Introduction To CPSC 231 & Computer Hardware

In this section of notes you will learn what are the basic parts of a computer and how they work.

Administrative (James Tam)

- Contact Information
  - Office: ICT 707
  - Email: tamj@cpsc.ucalgary.ca
- Office hours
  - Office hours: TR 14:00 – 14:50, (Right after class)
  - Email: (any time)
  - Appointment: email, phone or call
  - Drop by for urgent requests (but no guarantee that I will be in if it's outside of my office hours!)

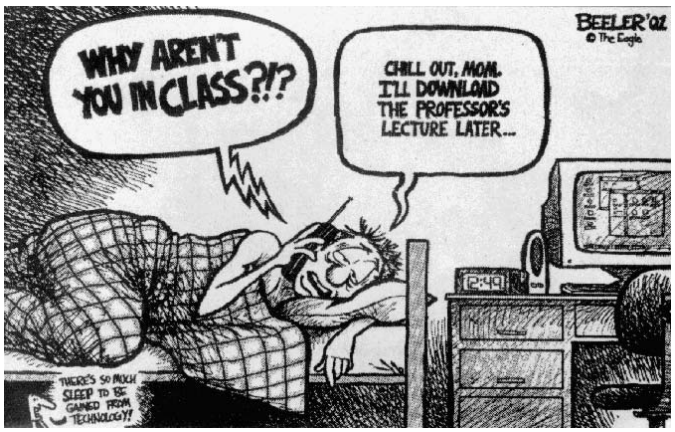
My Office

Course Resources

- Course website: http://pages.cpsc.ucalgary.ca/~tamj/231
- Course directory: /home/231/tamj (accessed via your UNIX Computer Science account)
- Recommended course textbook:
  - Pascal Programming & Problem Solving, 4th Edition, Leestma/Nyhoff (Prentice Hall)

How To Use The Course Resources

- They are provided to support and supplement this class.
- Neither the course notes nor the text book are meant as a substitute for regular attendance to lecture and the tutorials.



How To Use The Course Resources (2)

```
procedure add (var head    : NodePointer;
               var newNode : NodePointer);
var
  temp : NodePointer;
begin
  if (head = NIL) then
    head := newNode
  else
  begin
    temp := head;
    while (temp^.next <> NIL) do
      temp := temp^.next;
    temp^.next := newNode;
  end;
  newNode^.next := NIL;
end;
```


How To Use The Course Resources (2)

procedure add (var head : NodePointer;
var newNode : NodePointer);
var
begin
begin
temp := head;
while (temp^.next <> NIL) do
temp := temp^.next;
temp^.next := newNode;
end;
newNode^.next := NIL;
end;

If you miss a class make sure that you catch up on what you missed (get someone's class notes)

...when you do make it to class make sure that you supplement the slides with your own notes (cause you aint gonna remember it in the exams if you don't)

But Once You've Made An Attempt To Catch Up

- Ask for help if you need it
- There are no dumb questions

Me?

Me in first year?

Is This The Course The One For You?

- Introductory Computer Science courses for non-Computer Science majors (*do not want to get a Computer Science degree*)
  - CPSC 203
  - CPSC 217

- The introductory Computer Science course for Computer Science majors (*do wish to get a Computer Science degree*)
  - CPSC 231

CPSC 203

- **The focus is on how to *use* computer programs.**

- One important objective is to learn how computers and technology works *from the user's perspective*
  - Issues related to how computers work are largely introduced in the context of using applications.
    - E.g., Why is my computer so slow when I'm editing my movies?
    - E.g., Why did that computer game look and sound so much better on the store computer than on my machine at home?

- Assignments involve *using* popular software:
  - Productivity (business) software: MS-Office
  - Fun software: building a web site, making a computerized video etc.

CPSC 217

- **An introduction to *writing* computer programs for students whose major is not Computer Science ('non-techies')**
  - This person will not get a degree in Computer Science
  - This person will not develop/write software for a living (become a programmer)
  - This person may work with complex specialized software (e.g., running a biological simulation) which may require customization

- One important objective is to learn how computers and technology works *from the programmer's perspective*
  - Issues related to how computers work are largely introduced in the context of creating applications.
  - E.g., How do I write a program that will let me do my work on a computer?

- Assignments involve *writing* simple programs:
  - Possible examples:
    - Displaying text onscreen
    - Saving and reading information to/from a file
    - (Writing computer programs in the context of other disciplines): Creating a simulation (biological, chemical, economic, business)

CPSC 231

- **An introduction to *writing* computer programs for Computer Science majors**
  - This person will get a degree in Computer Science
  - This person will likely develop/write software for a living (become a programmer)

A computer geek

- Typically the course is more in-depth and cover more topics than CPSC 217.
- One important objective is to learn how computers and technology works *from the programmer's perspective*
  - Issues related to how computers work are largely introduced in the context of creating applications.
  - E.g., If I write my program one way it will run faster than if I write it another way?
- This may result in having more challenging assignments than the ones in CPSC 217.
  - (Writing computer programs in the context of Computer Science): writing a computer game, doing simple graphics etc.

A Bit More About CPSC 231

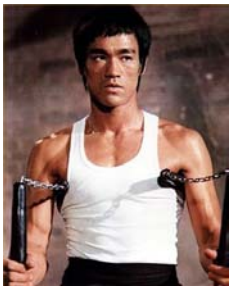
- It is a course geared primarily towards CPSC majors
- But it is not assumed that you have prior knowledge of Computer Science:
  - Almost all students do not have previous experience
  - Most students do get through the course
- It can be a lot of work

Wav file from "The Simpsons"

How To Succeed

•Successful people


Leonardo da Vinci


Bruce Lee


J.R.R. Tolkien


Amadeus Mozart

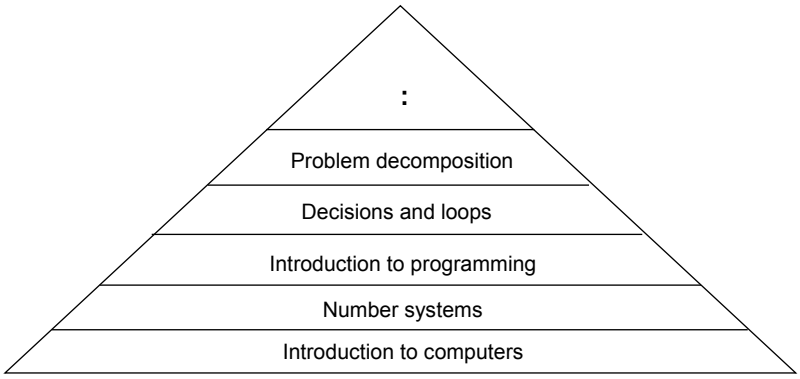

Wayne Gretzky

How To Succeed In This Course

1. Practice things yourself.
   - Write lots programs
     - At the *very least* attempt every assignment.
     - Try to do some additional practice work (some examples will be given in class, some practice assignments will be available on the course web page).
     - Write lots of little 'test' programs to help you understand and apply the concepts being taught.
   - Trace lots of code
     - Reading through programs that other people have written and understanding how and why it works the way that it does.

