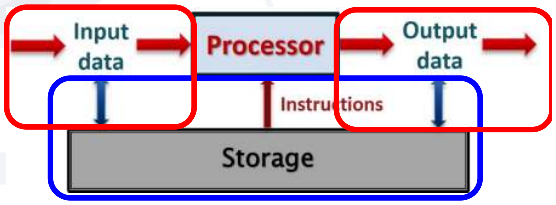
ENS1161 Computer Fundamentals

Module 7 Input/Output Devices and Interfacing

Moving forward..

- Last module:
  - Types of storage
  - How storage systems work
  - How data and instructions are moved from main memory to the processor and back
- Focus of this module: I/O systems and interfacing
  - Basics of I/O interfacing
  - Types of I/O devices and the operation



Module Objectives

On completion of this module, students should be able to:

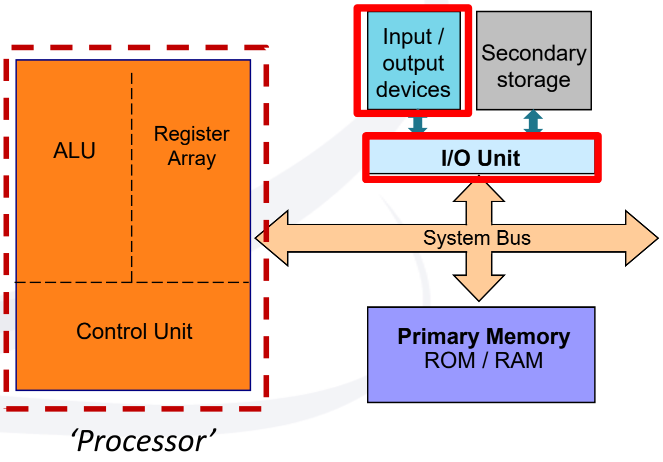
- Explain the basic principles of I/O interfacing and the importance of standards.
- Describe how serial and parallel communication is carried out and their relative advantages and disadvantages.
- List the various common I/O interfaces found in computer systems and explain how they are used.
- Describe the principles of operation of common I/O devices and related performance considerations.

Introduction

- **Module Scope**
  - What is I/O
  - The need for I/O interfacing
  - Common interface standards
  - Common I/O devices and corresponding interfaces

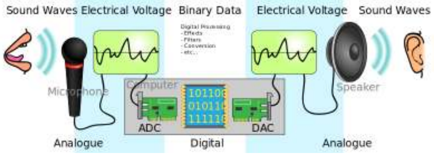
Basic Components of a Computer *(recap – Module 2)*

- Every computer contains the same basic components:
  - Arithmetic logic unit (ALU)
  - Register array
  - Control unit
  - Memory
  - Input/Output (I/O) unit
  - System Bus



I/O Devices

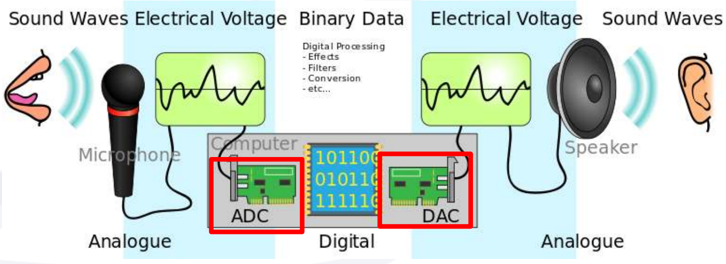
- There are a wide variety of input and output devices
  - Keyboard, mouse, microphone, scanner, display, printers, displays, speakers, etc.
- Most input devices translate a variety of analogue physical inputs to digital form for use by a processor
  - E.g. key presses, movements of mouse, voice commands, images
- Output devices translate binary data into a variety of physical output forms
  - E.g. text, images, sound, etc.
- Some devices are both input and output devices
  - transfer binary data to and from processor
  - E.g. secondary storage, network devices



Milesjpool, 2016
https://commons.wikimedia.org/wiki/File:CPT-Sound-ADC-DAC.svg

I/O signal format conversion (a)

- ‘Real-world’ analog (continuous) signals need to be converted to / from binary signals that computers understand
  - ADC – Analog to Digital Converter
  - DAC – Digital to Analog Converter
- Could be done in computer
  - Using interface cards as shown
- Could be done in the device itself
  - E.g. mouse movements converted to binary within mouse circuitry



Milesjpool, 2016
https://commons.wikimedia.org/wiki/File:CPT-Sound-ADC-DAC.svg

I/O Communication

- The processor must communicate with peripheral I/O devices
- Information transferred:
  - ***Data***
    - usually encoded in some suitable coding system
      - e.g. ASCII, PDL, scan codes, etc.
  - ***Control*** information
    - commands from the processor
    - requests for service from peripheral devices
    - control codes from the processor
    - status codes from I/O devices
      - *Covered in more detail in Module 8*

