Memory Management in Modern Computers

- One of the biggest problems facing modern computer designers is that of providing large amounts of high speed memory.
- This is a problem that has evolved over the last 20-25 years.
- Earlier in the history of computing, most processors were relatively slow compared to the speed of available memories (Except for bulk storage mechanical memories, i.e., disks and drums).
- Especially in the early days of the personal computer, the CPU was relatively slow compared to early electronic memories.
- The speed of random-access memory was not an issue; the biggest problem was just getting enough memory, period (early PC's with large memories had 256-512 Kbytes)!

Relative Speeds of the CPU and DRAM

- **Over the last two decades, central processor chips have caught up with and passed DRAM speed dramatically.**
- **Example: current CPU speed is 3-4 GHz, depending on the processor type, and should increase somewhat, although manufacturers are now abandoning the "speed race" in favor of multiple processors.**
- **On the other hand, the normal bus speed for CPU memory is about 1.6 GHz currently, and there is still a great deal of 400 MHz memory sold in lower-performance computers and laptops.**
- **The CPU performance edge over memory is on the order of 2-3, and much more than that on systems with the lower bus speeds.**

The Memory Speed/Cost Dilemma

- There are further problem facing the modern computer designer:
  - Users need very high memory speeds to improve performance (for example, in graphical computing, games, video editing).
  - At the same time there is also great demand for maximum memory capacity by many users (PC's do not just manipulate text any more; complex graphics, video games, movie editing and animation all require enormous amounts of both DRAM and bulk storage (hard drives [HDD's]).
- However, there is a conflict in these requirements:
  - Fast memories are very expensive.
  - High-capacity, cheap memories (esp. HDD's) are very slow.

Stating the Memory Management Problem

- **The computer designer of today is therefore faced with a problem that is not easy to solve:**
  - **There must be enough high-speed memory available to avoid slowing down the processing rate of current CPU's.**
  - **There must be sufficient DRAM to avoid the deadly "disk access" (i.e., having to go to the HDD to get program or data material), at least very often, since HDD access is very slow.**
  - **There must be enough bulk memory (HDD) for all storage needs, and accessing this memory and transferring it to DRAM/other memory must be as painless as possible.**
  - **The cost must be reasonable.**

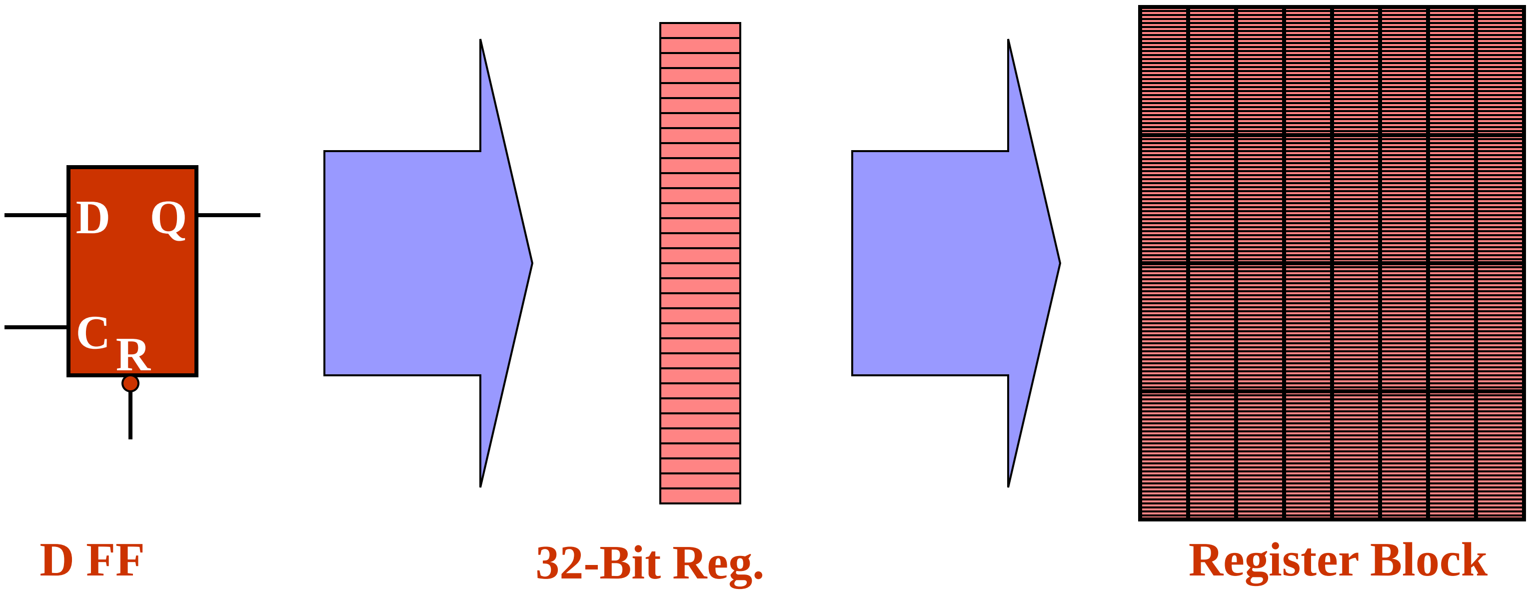
Solving the Problem of Memory Management

- **The current approach to memory management :**
  - **CPU has a large register complement, which allows more data in the CPU at a time and improves performance.**
  - **Very-high-speed D flip-flop arrays, called cache, hold currently executing program segments. There are two kinds:**
    - **L1 cache – On CPU chip, adjacent to ALU. ~ 16-64 Kbytes, very fast.**
    - **L2 cache – Opposite side of CPU chip. ~ 1-12 Mbyte, very fast.**
  - **High-speed electronic memory ("DRAM," up to 4 Gbytes, fast) provides capacity for programs currently in process.**
  - **Bulk storage memory (disk drives and flash memory, ~0.3-2+ Tbyte, slow but cheap) for maximum storage and archives.**
  - **Slower memories such as CD's and DVD's for long-term storage.**

Types of Memory

- **As we have just seen, even in the everyday PC, the use of sophisticated memory management is common.**
- **This means that there are five kinds of memory in the modern PC or workstation computer: Registers, L1 and L2 cache, DRAM, and the disk or HDD. And this does not count CD's, DVD's, Zip drives, thumb drives (flash EPROM), or floppy disks!**
- **The challenge to the computer engineer is to mesh the first five storage media and to make the use of them "transparent" – that is, invisible to the user, who will appear to have <u>massive amounts of high-speed, cheap memory available to solve any problem</u>.**
- **Before we discuss how to manage this extremely challenging engineering problem, we will discuss the types of memory that are used and learn a little about them.**

Registers



- **We already know that registers are simply collections of D FF's.**
- **Most CPU's today contain many registers, (e.g. the R-2000's 32).**
- **Registers are inside the CPU, adjacent to the ALU, so their speed is basically that of the CPU (in fact, they <u>determine</u> ALU speed).**

