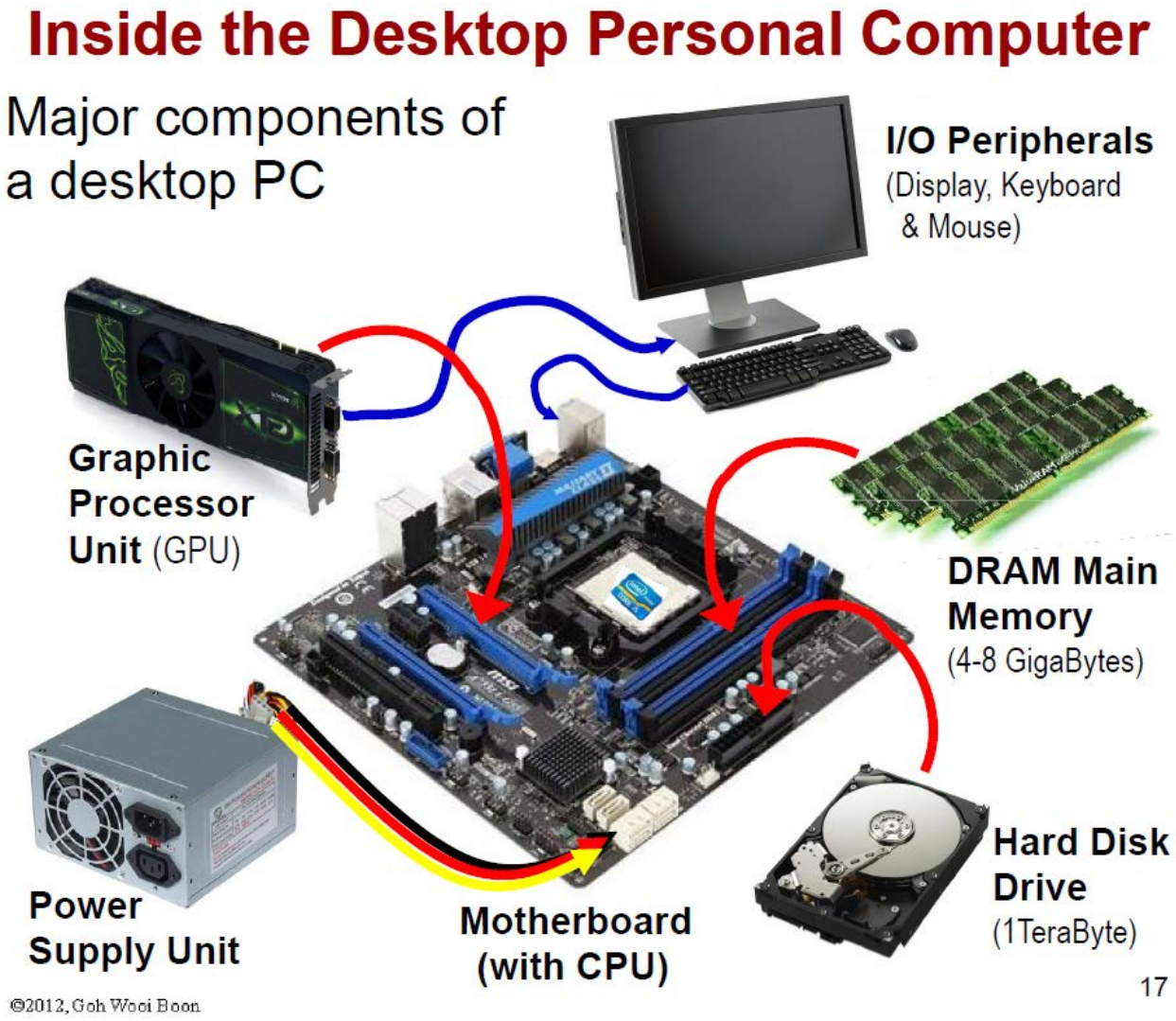
CE1006/CZ1006
Computer Organisation and Architecture

Data Transfer and I/O Interfaces
Introduction

Oh Hong Lye
Lecturer
SCSE, Nanyang Technological University.

Computer System



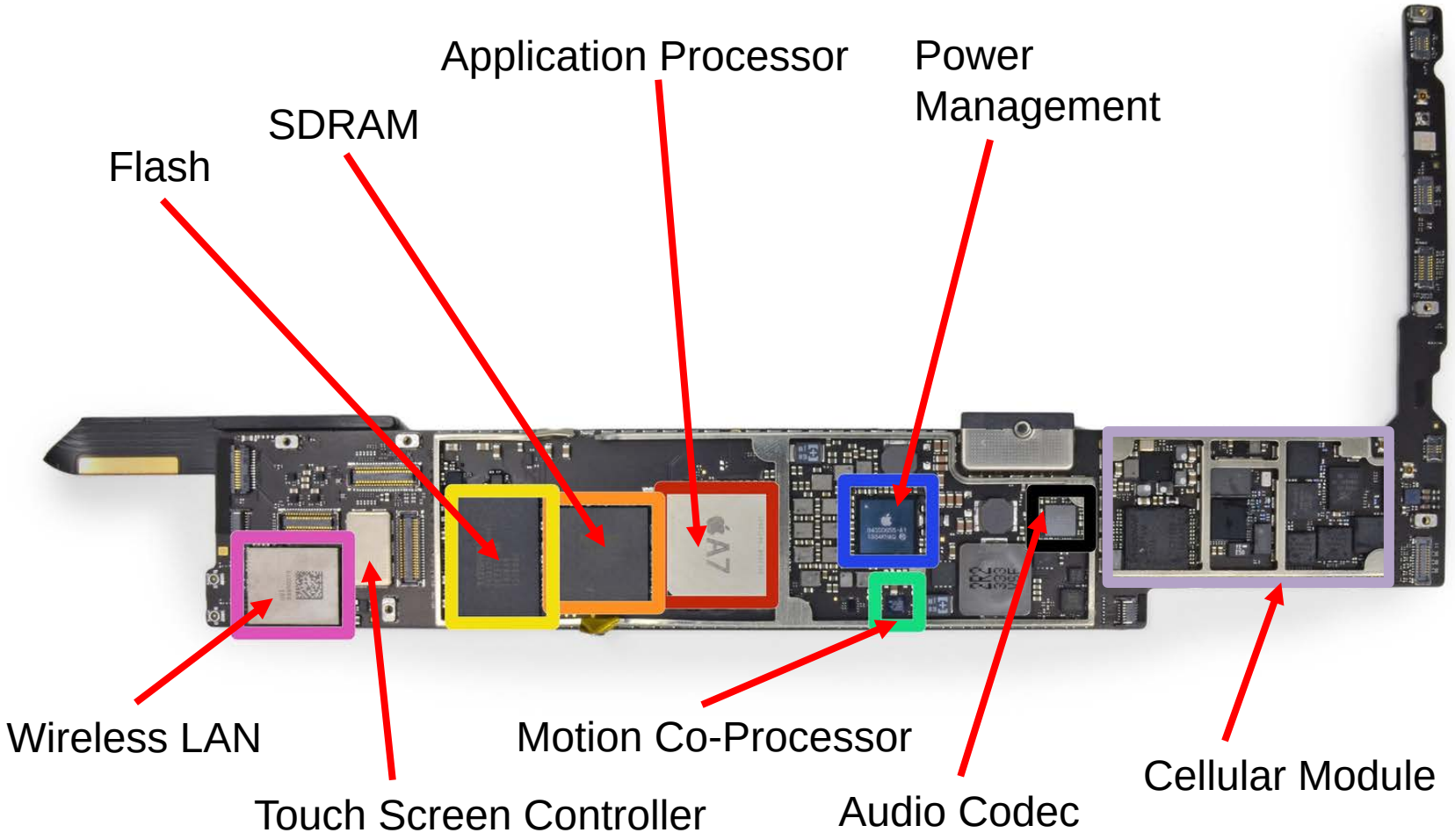
Tablet/Smartphone Teardown

Source: https://www.ifixit.com/Teardown/iPad+Air+LTE+Teardown/18907

Check out the link below for an animated teardown of iPhone

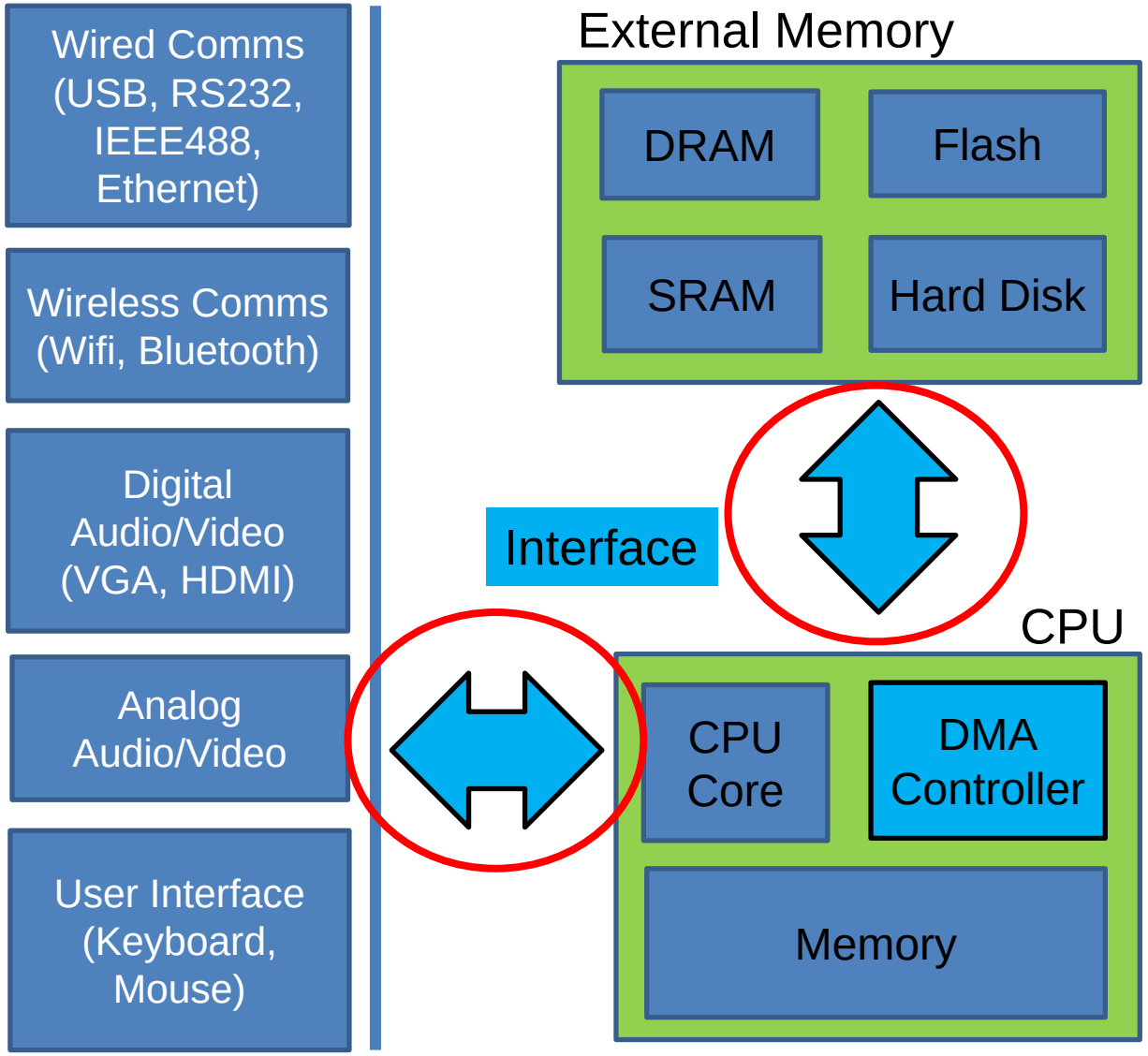
http://www.designnews.com/document.asp?doc_id=236187&dfpPParams=ind_184,industry_consumer,aid_236187&dfpLayout=article

iPad Air Main PCB



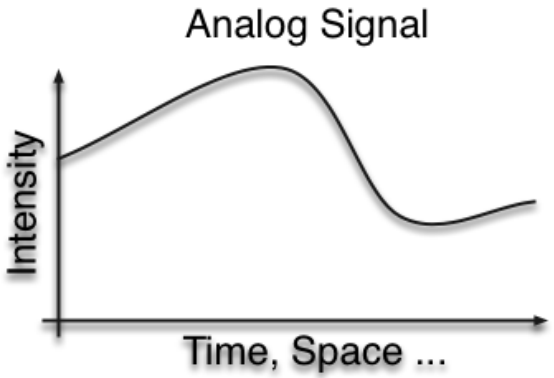
Source: https://www.ifixit.com/Teardown/iPad+Air+LTE+Teardown/18907

