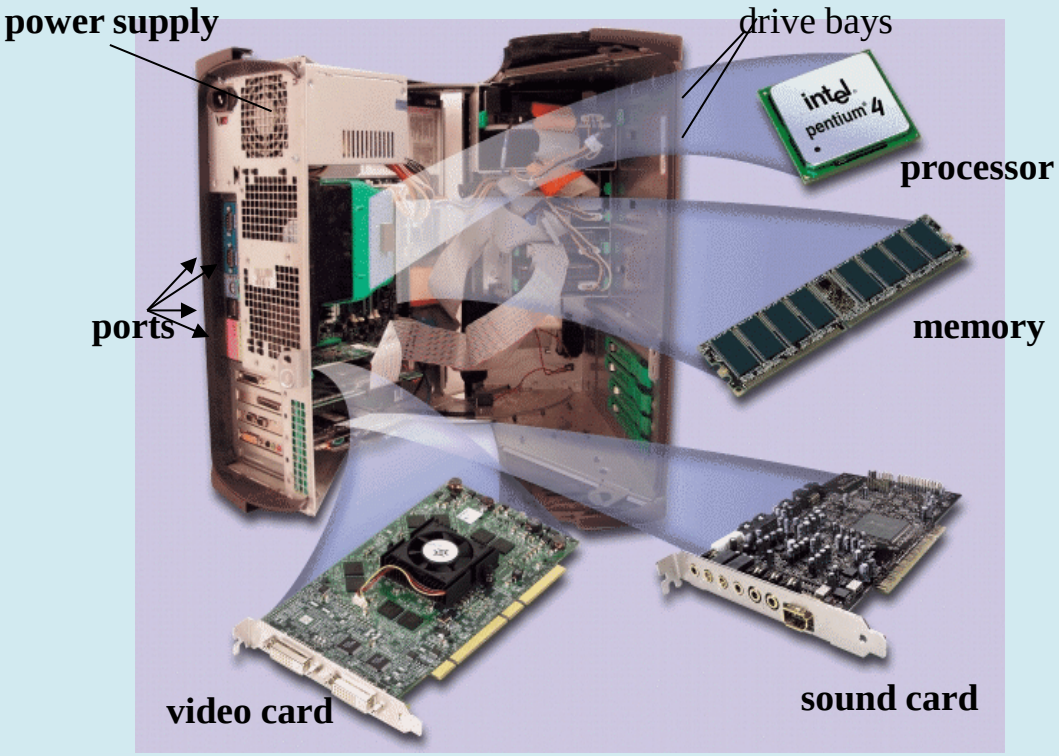
Lecture 1
System Architecture 2
DM2112
Digital Entertainment Systems

