Need: Projectables: Dynamic RAM cell, ROM Basic Cell, EPROM Basic Cell, memory configuration examples; Disk head

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 fact, the ability to store and transmit various kinds of information has become of increasing importance. Many computers are used primarily for information storage and retrieval, rather than for numeric computation. (Recall the homework problem asking about what name might have been given to "computers" if they were developed today.) This has been made possible, in part, by dramatic increases in the capacity of computer storage media.

C. 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.

D. We will look at the various semiconductor technologies currently used for the memory system. We will also look briefly at disk technology, which is often used as part of the memory hierarchy we will discuss in the next series of lectures.

I. Overview of Semiconductor Memory Devices

A. Frequently, a computer system will have from one to several hundred semiconductor memory chips. There are also monolithic chips available which have a complete memory system 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.

1. A type of memory is characterized by:

a. 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.

Often, read-only memory is called ROM, which stands for read-only memory. Read-write memory is often called RAM (which stands for random access memory, distinguishing it from technologies like disk) - but this is actually a misnomer since ROM is also typically random-access.)

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

c. Power requirements.

i. In the case of volatile memory, does the memory system require significant power at all times, or only when it is being accessed?

ii. This is, of course, a particularly important characteristic for memories used in battery-powered devices.

c. Density - how much memory can be put on a chip.

   i. 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 the dominant limiting factor on density.

   ii. Due to manufacturing an installation costs, this also affects the cost per bit of a type of memory.

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

e. Write time - for some kinds of read-write memory, 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.)

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

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

It turns out that fast access/cycle time and low density - hence high cost per bit - go hand in hand, which means that most practical systems will use several different technologies with relatively small quantities of the faster technologies and larger quantities of the slower ones.

2. In the case of a memory chip, two other issues are important

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

   i. One bit.

   ii. 8 bits (one byte)

   iii. Some other size, such as 2 bits, 4 bits, etc.

   iv. Notice that the word size of a memory chip need not be (and often is not) the same as the word size of the system on which it is used. (But it is almost always not greater than the word size of the system.) 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.

   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.

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

   128 meg x 4 bit = 512 megabit

i. In advertising, chip capacities are typically expressed in bits (makes them sound bigger!) [Note: B = byte; b = bit]

ii. 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.

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

a. A number of data lines, dictated by its word size - e.g. a 4 bit wide chip will have 4 data lines. (Some 1-bit chips use separate lines for data in and data out).

b. A number of address lines. To uniquely address each word on the chip, we need a number of address bits equal to log2 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 the address bits 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 typically the case with DRAM because of their capacity and the way the chip is internally organized.) This is called ADDRESS MULTIPLEXING.

c. Various control lines - eg:

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

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

d. 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

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

1 data line
16 address lines (multiplexed to form a 32 bit address)
1 chip select line
1 output (read) enable line
1 write enable line

e. 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 32 or more bits per word are typically composed of multiple chips, with several being accessed at the same time.

4. 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) 4 G bit chips. (18 would allow an extra parity bit per byte.)

B. There are six basic types of semiconductor memory, distinguished by how the data is actually stored. (With variations on the basic theme).

   1. Static RAM stores data in individual flip-flops on the chip.

      a. Data can be both written and read.

      b. SRAM is the fastest semiconductor memory technology (on the order of a few ns access time for the fastest designs - less when on the same chip as the CPU.)

      c. 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.

      (In August, 2001, IBM's web site listed SRAM's with capacities up to 8 megabits per chip, with cycle times as low as 3ns. A check in October, 2005 yielded very similar results, but showed one chip with 16 megabit capacity. A check of NEC's site in 10/07 showed the largest to be 36 megabits (4 Mb x 9). A check of Newark electronics on 8/7/15 showed about the same results, but some much higher capacity chips available for very low voltage use, but with no access time shown. Note that the highest capacity SRAM chip listed in a catalog consulted 11/17/95 had 1 megabit capacity. SRAM chip capacities are not increasing as rapidly as DRAM because power dissipation is the key limiting factor in density.)

