ELECTRONICS STUDY GUIDE

Busses

questions he used on the other test:

1. A **Transducer** is a device that takes variations of physical quantity, such as sound, pressure, light etc. and converts them into an electrical signal.
2. The term **Bus** can be used when making reference to the means by which two or more digital devices are connected together and can communicate by a set of defined protocols.
3. With simplex communication, all data flow is **Unidirectional** from the designated transmitter to the receiver.

Definition - the means by which two or more digital devices are connected together so that data can be communicated between them. (Using a defined set of protocols)

- a "bus" encompasses:
  - the physical wiring,
  - the voltage levels,
  - their timing sequences,
  - the connector pinout specs,
  - and all other technical features.

when we talk about a bus we are talking about generally a communication standard

Short Distance Busses

Some examples of short-distance busses:

- PCI - a bus used in PCs - data transfer rate of 100 Mbytes/s (32 bit) and 200 Mbytes/s (64 bit)
- SCSI - bus used for PC disk drives

Extended Distance network

Some examples of an extended distance network:

- RS-232(C) - serial network connection used for connecting to printers or mice - 20kbps @ 50ft - up to 100kbps @ 6ft
- Ethernet (IEEE 802.3) - high-speed network which links computers together.

- "Normal" ethernet - 10 mil bits/s - (10BASE5 = thick coax) || (10BASE2 = thin coax) || (10BASE-T = twisted pair) || (10BASE-F = fiber)
- "Fast" ethernet - 100 mil bits/s (100BASE-TX = 2 pair twisted pair) || (100BASE-T4 = 4 pair twisted pair) || (100BASE-FX = fiber)

Communication Types

Simplex Communication

- **uni**-directional data flow - only flows in one direction ever
- transmitter -> receiver

Transmitter	---->	Receiver

Half-Duplex

- **bi**-directional data flow - either device can transmit/receive but only one at a time

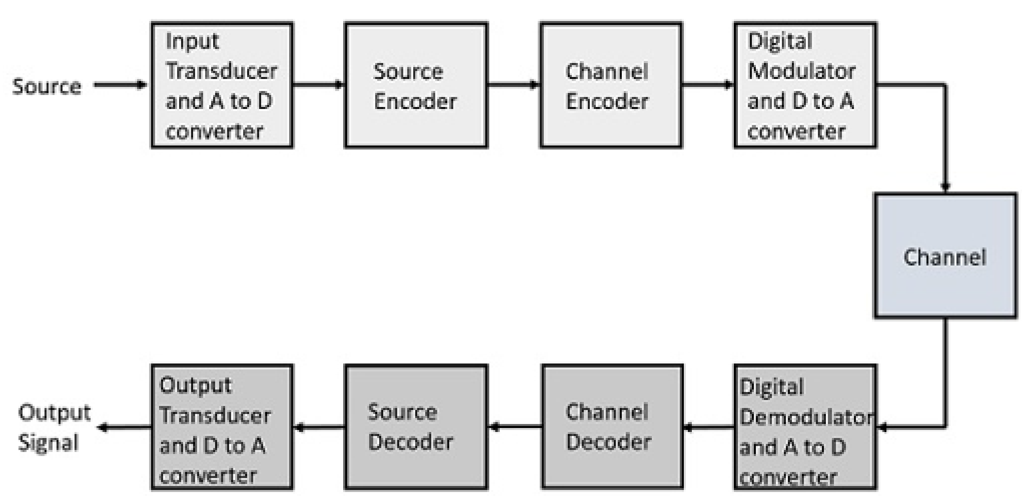
Receiver / Transmitter	< takes turns >	Receiver / Transmitter

Full-Duplex

- **Bi**-directional - flow of info is at the same time both ways
- Often through the use of two separate channels (a wire for each direction of communication flow)

Transmitter	----->	Receiver
Receiver	<-----	Transmitter

The whole digital communication system:



Transmitter Side

Source

- can be an analog signal - ex. sound signal

Input Transducer

- takes a physical input and converts it to an electrical signal - ex. microphone
- analog to digital converter - digital output