General purpose vs embedded systems *(recap Week 1)*

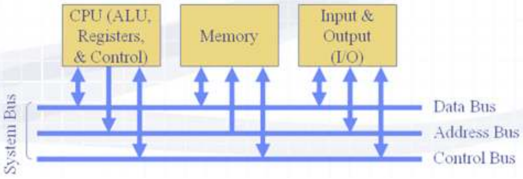
- **General purpose computers**
  - Designed to be used for a variety of applications
    - E.g. word processing, presentation, simulation, entertainment
  - Based on general purpose processor chips
  - 'Normal' computer
- **Embedded systems**
  - **Integrated into a larger device or system to perform a specific function**
  - **Generally use single chip integrated microcontroller devices**
  - Used to provide the 'smarts' in a range of appliances and products
    - E.g. microwaves, air-conditioners, TVs, cars, phones, routers
  - General purpose computers use these special purpose processors for subsystems
    - E.g. graphics coprocessor, DMA controller, keyboard controller
- Both have same general architecture and principles of operation

Buses *(recap – Module 2)*

- A bus is essentially a collection of conducting channels or 'wires'
- Each channel or line has can be at a 'high' voltage (1) or 'low' voltage (0)
  - So 1 line can 'carry' 1 bit of information
  - Multiple lines are needed to carry multiple bits (8, 16, 32, etc)

Parallel transmission

- Buses are used for communication between devices in a computer system
- The main processor system bus consists of
  - the address bus
  - the data bus
  - the control bus



I/O signal format conversion (b)

- Processors mostly operate with **parallel** digital data
- Many I/O devices transmit and receive data in **serial** format
  - Data bits sent down 1 line, 1 bit at a time
  - Used with devices that are further from the processor
- Advantages of serial communication
  - Serial communication is more reliable
    - Especially as distance increases
  - Serial communication is less costly
    - Less wires / transmission channels
    - Simpler circuits
    - Longer transmission lines

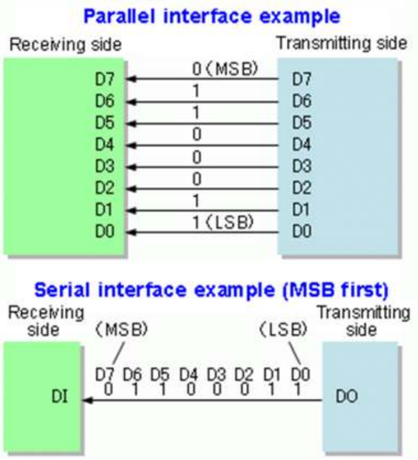
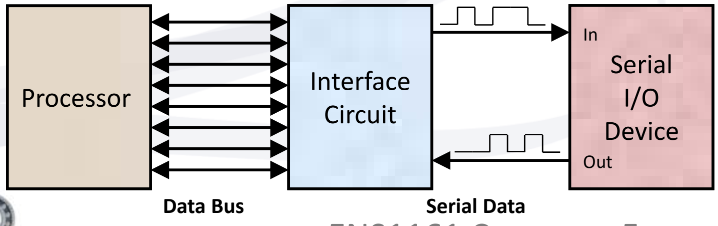


Image: SyamilAshri at English Wikibooks https://commons.wikimedia.org/wiki/File:Parallel_and_Serial_Transmission.gif

I/O signal format conversion (b)

- Need an interface circuit / device
  - To convert parallel to serial and vice versa
  - Also handles handshaking
    - Communications protocol to ensure data sent / received properly
  - Note: Interface circuitry normally includes some sort of controller
    - Simple(r) processor, normally designed for a specific purpose
    - Program stored as firmware



Parallel/Serial Interface – The UART

- Asynchronous serial transmission is extensively used
  - *Asynchronous* – without synchronised clocks
  - Data framed by Start and Stop bits for data flow control
    - Sometimes parity bit for error checking
  - Suitable for randomly timed small bursts of data
- Single chip UART or ***universal asynchronous receiver transmitter*** devices
  - Can perform serial/parallel conversions required
  - Also handles all handshaking with the I/O device

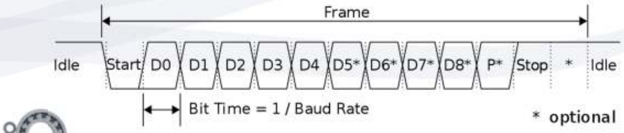
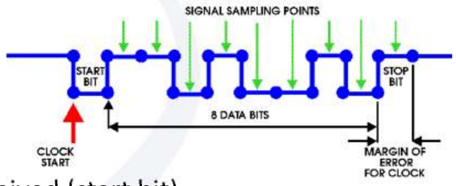


Image: *Public domain* https://commons.wikimedia.org/wiki/File:UART_Frame.svg

Data Communication terminology

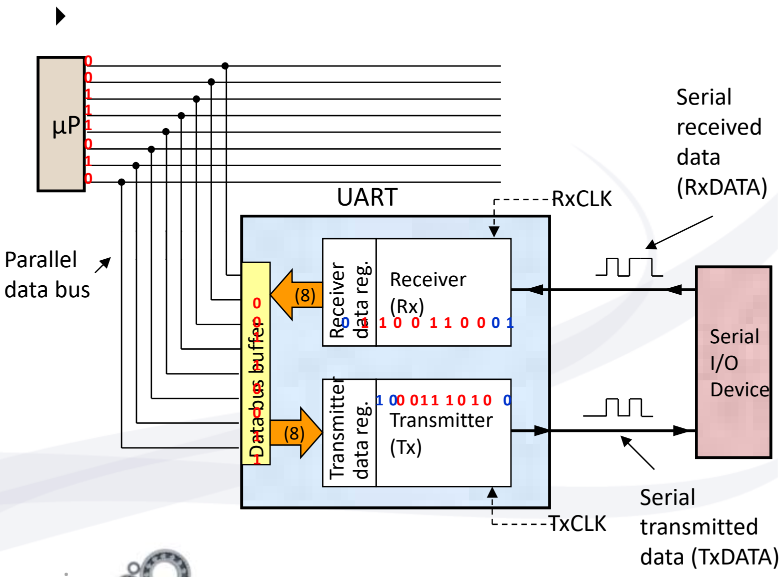
- Bit time
  - Time of 1 bit / time between sampling of serial signal
  - If not accurate, will lead to error reading signal
- Asynchronous transmission
  - No clock signal transmitted, timing worked out from data received (start bit)