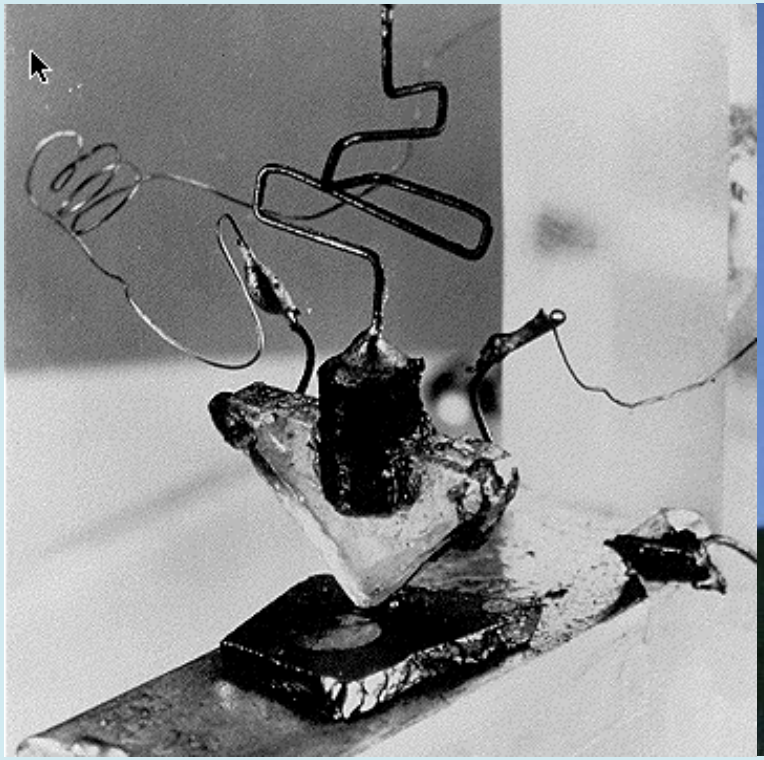
PC Today

- Today's PC are designed with customisable parts



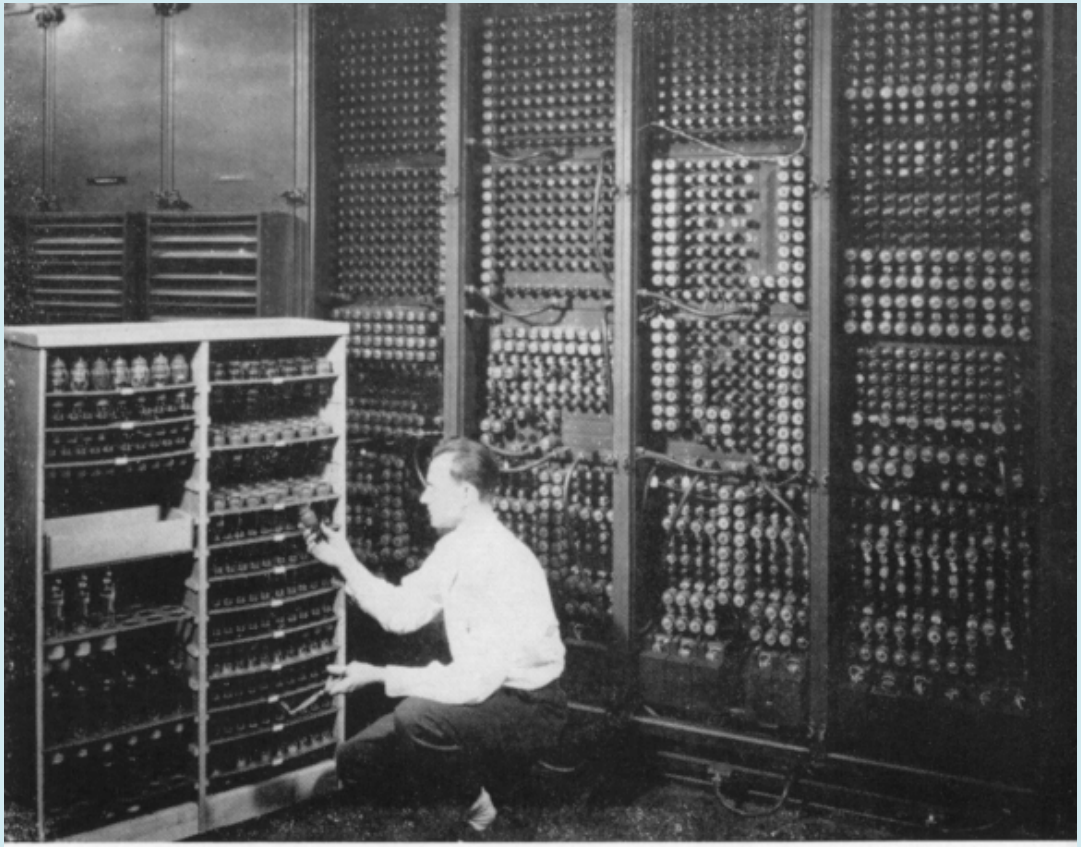
Transistor

- It all began with the transistor
  - Basic building block of electronics today

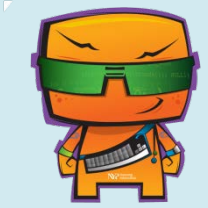


Transistor

- First transistor were in vacuum tubes



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

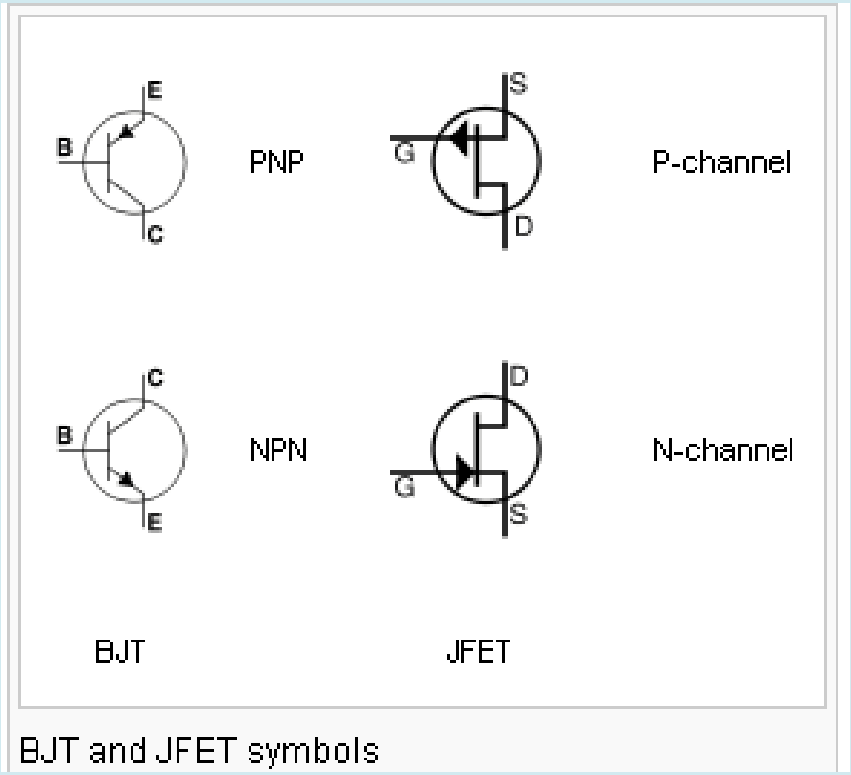


Transistor

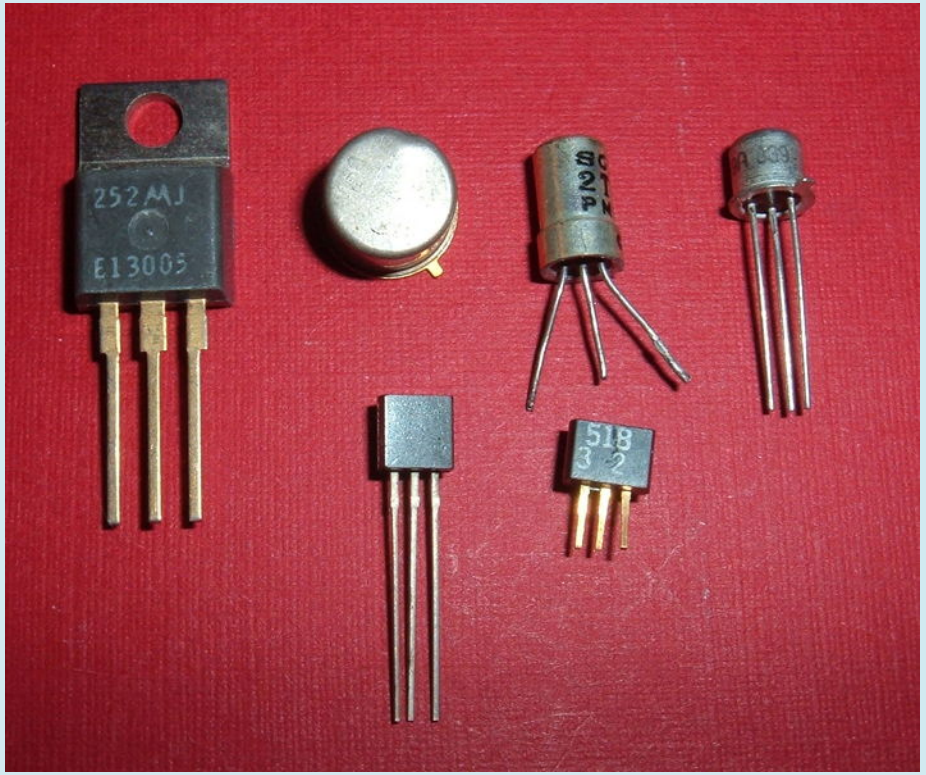
- Silicon-based transistors were key to creating complex circuitry required for modern processors
- Today's processors easily have hundreds of millions of transistors

Transistor

- Examples of Transistors



BJT and JFET symbols



History of Computer

- Have a read of the history of computers
- http://www.computersciencelab.com/ComputerHistory/History.htm

Graphic Cards

Graphic Cards

- Computer Expansion Card
- Dedicated graphic rendering device
- Graphics Processing Unit (GPU)
  - Accelerate building of images in a frame buffer for output to a display
  - First GPU is GeForce 256 by Nvidia in 1999

