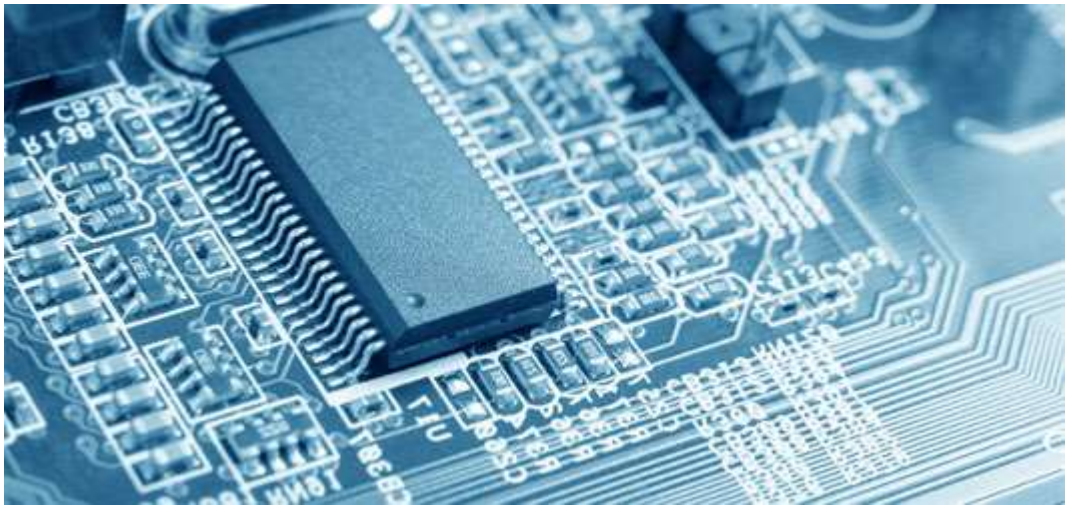
NAZARBAYEV UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY



# ROBT206 – Microcontrollers with Lab

## Lectures 21 – Memory Basics

**10 April, 2018**

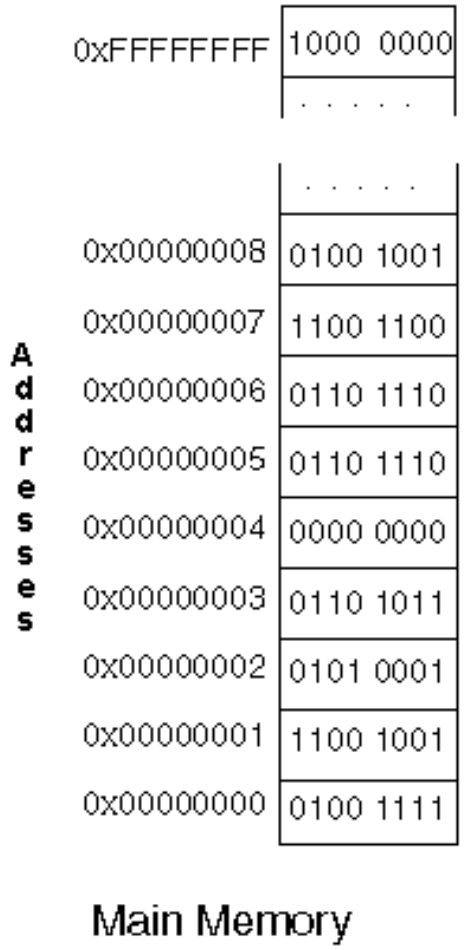
# Topics

## Today's Topics

- Memory definitions
- Random Access Memory (RAM)
- Static RAM (SRAM) integrated circuits
  - Cells and slices
  - Cell arrays and coincident selection
- Arrays of SRAM integrated circuits
- Dynamic RAM (DRAM) integrated circuits
- DRAM Types
  - Synchronous (SDRAM)
  - Double-Data Rate (DDR SRAM)
  - RAMBUS DRAM (RDRAM)
- Arrays of DRAM integrated circuits

# Memory Definitions

- **Memory** — A collection of storage cells together with the necessary circuits to transfer information to and from them.
- **Memory Organization** — the basic architectural structure of a memory in terms of how data is accessed.
- **Random Access Memory (RAM)** — a memory organized such that data can be transferred to or from any cell (or collection of cells) in a time that is not dependent upon the particular cell selected.
- **Memory Address** — A vector of bits that identifies a particular memory element (or collection of elements).

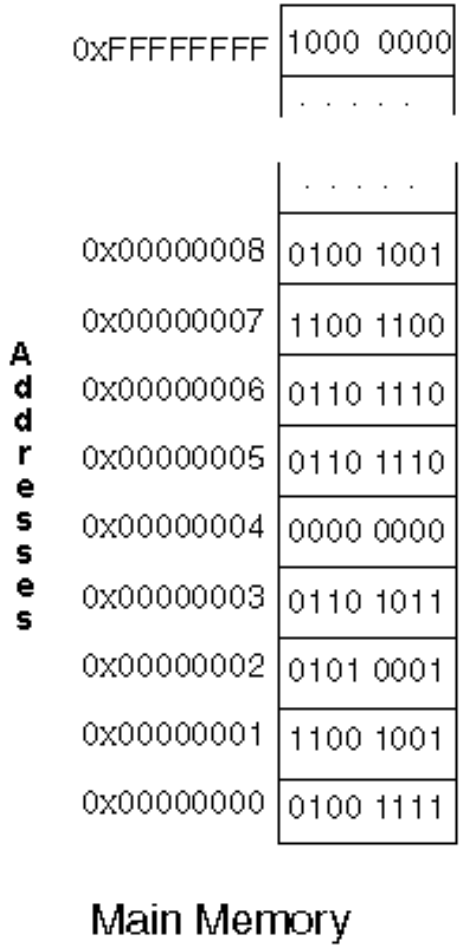


# Memory Definitions

- Typical data elements are:
  - **bit** — a single binary digit
  - **byte** — a collection of eight bits accessed together
  - **word** — a collection of binary bits whose size is a typical unit of access for the memory. It is typically a power of two multiple of bytes (e.g., 1 byte, 2 bytes, 4 bytes, 8 bytes, etc.)
- **Memory Data** — a bit or a collection of bits to be stored into or accessed from memory cells.
- **Memory Operations** — operations on memory data supported by the memory unit. Typically, ***read*** and ***write*** operations over some data element (bit, byte, word, etc.)

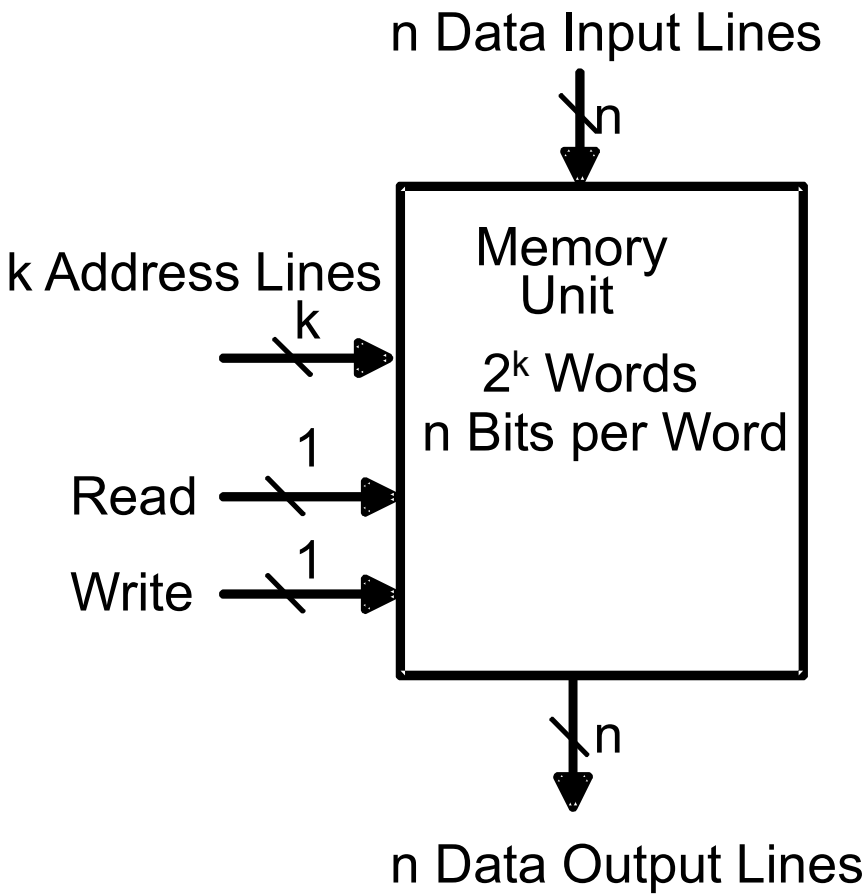
# Memory Organization

- Organized as an indexed **array of words**. Value of the index for each word is the **memory address**.
- Often organized to fit the needs of a particular computer architecture. Some historically significant computer architectures and their associated memory organization:
  - Digital Equipment Corporation PDP-8 – used a 12-bit address to address 4096 12-bit words.
  - IBM 360 – used a 24-bit address to address 16,777,216 8-bit bytes, or 4,194,304 32-bit words.
  - Intel 8080 – (8-bit predecessor to the 8086 and the current Intel processors) used a 16-bit address to address 65,536 8-bit bytes.



