

Cx1106

Computer Organization and Architecture

Computer Systems Overview

Oh Hong Lye

Lecturer

School of Computer Science and Engineering, Nanyang Technological University.

Email: hloh@ntu.edu.sg

Computer System Block Diagram

External Devices

User Interface Devices
(KeyBoard, Mouse,
Touch Screen)

Digital Audio/Video
Input/Output Devices

Analog Audio/Video
Input/Output Devices

Hard Disk Drive
Solid State Drive

Interfaces

Analog-to-Digital
Digital-to-Analog

UART, I₂C, SPI,
Parallel

Wired Comms
(USB, SATA, HDMI)

Wireless Comms
(Wifi, Bluetooth)

will have
"modules or
the interface to
allow communication
with external devices

to diff
devices

External Memory

DRAM
SRAM

Flash

store codes &
data in
entire
program

allows processor core
to talk to external
memory and the interface
modules

System Bus

Processor Core

CPU

Interrupt
Controller

DMAC

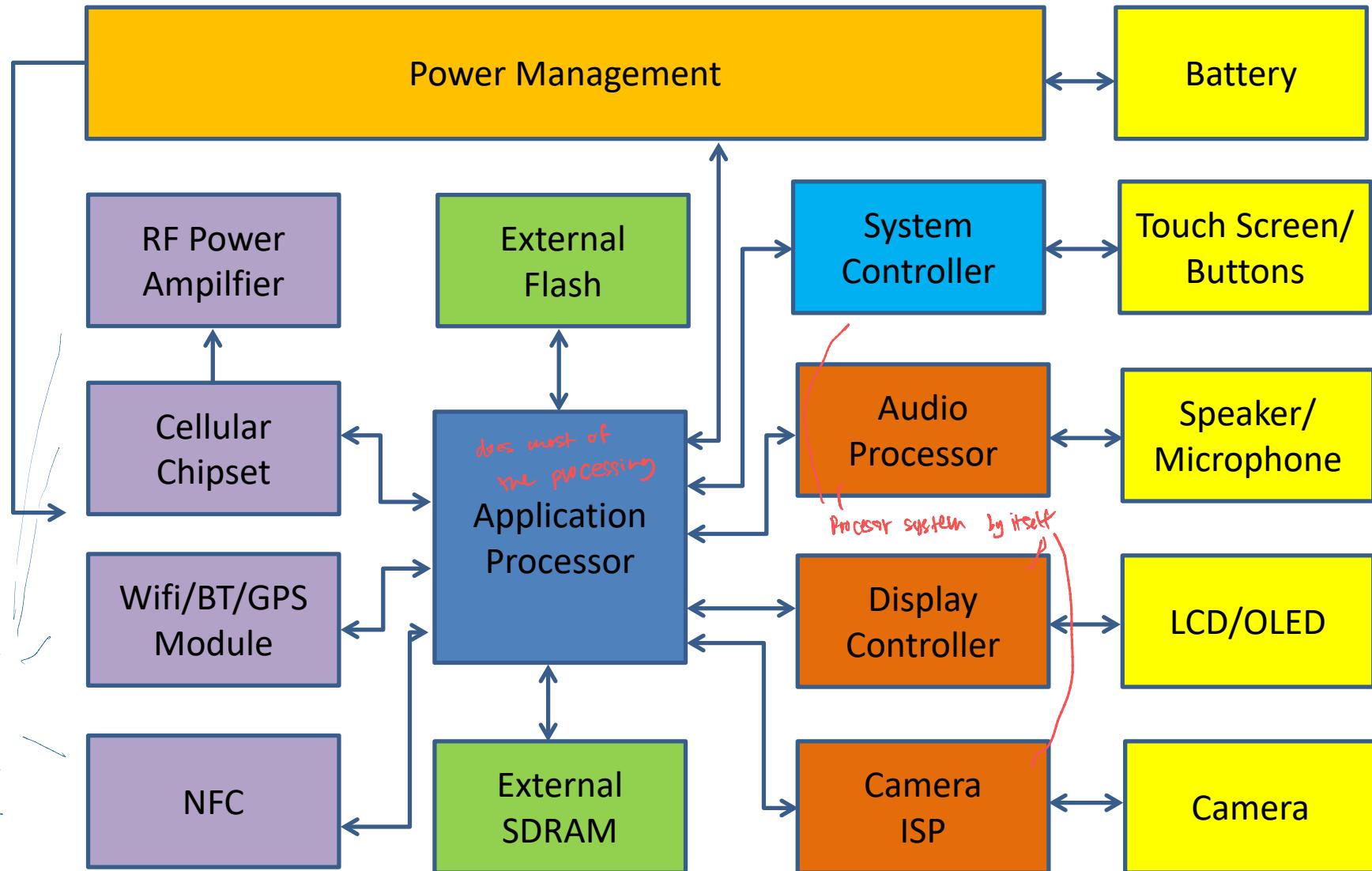
Cache

VM
Mgmt

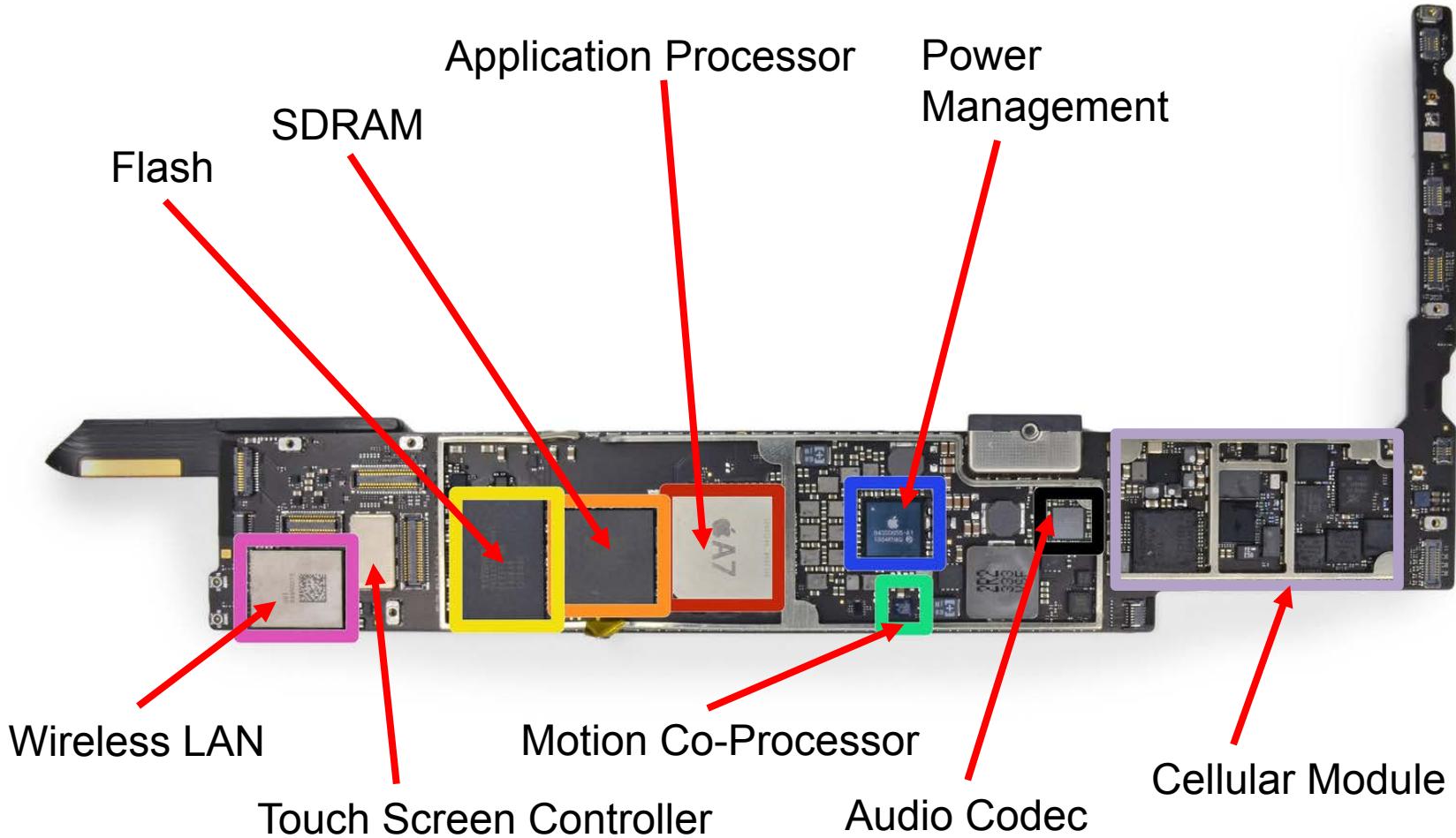
Internal
Memory

Phone can 8 to 11, or even 18 processors

Example: Smart Phone

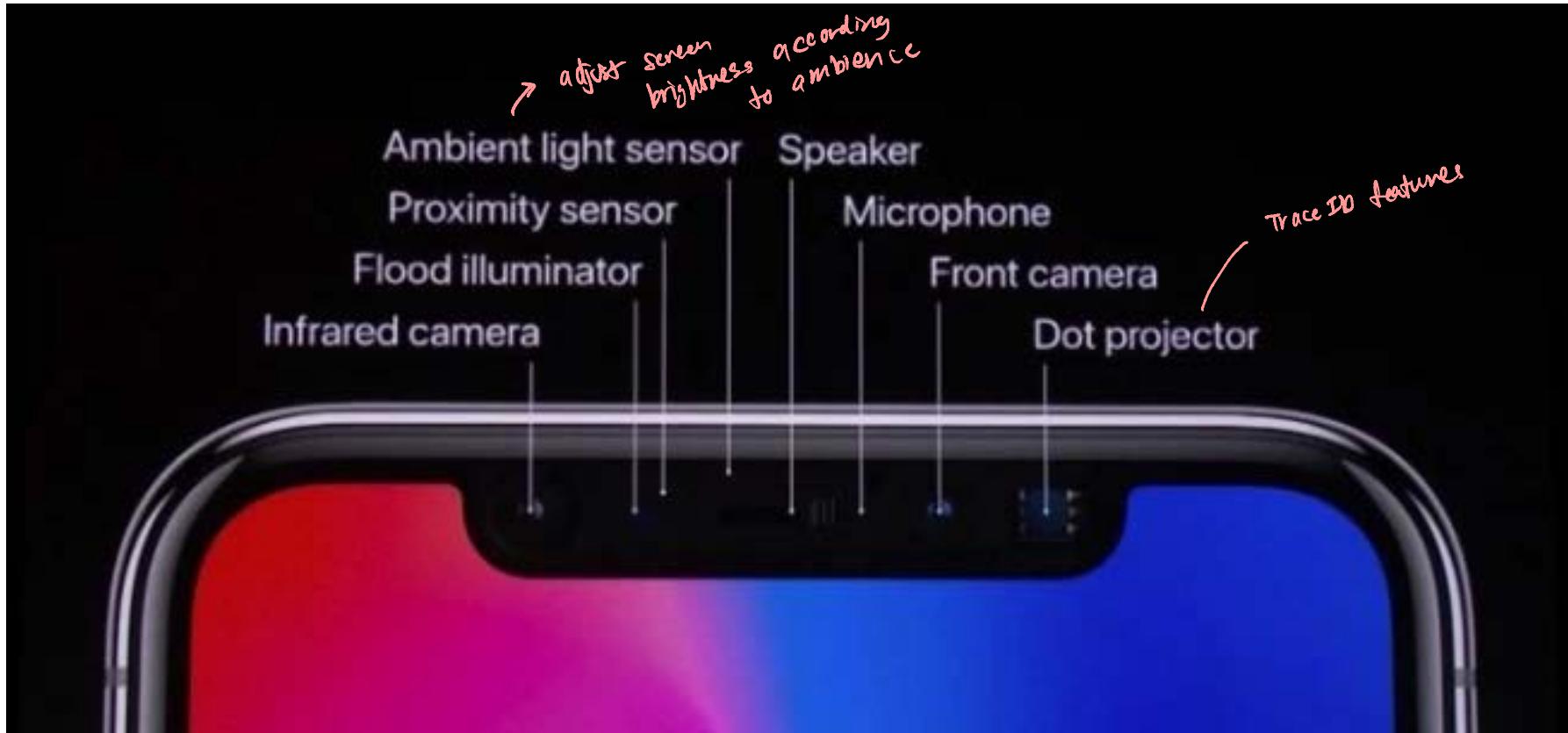


iPad Air Main PCB



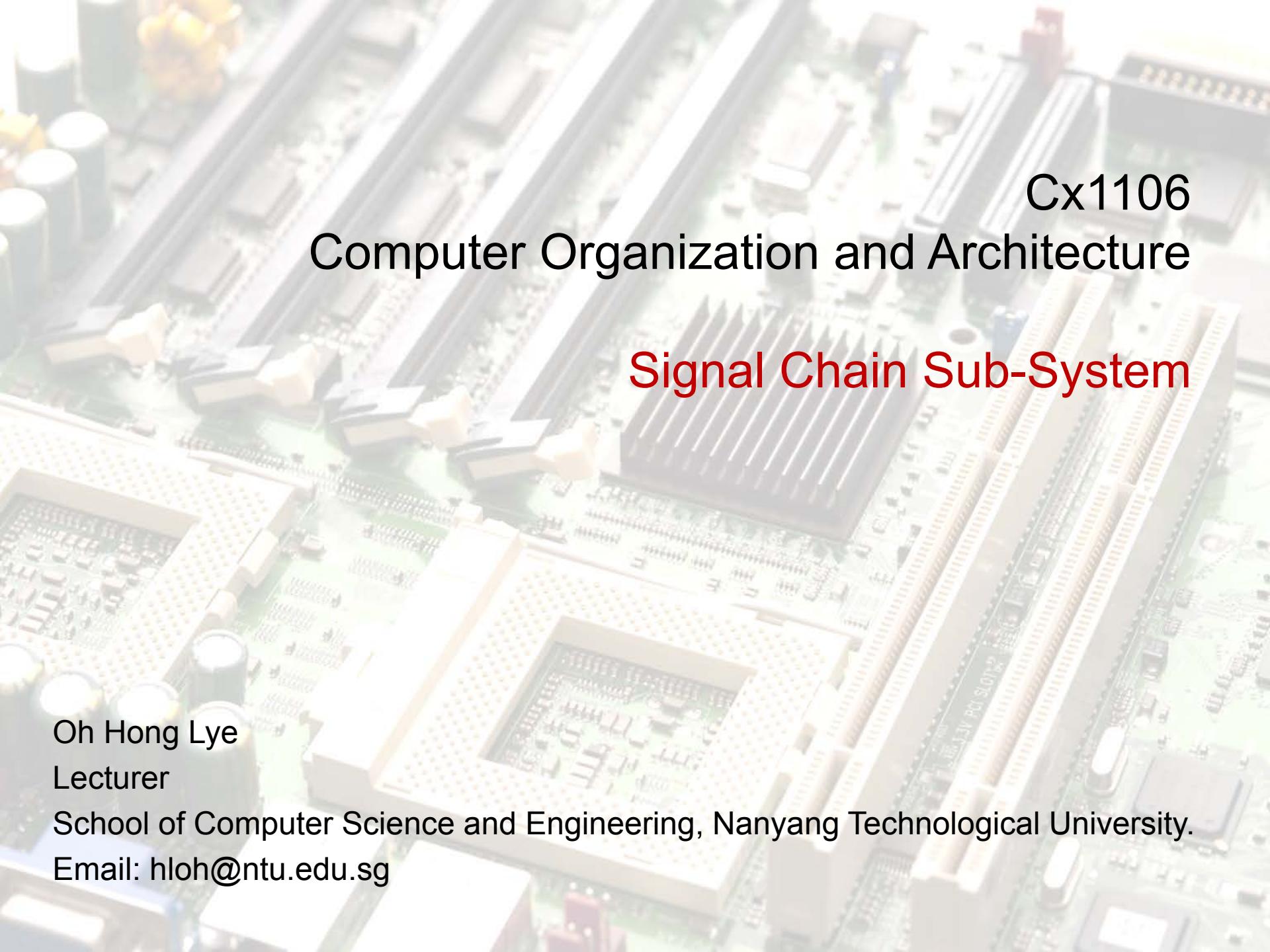
Source: <https://www.ifixit.com/Teardown/iPad+Air+LTE+Teardown/18907>

iPhone front facing peripherals



Sub-Systems in a Computer System

- Processor Core
 - Signal Chain Sub-System
 - ADC-DAC
 - Parallel and Serial (UART/SPI) Digital Interfaces
 - Direct Memory Access Controller
 - Interrupt Controller
 - Memory Sub-System
 - Semiconductor Memories (SRAM, DRAM, FLASH)
 - Flash memory based Solid State Drives
 - Magnetic Hard Disk Drives
 - Cache Memory Management
 - Virtual Memory Management
 - Communication and User Interface Sub-System
 - Wired (USB, SATA, HDMI)
 - Wireless (Wifi, Bluetooth)
 - Input Devices (KBM, Capacitive Touch, Camera, Microphone)
 - Output Devices (Display, Speakers)
- ¹
INT SO
in output



