

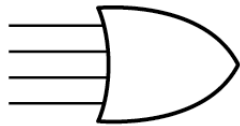
# ***Random Access Memory (RAM)***

# Overview

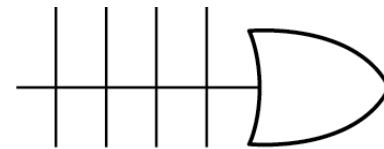
- **Memory** is a collection of storage cells with associated input and output circuitry
  - Possible to **read** and **write** cells
- **Random access memory (RAM)** contains **words** of information
- **Data accessed using a sequence of signals**
  - Leads to **timing waveforms**
- **Decoders are an important part of memories**
  - Selects specific data in the RAM
- **Static RAM loses values when circuit power is removed.**

# Preliminaries

- **RAMs contain a collection of data bytes**
  - A collection of bytes is called a **word**
  - A sixteen bit word contains two bytes
  - Capacity of RAM device is usually described in bytes (e.g. 16 MB)
- **Write operations write data to specific words**
- **Read operations read data from specific words**
- **Note: new notation for OR gate**



(a) Conventional symbol



(b) Array logic symbol

Fig. 7-1 Conventional and Array Logic Diagrams for OR Gate

# RAM Interface Signals

- Data input and output lines carry **data**
- Memory contains  $2^k$  words
  - $k$  address lines select one word out of  $2^k$
- **Read** asserted when data to be transferred to output
- **Write** asserted when data input to be stored