# Memory Block Diagram

- A basic memory system is shown here:
- k address lines are decoded to address $2^k$ words of memory.
- Each word is n bits.
- Read and Write are single control lines defining the simplest of memory operations.



# Memory Organization Example

- Example memory contents:
  - A memory with 3 address bits & 8 data bits has:
  - k = 3 and n = 8 so $2^3 = 8$ addresses labeled 0 to 7.
  - $2^3 = 8$ words of 8-bit data

| Memory Address Binary | Decimal | Memory Content |
|---|---|---|
| 0 0 0 | 0 | 1 0 0 0 1 1 1 1 |
| 0 0 1 | 1 | 1 1 1 1 1 1 1 1 |
| 0 1 0 | 2 | 1 0 1 1 0 0 0 1 |
| 0 1 1 | 3 | 0 0 0 0 0 0 0 0 |
| 1 0 0 | 4 | 1 0 1 1 1 0 0 1 |
| 1 0 1 | 5 | 1 0 0 0 0 1 1 0 |
| 1 1 0 | 6 | 0 0 1 1 0 0 1 1 |
| 1 1 1 | 7 | 1 1 0 0 1 1 0 0 |

# Basic Memory Operations

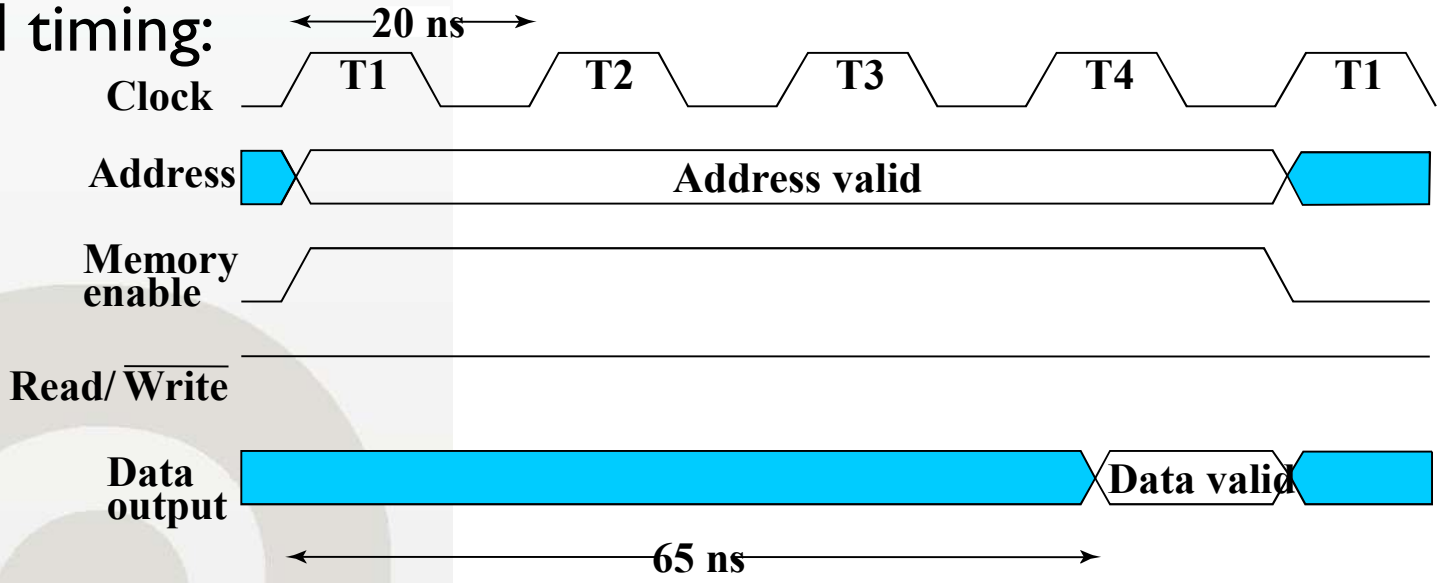
- Memory operations require the following:
  - ***Data*** — data written to, or read from, memory as required by the operation.
  - ***Address*** — specifies the memory location to operate on. The address lines carry this information into the memory. Typically: n bits specify locations of $2^n$ words.
  - **An operation** — Information sent to the memory and interpreted as control information which specifies the type of operation to be performed. Typical operations are READ and WRITE. Others are READ followed by WRITE and a variety of operations associated with delivering blocks of data. Operation signals may also specify timing info.

# Basic Memory Operations

- **Read Memory** — an operation that reads a data value stored in memory:
  - Place a valid address on the address lines.
  - Wait for the read data to become stable.
- **Write Memory** — an operation that writes a data value to memory:
  - Place a valid address on the address lines and valid data on the data lines.
  - Toggle the memory write control line
- Sometimes the read or write enable line is defined as a clock with precise timing information (e.g. Read Clock, Write Strobe).
  - Otherwise, it is just an interface signal.
  - Sometimes memory must acknowledge that it has completed the operation.

# Memory Operation Timing

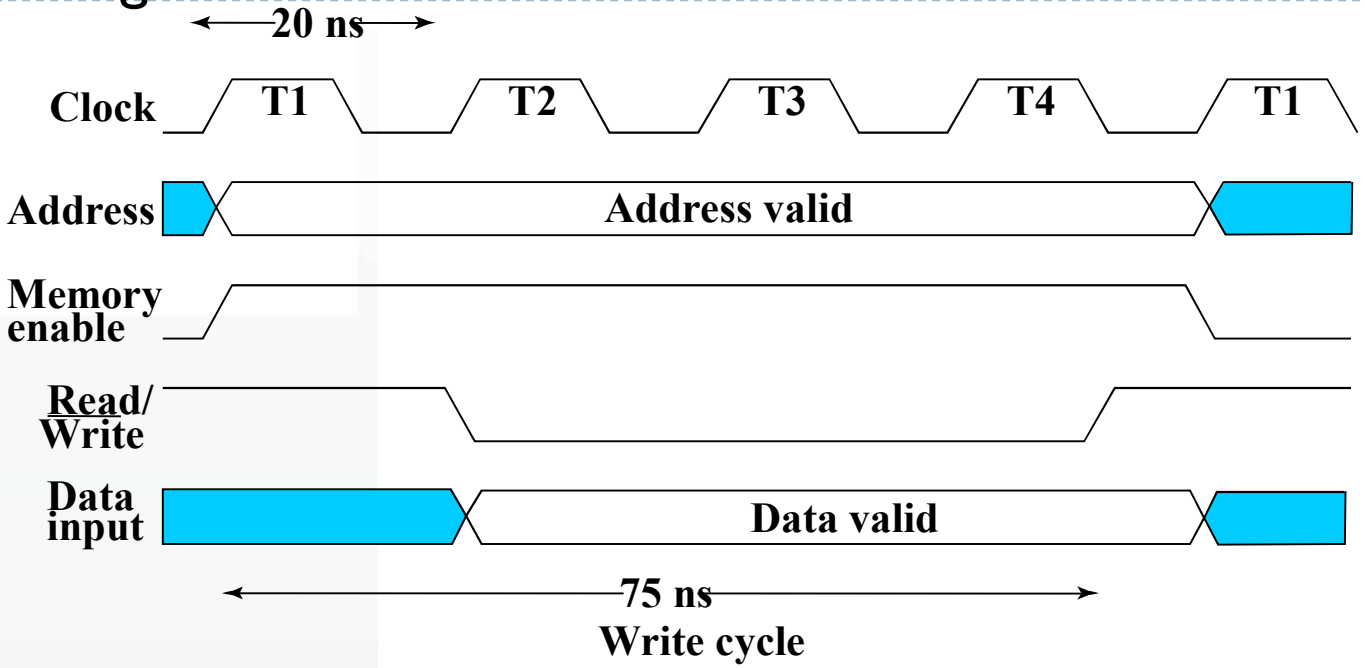
- Most basic memories are asynchronous
  - Storage in latches or storage of electrical charge
  - No clock
- Controlled by control inputs and address
- Timing of signal changes and data observation is critical to the operation
- Read timing:




Access time: maximum time from the application

# Memory Operation Timing

- Write timing:



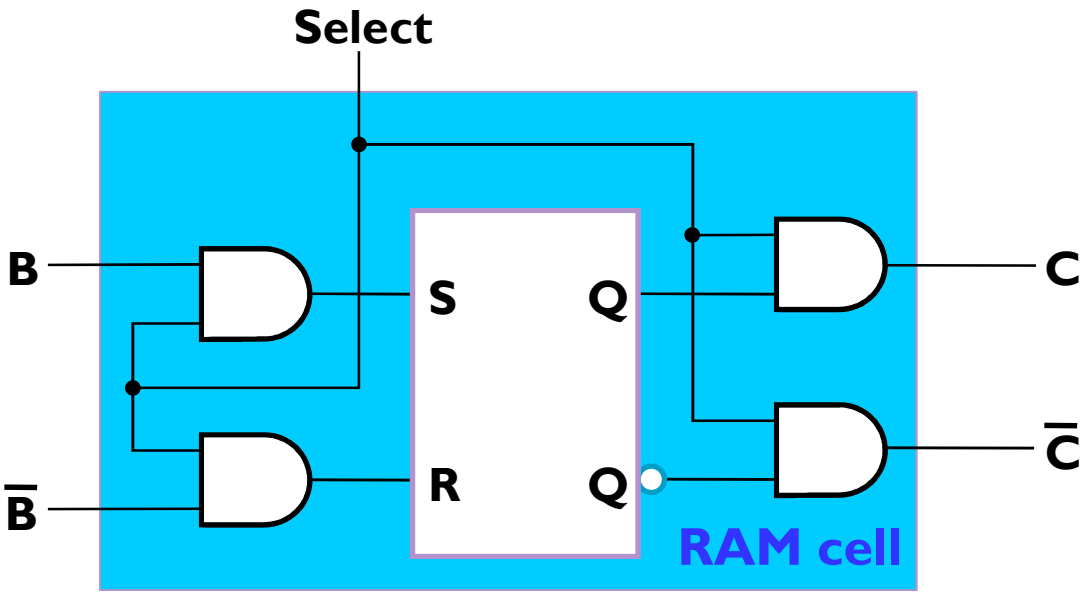

- Critical times measured with respect to edges of write pulse (1-0-1):
  - Address must be established at least a specified time before 1-0 and held for at least a specified time after 0-1 to avoid disturbing stored contents of other addresses
  - Data must be established at least a specified time before 0-1 and held for at least a specified time after 0-1 to write correctly

# RAM Integrated Circuits

- Types of random access memory
  - ***Static*** – information stored in latches
  - ***Dynamic*** – information stored as electrical charges on capacitors
    - Charge "leaks" off
    - Periodic *refresh* of charge required
- Dependence on Power Supply
  - ***Volatile*** – loses stored information when power turned off
  - ***Non-volatile*** – retains information when power turned off

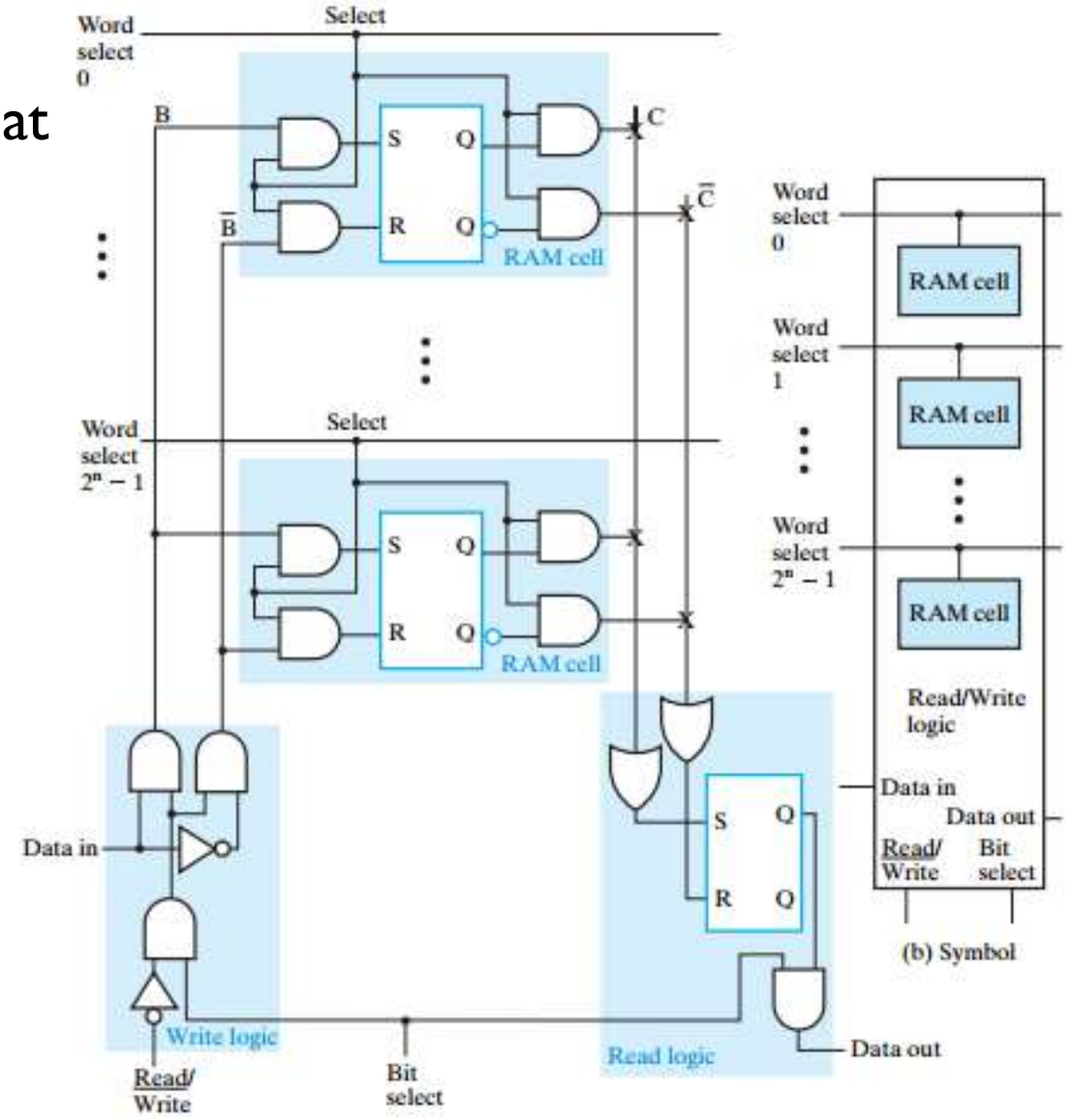
# Static RAM Cell

- Array of storage cells used to implement static RAM
- Storage Cell
  - SR Latch
  - Select input for control
  - Dual Rail Data Inputs B and $\overline{B}$
  - Dual Rail Data Outputs C and $\overline{C}$



