**CSCI 463 Final Exam Study Guide**

**Signal Properties**

This data encoding technique uses one voltage level to represent 1 and another to represent 0. This protocol has NO neutral level or non-signaling level.

NRZ (non-return to zero)

This encoding technique uses a transition in the signal level at the clock time boundary to signal one bit value and a lack of transition to signal the other. Which bit value is represented by a change is specific to a particular implementation.

NRZI (non-return to zero inverted)

This encoding technique uses two different frequencies to represent either zero or one.

FSK (frequency-shift keying)

This encoding technique uses an abrupt shift in the carrier’s wave form to represent a one.

PSK (phase-shift keying)

This encoding technique imposes the data on a clock signal. A one is transmitted when the carrier transition is low to high or stays high and a zero when transition is high to low or stays low. There is always a transition change in the middle of the bit transfer to provide clocking. Some bus protocols reverse the low and high assignment.

Manchester encoding

This encoding technique imposes the data on a clock signal. If the next bit is a 1, there is no transition in signal level at the end of the current bit transfer. If the next bit is 0, there is a transition in the signal level at the end of the current bit transfer. There is always a transition in the middle of the bit transfer. Some bus protocols reverse the low and high assignment.

Differential Manchester encoding

This encoding technique encodes a series of sequential bits, usually a byte, in a predefined longer sequence of bits, like a symbol, so that the number of sequential 0s or 1s is limited.

RLL (run length limited)

This technology uses a pair of lines to transmit signals. The signal is applied to one line and its complement to the other. The paired signals are then compared on the receiving end. This technique provides great protection against external signal corruption. Note the actual data may use level based or change based encoding.

Differential signaling

This technology uses a single line or trace to transmit data. It depends on a separate ground line or ground plane to return the signal’s energy to sender.

Single-ended signaling

This technology depends on a transmitted signal to bounce back from the end of the line or trace to amplify the signal. Signal is terminated at the source. This technique limits the length of the bust.

Reflected-wave signaling

This encoding technique uses two different signal levels to represent zero or one. Additionally, it has an intermediate level that signifies no data transmission.

RZ (return to zero)

**Bus Properties**

BUS uses packet based communication.

Hypertransport

PCIe

USB 1.1, 2, 3

SATA

BUS uses NRZ signaling.

GPIB

ISA

PCI

PCIe

SATA

BUS uses NRZI signaling.

USB

BUS supports burst mode transmission.

GPIB

PCI

PCIe

USB 2, 3

SATA

BUS uses RLL encoding.

PCIe

SATA

Bus allows or provides for bus mastering.

ISA

GPIB

PCI

PCIe

Bus interfaced logic can signal interrupt request an any time.

USB

GPIB

BUS multiplexes address, data, and control over same lines.

USB

PCI

BUS uses differential signaling for its data, address, and control.

PCIe

USB

SATA

Bus uses reflected wave signaling on its data lines.

PCI (according to his notes), SATA (according to chegg)

Bus uses single end point signaling for its data lines.

ISA

BUS uses bit stuffing.

USB 1.1, 2

BUS uses scrambling which helps with error detection and reduced cross-talk.

Hypertransport

Bus uses single serial bidirectional data link.

USB 1.1, 2

SATA

Bus has separate pairs of unidirectional serial data connections(paths), but only one of each.

USB 3

PCIe

Hypertransport

Bus uses serial communication but supports multiple serial lanes, but transmits individual bytes on individual paths or lanes.

USB

SATA

Bus uses variable width bus, individual bits of bytes distributed across available lines or lanes.

Hypertransport

BUS switches from half-duplex link to separate full duplex pair once both ends recognize its supported.

USB 3

Bus has a dedicated clock line or signal.

ISA, PCI, PCIe, GPIB

**Bus Basics**

4 types of bus

* Address - used to identify device/circuit being contacted.
* Data - used to transfer data between devices/circuits.
* Control - used trigger actions or transmit status of activity.
* Power - used to power circuits/devices.

Arbitration - used when more than one device can master the bus.  
4 types

Daisy-chain

* + Grant line linked to masters in serial chain. If a master claims the grant, it doesn't pass the grant on to then next master.
  + Lesser masters get bus only if higher master not interested.
  + Not fair, but efficient for certain bus designs.

