CPS311 Lecture: Memory Devices

Last Revised October 16, 2019

Objectives:

1. To introduce various characteristics of semiconductor memory chips: word size, capacity, number of data and address bits, etc.
2. To introduce various types of semiconductor memory: Static RAM, Dynamic RAM, ROM, PROM, EPROM, Flash
3. To show how semiconductor memories can be combined into complete systems
4. To overview basic characteristics of rotating disks.

Materials

1. Projectable of Dynamic RAM Cell
2. Projectable of DRAM chip showing square array of cells
3. Projectable of ROM Basic Cell
4. Projectable of EPROM Basic Cell
5. Projectable of 64 KB system built from 8-bit wordsize chips
6. Projectable of 64 KB system built from 2-bit wordsize chips
7. Projectable of memory module
8. Projectable of 16 GB system build from 1G x 4 chips
9. Projectable of Disk Head

I. Introduction

A. We already seen how many different types of information can be represented by appropriate binary bit patterns. This allows computer systems to store and process numbers, text, graphics, sounds, and video.

B. In the VonNeumann computer model, one of the major components was the memory. We now consider various technologies that are used for this purpose in modern computers.

C. A memory technology can be characterized by:

1. Cost per unit of storage (often expressed in bits or bytes - be very careful to notice which unit is being used, since there is an 8:1 difference!).

2. Whether the memory is capable of being written and read under program control, or whether it is read only, with data be written into it either at manufacture time or by a separate step outside of normal program operation.

   a) Often, read-only memory is called ROM, which stands for read-only memory.

   b) In the case of semiconductor memory, read-write memory is often called RAM (which stands for random access memory, distinguishing it from technologies like disk) - but this actually involves a misnomer since ROM is also typically random-access. However, limiting the term RAM to read-write memories is a well-established practice.

3. Volatility: is the contents lost when the power is turned off (or in case of a power failure)?

4. Power requirements.

   a) In the case of volatile memory, a further refinement of the question is to ask whether the memory system require ssignificant power at all times, or only when it is being accessed. (All volatile technologies require some power at all times, but in some cases this can be much smaller when the memory is not being accessed)

   b) Power requirements are, of course, a particularly important characteristic for memories used in battery-powered devices.

5. Density - how much memory can be put on a single package (chip or disk).

   a) For semiconductor technologies, this is actually closely related to power requirements, since the power applied to the memory is ultimately turned into heat. The need to dissipate this heat is a dominant limiting factor on density.

b) Also, for semiconductor technologies, due to manufacturing and installation costs, this also impacts the cost per bit of a type of memory.

6. Access time - the delay between the time a particular item to be read is requested and the time it becomes available.

7. Write time - for some read-write technologies, this is the same as access time, while for certain other types, it is much longer. (Of course, if the memory is not writable under program control, this is irrelevant.)

8. Cycle time - this includes access time plus any "recovery" time needed before accessing the next item. For most technologies, cycle time is the same as access time, but for one very important type of semiconductor technology it is about twice as long.

9. Transfer rate - the rate at which large quantities of data can be transferred once the beginning of a data area has been accessed.

D. At the present time, the technologies primarily being used are of two kinds:

1. Semiconductor technologies.

2. Various kinds of rotating disks (magnetic or optical)

3. Other technologies either are or have been used - e.g. magnetic tape (still very much in use for backup and archival storage) and magnetic cores (of historical interest.) But we will focus on the two technologies just listed - realizing that at some point in the future some other approach may be developed and used (e.g. technologies based on biological cells).

E. A few observations about the two broad categories:

1. Rotating disks are cheaper than semiconductor technologies - often dramaticalliy so.

2. Many - but not all - semiconductor technologies are volatile. Disk technologies are not.

3. In the case of both categories, density improvements over time have been huge: orders of magnitude. But rotating disk density's are much higher than those for semiconductor technologies.

4. One major difference between semiconductor and rotating disk technologies concerns access time (and write time), where the difference is orders of magnitude (e.g. < 1 ns for the fastest semiconductor technology, while access times for disk technologies are typically measured in milliseconds and also depend on the exact location of the data being accessed).

II. Overview of Semiconductor Memory Technologies

A. Generally, a computer system will have some number of semiconductor memory chips (typically on the order of several to dozens) implementing various technologies.

B. There are also monolithic chips available which place a complete memory system (sometimes using several different technologies) on the same chip as the CPU. Though we will develop our discussion assuming separate memory chips, most of what we say also applies to the memory portion of a monolithic system.

C. We've listed above a number of metrics for comparing technologies. In the case of a memory chip, two other issues are important

1. Its WORD SIZE: The unit of data that is read from or written to the chip in a single operation. This may be:

   a) A single bit

   b) 8 bits (one byte)

c) Some other size, such as 2 bits, 4 bits, etc.

d) Notice that the word size of a memory chip need not be (and generally is not) the same as the word size of the memory system on which it is used. (But it is not greater than the word size of the system, since that would mean some memory on the chip is not accessible.) If the word size of the system is bigger than the word size of the individual chips, then the system is configured so that several chips are accessed in parallel to provide a complete word. For this reason, we will use the terms "chip word size" and "memory system word size" to distinguish the two.

Example: A memory system transfers data in units of 64-bit words, but the chips from which it is constructed use 4 bit chip words. The system will be configured so that each operation accesses 16 chips, each of which is responsible for 4 of the 64 bits in the complete operation.

2. Its CAPACITY - can be expressed either in bits or in words - e.g

128 meg (chip words) x 4 bit = 512 megabit. This might be described as a 128 meg x 4 chip or as a 512 megabit chip.

a) In advertising, chip capacities are typically expressed in bits (makes them sound bigger!) [Note: B = byte; b = bit - be sure to note which is being used!]

b) Often complete memory systems or individual memory modules (eg. DIMMS) are built out of multiple chips, with the capacity generally expressed in bytes. In many cases, only a subset of the total chips in a system will be accessed in any given operation.