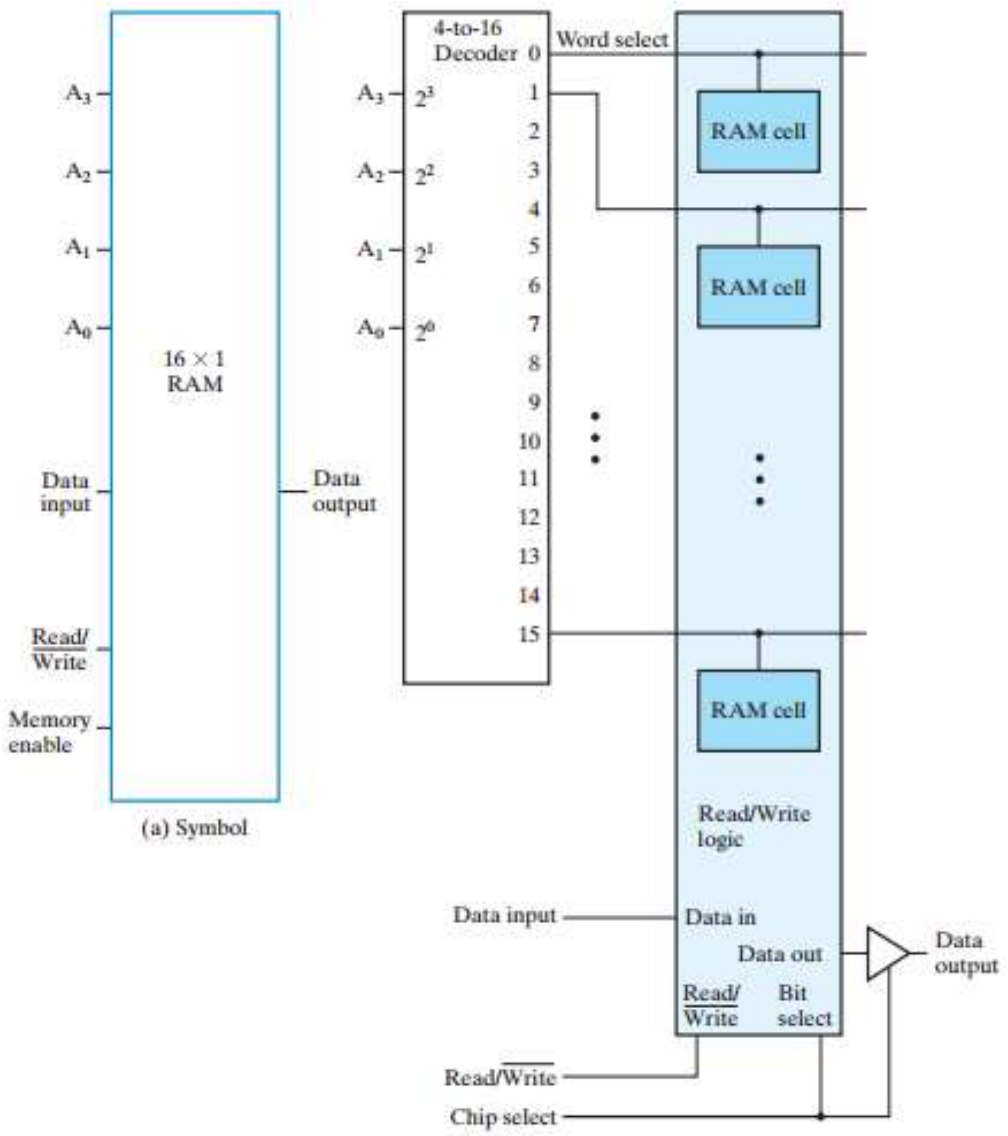
# Static RAM Bit Slice

- Represents all circuitry that is required for $2^n$ 1-bit words
  - Multiple RAM cells
  - Control Lines:
    - Word select i – one for each word
    - Read/$\overline{Write}$
    - Bit Select
  - Data Lines:
    - Data in
    - Data out



# $2^n$-Word × 1-Bit RAM IC

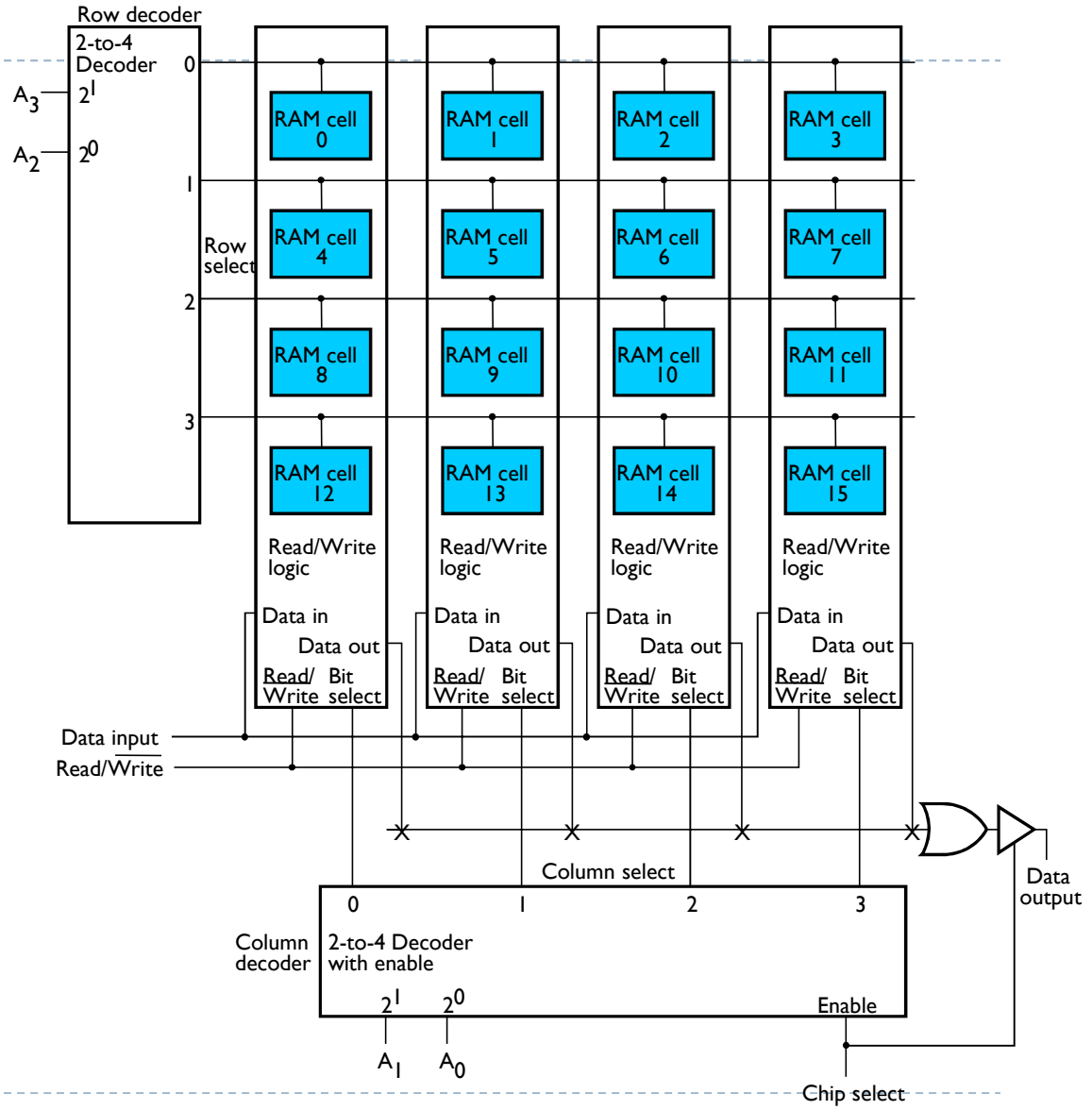
- To build a RAM IC from a RAM slice, we need:
  - Decoder decodes the n address lines to $2^n$ word select lines
  - A 3-state buffer
  - on the data output permits RAM ICs to be combined into a RAM with c × $2^n$ words



# Cell Arrays and Coincident Selection

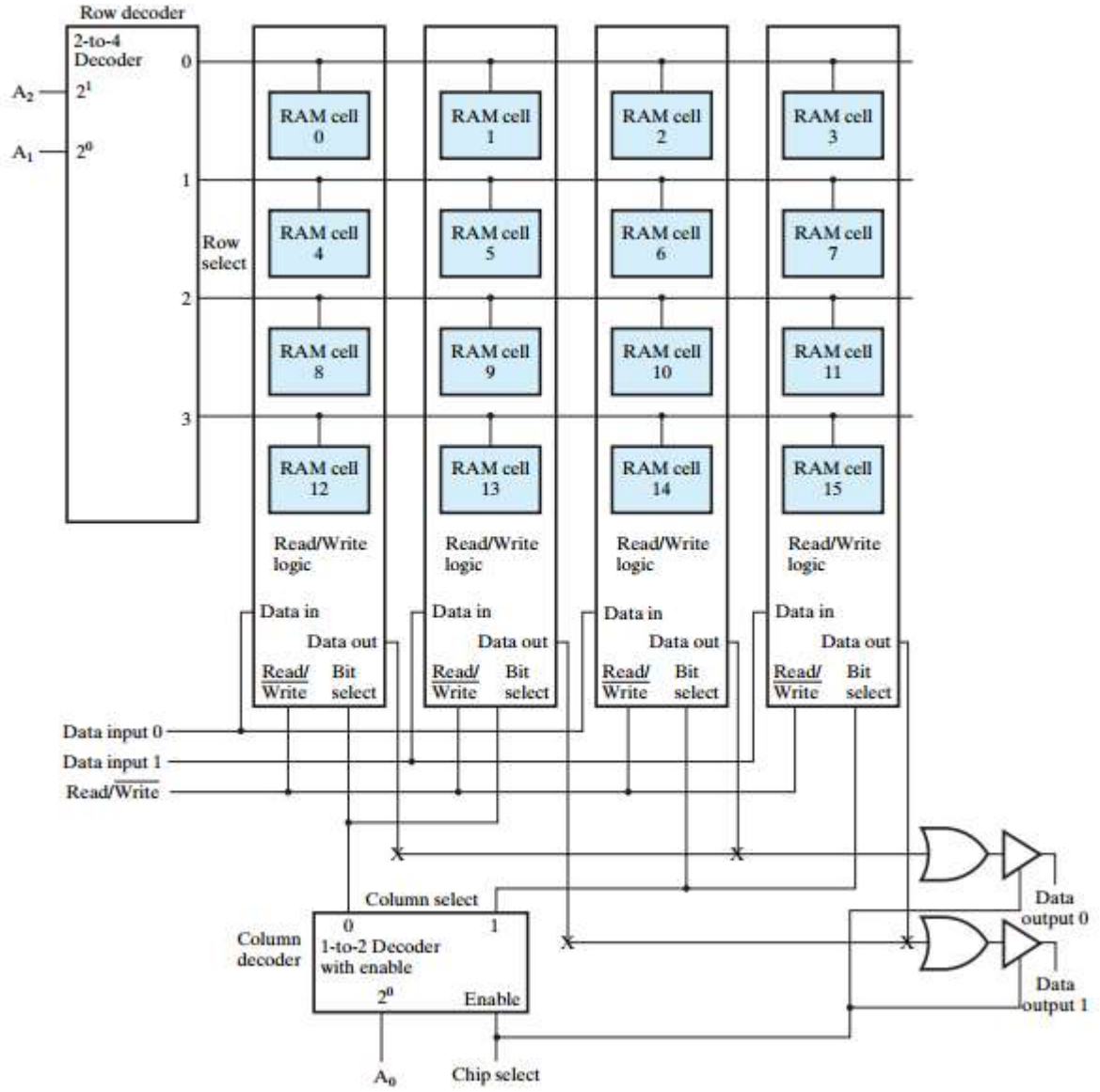
- Memory arrays can be very large =>
  - Large decoders
  - Large fanouts for the bit lines
  - The decoder size and fanouts can be reduced by approximately $\sqrt{n}$ by using a coincident selection in a 2-dimensional array
  - Uses two decoders, one for words and one for bits
  - Word select becomes Row select
  - Bit select becomes Column select
- See next slide for example
  - $A_3$ and $A_2$ used for Row select
  - $A_1$ and $A_0$ for Column select