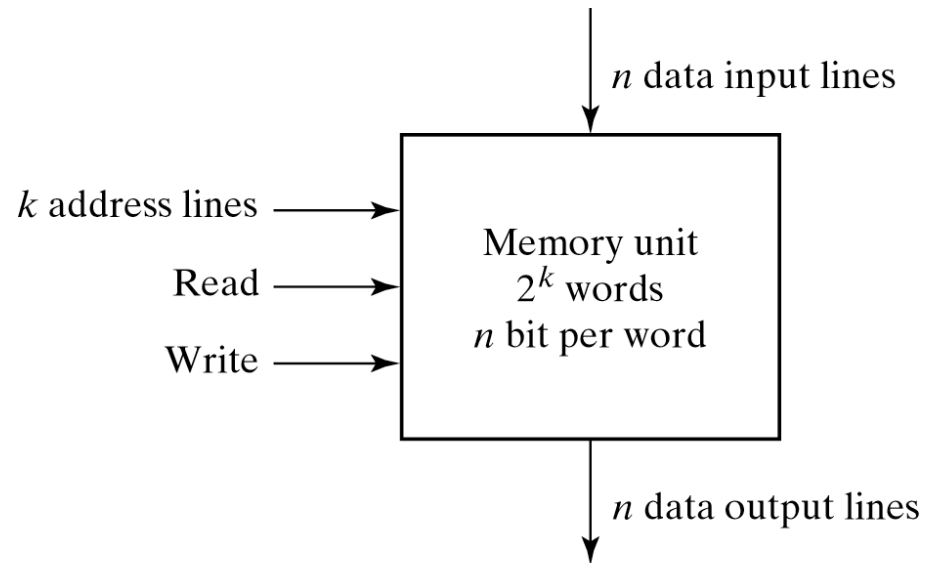


Fig. 7-2 Block Diagram of a Memory Unit

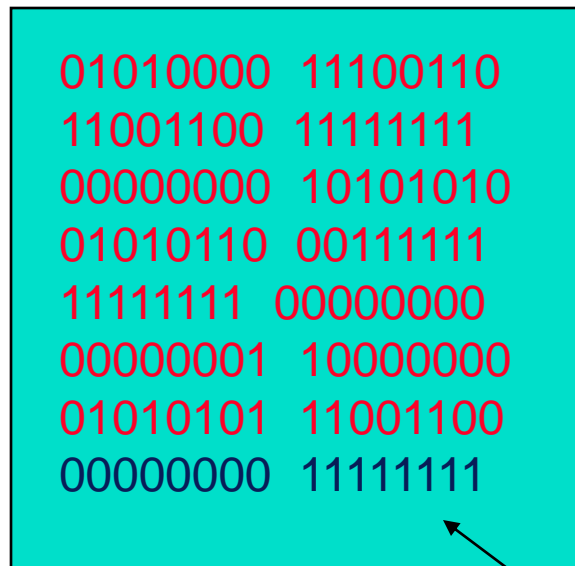
# Random Access Memory Fundamentals

## ◦ Lets consider a simple RAM chip

- 8 words of 2 bytes each (each word is 16 bits)
- How many address bits do we need?

Pick one of 8 locations

Dec	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111



16 Data and Input signals

\_\_\_\_\_ address signals

word

Each bit stored in a binary cell

# RAM Size

◦ If memory has  $2^k$  words,  $k$  address bits are needed

◦  $2^3$  words, 3 address bits

◦ Address locations are labelled 0 to  $2^k-1$

◦ Common subscripts:

◦ Kilo –  $2^{10}$

◦ Mega –  $2^{20}$

◦ Giga -  $2^{30}$

Memory address		Memory content
Binary	decimal	
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

Fig. 7-3 Content of a  $1024 \times 16$  Memory

# Write Operation

1. **Apply binary address of word to address lines**
2. **Apply data bits to data input lines**
3. **Activate write input**