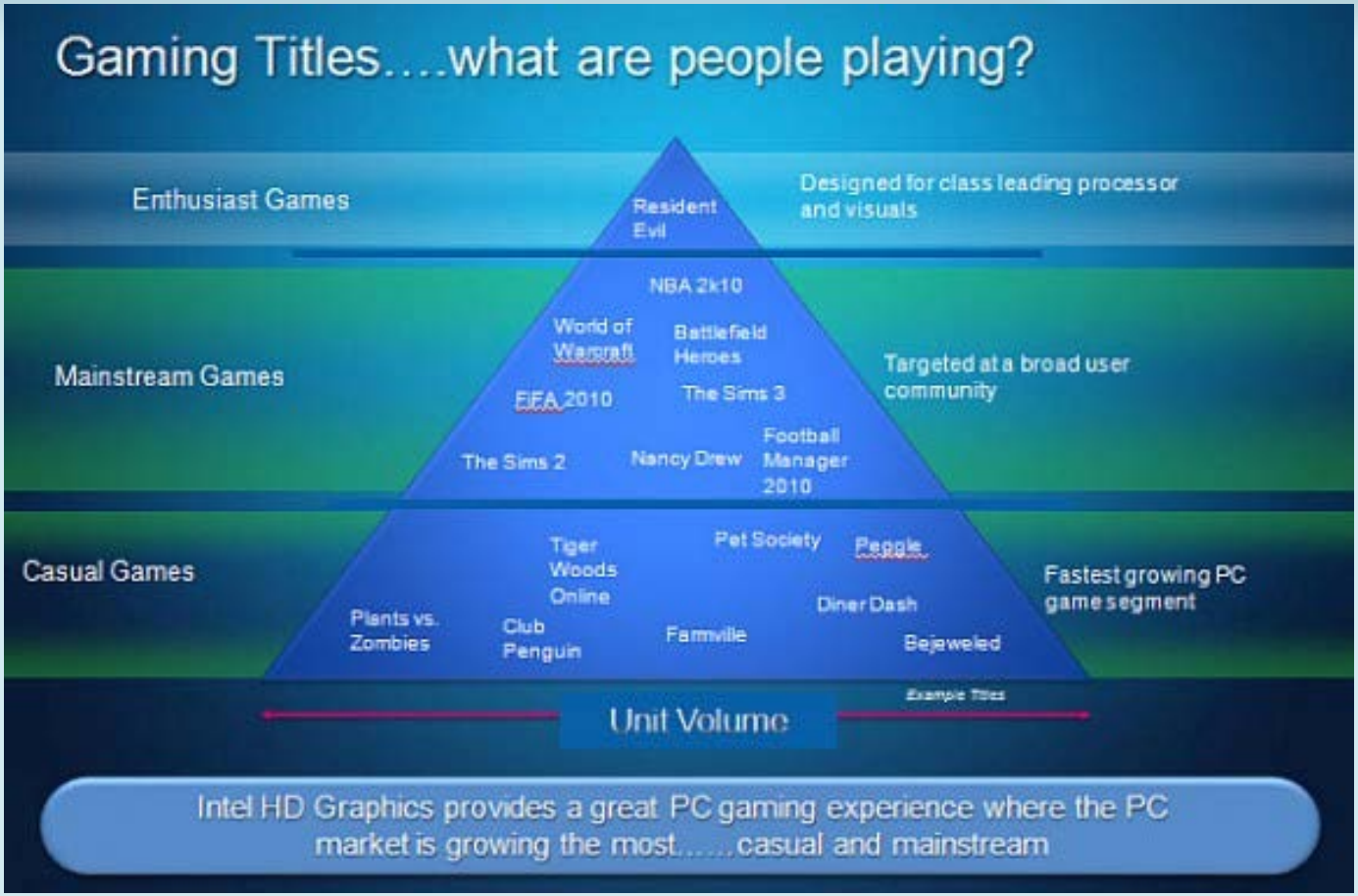
Intel Integrated Graphics

- Built into the motherboard's northbridge
  - Will touch on northbridge in future
- Later moved to the CPU

- 2010 – HD Graphics released
- 2011 – HD Graphics 2000, 3000 released with Sandy Bridge
- 2012 – HD Graphics 2500, 4000 releasing with Ivy Bridge

Intel Integrated Graphics

- Intended for Mainstream Games



http://www.geek.com/articles/games/leaked-slide-shows-intel-hd-graphics-aimed-squarely-at-mainstream-games-20091230/

Dedicated Graphic Cards

- Computer expansion card dedicated to fast rendering of images for realtime display on computer display

- Has its own processor (GPU)
- Commonly handles vector operations
  - Will touch on this in future topics