How To Succeed In This Course (2)

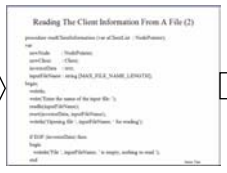
2. Make sure that you keep up with the material
   - Many of the concepts taught later depend upon your knowledge of earlier concepts.
   - Don't let yourself fall behind!
   - *At least* attempt all assignments!



James Tam

How To Succeed In This Course (3)

3. Look at the material before coming to lecture so you have a rough idea of what I will be talking about that day:
   a) Read the slides
   b) Look through the textbook (if you bought it)



...

James Tam

How To Succeed In This Course (4)

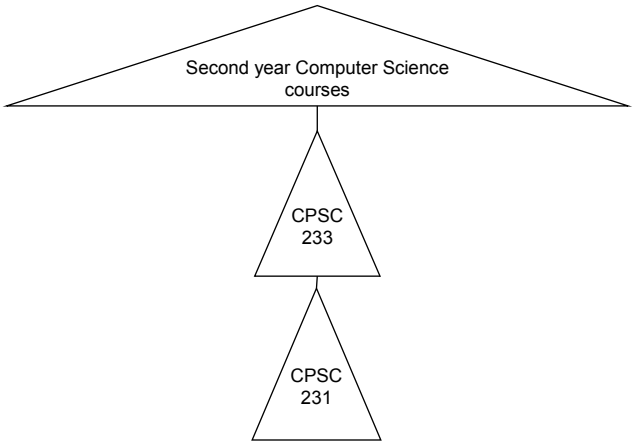
4. Start working on things as early as possible:
   - Don't cram the material just before the exam, instead you should be studying the concepts as you learn them throughout the term.
   - Don’t start assignments the night (or day!) that they are due, they may take more time than you might first think so start as soon as possible.

How To Succeed In This Course: A Summary

1. Practice things yourself
2. Make sure that you keep up with the material
3. Look at the material before coming to lecture
4. Start working on things early

How To Succeed In Computer Science: Additional Note For Computer Science Majors

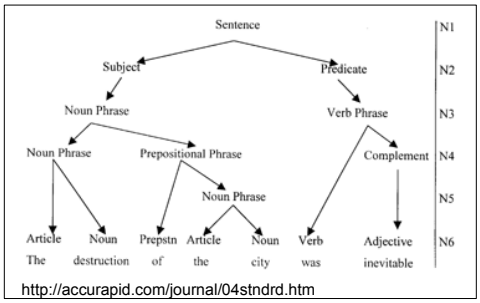
- The material in this course is used as a foundation for material in later courses
  - It will be assumed in the subsequent classes that you already know the material in this class.



You Will Learn About Program Structure

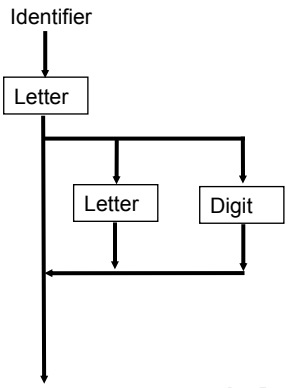
- The required structure and rules for a creating a computer program (*the syntax of the language*)
- You will then need to apply these principles throughout the term

Learning the rules of the English language: grammar



http://accurapid.com/journal/04stndrd.htm

Learning the rules of the computer language: syntax



You Will Learn About Programming Style

- Learning good programming practices
- Learning why something is regarded as 'good' or 'bad' style

Bad ☹

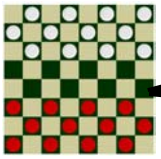
```
program p;
var
  x : integer;
  y : integer;
begin
  while (true) do
  begin
    if (x < 0) then
      break;
      :
  end;
end.
```

Good ☺ (or at least better)

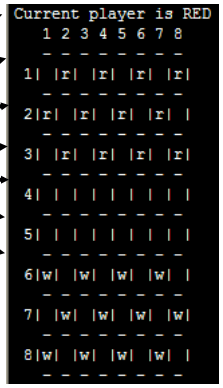
```
(*
   This program is a simple interest
    calculator
*)
program banking;
begin
  var interest      : real;
  var principle     : real;
  var amount        : real;
  var time          : real;
  amount := principle * rate * time;
       :
end.
```


You Will Learn How To Problem Solve

- With a knowledge of programming practices and programming style you will work out solutions to given problems (e.g., assignments).

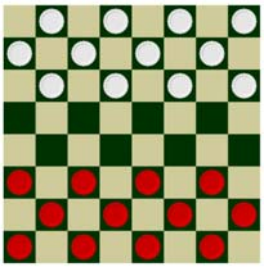


Assignment: create a computerized checkers game (western version).

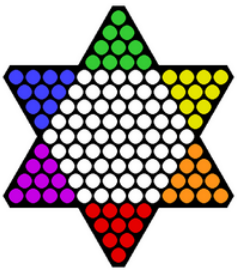


Solution: A text-based Pascal program.

Problem Solving Is A Skill That You Need To Learn And To Improve Upon



Your assignment: Implement a checkers game that follows European rules

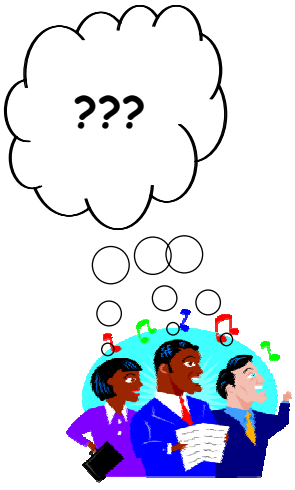


Lecture example: A partial implementation of the Chinese Checkers game

The Problems Are Not Impossible For Beginners (They Just Seem That Way At First :p)

The Game of Life	
European checkers	
The Quest for the Fountain of Fulfillment	
Star Trek, mission: Find the dilithium and save the earth	
The Lord of The Rings: Quest to Mount Doom	
Star Wars: The Assault on the Death Star	
The Lord of The Rings: The Journey Through Khazad-dum	
The Hobbit	
Squirt the weed!	