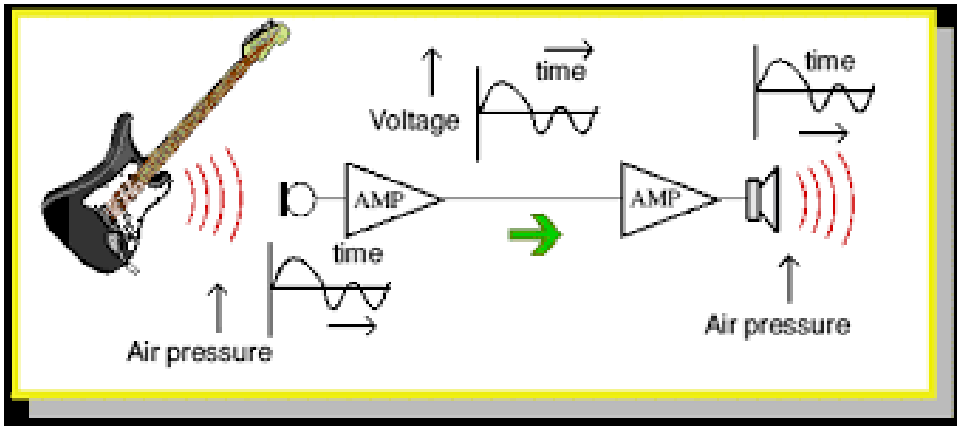
Computer Interfaces



- Data Transfer and IO Interface
- Polling, Interrupts and DMA Mechanism
- Computer Memory
- Computer Arithmetic
- Performance Measurement Analysis
- Parallelism

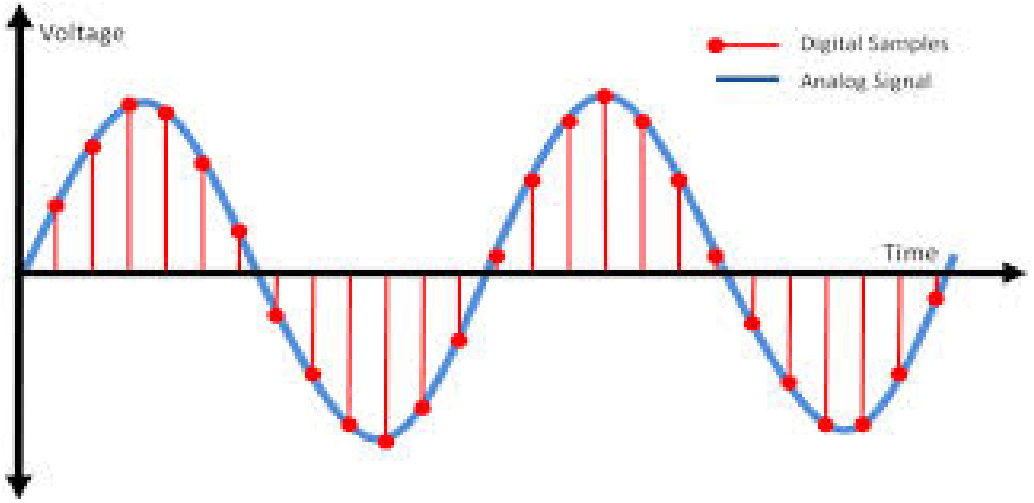
Analog and Digital Signal





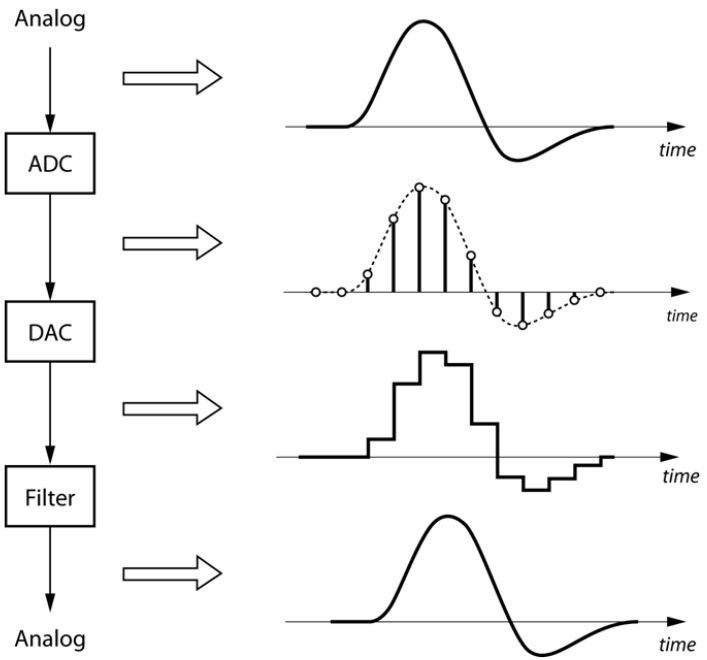
- Real world signals are Analog in nature.
  - Examples: Sound, light, heat, pressure etc
  - In the past, most processing are done in analog domain.
  - With the introduction digital processors and decreasing cost to build them, digital processing became increasingly popular as it offer more flexibility in implementation and are more tolerant to noise and component aging.
  - Analog signal has continuous voltage level and are continuous in time domain.

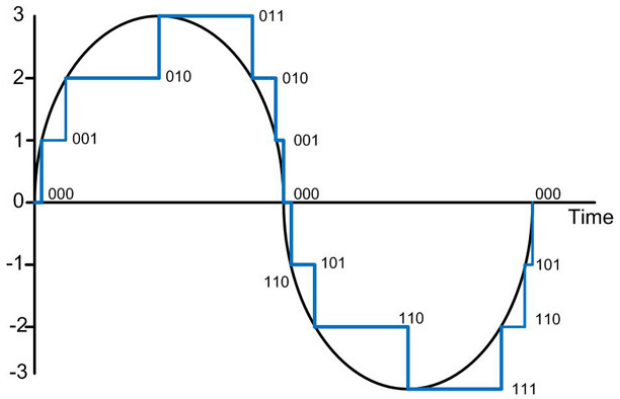
Digital Signal



- Digital signals are discrete time representation of analog signal.
- Obtained through process of digitisation, commonly know as Analog to Digital Conversion.
- Analog signals are digitised to its digital equivalent using a Analog-to-Digital Converter (ADC).
- Digital signal has discrete voltage levels.
- Similarly, analog signal can be reconstructed from digital signal via Digital to Analog Conversion.

Transformation between Digital and Analog Domain





- Figure on the left shows conversion between analog and digital domain.
- Digital signal is obtained by sampling the analog signal level at discrete time, known as the sampling interval. Typically, the analog signal level assigned to the nearest discrete voltage level allowed in that particular digitisation process.
- Analog signal can be reconstructed back by applying a filter on the digital signal.

Computer Interface

- Digital Processors works only in the digital domain so typical process these days is to convert the real world analog signal to digital signal, allow the processor to work on the digital data, and reconstruct back the analog signal to be output to the real world.
- So most of the interfaces we are dealing with in this course are in the digital domain.

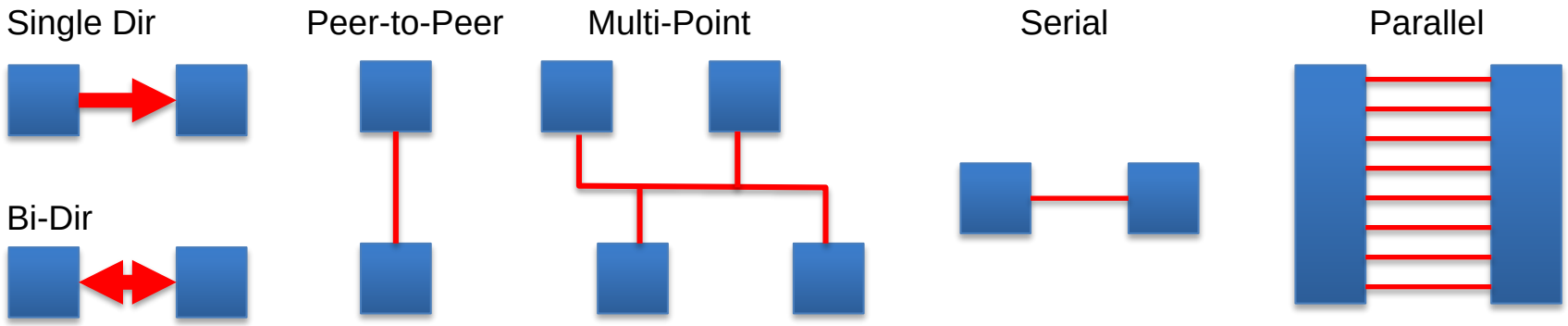
CE1006/CZ1006
Computer Organisation and Architecture

Data Transfer and I/O Interfaces
Electrical Signal Interface

Oh Hong Lye
Lecturer
SCSE, Nanyang Technological University.

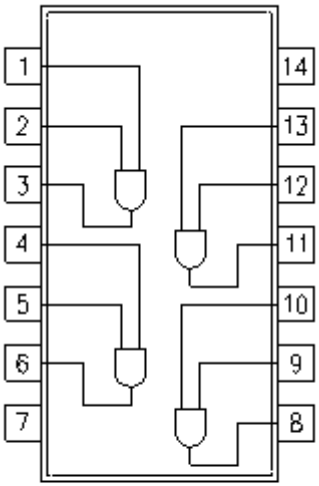
Interface

- A boundary where two or more devices meet to exchange data. Some modes of connection below
  - Peer-to-peer or Multi-point
  - Single-bit Data(Serial) or Multiple-bits Data(Parallel)
- For each mode above, the connection could be Single direction or Bi-direction.

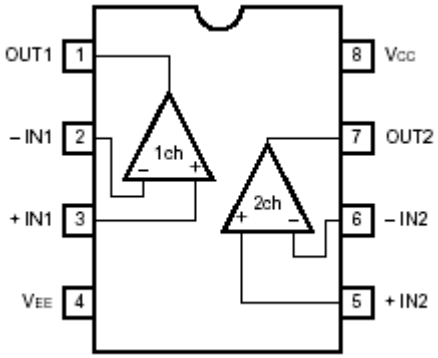


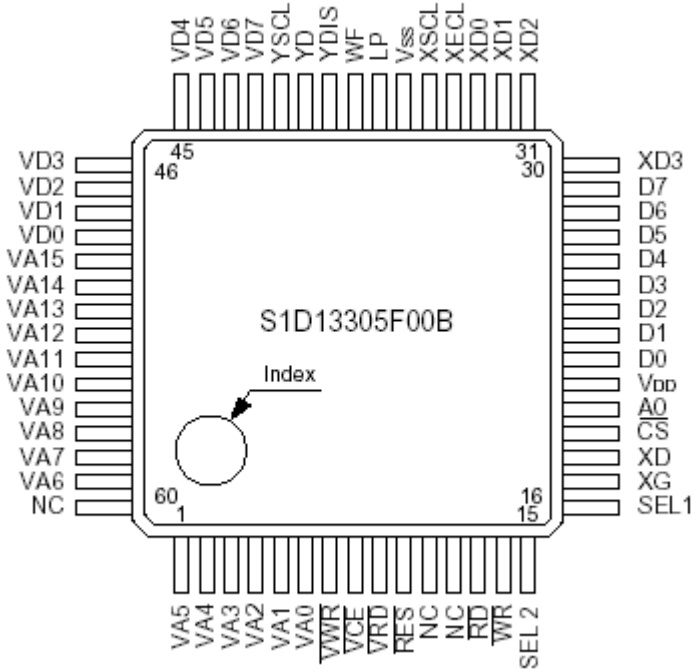
Input, Output and Bi-Directional Pins

- Semiconductor devices interface to the outside world via pins.
- Depending on the device design and configuration, these pins are either an input, output or bi-direction (input and output) pin.



