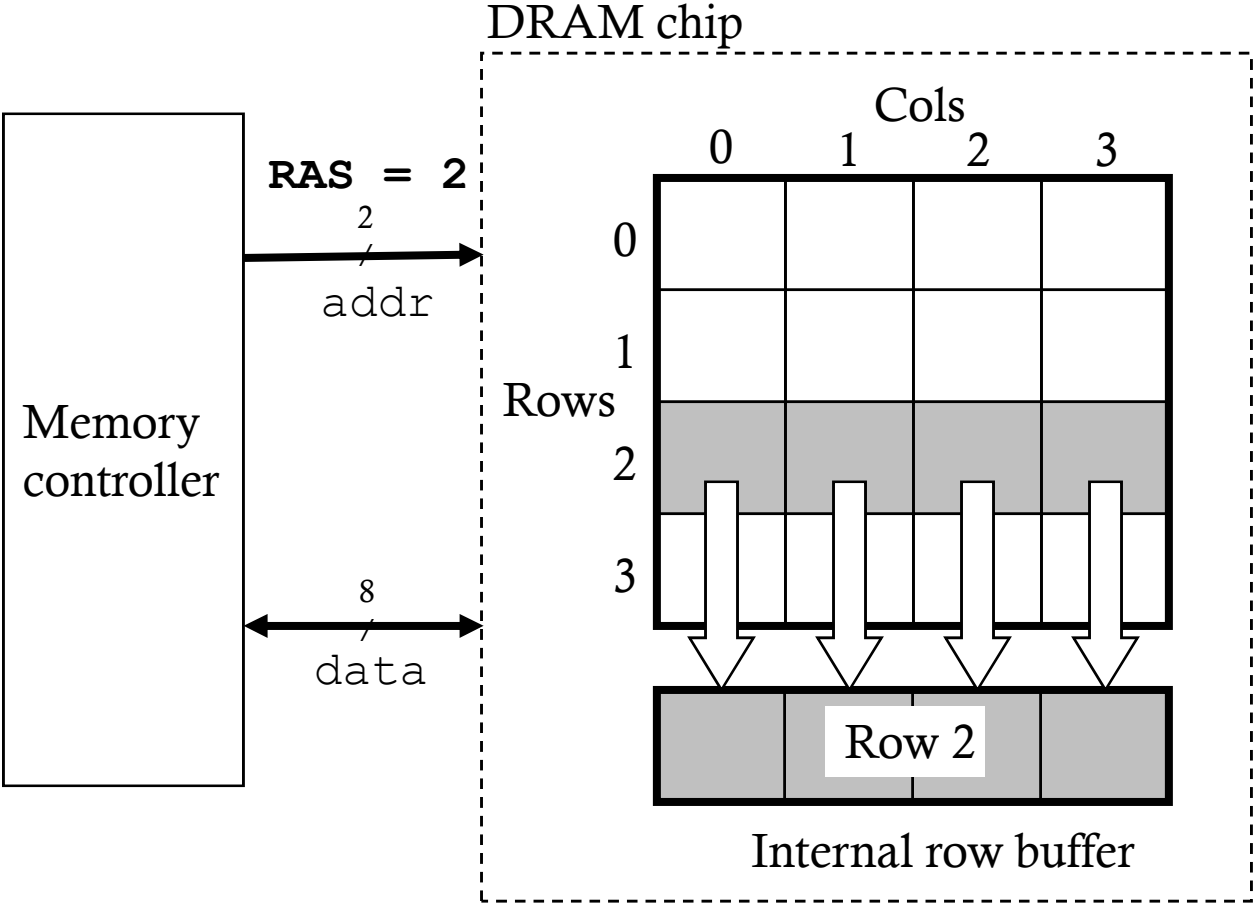
# 128 Bit DRAM chip with 8 bit supercells