Data output lines unused

Read input signal should be inactive

Delay associated with write

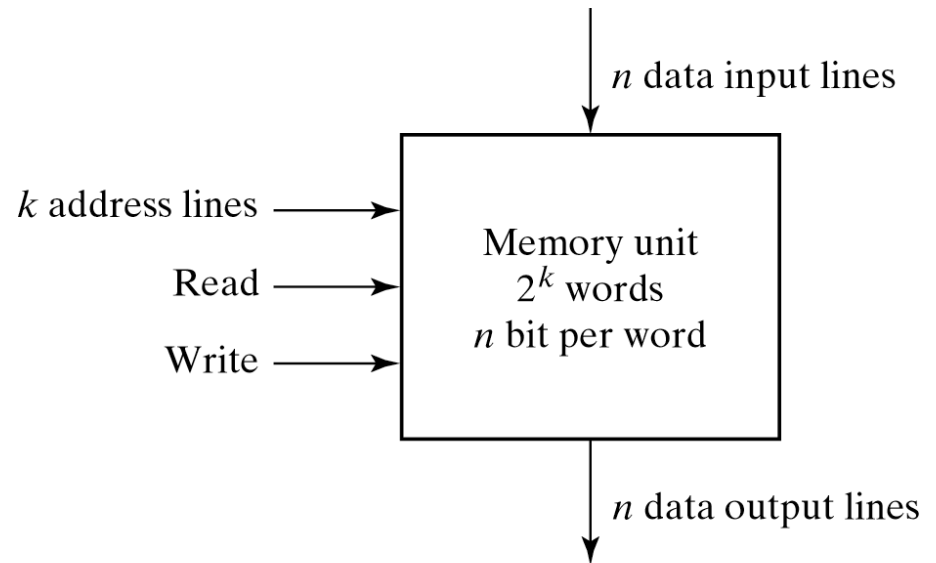


Fig. 7-2 Block Diagram of a Memory Unit

# Read Operation

**1. Apply binary address of word to address lines**