7408
Quad 2-Input
NOR Gate





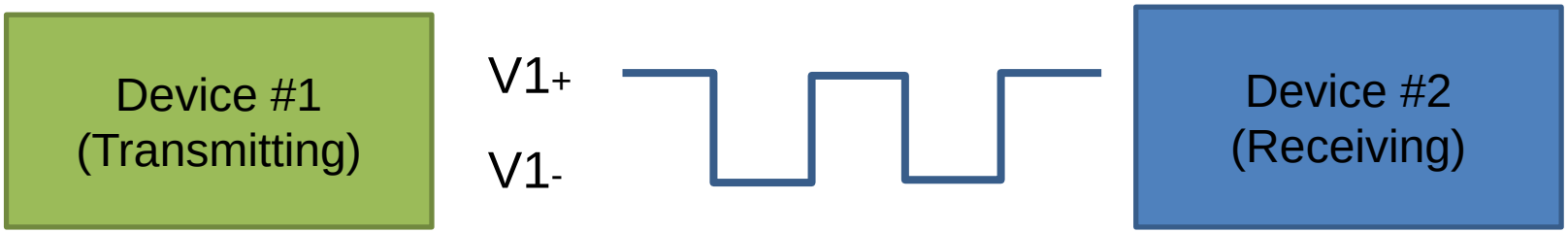
Interface Compatibility

- Interfacing one electronic device to another requires compatibility in
  - Electrical signal level
  - Communication protocol (Handshaking and Data signals).

Electrical Signal level (Safety)

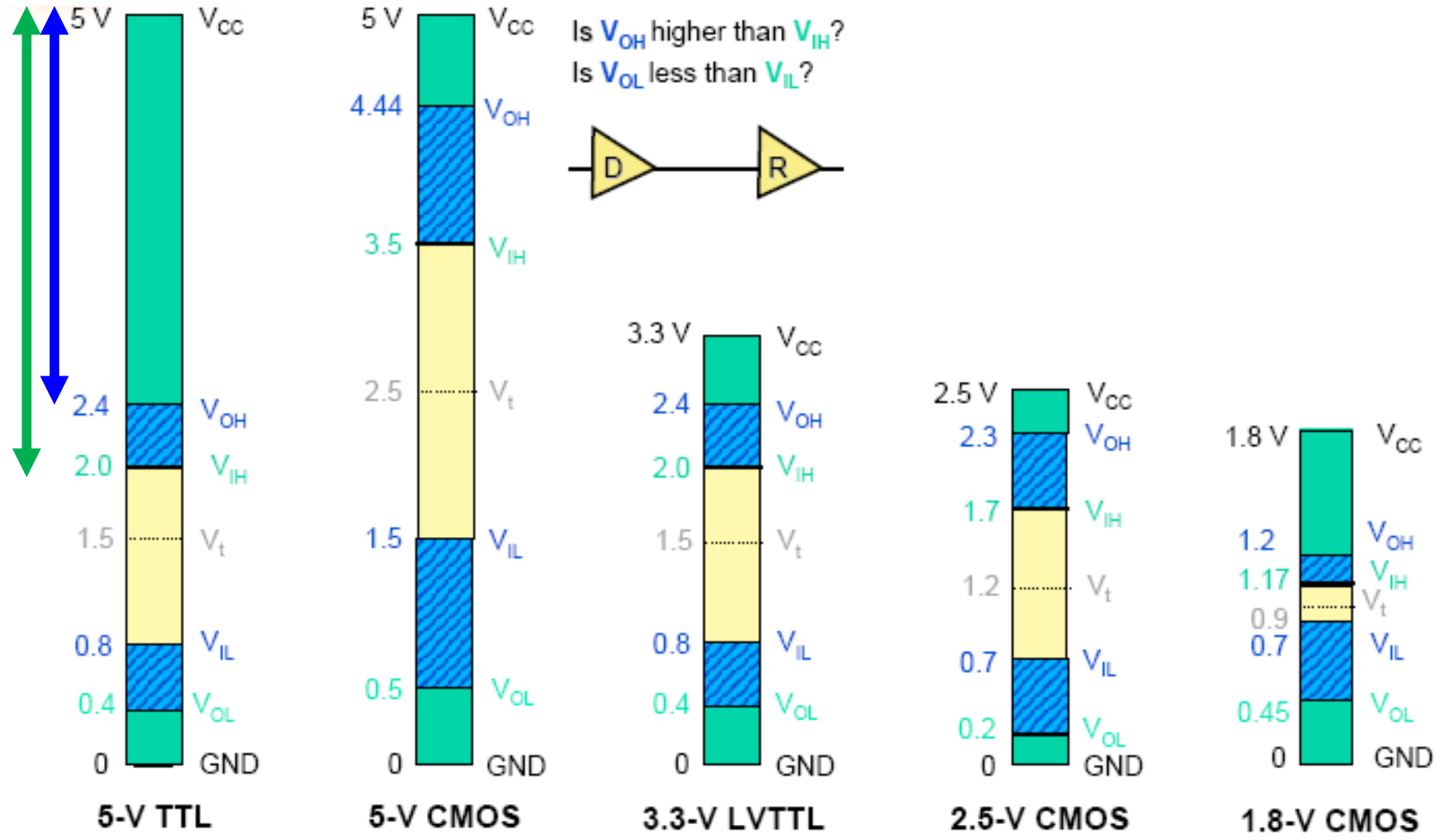
- Primary consideration when connecting two electrical device together.
- Ensure that the output voltage level of output device do not exceed the maximum input voltage level of the input device.
- Electronic devices will either get ‘fried’ i.e. spoilt or have its reliability reduced if the input voltage is higher than what they are designed for.
- So do check the voltage level which the device input/output is operating on (1.8V, 3V, 5V, 15V etc) before connecting them together.

Electrical Signal Level (Digital Data Transfer)



- Device #1
  - Output Voltage level for logic '1' = V1+ (e.g. 5V)
    For typical value of V1+, look for V(OH) parameter in device datasheets.
  - Output Voltage level for logic '0' = V1- (e.g. 0V)
    For typical value of V1-, look for V(OL) parameter in device datasheets.
- Device #2
  - Min Input voltage range to recognised as logic '1' = V(IH)
  - Max Input voltage range to recognised as logic '0' = V(IL)
- In order for Device #2 to sense the logic level correctly,
  - Condition 1: V1+ ≥ V(IH) (e.g. V1+ ≥ 2.0V)
  - Condition 2: V1- ≤ V(IL) (e.g. V1- ≤ 0.8V)

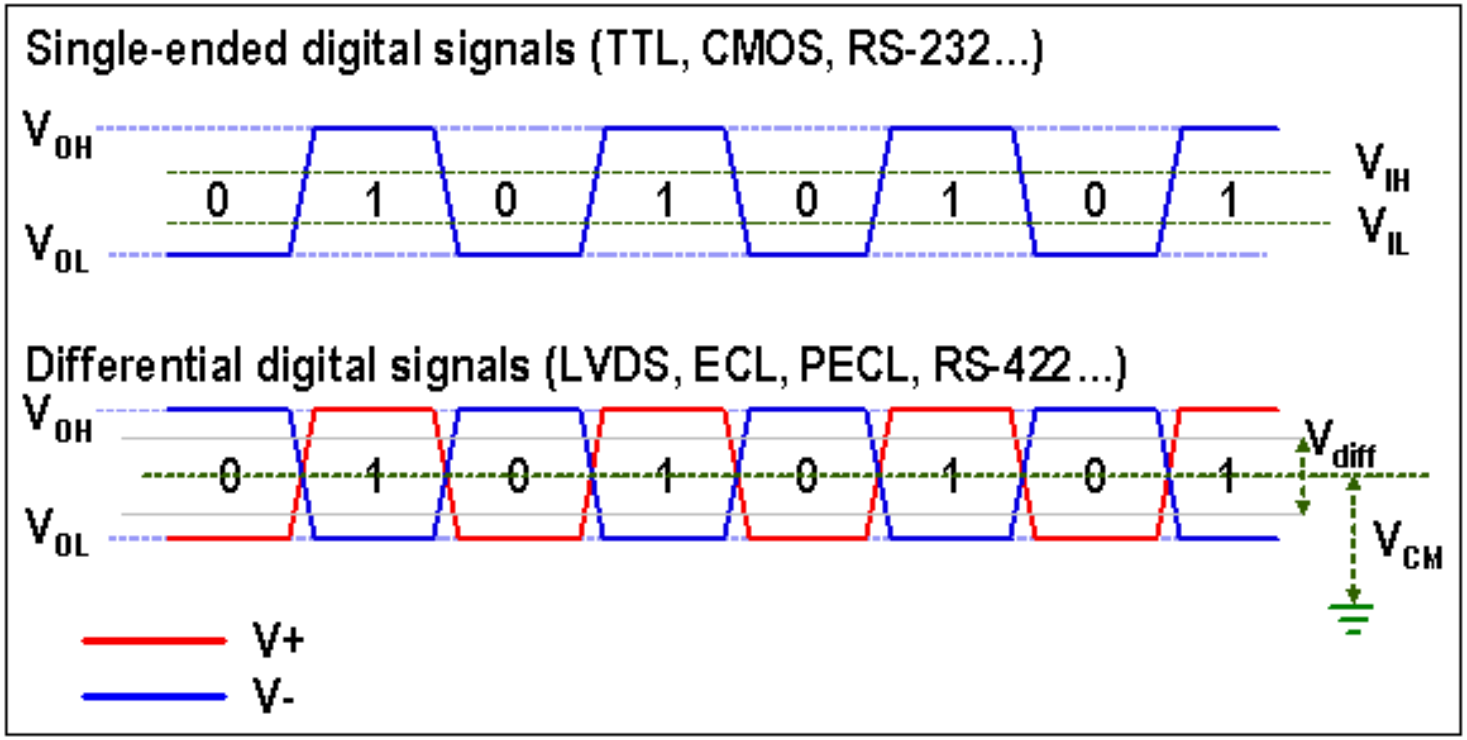
Electrical Signal Level Standards



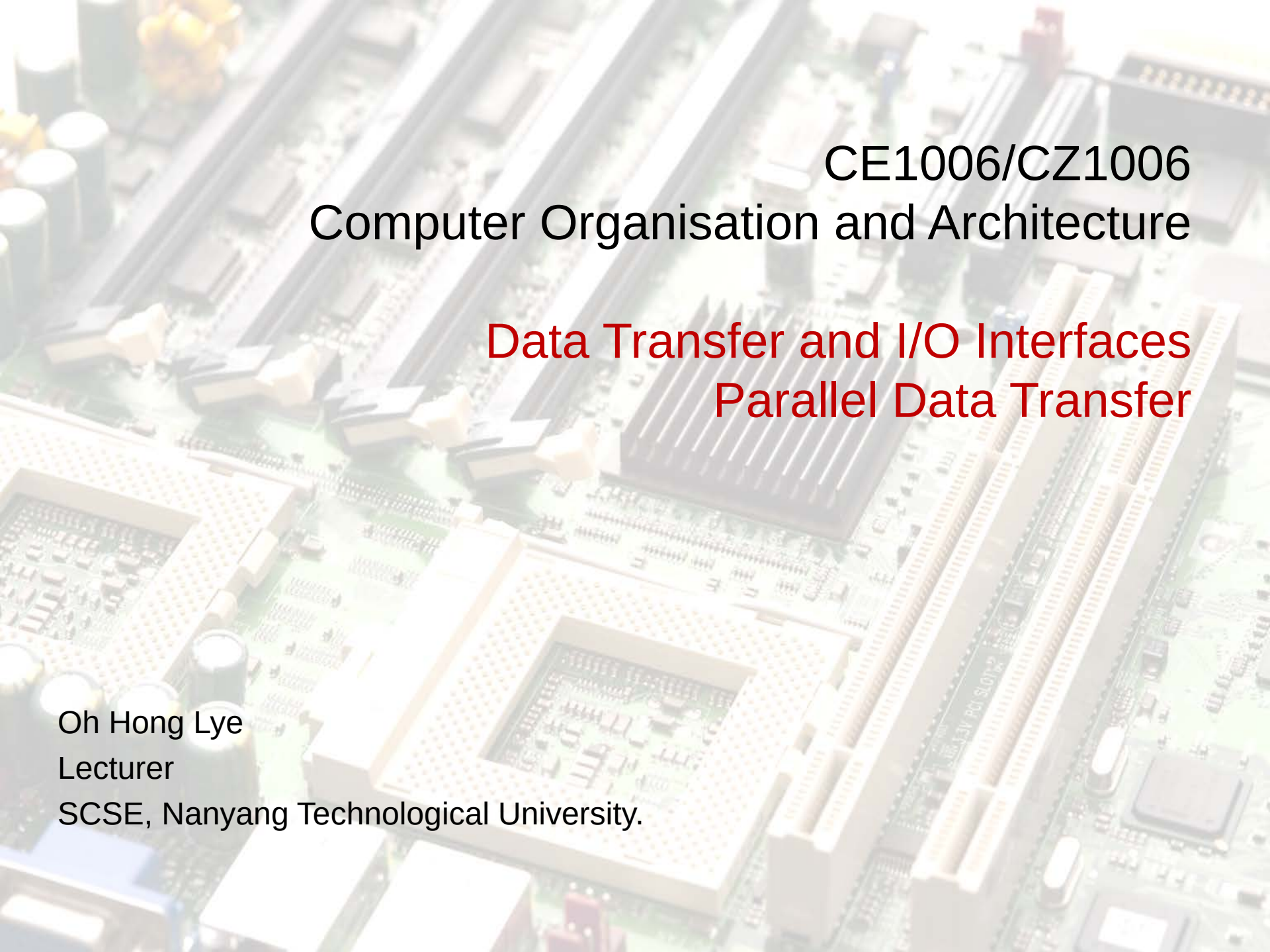
Source: http://denethor.wlu.ca/pc200/logic_charac/voltage_char.png

Differential Signals

- Differential signals has better noise tolerance so is able to be clocked at higher frequency.
- V(CM) = Common Mode Ground.