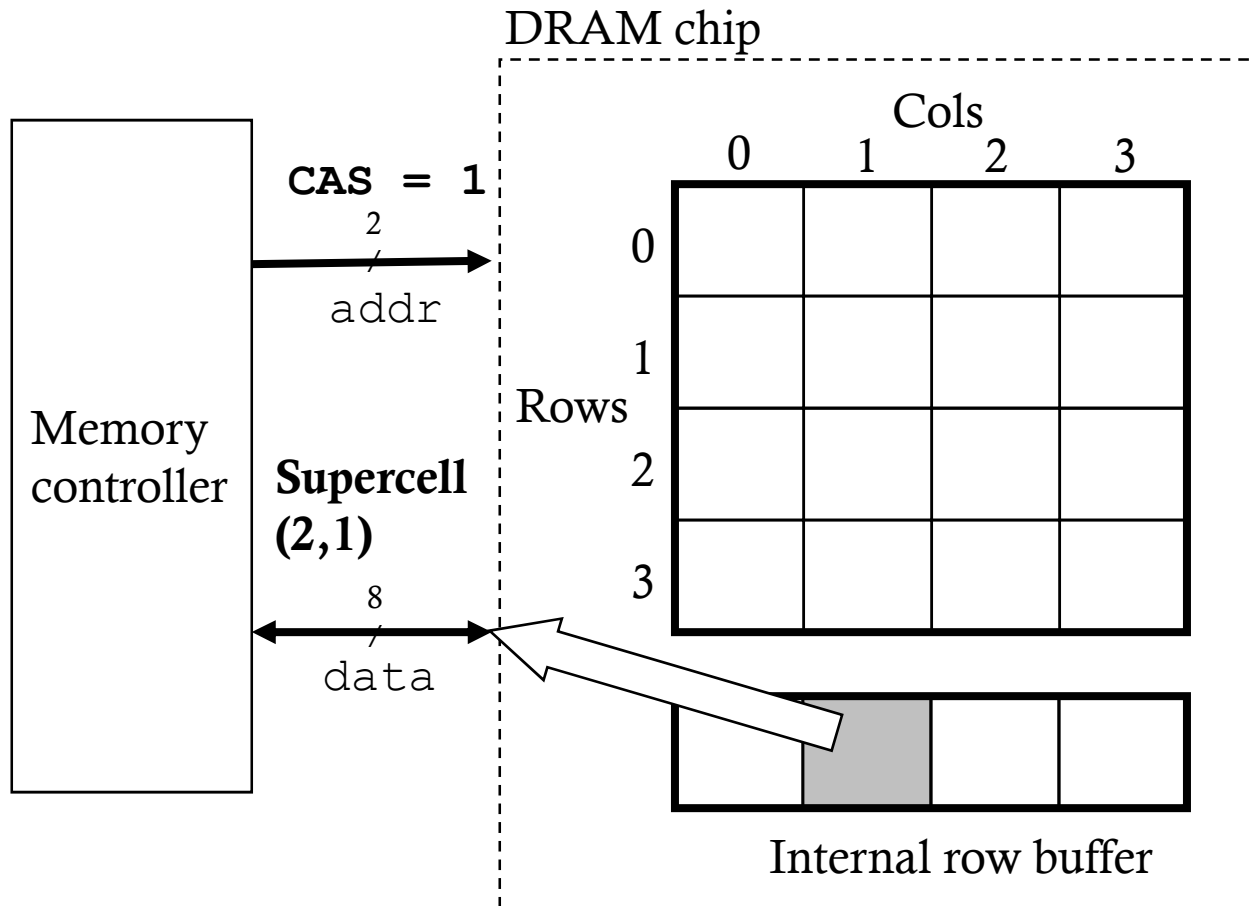


# Conventional DRAM ( Shown diagrammatically in next two slides)

1. Send RAS  $i=2$
2. Entire row is fetched in to the Internal Row Buffer
3. Send CAS  $j=1$
4. The 8-bits in the supercell (2,1) are sent through the data pins.