Source Encoder

- compresses data into minimum number of bits - removes redundant bits, helps with bandwidth utilization

Channel Encoder

- adds error correcting bits - to minimize signal alteration

Digital Modulator

- modulates signal to be carried
- converted to analog to travel over medium

Channel

- the channel / medium - allows the analog signal to transmit from the transmitter to receiver

Receiver Side

Digital Demodulator

- signal is demodulated
- signal is converted from analog to digital again

Channel Decoder

- does error correcting with the previous bits
- recovers signal back to original

Source Decoder

- signal is once again digitized via sampling and quantizing without loss of information
- recreates source output

Output Transducer

- converts signal back into original physical form (same as input of transmitter) - ex. speaker
- digital to analog converter - analog output

Output Signal

- the final output - ex. the sound signal received

USB

USB 1 - Low Speed

- 1.5 Mbit/s

USB 1.2 - Full Speed

- 12 Mbit/s

USB 2 - High Speed

- 480 Mbit/s

Architecture

- based on a "tiered star topology"
- single host controller - theoretically up to **127 "slave" devices**
  - most hubs would run out of power way earlier (127 because there are 7 bits, 0 is reserved)
  - hubs would have to be stacked to achieve this - max number of tiers is 6
  - length of any single cable limited to 5 meters

Host is the Master

- all communications on the bus are initiated by the host - no direct communication from device to device.
- a device cannot initiate transfer - only host can.
- the only exception is if the USB device has gone into a "suspended" state to save power - it can send a "remote wakeup" signal

Types of host controller:
OHCI (Open Host Controller Interface) - Low speed and Full speed - standard
UHCI (Universal Host Controller Interface) - Low speed and Full speed - requires a license from intel
EHCI (Extended Host Controller Interface) - only High Speed - passes to OHCI/UHCI for slower speeds.

USB Characteristics

contains 4 wires

- D+, D- (twisted pair for data, low speed may not be twisted) - pins 2 and 3
- GND - ground wire - pin 4
- VBUS - 5v power supply - pin 1

name	use	pin	colour
VBUS	5v power supply	1	red
D-	data	2	white
D+	data	3	green
GND	ground wire	4	black

Power Distribution

- a device (or hub) can only consume current from its upstream port
- a **"Self-Powered"** device is one that provides its own power and does not draw power from the bus
  - in "suspended" mode, the device must reduce its current to 0.5mA (it only needs this to be woken up)
- a **"Bus-Powered"** device is one that needs extra power, it can pull **100mA or 500mA** if permitted by host.
  - the device must reduce its current draw to 2.5mA whenever it is "suspended" - (0.3 mA is used by resistor)

Device Powering

- up to 5V
- voltage can drop to 4.35V
- no device can take more than 100mA **before it has been configured by host.**
- a device may draw up to 500mA if it configured as a higher power device
- a "self-powered" hub can provide 500mA

- devices that draw more than 500mA need to be self powered - don't use a Y splitter and pull from multiple ports.

Hot pluggable devices

- pulling a plug out while current is being drawn can cause an arc ("Flyback") and cause damage / data loss
- to be hot pluggable, devices need to follow rules:
  - minimum of 1 micro F capacitor on VBUS and GND
  - maximum of 10 micro F capacitor on VBUS and GND

USB Data flow

- only one host or device can be transmitting at a time
- works like a hub - blasts data to everything but only gets accepted by the destination

Transceiver

- at the end of the data link between host and device is a transceiver circuit
- typical upstream end transceiver is located nearer to the host. Upstream end has two 15K pull-down resistors.
- the equivalent downstream end transceivers are found in end devices
- individual receivers on each line are able to detect single ended signals (single ended zero - SE0)

