

Module 6 : Semiconductor Memories

Lecture 27 : Basics of Semiconductor Memories

Objectives

In this lecture you will learn the following

- Introduction
- Memory Classification
- Memory Architectures and Building Blocks
- Introduction to Static and Dynamic RAMs

27.1 Introduction

Semiconductor based electronics is the foundation to the information technology society we live in today. Ever since the first transistor was invented way back in 1948, the semiconductor industry has been growing at a tremendous pace. Semiconductor memories and microprocessors are two major fields, which are benefited by the growth in semiconductor technology.

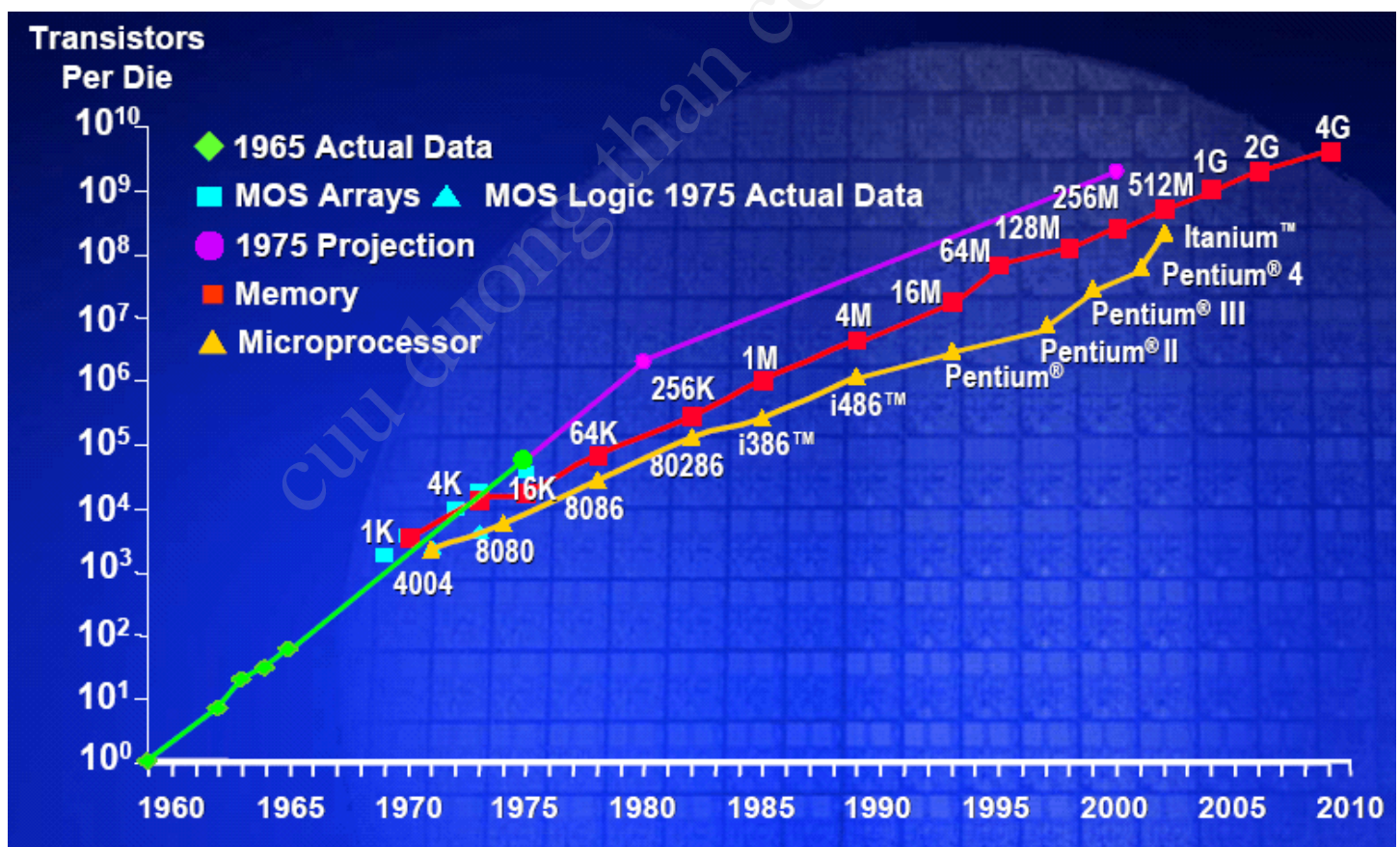


Fig 27.11: Increasing memory capacity over the years

The technological advancement has improved performance as well as packing density of these devices over the years. Gordon Moore made his famous observation in 1965, just four years after the first planar integrated circuit was discovered. He observed an exponential growth in the number of transistors per integrated circuit in which the number of transistors nearly doubled every couple of years. This observation, popularly known as **Moore's Law**, has been maintained and still holds true today. Keeping up with this law, the semiconductor memory capacity also increases by a factor of two every year.