Cx1106

Computer Organization and Architecture

Signal Chain Sub-System

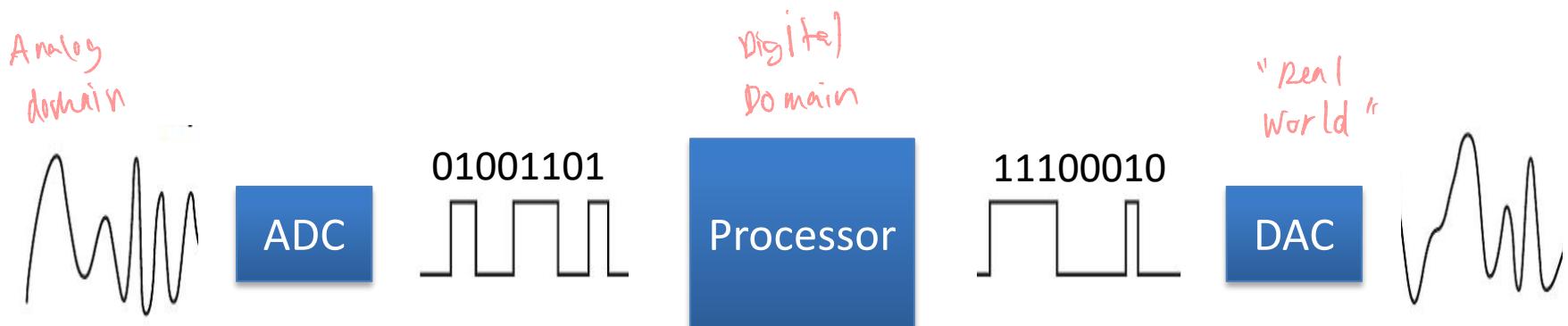
Oh Hong Lye

Lecturer

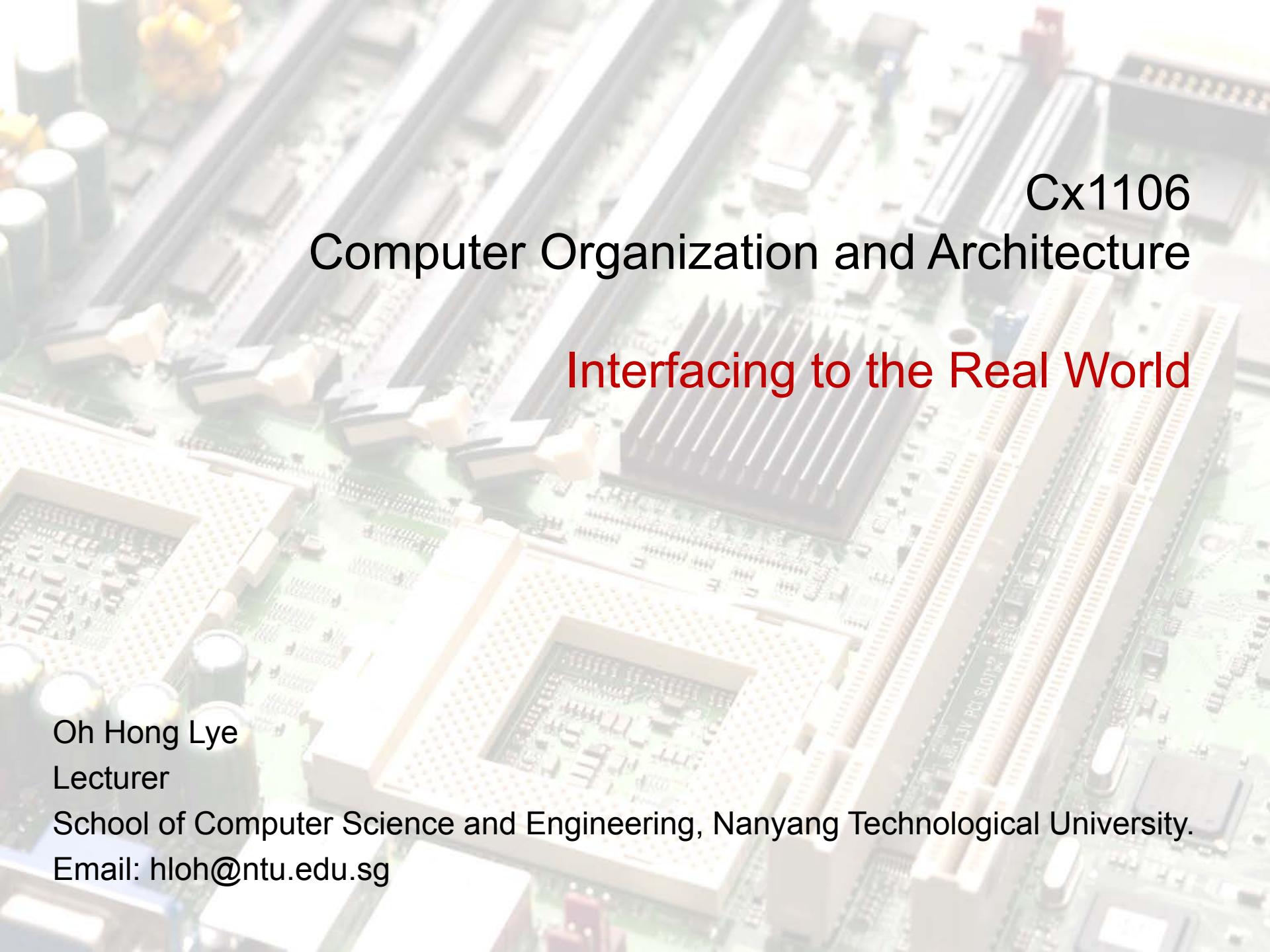
School of Computer Science and Engineering, Nanyang Technological University.

Email: hloh@ntu.edu.sg

Signal Chain Sub-System



- **Signal Chain Sub-System**
 - ADC-DAC
 - Parallel and Serial (UART/SPI) Digital Interfaces
 - Interrupt Controller
 - Direct Memory Access Controller



Cx1106

Computer Organization and Architecture

Interfacing to the Real World

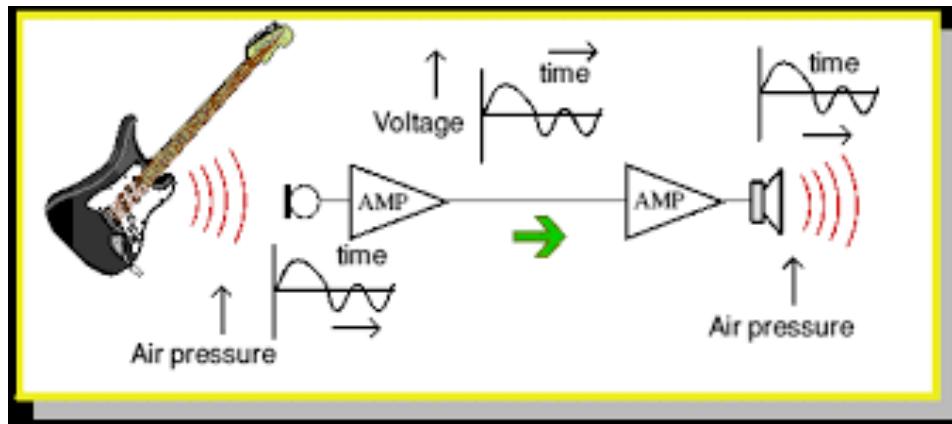
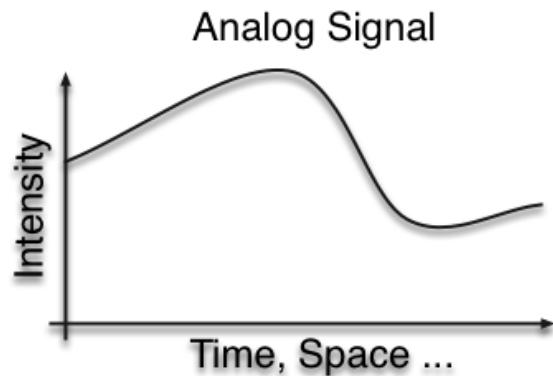
Oh Hong Lye

Lecturer

School of Computer Science and Engineering, Nanyang Technological University.

Email: hloh@ntu.edu.sg

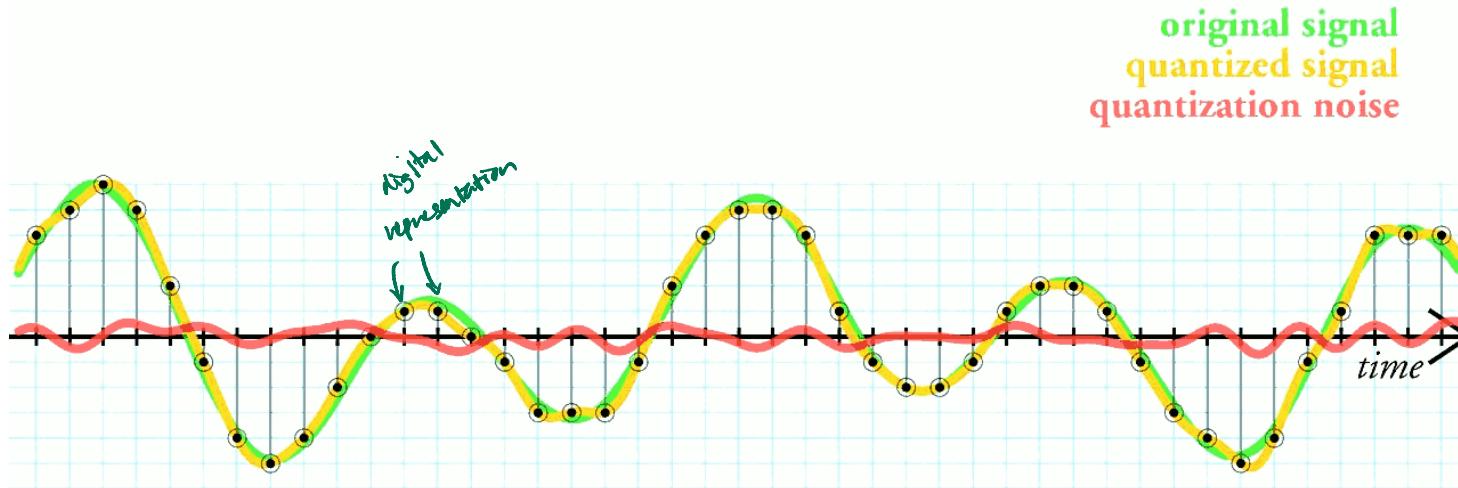
Analog and Digital Signal



- 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.**
↳ can introduce software coding
operation in digital domain more defined, e.g. 0 or 1 compare to analog
 - Analog signal has **continuous voltage level** and are **continuous in time domain.**

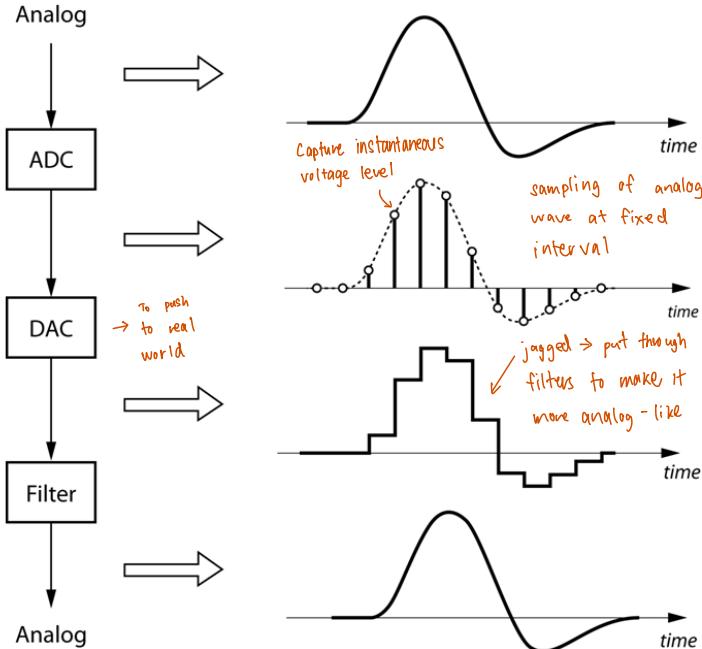
Analog continuous by nature,
be it time / space wise
or intensity wise

Digital Signal



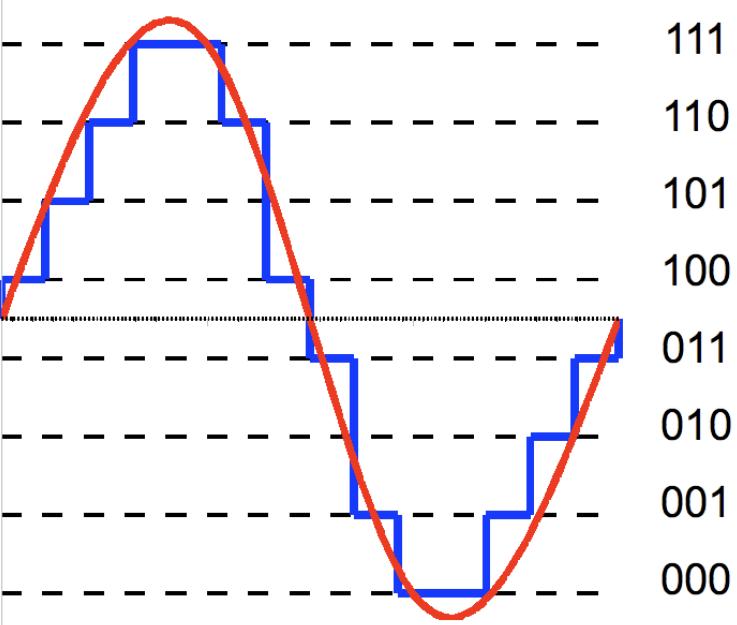
- Digital signals are **discrete time representation** of analog signal.
- Obtained through process of **digitization**, commonly known as **Analog to Digital Conversion**.
- Analog signals are digitized to its digital equivalent using a **Analog-to-Digital Converter (ADC)**.
Digital signal obtained; passed to processor
- 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**). *e.g. signal @ 1K hertz, sample @ 2K hertz per sec*
- Sampling frequency needs to be at least twice the signal frequency (**Nyquist theorem**). *from instantaneous voltage level*
- Typically, the analog signal level is assigned to the **nearest discrete voltage level allowed** in that particular digitization process.
- Analog signal can be reconstructed back by applying a filter on the digital signal.