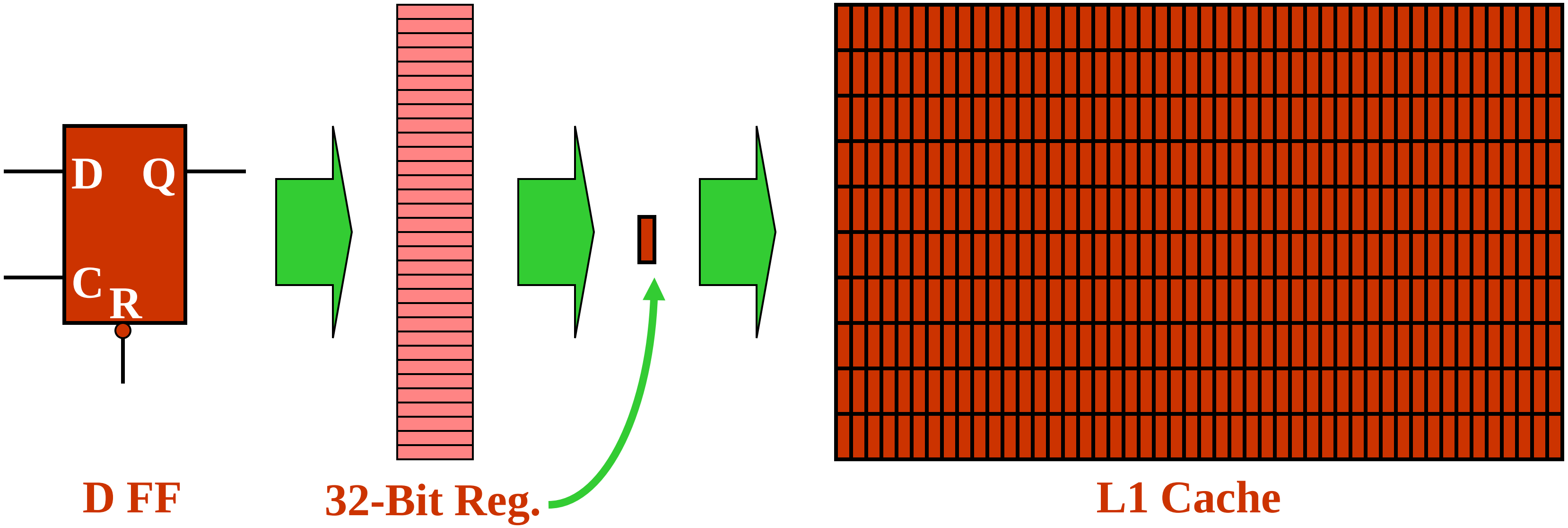
Random-Access Electronic Memory

- Random-access memories (RAM) make up the "working memory" of most computers.
- These memories are referred to as "random-access" because the entire array of memory is immediately available to be used; any single byte in the memory may be loaded or stored ("randomly accessed") in the same amount of time.
- There are two primary types of RAM: Static RAM (SRAM), and dynamic RAM (DRAM). Both SRAM and DRAM are used in modern computers such as the PC.
- SRAM is used in what are referred to as caches – small, very-high-speed memories that are physically close to the CPU.
- DRAM, though very fast, is slower than SRAM, but because it is inexpensive, it is the primary memory in most personal computing systems.

L1 Cache

- L1 cache ("level 1 cache") is SRAM memory that is very close to the CPU. For example, it is next to the ALU in most processors.
- L1 cache is basically sets of D FF's – but many more than in the CPU register block.
- For example, a typical register block might have 16-32 registers of 4 or 8 bytes each for a total of 64-128 bytes of storage. The Intel Quad-core cache, on the other hand, has 512 Kbytes – the equivalent of 4,000,000 registers.
- Access speed of L1 cache is slower, however, due to the complex arrangement of data buses which is necessary to access specific bytes in the L1 memory array. It is typically about one-third as fast as CPU registers in terms of load/store cycle.

L1 Cache (Continued)

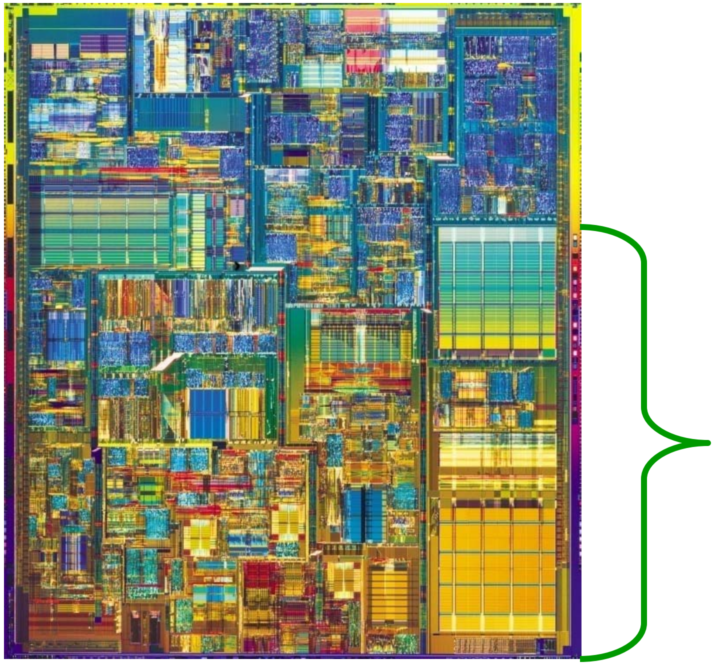


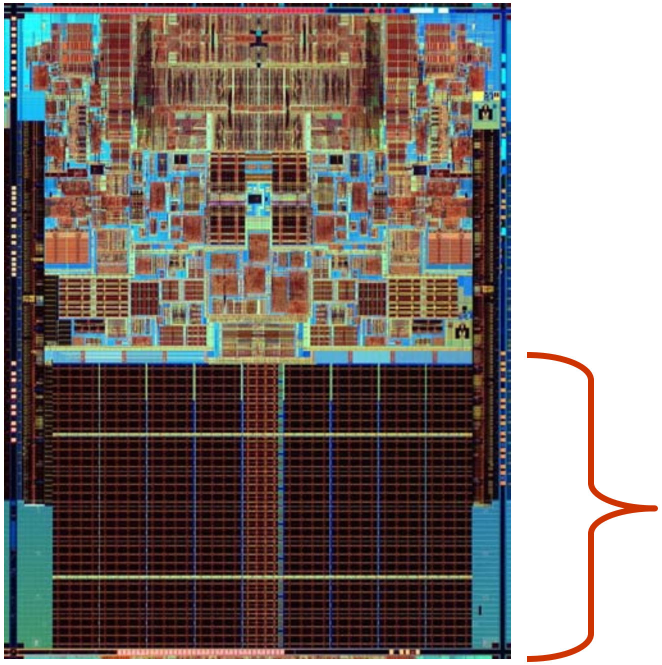
- L1 cache is a much bigger collection of D FF's that the register block. Typical L1 capacity in recent processors is 16-128K/core.
- In terms of memory arrangement, cache has <u>regressed</u>. Modern computer chips have separate instruction and data caches!

L2 Cache

- **The level-2 cache, although on the CPU chip, is on the opposite side of the chip. L2 cache is also SRAM.**
- **L2 cache is much greater than L1 cache, since more "real estate" is devoted to memory. Intel caches are 1-12 Mbyte. The AMD 4-core has 12 M; AMD six-core also has 12 Mbyte.**
- **Due to even more elaborate bus arrangements and the fact that L2 cache is not as close to the CPU, load/store access is > L1 cache, but still << DRAM.**

Cache Location on Intel CPU's





L2 cache physically located on P-IV chip.

L2 cache area on Duo-Core ("Conroe") circuit.

Why Not More Cache?