# Cell Arrays and Coincident Selection (continued)



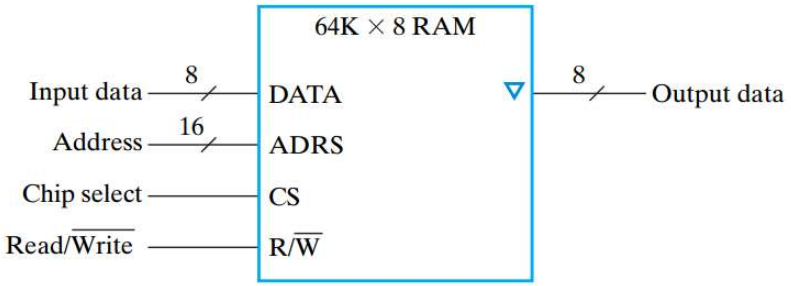
# RAM ICs with > 1 Bit/Word

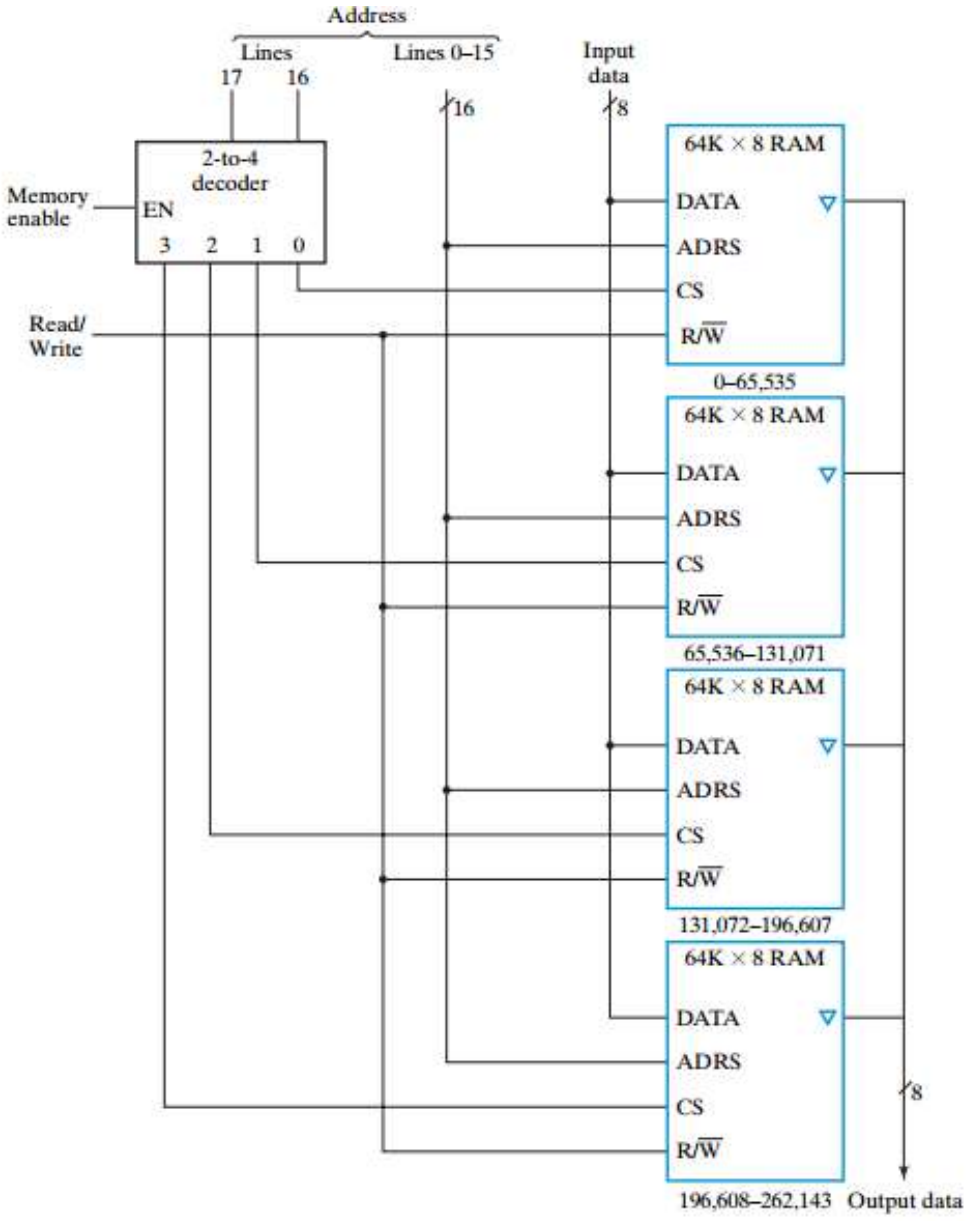
- Word length can be quite high.
- To better balance the number of words and word length, use ICs with > 1 bit/word
- Example: 8 x 2 bit RAM
  - 2 Data input bits
  - 2 Data output bits
  - Row select selects 4 rows
  - Column select selects 2 pairs of columns



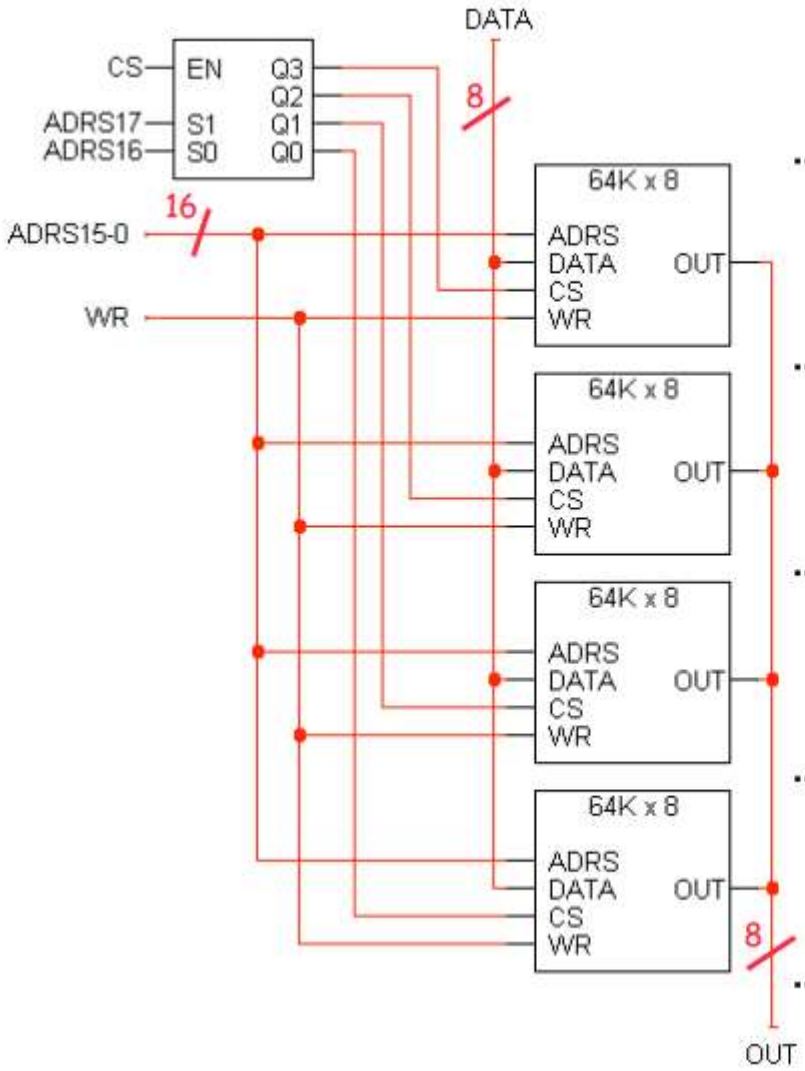
# Making Larger Memories

- Using the CS lines, we can make larger memories from smaller ones by tying all address, data, and R/W lines in parallel, and using the decoded higher order address bits to control CS.
- Example: using the 64K Word by 8-Bit RAM, we construct a 256K x 8 bit RAM. ⇒