D. A memory chip will have, in addition to power and ground pins, the following external connections:

1. A number of data lines, dictated by its chip word size - e.g. if the chip's word size is 4 then it will have 4 bit lines for data in and data out. These can be either separate lines for each purpose (so 8 total), or a single line for each bit that can go each way.

2. A number of address lines. To uniquely address each word on the chip, we need a number of address bits equal to $\log_2$ of the capacity in words - e.g. a 256 M word chip would need 28 address lines.

   Large capacity chips typically use half as many address lines as would be otherwise needed, and require that transmission of the address bits to the chip be split in two, with half the address being put on the lines in the first part of the memory cycle and the other half in the second part of the cycle. (This is often the case with DRAM (dynamic RAM) because of their high capacity and the way the chip is internally organized.) This is called ADDRESS MULTIPLEXING.

3. Various control lines - eg:

   a) Chip select - indicates that the chip is to participate in the current operation.

   b) Read/write control lines - indicates whether the operation being done is a read from the memory or a write to the memory.

4. Example: a 128 K x 8 bit ROM chip might have:

   8 data lines
   17 address lines
   1 chip select line
   1 output (read) enable line

5. Example: a 4G x 1 bit DRAM chip might have

   1 data line for read and one data line for write (or 1 serving both purposes)
   16 address lines (multiplexed to form a 32 bit address)
   1 chip select line
   1 output (read) enable line
   1 write enable line

6. It turns out that pinout is often a determining factor in chip configuration - e.g. a chip for 32 bit words would require 32 pins for data in/out and thus would not be practical to manufacture in a sufficiently small container. Hence memory systems supporting more than 8 bits per word are typically composed of multiple chips, with several being accessed at the same time.

7. Actually, unless you are a manufacturer, you will probably not purchase memory in the form of individual chips, but rather in the form of MEMORY MODULES consisting of several chips on a single circuit board that plugs into a PC motherboard.

   Example: A 8 GB memory module may consist of 16 (or 18) 4G bit chips. (18 would allow an extra parity bit per byte.)

E. There are six basic types of semiconductor memory, distinguished by how the data is actually stored. (With variations on the basic theme). We will consider each in turn. It turns out that 2 are read-write (RAM) technologies and 3 are read-only (ROM) and 1 (Flash) is used both as a form of read-only memory and for long-term storage of files as part of the file system rather than the memory.

III. Volatile Semiconductor Memory Technologies

A. Static RAM (SRAM)

1. Stores data in individual flip-flops on the chip.

2. Data can be both written and read.

3. SRAM is the fastest semiconductor memory technology (on the order of a few ns access time for the fastest separate chip designs - equal to the CPU cycle time when on the same chip as the CPU (so < 1 ns)

4. Static RAM has fairly high power consumption, because one side of each flip flop is always conducting. This means that heat dissipation is a key limiting factor in chip capacity.

   SRAM chip capacities are not increasing as rapidly as DRAM because power dissipation is the key limiting factor in density. For example, in 1995 the largest SRAM chip listed in a catalog I consulted had 1 megabit capacity; when I revised these notes (almost 25 years later - the largest chip I found was 72 M bit and there were still lots of 1 M bit chips (and less) being offered.

B. Dynamic RAM (DRAM) stores data as a charge on an internal capacitor - one capacitor per bit.

1. This type of semiconductor memory is considerably slower than static RAM, but also allows much greater bit capacities per chip and hence lower cost.

2. Data can be both written and read.

3. The basic DRAM bit cell is quite simple:

   PROJECT: MOS Dynamic RAM cell - Bartee Page 269

a) A MOS field effect transistor behaves basically like a simple switch. When its gate is biased appropriately, current may flow in either direction between its source and drain. Otherwise, the path from source to drain is effectively an open circuit. In this example, the gate is connected to a select line.

b) Data can be written to the cell shown by placing an appropriate value on the data line and activating the select. Depending on the value on the data line, current will flow either into or out of the capacitor, leaving it charged or discharged as appropriate.

c) Data can be read from the cell by connecting the the data line to a sense amplifier. When the word line is activated, any charge present in the capacitor will flow down the data line to the amplifier and be detected as a pulse; of course, if the capacitor is discharged to begin with no pulse will be produced. Note that this read operation is destructive; it is necessary to rewrite the data back to the cell after it has been read. (This is handled automatically by circuitry on the chip during the interval between one access and the next, but results in a total cycle time that is about twice the access time.)

d) To write data, first the existing charge on the cells is removed by an operation similar to read (but is ignored). Then the desired data (1 or 0) is placed on the data line, and the word line is activated. If the data line is 1, the capacitor is charged, otherwise it is left discharged.

e) Because both read and write operations actually involve both reading and re-writing, it is common for DRAM chips to also support what is called a "read-modify-write" functionality, which can be done in the same time as a simple read or write. Data is read (destructively), but then the write-back step writes the new data rather than re-writing the old data. [ISA's that allow a memory location to be both a source and a destination of an operation - e.g. IA32 - reflect this capability].

f) The individual memory cells on a chip are generally organized as a matrix, with the location of an individual cell specified by a row number and a column number.

Data is read or written to the cell array in units of an entire row at a time, and the chip has an on-chip buffer that can hold a complete row of data.

PROJECT - Picture of DRAM chip showing square array.

g) Because of the way data is organized on the chip, most DRAM chips use multiplexed addressing (which also conserves pins). For example, consider a 1 G x 1 chip, with 15 address pins ($\log_2 1\ G$) / 2

(1)first, the address of the desired row (15 bits) is placed on the address pins, and a row is read.

(2)then, the number of the desired column (15 bits) is used to select a column within the row, which can be either read or written or modified. The two halves of the address are called the "row address" and the "column address".

(3)The best-case time needed to access a row of data for 5 Volt chips has remained relatively constant at 50-60 ns.