# Memory Modules

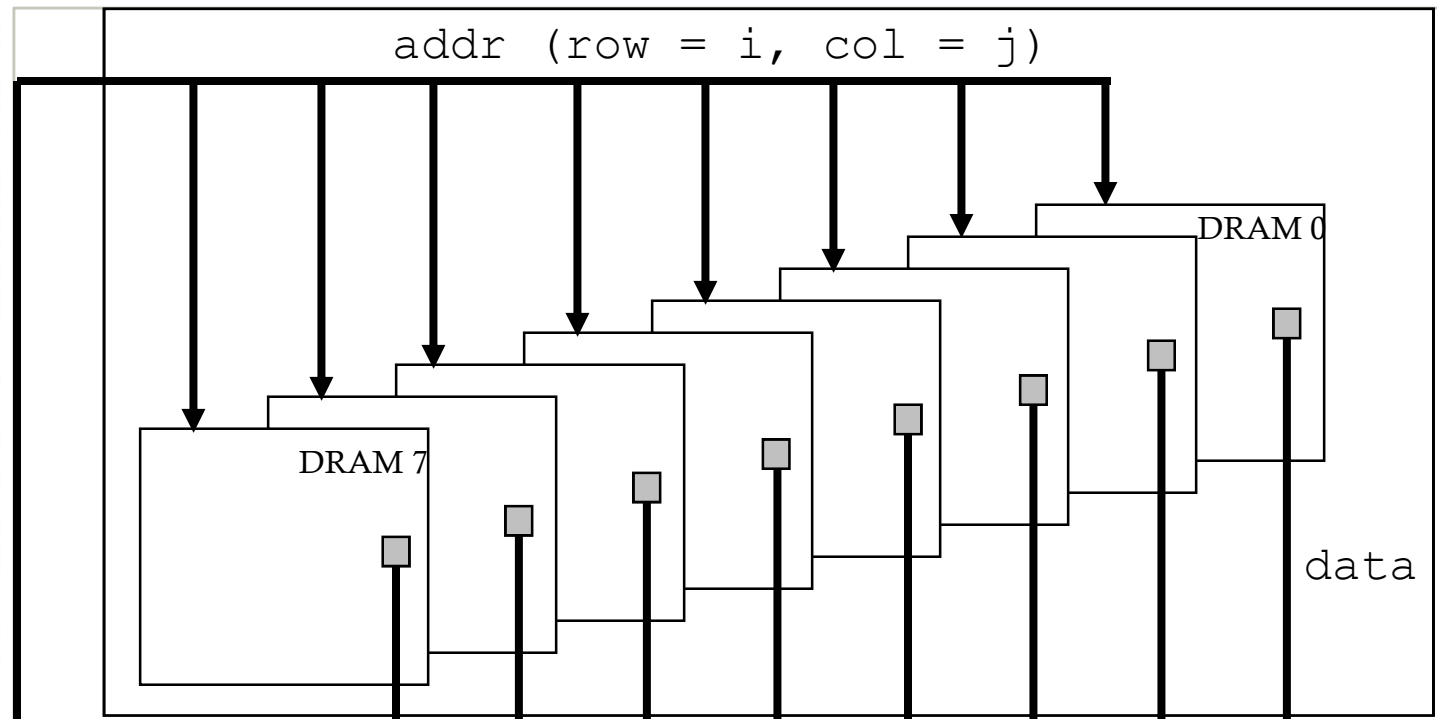
1. DRAMs packaged into memory modules that plugin into expansion slots in motherboard.

(Data at Address  $A$  is split in bytes and each byte is stored into each of the DRAM chips in the same supercell.)



2. Multiple memory modules can themselves be aggregated by connecting to memory controller.

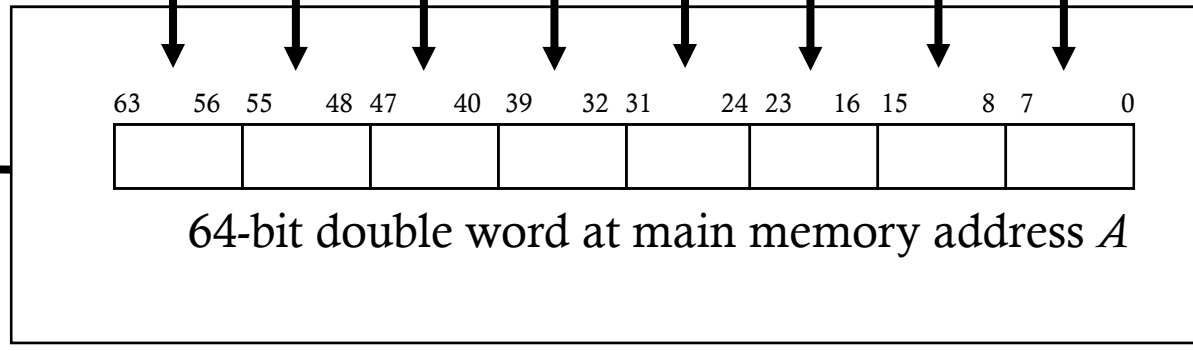
(Data at Address  $A$  is in a specific memory module  $k$ )



■: Supercell (i,j)

64 MB  
memory module  
consisting of  
8 8Mx8 DRAMs

bits 56-63    bits 48-55    bits 40-47    bits 32-39    bits 24-31    bits 16-23    bits 8-15    bits 0-7



Memory  
controller

64-bit doubleword to CPU chip

# Enhanced DRAM

1. **Fast Page DRAM:** consecutive accesses to same row is fast by skipping the RAS
2. **Extended data out DRAM:** CAS signals closer in time
3. **Synchronous DRAM(SDRAM):** control signals sent with rising edges of the memory controller's clock making it fast.
4. **Double Data-Rate Synchronous DRAM( DDR SDRAM):** both clock edges used as control signals
5. **Rambus DRAM(RDRAM):** alternative proprietary technology with higher maximum bandwidth
6. **Video RAM:** Allows concurrent reads and writes. Output produced by shifting entire contents of internal buffer in sequence.