Digital Quantization

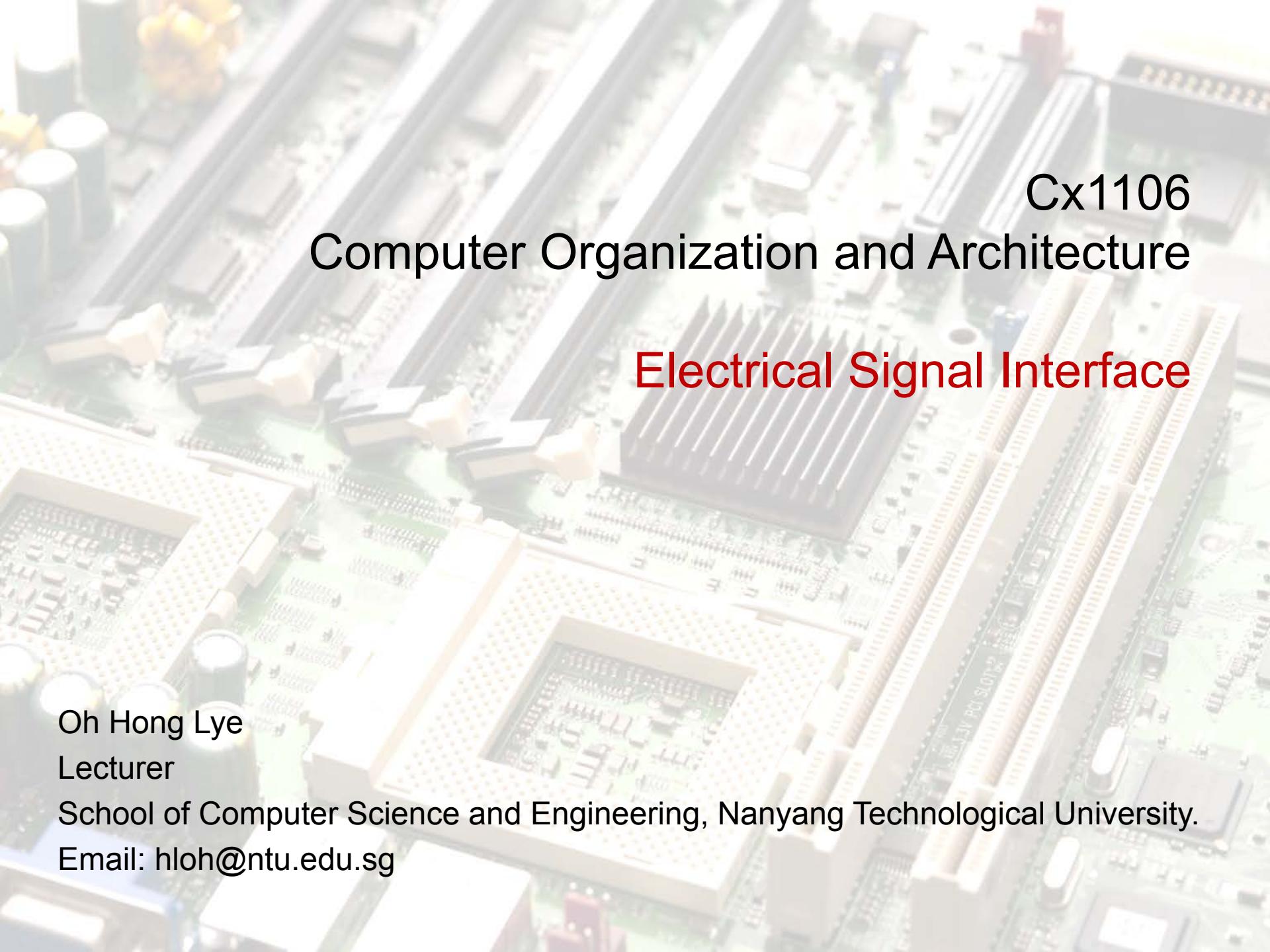


- Figure on the left shows a typical digital quantization process of a analog signal.
- A **3-bit system** is used in this example. So all signal will be mapped to one of the **8 possible representations (000 to 111)**.
- At each sampling point, the analog signal level is approximated to the nearest digital equivalent.

Signal Chain between Processor and Real World



- 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**.



Cx1106

Computer Organization and Architecture

Electrical Signal Interface

Oh Hong Lye

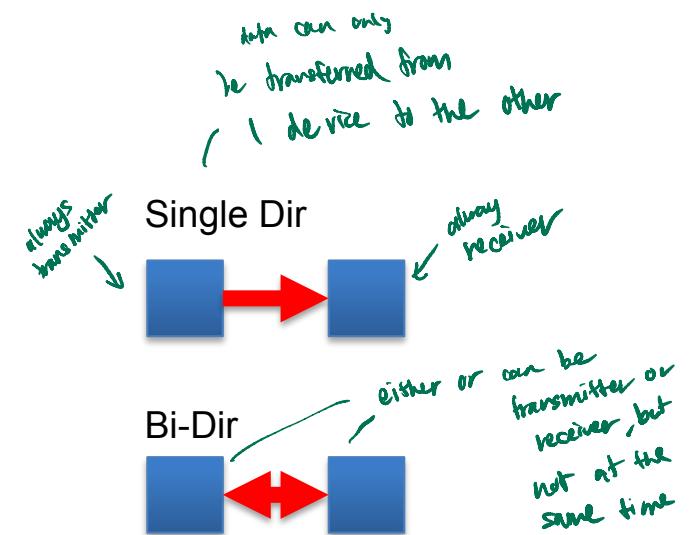
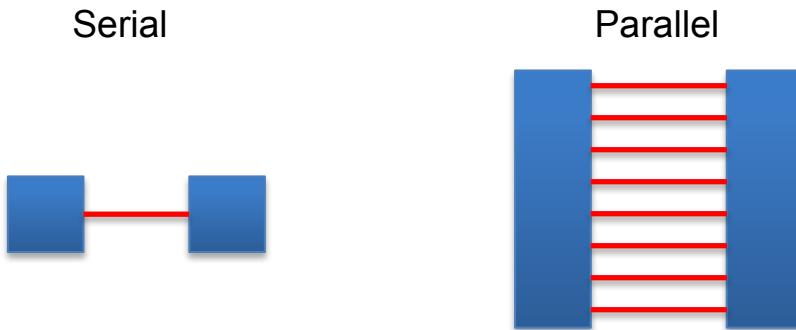
Lecturer

School of Computer Science and Engineering, Nanyang Technological University.

Email: hloh@ntu.edu.sg

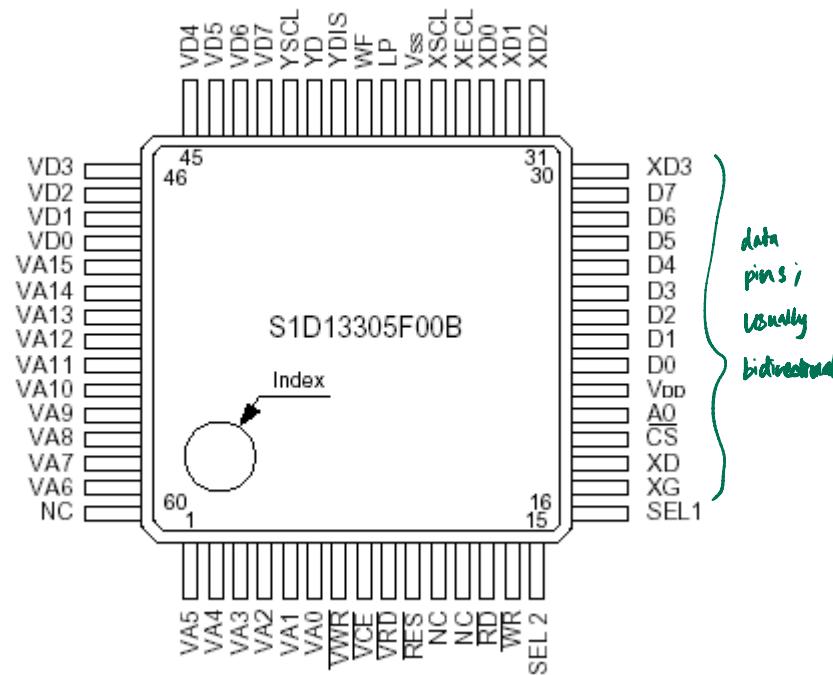
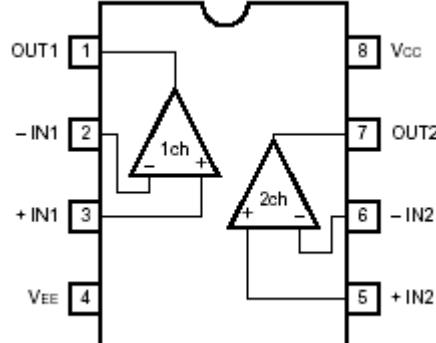
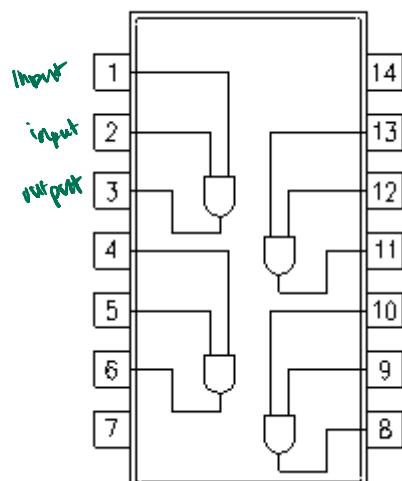
Interface

- A **boundary** where two or more devices meet to exchange information. Some modes of connection below
 - Single-bit Data transfer (**Serial**)
 - Multiple-bits Data transfer (**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.



Interface Compatibility

- Interfacing one electronic device to another requires compatibility in
 - Electrical signal level – have to be compatible
 - Communication protocol (Handshaking and Data signals).
 ↳ have to be the same

Electrical Signal level (Safety)

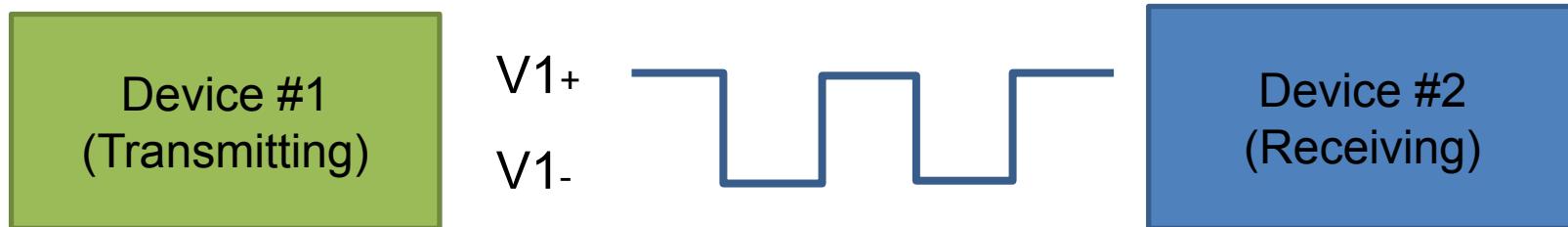
- Primary consideration when connecting two electrical device together.
- Ensure that the **output voltage** level of output device **do not exceed** the **maximum allowable input voltage** level of the input device.
- Electronic devices will either get ‘fried’ i.e. **spoilt** or have its **reliability 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)

have to be compatible from device to device

- 5V => Logic '1' or '0' ?
- Four parameters
 - **VOH**.
 - Transmitting a Logic '1' yield a signal level of \geq VOH
 - **VOL**.
 - Transmitting a Logic '0' yield a signal level of \leq VOL
 - **VIH**.
 - A received signal with level \geq VIH will be recognized as a Logic '1'
 - **VIL**.
 - A received signal with level \leq VIL will be recognized as a Logic '0'

Electrical Signal Level (Digital Data Transfer)

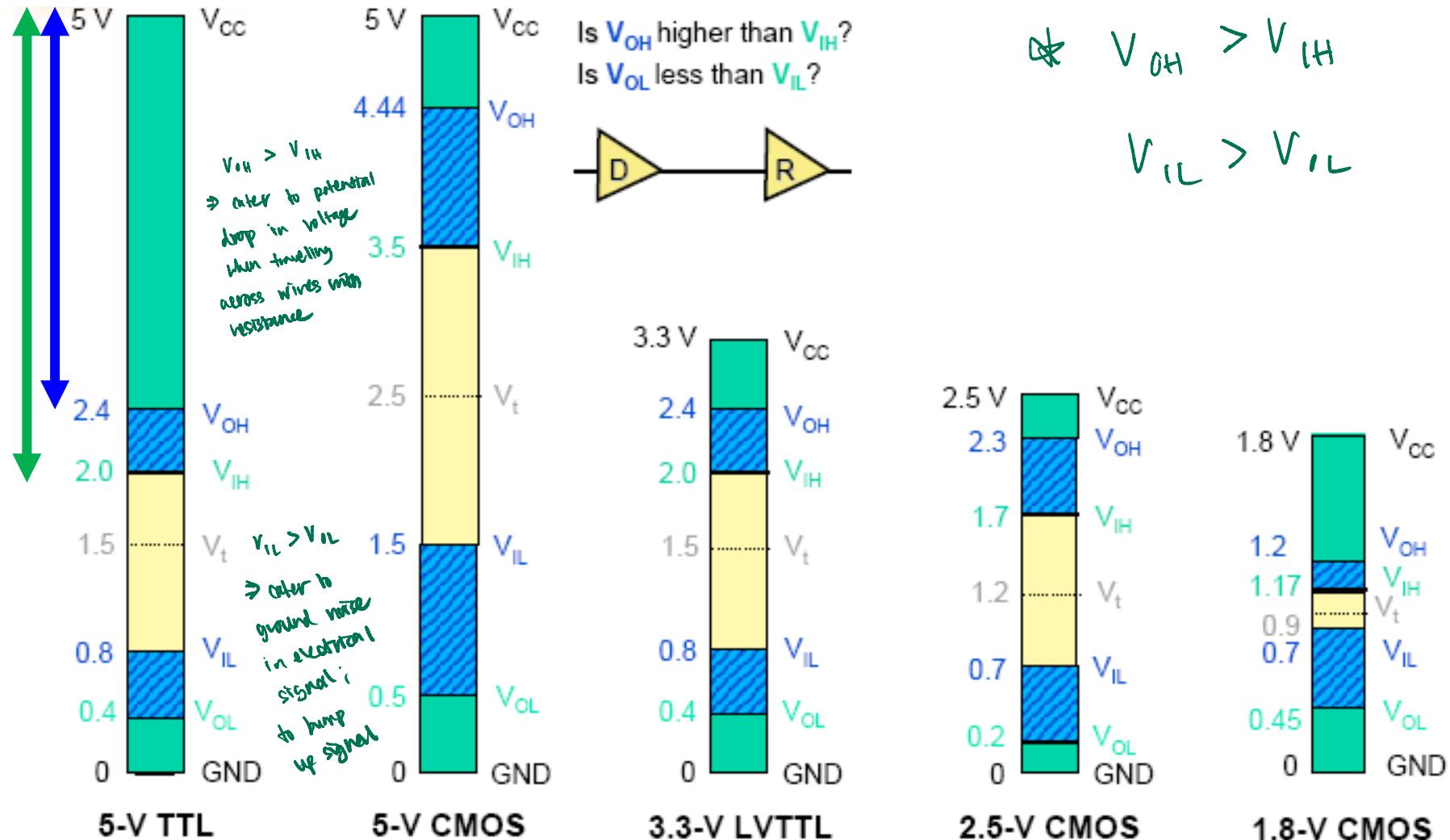


- Device #1
 - Output Voltage level for logic '1' = $V1+$ (e.g. 5V)
For typical value of $V1+$, look for $V(OH)$ parameter in device datasheets.
 - Output Voltage level for logic '0' = $V1-$ (e.g. 0V)
For typical value of $V1-$, look for $V(OL)$ parameter in device datasheets.
- Device #2
 - Min Input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

min. voltage device outputs for output logic 1
max. voltage device outputs for output logic 0