Source: http://www.ni.com/cms/images/devzone/tut/a/07c0be30313.gif



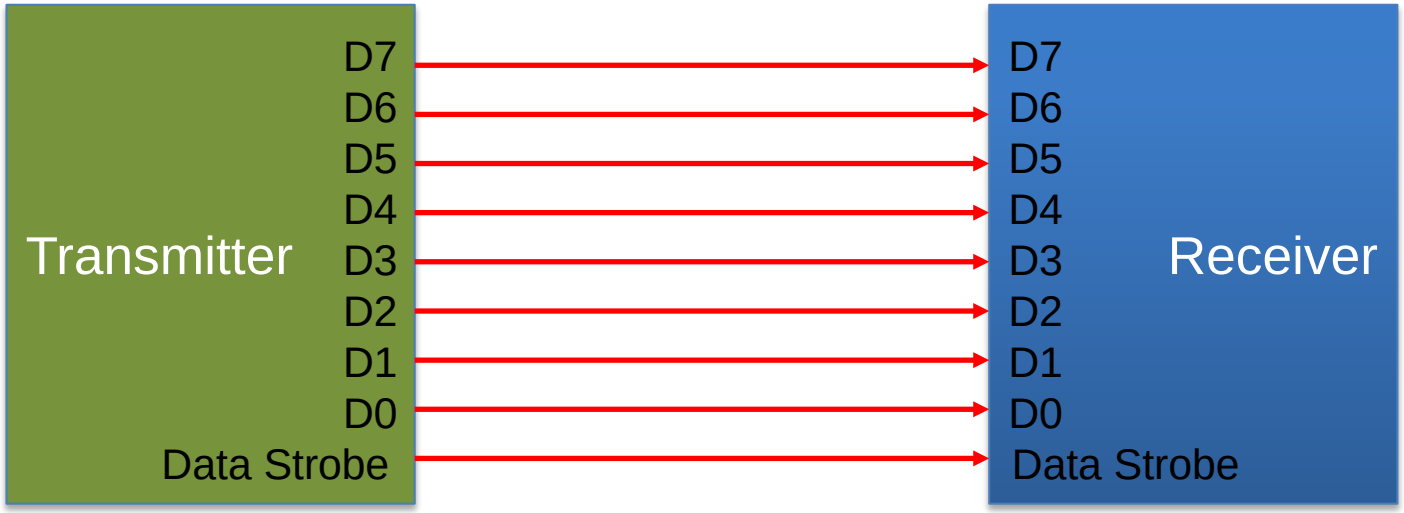
CE1006/CZ1006
Computer Organisation and Architecture

Data Transfer and I/O Interfaces
Parallel Data Transfer

Oh Hong Lye
Lecturer
SCSE, Nanyang Technological University.

Parallel Data Transfer

- Multiple bits of data are transferred simultaneously between two devices.
- Synchronous in nature as some sort of strobe signal is needed to inform the receiver when to latch in the data. E.g. rising edge of strobe signal.
- Able to achieve higher transfer rate than Serial Interface (using same clock).
- But more prone to Signal Skew and Crosstalk (see later slides).

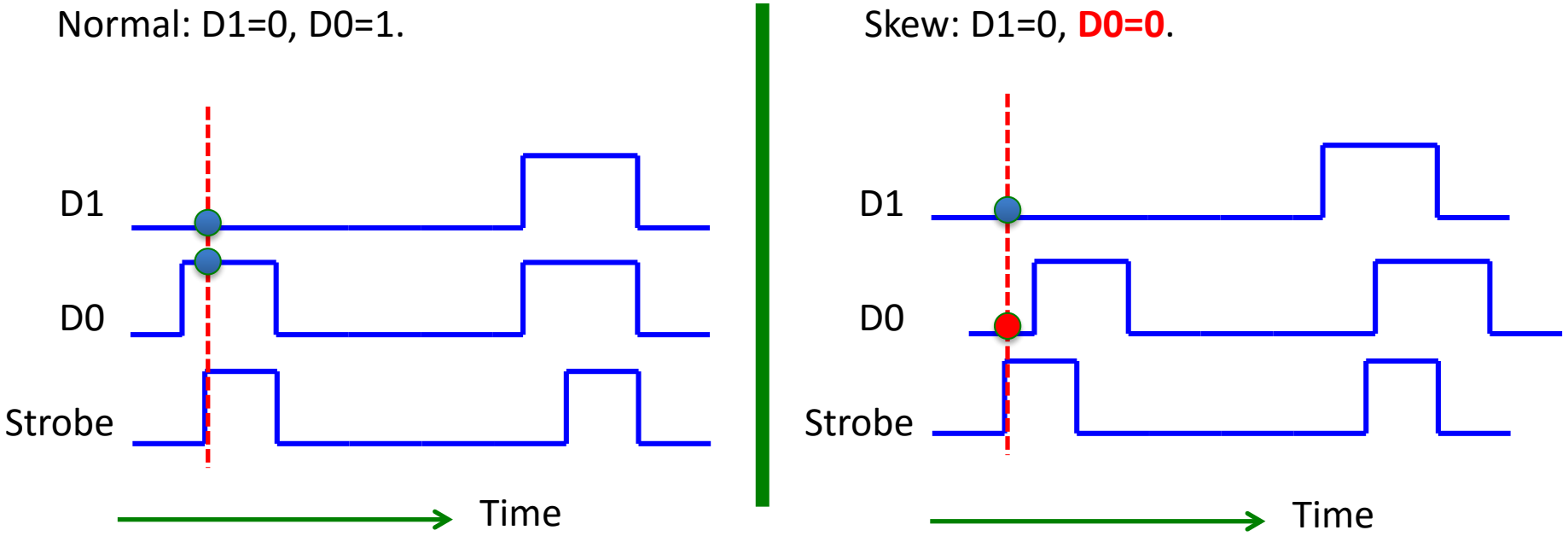


Parallel Data Transfer – Pros and Cons

- Advantages
  - Fast data transfer rate (more bits can be transferred at one time)
  - Hardware interface design tend to be simpler as only strobe signals are needed.
- Disadvantages
  - Affected by Signal Skew and Cross Talk, which limits the maximum clocking speed and transfer distance.
  - Hardware (data cable) can be bulky if data width is large.
  - Need more space to route the PCB traces.
  - Higher hardware cost compared to Serial data implementation.

Signal Skew

- If for some reason (due to circuit design, external interference), the signal in one or more data lines took different amount of time to reach the receiver, then there is a skew between the signals in the parallel data bus.
- Below example illustrate the effect of signal skew. Data is latched by the receiver on rising edge of the strobe signal.
- This result in wrong data (D0=0 instead of 1) being latched by the receiver.



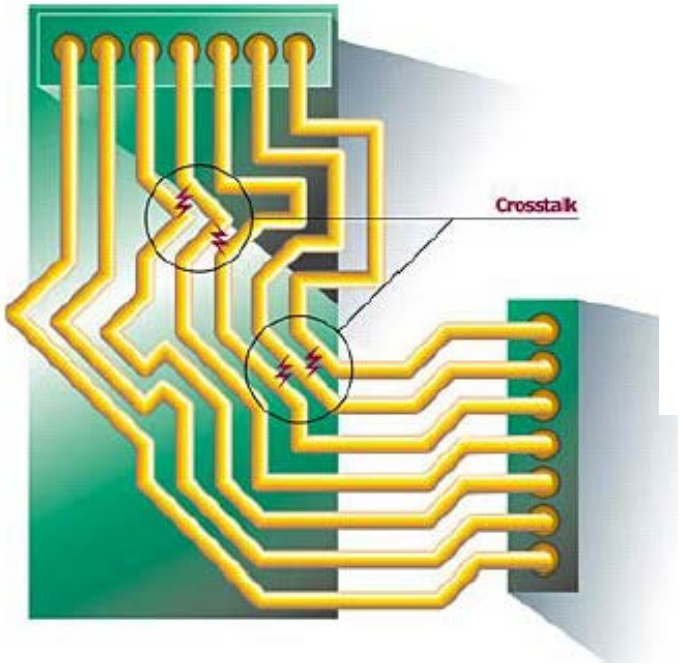
Signal Skew – Contributing Factors

- Signal Skew is due to variation in propagation delay between signals from the same data bus.
- Propagation delay is the time taken for signal to travel between two points in a circuit.
- Capacitance and Resistance of the physical data line is a major contributor to circuit propagation delay.
  - The larger the resistance and capacitance, the larger the delay. Illustrated in the equation $\tau=RC$ where τ is the time constant dictating rate of change of voltage levels in the data line concerned.
- Variation in resistance and capacitance of the signal lines can be due to
  - Variation in PCB trace length/width (lead to change in impedance).
  - Connecting active components (capacitors, inductors, IC etc) to some of the signal lines.