- **The question arises: If cache memory is so great, why isn't all computer memory fast cache?**
- **Answer: Cache memory has two major problems:**
  - **It consumes huge amounts of power compared to DRAM memory (a flip-flop has about sixteen transistors; a DRAM cell uses only one).**
  - **This means if more cache were used, the cost of a computer (think PC) would go up dramatically, due to the cost of extra power to run it, and cost of cooling the computer!**
  - **Also, cache is much more expensive than DRAM (5:1 or more).**
- **For that reason, DRAM memory is an excellent compromise solution to fast storage problems.**

Comparison of SRAM and DRAM

SRAM	Parameter	DRAM
Very fast	Speed	Fast
High; ~16 transistors per storage cell	Complexity	Low; 1 transistor per cell
High	Power Used	Very low
Excessive	Heat Generated	Virtually none
High	Cost	Very low

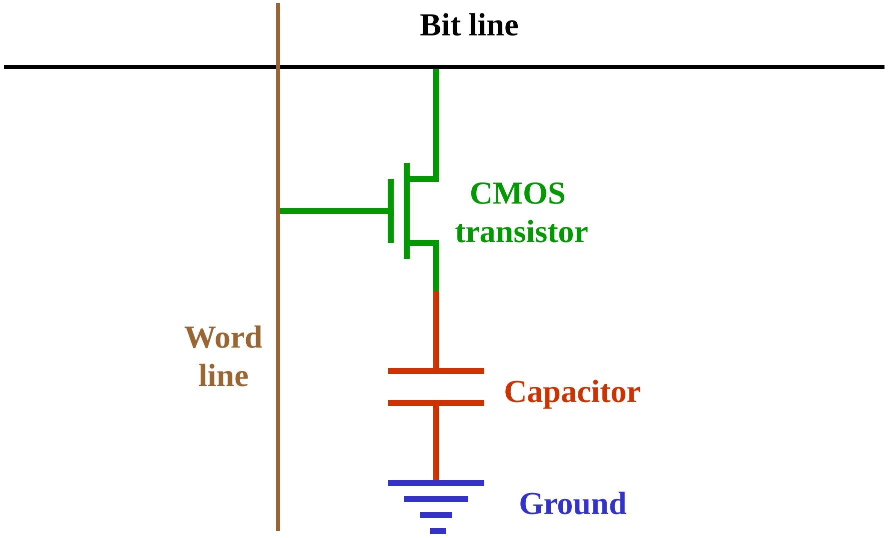
DRAM Memory

- **The term DRAM stands for "dynamic random-access memory" (pronounced "D-ram," not "dram"). This means that the title above is actually redundant!**
- **DRAM is electronic memory that is capable of very fast access (load or store), but is not as fast as cache. One exception is "Rambus" memory, a special DRAM memory whose manufacturer has announced cache-speed products (up to 7.2 GHz!). It is very expensive, however.**
- **The simple construction of DRAM makes it ideal in modern, workstation-based computing, where most users have their own computer system (PC, Mac, Sun, etc.).**
- **DRAM consists of a simple charge-storage device (stored charge = "1"), with a switch to store/test the charge. Only a single transistor is required for a DRAM bit cell.**

DRAM (Continued)

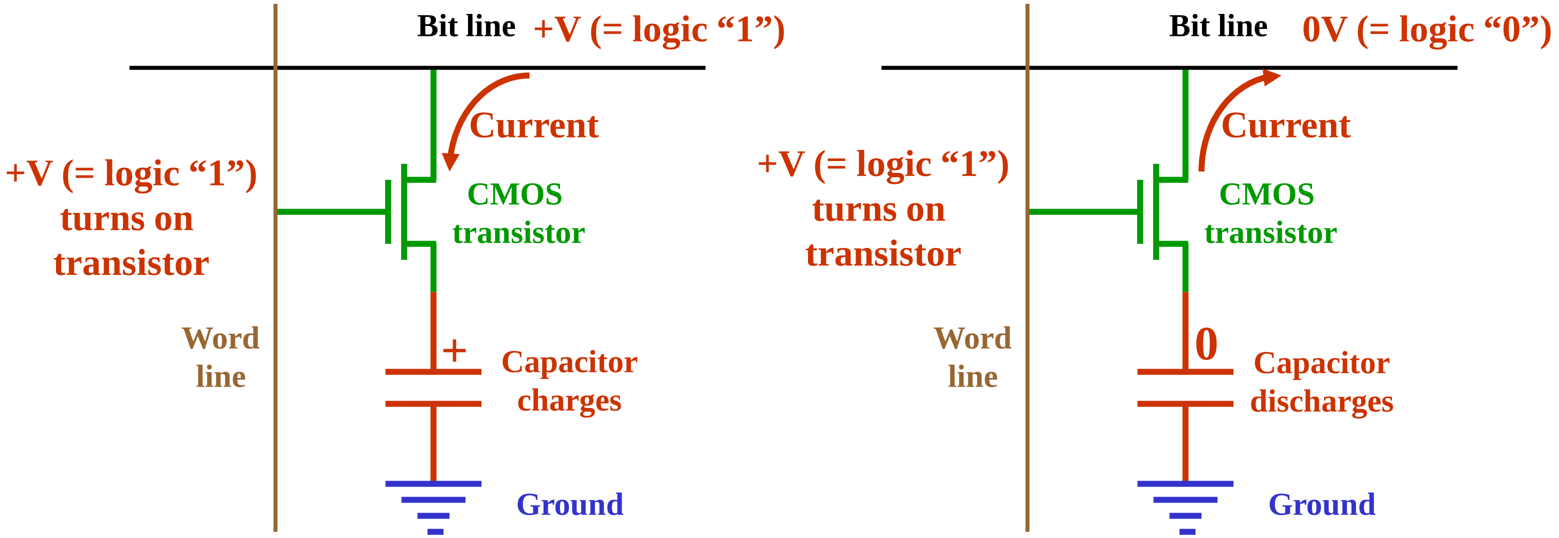
- **The term "dynamic" in DRAM is due to the fact that the memory is not truly a flip-flop; it is not static. DRAM "remembers" a 1 by storing charge on a capacitor.**
- **Capacitors, however, are not perfect storage elements – the charge leaks off after a short time. Thus the DRAM element is "dynamic" – its memory lifetime is limited and it must have its memory refreshed periodically.**
- **On the next several slides, we explore the way DRAM is constructed and the odd way that it must be treated to be sure that it retains its memory.**

DRAM Memory Cell Construction



- The DRAM cell is quite simple, consisting of a single CMOS transistor and a capacitor, which can store electronic charge.
- The capacitor is grounded on one end. Wires connect two terminals of the transistor to lines that can apply voltage.

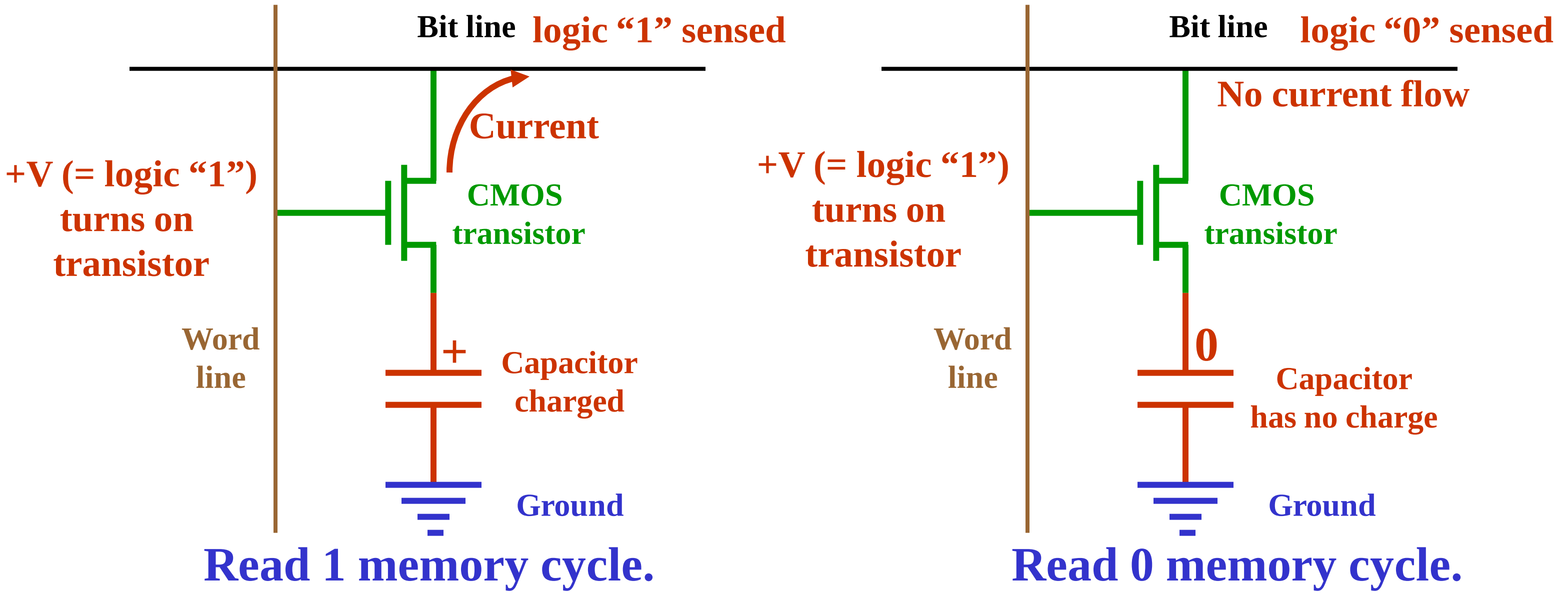
DRAM Cell Operation



To "write logic 1 data" to a DRAM cell, a voltage is applied to the word line, which turns the transistor on (it is like an "electronic switch"). If a voltage V is applied to the bit line, current flows into the capacitor and charges it, creating a "logic 1."