27.2 Memory Classification

- **Size:** Depending upon the level of abstraction, different means are used to express the size of the memory unit. A circuit designer usually expresses memory in terms of bits, which are equivalent to the number of individual cells need to store the data. Going up one level in the hierarchy to the chip design level, it is common to express memory in terms of bytes, which is a group of **8** bits. And on a system level, it can be expressed in terms of words or pages, which are in turn collection of bytes.
- **Function:** Semiconductor memories are most often classified on the basis of access patterns, memory functionality and the nature of the storage mechanism. Based on the access patterns, they can be classified into random access and serial access memories. A random access memory can be accessed for read/write in a random fashion. On the other hand, in serial access memories, the data can be accessed only in a serial fashion. **FIFO** (First In First Out) and **LIFO** (Last In Last Out) are examples of serial memories. Most of the memories fall under the random access types.

Based on their functionalities, memory can be broadly classified into Read/Write memories and Read-only memories. As the name suggests, Read/Write memory offers both read and write operations and hence is more flexible. **SRAM** (**S**tatic **RAM**) and **DRAM** (**D**ynamic **RAM**) come under this category. A Read-only memory on the other hand encodes the information into the circuit topology. Since the topology is hardwired, the data cannot be modified; it can only be read. However, **ROM** structures belong to the class of the **nonvolatile** memories. Removal of the supply voltage does not result in a loss of the stored data. Examples of such structures include **PROMs**, **ROMs** and **PLDs**. The most recent entry in the filed are memory modules that can be classified as nonvolatile, yet offer both read and write functionality. Typically, their write operation takes substantially longer time than the read operation. An **EPROM**, **EEPROM** and **Flash** memory fall under this category.