**2. Activate read input**

Data input lines unused

Write input signal should be inactive

Delay associated with read

Memory enable used to allow read and writes

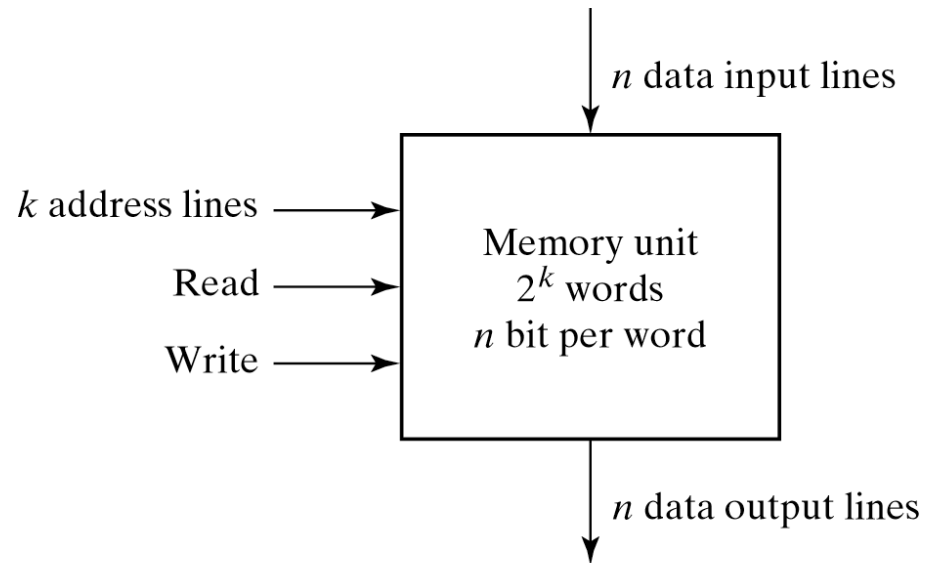
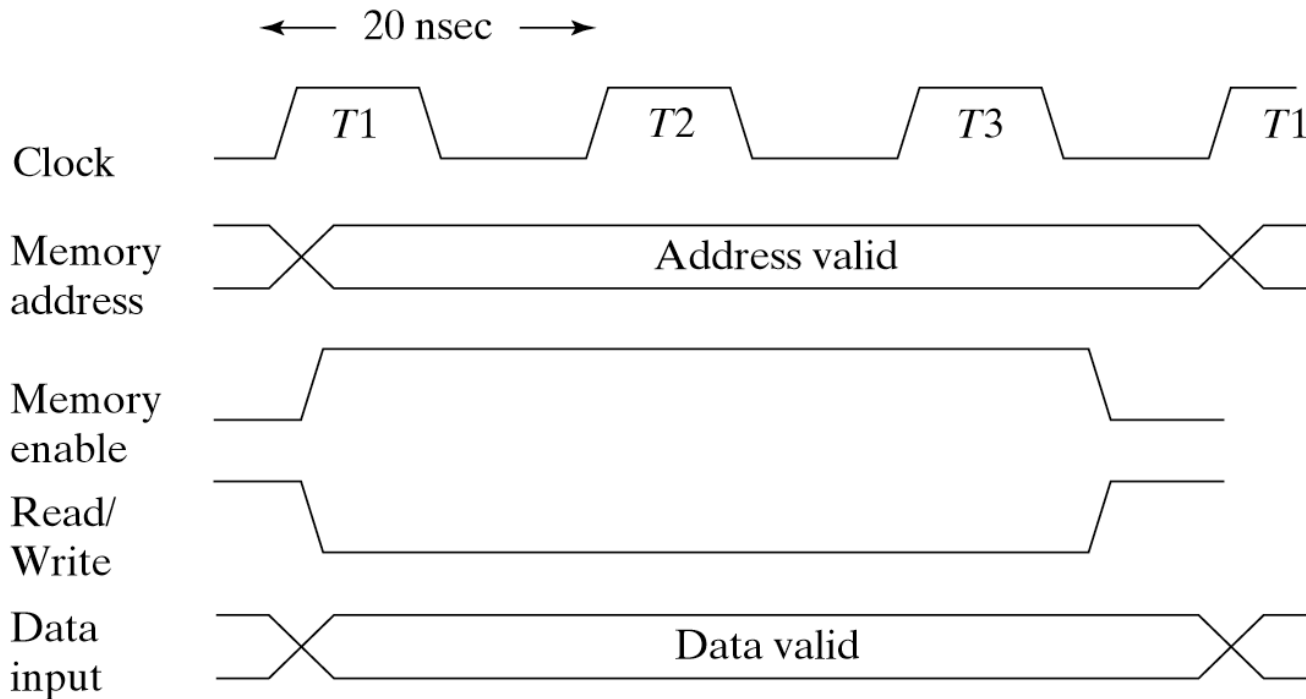