To "write logic 0 data" to a DRAM cell, a voltage is applied to the word line, which turns the transistor on (once again, like an "electronic switch"). Now, if 0 volts ("ground") is applied to the bit line, current flows out of the capacitor and discharges it, creating a "logic 0."

DRAM Cell Operation (2)

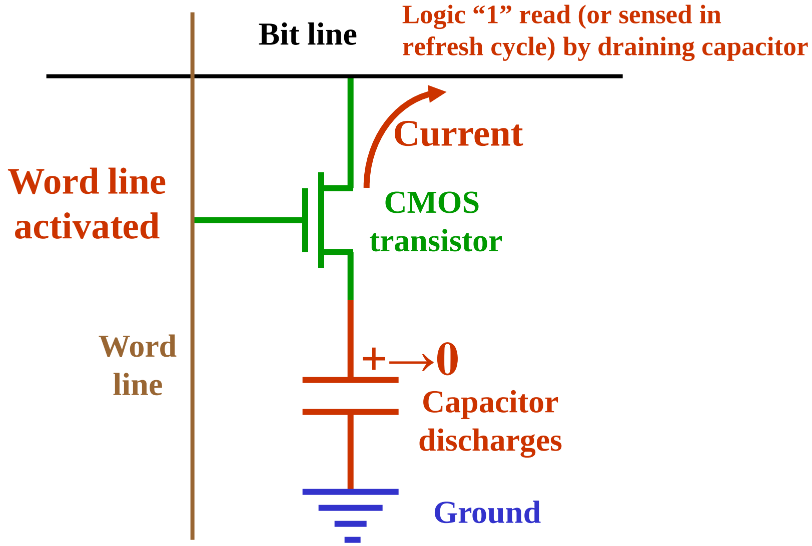


To "read," or sense the value of the DRAM cell, the word line once again has a voltage applied to it, which turns on the transistor. If the capacitor is charged, current flows OUT of the transistor, and this current is sensed and amplified, showing that a "1" is present. If the capacitor is discharged, no current flows, so that the sensing element determines that a logic 0 is present.

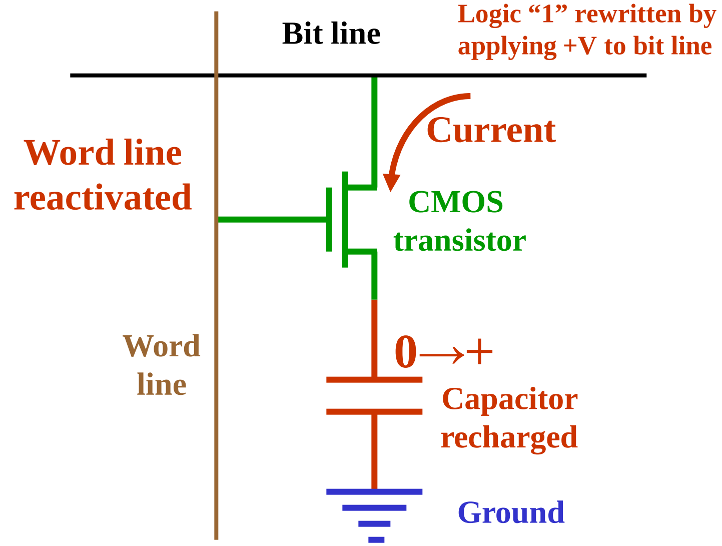
DRAM Cell Operation (3)

- **Note that in reading a DRAM memory cell with a "1" in it (charge stored on capacitor), the act of reading destroys the "1" by draining the charge off the capacitor.**
- **Therefore, after reading a "1," it must be rewritten.**
- **Also, as time passes, whether used or not, the capacitor loses charge so that the logic "1" eventually disappears.**
- **We see that even if a 1 is not read, the charge must be periodically replaced or the DRAM memory "loses its mind!"**
- **In a modern DRAM cell, this "refresh" must occur every few milliseconds.**
- **The refresh cycle is not long, however, taking 4-5% of total memory read/write time, which does not reduce memory speed or efficiency to any great degree.**

DRAM Cell Operation (4)



Read 1 memory cycle or refresh cycle logic "1" detect.



Read or refresh cycle logic "1" rewrite.

- **The refresh cycle occurs after a logic 1 read or periodically if the memory cell is not accessed. The refresh cycle is typically every few milliseconds. Obviously if the cell is a 0, it is not recharged.**

Exercise 1

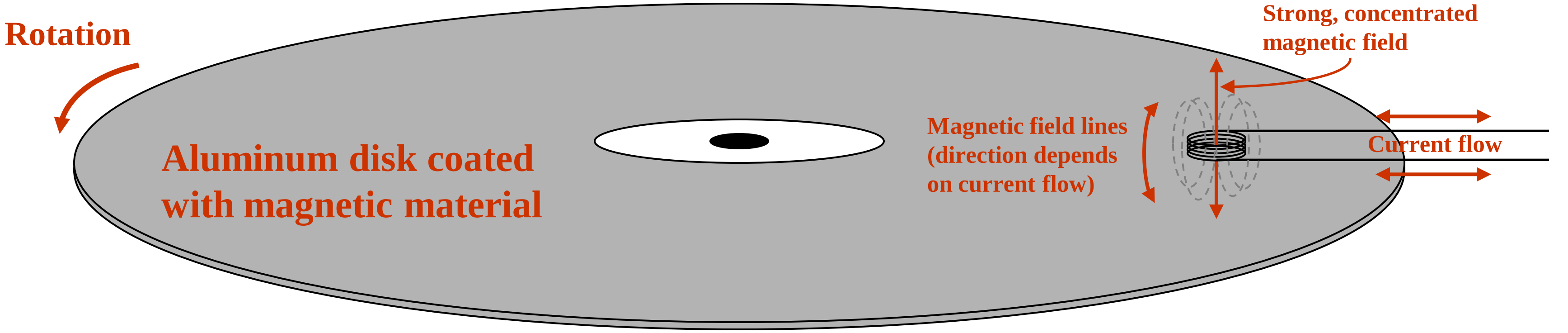
1. Rank these memories by speed: L2 cache, DRAM, L1 cache, registers, and hard disk drives.
2. A DRAM memory chip is accessed and a bit read out. The bit that is read is a 1. What happens now?
3. That same memory bit is then left "alone" (i.e., not accessed by its addressing mechanism for either read or write) for several milliseconds. What happens next?

1. (This will help with homework #8) – Registers in the computer are adjacent to the ALU, L1 is on-chip, L2 is nearby, and DRAM and HDD are farther away from the CPU. Thus the speed ranking is registers, L1, L2, DRAM, HDD.
2. The 1 data is erased by the read, so that the 1 is immediately rewritten after it is read.
3. The capacitor begins to lose charge (the "1") and so it is rewritten periodically.