KEY:
 $V_{OH} \geq V_{IH}$
 $V_{OL} \leq V_{IL}$

Electrical Signal Level Standards

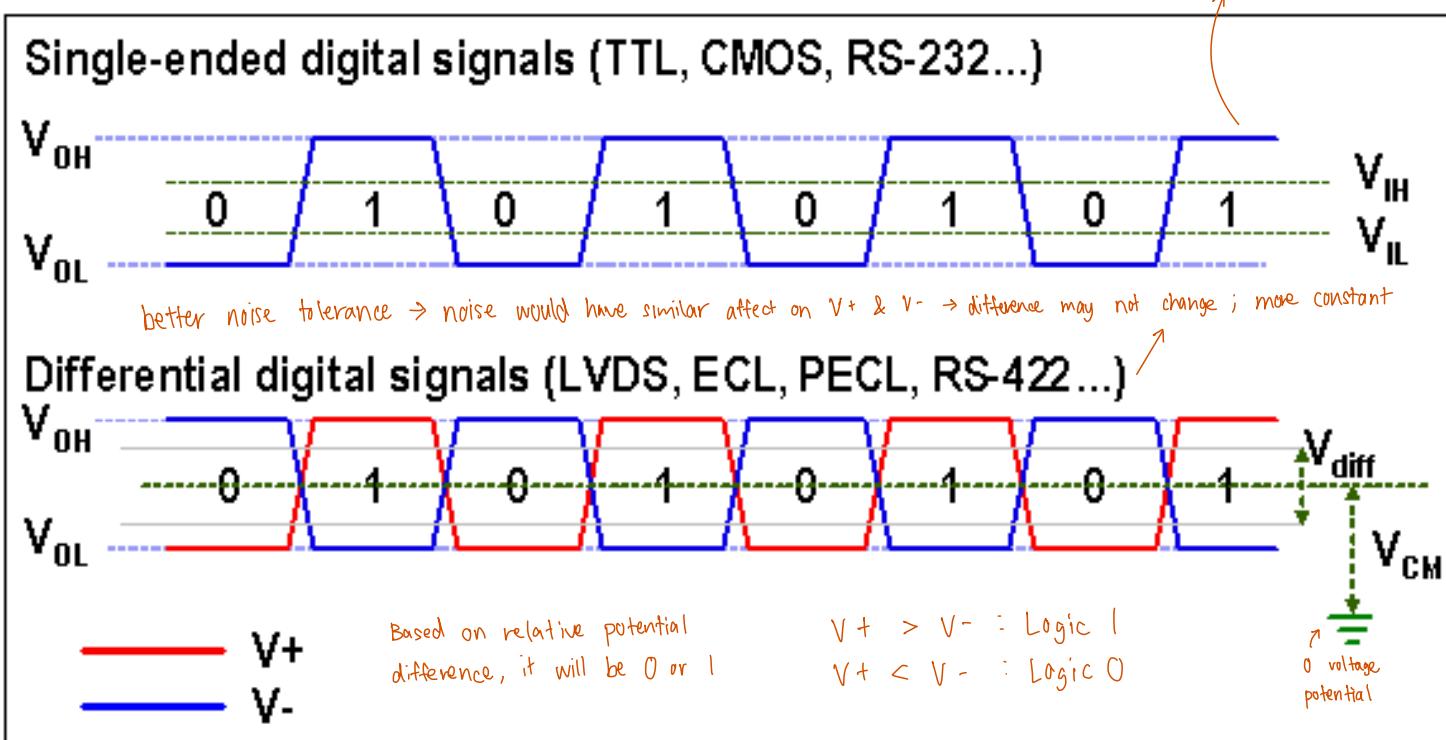


Source: http://denethor.wlu.ca/pc200/logic_charac/voltage_char.png

Differential Signals

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

one line used to transfer one bit of data; logic 1 or 0 with reference voltage of line along with respect to ground



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

Communication Protocols

- Communication Protocols refers to how the data are **formatted** during transmission.
- Some examples
 - Number of bits in a transmission frame
 - What synchronization to use
 - Data width
 - Types of data and its formatting
- Examples of communication protocols are USB, UART, SPI etc.

cannot
be used
together

Protocol 1

100**1001001001001000**

Blue: Synchronization Bits

Protocol 2

100**1001001001001000**

Green: Data Bits

Red: Error Correction Bits

Protocols

English

AIR

FART

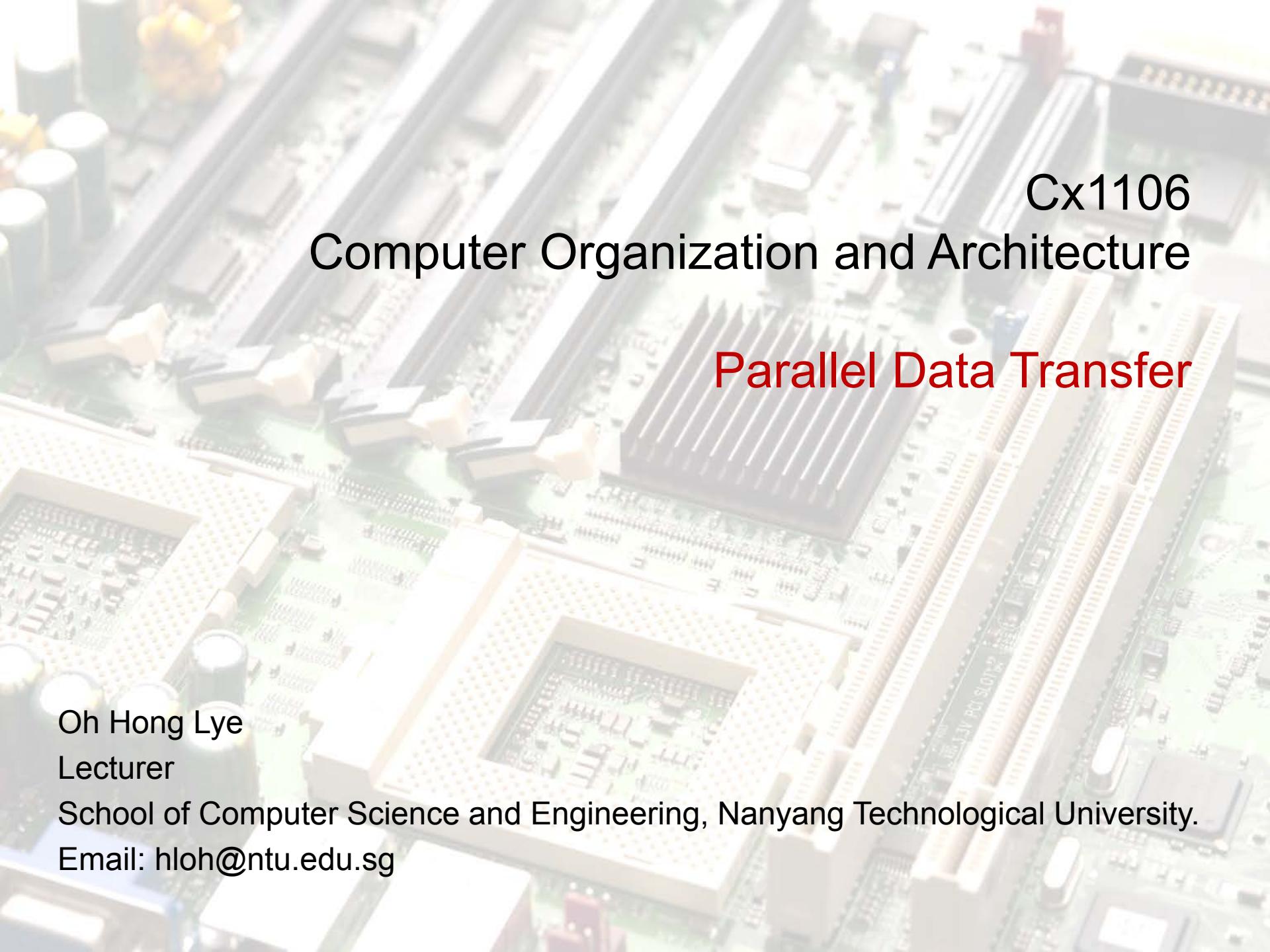
CRAP

Malay:
Water

Swedish:
Speed



Romanian:
Carp



Cx1106

Computer Organization and Architecture

Parallel Data Transfer

Oh Hong Lye

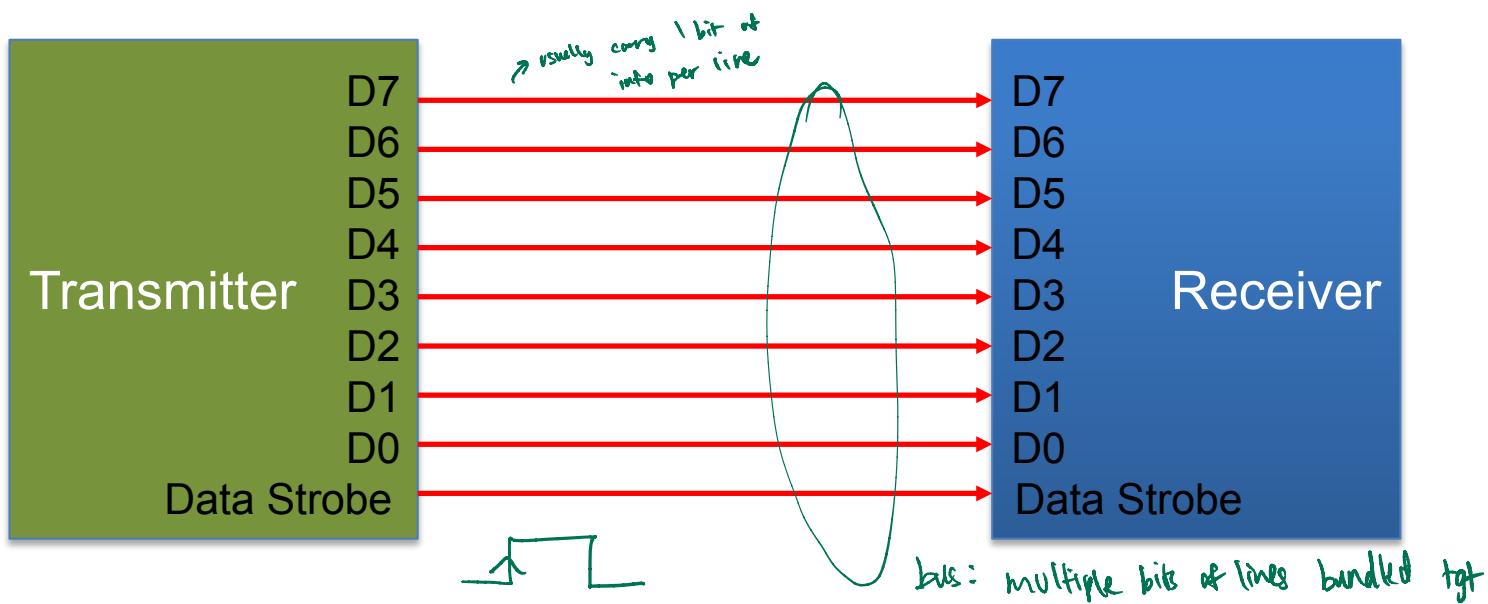
Lecturer

School of Computer Science and Engineering, Nanyang Technological University.

Email: hloh@ntu.edu.sg

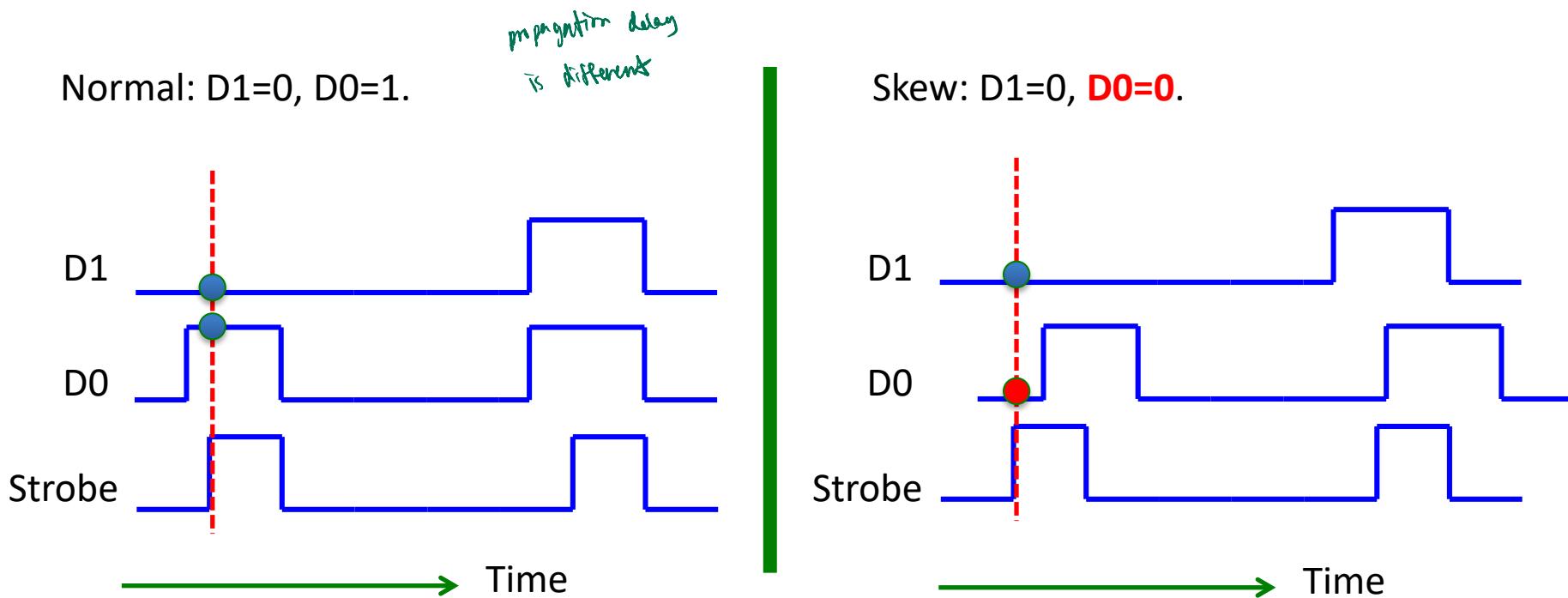
Parallel Data Transfer

- 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).