Fig. 7-2 Block Diagram of a Memory Unit



# Memory Timing – write operation

- Memory **does not** use a clock
  - Control signals may be generated on clock edges
- Cycle time – time needed to write to memory
- If cycle time is 50 ns, **3** clock edges required (**T1, T2, T3**)



(a) Write cycle

# Timing Waveforms – read operation

- **Access time** indicates time to read
- **Address** indicates location
- **Data valid on Data Output** following access time

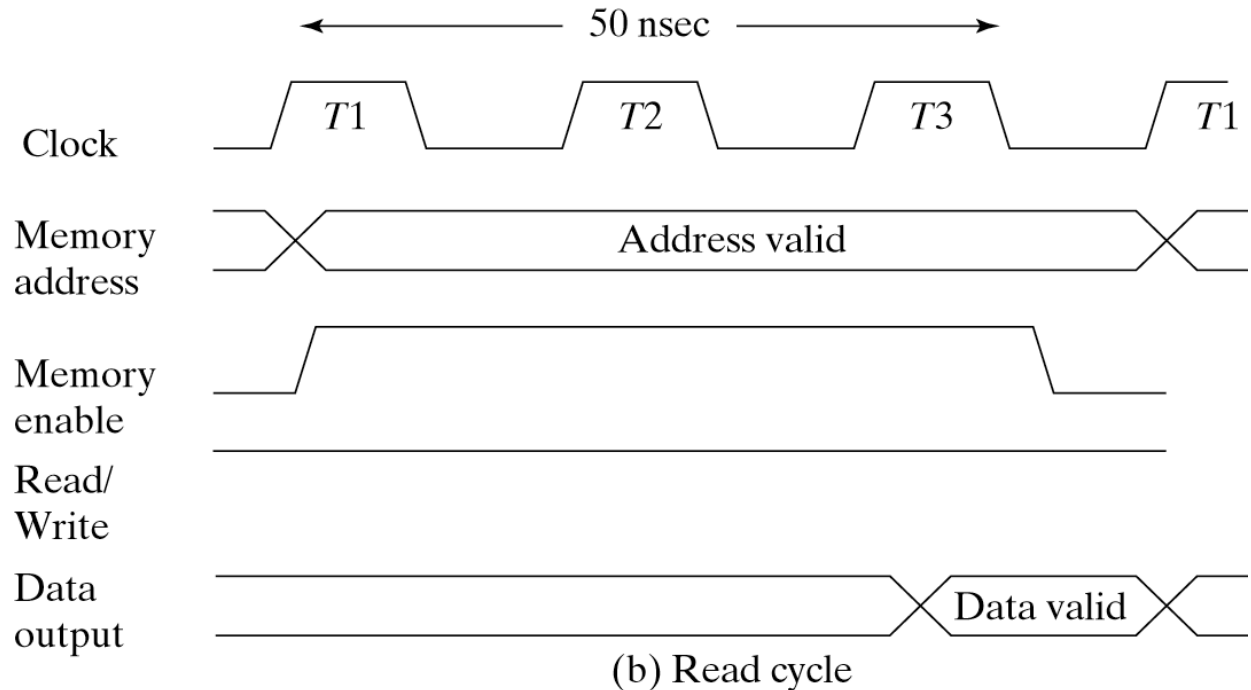


Fig. 7-4 Memory Cycle Timing Waveforms

Multiple clock signals needed for data read in this example

\* Note ordering of signals (address, mem enable)

# Comments about Memory Access and Timing

- **Most computers have a central processing unit (CPU)**
  - Processor generates control signals, address, and data
  - Values stored and then read from RAM
- **The timing of the system is very important.**
  - Processor provides data for the cycle time on writes
  - Processor waits for the access time for reads