Bulk Storage (Disk Storage or HDD)

- Electromechanical data storage is normally not random-access like SRAM or DRAM.
- This means that data cannot normally be accessed in arbitrary order, but must be loaded or stored according to rules, which generally have to do with positioning a recording mechanism over the correct location in an expanse of recording media prior to being able to perform the memory access.
- That is, the correct segment of data must be located (normally by mechanically moving a recording head) before it can be read.
- This load/store operation is particularly time-consuming, because it involves mechanical movement rather than simply electronic switching.

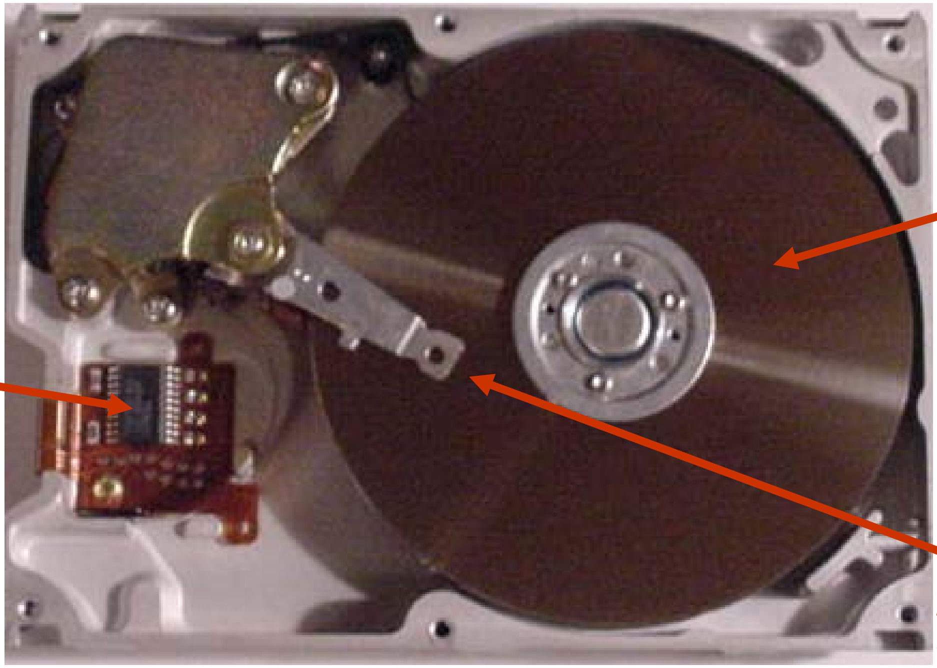
HDD Read/Write Mechanism



- The HDD stores data on a rotating disk coated with magnetic material.
- A magnetic coil is used to record each one and zero. Current in the coil generates a magnetic field, which magnetizes material in the HDD surface. One direction of current writes a 1, the other a 0.
- When the coil is later positioned over the disk to read, the opposite-polarity 1's and 0's cause back-and-forth current flow according to whether a 1 or 0 is present. In this way, the data is detected.

Hard Disk Drive Example

Portion of read/write electronic circuitry (the rest is on the back side of the unit on a separate circuit board).



Metal disk covered with magnetic coating

Recording head

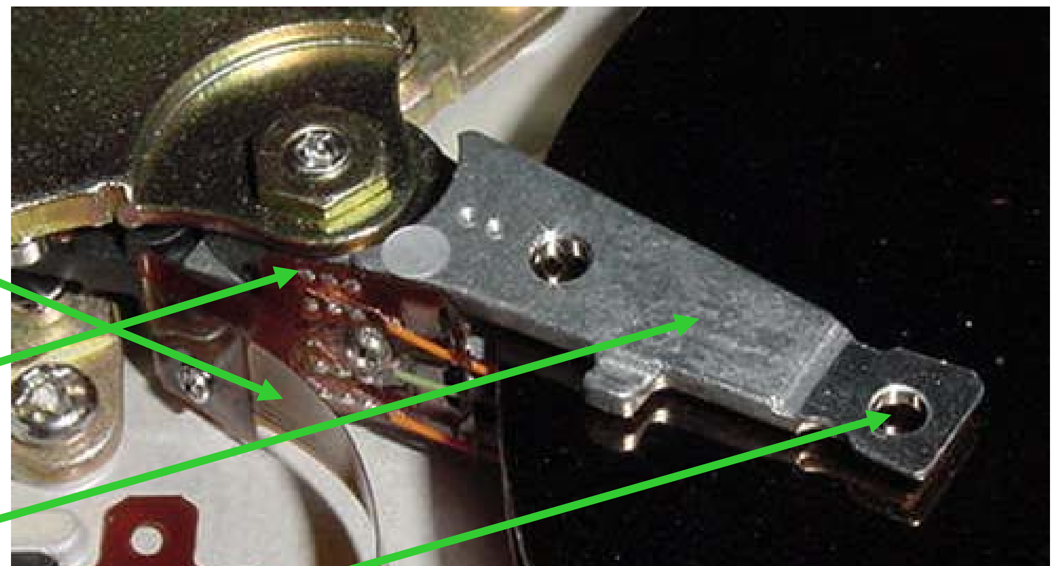
Detail of Disk Read/Write Head

Flexible cable carries signals to amplifier circuitry to be converted to digital signals