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.



large RC \rightarrow takes a longer time for electrical signal to transition from 0 or 1 vice versa

Signal Skew – Contributing Factors

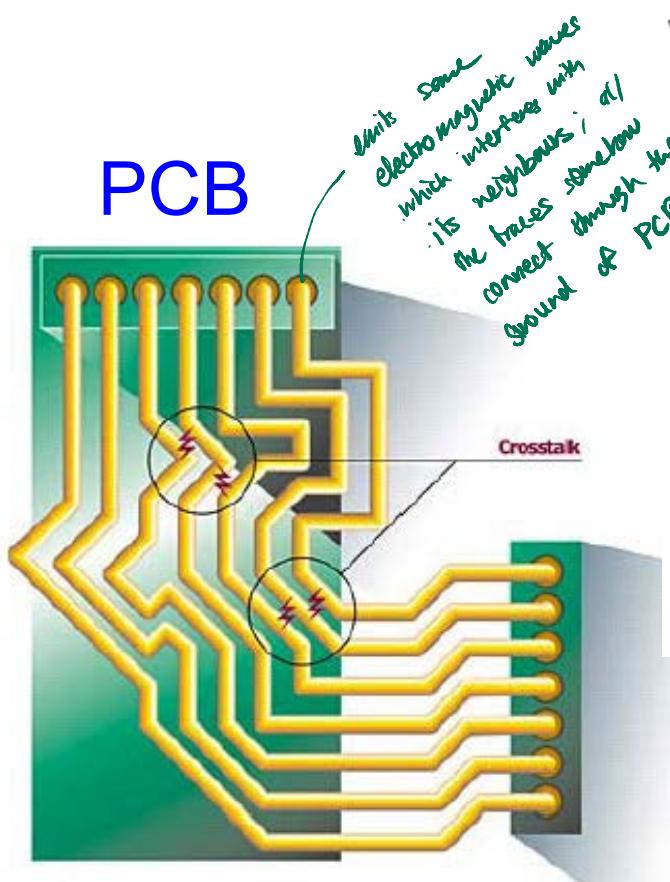
- Signal Skew is due to variation in propagation delay between signals from the same data bus.
- Propagation delay is the time taken for signal to travel between two points in a circuit.
- Capacitance and Resistance of the physical data line is a major contributor to circuit propagation delay.
 - The 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.

electrical wires

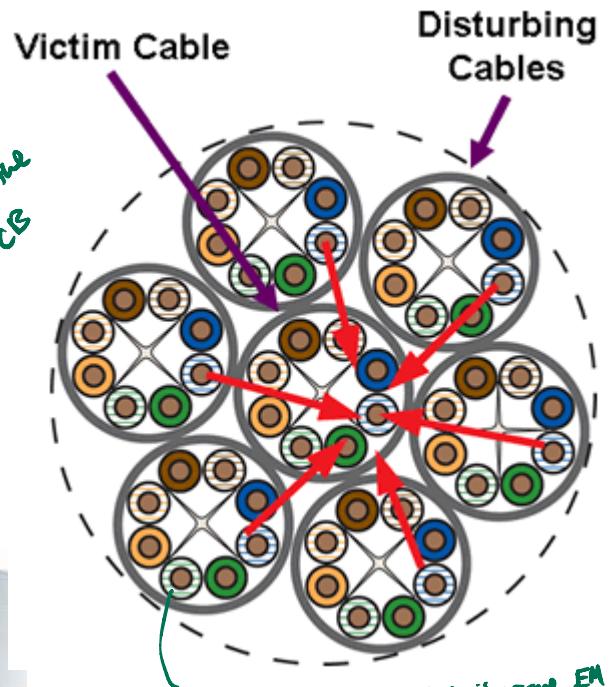
Crosstalk

- interference in the signal transmission

Party



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).

limitations signal skew & crosstalk place on system design?

- Max. no. of lines on data bus
- Max. clk rate
- Max. no. of bytes that can be transferred in one block
- Min. delay between each byte transfer

Parallel Data Transfer – Pros and Cons

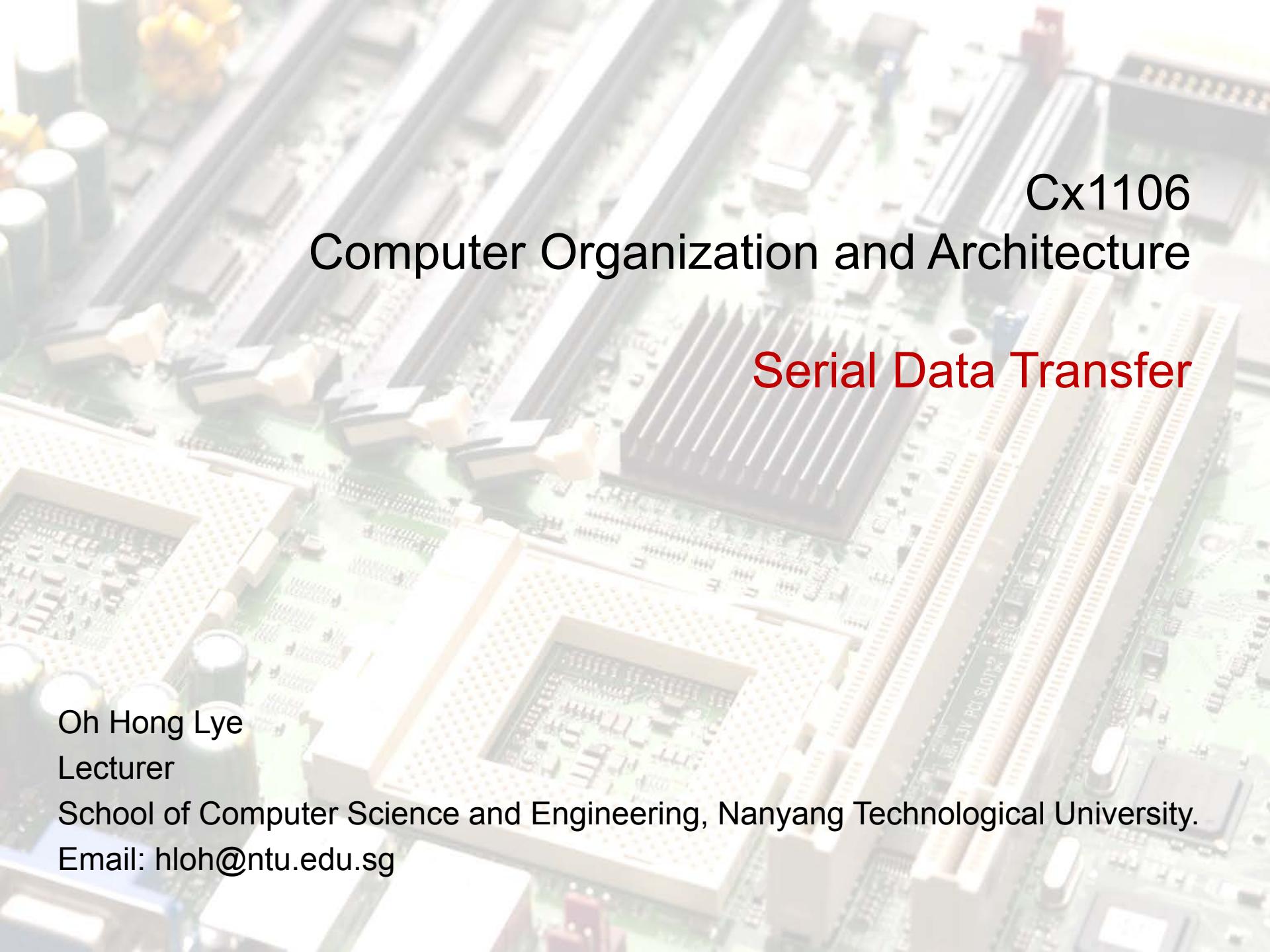
- Advantages

- 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.

/ it same clk rate comparing to serial

- 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.



Cx1106

Computer Organization and Architecture

Serial Data Transfer

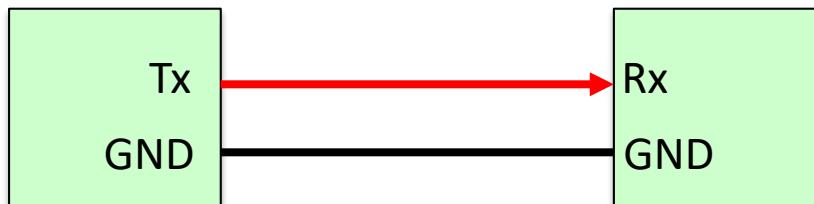
Oh Hong Lye

Lecturer

School of Computer Science and Engineering, Nanyang Technological University.

Email: hloh@ntu.edu.sg

Serial Data Transfer

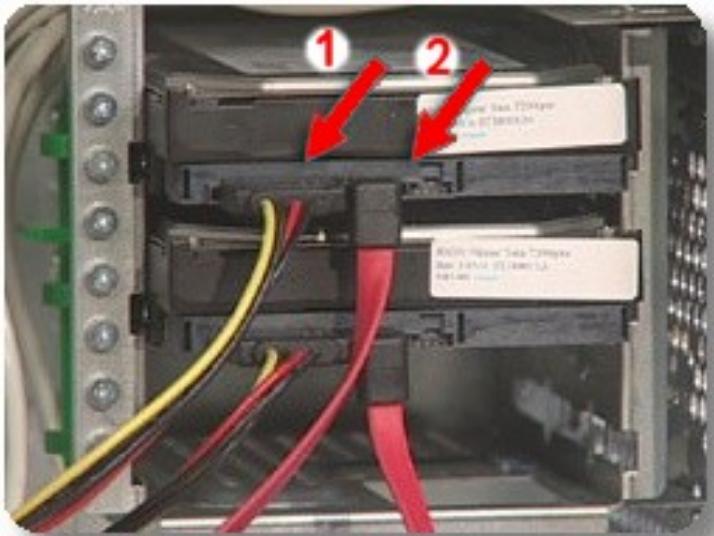
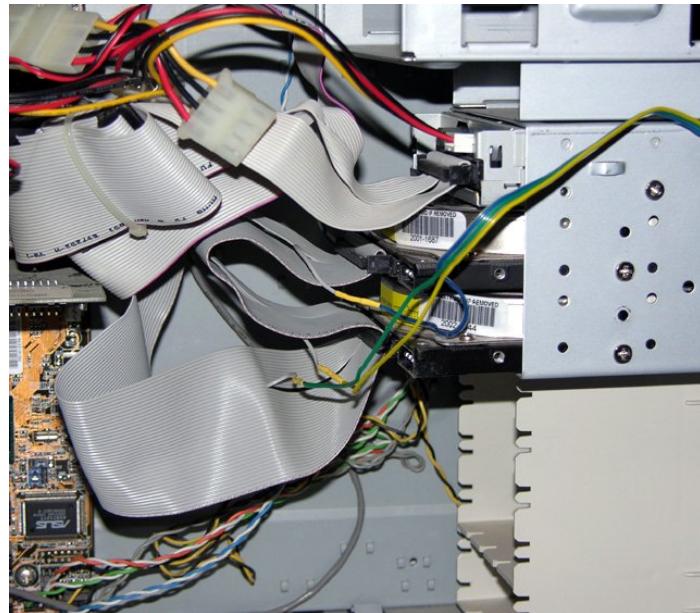


- Data is transferred **one bit at a time** over a single data line. Comparatively, parallel data interface transfer multiple bits simultaneously.
- Less affected by **signal skew and crosstalk** because there are less electrical wires involved compared to parallel data transfer. Hence, able to support 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

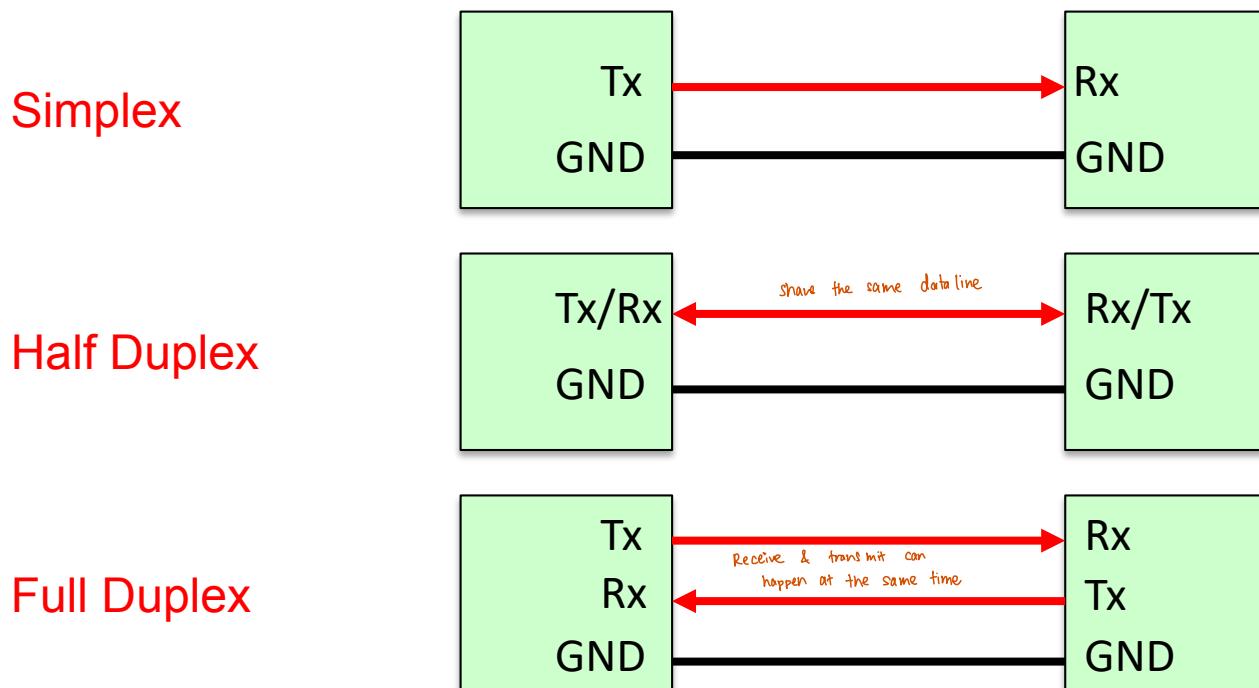
- Advantage
 - Less affected by signal skew and crosstalk because there are less electrical wires involved compared to parallel data transfer. Hence, able to support higher frequency clocking.
 - Able to transfer data reliably over a longer distance.
 - Lower cost since less wires and connectors are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