   2. Dynamic RAM stores data as a charge on an internal capacitor - one capacitor per bit.

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

      b. Data can be both written and read.

      c. The basic DRAM bit cell is quite simple:
      PROJECT: MOS Dynamic RAM cell - Bartee Page 269

         i. 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.

         ii. 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.

iii. Data can be read from the cell by connecting the the data line to a sense amplifier. When the row select 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. The individual memory cells on a chip are generally organized as a square array, with the location of an individual cell specified by a row number and a column number.

i. 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.

ii. The process of reading data from the chip then involves the following steps:

- Read the appropriate row from the memory array into the on-chip buffer.
- Access the desired bit(s) from the on-chip buffer.
- Write the row back to the memory array (because the read was destructive.)

iii. The process of writing data to the chip involves the following steps:

- Read the appropriate row from the memory array into the on-chip buffer.
- Change the values of the desired bit(s) in the on-chip buffer.
- Write the row back to the memory array (because the read was destructive.)

iv. Because both read and write operations actually involve both reading and re-writing a whole row (possibly with some changes), 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. [ISA's that allow a memory location to be both a source and a destination of an operation - e.g. IA32 - reflect this capability].

v. 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 (log2 1 G) / 2

- first, the address of the desired row (15 bits) is placed on the pins, and a row is read.
- then, the number of the desired column (15 bits) is used to select a column within the row.
- the two halves of the address are called the "row address" and the "column address".

e. The time needed to access a row of data for 5 Volt chips has remained relatively constant at 50-60 ns.

i. 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. 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 successive locations in memory are accessed succesively

      - Programs typically occupy successive locations in memory.
      - Data structures such as objects and arrays occupy successive locations in memory.

   (c) When successive locations in memory are being accessed one after another, average access time 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:

      - Fast Page Mode (FPM) DRAM
      - Extended Data Output (EDO) DRAM
      - Synchronous DRAM (SDRAM)
      - Double Data Rate DRAM (DDR RAM) and variants (DDR2, DDR3)

      It is common to find memory modules given speed ratings in MHz (e.g. 133 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, or sometimes multiple blocks being transferred on the same bus clock (various DDR technologies)

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

   (d) These strategies can reduce the time for the second and subsequent accesses to data in the same row in the DRAM to 10 ns or less. (The first access to the row still costs on the order of 60-80 ns).

   (e) The chip typically has two control signals used to specify whether the address being presented on the address pins is a row address or a column address. These signals are called row address strobe (RAS) and column address strobe (CAS).

ii. Chips designed for lower voltage operation (e.g. in battery-powered devices) can achieve much better access times - often by an order of magnitude.

f. 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 the 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 to do when I revise these notes every two years is multiply the DRAM capacities in my examples by 2 to 4]

g. 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. 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.

   i. If the programmer could guarantee that each row of the memory would be accessed regularly during normal program execution, this would suffice.

   ii. But since such a guarantee is virtually impossible, systems 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.

3. Mask-programmable 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.

   PROJECT ROM Basic Cell

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

      i. 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.

      ii. 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.

   b. 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: Most microcomputers contain firmware that handles booting the operating system from disk and provides certain basic services to the operating system. This firmware is typically implemented as a mask-programmable ROM.

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

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

   b. Each connection includes a fusible link 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.

      i. During programming, the power supply voltage is raised from 5V to 10V. 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.

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

      iii. As each location is addressed, a large current (say 65ma) 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.

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

   d. 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 out of the system; 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.)

5. Erasable PROM (EPROM), like Dynamic RAM, stores data in the form of electric charge in an on-chip capacitor. 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

      i. 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.

ii. 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: raising the power-supply voltage to 21-25V 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.

- Before programming, all gates are uncharged, so all locations on the chip contain 1.
- During programming, those bits to contain a 0 receive a high voltage pulse that charges the gate.
- 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.

iii. 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

iv. 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.)

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

6. Various forms of electrically erasable programmable ROM (EEPROM, EAROM, Flash memory). This is similar to EPROM, except that the charge may be both stored and erased during normal system operation.

a. 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.

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

c. 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.

d. 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.

II. Organizing Semiconductor Chips into a Complete Memory

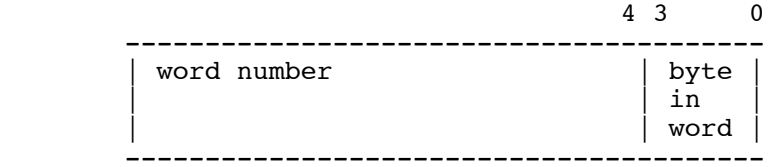
A. When we looked at the structure of a VonNeumann machine, we saw that two of its major building blocks were the memory system and the ALU and control units which together constitute what we now call the CPU.