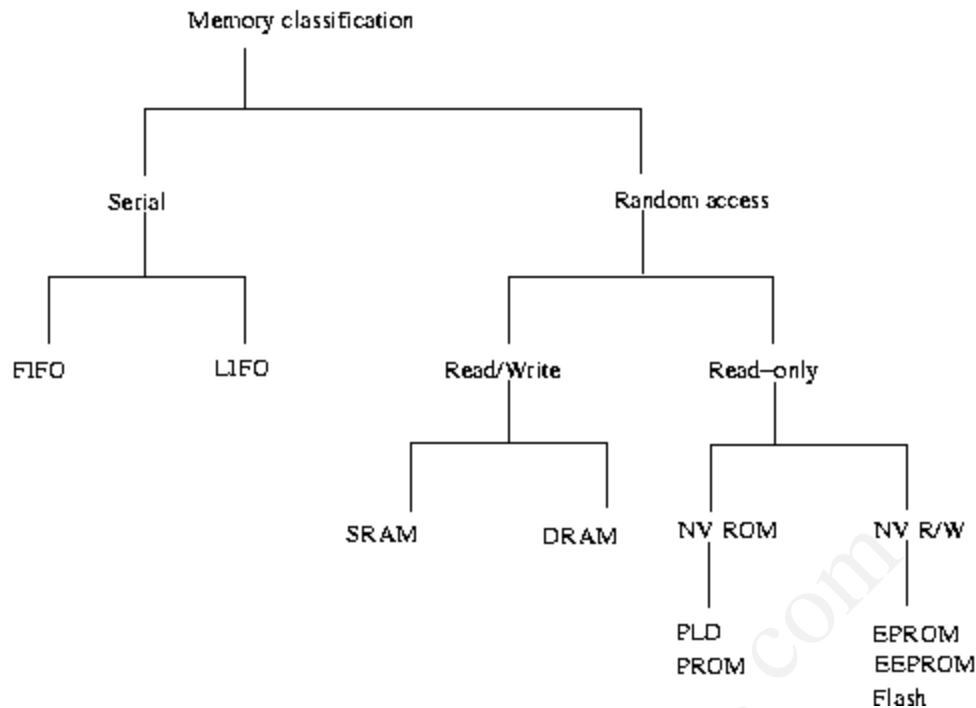
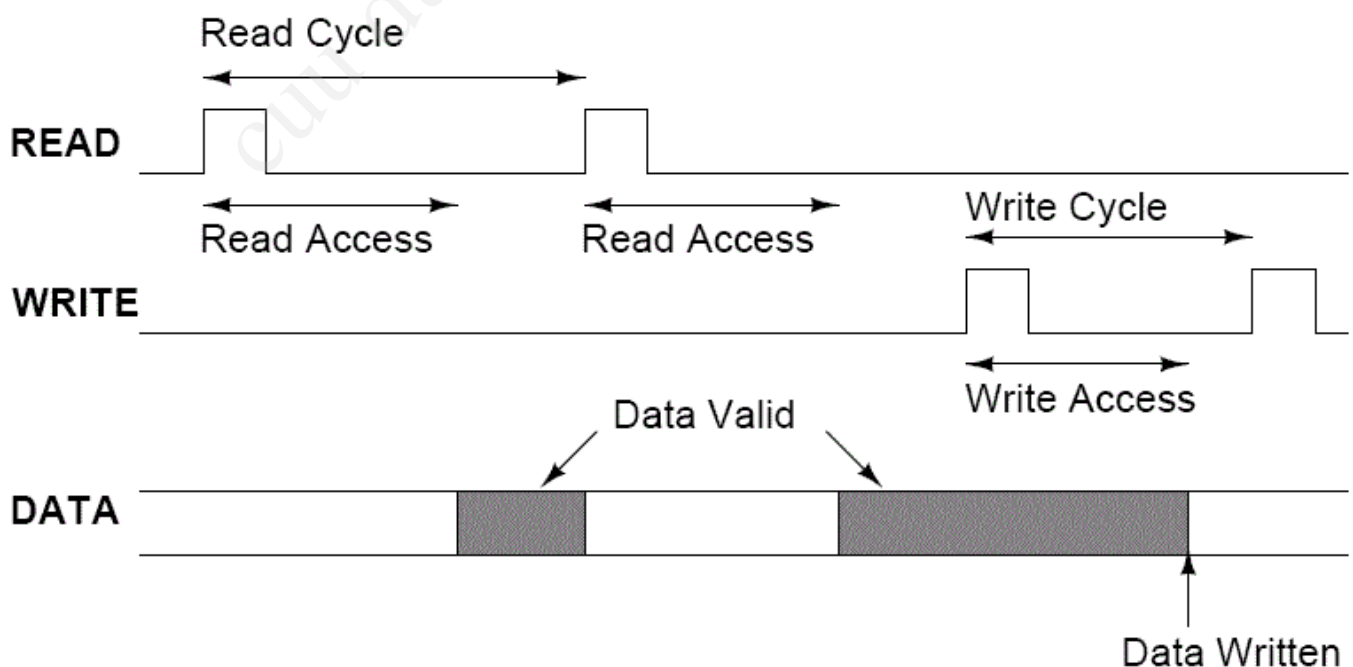


Fig 27.21: Classification of memories

- Timing Parameters:** The timing properties of a memory are illustrated in Fig 27.22. The time it takes to retrieve data from the memory is called the read-access time. This is equal to the delay between the read request and the moment the data is available at the output. Similarly, write-access time is the time elapsed between a write request and the final writing of the input data into the memory. Finally, there is another important parameter, which is the cycle time (read or write), which is the minimum time required between two successive read or write cycles. This time is normally greater than the access time.



27. 3 Memory Architecture and Building Blocks

The straightforward way of implementing a N-word memory is to stack the words in a linear fashion and select one word at a time for reading or writing operation by means of a select bit. Only one such select signal can be high at a time. Though this approach as shown in Fig 27.31 is quite simple, one runs into a number of problems when trying to use it for larger memories. The number of interface pins in the memory module varies linearly with the size of the memory and this can easily run into huge values.