Centralized arbitration

* + Uses a dedicated arbiter.
  + Can allow for 'programmable' prioritization.
  + Masters need only limited additional circuitry (request/grant)
  + WiFi uses this, clients wait for a clear channel and requests a reseved time slot to use WiFi. Access point grants or denies(ignores) request. If granted, all other devices see the grant and wait.
  + PCI arbitration is performed by the PCI controller.

Distributed arbitration

* + Each master needs additional logic (decoder logic) to break ties.
  + Each master assigned an ID to establish priority.
  + All masters share arbitration lines.
  + Some versions all priority to be reassigned dynamically.
  + Closely resembles daisy chain, but uses additional logic on each device.
  + Examples : SCSI

Distributed using collision detection.

* + Any device may use 'bus'.
  + Device transmits only on a clear bus
    - As it transmits, it continues to monitor for collision.
    - If detected, transmission pause for short time.
    - Re-transmission attempted again.
    - If collisions continue, transmission abandoned.
  + Effective if number of 'masters' not too large.
  + Early ethernet used CSMA/CD.

**Error Checking**

Parity Bit

* Even or odd.
* Parity bit sent so total 1 bits counted matches parity.
* Detects single bit error
* Cannot identify specific bad bit
* Used in early PCs as a check for main memory.

CRC - cyclic redundancy check

* Effective for quick detection of data corruption when transferring data.
* Uses polynomial math to calculate a short check value.
* Added to data stream. Recalculation results null out on good data transfer.
* Cannot recover data, just detects corruption.
* Useful where re-transmission cheap.
* Used in early hard drive storage.

Hamming code

* Provide error detection and recovery where error rate low.
* Better protection if errors are not close together.
* Used on main memory on high-end servers where accurate data is critical.

Reed-Solomon

* Applied to blocks of data
* Provides check and correction for multiple errors in block.
* Used where re-transmission of bad data not practical.
* Variations used in
  + DVD
  + CD
  + QR code
  + DSL
  + WiMax
  + Digital TV
  + deep space satellite

Turbo-code

* Highly efficient technique for checking and repairing errors.
* Effective where corrupting noise is present.
* Uses a type of cross check on receiving for best-guess fix.
* Used in 3G/4G mobile and newer deep-space satellites.

Low-density parity check code LDPC

* Highly efficient technique for checking and repairing errors.
* Effective where corrupting noise is present.
* Conceived in 60's but implementation only possible with current technology.
* Used in newer satellite communications.
* More efficient than turbo-code for very high speed transmission

False: GPIB interface bus uses packet based communication.

True: Legacy system bus, ISA/EISA, uses packet based communication.

False: PCI expansion bus uses packet based communication.

True: PCI-e expansion bus uses packet based communication.

True: USB2 bus uses packet based communication.

True: GPIB, general purpose interface bus uses NRZ signaling.

True: Legacy system bus, ISA/EISA, uses NRZ signaling.

True: PCI expansion bus uses uses NRZ signaling.

False: PCI-e expansion bus uses NRZ signaling.

False: USB2 bus uses NRZ signaling.

False: GPIB general purpose interface bus supports burst mode transactions.

False: Legacy system bus, ISA/EISA, supports burst mode transactions.

True: PCI expansion bus supports burst mode transactions.

True: PCI-e expansion bus supports burst mode transactions.

True: USB2 bus supports burst mode transactions.

False: GPIB general purpose interface bus uses RLL encoding.

False: Legacy system bus, ISA/EISA, uses RLL encoding.

False: PCI expansion bus uses RLL encoding.

True: PCI-e expansion bus uses RLL encoding.

True: USB2 bus uses RLL encoding.

True: GPIB general purpose interface bus supports bus mastering.

True: Legacy system bus, ISA/EISA, supports bus mastering.

True: PCI expansion bus supports bus mastering.

True: PCI-e expansion bus supports bus mastering.

False: USB2 bus supports bus mastering.

True: GPIB general purpose interface bus supports polled interrupts.

False: Legacy system bus, ISA/EISA, supports polled interrupts.

False: PCI expansion bus supports polled interrupts.

True: PCI-e expansion bus supports polled interrupts.

True: USB bus supports polled interrupts.

True: GPIB general purpose interface bus multiplexes both address and data over same lines.

False: Legacy system bus, ISA/EISA, multiplexes both address and data over same lines.

True: PCI expansion bus multiplexes both address and data over same lines.

True: PCI-e expansion bus multiplexes both address and data over same lines.

True: USB bus multiplexes both address and data over same lines.