4. When CPU clock speeds were on the order of 16 Mhz or less, DRAM memory could keep up with the CPU in terms of accessing a location in memory within the time taken by one CPU cycle. (At 16 MHz, the CPU cycle time is 66 ns). However, as CPU speeds have increased, there has developed a widening gap between CPU speeds and memory speeds.

a) A number of strategies have been developed to address this problem - some of which we'll talk about in conjunction with our discussion of memory hierarchies later in the course.

b) For now we note, though, that one strategy to help narrow the CPU / memory speed gap relies on the fact that often uccessive locations in memory are accessed succesively

(1)Programs typically occupy successive locations in memory.

(2)Data structures such as objects and arrays occupy successive locations in memory.

(3)As we shall see later when we talk about cache memories, a cache sometimes use lines whose size is several main memory words.

(4)When successive locations in memory are being accessed one after another, average access time in DRAM can be speeded up by reading the row containing the locations to be accessed and then performing multiple accesses to different columns in the on-chip buffer. Different ways of handling this give rise to different variants of DRAM, such as:

(a)Fast Page Mode (FPM) DRAM

(b)Extended Data Output (EDO) DRAM

(c)Synchronous DRAM (SDRAM)

(d)Double Data Rate DRAM (DDR SRAM) and variants (DDR2, DDR3)

(e)etc.

(5)It is common to find memory modules given speed ratings in MHz (e.g. 133 MHz or 1066 MHz). This is the speed of the bus with which the module is designed to be used, with a single block of data (often 64 bits) being transferred on each bus clock. This means that memory technologies that support fast access to multiple words in the same row can transfer the second and subsequent memory-words of successive accesses much faster than the first transfer.

Note that, the bus clock is slower than the CPU clock - so each unit of data transferred corresponds to several CPU cycles. (Sometimes an order of magnitude difference).

c) Chips designed for lower voltage operation (e.g. in battery-powered devices) can achieve much better access times.

d) Because no power is used except when data is being written and read, DRAM uses much less power per bit. This, coupled with the fact that he basic bit cell is also simpler than that for SRAM, makes this the best technology for high capacity chips. Currently, chips with up to 4G bits per chip are widely used, with larger capacity chips also available. (DRAM is also available in smaller capacity chips, but for small memories SRAM is often preferred for simplicity.) [historically, what I have had to do when I revise these notes every two years is multiply the DRAM capacities in my examples by 2 to 4]

e) Because charge slowly leaks from the capacitors, the data in the memory must be refreshed periodically - typically once every few ms. Since the read operation is destructive and requires that the chip rewrite its data internally, to refresh a location on the chip, it is sufficient simply to read it periodically. Since data is read/refreshed a row at a time, it is sufficient to ensure that each row of the chip is accessed often enough to guarantee that its data is not lost. But since guaranteeing this by the logic of a program is virtually impossible, systems using DRAM must incorporate some provision for doing special refresh cycles on all the rows on a regular basis. Thus, even if the program goes many minutes without accessing the a given row, its data will still be kept refreshed.

IV. Non-Volatile Semiconductor Memory Technologies

A. Mask-programmable ROM (ROM) stores data as the presence or absence of actual physical connections on the chip. Each bit's transistor has a connection which, if present, will cause the bit to be read as one value and, if absent, as the opposite value. (Typically transistor present is read as 0 and transistor not present as 1).

PROJECT ROM Basic Cell

1. The data is thus programmed into the chip at manufacture, and cannot be altered thereafter.

   a) This is done by preparing a special mask used in one of the last production steps to control where a layer of metal is or is not deposited.

   b) The customer specifies the desired pattern of 0's and 1's when the chip is ordered. This specification is generally submitted in some machine-readable form. The manufacturer, in turn, has automatic equipment that converts this data into a mask for use in producing the chips.

2. Mask-programmable ROM is only practical for applications calling for 1000's of chips containing the same data - e.g. production runs of commercial products.

   Example:Many embedded systens contain their software in a ROM that was programmed during manufacturing. (Though sometimes an alterable form is used to allow bug fixes in the field)

B. Field programmable ROM (often called PROM) is similar in principle to mask programmed ROM, except for how it is programmed.

1. The chip is built similarly to a mask-programmable ROM, but with all connections to its bit transistors in place.

2. Each connection includes a fusible link (narrow part that can be burned out by passing a higher than normal current through it. This can be done in the field by using a special device called progammer or "burner" to burn out the undesired links.

   a) During programming, the power supply voltage is raised above its normal value. This activates a special switch on the chip which reverses the direction of the data lines so that data can be fed into the chip over the lines normally used for output.

   b) Then, each location on the chip is addressed in turn, as if for a normal read cycle, but for a longer time.

   c) As each location is addressed, a large current is applied to the data output pin of a bit that is to have its fusible link blown. This current flows backward through the output data pin and the fusible link and burns it out.

   d) Once programmed, a PROM cannot be altered; if it contains incorrect data it must be discarded and a new PROM burned.

3. PROM is not used in mass-produced systems due to the programming required - instead mask-programmable ROM is used. Manufacturers typically offer mask-programmable ROMs and PROMs with identical pinouts and performance characteristics. During product development, PROMs will be used until all the bugs are (hopefully) out of the software. For production equivalent mask-programmed ROMs will be used. (Sometimes a correctly programmed PROM is submitted with the order for ROMs and the manufacturer derives the mask directly from it.)

C. Erasable PROM (EPROM), behaves similarly to ROM and PROM - but its operating principles are quite different.

1. Like Dynamic RAM, it stores data in the form of electric charge in an on-chip capacitor.

2. However, with EPROM the capacitor is insulated from the rest of the circuit, and so cannot be charged or discharged during ordinary operation. As a result, the EPROM behaves much like a fusible-link PROM.

   a) The heart of the cell is a MOS transistor with a floating gate - i.e. there is no external connection to the gate. In fact, it is surrounded by insulating material.