Positioning mechanism

Positioning arm

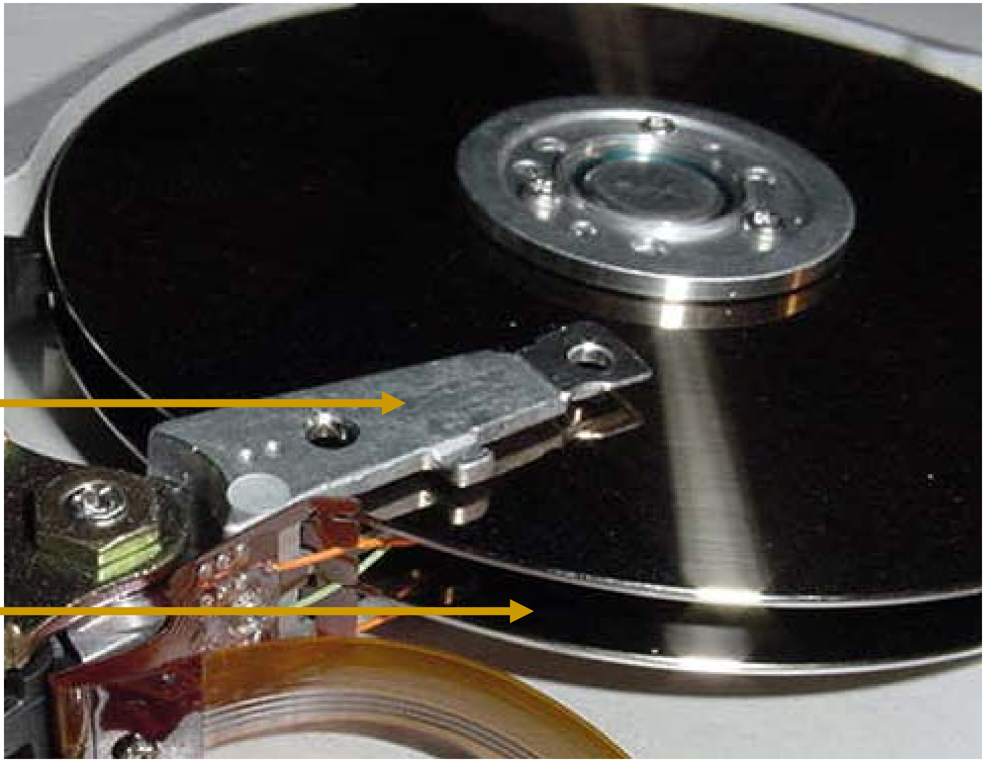
Recording head



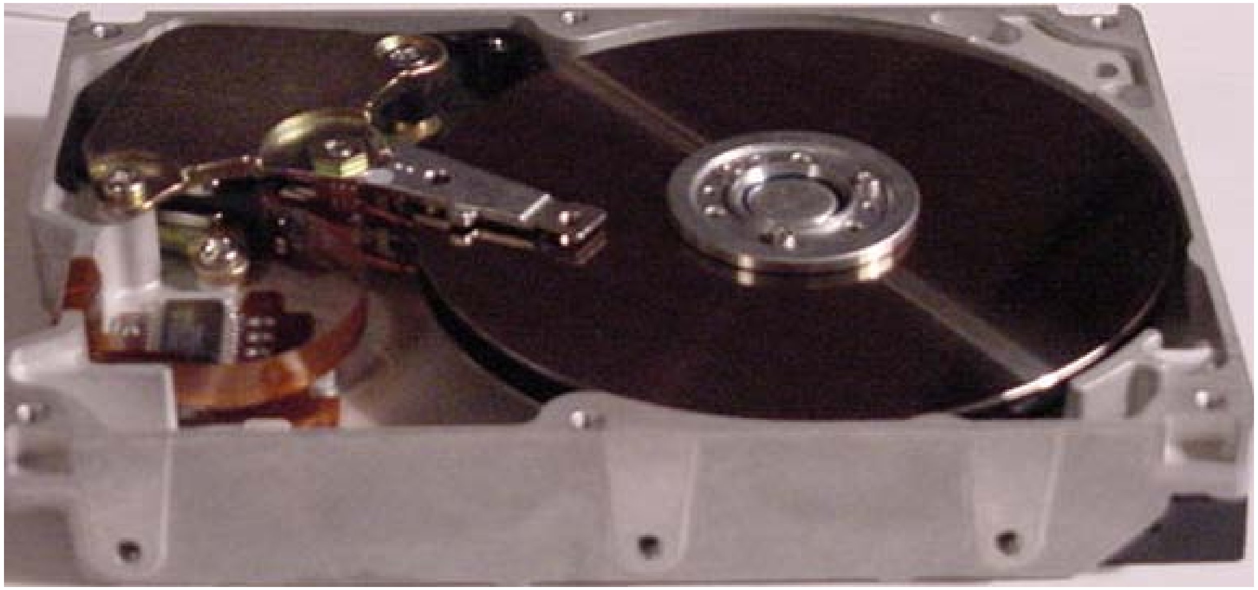
HDD Side View, Showing Multiple Disk Platters

Upper recording and reading head; note that positioning mechanism moves both heads simultaneously.

Second recording disk surface (recording head not visible)



HDD Package



The HDD is usually packaged in a metal case. Higher-quality units are typically packaged in an aluminum casting, or similar rigid container, which provides stability and better data integrity.

HDD Storage/Retrieval is Slow

- **The latency ("time to get/store data") of a HDD is given by the formula:**

latency = seek time + rotational delay + transfer time + controller delay

Where:

- **Seek time = time for the positioning arm to move the head from its present track to the track where the load/store data is located.**
- **Rotational time = time for the requested sector to rotate underneath the read/write head after the head is positioned over the track.**
- **Transfer time = time for data transfer from disk to main memory.**
- **Controller delay = time to set up transfer in the HDD electronic interface.**

HDD Storage/Retrieval is Slow (2)

- **Example: latency of writing one 512-byte sector on a magnetic disk rotating at 7200 rpm, with the following parameters:**
  - **Average seek time = 12 ms (typical for movement across half the disk)**
  - **Transfer rate = 5 Mbytes/sec; transfer time = [0.000512Mbyte/5 Mbyte/sec] = 0.1 ms**
  - **Controller delay = 2 ms**
  - **Rotational time depends on the position of the first byte to be transferred, but on average will be ([1/7200)]×60×[1/2]) = 4.2 ms (average rotation = ½ of circle).**

Then average latency = 12 ms + 4.2ms + 0.1 + 2 ms = 18.3 ms. Note that actual transfer time is small!

Other Disk Storage Units and Media

- Other storage media include CD's, DVD's and "thumb drives."
- Most of these storage units have (or are) removable media.
- Floppy disks, hard drives, Zip drives, and tapes are magnetic media.
- The CD-ROM and the DVD use optical recording/reading involving a laser beam to record and read data. They are relatively slow.
- The "thumb drive" is a newer archival media. It uses electronic memory called EPROM ("erasable, programmable read-only memory"), and is a true solid-state memory with no moving parts.
- Except for HDD's and EPROM's, these are primarily for archiving and not for immediate data access, due to their relatively slow read and record times (<u>and possible need to insert or remove media</u>).
- Note: Very fast EPROM's are beginning to be available for fast bulk storage, replacing HDD's on laptops. <u>They are expensive</u>.

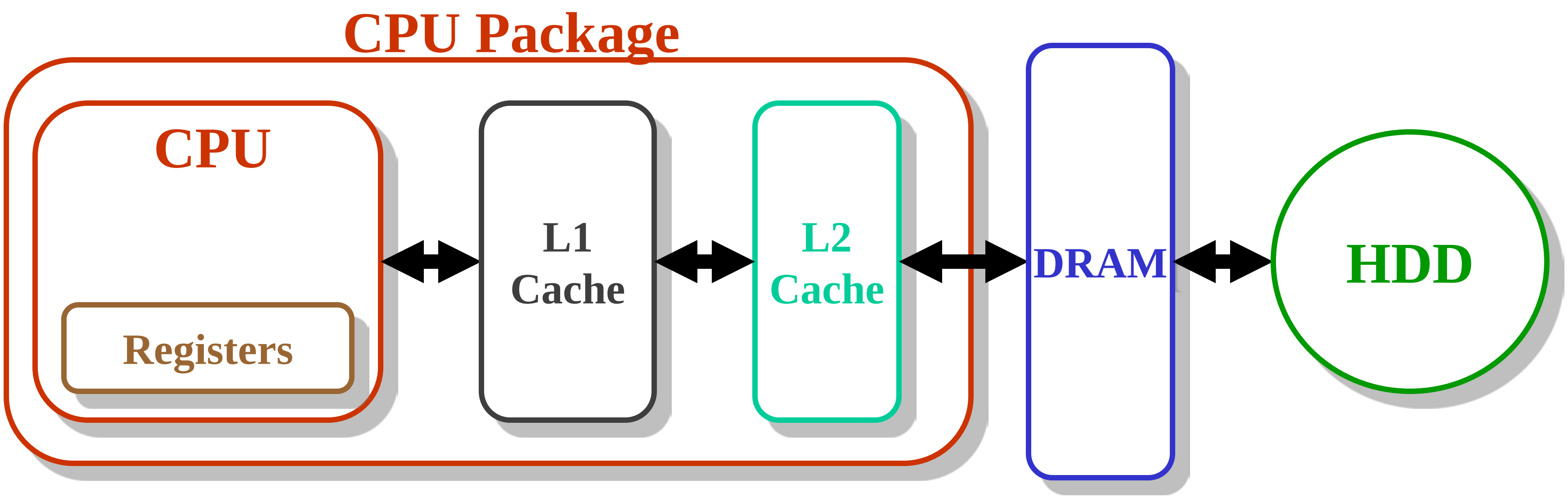
The Memory Hierarchy

- **We have described a number of memory devices which are useful for storing and reading computer data.**
- **All of these (other than archival types) are used in a mix on the modern computer for real-time storage and retrieval of data.**
- **Since SRAMs – the best data storage media if not so power-hungry and costly – cannot be used exclusively, a mix of L1 and L2 cache, DRAM, and HDDs make up the "memory hierarchy" of most computers.**
- **The trick is to design a mix of these types which will give the highest performance for a reasonable price.**

Arrangement of the Memory Hierarchy

- Memory arrangements make use of the fact that programs exhibit two common behaviors:
  - Temporal locality – Recently-used code and data is often reused (e.g., a loop program continues to use the same steps).
  - Spatial locality – Recently-accessed data items are usually close to other recently-accessed (or about-to-be-accessed) data items.
- Modern schemes use a "shuffling" methodology that moves data from slower storage media to faster media.
- Higher-speed memories are also placed closer to the CPU, since memory access also depends on the proximity of the storage element; electronic signals propagate at about 33 ps/cm.