# Non-volatile Memory

1. **PROM:** Programmable Read only Memory
2. **EPROM:** Erasable Programmable ROM  
(Reprogrammable 1000 times)
3. **EEPROM:** Electrically Erasable Programmable  
ROM (Reprogrammable 100000 times)
4. **Flash Memory:** based on EEPROMs and found in  
digital cameras, phones, ipods etc.
5. **Solid State Drives:** Alternative to magnetic disks  
based on flash. We will cover this in more detail  
later.

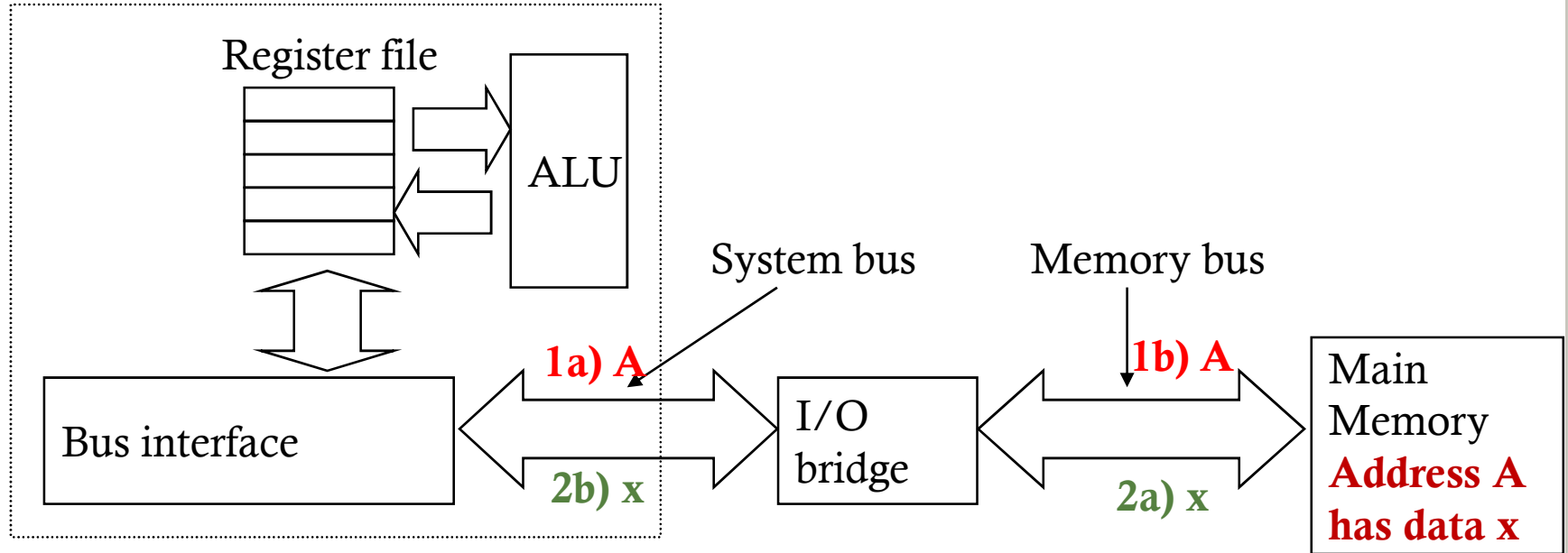
# Accessing Main Memory

What happens with: **movl Address A, %eax**

Bus interface in CPU initiates a **read** transaction with three steps:

1. CPU places address *A* on the system bus and I/O bridge passes it along to memory bus
2. Main memory senses address signal, reads address *A* from the memory bus, fetches data word from the DRAM and writes to the memory bus. I/O bridge passes it along the system bus.
3. CPU senses the data on the system bus, reads it and copies into the register `%eax`.

CPU chip





# Accessing Main Memory

What happens with: **movl %eax, Address A**

Bus interface in CPU initiates a **write** transaction with three steps:

1. CPU places address A on the system bus and I/O bridge passes it along to memory bus. Main memory senses address signal, waits for the data to arrive.
2. CPU copies the data word in %eax to the system bus and the I/O bridge passes it along.
3. Main memory reads the data word from the memory bus and stores the bits in the DRAM supercell.

CPU chip