# Address Ranges




11 1111 1111 1111 1111 (0x3ffff)
to
11 0000 0000 0000 0000 (0x30000)

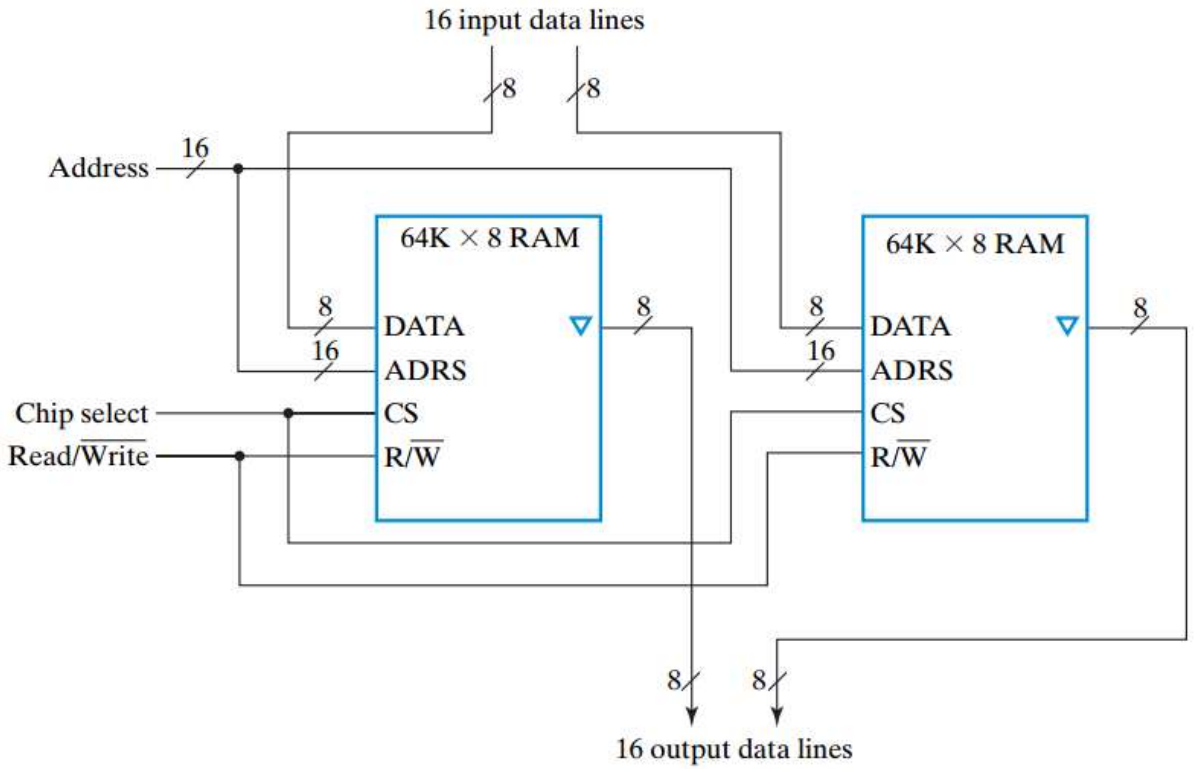
10 1111 1111 1111 1111 (0x2ffff)
to
10 0000 0000 0000 0000 (0x20000)

01 1111 1111 1111 1111 (0x1ffff)
to
01 0000 0000 0000 0000 (0x10000)

00 1111 1111 1111 1111 (0x0ffff)
to
00 0000 0000 0000 0000 (0x00000)

# Making Wider Memories

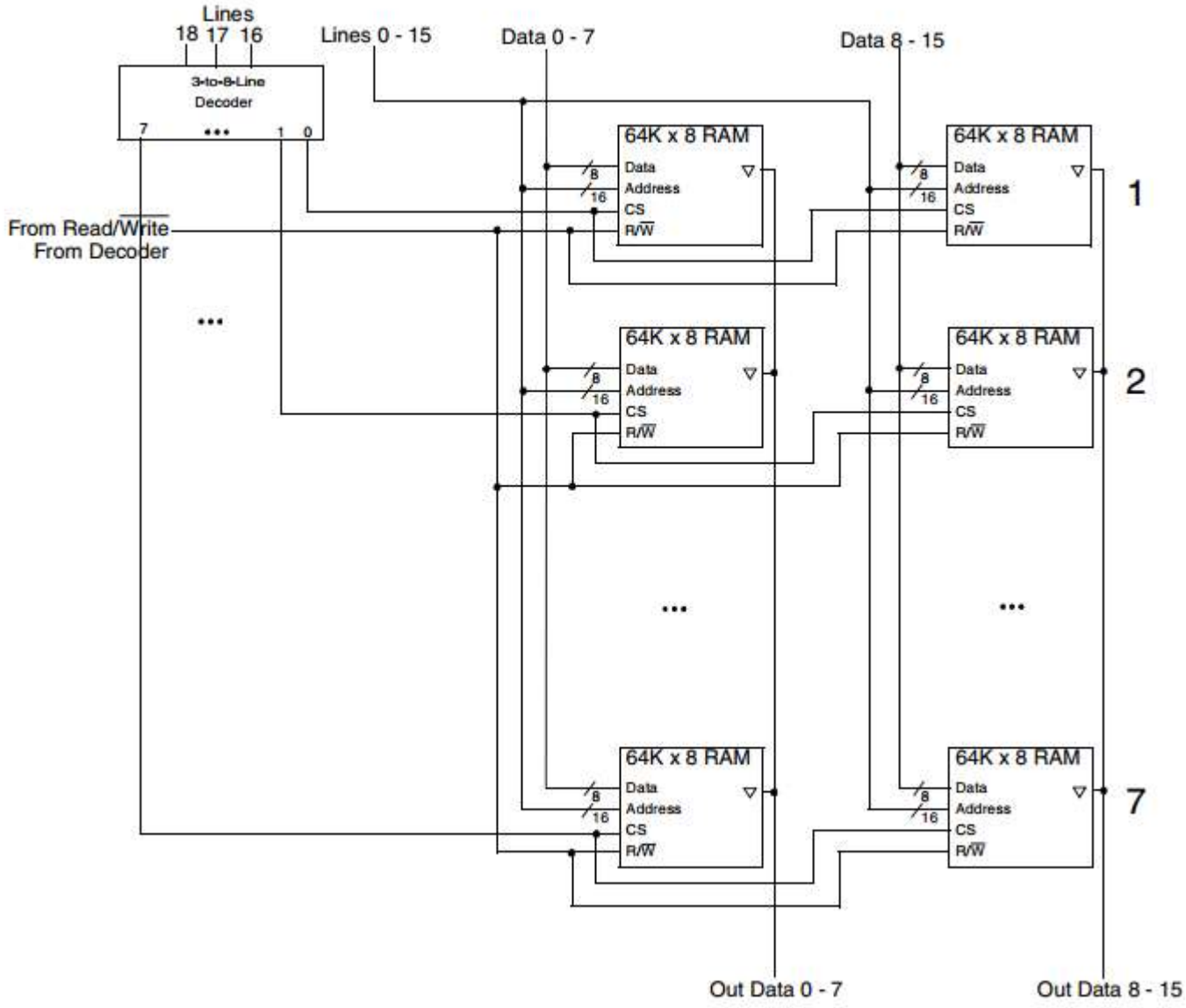
- To construct wider memories from narrow ones, we tie the address and control lines in parallel and keep the data lines separate.
- For example, to make a 64K word by 16-bit memory from two 64K x 8 bit RAM
⇒



# Memory Expansion Example 1

- Using the 64K x 8 RAM chip plus a decoder, construct the block diagram for a 512K x16 RAM

# Memory Expansion Example 1



# Memory Expansion Example

We want to build a memory with 4-byte words and a capacity of 2 MB.

1) How many 2K x 8 RAM chips are needed?

2) How many address lines are needed for the memory?

3) How many of these address lines are connected to the address inputs of the RAM chips?

4) How many of these address lines will be used to select the appropriate chip(s)?

5) What size decoder is needed to build the memory?

# Memory Expansion Example

We want to build a memory with 4-byte words and a capacity of 2 MB.

1) How many 2K x 8 RAM chips are needed?

   Capacity1/Capacity2  Capacity2 = length*width (in bits)

   2^21/(2^3*2^11) = 2^7 = 128 chips

2) How many address lines are needed for the memory?

3) How many of these address lines are connected to the address inputs of the RAM chips?

4) How many of these address lines will be used to select the appropriate chip(s)?

5) What size decoder is needed to build the memory?

# Memory Expansion Example

We want to build a memory with 4-byte words and a capacity of 2 MB.