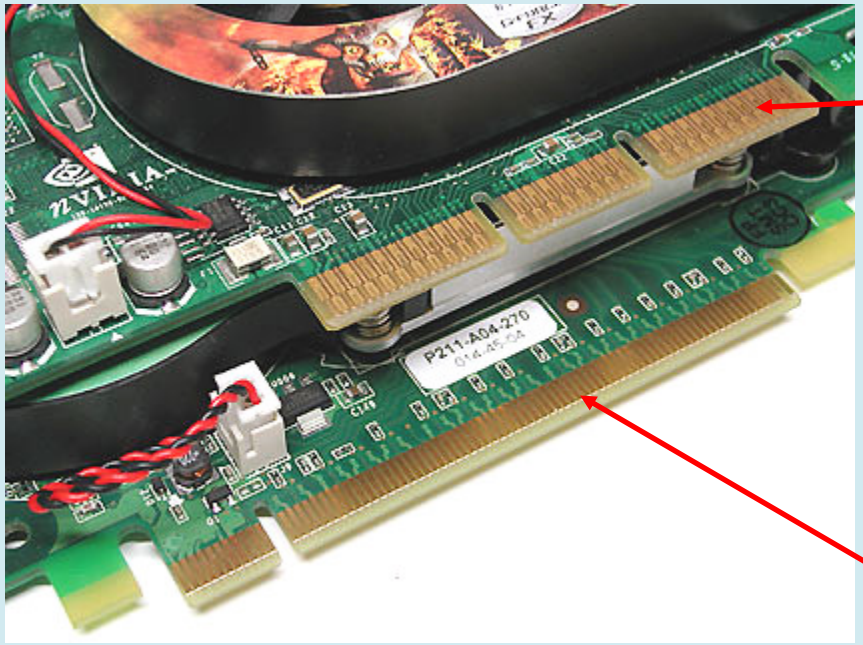
Graphic Cards Connection

- Taken from wiki
- http://en.wikipedia.org/wiki/Video_card#Motherboard_interface

Chronologically, connection systems between video card and motherboard were, mainly:

- S-100 bus: designed in 1974 as a part of the Altair 8800, it was the first industry-standard bus for the microcomputer industry.
- ISA: Introduced in 1981 by IBM, it became dominant in the marketplace in the 1980s. It was an 8 or 16-bit bus clocked at 8 MHz.
- NuBus: Used in Macintosh II, it was a 32-bit bus with an average bandwidth of 10 to 20 MB/s.
- MCA: Introduced in 1987 by IBM it was a 32-bit bus clocked at 10 MHz.
- EISA: Released in 1988 to compete with IBM's MCA, it was compatible with the earlier ISA bus. It was a 32-bit bus clocked at 8.33 MHz.
- VLB: An extension of ISA, it was a 32-bit bus clocked at 33 MHz.
- PCI: Replaced the EISA, ISA, MCA and VESA buses from 1993 onwards. PCI allowed dynamic connectivity between devices, avoiding the manual adjustments required with jumpers. It is a 32-bit bus clocked 33 MHz.
- UPA: An interconnect bus architecture introduced by Sun Microsystems in 1995. It had a 64-bit bus clocked at 67 or 83 MHz.
- USB: Although mostly used for miscellaneous devices, such as secondary storage devices and toys, USB displays and display adapters exist.
- AGP: First used in 1997, it is a dedicated-to-graphics bus. It is a 32-bit bus clocked at 66 MHz.
- PCI-X: An extension of the PCI bus, it was introduced in 1998. It improves upon PCI by extending the width of bus to 64-bit and the clock frequency to up to 133 MHz.
- PCI Express: Abbreviated PCIe, it is a point to point interface released in 2004. In 2006 provided double the data-transfer rate of AGP. It should not be confused with PCI-X, an enhanced version of the original PCI specification.

Graphic Cards Connection



GPU (Graphics Processing Unit)

- GPUs operate on a pipeline
- Prior to shaders, this pipeline is fixed

- Programmable step = shaders
  - Vertex shader
  - Pixel/Fragment shader
  - More on these in other modules