people.seas.harvard.edu

- Synchronous transmission – 2 types
  - 1. A shared clock signal used
  - 2. Clocks on either end synchronised
- Parity bit
  - Optional extra bit (0 or 1) to check if any errors
  - Can either be Even parity – total number of 1s including parity is even - or Odd parity
- Frame: additional bits around actual data bits to create a total 'packet' of data

Parallel/Serial Interface – The UART


µP

Parallel data bus

UART

RxCLK

Serial received data (RxDATA)

Receiver data reg.

Receiver (Rx)

Data bus buffer

Transmitter data reg.

Transmitter (Tx)

TxCLK

Serial I/O Device

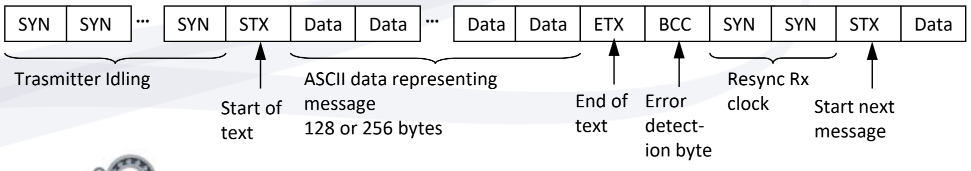
Serial transmitted data (TxDATA)

UART components

1. A serial receiver (Rx)
   - takes serial input and converts it to parallel format
   - stores it in the receiver data register (RxDR) ready for transmission to the processor
2. A serial transmitter (Tx)
   - takes a parallel data word from the transmitter data register (TxDR)
   - converts it to serial format for transmission
3. A bidirectional data bus buffer
   - passes parallel data from the processor to the TxDR, or from RxDR to the processor over the system data bus
4. Two externally applied clock inputs, RxCLK and TxCLK
   - Speed worked out (negotiated) via handshaking

Synchronous Serial Transmission

- Used for transmission of large blocks of data
- Uses synchronised clocks
  - Receiver synchronised to the transmitter through the use of special sync characters
    - sent before each block of data, or embedded in it and whenever the communications channel is idle
- More efficient, but more costly



SYN | SYN | ··· | SYN | STX | Data | Data | ··· | Data | Data | ETX | BCC | SYN | SYN | STX | Data

Trasmitter Idling

Start of text

ASCII data representing message
128 or 256 bytes

End of text

Error detect-ion byte

Resync Rx clock

Start next message

I/O Interfacing

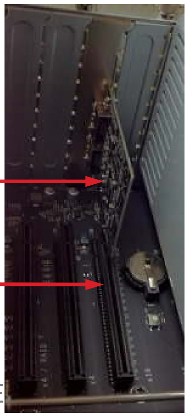
- *Interfacing*: connecting devices together so that they can share information
- An interface includes:
  - the physical connection (the hardware)
    - physical dimensions, pinning, voltages
      - *Broadly covered in this module*
  - a set of rules or procedures governing the transfer of information over that connection (the software/algorithm)
    - I/O modes, protocols, data format, software interface, etc.
      - *Broadly covered in this module and next*

Interfacing standards

- Many types of peripherals are interfaced with a computer
  - therefore many interfacing *standards* have been created
- A *standard*: an accepted way of interfacing
  - usually agreed to by industry associations
    - E.g. USB, HDMI, Wi-Fi, Bluetooth
- Why are standards required?
  - connecting devices built by different manufacturers
  - standards specify how devices should connect and communicate
    - Hardware
    - Software / algorithm
  - allows interoperability across multiple vendors

Motherboard and expansion buses

- The main circuit board that contains the processor is called the motherboard
- To connect various peripherals, a variety of interface circuits may be required
- For flexibility, most computer systems have some sort of expansion bus on the motherboard
- Interface cards may be connected to these expansion slots
  - physical interface where cards plug in
- Expansion buses have standards
  - Specify physical layout, data transfer speeds, signalling protocols, etc.

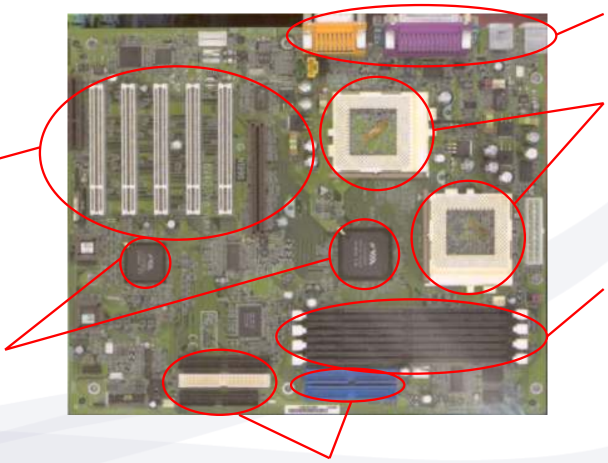


Interface card

Expansion bus slot

Image: Glenn Batuyong, 2011 https://www.flickr.com/photos/glennbatuyong/6114962032