1) How many 2K x 8 RAM chips are needed?

   Capacity1/Capacity2  Capacity2 = length*width (in bits)

   2^21/(2^3*2^11) = 2^7 = 128 chips

2) How many address lines are needed for the memory?

   Log(2, Capacity/(width)  width = words length in bits

   Log(2, 2^21/(2^3*2^2)) = Log(2, 2^21/2^5) = 16

3) How many of these address lines are connected to the address inputs of the RAM chips?

4) How many of these address lines will be used to select the appropriate chip(s)?

5) What size decoder is needed to build the memory?

# Memory Expansion Example

We want to build a memory with 4-byte words and a capacity of 2 MB.

1) How many 2K x 8 RAM chips are needed?

   Capacity1/Capacity2  Capacity2 = length*width (in bits)

   2^21/(2^3*2^11) = 2^7 = 128 chips

2) How many address lines are needed for the memory?

   Log(2, Capacity/(width)  width = words length in bits

   Log(2, 2^21/(2^3*2^2)) = Log(2, 2^21/2^5) = 16

3) How many of these address lines are connected to the address inputs of the RAM chips?

   Log(2, length)  length = number of lines

   Log(2, 2^11) = 11

4) How many of these address lines will be used to select the appropriate chip(s)?

5) What size decoder is needed to build the memory?

# Memory Expansion Example

We want to build a memory with 4-byte words and a capacity of 2 MB.

1) How many 2K x 8 RAM chips are needed?

   Capacity1/Capacity2  Capacity2 = length*width (in bits)

   2^21/(2^3*2^11) = 2^7 = 128 chips

2) How many address lines are needed for the memory?

   Log(2, Capacity/(width)  width = words length in bits

   Log(2, 2^21/(2^3*2^2)) = Log(2, 2^21/2^5) = 16

3) How many of these address lines are connected to the address inputs of the RAM chips?

   Log(2, length)  length = number of lines

   Log(2, 2^11) = 11

4) How many of these address lines will be used to select the appropriate chip(s)?

   #Addrs - #RAM

   16-11 = 5

5) What size decoder is needed to build the memory?

# Memory Expansion Example

We want to build a memory with 4-byte words and a capacity of 2 MB.

1) How many 2K x 8 RAM chips are needed?

   Capacity1/Capacity2  Capacity2 = length*width (in bits)

   2^21/(2^3*2^11) = 2^7 = 128 chips

2) How many address lines are needed for the memory?

   Log(2, Capacity/(width)  width = words length in bits

   Log(2, 2^21/(2^3*2^2)) = Log(2, 2^21/2^5) = 16

3) How many of these address lines are connected to the address inputs of the RAM chips?

   Log(2, length)  length = number of lines

   Log(2, 2^11) = 11

4) How many of these address lines will be used to select the appropriate chip(s)?

   #Addrs - #RAM

   16-11 = 5

5) What size decoder is needed to build the memory?

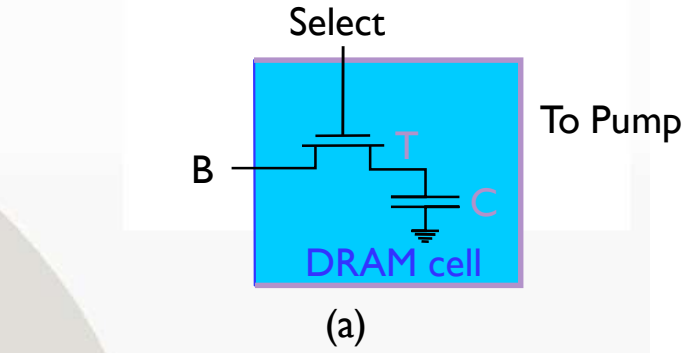
   #Select x 2^#select

   5 x 2^5 = 5 x 32

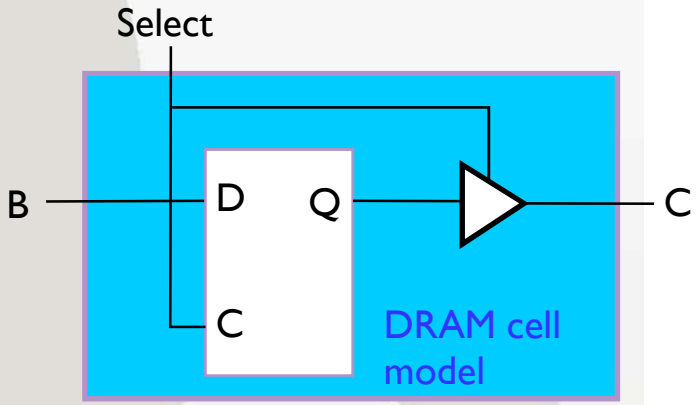
# Dynamic RAM (DRAM)

- Basic Principle: Storage of information on capacitors.
- Charge and discharge of capacitor to change stored value
- Use of transistor as "switch" to:
  - Store charge
  - Charge or discharge
- See next slide for circuit, hydraulic analogy, and logical model.

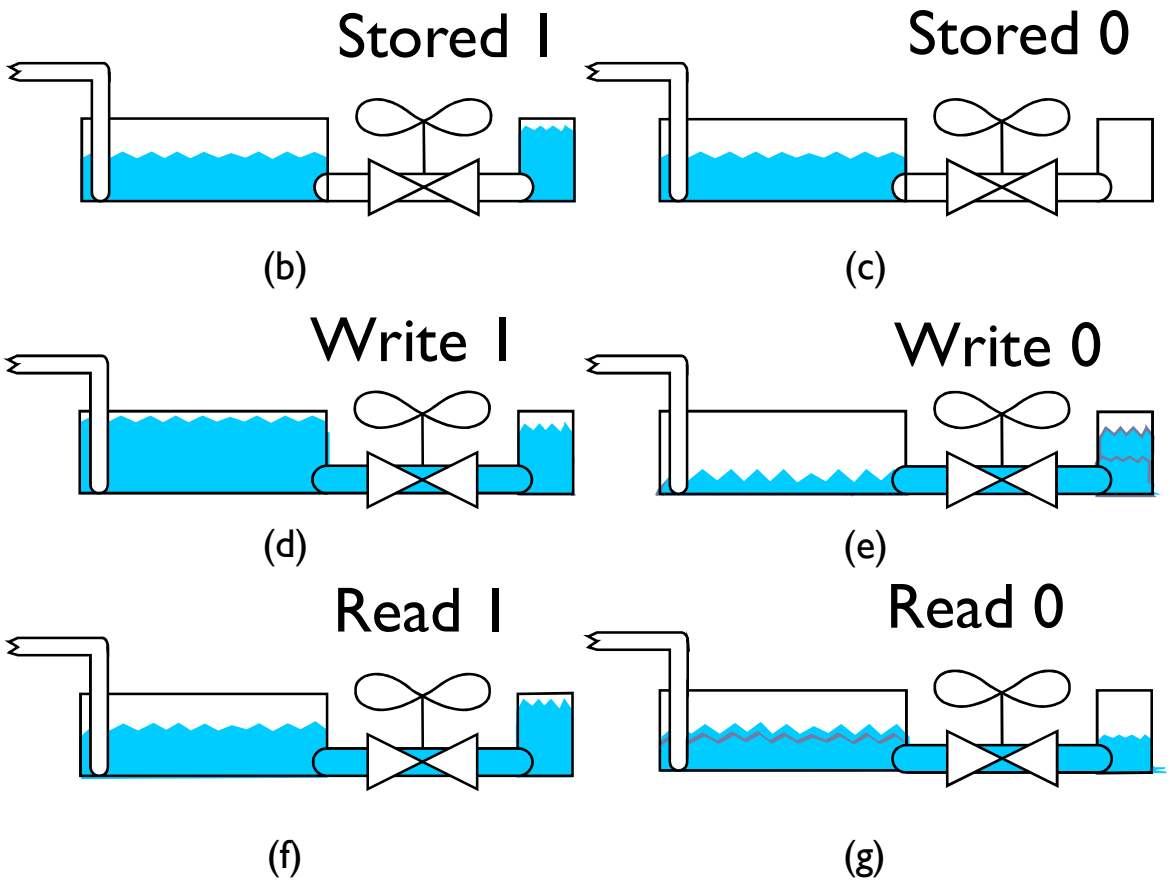
# Dynamic RAM Cell




(a)