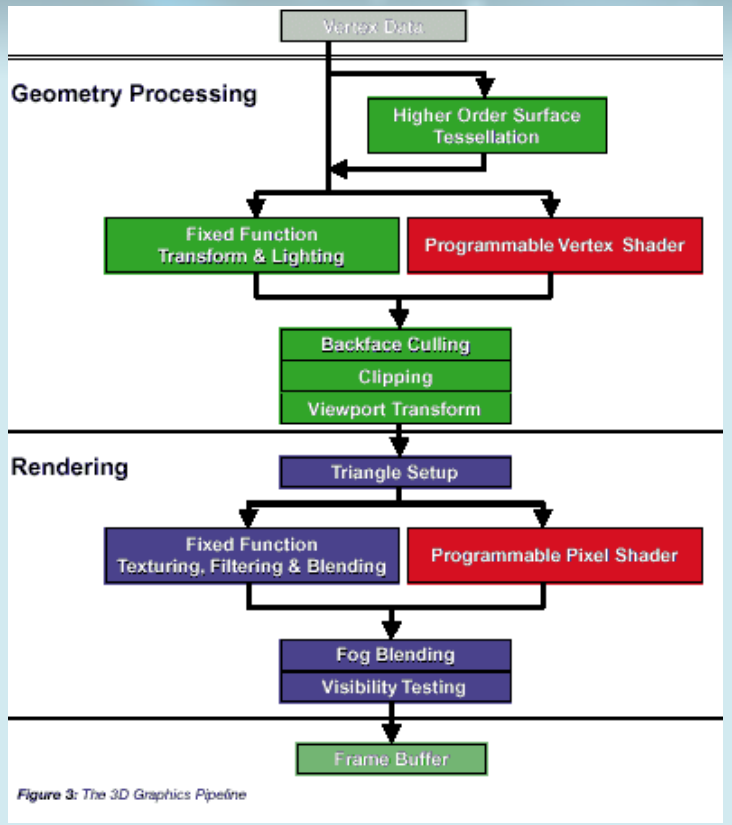
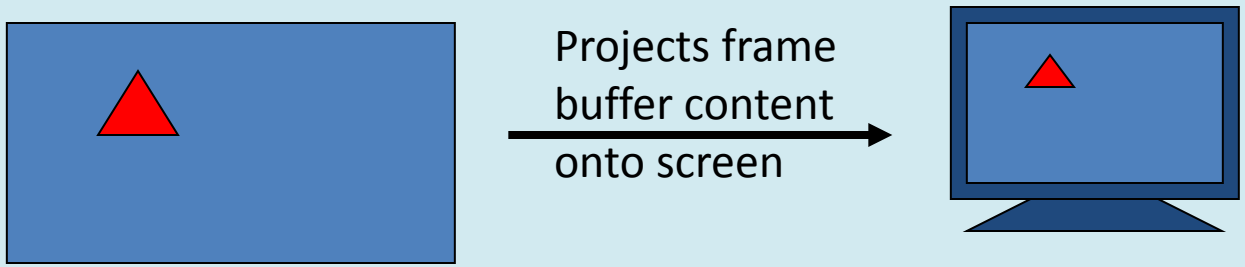


Figure 3: The 3D Graphics Pipeline

Work of a GPU

- Takes graphic information
  - i.e. graphics primitives – building blocks to create complicated scenes
  - 2D primitives – lines, circles
  - 3D primitives – points (vertices), lines, triangles, quads

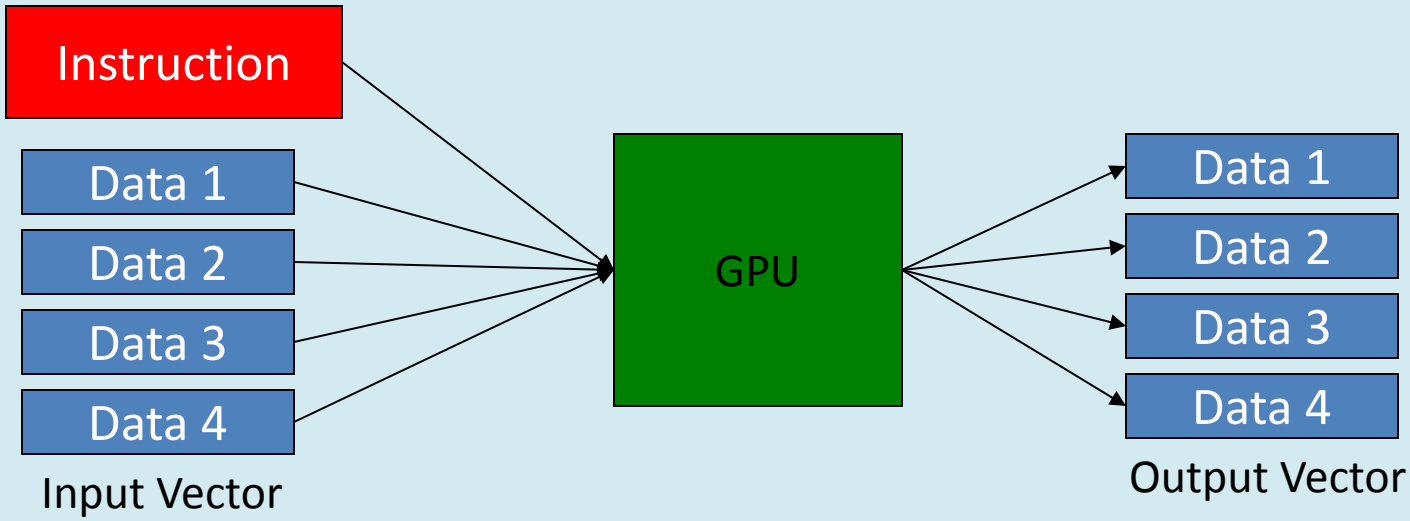
- Draw these on a memory space called a frame buffer