      PROJECT EPROM basic cell

   b) Current flow between the source and drain of a MOS transistor is modulated by the charge on the gate. In the absence of charge, no current can flow; when charge is present, it can flow. As the chip is structured, an uncharged gate corresponds to a stored value of 1, and a charged gate to a stored 0. In the erased state, all gates are uncharged, and so all cells on the chip will read as 1's. This is also the case for nonexistent memory. (Recall what you observed about nonexistent memory in Lab 1!)

   c) The problem, of course, is that with a floating gate there appears to be no way to put a charge on the gate in order to write a zero into a cell! However, the insulating material around the gate is made thin enough that, under higher- than-normal voltages, it can temporarily break down without being permanently damaged. This is what happens during programming: using a higher-than-normal voltage allows the insulation around the gate to be breached so that charge can be transferred to the gate. However, when the programming voltage is removed, the insulation is restored and the charge is trapped.

      (1)Before programming, all gates are uncharged, so all locations on the chip contain 1.

      (2)During programming, those bits to that should contain a 0 receive a high voltage pulse that charges the gate.

(3)This is handled by special circuitry on the chip that senses the high programming voltage a special programming pin. At this time, the data output pins are converted into inputs. The address pins select a byte on the chip, and any bit whose data pin has a zero on it receives the programming voltage.

d) To erase the chip, ultraviolet light is allowed to shine on it. This knocks the charge off those gates that are charged, and restores the chip to the unprogrammed (all 1's) state. This is done through a quartz window in the package directly over the semiconductor

(1)Except during programming, this window is kept covered by a label to prevent gradual deterioration of the data from ordinary light. Thus, in ordinary use the chip behaves like a ROM; but if errors in programming are discovered they can be corrected without discarding the chip. (This is good for initial R & D work on a new system.)

(2)Over a long period (10 years) some leakage of charge can occur; thus other types of ROM are preferable for long-term permanent data storage.

D. There are various forms of electrically erasable programmable ROM (EEPROM, EAROM, Flash memory) that are similar to EPROM, except that the charge may be both stored and erased during normal system operation.

1. These have characteristics of both ROM and RAM. Like ROM, they are nonvolatile - once written the memory will not lose its data even if power is totally turned off; like RAM, it can be written to under program control.

2. These technologies are slower than other types of memory for writing (which can take time on the order of milliseconds.)

3. Flash memories are often used in situations where otherwise an EPROM might be used to allow "in the field" reprogramming of a device such as a cell phone or various components of an automobile. This reprogramming operation is called "reflashing".

   Example: A previous car I owned had to have its electronic ignition system reflashed to address a software problem that caused cars of that year to behave incorrectly in certain rare situations and there is a recall out on my model year of a car I currently own to reflash a the entertaintment system!

4. As you know, flash memories are also widely used in devices such as USB memories, digital cameras, etc, and more recently as an alternative to magnetic disk for file storage on laptops. Though flash memory is slower than other semiconductor memory technologies, it is much faster and lighter in weight than magnetic disk, while still being nonvolatile but still being writable when one intends to do so.

V. Organizing Semiconductor Chips into a Complete Memory

A. When we looked at the structure of a VonNeumann machine, we saw that one of its major building blocks was the memory system.

   1. The memory can be regarded as an array of numbered slots, each of some fixed size. The slots are numbered 0, 1 ... - where the number of a slot is called its PHYSICAL ADDRESS. (The size of the slot is fixed by the design of the memory system.)

   2. Each slot in memory has its own phyical address, and the memory system provides the primitive operations of `read(physical_address)` and `write(value, physical_address)`.

   3. The size of a physical slot in the memory may not be the same as the addressable unit of the ISA of the CPU. (In modern systems it is often more but never less.)

For example, it is common today to find a system with a byte-addressable CPU ISA being implemented using a memory system based on 64 bit physical words (or even 128), with a data path connecting the CPU and memory that supports transferring 64 (or 128) bits at a time.

4. The CPU and the memory typically exchange information using a set of datapaths called the MEMORY BUS.

   a) This consists of a large number of wires (or PC board traces) broken into several parts.

      (1)The address bus.

      (2)The data bus.

      (3)Various control signals, which we won't discuss now.

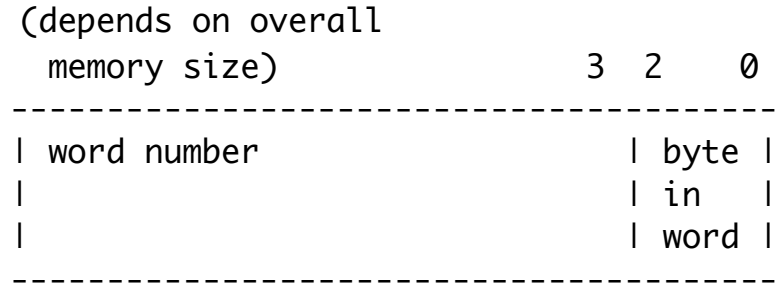
   b) To read a word from memory, the CPU puts the address of the word it wants on the address bus, sends the appropriate control signals to the memory, waits an appropriate period of time (based on the access time of the memory), and then copies the data off the data bus. (The memory is responsible for responding to the control signals by fetching the data, and putting it on the bus.)

   c) To write a word to memory, the CPU puts the address of the word it wants to write to on the address bus, puts the data to be written on the data bus, sends the appropriate control signals to the memory, and leaves the information on the busses for an appropriate period of time. This may be the access time of the memory or - typically - much less if the memory system controller contains an internal register to store the data being written so that the CPU can get on with other things. The memory system is responsible for responding to the control signals by copying the data from the data bus into the appropriate location in memory.

d) To write just a portion of a word to the memory (e.g. a single byte) iit is necessary to read an entire words, modify just the part that is to be changed, and then write the modified word back. A memory system may support this by allowing the CPU to do partial word writes that it converts to whole word accesses - but for simplicity we will not go into this further.