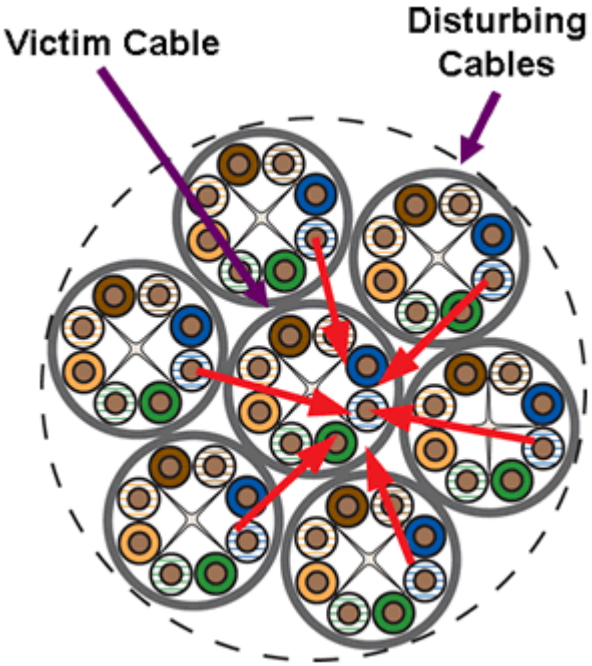
Crosstalk

Party

PCB

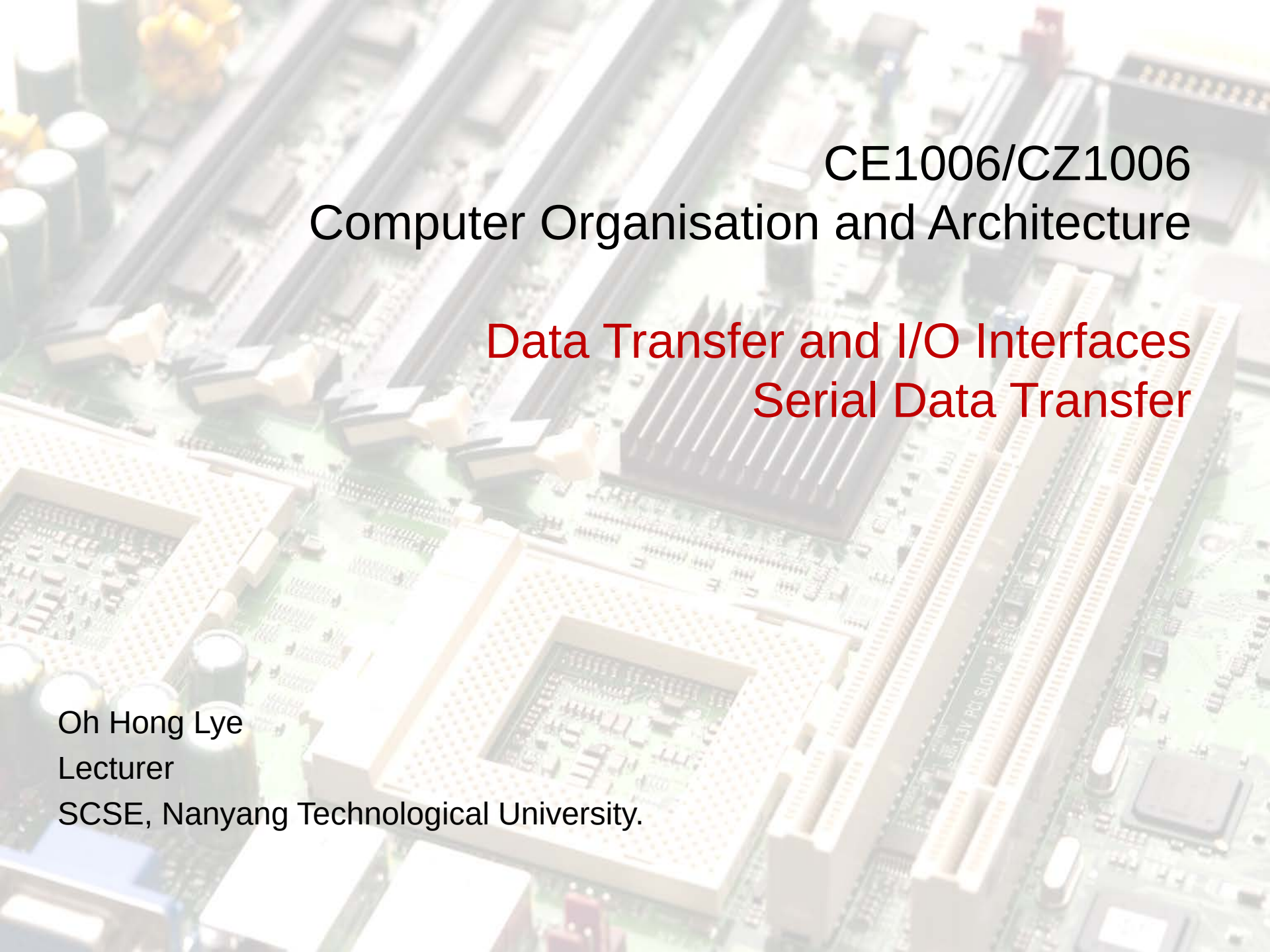


Cable



Crosstalk – Electrical Circuit

- Crosstalk are undesired coupling of signals from one circuit to another circuit.
- In a parallel bus context, the close placement of the data lines in PCB routing or cabling enables the effect of electrical signal in one trace/wire to be coupled over to the other. Creating undesired interference (crosstalk).
- Crosstalk can be transmitted electrically or via electromagnetic radiation (the trace/wire acts like an antenna).

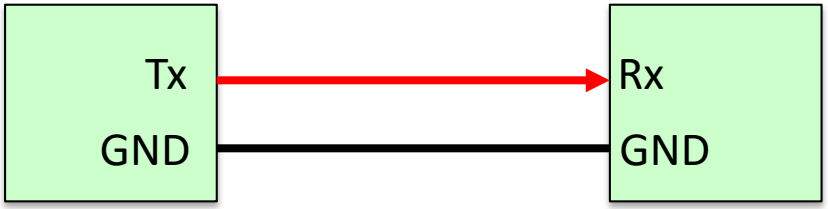


CE1006/CZ1006
Computer Organisation and Architecture

Data Transfer and I/O Interfaces
Serial Data Transfer

Oh Hong Lye
Lecturer
SCSE, Nanyang Technological University.

Serial Data Transfer



- Data is transferred one bit at a time over a single data line. Comparatively, parallel data interface transfer multiple bits simultaneously.

- Less affected by signal skew and crosstalk because there are less electrical wires involved compared to parallel data transfer. Hence, able to support higher frequency clocking.

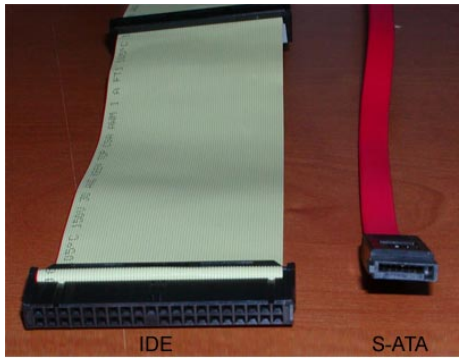
- Data transfer rate lower (compared to parallel interface) given the same clock rate since only one data line is available.

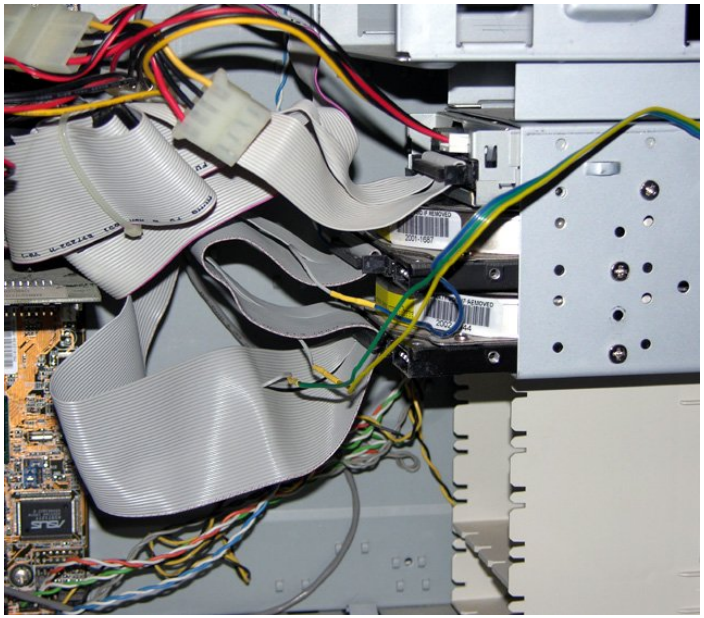
Serial Data Transfer – Pros and Cons

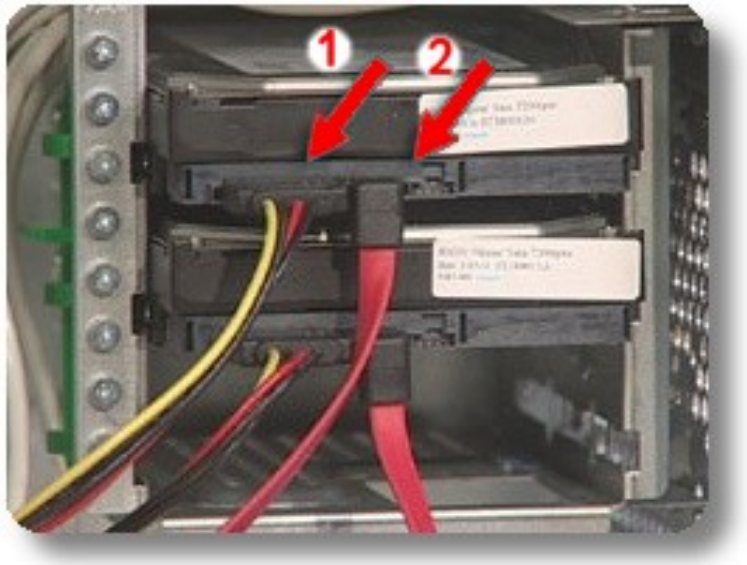
- Advantage
  - Less affected by signal skew and crosstalk because there are less electrical wires involved compared to parallel data transfer. Hence, able to support higher frequency clocking.
  - Able to transfer data reliably over a longer distance.
  - Lower cost since less wires and connectors are needed.

- Disadvantage
  - Data transfer rate lower given the same clock rate since only one data line is available.
  - Hardware interface design typically more complex as it need to handle serial to parallel conversion (Processor only process in words).

Parallel and Serial Comparison

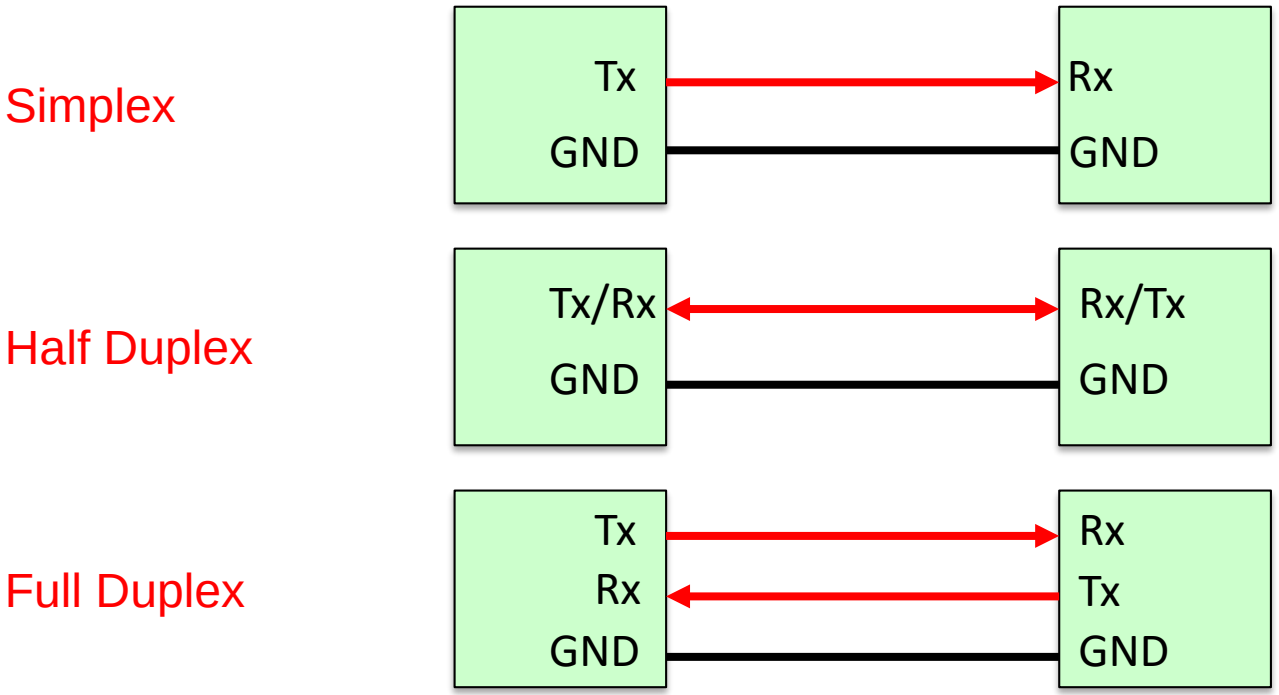




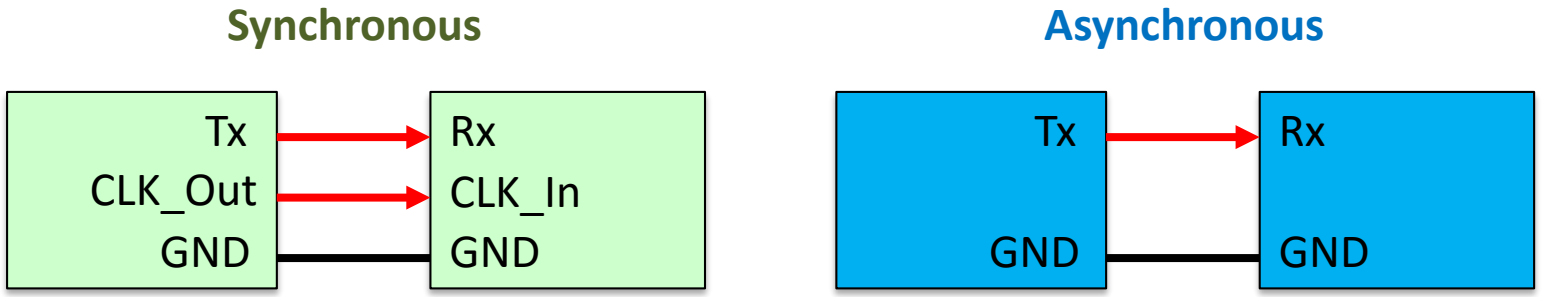


Serial Data Transfer Mode

- Simplex: Data transfer in one direction only.
- Half-Duplex: Data transfer in both direction, but RX and TX is mutually exclusive.
- Full Duplex: Simultaneous RX and TX



Synchronous vs Asynchronous



- If there is a common clock signal between the Transmitter and Receiver, then the communication is termed synchronous. Else, the communication is termed asynchronous.