Scalar vs Vector Processing

Scalar vs Vector Processing

- GPUs commonly process multiple sets of data units at once (i.e. as a vector)



- Vector processing is also commonly used today on CPUs

Scalar vs Vector Processing

- Scalar processing handles the processing of 1 piece of data at a time.
- Vector processing handles a chunk of data at a time. Large amounts of data are processed simultaneously.
- Vector processing is extremely useful when processing graphics data for realtime display
  - Usually repetitive task to be done on large sets of data
    (e.g. set a colour value to all pixels on screen)
  - Perform in vector = can be processed faster

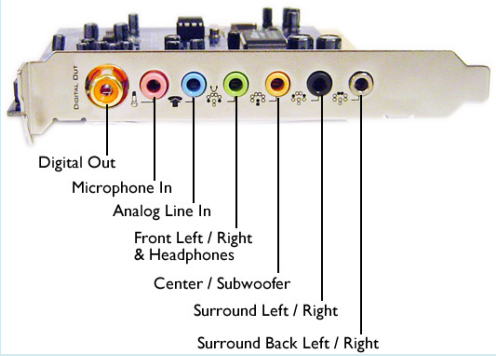
Scalar vs Vector Processing

- But if vector processors are better than scalar processors, why are scalar processors still in use?
  - Only useful for data-intensive task that require vector processing
  - Most of our daily computing processes tends to be sequential in nature
    - single data-single instruction

Sound Cards

Sound Cards

- Used to playback audio and convert real-world audio into computer data



- Contains
  - Audio converters (ADC & DAC)
  - A chip for wavetable synthesis (MIDI music)
  - Maybe a chip to manipulate the audio

- Important characteristics
  - Quality of converters
  - Sampling rate
  - Sampling frequency
  - Polyphony (of MIDI playback)
  - Quality of wavetable

Common Audio Formats

- Many different formats
  - usually are slight variants of each other
- Common ones are
  - MIDI
  - Wav
  - MP3

MIDI vs Wav Files

- MIDI and wav files are 2 completely different kinds of audio files

- MIDI operates with a MIDI table and MIDI data
  - MIDI table is a table that contains a map of audio samples from actual instrumental sounds
  - MIDI data files is data that describe the following
    - What notes to play, using which instrument, when to play it
    - Together with MIDI table, processed through wavetable synthesizer to produce an actual sound
    - Output usually sounded synthetic / unnatural
    - Wave table synthesizer is a device (e.g a computer chip on the sound card) that has small files of actual recordings of instruments stored in its ROM. These records are called upon by the sound card when a particular sound effect or note of a particular instrument needs to bbe played back
- Useful when space is limited
  - MIDI was popular in 1990s when HDD size =~ 40MB

MIDI vs Wav Files

- Wav files consist of computer data that contains actual audio recording
  - Usually very large (if uncompressed)
  - MP3 is an example of compressed wav audio data
  - Superior audio reproduction (versus MIDI)

- MIDI files nowadays are used to record / store the musical performance
  - MIDI data can be edited (e.g wrong notes can be fixed easily because it's just computer data)
  - MIDI file can then be played back and output through a very quality software synthesizer (soft synth) that is based on wavetable synthesis