B. If the addressable unit of the CPU's ISA differs from the physical size of a memory slot, then some translation must be done between the address generated by a program and the physical memory address of the slot addressed in memory. This is done by splitting an address generated by a program into two parts - a word number and a byte in word.

1. For example, if a byte addressable CPU is to be used with a memory system that is organized around 64 bit words it would interpret an address generated by a program as follows;



(A memory word size of 64 bits is 8 bytes, so the low order 3 bits of the address of a byte specify the position of the byte in the word, and the remaining bits specify which word.)

a) If a program references the byte at 0x3 in memory, this will be interpreted as byte 3 in word 0.

b) If a program accesses a half word at 0x94 in memory, this will be interpreted as bytes 6 and 7 in word 11, since (94 / 8 = 11 remainder 6)

2. The task of splitting the address into a word number and byte number, and later of extracting the desired byte(s) from the word accessed, may be performed by the CPU or by the memory system. In the former case, the address sent to the memory consists of only the "word number" portion of the address. For simplicity, we will develop our examples this way by assuming that the translation is done at the CPU end.

3. In our present discussion, we will use the terms "word" and "word size" to mean a memory word and the memory word size respectively- not a word as defined by the ISA. In like manner, we will talk about an address as being the slot number for a memory word, which may be a subset of the address calculated by the CPU (e.g. the leftmost 29 bits of a 32 bit address if a byte addressable CPU is used with a 64-bt word memory system.)

C. Also of importance is the chip word size of the chips used to implement the memory system. The simplest implementation would arise if the chip word size and the memory word size are the same.

1. For example, the Microprofessor systems we used in Lab 1 use a Z80 CPU whose addressable unit is a single byte. A fully-populated memory system would be 64 KB, because the Z80 uses a 16 bit address.

a) Suppose we built a fully-populated system using 128 KBit = (16 KB) chips, organized 16 K words x 8 bits each.

(1)How many data lines will the overall system need?

ASK

8 - since the memory word size is 8

(2)How many address lines will the overall system need?

ASK

16 - since the overall system is 64 KB, and $2^{16} = 65{,}536$

b) How many chips would we need to build a 64 KB memory?

ASK

4 - since the overall memory is 64 KB and each chip is 16 KB

(1)How many data lines will each chip have?

ASK

8 - since the chips are organized 16 K x 8

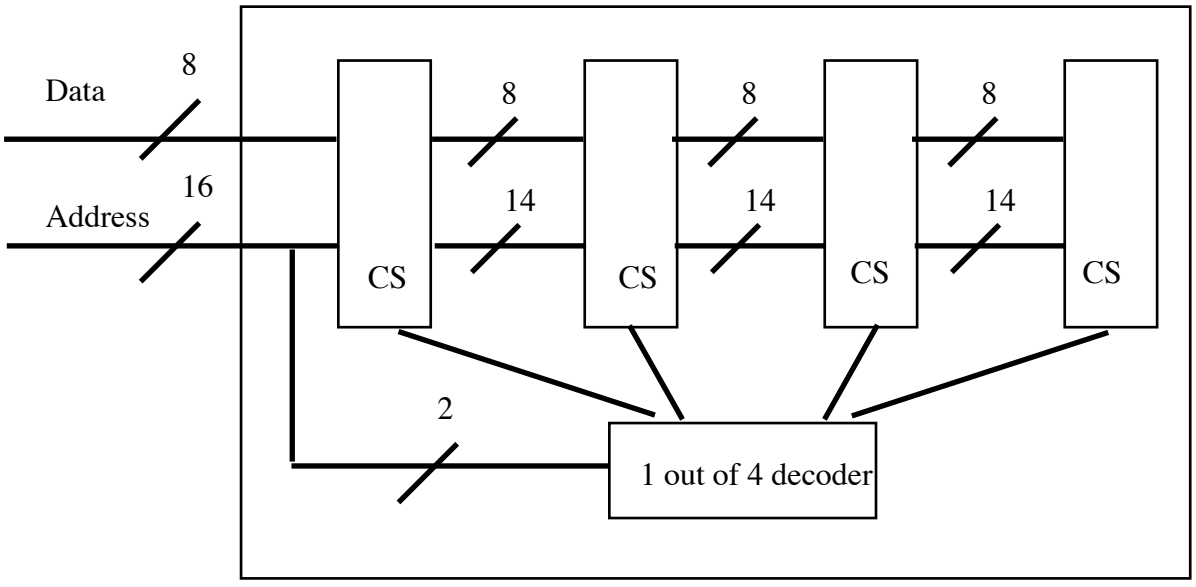
(2)How many address lines will each chip have?

ASK

14 - since each chip holds 16 K chip words and 2^{14} = 16,384

c) When we want to access a word in memory, we only need to access one chip because the chip word size is the same as the memory word size. The remaining two bits of the system adddress (16 - 14) are decoded to select which chip is accessed. (Observe that $2^2 = 4$, the number of chips).

2. The following configuration would work



PROJECT

The correct chip is selected by the decoder on the basis of the high 2 bits of the address. The outputs of the decoder go to a "chip select" input on each chip. The remaining 14 bits of the address and the 8 bits of data connect to the selected chip, which accesses the desired word.

D. If is often the case that the memory chip size is less than the memory system word size.

1. One practical reason for this is that the number of data pins a memory chip needs is at least its chip word size - so if we tried to build a memory system having 64 bit memory words out of chips having 64 bit chip words, each chip would need 64 data pins! This would result in requiring excessively large chips.

2. Now consider what would happen if we built the same system we just discussed using the same capacity chips (128 KB), but this time organized 64 K x 2 bit.

   a) How many chips would we need to build a 64 KB memory?

      ASK

      Same as before - 4 - since the overall memory is 64 KB and each chip is still 16 KB (the chip word size is 1/4 byte and 64 KB / 4 = 16 KB)

      (1)How many data lines will each chip have?