Parallel and Serial Comparison

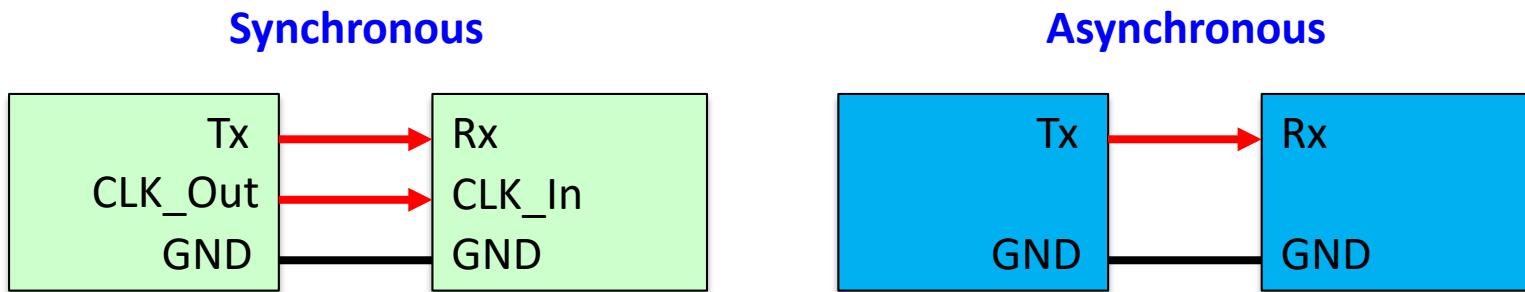


Serial Data Transfer Mode

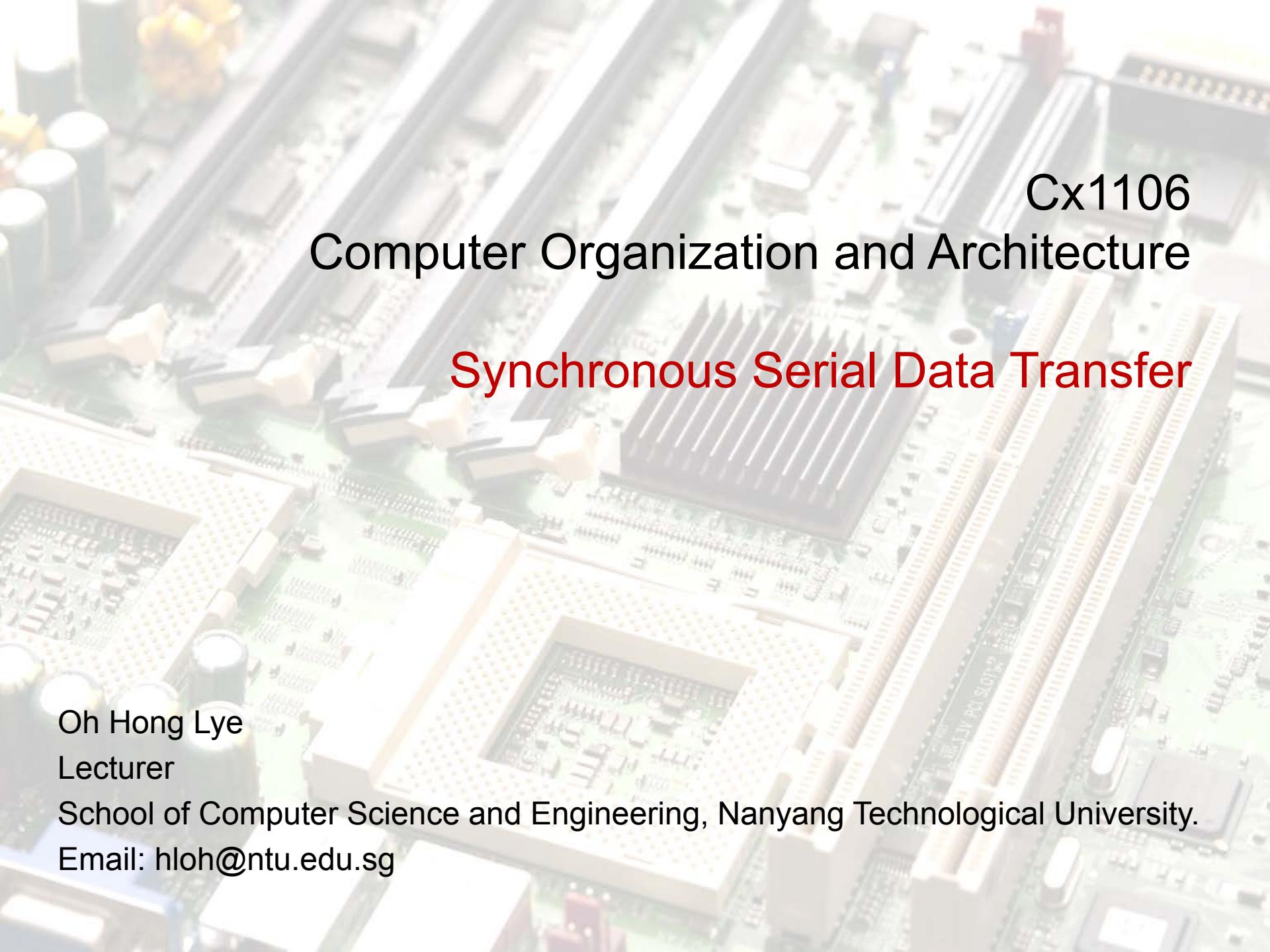
- **Simplex**: Data transfer in one direction only.
- **Half-Duplex**: Data transfer in both direction, but RX and TX is mutually exclusive. → at any point in time, device A can be only either transmitter or receiver
- **Full Duplex**: Simultaneous RX and TX



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 synchronize 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.



Cx1106

Computer Organization and Architecture

Synchronous Serial Data Transfer

Oh Hong Lye

Lecturer

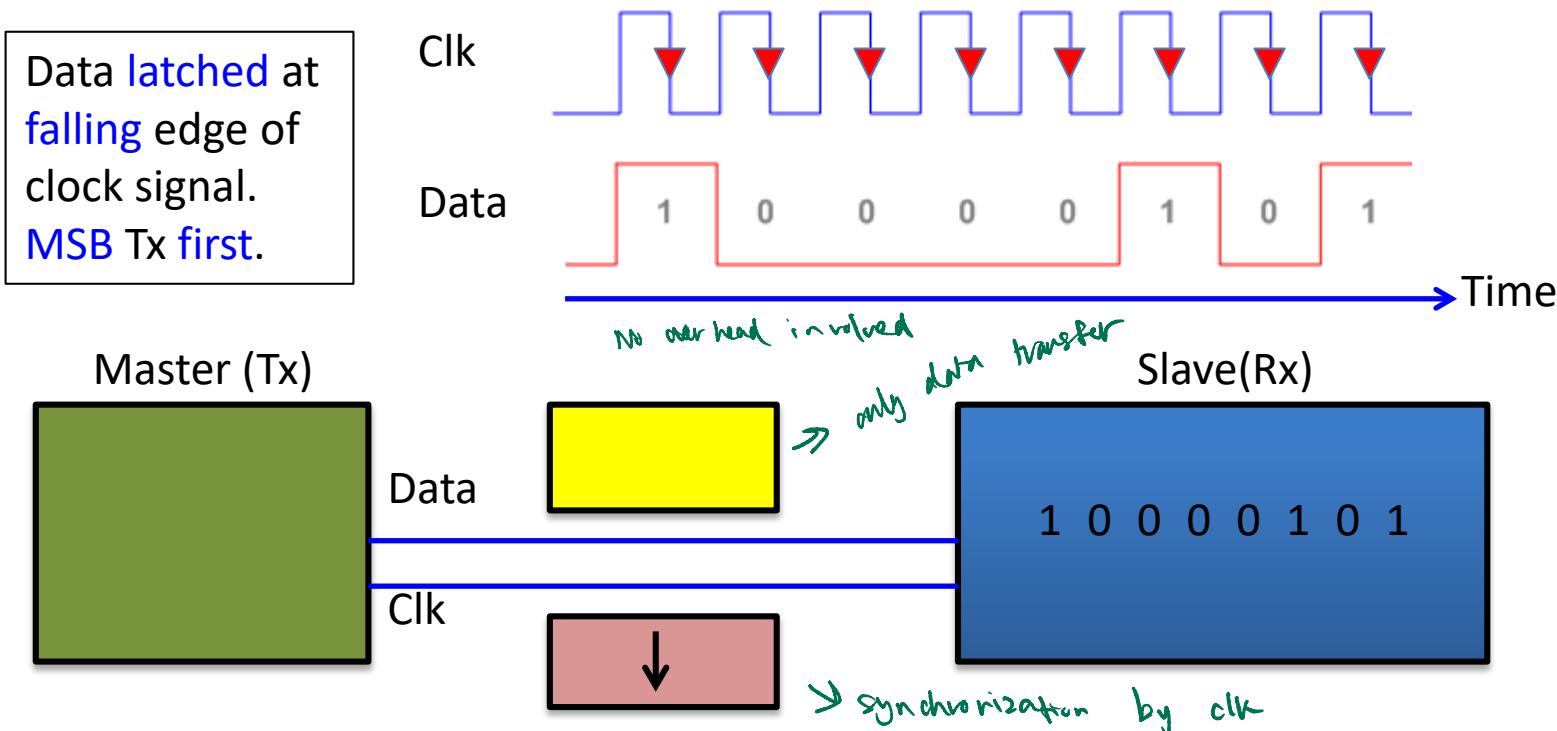
School of Computer Science and Engineering, Nanyang Technological University.

Email: hloh@ntu.edu.sg

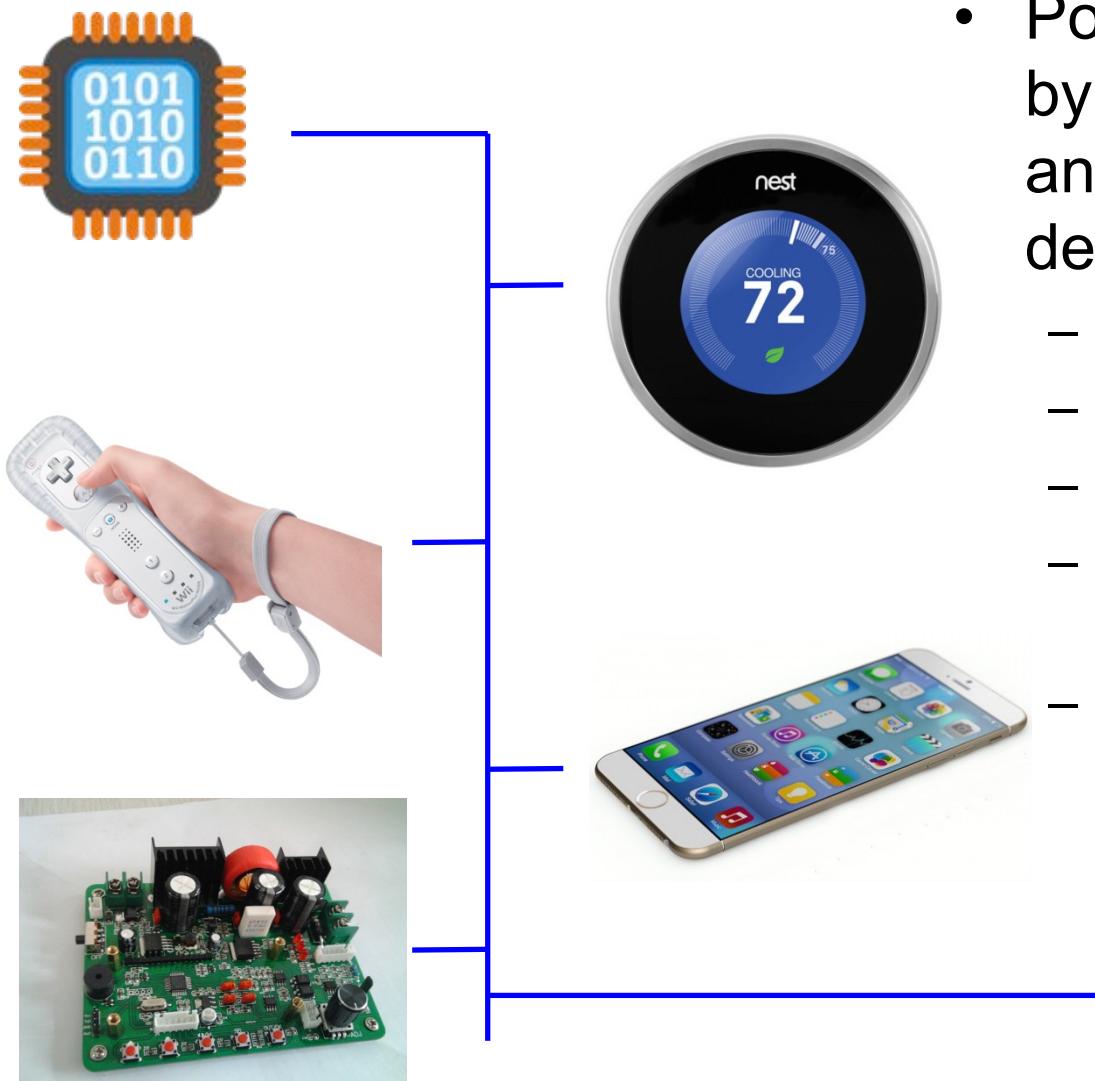
Synchronous Serial Transfer

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

can have
≥ 1 slave



I2C and SPI Bus

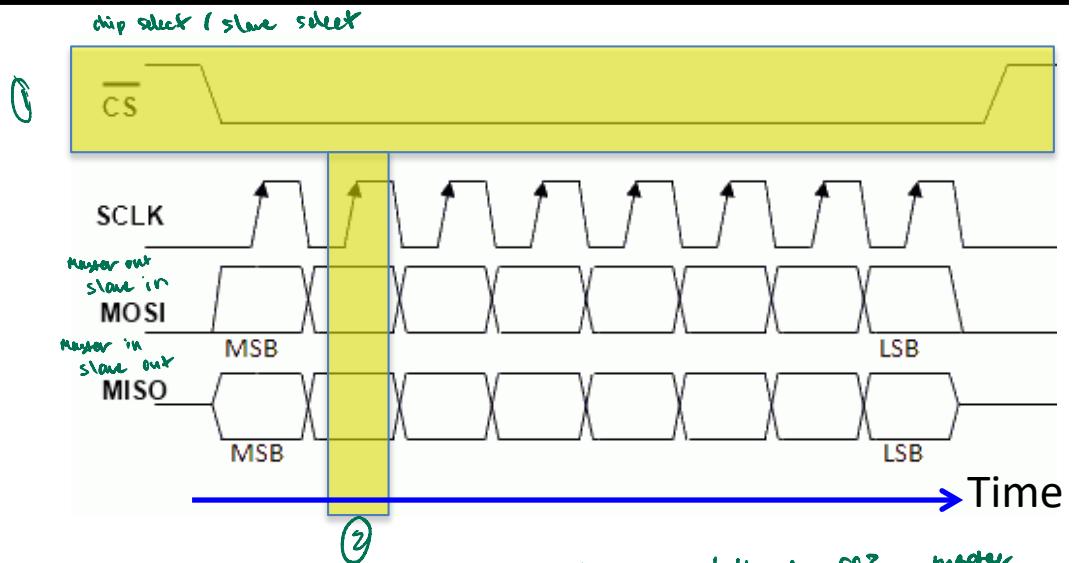
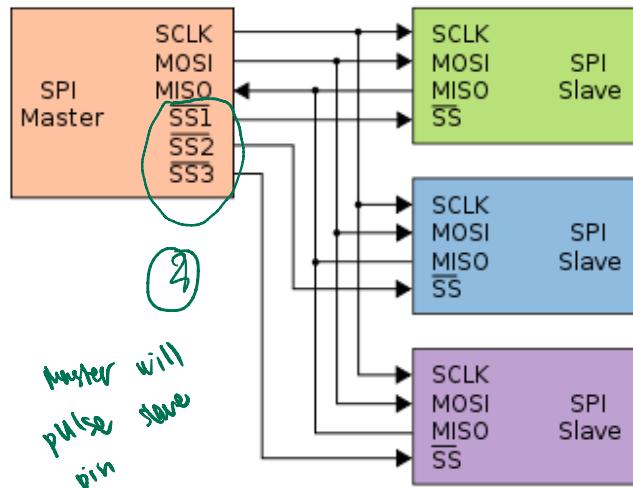