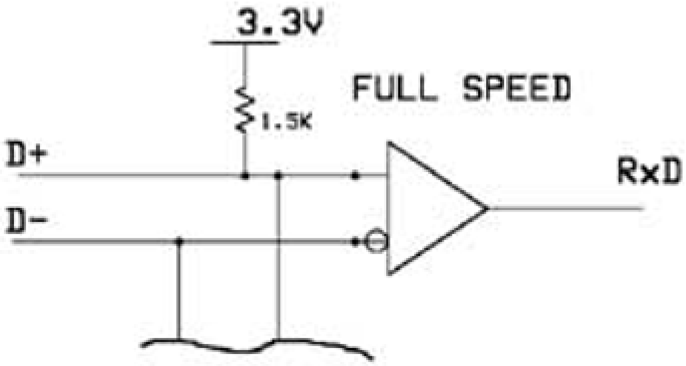
Speed Identification

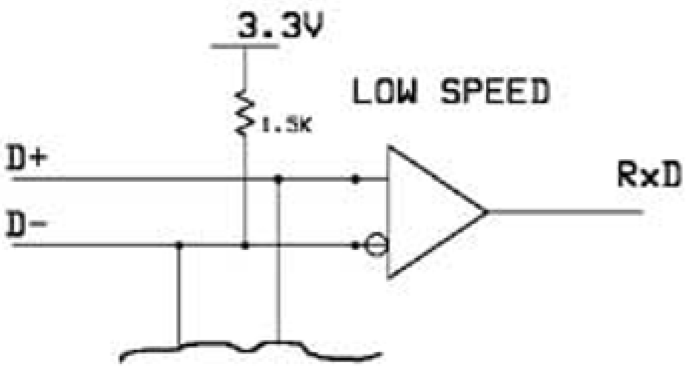
- The device end of the link, a 1.5K ohm resistor pulls one of the lines up to a 3.3V supply from VBUS
- **D- for low speed devices**, **D+ for a full speed device**
- high speed device will initially present itself as a full speed device with the pull-up resistor on D+

Important to remember:

- Version 1 - low speed - 1.5K resistor connected to **D-**
- Version 1.2 - full speed - 1.5K resistor connected to **D+**

- Version 2 - high speed - 45 ohm resistor on **each end of both D+ and D-**





- **A COMBINATION OF THOSE TWO IS HIGH SPEED**

Line States

- Detached - no device plugged in - both data lines low
- Attached - device plugged in - either D+ or D- go to level 1
- Idle - pulled up line is high and other is low - i dont know man
- J-State - same polarity as the idle state - being driven to that state by either host or device
- K-state is just the opposite polarity to the J-state

RS232

- also known as a serial connection - used a lot for like computer connections (think serial to x adapter)
- started at 25 pins, now has 9 pins

Mark - logical 1

- a mark is a low voltage state for a logical 1 (binary 1)
- ranges from -3v to -15v

Space - logical 0

- a space is a high voltage state for a logical 0 (binary 0)
- ranges from +3v to +15v

ETHERNET

10Base-T - 10Mbit/s

- IEEE 802.3i
- uses Manchester encoding signal

LLC - (Logical Link Control) sublayer - L3?
MAC - (Medium Access Control) sublayer - L2

- **MII** - Media Independent Interface
- **PLS** - Physical Layer Signaling
- **AUI** - Attachment Unit Interface
- **MAU** - Media Attachment Unit
    - **PMA** - Physical Medium Attachment
- **MDI** - Medium Dependent Interface

^ top down order of interactions for physical layer (???)

100Base-T - 100Mbit/s

- IEEE 802.3u
- uses a combo encoding method
    - NRZ-I (PMA layer)
    - 4B/5B (PCS layer)
    - MLT3 and Scrambling (PMD Layer)
- MII = Media Independent Interface
- PHY = Physical Layer Device
    - PCS = Physical Coding Sublayer
    - PMA = Physical Medium Attachment
    - PMD = Physical Medium Dependent
    - AUTONEG = Auto Negotiation
- MDI = Medium Dependent Interface

^ another classic top down order of interactions for physical layer (>???!??!?!??)