# Types of Random Access Memories

## ◦ Static random access memory (SRAM)

- Operates like a collection of latches
- Once value is written, it is guaranteed to remain in the memory as long as power is applied
- Generally expensive
- Used **inside** processors (like the Pentium)

## ◦ Dynamic random access memory (DRAM)

- Generally, simpler internal design than SRAM
- Requires data to be rewritten (refreshed), otherwise data is lost
- Often hold larger amount of data than SRAM
- Longer access times than SRAM
- Used as **main memory** in computer systems

# Inside the RAM Device

- **Address inputs go into decoder**
  - Only one output active
- **Word line selects a row of bits (word)**
- **Data passes through OR gate**
- **Each binary cell (BC) stores one bit**
- **Input data stored if Read/Write is 0**
- **Output data driven if Read/Write is 1**

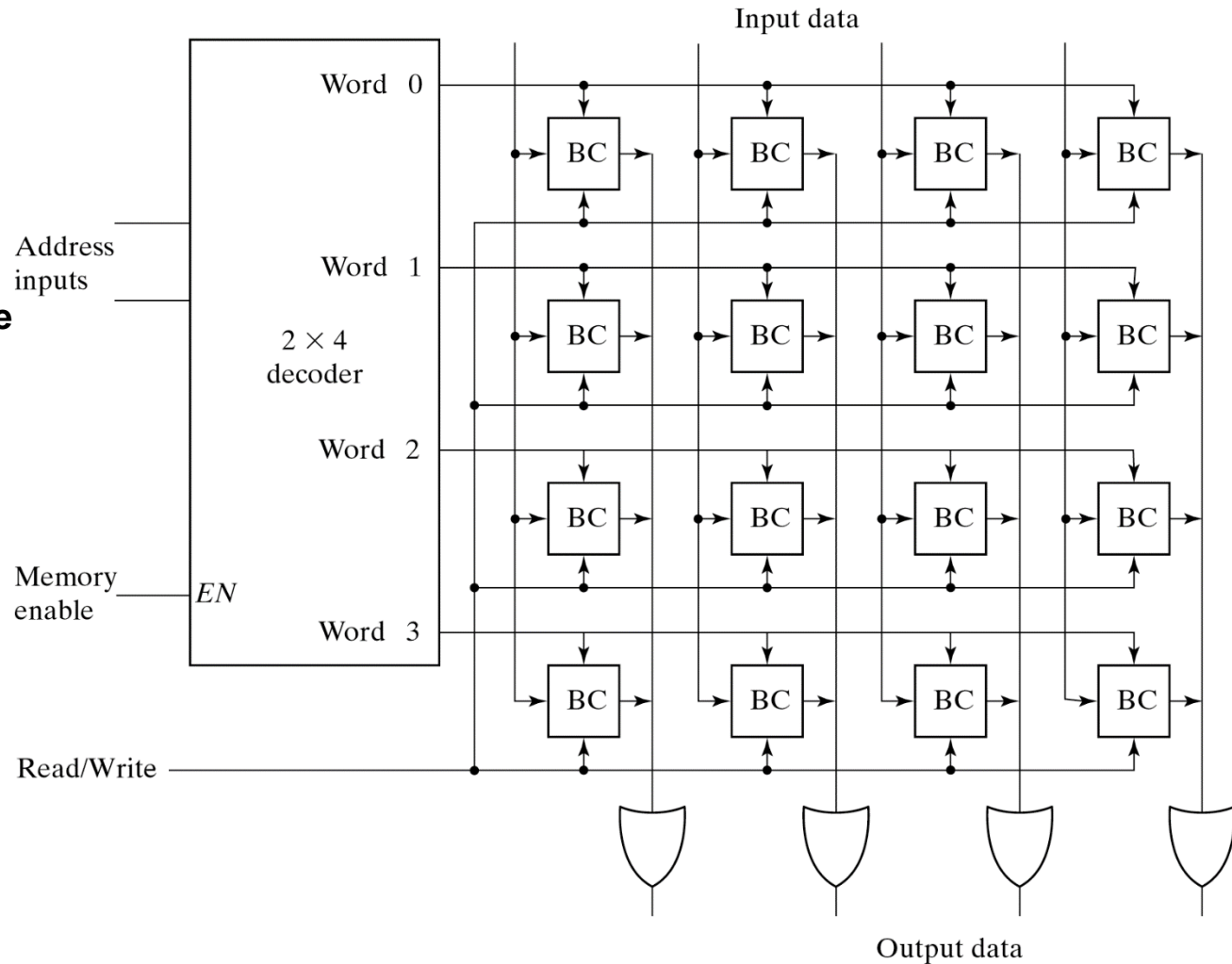
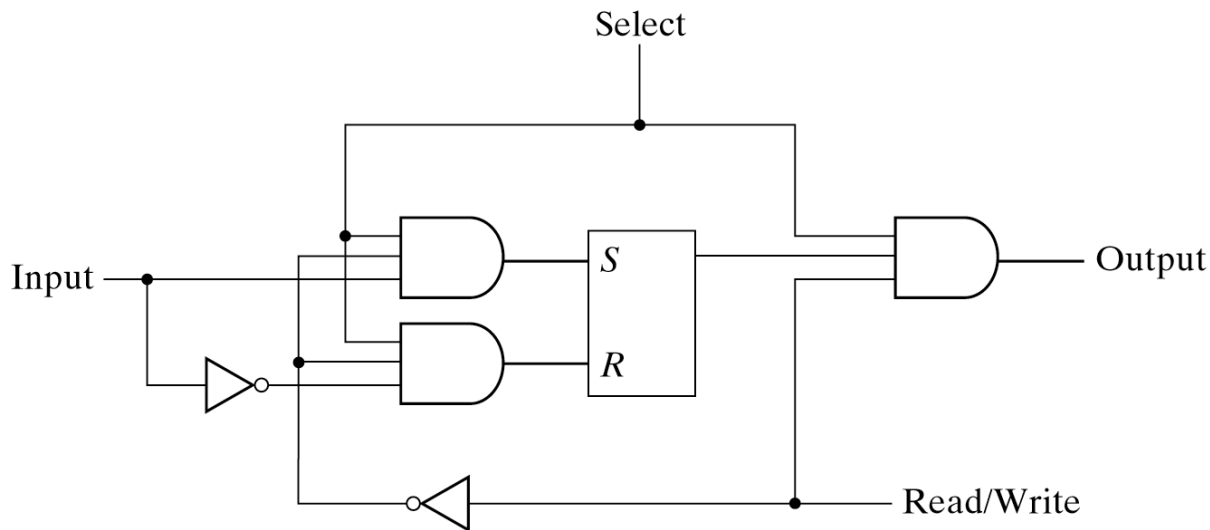