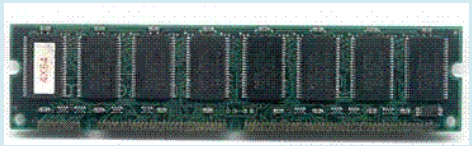
Audio in Games Today

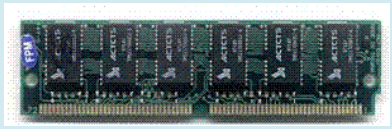
- Today's gaming commonly use a mix of mp3 (or .ogg format) & wav files
- Space is no longer a major concern
  - HDD is reaching in units of terabyte
  - Memory of greater than 4GB is becoming common
- Quality is more important than saving space & processing power

Memory

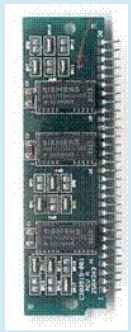
Memory (RAM)

- Data storage that allows for data to be accessed in any order and without physical movement
- Any piece of data can be fetched quickly and in a constant time



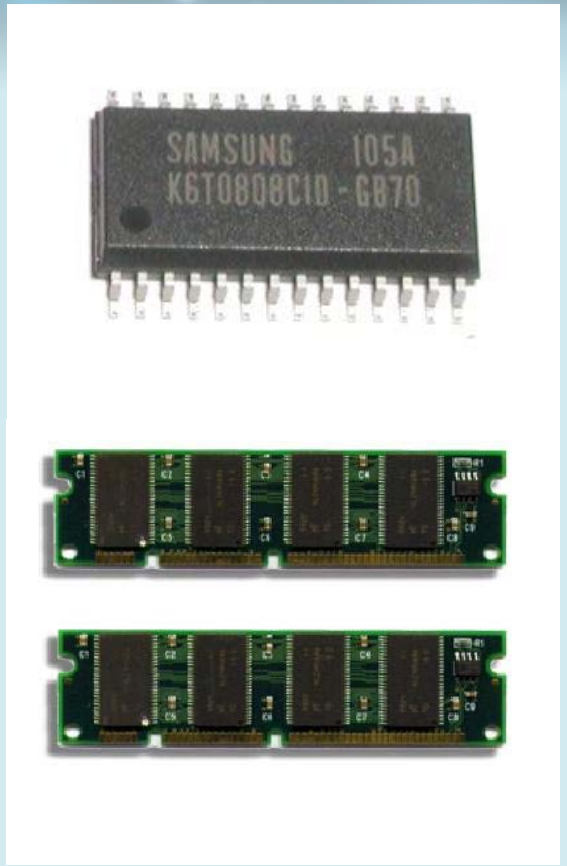


- Disadvantages:
  - High cost
  - Loss of data when power is turned off
  - Memory wall: CPU speeds have increased, but memory speeds have not been able to catch up
    - Memory latency



Memory (RAM)

- Comes in many “flavours”
- SRAM
  - Static Ram
  - Faster & consume less power (than DRAM)
  - More expensive
  - Commonly used for processor cache (L1/L2/L3 etc)
- DRAM
  - Dynamic RAM
  - Many revision & types (i.e. DDR, DDR2, DDR3, RDRAM, …)
  - Cheap to build in large volume
  - Main memory commonly built from these



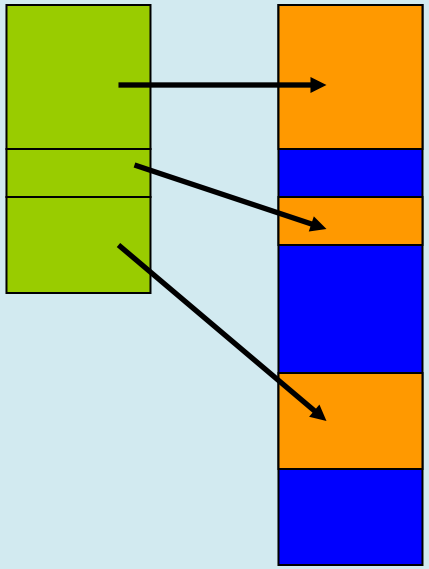
Cache Memory

- Concept of keep regularly accessed data (i.e. common instructions) in a temporary location
  - Location needs to be faster than main memory
  - Therefore Cache memory (L1, L2, L3)

- Why multiple levels?
  - Trade off speed (latency) vs volume / size
  - Multiple levels maximise cache hit (i.e. successfully locate the data in cache) and overall performance
  - L1 =~ 64-128kb
  - L2 = ~256 kb
  - L3 = ~2-8MB (up to 20MB in high end processors)

Virtual Memory

- Main memory space is frequently accessed
  - Read, write, erase
- Difficult to ensure an application always get a contiguous space allocated
- Virtual memory is a technique to make applications see contiguous space when actual allocation is fragmented



- That's all for today ☺