CHAPTER 2
MEMORY ORGANIZATION

Section 1. Organization of Main Memory

Memory is one of the major components of a computer system. More precisely, we mean a relatively large, fast memory used for program and data storage during the computer's operation. To help satisfy the desire for speed, this memory is organized so that every storage unit can be accessed with a speed that is constant and independent of the unit's location. Memory with such an organization is known as *random access memory* (or RAM).

- The memory's storage units, known as *words*, normally have the same size; a word comprised of 8 bits is called a *byte*.

- Communication between a memory unit and its environment is achieved through control lines, address selection lines, data input lines and data output lines.

    a. The *control lines* specify (at least) whether a read or write operation is to be performed;

    b. The *address lines* select a particular word out of all those available;

    c. The *input lines* supply information to be stored;

    d. The *output lines* supply the information coming out of memory.

These differences in functions among lines should be viewed logically rather than physically however since in some implementations it is common for some of the lines to be used for several functions (e.g. using the same lines for input and output, using one line for designating a read or write operation; or using the same lines for data and addresses).

Now that we know what a register is we can view RAM (once again, logically, but not necessarily physically) as a collection of registers together with the associated circuitry needed to transfer information in and out of these registers. The main components of memory now become are:

1. the registers.

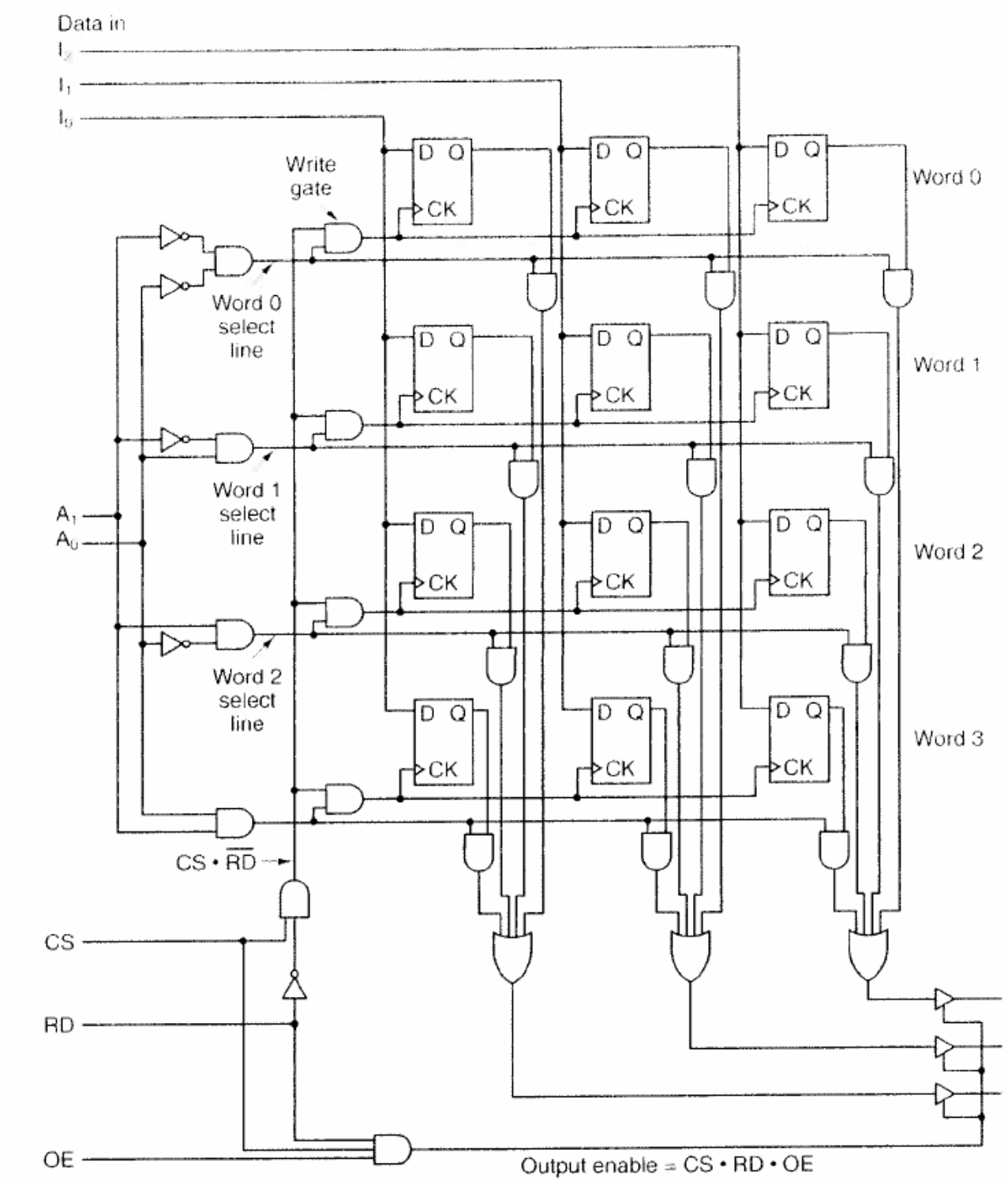
2. a decoder for selecting one register out of all those in the memory.