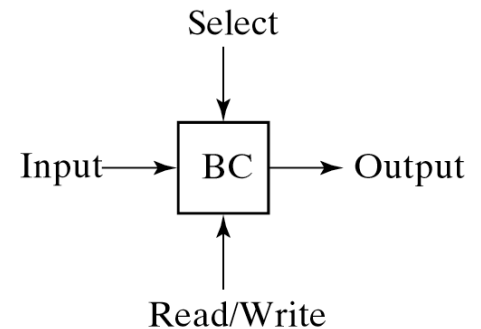
Fig. 7-6 Diagram of a 4 × 4 RAM

## Inside the SRAM Device

- Basis of each SRAM cell is an S-R latch
- Note that data goes to both S and R
- Select enables operation
- Read/write enables read **or** write, but not both



(a) Logic diagram

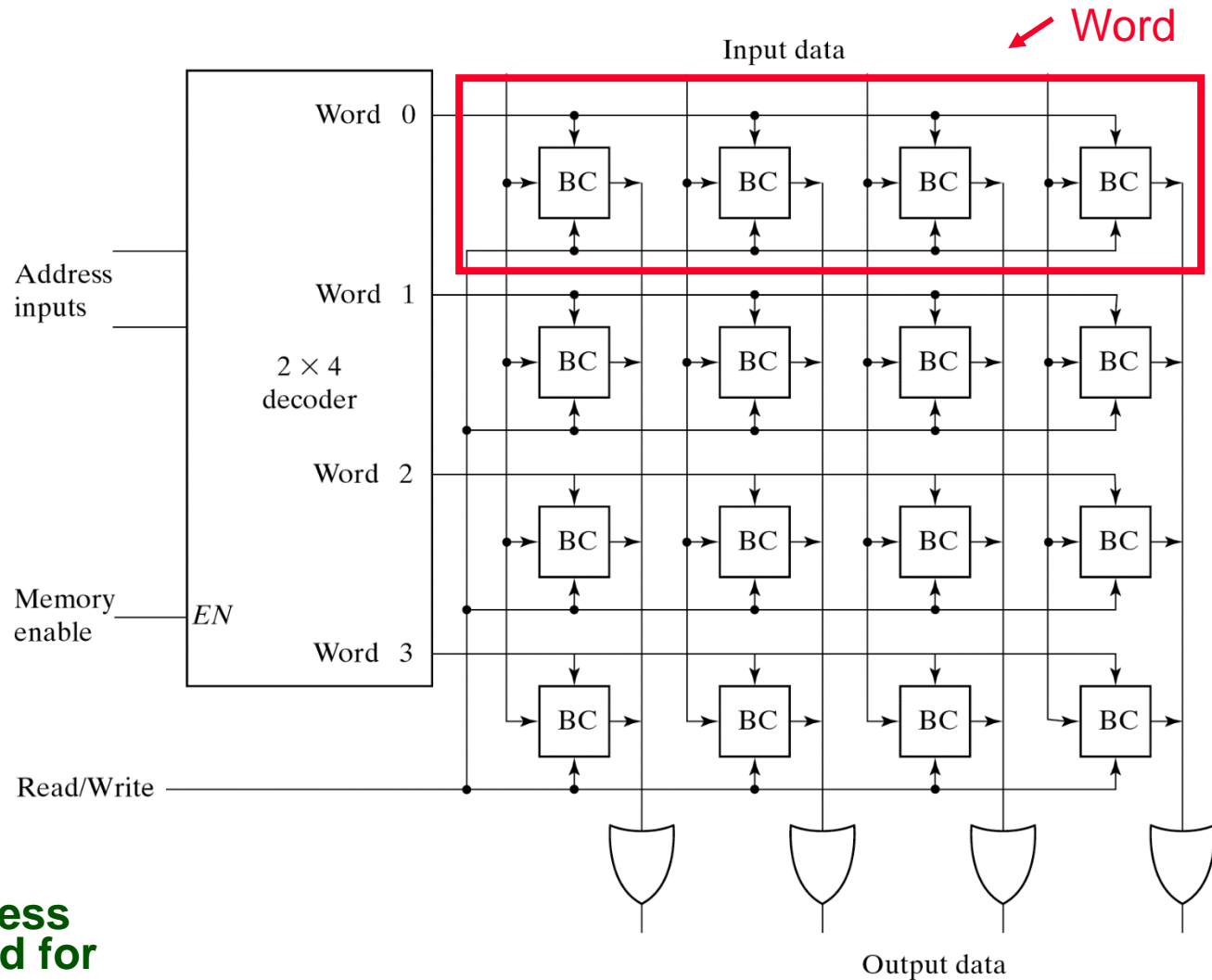


(b) Block diagram

Fig. 7-5 Memory Cell

# Inside the SRAM Device

- **Note: delay primarily depends on the number of words**
- **Delay not effected by size of words**



- **How many address bits would I need for 16 words?**

Fig. 7-6 Diagram of a 4 × 4 RAM

# Summary

- **Memories provide storage for computers**
- **Memories are organized in words**
  - Selected by addresses
- **SRAMs store data in latches**
  - Accessed by surrounding circuitry
- **RAM waveforms indicate the control signals needed for access**
- **Words in SRAMs are accessed with decoders**
  - Only one word selected at a time