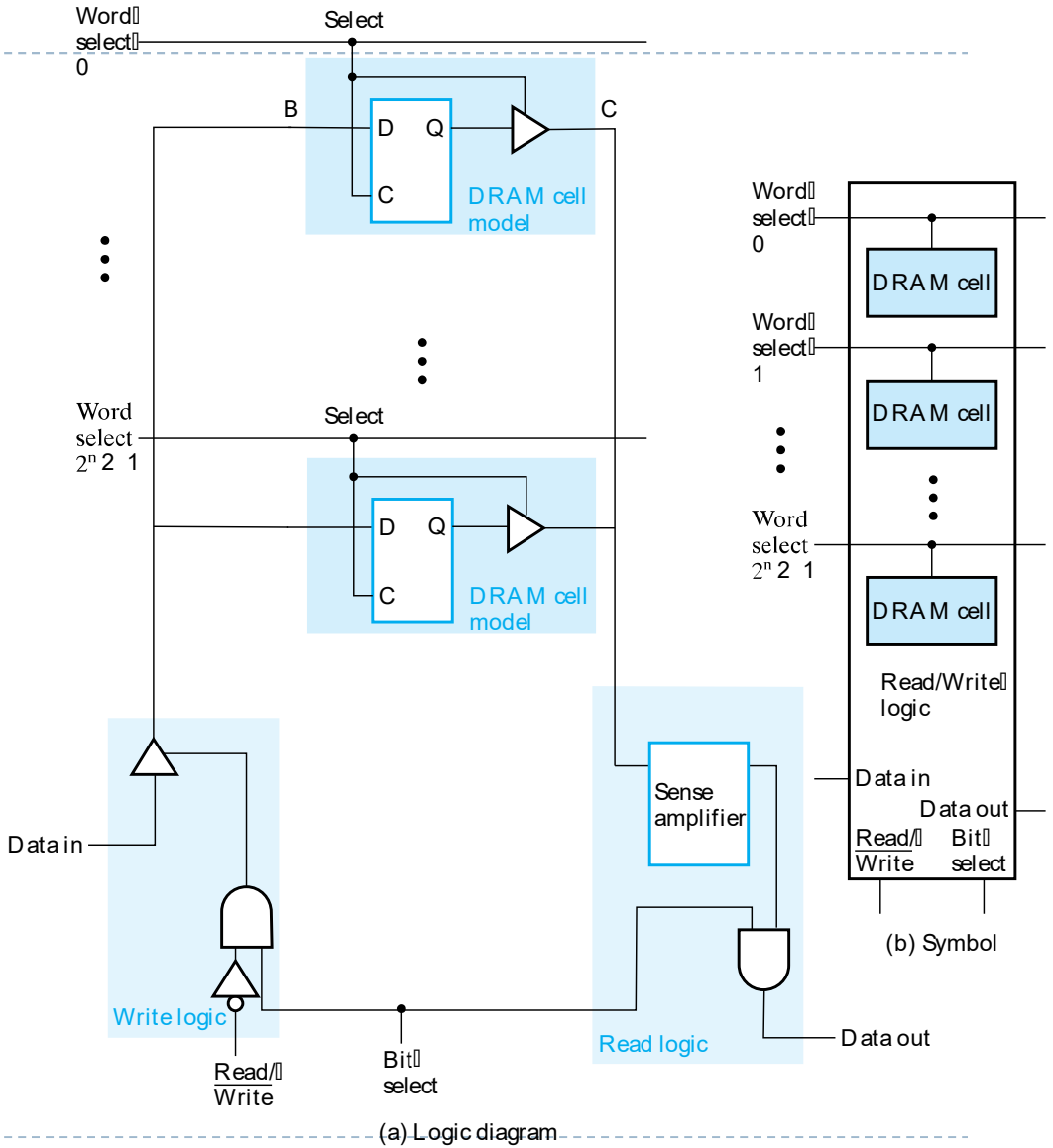



(h)

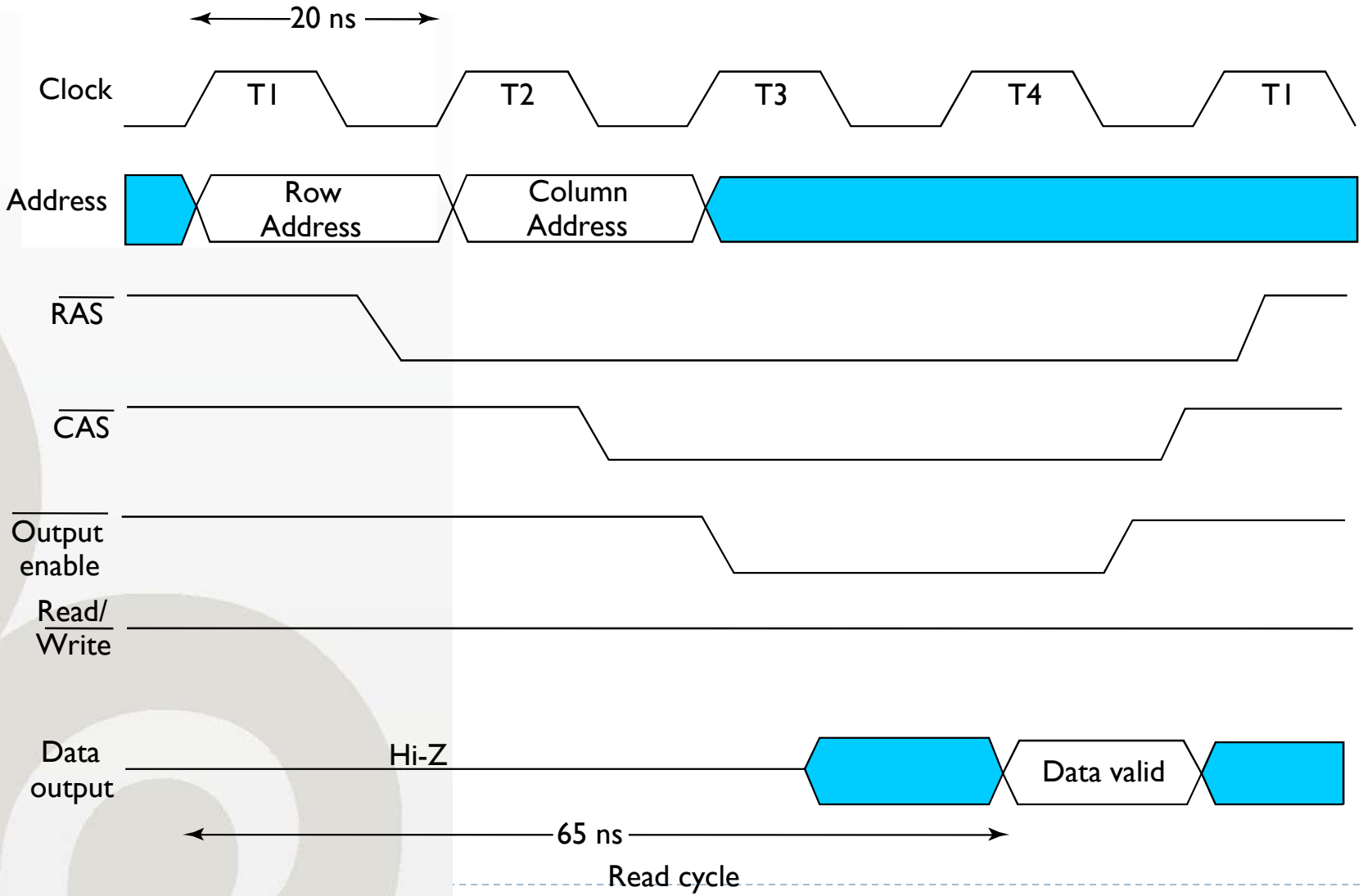


# Dynamic RAM - Bit Slice

- C is driven by 3-state drivers
- Sense amplifier is used to change the small voltage change on C into H or L
- In the electronics, B, C, and the sense amplifier output are connected to make destructive read into non-destructive read



# Dynamic RAM Read Timing



# DRAM Types

- Synchronous DRAM (SDRAM)
- Double Data Rate SDRAM (DDR SDRAM)
- RAMBUS® DRAM (RDRAM)

- ***burst read*** – the resulting multiple word read from consecutive addresses

# Synchronous DRAM

- Transfers to and from the DRAM are synchronized with a clock
- Synchronous registers appear on:
  - Address input
  - Data input
  - Data output

# Double Data Rate Synchronous DRAM

- Transfers data on both edges of the clock
- Provides a transfer rate of 2 data words per clock cycle

# RAMBUS DRAM (RDRAM)

- Uses a packet-based bus for interaction between the RDRAM ICs and the memory bus to the processor
- The bus consists of:
  - A 3-bit row address bus
  - A 5-bit column address bus
  - A 16 or 18-bit (for error correction) data bus
- The bus is synchronous and transfers on both edges of the clock
- Packets are 4-clock cycles long giving 8 transfers per packet representing:
  - A 12-bit row address packet
  - A 20-bit column address packet
  - A 128 or 144-bit data packet
- Multiple memory banks are used to permit concurrent memory accesses with different row addresses

# Arrays of DRAM Integrated Circuits

- Similar to arrays of SRAM ICs, but there are differences typically handled by an IC called a *DRAM controller*:
  - Separation of the address into row address and column address and timing their application
  - Providing $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ and timing their application
  - Performing refresh operations at required intervals
  - Providing status signals to the rest of the system (e.g., indicating whether or not the memory is active or is busy performing refresh)

# Any Questions?