We then need to combine the control lines, address lines, and data lines with these components to make the memory unit function as desired. The diagram on page 2 of these notes gives a possible organization for a small memory of four 3-bit words and the same as that on page 176 of your textbook.

A memory of this type, essentially implemented from latches or flip-flops, is known as a ***static random access memory***, or SRAM. There is another way of implementing memory, known as dynamic RAM, or DRAM, that is actually the way most computers' memory is implemented, that we describe later. Operationally, however, the two technologies are the same.

The details of the registers implementing each word are shown and from this we see these registers are simply parallel-in, parallel-out registers with the loading of data being controlled by simultaneous edge-triggering of each input bit.

The address lines A_1 and A_0 give the binary address (0 - 3) of the word on which a given operation is to be performed, while the control lines CS, RD, and OE whether a read or write operation is to be performed, or neither. In this configuration the control line CS, (for Chip Select, also known as chip enable), in effect disables the operation of the memory unit by preventing both read and write operations. The signal RD is used when a read operation is to be performed; a read operation will be performed when the line has value 1. While a value of 0 on the RD line could suggest a write operation, this is not what happens. Instead a write is performed only when RD = 0 and a value of 1 appears on the OE line (which stands for Output Enable). Likewise, CS must be active (= 1).


Data in
I_2
I_1
I_0
Write gate
D Q
CK
Word 0
Word 1
Word 2
Word 3
Word 0 select line
Word 1 select line
Word 2 select line
A_1
A_0
CS • $\overline{RD}$
CS
RD
OE
Output enable = CS • RD • OE

A curious feature of the memory example is the use of the "buffer looking devices" connecting the four-input OR gates to the output lines O_0, O_1, and O_2. These are examples of what are known as" tri-state buffers" and their purpose is to make the buffer act in the normal manner when a control input is high, but to act like an open circuit when the control input is low. Technically we don't need these tri-state buffers since we show separate lines for input and output, but in practice the same lines are used for both, in which case the tri-state buffers prevent the output values from interfering with the input values on a read operation. The following tables show some of the various tri-state devices available and their operation.

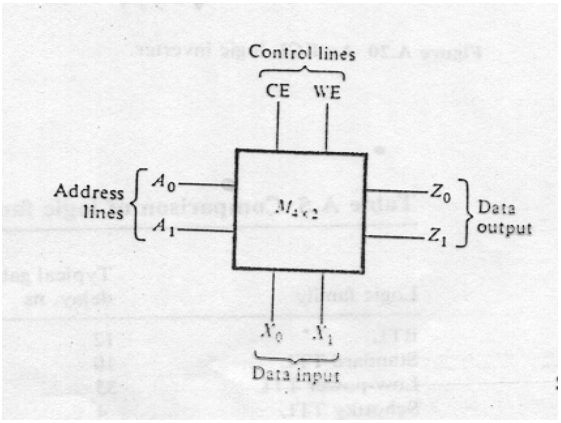
tri-state buffer (active high)		
E	A	X
0	X	HI-Z
1	0	0
1	1	1