         ASK

         2 - since the chips are organized 64 K x 2

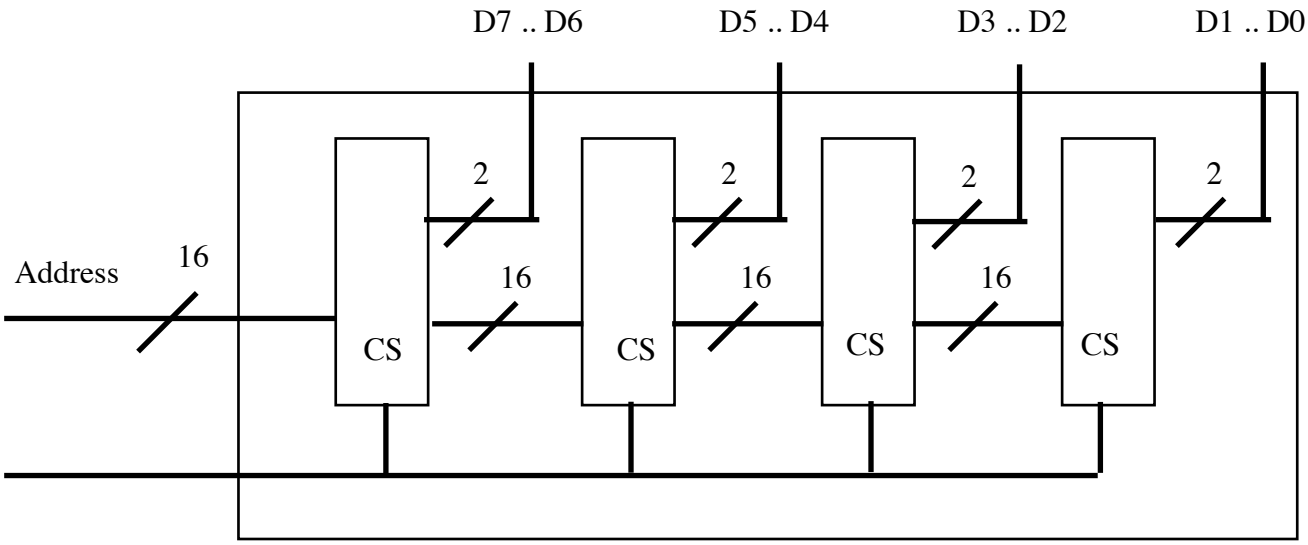
      (2)How many address lines will each chip have?

         ASK

         16 - since each chip holds 64 K chip words and 2^{16} = 65,536

b) When we want to access a word in memory, we need to access all 4 chips because the chip word size is 1/4 of the memory word size. But all the bits of the memory address go to each chip to specify which word we want.

3. The following configuration would work:



E. We have considered two implementations which illustrate different issues:

1. In the first implementation, the chip word size was the same as the memory word size, but the chip word count was 1/4 of the memory system word count. So each access to memory accessed just 1 chip.

2. In the second implementation, the chip word size was 1/4 of the memory word size, but the chip word count was the same as the system word count. So each access to memory accessed all the chips.

3. Actually, it turns out that the first of these implementations has an advantage. Suppose we wanted to use fewer than 4 chips - perhaps to save money or to use less space on a circuit board.

   a) With the first implementation, we could simply omit 1 or more chips. This means that certain memory addresses don't exist - but the ones corresponding to chips that are present would still work properly.

(Recall that you saw something like this with the Microprofessor - it actually used 3 chips of which 2 were 16 K bit and 1 was 32 K bit for a total of 64 K bits or 8 KB out of a possible 64 KB - so there were many addresses that did not exist.)

b) What would happen with the second implementation if we omitted a chip?

ASK

Every word would have a "hole" in it corresponding to the unimplimented chip!

F. The most complex situation - but one that often arises in practice - is one where both issues arise - the chip word size is a fraction of the memory word size and the chip word count is a fraction of the memory system word count. As a result, each access to memory accesses several chips - but a proper subset of the total.

1. Now consider the following situation:

a) A memory system with an 16 GB capacity and a 64 bit word size is built using 4 GBit chips organized 1 G x 4 .

(1)How many memory words are there?

ASK

Overall capacity is 16 GB. Since each word is 8 bytes this is 2 G words

(2)How big is a physical word address?

ASK

31 - since $2^{31} = 2G$

(3)How many chips are needed?

ASK

32 - A 16 GB memory is 128 G bits, and each chip holds 4 G bits.

(4)How many chip words does each chip have?

ASK

Each chip has 1G chip words

(5)How many addess lines will each chip have?

ASK

30 - since 2^{30} = 1 G

(6)How many chips will be accessed for each access to the memory system?

ASK

16 - a word of 64 bits will consist of 4 bits from each of 16 chips

b) The 32 chips are divided into two groups (banks) of 16 each. Any access to memory accesses one of the two banks.