Sample Legacy PC Motherboard



Peripheral Interface Ports

Expansion slots

Processor Sockets

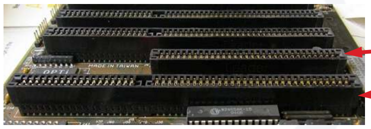
Memory Sockets

Support Chipsets

Internal Storage Device Interface

Example Bus Standards

- ISA - Industry Standard Architecture
  - original bus in IBM PCs (8-bit bus)
  - extended to 16-bit bus from PC/AT onwards
  - 8MHz, 16MBps throughput
- EISA - Extended ISA
  - 32-bit bus
  - 8MHz, 32 MBps throughput
  - *backward compatible* with ISA
- MCA - Micro Channel Architecture
  - 32-bit bus from IBM
  - not compatible with ISA
    - fatal mistake!



8-bit slot

16-bit slot

Image: Miha Ulanov https://commons.wikimedia.org/wiki/File:Isa1.jpg

Image: Appaloosa, 2005 https://commons.wikimedia.org/wiki/File:EISA_Bus.jpg

Image: K.Reichert, 2007 https://commons.wikimedia.org/wiki/File:MCASlot16.JPG

Bus Standards (cont.)

- PCI - Peripheral Component Interconnect
  - a local bus, high performance
  - 33/66 Mhz, 133 – 533 MBps
  - only accepts PCI cards, but can exist alongside ISA bus
- PCI Express (PCIe)
  - Updated and improved standard
  - Smaller footprint, higher throughput
  - 8 to 32 GBps
- PC Card Bus (PCMCIA)
  - for laptop, notebook computers
  - credit card sized slots
  - Being replaced by *ExpressCard*

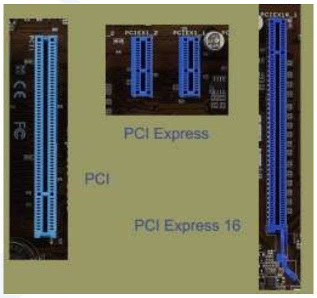


Image: Lucianolll, 2013
https://commons.wikimedia.org/wiki/File:Pci_Express_Slot.png

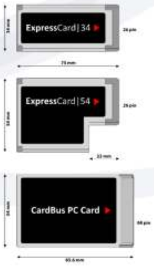
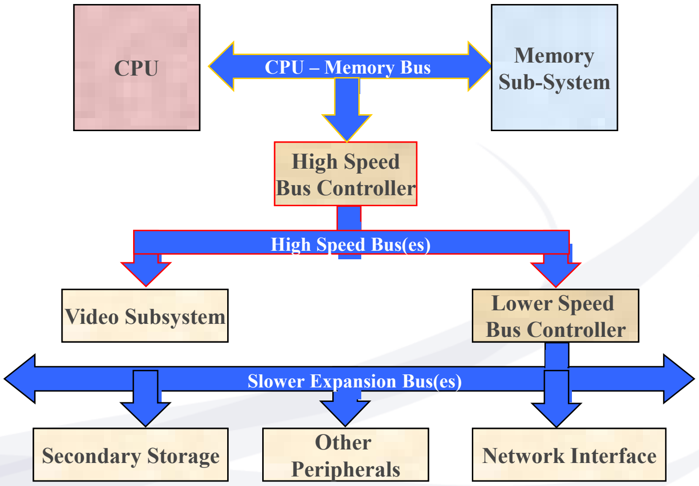


Image: Henrik Wannheden, 2006
https://commons.wikimedia.org/wiki/File:PCCard-ExpressCard_ZP.svg

Image: Jim.henderson, 2013
https://commons.wikimedia.org/wiki/File:3Com_modem_-_Ethernet_PCMCIA_in_IBM_9547.jpg

Example of multiple buses



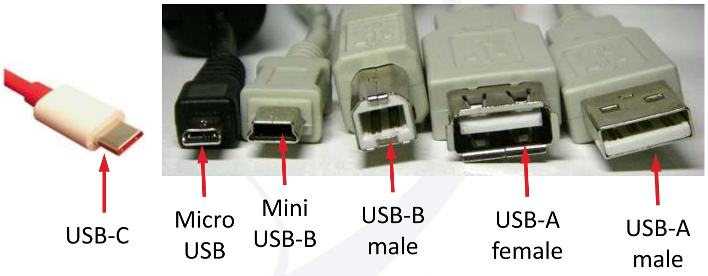
USB interface

- **U**niversal **S**erial **B**us
  - One of the most common interface standards today
- Used to connect a whole host of I/O devices
  - Keyboard, mouse, printers, flash drives, external hard drives, etc.
- Provides reliable, relatively high speed data transmission
- Can provide power to low power devices
- Easily expandable
  - Hubs provide additional ports
  - Hubs can be daisy-chained for further expansion

Image: Liftarn , 2007
https://commons.wikimedia.org/wiki/File:
USB_hub.jpg

USB Interface

- The standard evolves with time
- Physical connections:



- Data transfer rates:
  - USB 1.0 : 1.5 Mbps (Megabits per second)
  - USB 1.1 : 12 Mbps
  - USB 2.0 : 480 Mbps
  - USB 3.0 : 5 Gbps
  - USB 3.1 : 10 Gbps