Media Independent Interface (MII)

- used for connecting different types of PHYs devices to MACs
- being media independent means that different types of PHY devices for connecting to different media (twisted pair, fiber, etc.) can be used without replace the MAC hardware

Media Dependent Interface (MDI)

- the interface in a computer network from a physical layer implementation to the physical medium used to carry the transmission

Physical Coding Sublayer (PCS)

- networking protocol sublayer in Fast Ethernet, Gigabit Ethernet, and 10 Gigabit Ethernet standards
- Provides an interface between the PMA sublayer and the MII

Physical Medium Attachment (PMA)

- comprised of one PMA Reset function and five simultaneous and asynchronous operating functions
  - PHY Control
  - PMA Transmit
  - PMA Receive
  - Link Monitor
  - and Clock Recovery

Physical Medium Dependent sublayers (PMD)

- further helps to define the physical layer of computer network protocols
- define the details of transmission and reception of individual bits on a physical medium
- responsible for:
  - bit timing
  - signal encoding
  - interacting with the physical medium
  - and the properties of the cable - fiber, or copper

Physical Layer (PHY/PLS)

- responsible for sending computer bits from one device to another along the network
- doesn't understand the bits, rather its role is determining how physical connections to the network are set up
- covers a variety of devices and mediums
  - cabling

  - connectors
  - receivers
  - transceivers
  - and repeaters

Auto Negotiation - know this one

- feature that allows two devices to determine the optimal duplex mode / speed for the connection

Attachment Unit Interface (AUI)

- specifies how a cable is going to connect to an Ethernet Card
- AUIO specifies a coaxial cable connected to a transceiver that plugs into a 15 pin socket on the NIC

Medium Attachment Unit (MAU)

- transceiver that converts signals on an ethernet cable to and from AUI signals

in most modern networks - neither MAU nor AUI exist.

General overview

- PHYs work in layer 1
- Ethernet MACs are placed in Layer 2
- 10 Mbit/s - **Manchester**
- 100Mbit/s
  - **4B/5B** - 5 bits - guarantees at least two signal edges per 5 bit word (??)
  - **NRZ-I**, - three values | -1, 0, 1
  - **MLT-3** - half the NRZ-I transfer frequency from 62.5 MHz to 31.25 MHz
- 1000Mbit/s - (**4D**?)-**PAM 5**

10Mbit/s - Half-Duplex
100Mbit/s - Full-Duplex
1000Mbit/s - Full-Duplex with all 4 pairs

1000Mbit/s - needs auto-negotiation to establish the master-slave signal timing control that makes the link work

- does this through "Fast Link Pulse" signals - used to verify link integrity

10BASE-T and 100BASE-T

Pin	function	colour
1	+TD	white and green
2	-TD	green
3	+RD	white and orange
4	not used	blue
5	not used	white with blue
6	-RD	orange
7	not used	with with brown
8	not used	brown
**PAIRS ARE: 1:2	3:6** (2 pairs)	

TD means transmit
RD means receive

1000BASE-T

Pin	function	colour
1	+Bi_DA	white and green
2	-Bi_DA	green
3	+Bi_DB	white and orange
4	+Bi_DC	blue
5	-Bi_DC	white with blue
6	-Bi_DB	orange
7	+Bi_DD	with with brown
8	-Bi_DD	brown
**PAIRS ARE: 1:2	3:6	4:5

Bi means bidirectional
D(A,B,C,D) means Data (A,B,C,D)

How is 1000BASE-T gigabit

- 10BASE-T and 100BASE-T both use a dedicated pair for transmitting and a dedicated pair for receiving data
- each of the wires use 125MHz frequency and sends 1 bit over a pair

1000BASE-T uses all 4 pairs for both transmitting and receiving data

- each wire still uses 125MHz frequency, but since they are both transmitting and receiving, they send 2 bits per pair.
- **125MHz X 2 bits per pair X 4 pairs = 1000Mbit/s**

Difference between PCI and PCI-e

PCI

- Runs at a transfer rate of 133MB/s
- **SHARED BUS** - that means if there were 4 PCI slots - the bandwidth would be cut each connection it has

PCI-e

- runs at a transfer rate of 250MB/s
- point-to-point connection - doesn't share the bandwidth

Why PCI-e over PCI

- gigabit uses 125MB/s, so if you had 2 NICs in PCI you would not be able to get max speeds for either bus
- PCI-e you would be able to