- In synchronous transmission, there is a common clock signal to synchronise the data transfer. E.g. receiver to latch in the data at every rising edge of the clock.

- In asynchronous transmission, there is no common clock signal so devices have to agree on a pre-fixed clock frequency to use for data transfer.

CE1006/CZ1006
Computer Organisation and Architecture

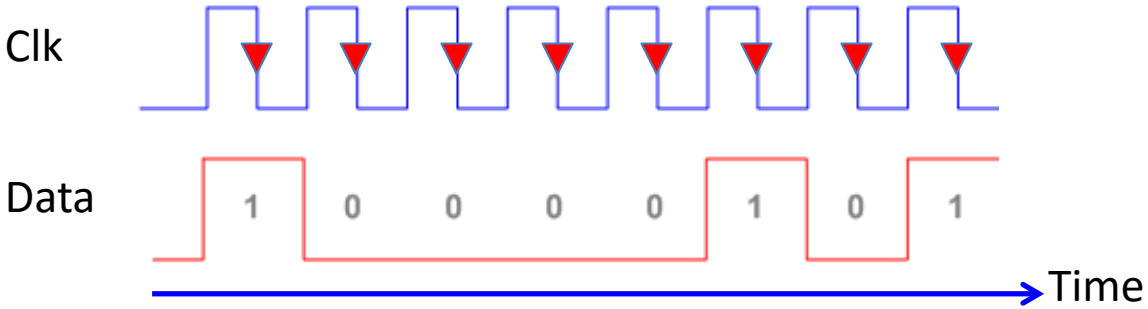
Data Transfer and I/O Interfaces
Synchronous Transfer

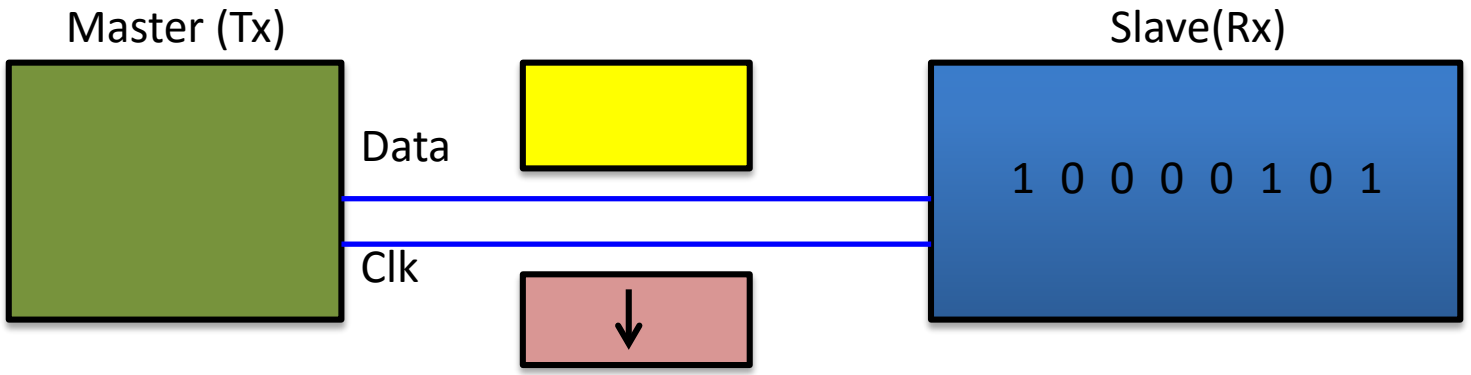
Oh Hong Lye
Lecturer
SCSE, Nanyang Technological University.

Synchronous Transfer

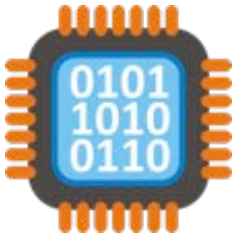
- Common clock signal between transmitter and receiver to synchronise the data transfer.
- Master-Slave configuration. With Master providing the clock signal.
- Potentially allows faster transfer rate since no data overhead is needed to synchronise the transfer.

Data latched at falling edge of clock signal. MSB Tx first.



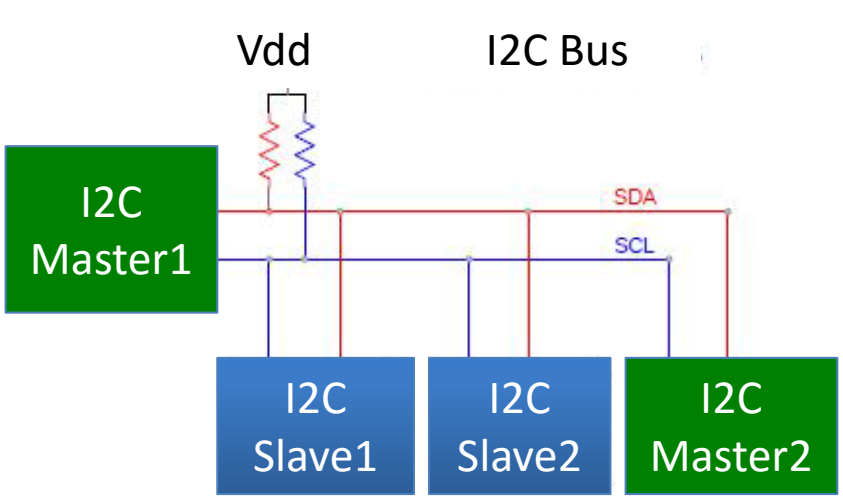


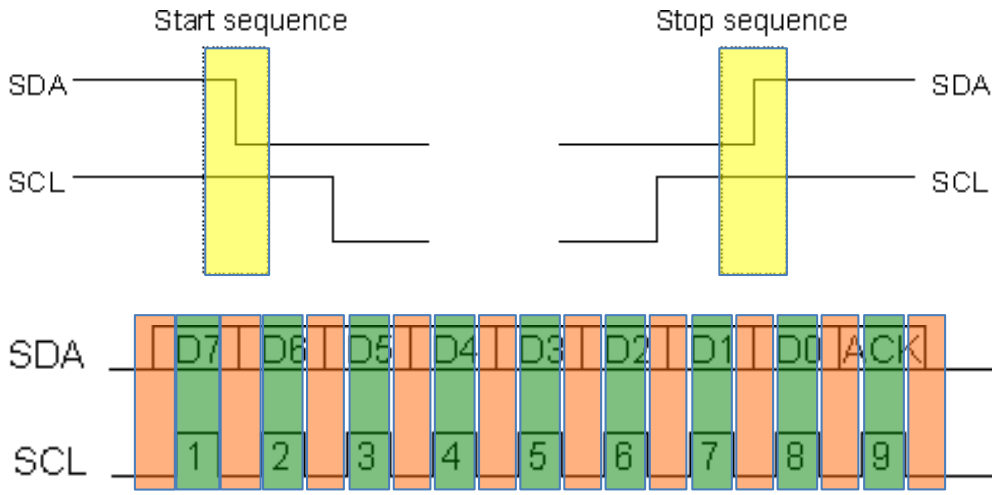
I2C and SPI Bus



- Popular serial bus used by micro-processor to transfer data and control many peripheral devices.
  - Accelerometers
  - Temperature Sensors
  - Touch Screen Controllers
  - Power Supply Modules Configuration
  - Audio/Video Codecs Configuration