Arrangement of Levels in Memory Hierarchy



Size:	<300 Bytes	8-64 Kbytes	0.5-3 Mbytes	0.5-4 Gbytes	160-2000 Gbytes
Speed:	100 ps	200 ps	0.2-0.5 ns	1-10 ns	~10-20 ms

- Memory is physically arranged so that fastest elements (registers) are closest to the CPU and slower elements are progressively farther away.

The Importance of Cache

- As mentioned previously, the key to modern computer performance is not the CPU – CPU performance has far-outstripped the speed of most computer memories.
- The key is the use of cache. The secret of today's high-performance PC's and workstations is the design of an architecture that allows maximum use of DRAM and HDD (cheap) plus just enough SRAM cache (expensive and power-consuming), thus enabling the CPU to realize most of its performance advantage.
- The method used is the "shuffling" technique alluded to two slides back. This method uses a very high speed, complex arrangement to constantly move program and data content from slower to faster memory as the CPU executes a process.

Cache Utilization

- **Cache designers make use of the principles of temporal and spatial locality to assure that the most-probably needed instructions and data are available to the computer in cache (to speed execution).**
- **Special hardware is designed to manage cache content with the goal of forecasting upcoming instructions and data required by the processor during program execution and moving it from slower DRAM into cache.**
- **This hardware has two special goals: (1) examining the currently-executing process and predicting instruction and data need, and (2) moving the required information from DRAM to cache in a timely manner to foresee that anticipated need.**

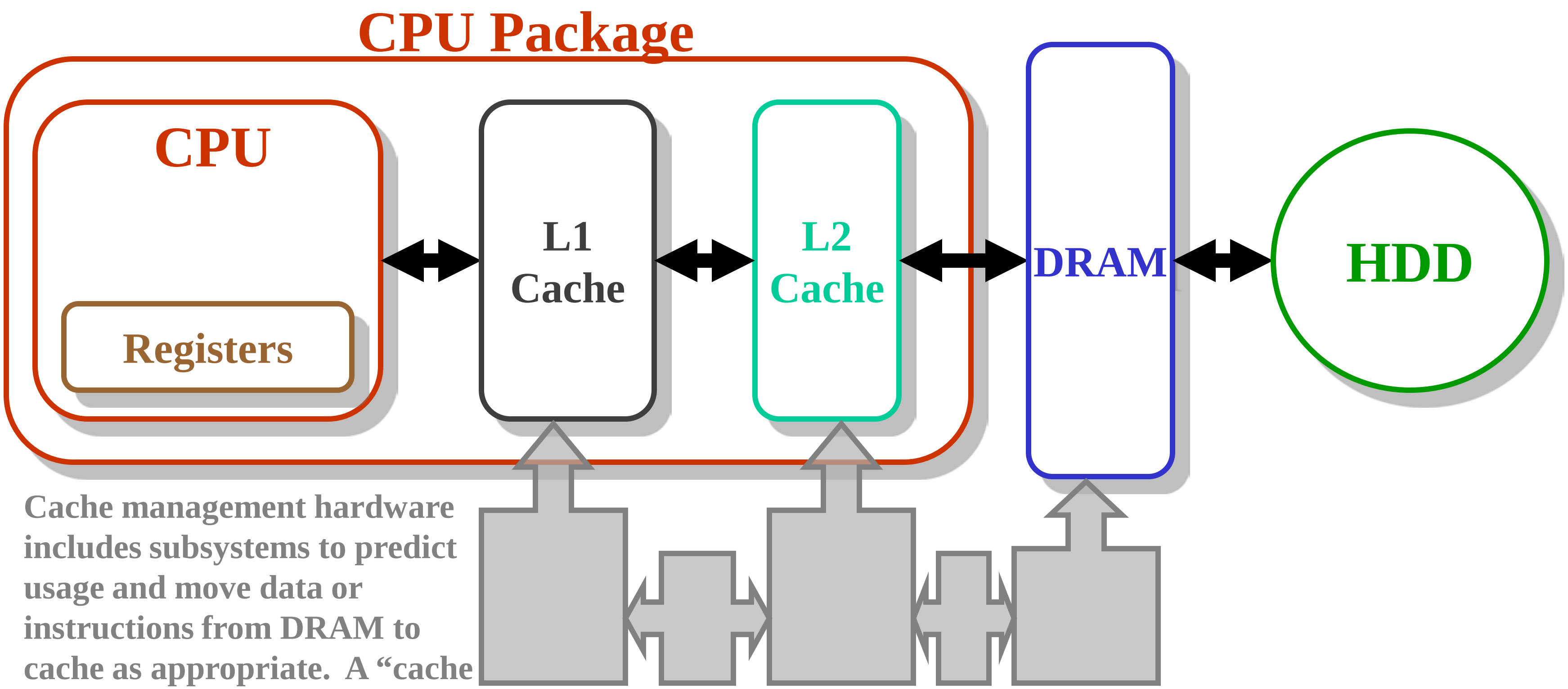
Looking for Data/Instructions in the Cache

- Clearly the purpose of cache management is to make sure that ALL upcoming instructions and data are in the cache.
- This brings up two questions: (1) how does the processor know that data is in the cache, and (2) if it is NOT there, how does the processor get it and what sort of performance penalty is there?
- There are several ways in which the cache can be assigned DRAM memory correspondence. The simplest is direct mapping, in which each block of memory in cache is assigned to some number of DRAM locations.
- When a program needs a particular DRAM location to be loaded, it goes to the corresponding cache location to get the data. This leads to further complications, in that now we need "validity indicators" for each cache location. This is because since each cache block is assigned to several memory blocks in DRAM, the program needs to know if the right data is available in cache at the time it is needed.

Looking for Data (2)

- **If the correct data is not in cache, the hardware memory manager declares a "cache miss." This means that the program must be delayed for several clock cycles while the required instruction or data is moved from DRAM to cache.**
- **We see that a cache miss is highly undesirable, since it can substantially slow down the program.**
- **A key part of cache memory management, then, is to minimize the cache misses, which correspondingly increases the speed of execution of a program.**
- **There are a number of clever and effective cache management designs, which dramatically reduce cache misses and improve computer performance. They are, however, beyond the scope of EE 2310.**

Cache Diagram



Cache management hardware includes subsystems to predict usage and move data or instructions from DRAM to cache as appropriate. A “cache miss” will initiate DRAM access for transfer to cache.

Virtual Memory

- The concept of virtual memory takes the advantages of cache one step further.
- Sometimes not all required data or program instructions are in DRAM – sometimes they are on the hard drive.
- This is because in modern computers, sometimes many programs are running at once ("multiprogramming"), and regardless of DRAM memory size, there is not enough random-access memory.
- The concepts of virtual memory allow many active processes to share limited memory, and to unburden the programmer from having to worry about memory limitations. Each process appears to have the full use of all working computer memory (i.e., all DRAM, and even all the HDD unit).

Summary

- Modern computer memory management is aimed at maximizing the speed of computer processing while keeping the cost of the system reasonable for the user.
- The approach is to use a small amount of very fast SRAM memory in "caches" which are physically near the computer, a substantial amount of DRAM, which is still very fast, as the main "working memory," and electromechanical data storage (HDD) for large program storage. Other electromechanical storage such as thumb and Zip drives, CD's and DVD's are used for archival storage.
- Effective (and complex) hardware and software suites have been developed to manage this memory hierarchy and maximize its effectiveness.

Exercise 2

1. Each small area of cache (say, 1K byte) represents a much larger area (say, 1 Mbyte) in DRAM. If an instruction, for example, is supposed to reside in a given Mbyte of DRAM, the corresponding cache extent is searched. Assume that, according to the validity indicator, the correct instruction is NOT in cache. What now?
2. Give simple definitions of the principles of temporal and spatial locality.