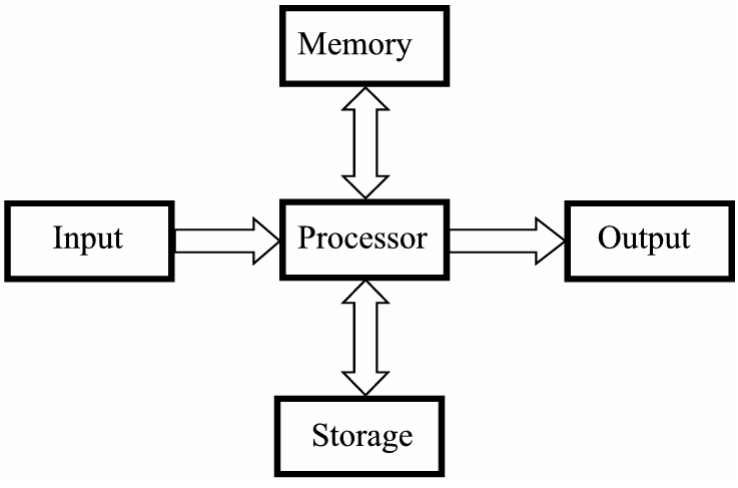
Feedback



Dilbert © United Features Syndicate

James Tam

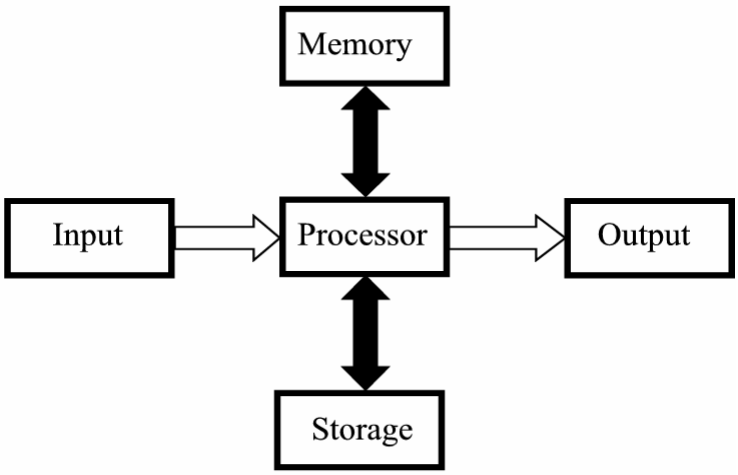
High Level View Of A Computer



James Tam

Buses

•Connect the different parts of the computer together



Types Of Buses

•Data buses
- Are used to transmit information to the different parts of the computer

•Address buses
- Indicate where the information is supposed go

Buses (2)

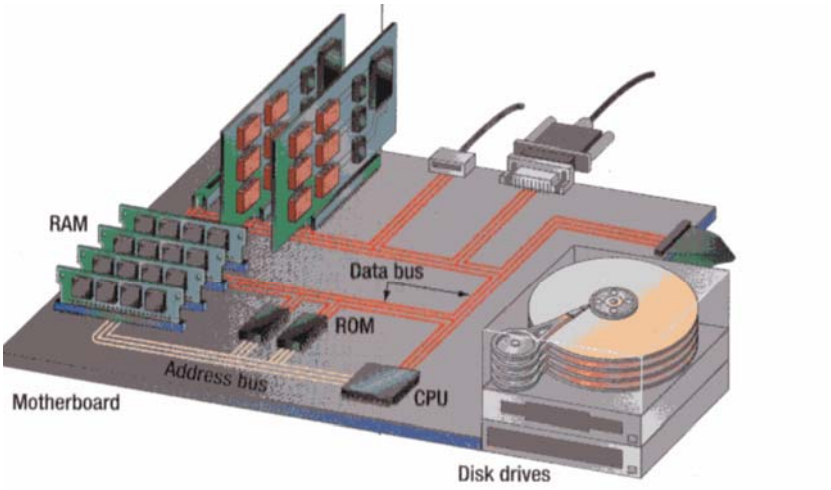


Image from Peter Norton's Computing Fundamentals (3rd Edition) by Norton P.

James Tam

Basic Units Of Storage

Bit  on OR off

- *Bi*nary digi*t*
- Smallest unit of measurement
- Two possible values

Byte 

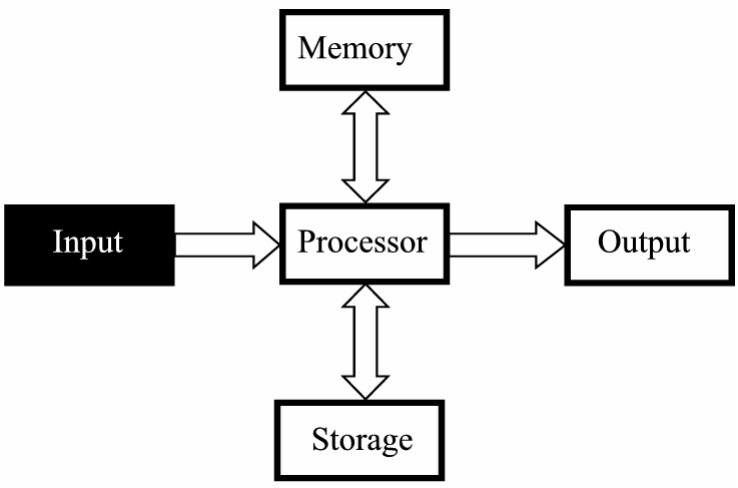
- 8 bits

Word

- The word size of a computer is the number of adjacent bits that can be stored and manipulated as a unit
- 32, 64 for home computers, 128 for faster machines or specialized systems

James Tam

Input



Input Devices

•Used by a person to communicate to a computer.

Person to computer

Example Input Devices

- Keyboard 
- Mouse 
- Etc.

How Keyboard Input Works

Keyboard: A key is pressed

The electrical impulse is sent via a wired or wireless connection

Keyboard controller: based on the electrical impulses it determines which key or combination of keys was pressed

A	p	p	l	e	...