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 ADDRESS.

   a. The size of the slot is fixed by the design of the memory system. For example, if the memory system is designed to support accessing 64 bit words, then the slot size is 64 bits. (It is, of course, possible to access a smaller unit of memory, but the memory system will access a 64 bit word and then the portion that is needed is extracted from it.)

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

   c. Note that there may be a difference between the size of the physical slots in memory and the addressable unit in the ISA of the CPU. For example, a byte addressable CPU might be used with a memory system that is organized around 64 bit words. In that case, an address generated by a program is split into two parts, as follows, with the word number portion being used to actually select the desired word in memory:



   Example: if the program references location 0x3 in memory, this will be interpreted as byte 3 in word 0.

   d. Systems often require that addresses be ALIGNED - e.g. if the size of the item being accessed is a full word, then the byte number must be 0, etc. Some ISA's enforce this; other ISA's allow unaligned addresses, but with the hidden performance penalty that two memory accesses will actually occur if the requested data spans two words in memory. (I.e. even if unaligned addresses are legal, they are not good).

   Example: MIPS requires aligned addresses; IA32 permits unaligned addresses, but using them results in lower performance.

   e. 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. We will develop our examples along this line for now.

2. The CPU and the memory typically exchange information using a set of datapaths called the MEMORY BUS. This consists of a large number of wires (or PC board traces) broken into two parts.

   a. The address bus.

   b. The data bus.

   c. In addition, the memory bus contains various control signals.

3. To read data from memory, the CPU puts the address of the data 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.)

   For dynamic RAM systems, systems often take advantage of the fact that multiple words can be read from the same row relatively quickly - in which case additional read cycles are done without specifying a new row address (just a column address).

4. To write data to memory, the CPU puts the address of the location 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. (Note: most memory systems allow the CPU to write a smaller amount of data than the word size of the memory, in which case the memory controller has to write back the original values in the positions that were not changed.)

B. Though it is possible to have a memory system that consists of a single chip, most semiconductor memory systems require multiple chips. We now consider how they are connected together. The following parameters will shape the design:

   1. The total size of the desired memory (usually measured in bytes, even if the memory is organized around a unit such as a 32 or 64 bit word.)

      Note: Often a system is built with room for expansion, so that the total amount of memory the system can handle is larger than the amount actually present on a typical system.

   2. The size of the "slots" in the memory.

      Note: 1 & 2, together, determine the number of address bits needed to uniquely specify a memory location, assuming that separating off a desired byte or group of bytes from a memory word is handled other than in the memory.

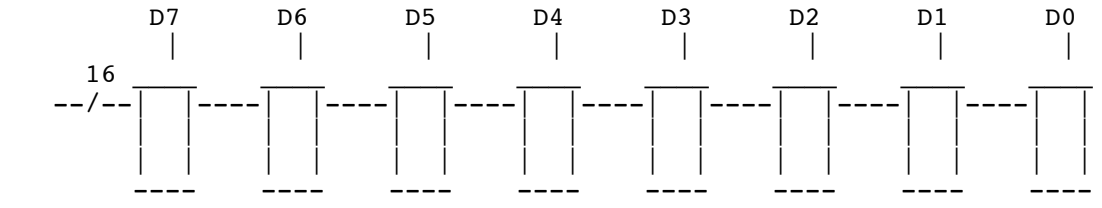
3. The width of the data bus connecting to the memory. This represents amount of data that can be transferred to/from memory in a single operation.

   Note: The data bus width is never smaller than the size of the addressable unit, but is often greater. In the latter case, the memory will sometimes transfer more information than requestor needs, with the requestor extracting the relevant pieces.