- Popular serial buses used by processor to transfer data and control many peripheral devices.
 - Accelerometers
 - Temperature Sensors
 - Touch Screen Controllers
 - Power Supply Modules Configuration
 - Audio/Video Codecs Configuration



I2C will not be covered in this module

Serial Peripheral Interface (SPI) Bus

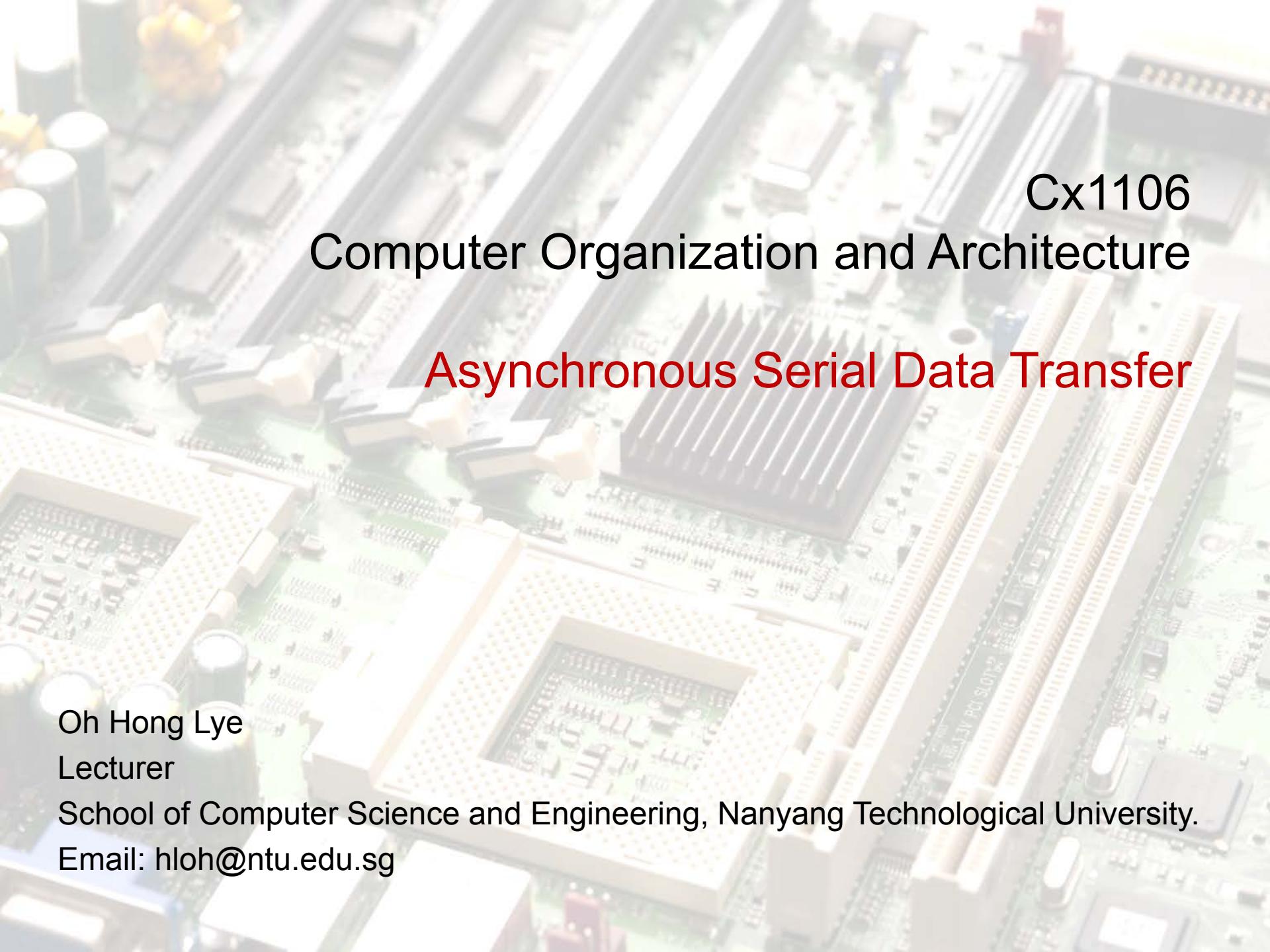


- ① • To **start** the transfer
 - Slave Select (SS) has to be pulled **Low**.
- ② • **Data transfer**
 - Data on MOSI and MISO **latched** in on **rising/falling** clock edge (configurable)
 - MOSI: Master-Out-Slave-In (Master Output, Slave Input)
 - MISO: Master-In-Slave-Out (Slave Output, Master Input)
- ③ • Allow **multiple slaves** via use of multiple Slave Select.

e.g. if wanna talk to SS3, master will pull SS3 line low and SS1 & SS2 remains high.

SS1 & SS2 disable & SS3 enabled to obtain data

clk frequency & transfer rate
⇒ 1 bit of data transmitted at one clock edge
⇒ e.g. clk : 1 MHz
transfer: 1 Mbit per second



Cx1106

Computer Organization and Architecture

Asynchronous Serial Data Transfer

Oh Hong Lye

Lecturer

School of Computer Science and Engineering, Nanyang Technological University.

Email: hloh@ntu.edu.sg

Asynchronous Serial Transfer

transferred by frame;
every frame consists
of data bits + some
overhead

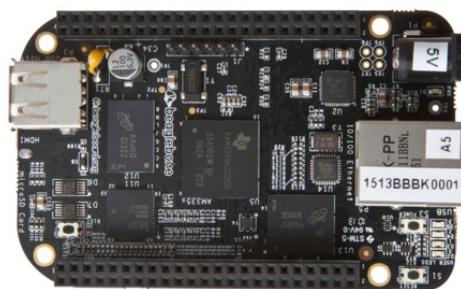
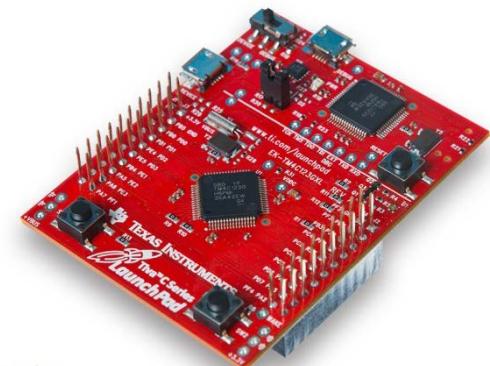
- No common clock is provided between transmitter and receiver.
- Prior to the transmission, the receiver needs to know the transmitting clock rate and the number of bits that are to be transferred with each data packet.
need some form of Synchronization
- Special SYNC words are used to indicate START/STOP condition.
- Upon receiving the START SYNC Word, the receiver then use its own local clock to track the timing.
may run at slightly diff rate
- 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 synchronize its clock to the transmitter clock (helps to reduce clock skew between the two clocks).
*↓ transmitter, at every clock edge,
will send a synchronize bit;
receiver receives and knows it is
the exact time that transmitter clk transition from 1 to 0 or vice versa*

keep frame size small,
clk become
in significant in UART

*receiver adjust internally
its own local clk to
align edge ↑*

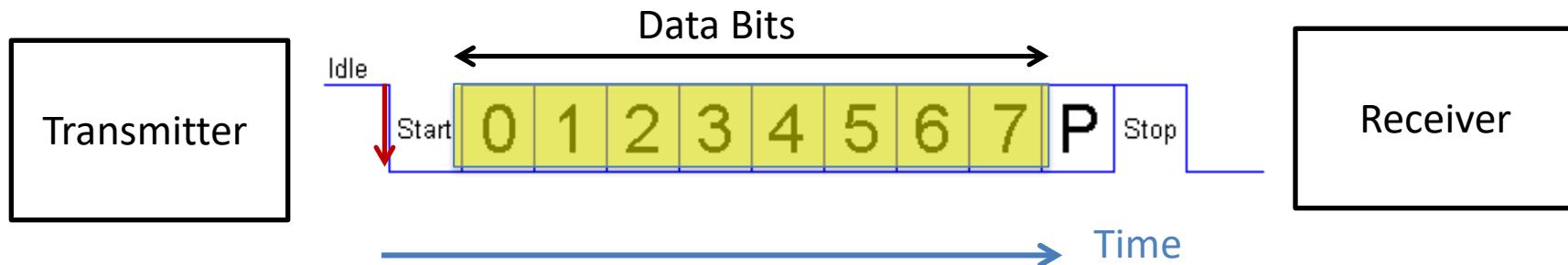
Universal Asynchronous Receiver Transmitter (UART)

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



UART Transmit

- During **IDLE** State, the data line is in ‘powered’ state i.e. **Logic ‘1’**.
signals initiation of transmission
- 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.
↓ error detection
- A **STOP** bit (**Logic ‘1’**) terminates the transmission.



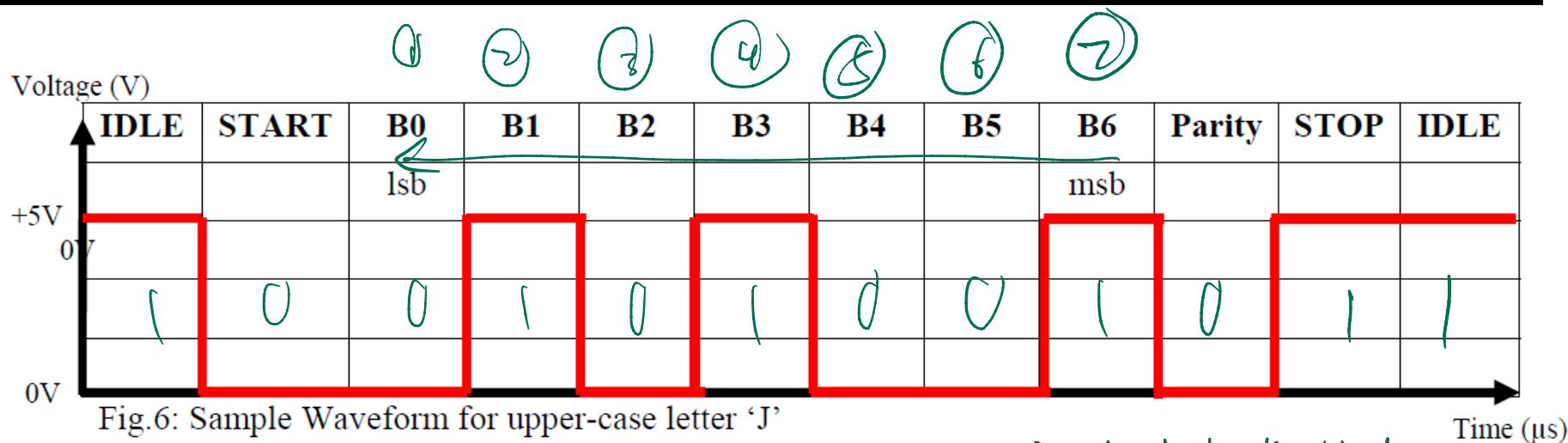
Parity Bit

- If parity scheme is enabled, the receiver will also sample the parity bit and checks for parity error.
- If even parity scheme is used, there should be an even number of '1's in the data field and the parity field.
- Hence, if there is an odd number of '1's in the data field, then the parity bit transmitted should be a '1' to make the total even.
- If receiver receives odd number of '1's in a even parity scheme, it'll flag a Parity Error, meaning one or more bits in the transmission is wrong.
- Vice versa for odd parity scheme. *↑ should be logic 1; receiver know where it is*
- The receiver then samples the STOP bit(s), if a '0' is detected instead of '1', then receiver will flag a Framing Error.

*error handling depends on processor
→ most likely discard the packet & get it retransmitted*

for error detection, NO error correction

UART Tx Example

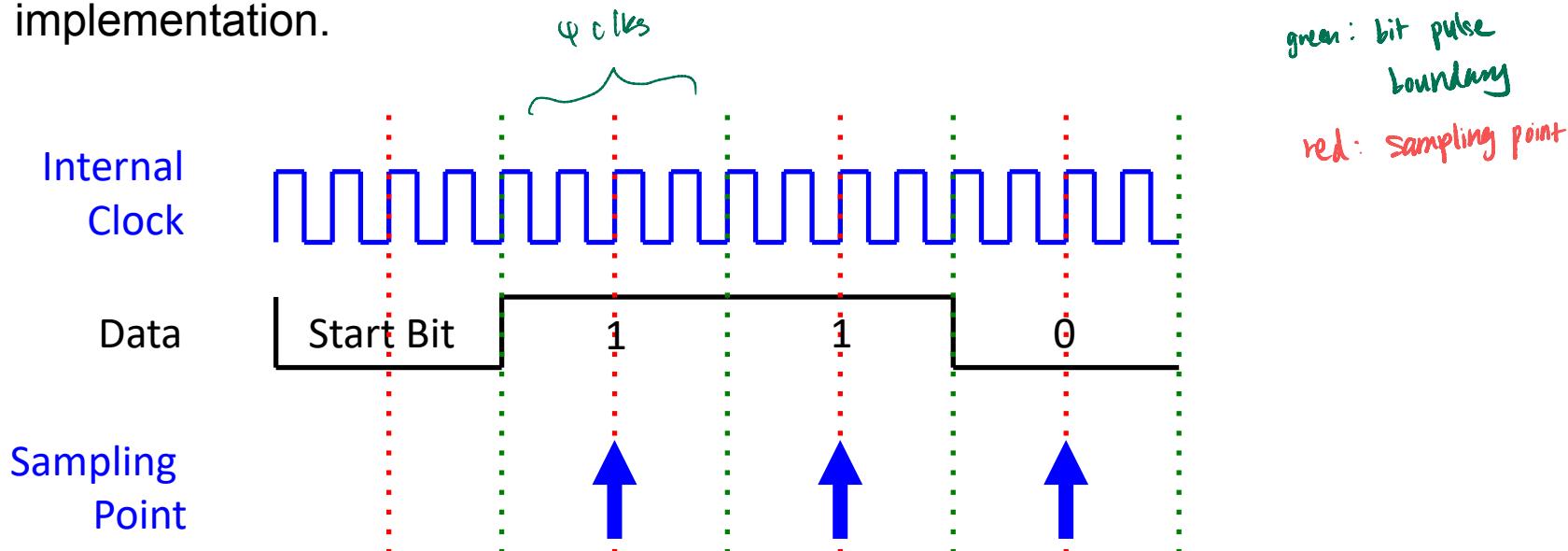


- Figure 6 above correspond to the configuration 701 (7 Data, 1 STOP and Odd parity).
- Capital Letter 'J' => Ascii value = 0x4A
- 0x4A => 100 1010 (binary)
- Presented in LSB first => 0101 001
- IDLE=1, START=0, STOP=1.
- Parity bit logic depends on number of 1's in Data Field and the Parity Scheme (Odd or Even) used.