Keyboard buffer: stores the keystrokes

The keyboard controller transmits an interrupt request

Operating system

How Keyboard Input Works

Operating system:

Q: Is the key combination a (an operating) system level command e.g., <alt>-<ctrl>-<del>?

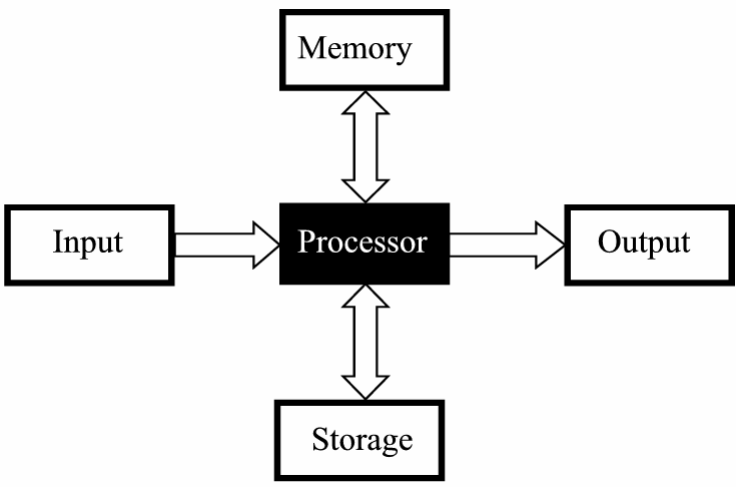
Yes

Execute operating system instruction

No

Pass the key combination onto current application

Processor



Processor

•The brains of a computer

Image from:
www.howstuffworks.com

•A common desktop processor

Small Units Of Measurement

•Milli: one thousandth
•Micro: one millionth
•Nano: one billionth

Small Units Of Measurement (Processor And Memory Speed)

- Millisecond (ms) – a thousandth of a second ($1/1{,}000 = 10^{-3}$)
- Microsecond (μs) - a millionth of a second ($1/1{,}000{,}000 = 10^{-6}$)
- Nanosecond (ns) – a billionth of a second ($1/1{,}000{,}000{,}000 = 10^{-9}$)

Processor Speed

- Determined by:
  1. Type of processor e.g., Intel: Celeron, Pentium; AMD: Athlon, Opteron
  2. Clock speed
     - 1 Hz = 1 pulse is sent out each second (1 second passes between each pulse)
     - 10 Hz = 10 pulses are sent out each second (0.1 seconds passes between each pulse)
     - :
     - 25 MHz = 25 million pulses sent out each second (0.000 000 04 seconds between each pulse or 40 ns between pulses)
     - 3.8 Ghz = 3.8 billion pulses sent out each second (0.26 ns between pulses)

The Processor And The Computer

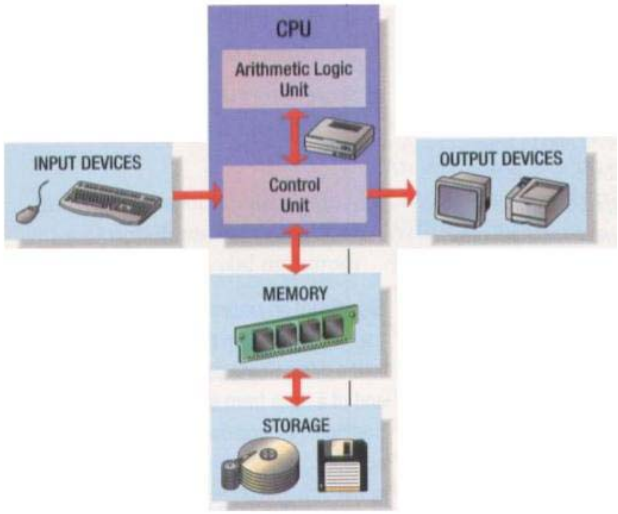
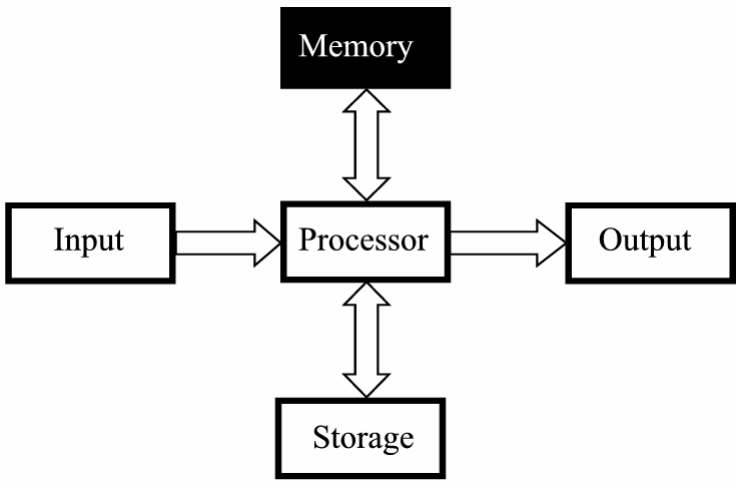


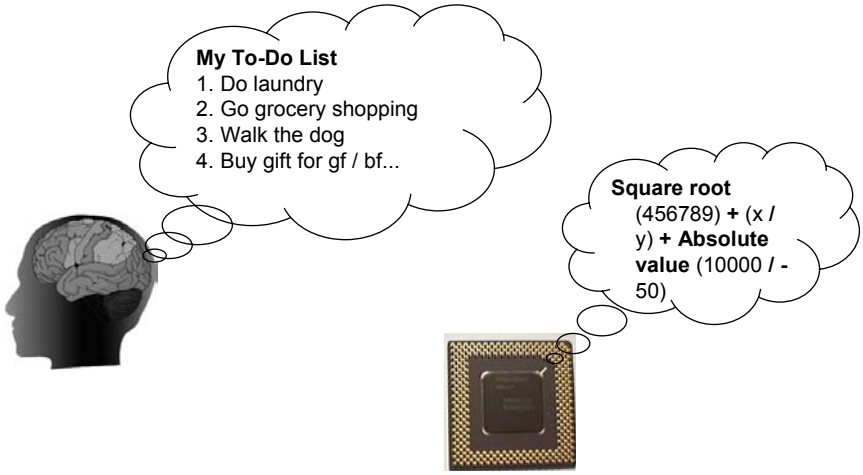
Image from Peter Norton's Computing Fundamentals (3rd Edition) by Norton P.