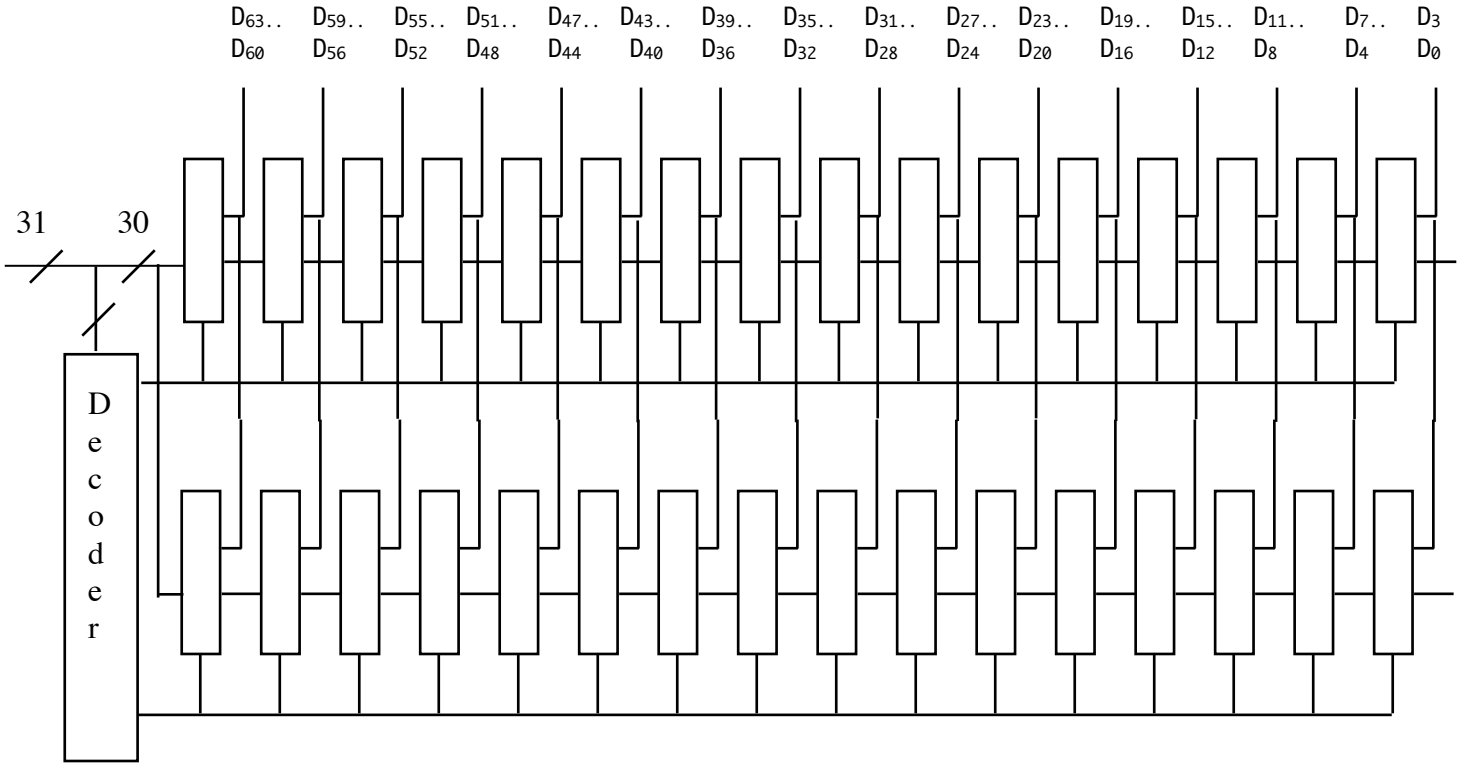
(1)Each bank comprises 8 GB of the overall 16 GB of the memory system. (16 chips x 4 G bits / chip / (8 bits / byte) = 8 GB)

(2)One bit of the physical address determines which bank is accessed. The remaining 30 bits specify a chip word - note that 2^{30} = 1G.

(3)In an actual system, each bank might be realized by a memory module card, or they might be realized by two sides of a single card.

PROJECT: Example of memory module

2. This yields the following configuration:



a) Each chip connects to four bits of the data bus.

b) The 31 bit physical address is divided into two parts. 30 bits are used to select which of the 1G chip words on each chip is accessed. The remaining bits selects which bank of chips is used.

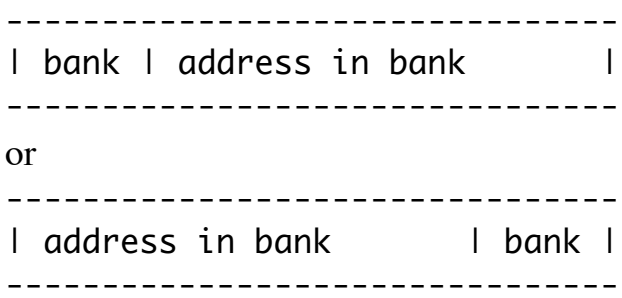
c) All chips in a given bank have chip select activated by a common line that is asserted when the "bank select" bit of the physical address lie in the range allocated to this 8 GB bank of the overall 16 GB memory system size.

d) The diagram shows the decoder as a separate component. In practice, the decoder may be distributed across the memory sticks - which means there is no decoder per se, though the function performed is the same.

3. The example I have developed uses 2 banks to achieve the desired overall size, but the same approach could be used with any (power of 2) number of banks.

   For example: a 512 M word x 64 bit/word memory (4 GB) might be realized using 128 M x 8 bit chips. A bank would need to consist of 8 chips (8 chips x 8 bits / chip = 64 bits per memory word), and a total of 4 banks would be needed to achieve the overall size of 4 GB.

4. You'll notice I haven't said anything about which bit of the physical address is used to select the bank to use. In the example just developed, the two viable possibilities are the most significant (leftmost) bit(s), and the least significant (rightmost) bit(s). The two alternatives can be diagrammed this way:

   where the width of the bank field is determined by how many banks are needed - e.g. if there are 4 banks then the bank field will be $\log_2$ #banks = 2, and the width of address in bank is the same as the number of address bits on each chip.

   a) The former is the choice made in laptops. If the address size and choice of chips called for two banks, say, it would be possible to build a usable memory using only 1 bank - meaning half of the potential addresses would not be used (so our example would be an 8 GB system and addresses above 1FFFFFFFF would not exist.)

      Laptops are often sold with the buyer having the option of adding a second bank for expansion (in the case of our example) to 16 GB, or adding a second bank purchased from a third party at the time of initial purchase or later.

b) Or the least significant bit(s) could be used for bank select. If there were two banks as in our example, the result would be that even numbered memory words (0, 2 ...) would be found in bank 0, and odd-numbered words in bank 1. In this case, omitting one of the banks would result in a system where every other memory word would not be present - a totally nonviable situation.