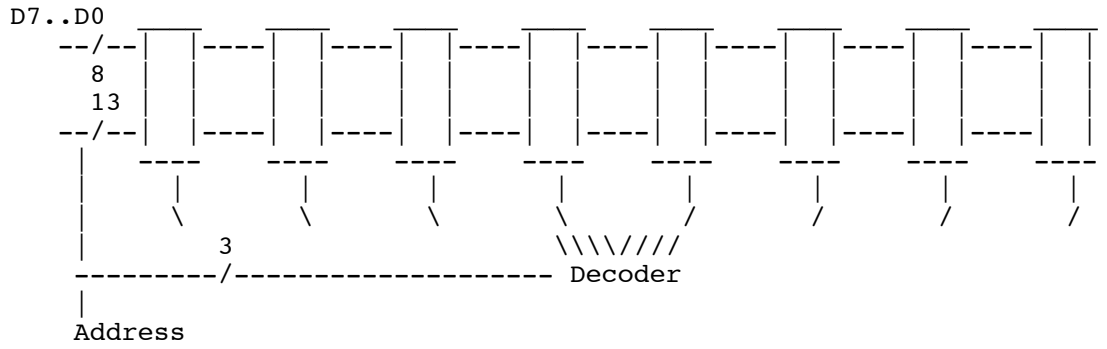
4. The capacity and organization of the chips to be used.

C. We begin with a very simple example: a 64 KByte memory, byte addressable, with a 16 bit address bus and an 8 bit data bus, constructed from 64K bit chips. We consider two possible variants, both of which will require 8 chips:

1. Using 8 64K x 1 chips. Each byte accessed is composed of one bit from each of the 8 chips



2. 8 8K x 8 chips. Each byte accessed comes from one of the 8 chips



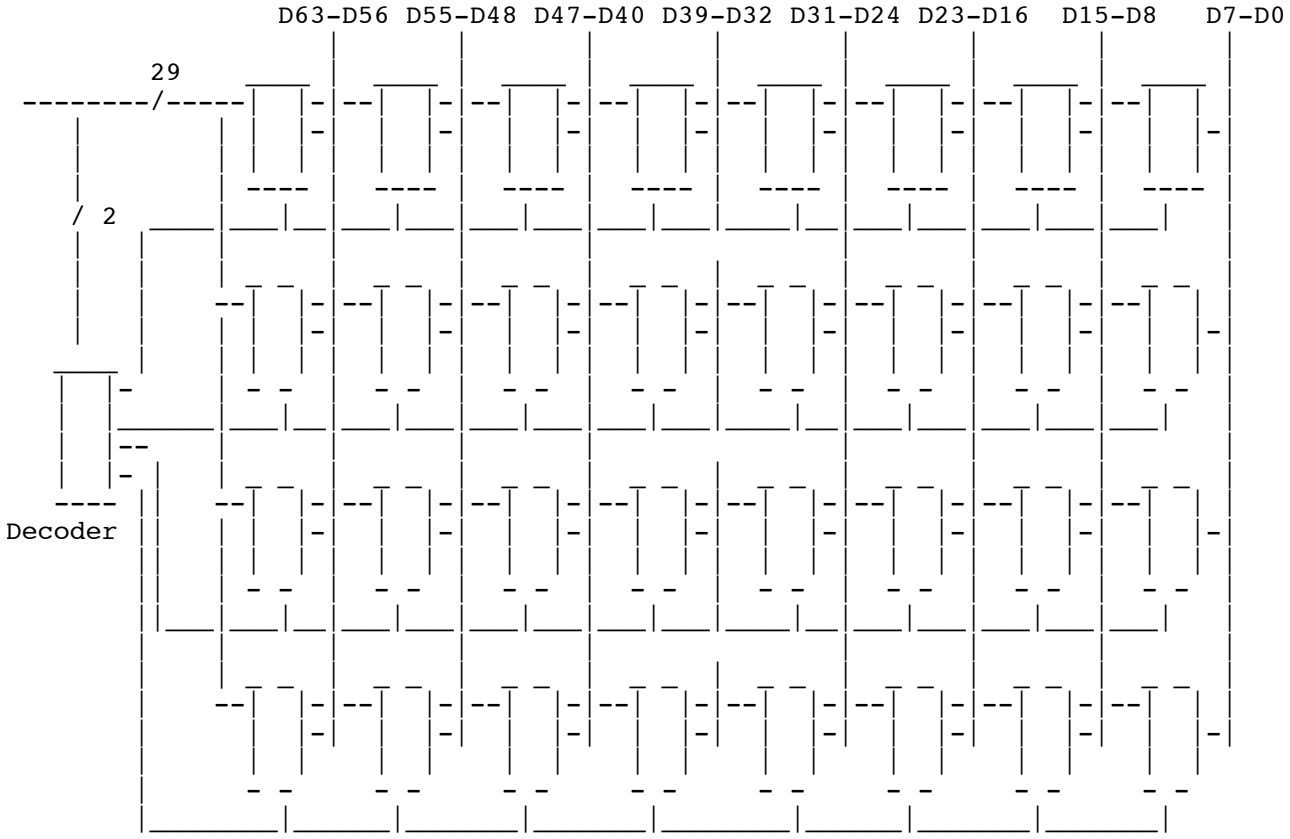
   The correct chip is selected by the decoder on the basis of the high 3 bits of the address. The outputs of the decoder go to a "chip select" input on each chip.