Memory



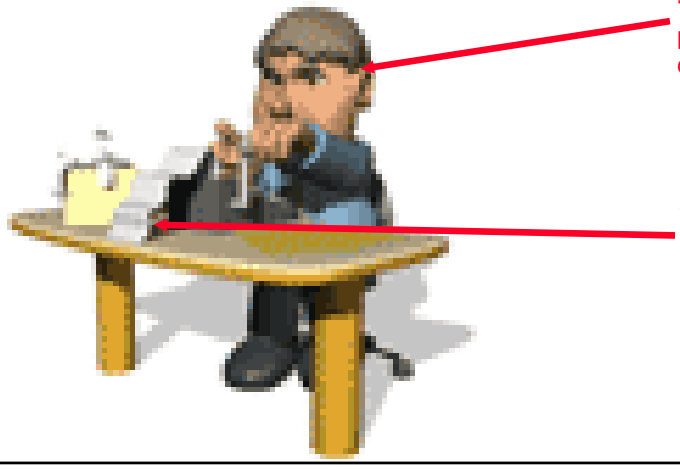
Memory

- The processor has a small amount of memory that is fast but very low in capacity



Memory

- It is used as temporary storage for storing information and instructors that won't fit in the processor but is needed now



Processor: 'brains' that performs the calculations

Memory: stores information needed by the processor

Memory: High-Level View

•Storing information in memory is based on bits (on/off state)

 on OR off

•Since bits cannot store enough information bits are combined into bytes

Large Units Of Measurement

•Kilo: One thousand
•Mega: One million
•Giga: One billion
•Tera: One trillion

Large Units Of Measurement (Memory, Storage)

- Note: powers of two are used because computer memory and storage are based on the basic unit (bit).
- Kilobyte (KB) – a thousand bytes ($1,024 = 2^{10}$)
- Megabyte (MB) - a million ($1,048,576 = 2^{20}$)
- Gigabyte (GB) – a billion ($1,073,741,824 = 2^{30}$)
  - ~ A complete set of encyclopedias requires about 700 MB of storage
  - ~ 30 minutes of video (~1/4 of the information stored on a typical DVD)
- Terabyte (TB) – a trillion ($1,099,511,627,776 = 2^{40}$)
  - ~ 20 million four-drawer filing cabinets full of text
  - ~ 200 DVD's (standard) of information

James Tam

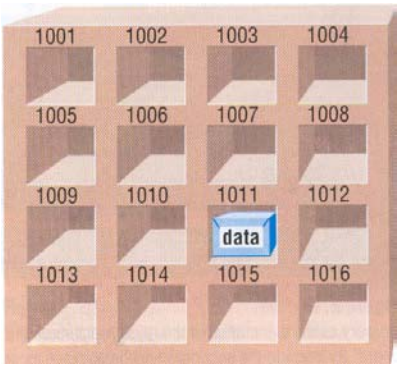
RAM

- *R*andom *A*ccess *M*emory
- Volatile
  - Used for temporary storage
- Typical ranges 256 MB - 4 GB

James Tam

RAM (2)

- Random access means direct access to any part of memory
- A common form of RAM is DRAM (Dynamic RAM)



Random access doesn't mean chaotic or haphazard but it means that access does not have to be sequential but can occur anywhere

Picture from Computers in your future by Pfaffenberger B

How Does DRAM Work?

- Acts like a leaky bucket

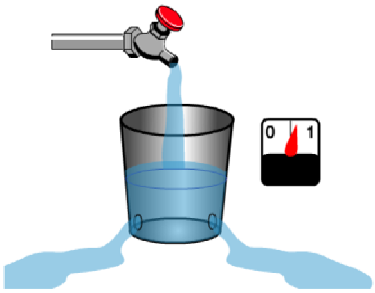


Image from www.howstuffworks.com

How Does DRAM Work?

•Acts like a leaky bucket

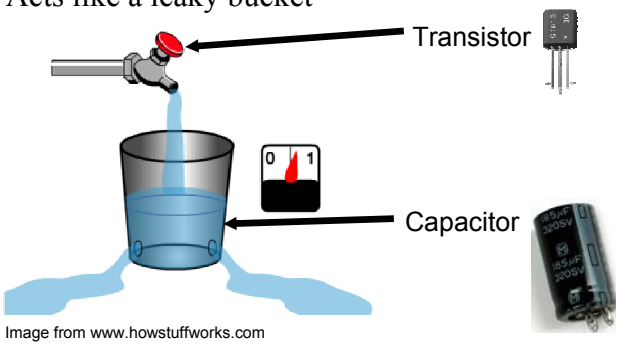
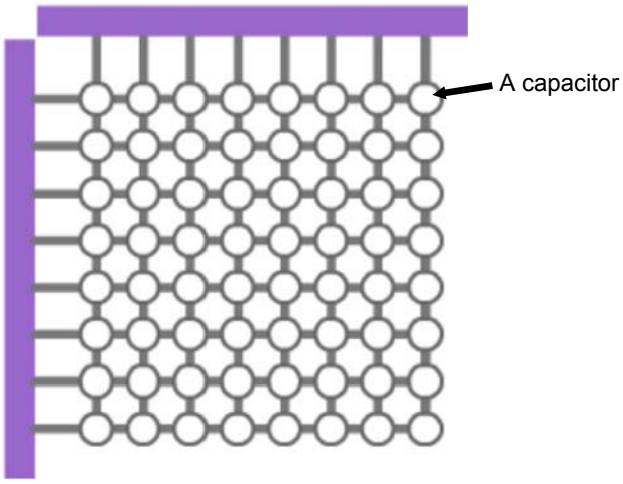


Image from www.howstuffworks.com

DRAM: A Collection Of Capacitors



Conceptual view of DRAM

Actual DRAM

Recall: The Smallest Useful Unit Of Storage

- Bits, bytes, word size
  - Bits are too small to be useful so 8 bits are collected into one byte
  - This is the smallest useful 'slot' in memory for storing information

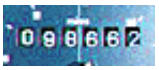
- Q: What determines the maximum number of slots or bytes in a computer?

The (Address) Word Size Of The Computer Determines The Maximum Amount of RAM

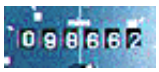
- Given that byte sized slots are used to store information how many slots can exist in a computer?
- Recall
  - $2^{30} \sim 1$ billion
  - $2^{31} \sim 2$ billion
  - $2^{32} \sim 4$ billion
  - This means that with a 32 bit computer the maximum amount of memory allowable is 4 billion (4 GB).