Devices

- Shall now explore the principles of operation of common I/O devices
- Input devices
  - Keyboard, mouse
- Output Devices
  - Video display, printer
- Storage devices
  - HDD, SSD
- Communication / Network Devices
  - Network cards, modems, WiFi

Keyboard

- Pressing key triggers a switch
- Keyboard controller detects switch pressed and generates a *scan code*
  - Each key has a unique code
    - One code for key press (*make*)
    - Another code for key release (*break*)
  - 'Mapping' of keys to code stored in firmware

- Keyboard controller then signals computer that a key has been pressed
- Transfers the scan code to the computer serially
- Interface
  - Most common now: USB
  - Previously DIN or Mini DIN

DIN connector (male)

Image: Pavel Ševela, 2012 https://commons.wikimedia.org/wiki/File: Chicony_keyboard_connector.jpg

Mini DIN connector (female)

Image: D-Kuru/Wikimedia Commons, 2007

Mouse

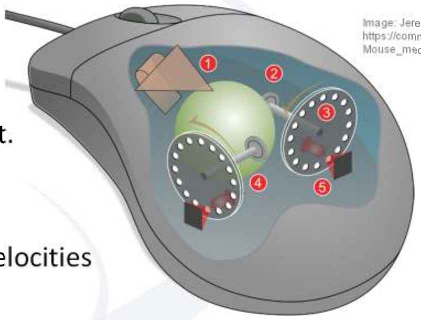
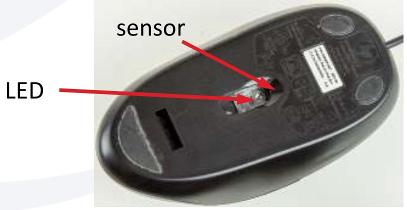


Image: Jeremy Kemp, Pbroks13, 2018 https://commons.wikimedia.org/wiki/File: Mouse_mechanism_diagram.svg

- *Mechanical mouse:*
  1. Pulling the mouse turns the ball.
  2. X and Y rollers grip the ball and transfer movement.
  3. Optical encoding disks include light holes.
  4. Infrared LEDs shine through the disks.
  5. Sensors gather light pulses to convert to X and Y velocities
- *Optical mouse:*
  - Optical sensor detects change in reflection of light from LED
- Buttons based on switches
- Interface
  - Most common now: USB
  - Previously: Mini DIN or RS-232 serial



Display

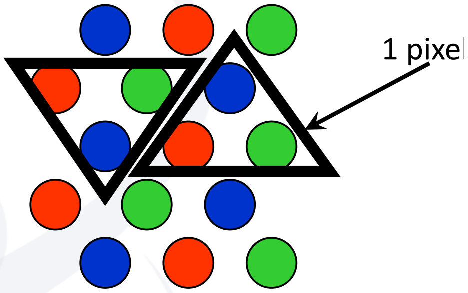
- Main output method / device for computer systems
- Computer displays have evolved over time
  - Older CRT (cathode ray tube) based monitors
    - Bulky and heavy
    - Really old ones only had one colour (monochrome)
  - Newer flat screen displays
    - Based on LCD or LED technology
    - Better resolution and capability
- Basic principle remains the same:
  - Control dots on a screen to create a pattern (image)
  - Dots have different intensity and colour

Image: Ceedrun, 2010 https://commons.wikimedia.org/wiki/File:Medion_CRT-Monitor_19%22_Silber-Metallic.jpg

Image: Boffy b, 2005 https://commons.wikimedia.org/wiki/File:IBM_PC_5150.jpg

Video System – terms and principles

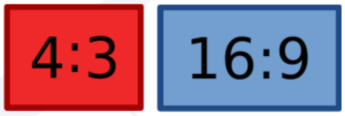
- **Pixel**
  - short for picture element
  - dots that make up picture on a monitor
  - Combination of red, green and blue (RGB) at different intensities results in different colours
    - E.g. all 3 on full = bright white, all off = black
  - Different technologies use different methods of creating the RGB dots



The red, green and blue dots are arranged in a triangle to make a single screen pixel

Video System (cont.)

- **Resolution**
  - number of dots displayed on a monitor
  - *number of dots across* x *number of rows*
    - e.g. 640 x 480, 1024 x 768, 1280 x 720 , 1920 x 1080
- **Aspect Ratio**
  - Ratio of width to height of display
    - Default used to be 4:3
    - Newer displays tend to be 16:9
      - HD standard
- **Screen size**
  - Based on diagonal measurement of screen



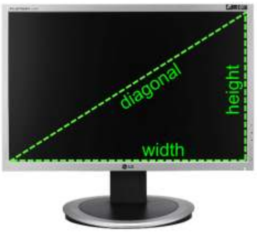
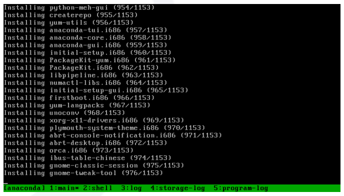


Image: Mikael Häggström, 2011 https://commons.wikimedia.org/wiki/File:Display_size_measurements.png

Video System (cont.)