tri-state inverter (active high)		
E	A	X
0	X	HI-Z
1	0	1
1	1	0

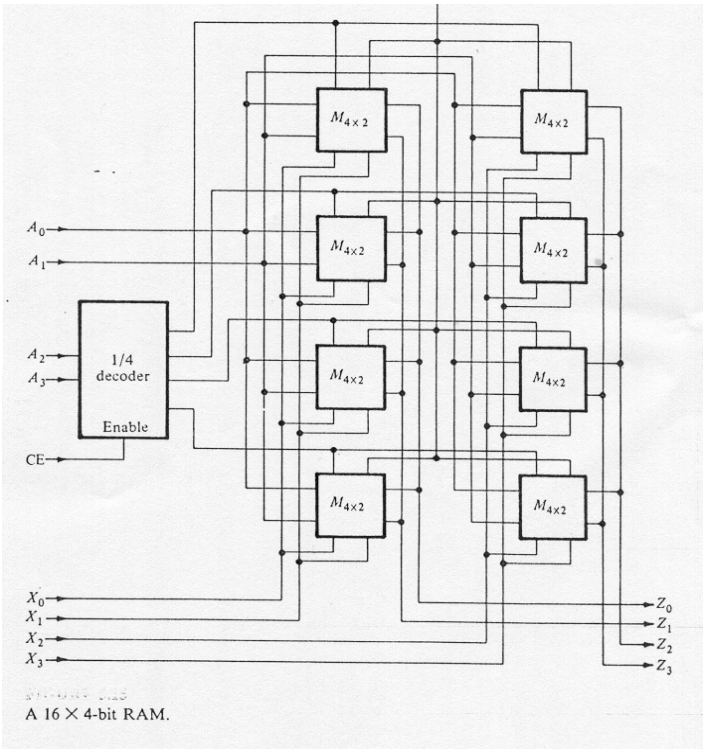
tristate buffer (active low)		
E	A	X
0	0	0
0	1	1
1	X	HI-Z

Forming Larger Memory Units from Smaller Ones

The following diagram shows a 4 × 2 RAM that uses a chip enable (CE) line and a write enable(WE) line to carry out memory operations, with a low signal on WE indicating a read operation.



We can then combine these to form a 16 x 4 RAM as follows:



A 16 × 4-bit RAM.

In another organization, the memory addressing logic regards the address space as two-dimensional rather than linear.