↑
data
↑
stop bit
↑
odd parity

UART Receive

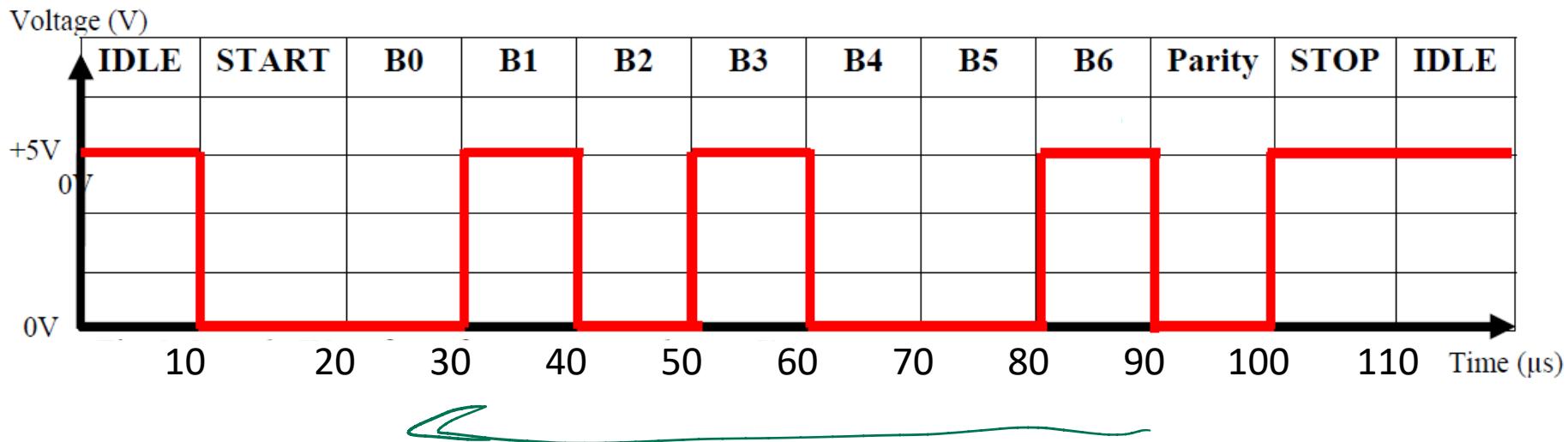
- Receiver monitors the Data line for the **Start Bit**. In real world design, the falling edge on the data line will trigger an interrupt in the microprocessor to start the receiving process.
- Needs to know the **baud rate** in order to **sample the data bits correctly**.
- **Internal clock** of the 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.
- 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.



UART Rx Example

Important to use same configuration & baud rate at receiver & transmitter

Tx baud rate = $1/(10 \text{ us}) = 100000 \text{ bps}$. Configuration = 7O1.



- RX Data = 1001010b
- RX Data = 0011000b ?
- Information Sampled @ 200000 bps:

00001100110000110011

wmt reflet
error

Samples above
since

$$\frac{1}{200000} = 5 \mu\text{s}$$

Sampling at

possible if there is a mismatch between transmitter baud rate & into receiver used to receive UART transmitter \Rightarrow receiver thought transmitter is transmitting at 200 000 bits/s instead of 100 000 bit/s

Which of the signals below is not analog in nature?

- A. Input signal to Digital-to-Analog Converter
- B. Microwave radiation
- C. Audio from headphone
- D. Disco strobe light

B, C and D are real world signals: electromagnetic waves, sound, light, all of which are analog in nature. Digital-to-Analog converter (DAC) converts digital data to its analog equivalency, hence its input are digital in nature.

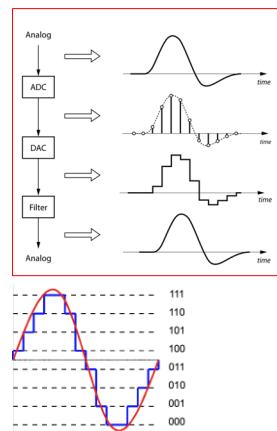
Cx1106

2

**Which of the following procedure is carried out when digitizing an analog signal?

- A. Sample the voltage level of the analog wave form at fixed time interval.
- B. Limit the voltage range of the analog waveform to +/- 5V
- C. Match the voltage level of the analog waveform to the closest level available in the digital domain.
- D. Reduce the analog waveform voltage level to a range used by the digital domain

Forcefully reducing the analog voltage level will distort the analog waveform, hence it is not carried out when digitizing the analog signal.

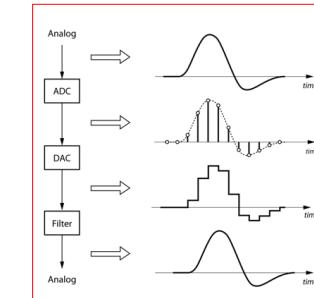


Cx1106

4

Which of the following sequence is used for processing real world data?

- A. ADC->Filter->Digital Processor->DAC
- B. ADC->Digital Processor-> Filter->DAC
- C. DAC->Filter->Digital Processor->ADC
- D. ADC-> Digital Processor->DAC->Filter



Cx1106

3

Which is a more accurate representation of a real world signal?

- A. Analog signal
- B. Digital signal sampled at 1 sec interval
- C. Digital signal sampled at 0.1 sec interval
- D. Digital signal sampled at 0.001 sec interval

Analog signal is equivalent to a digital signal with infinitely small sampling interval.

Cx1106

5

What is the minimum sampling rate required in order for a digital signal to be able to represent the analog equivalent adequately?

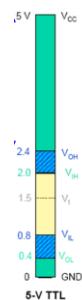
- A. 1X the frequency of the analog signal
- B. 2X the frequency of the analog signal
- C. 3X the frequency of the analog signal
- D. As fast as the system can support

Nyquist Theorem

6

Which conditions are needed in order for the input device (Din) to sense the logic of the output device(Dout) correctly?

- A. $V(OH)$ of Dout > $V(IH)$ of Din, $V(OL)$ of Dout > $V(IL)$ of Din
- B. $V(OH)$ of Dout < $V(IH)$ of Din, $V(OL)$ of Dout < $V(IL)$ of Din
- C. $V(OH)$ of Dout < $V(IH)$ of Din, $V(OL)$ of Dout > $V(IL)$ of Din
- D. $V(OH)$ of Dout > $V(IH)$ of Din, $V(OL)$ of Dout < $V(IL)$ of Din



$VOH \Rightarrow$ device's min output voltage level if it is outputting a logic '1'
 $VIH \Rightarrow$ the min voltage level at its input pin in order of the device to consider the signal as a Logic '1'.

8

Which of the following connection may result in compatibility issue?

- A. A 5V device input pin to 5V device bi-directional pin
- B. A 3V device output pin to a 3V device input pin
- C. A 3V device output pin to a 5V device input pin
- D. A 3V device ground pin and a 5V device ground pin

$$\begin{aligned} VOH &\geq VIH \\ VOL &\leq VIL \end{aligned}$$

(C) Electrical Signal level is not compatible, this may result in VOH/VIH , VOL/VIL compatibility issues.

7

	A			UNIT
	MIN	NOM	MAX	
VCC	Supply voltage	4.5	5	5.5
VIH	High-level input voltage	2		V
VIL	Low-level input voltage		0.8	V
IOH	High-level output current		-0.4	mA

PARAMETER	TEST CONDITIONS‡			UNIT
	MIN	TYP	MAX	
VIK	$V_{CC} = \text{MIN}$, $I_I = -12 \text{ mA}$		-1.5	V
VOH	$V_{CC} = \text{MIN}$, $V_{IL} = 0.8 \text{ V}$, $I_{OH} = -0.4 \text{ mA}$	2.4	3.4	V
VOL	$V_{CC} = \text{MIN}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 16 \text{ mA}$	0.2	0.4	V

Is the electrical signal level of device B's output compatible with device A's input?

$$\begin{aligned} VOH &> VIH \\ 2.4 &> 2 \\ VOL &< VIL \\ 0.2 &< 0.8 \end{aligned}$$

- A. Yes
- B. No

B's min VOH is larger than A's min VIH value.
 \Rightarrow Logic '1' can be detected correctly.
B's max VOL is lower than A's max VIL .
 \Rightarrow Logic '0' can be detected correctly.

9

**Why does differential signal has a better noise tolerance than single ended signal?

- A. Two signals allows redundancy, if one fails, the other could still be used
- B. Higher margin between V+ and V- to differentiate between logic '1' and logic '0'
- C. External Interference has the same effect on both the V+ and V- differential signal line so the voltage difference between the V+ and V- signal remains more or less the same
- D. The signal strength of V+ and V- adds up at the receiver, giving raise to higher voltage.

For Differential signal, $V+ > V- \Rightarrow$ Logic '1', $V+ < V- \Rightarrow$ Logic '0'.
 So there is a larger margin to differentiate between Logic '1' and '0' compared to the single ended signal (using VOH, VIH etc).
 Any external influence on the transmission will have the same effect on V+ and V-, so their difference remain more or less unchanged.

Cx1106

10

10

Which of the following scenario will result in the most crosstalk?

- A. 8 PCB traces running side by side to each other transmitting 1Mhz signal, with ground plane in the PCB layer below.
- B. 8 PCB traces running side by side to each other transmitting 10Mhz signal.
- C. 8 PCB traces running side by side to each other transmitting 1Mhz signal. *restrict amount of EM wave wire can radiate out*
- D. 8 shielded cables running side by side to each other transmitting 1Mhz signal

Crosstalk increases as signal frequency increases, interference (crosstalk) on neighbouring signal increases when the lines are close together and when the number of closely packed line increases (more interference sources).

Cx1106

12

12

**Which of the following scenario(s) will increase the skew between a set of well-behaved signals?

- A. Applying additional serpentine/trombone routing to the PCB traces.
- B. Have different PCB trace width but keep the length constant.
- C. Preserving the PCB trace length and width for each signal in the data bus.
- D. Loading some of the data lines within the same data bus with capacitors.

Well behave implies the signal propagation delay in the system has been tweaked so that there are no signal skew among signals that has some dependency, e.g. individual data bit lines in a data bus.
 So any change in parameters that may alter the propagation delay will increase the signal skew.

Cx1106

11

11

Which of the following is False?

- A. Parallel bus implementation leads to more bulky cabling compared to Serial Bus.
- B. Parallel Bus is more affected by signal skew and cross talk because of concentration of multiple signals in close vicinity.
- C. Signal Skew and Cross Talk affects the quality of signal transmission putting a limit on the minimum bus clocking speed. *should be maximum*
- D. Parallel Bus interface design tend to be simpler as it only needs a strobe signal.

Signal Skew and Cross Talk will limit the max bus clocking frequency because the effect of Cross talk increases when data signal frequency (proportional to clock speed) increases. Increase in frequency also result in a smaller allowed margin of error (tolerance) before any signal skew results in wrong data being latched, which means the probability of wrong data latched increases as data signal frequency increases.

Cx1106

13

13

Why is parallel bus more prone to cross talk and signal skew compared to serial bus?

- A. Higher transfer rate
- B. More data lines in close vicinity to each other
- C. No clock line to synchronise to
- D. Synchronous in nature

More data lines means more interference sources (more cross talk) and more signal that has dependency among each other (needs to propagate to the receiver simultaneously).

14

14

**What limitation(s) does signal skew and cross talk placed on the system design?

- A. Maximum number of lines on the data bus
- B. Maximum clock rate
- C. Maximum number of bytes that can be transferred in one block
- D. Minimum delay between each byte transfer

15

15

Advantage(s) of Parallel Bus over Serial Bus

- A. Faster data transfer rate
- B. Faster data transfer rate given the same system clock rate
- C. Cable less prone to breaking as there are more wires bundled together
- D. Cable connector more robust as it is bigger.

16

16

Which factor(s) below will cause a change in propagation delay?

- A. Presence of capacitors and inductors in the signal line
- B. PCB trace width
- C. PCB trace length
- D. Bad soldering of components/wires to PCB, thus increasing the resistance of the signal path.

Bad soldering will introduce more parasitic resistance into the system due to poorer electrical connection.
Changing PCB width/length will change the overall impedance (resistance, capacitance and inductance) of the system.

17

17

Which of the following is False?

- A. All things equal, Serial Bus can achieve a higher clocking rate compared to Parallel Bus.
- B. Given the same clock rate, Serial bus is able to transfer data **further and faster**.
- C. Serial bus is less affected by signal skew compared to Parallel bus.
- D. Serial bus implementation is less bulky compared to Parallel bus.

With the same clock rate, Serial Bus suffers less Cross Talk compared to Parallel bus which has more lines (more interference sources). As are less signal skew (only between the data bit and its clock) compared to parallel bus with multiple data lines.

Cx1106

18

What is true for synchronous and asynchronous serial transfer?

- A. Synchronous transfer use a common clock line to synchronize data transfer between transmitter and receiver.
- B. Asynchronous transfer does not need any form of synchronization at all to transfer the data.
- C. Asynchronous transfer requires transmitter and receiver to have a prefix transfer clock frequency on a common clock line.
- D. Synchronous transfer does not need a common clock between the transmitter and receiver.

Ansynch transfer still needs some form of synchronisation in order for transmission to be reliable, just that they transfer these synch info via the data line.

Cx1106

20

If bi-directional data transfers are supported in the system, which of the following Serial transmission will yield the fastest transfer rate?

- A. Half Duplex
- B. Full Duplex
- C. Simplex

Full duplex allows data transfer in both direction simultaneously, so effectively has higher data transfer rate. In half duplex case, some overhead will be incurred when switching data transfer direction.

Cx1106

19

Which of the following serial bus(es) are synchronous in nature?

- A. UART
- B. SPI
- C. USB
- D. RS232

Cx1106

21

20

21

Which of the following UART control/data bit(s) has a LOGIC LOW?

- A. Data '0'
- B. STOP Bit
- C. START Bit
- D. IDLE State

IDLE → 1
Start → 0
Stop → 1

Cx1106

22

22

What is parity bit used for?

- A. Error Correction
- B. Error Prediction
- C. Error Detection
- D. Error Detection and Correction

The parity scheme in UART can only detect that there are some issue with the transmission. It is not able to identify exactly which bit is wrong, that means it also cannot perform error correction.

Cx1106

24

24

What is the value of the Parity bit if the number of '1's in the data packet is even? Assume Even Parity Scheme is used.

- A. 0
- B. 1

In Even Parity scheme, transmitter monitor the number of '1's that are transmitted in the Data Field, the transmitter then transmit a '1' or '0' in the parity field so that the total number of '1's in the Data+Parity Field is an Even Number.

Cx1106

23

23

Which of the following UART control/data bit(s) is represented by a +15V signal when transmitted via PC COM port using RS232 transmission standard?

Logic '0' → +15V
Logic '1' → -15V

- A. DATA '1'
- B. STOP
- C. IDLE State
- D. START

only one with Logic 0

RS232 standard uses the UART protocol transmission, i.e. START bit has a Logic '0', STOP bit has a Logic '1' and IDLE has a Logic '1'. But it uses a electrical standard interface that employ a 'negative logic', i.e. Logic '1' is represented by a voltage smaller than that of Logic '0'. E.g. Logic '1' can be equal to -15V while Logic '0' can be equal to +15V.

Cx1106

25

25