- **VIDEO MODES**
  - **Text Mode**
    - for text only display (e.g. terminal mode)
    - screen size - no of characters displayed (e.g.80 x 25)
    - each character made up of pixels (VGA - 9 x16 pixels)
    - character set in ROM or software



```
Installing python-meh-gui (954/1153)
Installing createrepo (955/1153)
Installing yum-utils (956/1153)
Installing anaconda-tui.i686 (957/1153)
Installing anaconda-core.i686 (958/1153)
Installing anaconda-gui.i686 (959/1153)
Installing initial-setup.i686 (960/1153)
Installing PackageKit-yum.i686 (961/1153)
Installing PackageKit.i686 (962/1153)
Installing libpipeline.i686 (963/1153)
Installing numactl-libs.i686 (964/1153)
Installing initial-setup-gui.i686 (965/1153)
Installing firstboot.i686 (966/1153)
Installing yum-langpacks (967/1153)
Installing unoconv (968/1153)
Installing xorg-x11-drivers.i686 (969/1153)
Installing plymouth-system-theme.i686 (970/1153)
Installing abrt-console-notification.i686 (971/1153)
Installing abrt-desktop.i686 (972/1153)
Installing orca.i686 (973/1153)
Installing ibus-table-chinese (974/1153)
Installing gnome-classic-session (975/1153)
Installing gnome-tweak-tool (976/1153)

[anaconda] 1:main* 2:shell  3:log  4:storage-log  5:program-log
```

Image: Tom29739, 2015 https://commons.wikimedia.org/wiki/File:Anaconda_text_mode.png

  - **Graphics mode**
    - each pixel can be altered by the video adapter
    - each pixel uniquely addressable - can set each pixels attributes
    - mode for graphical user interfaces
      - E.g. Windows, MacOS
    - Different graphics modes have different resolutions, depth of colour, refresh rates

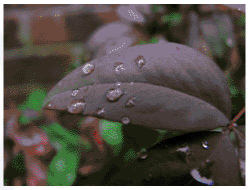
Video System (cont.)

- **Colour depth**
  - Colours made by combining red, green and blue dots at different intensities / ratios
  - No. of *bits per pixel (bpp)* used determines no. of different colours
  - *n* bits = 2^n different colours
  - more colours - finer gradation, more realistic pictures

No of bits	No. of Colours
4	16
8	256
16	64k (65,536)
24	16M (16 million) - *'True colour'*

  - New HD formats: 30 bit / 36 bits *('10-bit / 12-bit colour')*

4-bit



8-bit

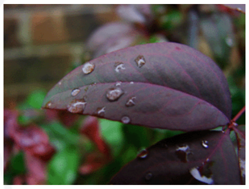
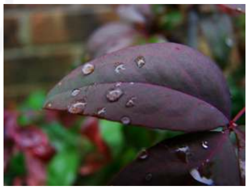


Image: Gnash, 20017 https://commons.wikimedia.org/wiki/File:8_bit.png

24-bit



Video System (cont.)

- CRT display : each pixel ‘displayed’ in turn
  - ‘Eye ‘persistence’ keeps image looking ‘whole’
- **Refresh Rate**
  - Number of times screen image is redrawn in a second
  - Higher rate - less flicker detectable - less eyestrain
    - Typically 50 – 72 Hz
- **Interlaced Display**
  - alternate lines redrawn in turn - odd then even
  - whole screen refreshed in two passes
- **Non-interlaced Display**
  - each line redrawn in turn - whole screen in one pass
  - better for static images

Interlaced display

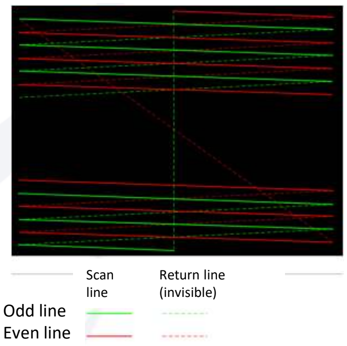
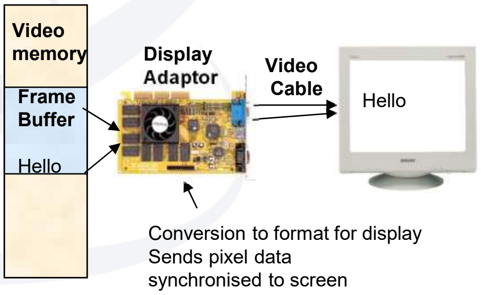


Image: Christophe Chan, 2007
https://commons.wikimedia.org/wiki/File:Balayage_entrelace_affichage_trames.svg

Video System (cont.)

- **Frame Buffer**
  - Memory that contains information to be written to screen
  - Contains a table of values for each pixel (*bitmap*)
  - Memory required for buffer = *no. of pixels on screen* x *no of bits for each pixel* (+ additional overhead)
- **Video adaptor**
  - Contains video memory for frame buffer
  - Provides interface for display unit
  - Many also have a *Graphics Processing Unit* (GPU)
    - Specialised controller
    - Takes over rendering of video information
    - Relieves main CPU, faster response
    - Crucial for video, gaming applications