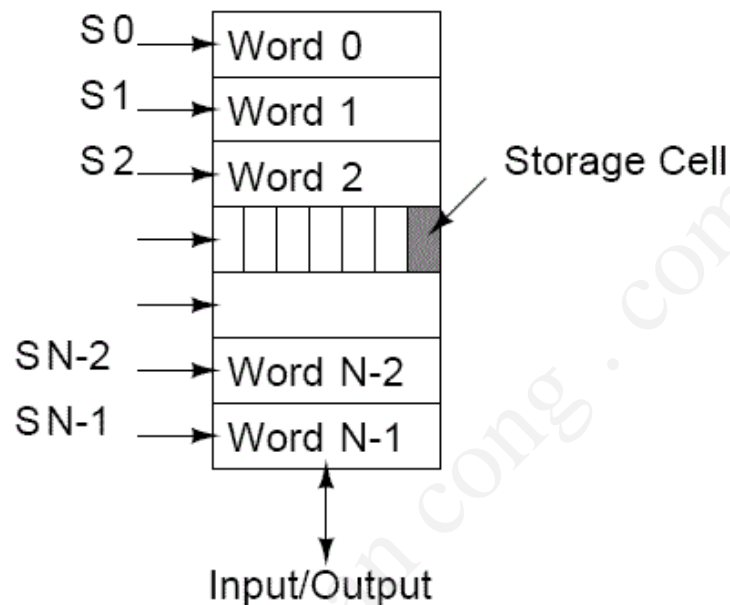


Fig 27.31: Basic Memory Organization

To overcome this problem, the address provided to the memory module is generally encoded as shown in Fig 27.32. A decoder is used internally to decode this address and make the appropriate select line high. With ' k ' address pins, 2^k number of select pins can be driven and hence the number of interface pins will get reduced by a factor of $\log_2 N$.

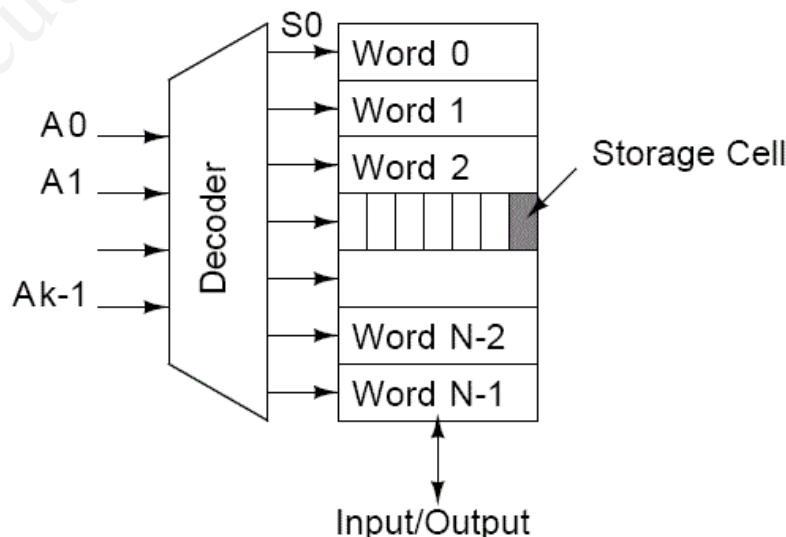


Fig 27.32: Memory with decoder logic

Though this approach resolves the select problem, it does not address the issues of the memory aspect ratio. For an **N**-word memory, with a word length of **M**, the aspect ratio will be nearly **N:M**, which is very difficult to implement for large values of **N**. Also such sort of a design slows down the circuit very much. This is because, the vertical wires connecting the storage cells to the inputs/outputs become excessively long. To address this problem, memory arrays are organized so that the vertical and horizontal dimensions are of the same order of magnitude, making the aspect ratio close to unity. To route the correct word to the input/output terminals, an extra circuit called column decoder is needed. The address word is partitioned into column address (**A₀ to A_{K-1}**) and row address (**A_{K-1} to A_{L-1}**). The row address enables one row of the memory for read/write, while the column address picks one particular word from the selected row.