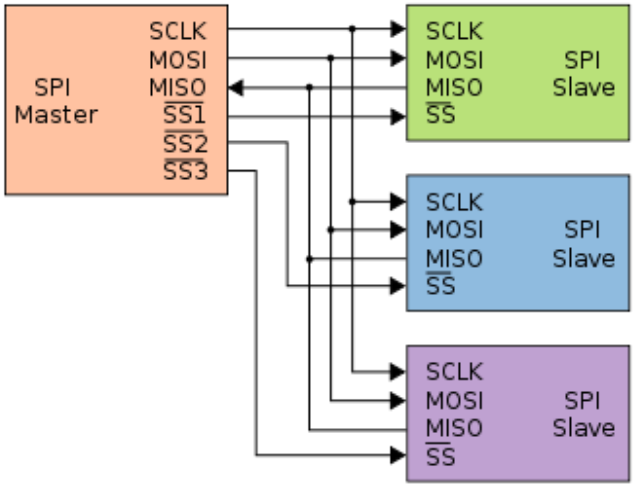
Inter-Integrated Circuit (I2C) Bus

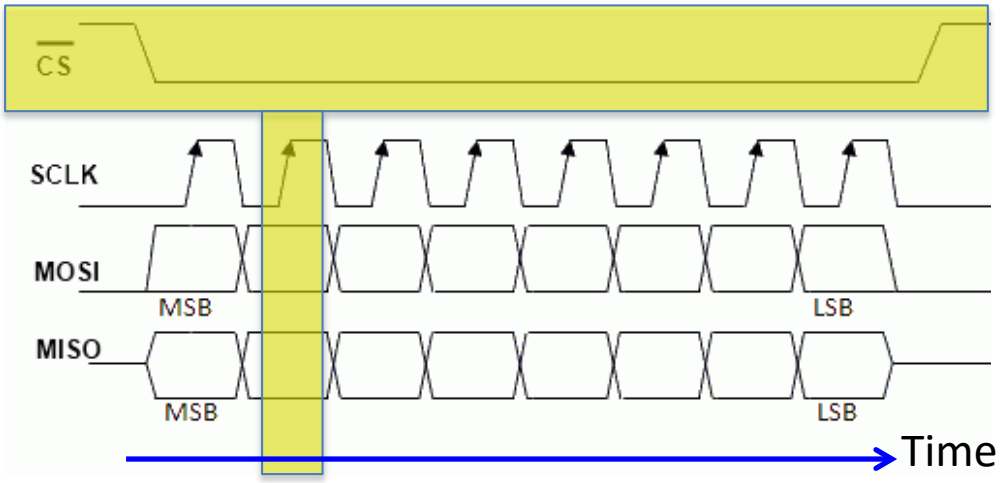




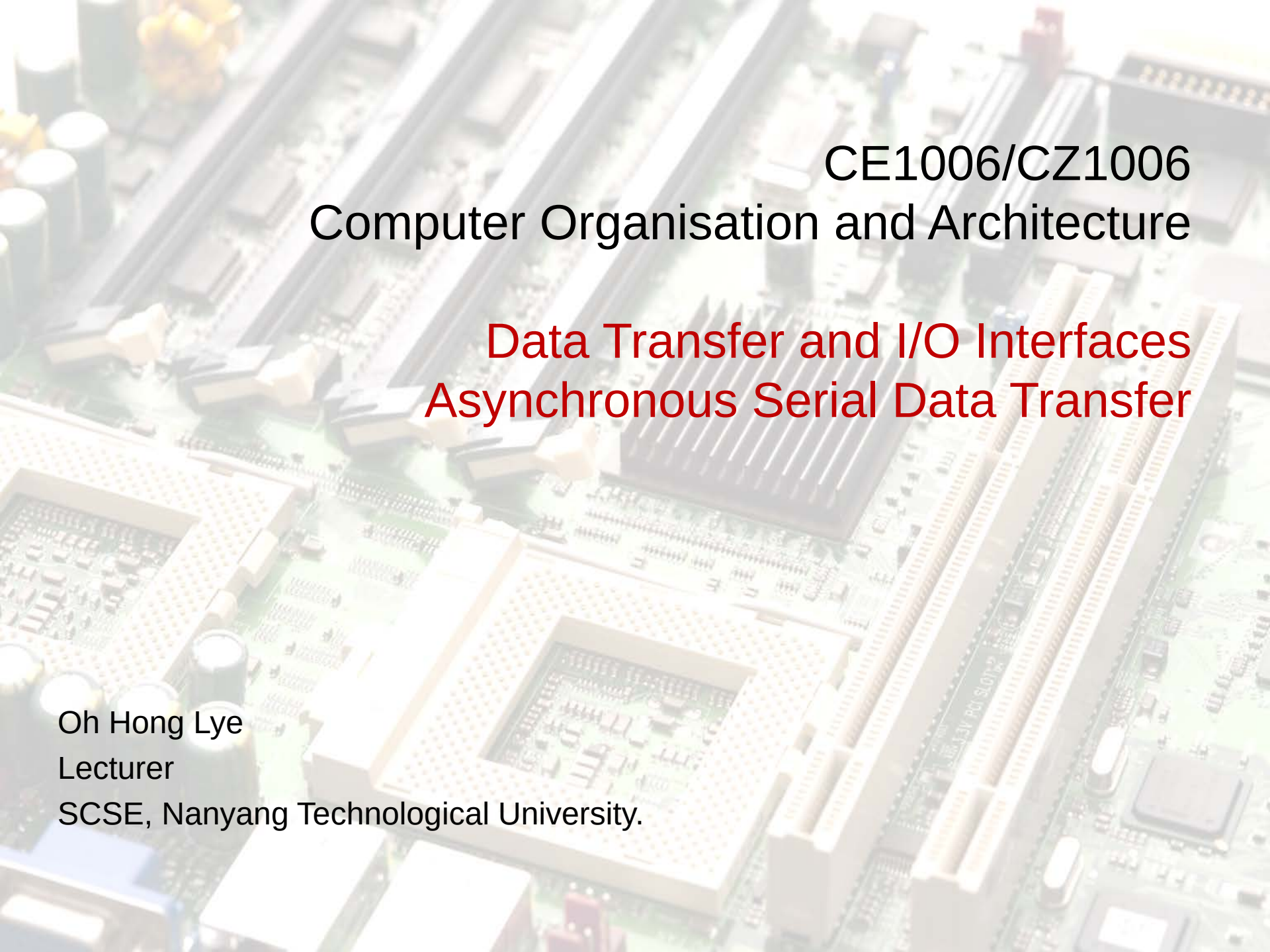
- Two Wire Bus (Multi-Master, Multi-Slave)
  - SDA: Data, SCL: Clock
  - Need Pull-up Resistor on the Bus.
- Data Transfer is initiated with a START bit signalled by SDA being pulled low while SCL stays high.
- SDA sets the 1st data bit level while keeping SCL low
- The data is sampled (received) when SCL rises for the first bit.
- This process repeats, SDA transitioning while SCL is low, and the data being read while SCL is high.
- A STOP bit is signalled when SDA is pulled high while SCL is high.

Serial Peripheral Interface (SPI) Bus





- To start the transfer
  - Slave Select (SS) has to be pulled Low.
- Data transfer
  - Data on MOSI and MISO latched in on rising/falling clock edge (configurable)
  - MOSI: Master-Out-Slave-In (Master Output, Slave Input)
  - MISO: Master-In-Slave-Out (Slave Output, Master Input)
- Allow multipe slaves via use of multiple Slave Select.



CE1006/CZ1006
Computer Organisation and Architecture

Data Transfer and I/O Interfaces
Asynchronous Serial Data Transfer

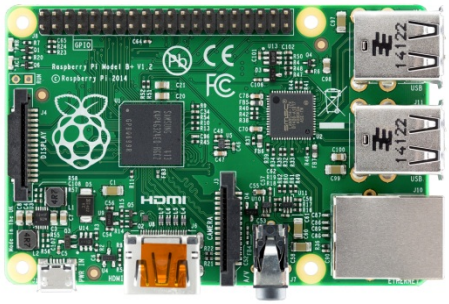
Oh Hong Lye
Lecturer
SCSE, Nanyang Technological University.

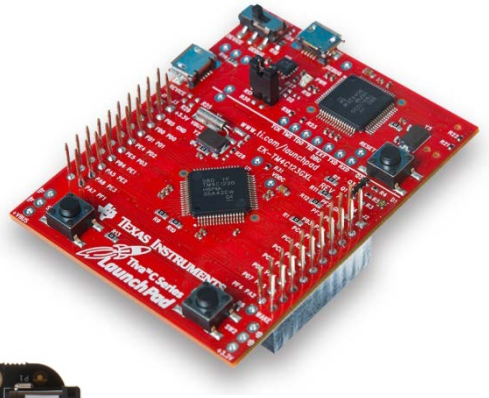
Asynchronous Transfer

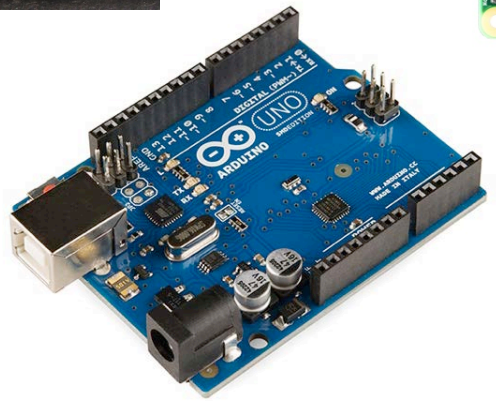
- No common clock is provided between transmitter and receiver.
- Prior to the transmission, the receiver needs to know the transmitting clock rate and the number of bits that are to be transferred with each data packet.
- Special SYNC words are used to indicate START/STOP condition.
- Upon receiving the START SYNC Word, the receiver then use its own local clock to track the timing.
- Potential skew issue between the two local clocks as transmission progress.
- Asynchronous Transmission typically also uses SYNC word/bits to provide occasional time stamp for receiver to synchronise its clock to the transmitter clock (helps to reduce clock skew between the two clocks).

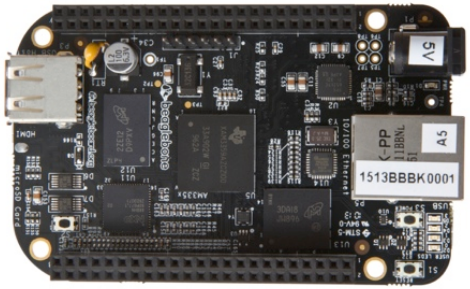
Universal Asynchronous Receiver Transmitter (UART)

- One of the most commonly used serial interface.
  - PC Serial COM Port (RS232) uses UART protocol.
  - Communication interface between PC and many processor development boards e.g. Arduino Board, TIVA-C Launchpad etc
  - Many USB devices uses a Virtual COM Port implementation to connect to the PC.



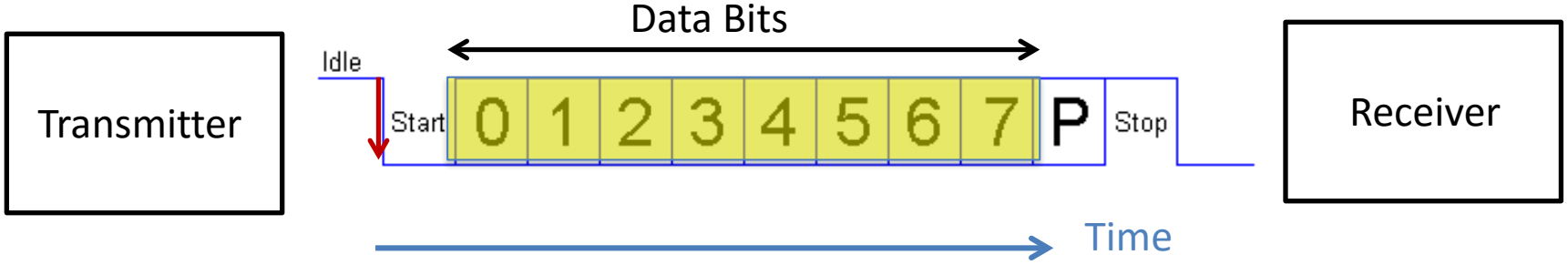






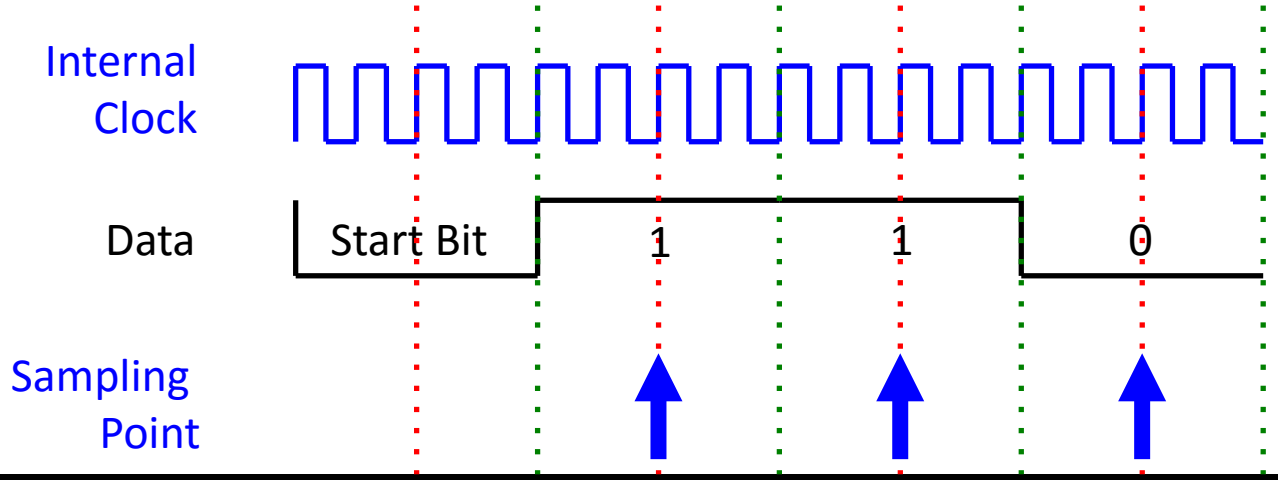
UART Transmit

- During Idle State, the data line is in ‘powered’ state i.e. Logic ‘1’.
- The transmitter send a **START** pattern (logic ‘0’) to alert the receiver.
- Sending a logic ‘0’ from idle state (logic ‘1’) will result in a falling edge on the data line. This is typically used by the receiver to detect the start of transmission.
- This is followed by the actual **DATA** at a frequency known to the receiver. The transmitting clock rate is also known as the **baud rate**, and determines the number of bits transmitted per second.
- A **PARITY** bit (optional) may also be sent for the receiver to check the integrity of the data packet.
- A **STOP** pattern (1 or more bits) terminates the transmission.