1. The CPU must wait until the correct instruction can be retrieved from DRAM.
2. "If the data or instruction was used recently, it might be used again soon." "If the data or instruction was from a particular area of memory, other data/instructions from that area will probably be used."

Computing in the Future

- We discussed the evolution of computing up to the present in Lecture #1.
- Now let's talk about the future.
- Note that much of the information presented here is from a recent article in PC World.

Memory

- The year 2008 marked the first development of the **memristor** – a completely new kind of circuit element (that was predicted in 1971).
- Memristor circuit elements can retain a state (i.e., memory) even when power is off. They could replace flash EPROM in the near-term (i.e., "thumb drives"), and eventually DRAM. Imagine a 1000 Gbyte main memory with no need for a disk drive!
- Memristors can remember multiple states (not just ones and zeros). Thus a memristor memory might eventually "remember" like a human neuron. This could lead to neural-type processors in the long term.

Memory (2)

- **Time frame for new memristor devices:**
  - **Flash memory replacements – circa 2012***
  - **Replacements for DRAM – 2015 or later***
  - **Disk unit replacements – 2015 or later**
  - **“Neural” (multi-state) memories – 2025-2030**
  - **“Realistic” estimated slip for these dates – five to ten years for near-term digital memory replacements. Ten to twenty (or more) years for true neural memories.**

*** Don’t hold your breath!**

Memory (3)

- Phase Change Memory ("PCM") is another new memory type.
- PCM is similar to flash memory, in that writing a one or zero bit is done in two ways:
  - Writing a 0 is done by heating up the PCM material and creating a crystalline structure, which has a low conductivity.
  - Writing a 1 is done with a higher temperature makes the structure an "amorphous: crystal – one that has a disorganized structure. This structure has much higher conductivity.
- Experimentation is still being done. Read/write cycles are only up to about 100 million (far too low to produce a product; DRAM and flash memory can do 1-10 quadrillion cycles!).

The CPU

- Manufacturers (chiefly Intel, but also AMD) have abandoned the GHz race in CPU's. Intel's goal of a 10 GHz CPU by 2010 is officially defunct.
- The "big deal" now is multiple CPU's. Four-core CPU's are now standard, and six-core CPU's increasingly popular (Intel Xeon upscale server CPU's are 8-core).
- Intel was said to have abandoned plans for a 32-core CPU. However, just two weeks ago, they announced a spectacular breakthrough (discussed shortly).
- The main force for further miniaturization (and therefore more cores per chip) is "minimum feature size."

Minimum Feature Size

- "Minimum feature size" is the smallest dimension that can be laid out on a chip in the manufacturing process. This is typically the width of a wire or the size of a part of a transistor.
- Currently the minimum feature size is about 32 nanometers in semiconductor manufacturing processes (one nanometer is one billionth of a meter in length [10^{-9} meters]).
- Intel has begun manufacturing at the 22-nanometer node. Both AMD and Intel have 22 nanometer products this year.
- At the 22 nanometer node, 16-core CPU's are possible. The default PC memory size is now 12-16 GB, with 64 available.
- Further feature size reduction will result in even more CPU's (and perhaps 128-bit multi-CPU's by 2016-2020 or so).

Multi-Core Advance

- Intel had previously announced that a planned 32-core chip had been abandoned. Now we know why:

"Wow: Intel unveils 1 teraflop chip with 50-plus cores!"*

- "A short time ago (1997), Intel was boasting about the first supercomputer with sustained 1 teraflop performance. That was a system with 9,298 Pentium II chips that filled 72 computing cabinets.
- Now Intel has developed equal performance in a matchbook-sized chip, ("Knights Corner"), based on its new "Many Integrated Core" architecture, or MIC, designed largely in the Portland.
- The company would not specify how many cores the chip has – just more than 50 – or its power requirements.
- This means that Intel could be producing teraflop chips for personal computers within a few years, although useful software would be a problem."

* Seattle Times, Monday, Nov. 21, 2011.

"Knight's Corner"

- To be used in a supercomputer at the Texas Advanced Computing Center (TACC) at the University of Texas at Austin by 2013.
- Called Stampede, the new computer will be built by TACC in partnership with Dell and Intel. When completed, Stampede will house several thousand Dell "Zeus" servers, each with dual 8-core Intel Xeon processors, as well as KC. This production system will offer almost 2 petaflops of peak performance.
- KC is built on Intel's latest 22-nanometer 3D transistor process – providing an additional 8 petaflops of performance (for a total of 10 petaflops in Stampede).
- Knights Corner uses modified Pentium-era cores. Intel went back to a simpler design that's more power efficient: a simpler core is a power saving opportunity.

Intel: "Ivy Bridge Coming Up!"

- What about the "bread and butter" Intel PC CPU's?
- Intel has odd names for CPU's.* Current CPU's are the "Sandy Bridge" family. Upcoming are "Ivy Bridge."
- May be out late this spring (by the time you are seeing this lecture!).
- "Ivy Bridge" CPU's will use less power, even at high clock speeds, giving laptops more power with better battery life.
- Also, "gamer laptops" will run even the most demanding games.
- Will have a new graphics architecture that is supposed to be much faster that current models.
- AMD's new chip, Trinity, (part of Fusion family) coming out, but details are sketchy, although it is reputed to have excellent graphics.

* Like Apple OS X updates – "Leopard," "Snow Leopard," "Lion."

The End of Graphics Processors?

- **Currently, to "unload" a computer CPU – especially for heavy-duty graphics generation as in PC video games, most users add a high-performance video card.**
- **But assume a CPU with 30-50 cores: Why bother with a graphics processor? Simply buy plenty of memory and assign as many cores as needed to graphics generation. Better still, put a several GPU's on the CPU chip. Interestingly enough, Intel and nVidia have just signed a cross-licensing agreement, giving Intel access to nVidia's GPU designs.**
- **The graphics processor may become superfluous for many applications, with large groups of CPU cores available. The exception would probably be high-performance gaming PC's and engineering design workstations.**

Miscellaneous

- **More "neat stuff" is coming to your nearest computer:**
  - **Wireless rechargers – and all wireless interconnection.**
  - **Mouseless cursor direction, and other efficient data input:**
    - **Eyeball tracking**
    - **Gesture recognition (remember the "The Minority Report?")**
    - **Really good speech recognition (finally!).**
- **A competitor to Windows? Not really.**
  - **Early on, Google Chrome OS was viewed as a competitor to Windows™.**
  - **However, Chrome OS is really intended for netbooks, which lack the computational power to run a resource-intensive program.**
  - **Others have concluded that Google's Chrome OS prototype met the basic requirements for Web surfing, gaming, and personal productivity, but fell short for more intensive tasks.**
  - **Further, most of Chrome OS's advantages can be found in other software environments. Its reliance on internet connectivity made it less useful, as well.**

64-Bit Software

- Intel introduced the first 32-bit CPU in 1986.
- However, the first full-32-bit Windows OS was not introduced until 1993.
- We will have a similar lag until the first full 64-OS is introduced to PC platforms.
- Note that MAC OS is already fully 64-bit.
- A 64-bit OS is severely needed for the PC platform – 32-bit PC's can only address 3-4 Gbyte.
- Until then, 64-bit PC software will be spotty, although evolving towards greater availability.
- Windows 8 will probably be exclusively 64-bit (next slide).

Windows 8: What's Next?

- **Windows 8 (fall, 2012 or spring, 2013) will be the biggest Microsoft release since Windows 95, according to some sources.**
- **Completely new start screen and "desktop."**
- **High-tech touch interface for touch-screen laptops or notepads.**
- **Will now support ARM ("advanced RISC Machine") so this will soon lead to hybrid laptops.**
- **Microsoft is also apparently going to encourage purchase by download, and has reduced the user interaction when installing Windows 8 dramatically from previous installations (like Windows 7!). Supposedly it now takes only 11 mouse clicks.**
- **No word on whether the download Windows will be cheaper. Don't hold your breath!**