3. Note that the latter configuration would allow for the possibility that the memory system could be built partially populated with chips, with more chips added later. This would mean that accessing certain addresses would result in no response due to the missing chip. (If this were attempted with the first organization, each byte would be missing some number of bits - which is certainly not good!)

D. A more complicated case: a 4G Byte memory (expandable to 16 G Bytes), byte addressable, but with the memory organized around 64 bit (8 byte) words. Thus, the minimum size of 4G of memory corresponds to (4GB / 8 bytes/word) = 512M words, wand the maximum of 16 GB of memory corresponds to (16GB)/ (8 bytes/word)= 2G words. The memory system will receive a 31 bit physical address to select one of the 2G words, with some possible addresses not corresponding to any portion if only 4GB is implemented.

Assume the memory is constructed using 4G bit chips. We will need 8 chips for the initial configuration (before expansion) - since 4 GB = 2^32 bytes; each chip is 4G Bit = 512 MByte = 2^29 bytes; therefore 2^3 = 8 chips are needed. (Full capacity would use 32 chips)

Assume the chips are organized 512M x 8 (the 4G bits on each chip are organized as 512M chip words of 8 bits each). This means that each operation will access 8 chips (64 / 8 = 8).



a. Each chip connects to four bit of the data bus.

b. The 31 bit physical address is divided into two parts. 29 bits are used to select which of the 512 chip words on each chip is accessed (2^29 = 512M). The remaining two bits select a "row" of chips. In the initial configuration, just one row is populated; but there is provision for up to three more rows to be added later. (The chips for these are shown using dashed lines).

c. All chips in a given row have chip select activated by a common line that is asserted when the upper 2 bits of the address lie in the range allocated to this 4G Byte block of the overall 16 G Byte address space.

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.

E. The above examples were based on a design decision that is frequently made to let the UPPER BITS of the address determined which memory module responds to a request, with the LOWER BITS determining a specific location within that module.

1. A physical address is interpreted by the memory system as:

Module # Address of word in module

2. One of the major advantages of this approach is that it facilitates expanding the memory by adding a new module.

F. However, for highly parallel computers, a different approach is often used. On such machines, we want to be able to allow several memory accesses to be done simultaneously. This is possible only if each simultaneous access falls in a different module.

1. The scheme above might not work well for this. For example, all the instructions of a given program would probably lie in the same module; as would any given data structure, so if multiple machines were running the same program, only one could access memory at a time.

2. An alternate approach - which serves to spread successive addresses over several modules - is to interpret addresses as follows:

Address of word in module Module #

3. This is called INTERLEAVED MEMORY.

III. Magnetic Disk

A. 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.

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

C. A magnetic disk system consists of 1 or more PLATTERS, each of has 1 or 2 SURFACES coded with a magnetic material.

   1. Floppy disks were made of a flexible vinyl material, and typically used both surfaces (though single-sided disks are sometimes used.)

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

C. 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.

   1. 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.

   2. 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 the computer which I am using to update these notes, the cluster size is 4096 bytes. Thus, if I create a 3 byte file, 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. 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. The head consists of a metal armature with a small gap right about the disk surface, and two coils of wire: one used for writing and one used for reading.

      PROJECT: DISK HEAD

      a. 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.

      b. 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.

   2. On floppy disks, the head actually made physical contact with the disk surface; on hard disks, a flying head arrangement is used where the head is kept a very small distance away from the surface.

3. Most current disk drives use a single head 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.)

E. 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 (SEEK) - typically on the order of several ms on hard disks, 100's on floppy disks.

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 is - say - 10 ms, then often less than 1% of this time is actually spent transferring the data.

   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.

F. 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 most current disk systems use sealed packs: head positioning tolerances can be smaller and contaminants that could get wedged under the head can be sealed out.

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

a. 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.

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

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

H. 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.