The (Data) Word Size Of The Computer Determines The Size Of The Information That Can Be Stored

- It determines how many byte sized slots can be combined to store information.
- Examples:
  - A computer that stores data with 16 bits (2^{16} = 65536 combinations)
    - 32,000 * 2: Okay
    - 65535 * 2: Not okay

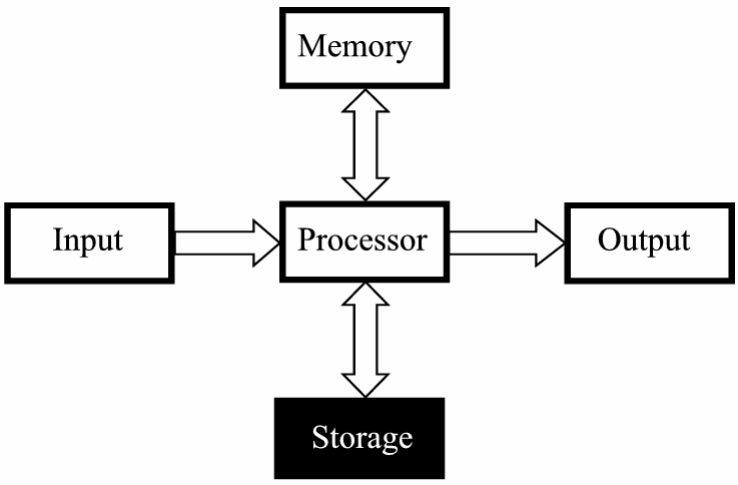


  - A computer that stores data with 32 bits ($2^{32} \approx$ 4 billion combinations)
    - 2 billion + 2 billion: okay
    - 3 billion + 3 billion: not okay



James Tam

Storage



James Tam

Storage Vs. Memory

Memory (e.g., RAM)

- The information stored is needed now
- Keep the information for a shorter period of time (usually volatile)
- Faster
- More expensive
- Low storage capacity (~1/4 of a DVD for 1 GB)

Storage (e.g., Hard disk)

- The information stored is not needed immediately
- The information is retained longer (non-volatile)
- Slower
- Cheaper
- Higher storage capacity (~50 standard DVD's for 200 GB)

James Tam

Categories Of Storage

1. Magnetic
   - Floppy disks
   - Zip disks
   - Hard drives
2. Optical
   - CD-ROM
   - DVD
3. Solid state storage devices
   - USB Key (a very common form of solid state storage)

James Tam

1. Magnetic Drives

1. Magnetic Drives: Storage Capacities

- Floppy disks
  - ~ 1 MB
- Zip disks
  - 100, 250, 750 MB
- Hard drives
  - ~ Hundreds of GB – 2+ TB

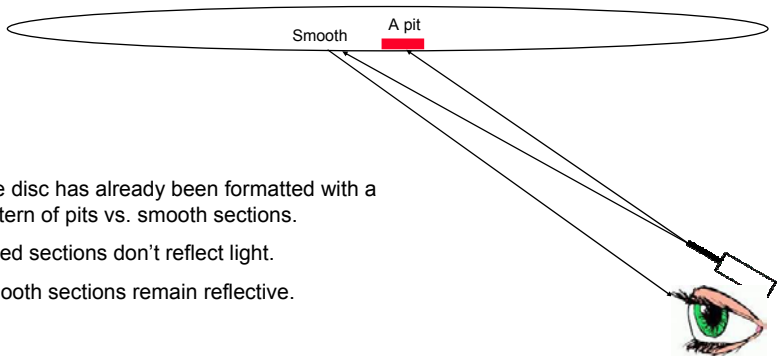
2. Optical Drives

- Use lasers to store and retrieve information (CD's and DVD's)
- Categories:
  - Can only read information off the disc (CD-ROM, DVD-ROM)
  - Can read and also *r*ecord information to the disk (CD-R, DVD-R, DVD+R)
  - Can read, record and also *r*e-*w*rite information multiple times (CD-RW, DVD-RW, DVD+RW)
- Storage capacities:
  - CD ~ 700 MB
  - DVD ~ 4 GB (drives with much larger capacities are becoming more common)

James Tam

Optical Drives: Reading Information

CD-ROM, DVD-ROM

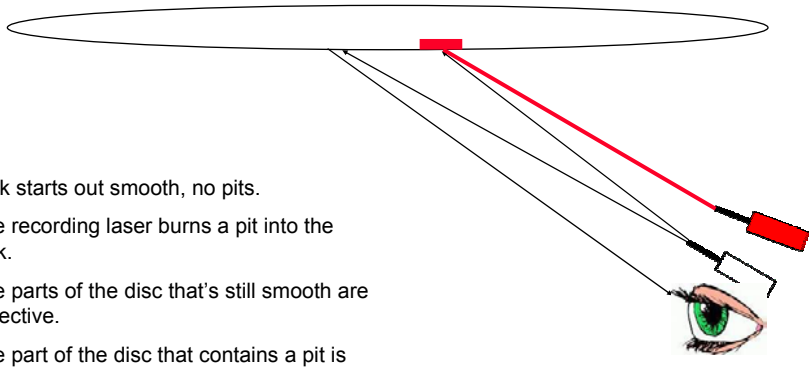


- The disc has already been formatted with a pattern of pits vs. smooth sections.
- Pitted sections don't reflect light.
- Smooth sections remain reflective.

James Tam

Optical Drives: *R*ecording And Reading Information

CD-R, DVD-R, DVD+R

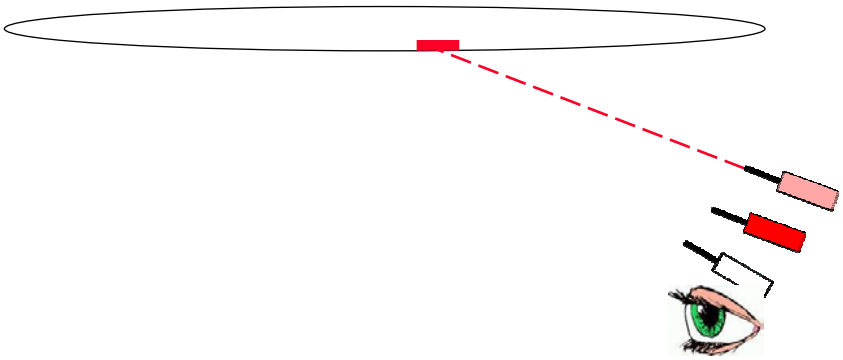


- Disk starts out smooth, no pits.
- The recording laser burns a pit into the disk.
- The parts of the disc that's still smooth are reflective.
- The part of the disc that contains a pit is non-reflective.