This approach is sometimes used for highly parallel computers. Recall that the cycle time for DRAM is about twice the access time, because reading a DRAM cell is destructive and it needs to be rewritten before the DRAM can be accessed again. Alternating between banks when accessing memory locations sequentially allows time for this.

This is called INTERLEAVED MEMORY.

VI. Rotating Disk Memory

A. Two basic types of rotating disk devices have been used for storage purposes: magnetic disks and optical disks. The former are readable and arbitrariily writable; the latter are readable and - in some cases - recordable. We'll focus on magnetic disk, though much of what we say applies to optical disks as well.

B. Historically, magnetic disk has been used for two purposes:

1. For storage of files.

2. As part of a hierarchical memory system to implement virtual memory.

C. For either purpose, the physical characteristics of the disk affect how it is used.

1. A magnetic disk system consists of 1 or more PLATTERS, each of has 1 or 2 SURFACES coded with a magnetic material. (Optical disks have only a single platter, but may use one or both surfaces (most often 1)).

2. (As a matter of essentially historical interest) Floppy disks were made of a flexible vinyl material, and typically used both surfaces (though single-sided disks were sometimes used.)

3. Hard disks are made of aluminum. Small disk systems generally have one platter, but larger systems often have multiple platters.

4. Each surface is divided into a number of concentric TRACKS. Each track, in turn, is divided into a series of SECTORS. Each sector holds some number of bytes of data, plus various formatting and control information.

5. Most disk systems use a fixed sector size - e.g. 512 bytes. This means that system software must map the data units used by the program (e.g. lines of text or database records) - which are generally of a different size - onto the fixed sector strucuture.

   Often, sectors on the disk are clustered into larger units that are read or written in a single operation. Space is assigned to files in units of a whole cluster.

   Example: On a computer which I used recently to update these notes, the cluster size is 4096 bytes. Thus, if I create a 3 byte file on this machine, it is still allocated 4096 bytes on disk; if I were to create a 4097 byte file, it would be given 8192 bytes on disk.

D. On a magnetic disk, data is written or read by positioning a HEAD above the correct track, and then waiting for the start of the desired sector to come up under the head.

1. On magnetic disk, the head consists of a metal armature with a small gap right above the disk surface, and two coils of wire: one used for writing and one usedfor reading.

   PROJECT: DISK HEAD

2. Data is written by passing a current through a coil in the head, which creates a magnetic field that magnetizes a spot on the disk.

3. Data is read by taking advantage of the fact that a moving magnetic field will generate an electric current in a nearby coil of wire.

4. During normal disk operation, the head is kept close to - but not in contact with - the surface. Because of the fast rotation of the disk and Bernoull's principle, the head actually "flies" above the surface and typically must be pushed toward the surface by a spring.

   Should the head ever come into contact with the surface, the result would be physical damage of both the disk and the surface, effectively destroying both. Such a problem is called a head crash.

E. Optical disks use a laser to read (and if appropriate) write the disk. Some optical disks actually read/write two different layers on the disk surface.

F. Most current disk drives use a single head/laser assembly for each surface, mounted on an arm that allows it to be positioned above any track. An actuator mechanism is also present to move the arm to any desired track under program control. (On multiple surface disks, all the head arms are usually ganged together to a common actuator.)

   A stepper motor is generally used for moving the arm, which results in the slight clicking you hear when the disk is being accessed.

G. Accessing data on disk involves a series of steps, each of which can take significant time when compared to internal processing.

1. Positioning the head/laser assembly (SEEK) - typically on the order of several ms, but varies widely depending on how far the head has to move from its previous position.

2. Waiting for the data to come up under the head (ROTATIONAL LATENCY or SEARCH.) The average latency is 1/2 revolution. (On a hard disk spinning at 6000 rpm, this would be about 5 ms.)

3. The actual transfer of the data. This is a small fraction of the rotation time. (On a hard disk with 100 sectors per track, spinning at 6000 rpm, transfer time for one sector would be about 0.1 ms.)

4. Note, then, that the bulk of the time spent accessing data on disk is involved in positioning the disk heads over the right track (seek), and waiting for the appropriate sector to come up under the head (search). If a typical access time on a magnetic disk is - say - 10 ms, then often less than 1% of this time is actually spent transferring the data. (Access times on optical disks are much higher, because the laser assemblies that must be moved are more massive.)

   This is one reason for relatively large sector and cluster sizes - the time cost of an operation is amortized over all the bytes in the cluster.

H. Magnetic disk capacities have improved dramatically over the years. The key factor is this: to increase capacity, the spacing between tracks and between bits within a track must be reduced. This means using heads with smaller gaps. But smaller gap heads must be closer to the disk surface to read and write data reliability, which calls for improved manufacturing precision. This is one reason why magnetic disk systems use sealed packs: head positioning tolerances can be smaller and contaminants that could get wedged under the head can be sealed out.

   Optical disks have not exhibited the same dramatic increases in capacity - perhaps in part due to the need for preserving media compability and also due to the inability to use sealed packages.

I. In recent years, there has been a move to replace magnetic disks with solid-state drives built around flash memory in lighter weight laptops and smaller devices such as cell phones. (When used in this way, these are known as Solid State Disks - SSD for short.)

1. SSDs are considerably more expensive per unit of data (about 10 x as much), so systems using SSD tend to use SSDs that are smaller than the hard disks that would otherwise be used.

2. However, SSDs are much faster than hard disks - on the order of 20-100 x as fast.

3. Repeated erase/write cycles can ultimately cause a block on an SSD to fail.

J. Though disks can be used as part of the memory system, they actually interface to the rest of the system as IO DEVICES. A system will include one or more disk controllers, each of which controls one or more disks. The controller responds to IO commands such as "seek the heads to track x"; "read x sectors of data from sector y on the current track" etc.

   We will discuss IO systems later in the course.