Video memory

Frame Buffer

Hello

Display Adaptor

Video Cable

Hello

Conversion to format for display
Sends pixel data
synchronised to screen

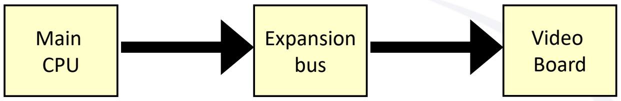
Video data throughput

- Trend is for increasing resolution screens / images
  - XGA : 1024 x 768 ≈ 786 thousand pixels
  - HD : 1920 x 1080 ≈ 2 million pixels
  - Quad HD : 2560 x 1440 ≈ 3.6 million pixels
  - Ultra HD (computer version of 4K) : 3840 x 2160 ≈ 8.2 million pixels!!
- Each pixel needs 3 bytes
  - For 24-bit colour depth
- Refresh rates
  - Minimum 60 Hz, could be as high as 120Hz
- Minimum video data throughput ≈ 140 Mbytes / sec
  - Could be 20 x or more higher

Addressing video throughput issues

- Separate video bus
  - Dedicated channel / bus to transfer video data
  - Does not interfere with / get blocked by other data transfers
- Graphics coprocessors / accelerators
  - Separate graphics processor to render images
    - Processor optimised for processing video type data
  - Main CPU only sends summary of changes to each screen
  - May include decompression of compressed video data
    - Compression algorithms ‘squash’ video data
      - E.g. MPEG-4, H.264
    - Decompressing them requires computation (processing)

Computer without graphics coprocessor

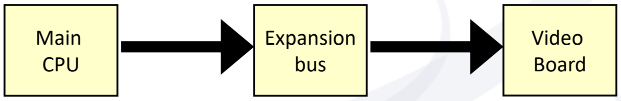


Main CPU: Calculates each pixel value. Performance bottleneck

Expansion bus: All display data details passed to video board. Performance bottleneck

Video Board: Displays information

Computer with graphics coprocessor



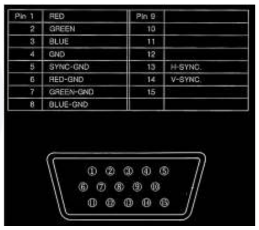
Main CPU: Works out general information about display. Free to do other work

Expansion bus: Transfers general info - far less than with conventional video board. Free to transfer other data

Video Board: Intelligent board calculates each pixel value and displays it, freeing main processor

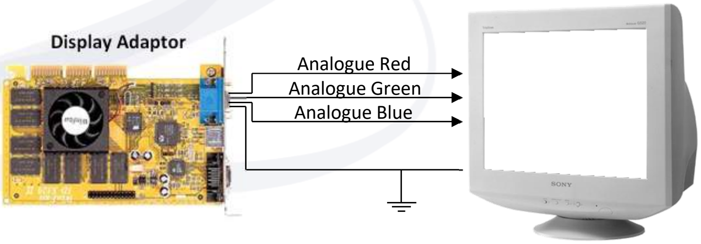
Video System (cont.)

- **VGA Interface**
  - **V**ideo **G**raphics **A**rray
  - Common interface for CRT-type displays
    - ‘Default’ from late 1980s
  - Each colour and other information converted to analog signals and sent to display via separate pins
  - VGA cards need 3 ADC converters
  - VGA standard has many variations
    - Standard VGA – 640 x 480 resolution
    - SVGA, XGA – 800 x 600, 1024 x 768
    - Etc.



Pin 1	RED	Pin 9	
2	GREEN	10	
3	BLUE	11	
4	GND	12	
5	SYNC-GND	13	H-SYNC.
6	RED-GND	14	V-SYNC.
7	GREEN-GND	15	
8	BLUE-GND		

Image: Daewoo, 2000
https://www.flickr.com/photos/easyplex/316055155/



Video System (cont.)

- **HDMI Interface**
  - **H**igh-**D**efinition **M**ultimedia **I**nterface
  - Interface for transferring both video and audio data in digital form
    - Does not require conversion to analog
  - Default interface for external screen in newer systems

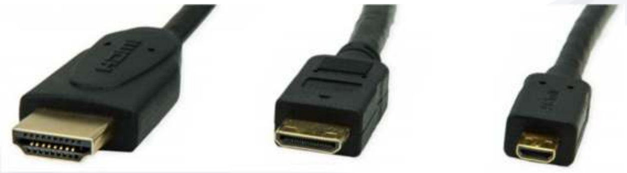


Image: SIMOBORTOLO, 2014
https://commons.wikimedia.org/wiki/File:HDMI-HDMImini-HDMImicro.png

Printers

- Printers have their own processor, firmware and memory
  - Some memory (*buffer space*) to store data from computer prior to printing
  - Data transferred in blocks
- Data is sent in some form of Page Description Language (PDL)
  - E.g. PCL, Postscript, PDF
- Physical interface:
  - Parallel printer interface
    - Obsolete
  - USB
  - Network connection
    - *Described in more detail in Module 10*

Image: Eike sauer, 2013
https://commons.wikimedia.org/wiki/File:Parallel_printer_cable_Centronics.jpg

Storage Interfaces

- **Parallel ATA (PATA)**
  - Or IDE (*Integrated Device Electronics*) / just ATA
  - Mainly for HDD or CD/DVD drives
  - Controller is integrated with drive
  - Cable transfers data in parallel (ribbon cable)
  - Can connect 2 drives in a daisy chain
    - 1 master, 1 slave device