James Tam

Optical Drives: *R*e-*W*riting

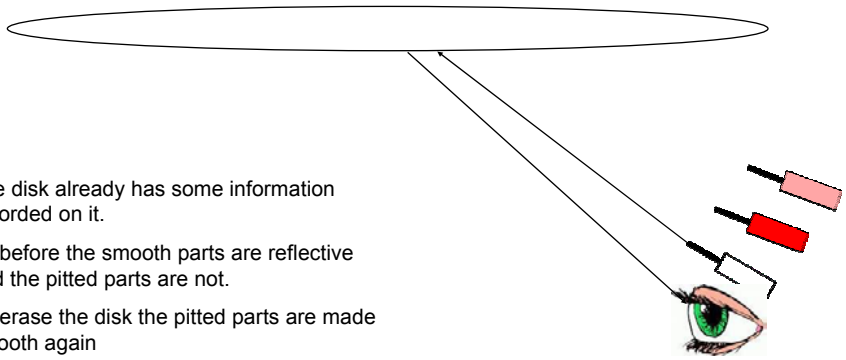
CD-RW, DVD-RW, DVD+RW



James Tam

Optical Drives: *R*e-*W*riting

CD-RW, DVD-RW, DVD+RW



- The disk already has some information recorded on it.
- As before the smooth parts are reflective and the pitted parts are not.
- To erase the disk the pitted parts are made smooth again

3. Solid State Storage Devices

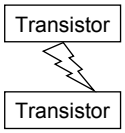
•Portable but can store a large amount of information (256 MB – 16 GB)

3. Solid State Storage Devices

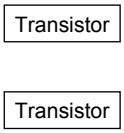
- Require no moving parts but instead uses transistors

- Use a pair of transistors to store each bit of information

Connected: stores '1'

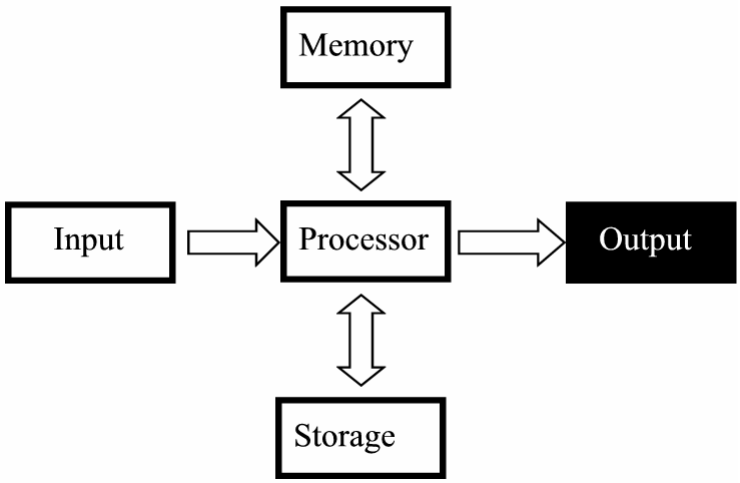


Disconnected: stores '0'



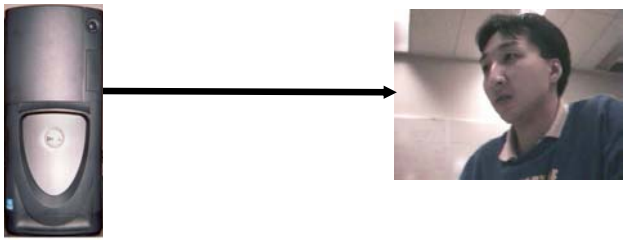
- An electrical current can be used to connect and disconnect the transistors
- The pair transistors will remain in their current state (connected or disconnected) until an electrical charge is applied.

Output



Output Devices

•Displays information from the computer to a person.



The Most Common Output Device: The Monitor

- Common monitor technologies:
  1. CRT (Cathode Ray Tube)

  2. LCD (Liquid Crystal Display)

  3. Plasma displays

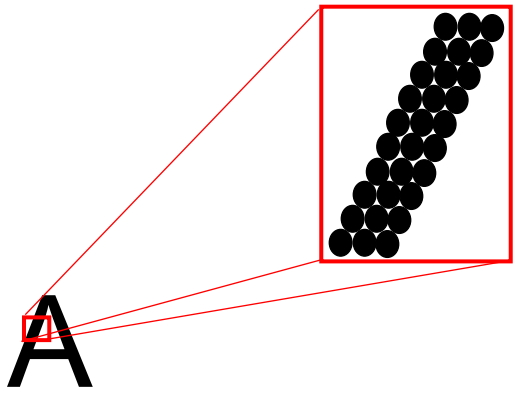
How Images Are Drawn On Monitors

- Images and text are drawn with tiny dots (Pixels: *Pi*cture *el*ement*s)*

A

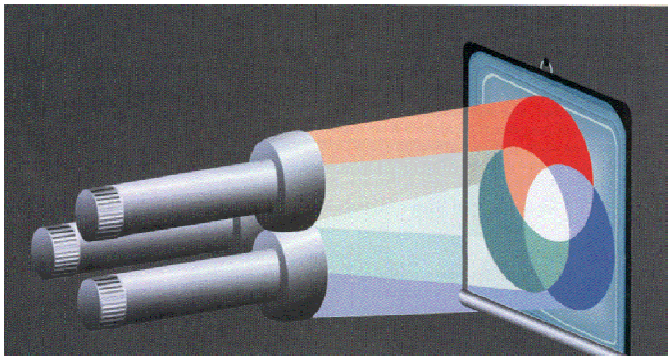
How Images Are Drawn On Monitors

- Images and text are drawn with tiny dots (Pixels: *Pi*cture *el*ement*s)*



1. CRT Monitors

•The pixels are drawn with light ‘guns’

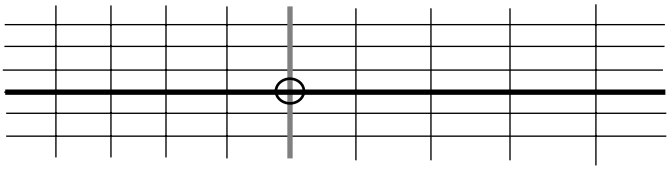


Picture from Computer Confluence by Beekman G.