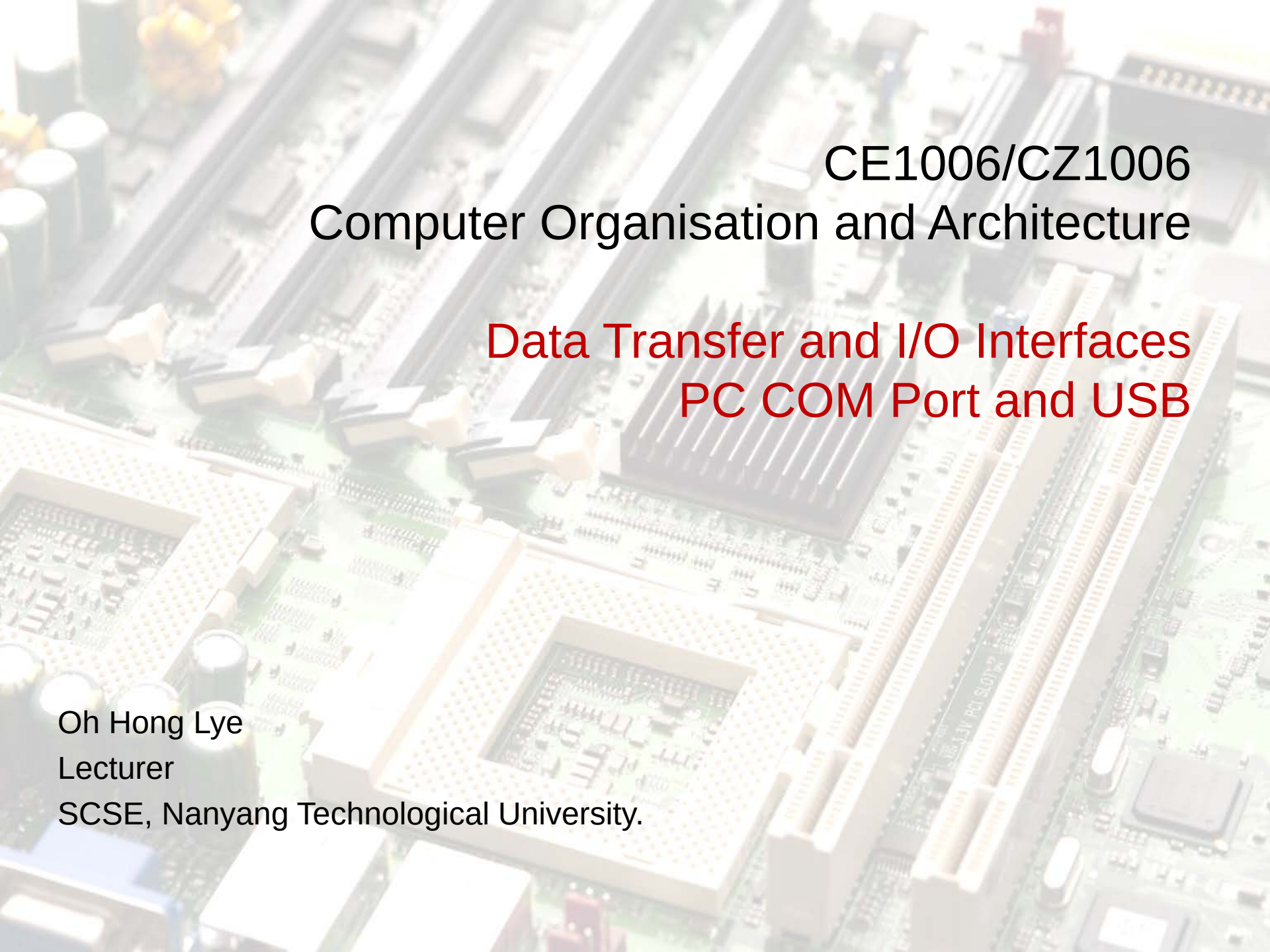
UART Receive

- Receiver monitors the Data line for the Start Bit. In real world design, the falling edge on the data line will trigger an interrupt in the microprocessor to start the receiving process.
- Needs to know the baud rate in order to sample the data bits correctly.
- Internal clock of the UART typically run at a multiple e.g. 16X of the baud rate so as to timed the sampling closed to the middle of each data bit.
- Oversampling is commonly used to increase accuracy.
- Below is an example of UART receive with internal clock running at 4X baud rate (for illustration only). Internal clock rate is typically faster than 4X baud rate in real world implementation.



Parity Bit

- If parity scheme is enabled, the receiver will also sample the parity bit and checks for parity error.
- If even parity scheme is used, and there is an odd number of 1's in the data packet, the parity bit should indicate 1 such that the data bits including the parity bit gives even number of 1's.
- Else, parity error occurs.
- Vice versa for odd parity scheme.
- The receiver then samples the stop bit(s) for framing error.



CE1006/CZ1006
Computer Organisation and Architecture

Data Transfer and I/O Interfaces
PC COM Port and USB

Oh Hong Lye
Lecturer
SCSE, Nanyang Technological University.

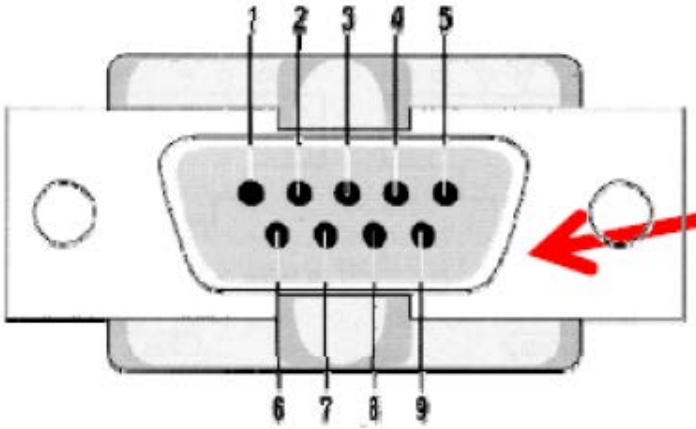
PC COM Port

- Many PC used to have a serial port (a.k.a. COM port).
- In essence, it is UART + RS232.
- The processor has UART pins which typically drive (or receive) the signals (transistor-transistor-logic or TTL) to 3.3V or 5V external to the processor.
- RS232 requires ±15V (max) to communicate up to a distance of 100-200 m.

LOGIC LEVELS			
STANDARD		LOGIC 0 (Volts)	LOGIC 1 (Volts)
TTL	INPUT OUTPUT	0.0 to 0.8 0.0 to 0.4	2.0 to 5.0 2.8 to 5.0
RS-232	INPUT OUTPUT	+3 to +15 +5 to +15	-3 to -15 -5 to -15

COM Port Connector Pinout

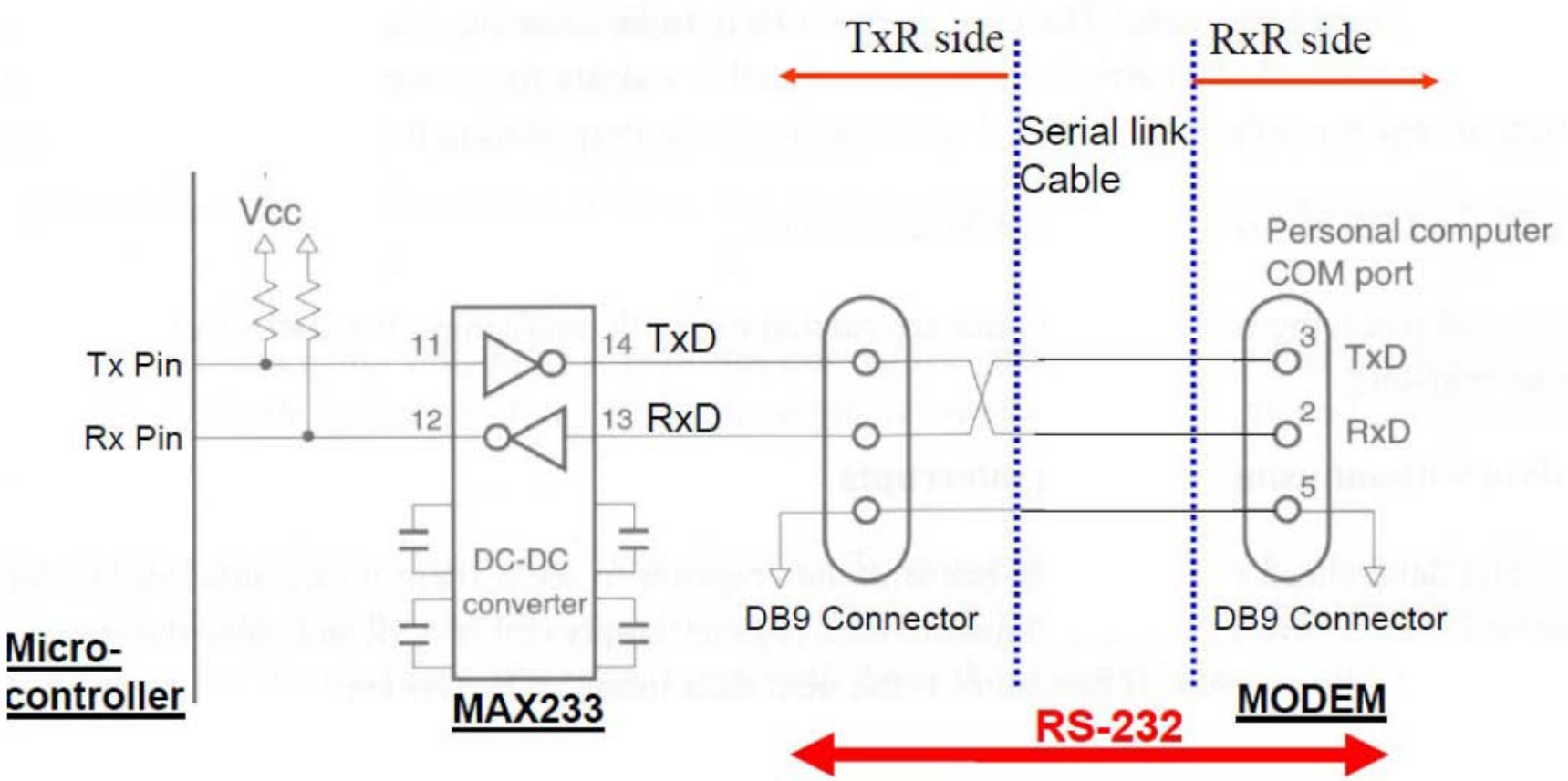
RS-232, a serial interface standard, specifies electrical and mechanical interface, i.e., D-type connector (DB 25 and DB 9), voltage levels used, maximum bit rate and maximum distance of operation, etc.



RS-232 port
(DB9)

1. Data Carrier Detect
2. Receive Data
3. Transmit Data
4. Data Terminal Ready
5. Signal Ground
6. Data Set Ready
7. Request to Send
8. Clear to Send
9. Ring Indicator

COM Port Hardware

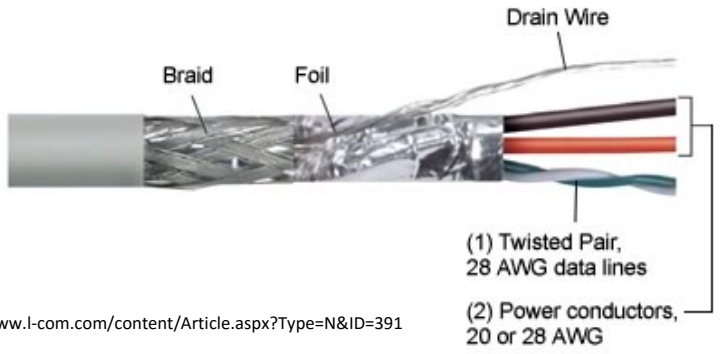


Universal Serial Bus (USB)

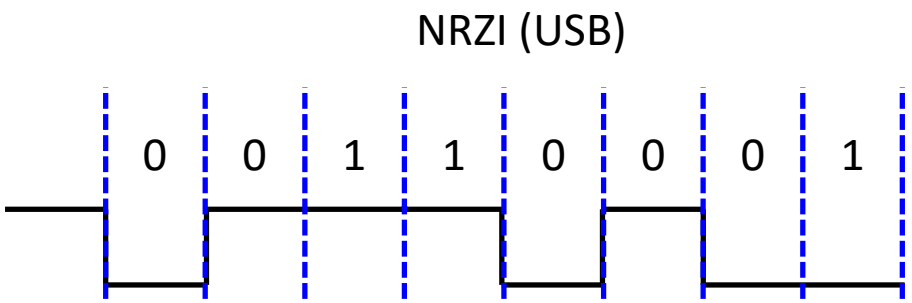
- Serial Bus designed to standardise connection of computer peripherals (Keyboard, Mouse, Printers, Disk Drives, Network Adapters, Digital Cameras etc).
- Effectively replaced a number of interface bus e.g. Serial Bus (COM Port), Parallel Port.
- USB Transfer rate
  - USB1.0: up to 12Mbps
  - USB2.0: up to 480 Mbps
  - USB3.0: up to 5Gbps
- What are the design consideration put in to allow USB to scale up to 5Gbps?

A look at USB

- Cable
  - Grounded Shielding foil to reduced effects of electromagnetic field (Crosstalk reduction)
  - Twisted Pair configuration for data cable (D+, D-) to reduce effect of crosstalk.
- Signaling
  - Differential Signaling (D+, D-) reduce effect of crosstalk.
  - NRZI (Non-Return-to-Zero Inverted) coding incorporate sync information into data stream. '0' => Transition of the data signal. '1' => No transition.
  - Bit Stuffing: '0' inserted when there is consecutive six '1's. This force a transition to allow slave to synchronise its internal clock.
  - Full Duplex transfer used in USB3.0 to increased the transfer rate. No time wasted in switching data transfer direction.



Pic Source: http://www.l-com.com/content/Article.aspx?Type=N&ID=391



Summary

- Needs for Interfacing
- Consideration when interfacing two electrical modules/components
  - Electrical Signal level
  - Communication Protocol
- Types of Signal
  - Analog
  - Digital
- Digital interface
  - Parallel
  - Serial
- Interface Standard
  - INTEL FSB
  - IEEE 488 (GPIB)
  - SPI
  - UART
  - USB