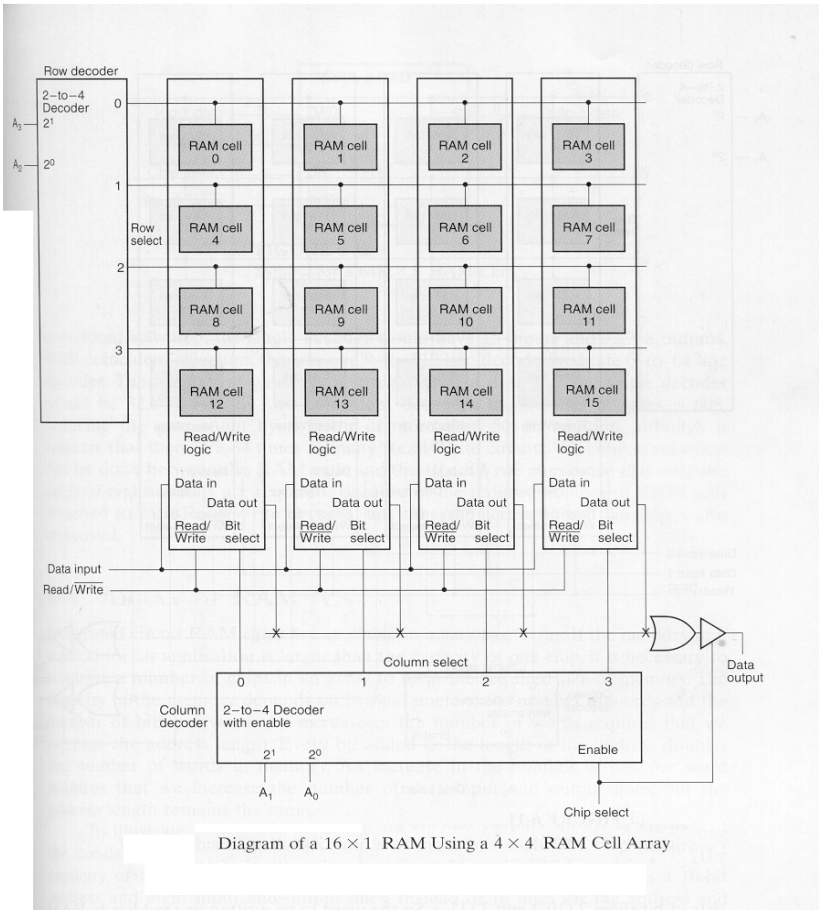


Diagram of a 16 × 1 RAM Using a 4 × 4 RAM Cell Array

Although we do not supply descriptions for all of the lines, by now you should be able to discern their role. You will notice, however, that both the row decoder and the column decoder use address lines labeled A_1 and A_0 rather than employ four address lines A3, A2, A1, A0 and distribute them between the row and column decoders. This is not unusual, as large memories will use the same lines for row and column addresses to save pins and reduce package costs. To distinguish the two types of addresses, a pair of signals called *row access strobe* (RAS) and *column access strobe* (CAS) are also provided to signal the memory that a row or column address is being supplied.

Example:. Consider the 4M × 1 RAM (2×2^{22} words of 1 bit1 each) and 512K × 8 RAM (2×2^{19} words of 8 bits each) diagrammed on page179 of your textbook. Note that each uses 4Mbits, but their internal organization is different as the book explains.

What are some of the ways 4 of the first type can be combined to form larger memory units? Before considering this case

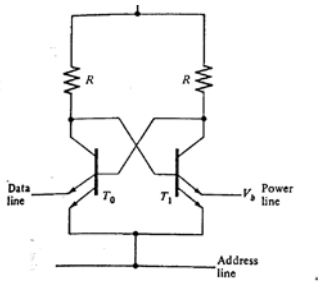
Done in class

Operational Properties of RAM

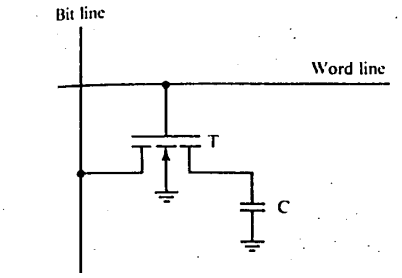
As our discussion above indicated, RAM is usually constructed as an n × m arrangement of *bits*, not an arrangement of n *words* of m bits each.

When fabricated from semiconductors two classes of RAM emerge, depending on the technology used in the construction of each.

1. *static RAM* (SRAM) - Here the bits are implemented essentially as flip-flops.



2. *dynamic RAM* (DRAM) - Here each bit is implemented using a single transistor and a single capacitor.



Operationally, they differ in that over time the capacitors used in dynamic RAM lose their charge, requiring that they be periodically "refreshed." At one time, to carry out a read operation, the capacitors must be discharged, destroying the value stored - a phenomenon known as *destructive read-out* (DRO). DRO requires that each read be followed by a restoration of the bit's original value. Static RAM is non-destructive during read operations (NDRO) and requires no refreshing. Both types of memory however are *volatile*, meaning the values stored in each bit will be destroyed if power to the memory is disrupted.

Static RAM is faster than dynamic RAM. Not only does static RAM require less time to complete a read or write operation (a property known as the RAM's *access time*), but as we noted above, static RAM requires neither refreshing nor restoration after a read. One consequence of this NDRO property is that the time which must elapse between successive read or write operations (known as the memory's *cycle time*) is the same as its access time; for dynamic RAM the cycle time is necessarily greater than its access time.

The major advantage of dynamic RAM over static RAM is that a bit of dynamic RAM requires only one transistor and one capacitor so that one can pack significantly more bits of DRAM into a given area than SRAM, and DRAM consumes less power than a bit of static RAM.

Because of the high-bit density achievable with dynamic RAM, it is currently the type used for a computer's main memory. Where greater speed, but fewer bits, are required (e.g. in *cache memories*, or in virtual memory tables) static memory is preferred.