2. LCD Monitors

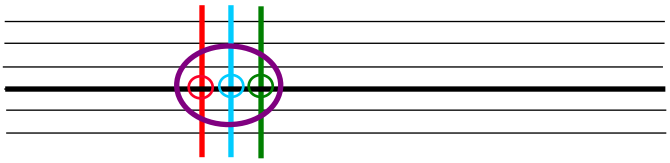
•Employ a conductive grid for each row and column

•The meeting of a row and column allows light to be emitted (a pixel can be seen)



2. Colour LCD Monitors

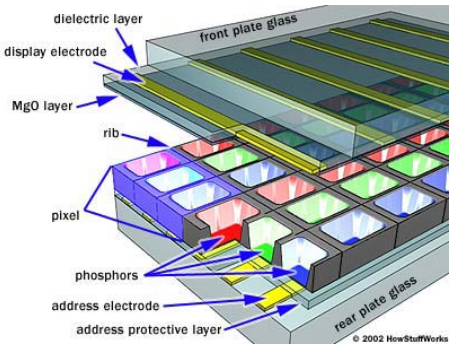
•Use three sub pixels:
- One wire for each row
- One wire for each sub-pixel
- One colour filter for each colour (red, blue, green)

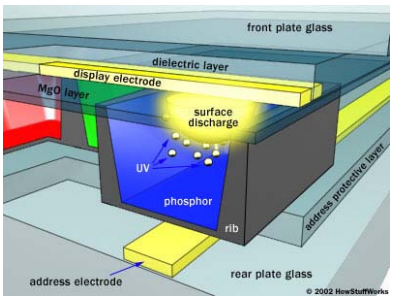


3. Plasma Monitors

•Sub-pixels are "drawn" by passing an electrical current through a gas.

•Again each pixel is formed by three sub-pixels





Images from www.howstuffworks.com

All The Basic Parts Together

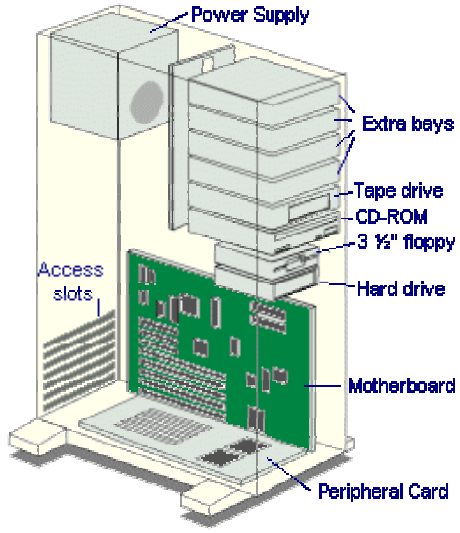
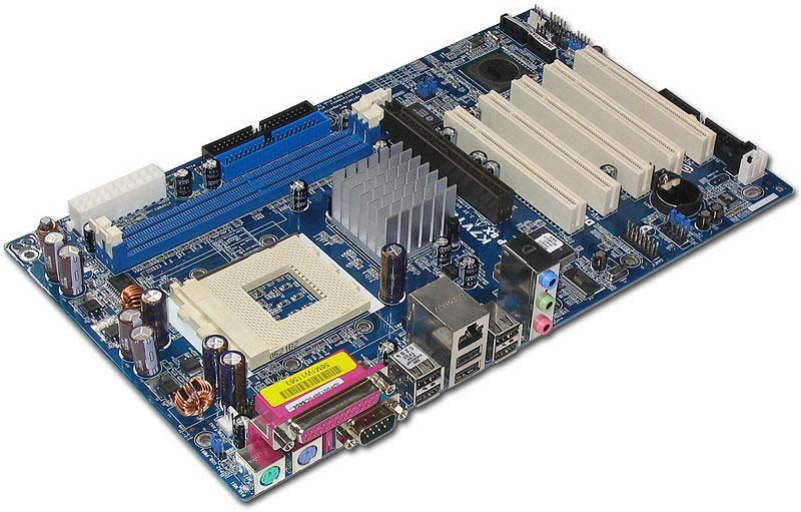


Diagram from http://www.jegsworks.com

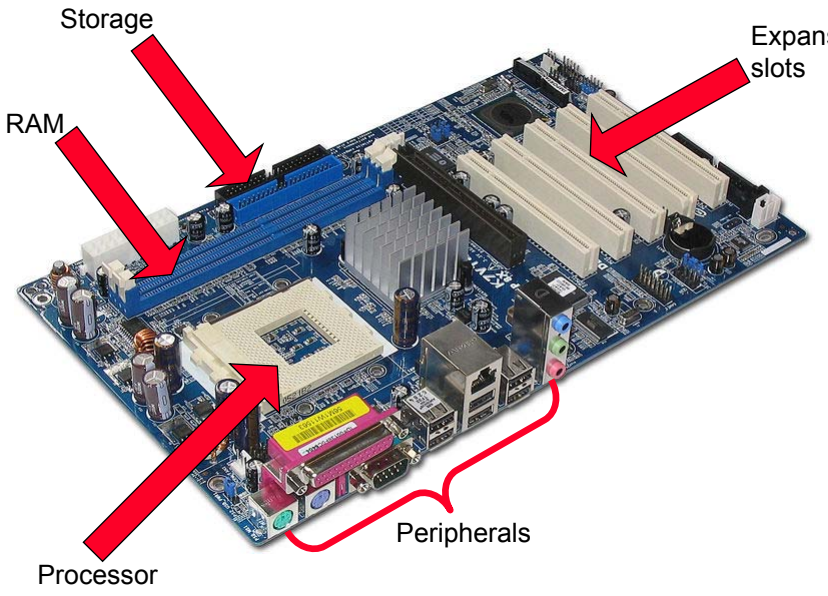
James Tam

The Motherboard



James Tam

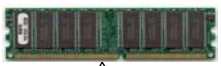
The Motherboard



James Tam

Relating The Speed Of The Computer To Its Components

Storage:
Hard drive



Memory:
RAM

Processor

James Tam

You Should Now Know

- What are common units of measurement for the computer
- What are the basic parts of the high level view of a computer
- Example input devices
- The role of the processor in a computer
- What determines processor speed
- What are the characteristics of RAM
- How does DRAM work
- The difference between storage and memory
- What are the different categories of storage devices as well as common examples of each
- How do different storage devices work
- The approximate storage capacity of memory and different storage devices
- How do computer monitors work
- How the different hardware components affects the speed of the system