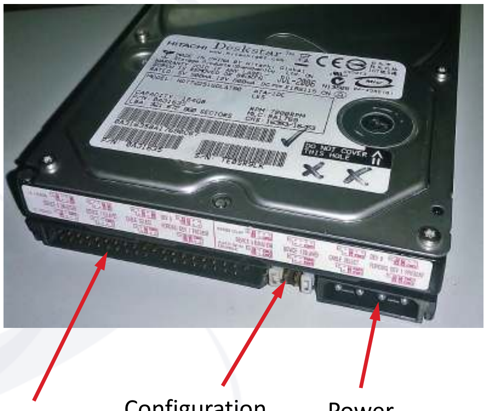


Image: JulianVilla26, 2016 https://commons.wikimedia.org/wiki/File:Hitachi_HDD_HDT722516DLAT80_IDE-ATA.jpg

ImageRainer Knäpper, Free Art License https://commons.wikimedia.org/wiki/File:IDE_cable_40_pin_%26_80_pin.jpg

Storage Interfaces

- **Serial ATA (SATA)**
  - Serial version of ATA
  - Used for HDD, optical drives and SSD
  - But higher data throughput
    - Faster signalling
  - Smaller cable
    - Less obstruction
  - *Hot plug* feature
    - Can connect / disconnect drive while computer running

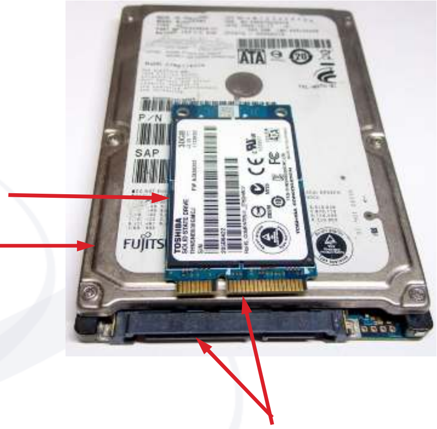


Image: Vladsinger, 2013 https://commons.wikimedia.org/wiki/File:MSATA_SSD_vs._2.5%22_SATA_drive.JPG

Storage Interfaces

- **SCSI**
  - Small Computer Systems Interface
  - Around from mid 80's
  - Can connect a variety of peripherals
    - Hard drives, optical drives, scanners
  - Can connect up to 8 devices
  - Original SCSI (parallel SCSI) mainly superseded
  - Now mainly used in business / enterprise computers
  - New standards:
    - Serial attached SCSI (SAS)
    - USB attached SCSI (UAS)

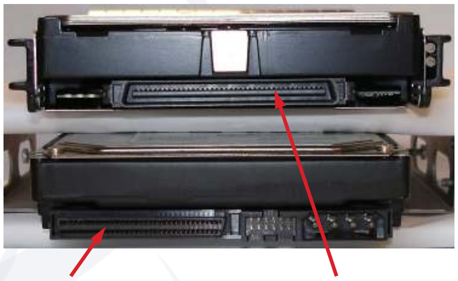


Image: Alistair1978,, 2006 https://commons.wikimedia.org/wiki/File:SCSIHardDriveConnectors.jpg

Image: Adamantios, 2009 https://commons.wikimedia.org/wiki/File:SFF-8484-internal-connector-0a.jpg

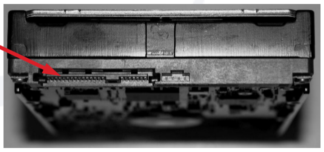


Image: Frettled, 2007 https://commons.wikimedia.org/wiki/File:SAS-drive.jpg

Network Interface

- Computers send and receive data to other computers via networks
  - *More details in Module 10*
  - Interfaces can be broadly divided into wired and wireless
- **Ethernet**
  - Most common interface for wired networks
  - Common name for the IEEE 802.3 protocol
    - Fast Ethernet: 100 Mbps (megabits per second) data transfer rate
    - Gigabit Ethernet: 1 Gbps
  - Requires an Ethernet adaptor / network interface card
  - Cable connection:
    - UTP (Unshielded Twisted Pair) – most common
    - Coaxial - not so commonly used
    - Fibre optic cable - for higher capacity connections

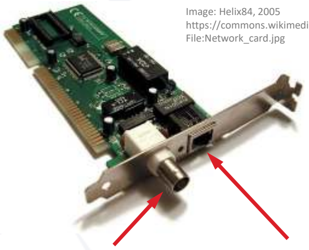


Image: Helix84, 2005 https://commons.wikimedia.org/wiki/File:Network_card.jpg

Image: Raimond Spekking, 2018 https://commons.wikimedia.org/wiki/File:BNC_connector_with_10BASE2_cable-92170.jpg

Image: Fo0bar, 2005 https://en.wikipedia.org/wiki/Category_5_cable#/media/File:Cat_5.jpg

Network Interface

- **Wi-Fi**
  - Wireless network protocol – IEEE 802.11
  - Computer needs a Wi-Fi adaptor (transceiver)
  - Wirelessly transmits data to an compatible access point
  - Uses radio waves to carry data
    - Typically 2.4 GHz or 5 GHz frequency
  - Various versions of protocol, now with newer simpler naming

Old name	*New name*
802.11n	*Wi-Fi 4*
802.11ac	*Wi-Fi 5*
802.11ax	*Wi-Fi 6*

Module Objectives

On completion of this module, students should be able to:

- Explain the basic principles of I/O interfacing and the importance of standards.
- Describe how serial and parallel communication is carried out and their relative advantages and disadvantages.
- List the various common I/O interfaces found in computer systems and explain how they are used.
- Describe the principles of operation of common I/O devices and related performance considerations.