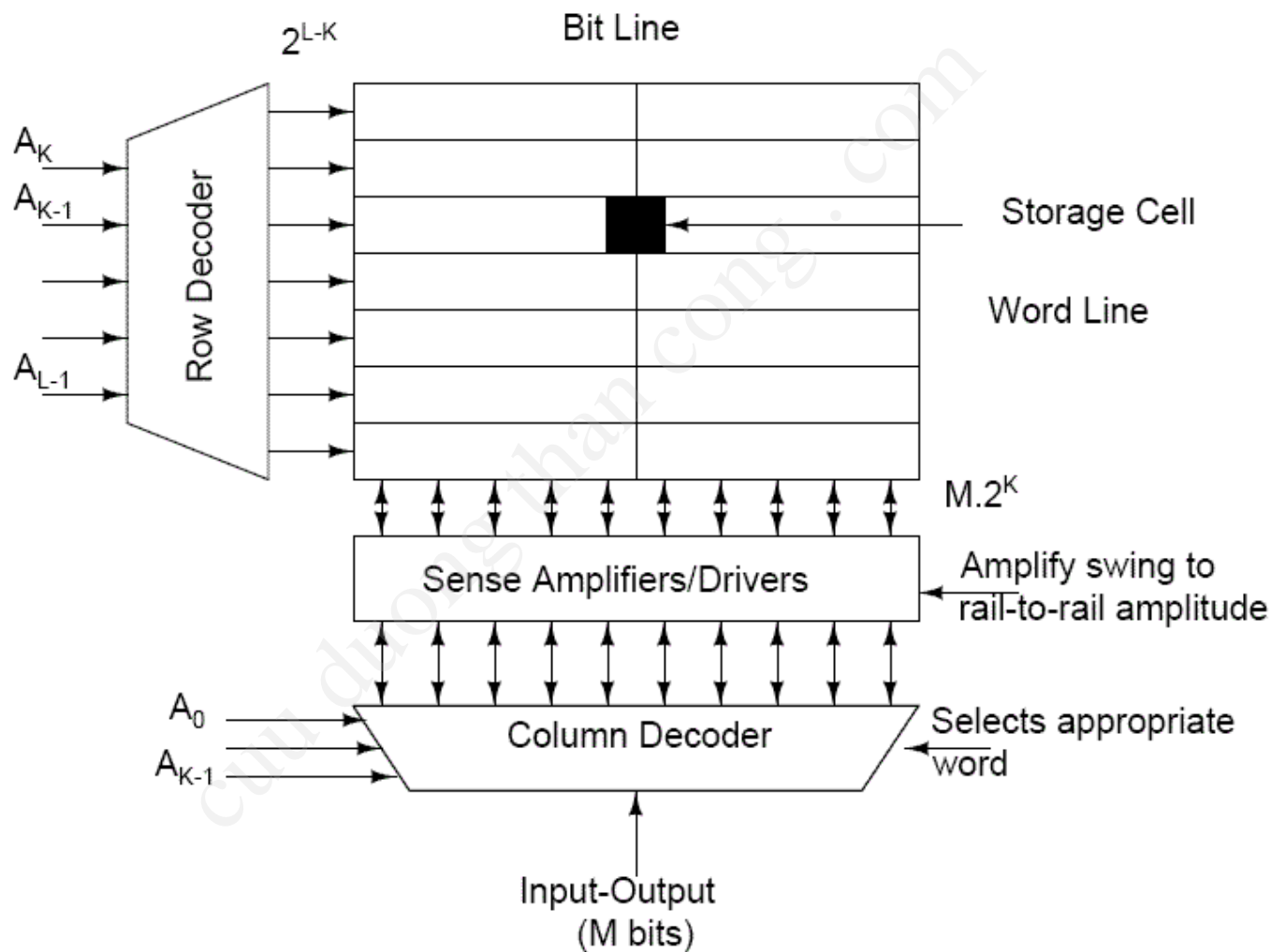


Fig 27.33: Memory with row and column decoders

27. 4 Static and Dynamic RAMs

RAMs are of two types, static and dynamic. Circuits similar to basic **D** flip-flop are used to construct static RAMs (**SRAMs**) internally. A typical SRAM cell consists of six transistors which are connected in such a way as to form a regenerative feedback. In contrast to DRAM, the information stored is stable and does not require clocking or refresh cycles to sustain it. Compared to DRAMs, SRAMs are much faster having typical

access times in the order of a few nanoseconds. Hence SRAMs are used as level 2 cache memory.

Dynamic RAMs do not use flip-flops, but instead are an array of cells, each containing a transistor and a tiny capacitor. '0's and '1's can be stored by charging or discharging the capacitors. The electric charge tends to leak out and hence each bit in a DRAM must be refreshed every few milliseconds to prevent loss of data. This requires external logic to take care of refreshing which makes interfacing of DRAMs more complex than SRAMs. This disadvantage is compensated by their larger capacities. A high packing density is achieved since DRAMs require only one transistor and one capacitor per bit. This makes them ideal to build main memories. But DRAMs are slower having delays in the order tens of nanoseconds. Thus the combination of static RAM cache and a dynamic RAM main memory attempts to combine the good properties of each.

Recap

- In this lecture you have learnt the following
- Introduction
- Memory Classification
- Memory Architectures and Building Blocks
- Introduction to Static and Dynamic RAMs

Congratulations, you have finished Lecture 27.