Wired Communication

By: Haron Abdel-Raziq

Collective Embodied Intelligence Lab

Schedule & Other Details

- We noticed the struggle with Lab 2
  - Lab 2 is now due on October 5th
  - Milestone 2 is Due on October 12th
- Next week (Monday) there is an FPGA lecture
  - Will be given by Professor Bruce Land
  - You should definitely attend

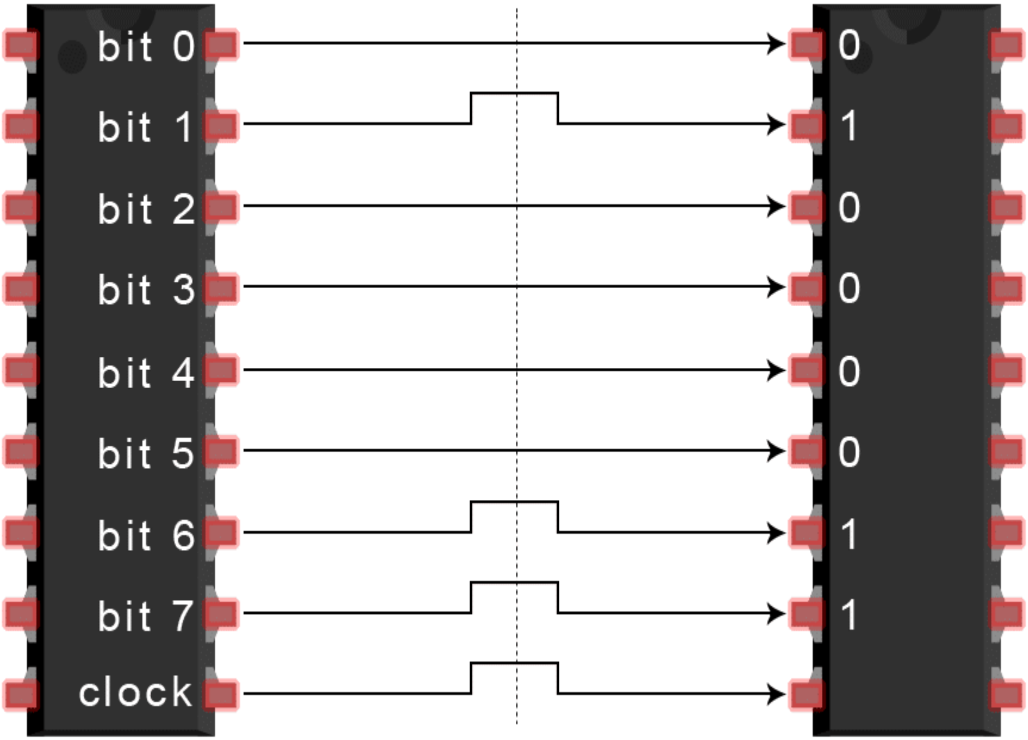
Wired Communication

Why Wired?

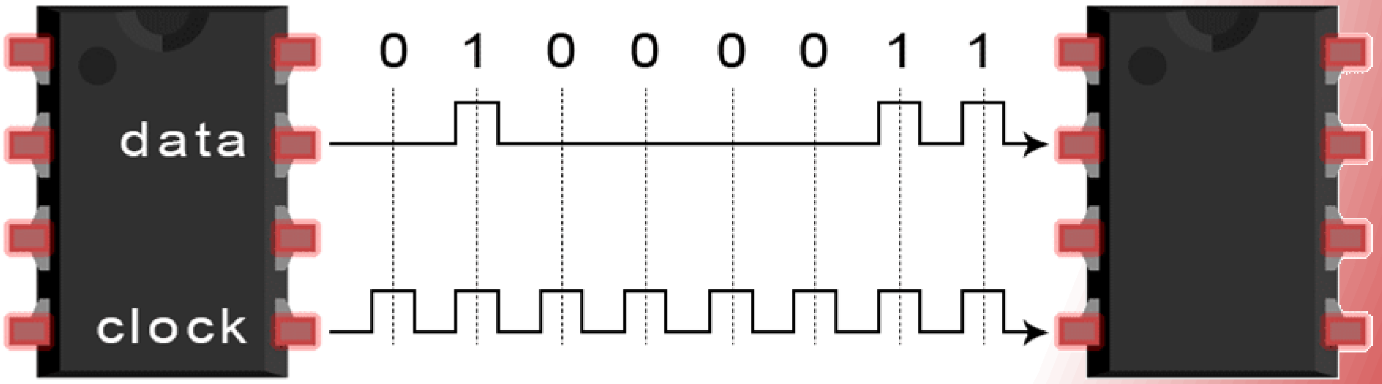
- Wired connectivity is by far the most prevalent type of interconnection on chip
- Important to understand how it functions if you intend on combining many chips together on a single board
- Allows for a wide variety of interfaces
- Reliable, fast, relatively straightforward to implement/understand

Parallel Vs. Serial

Sending Bit Pattern 01000011



Parallel



Serial

Parallel Vs. Serial

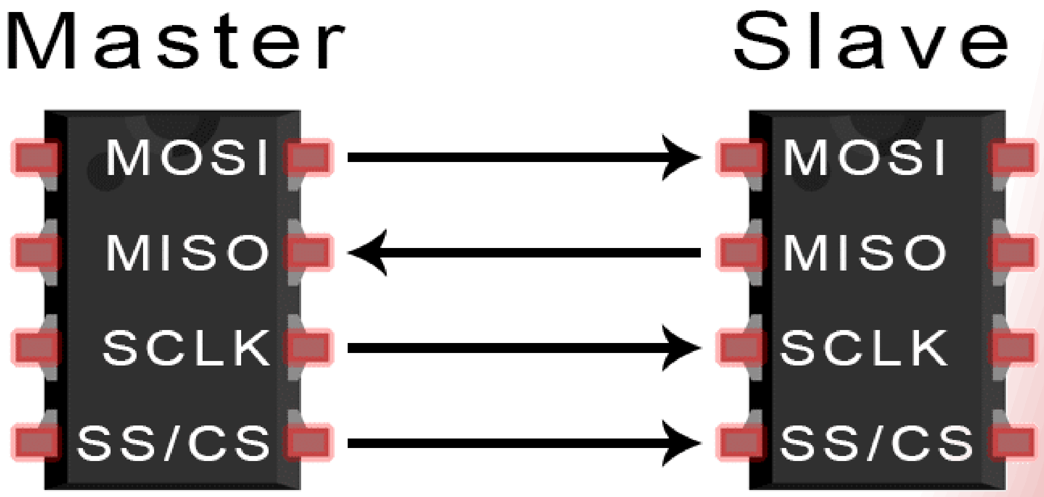
- Parallel
  - Pros
    - Transmit data over multiple channels simultaneously
  - Cons
    - Must always be clocked
    - Requires more wires and pins
    - Crosstalk
    - Intersymbol Interference (ISI)
- Serial
  - Pros
    - Simpler design (fewer wires and pins)
    - Can be clocked faster than parallel
    - Can be asynchronous
  - Cons
    - Sends a single stream of data
- PCI: Max Parallel Speed: 4.266 Gbits/s, Max Serial Speed: 256 Gbits/s

Serial Communication

	UART	SPI	I2C
Maximum Speed	115200b/s	10 Mbps	5 Mbps
Synchronous/ Asynchronous	Asynchronous	Synchronous	Synchronous
# Of Wires	2	4	2
Max # of Masters	1	1	Unlimited
Max # of Slaves	1	Unlimited*	1008

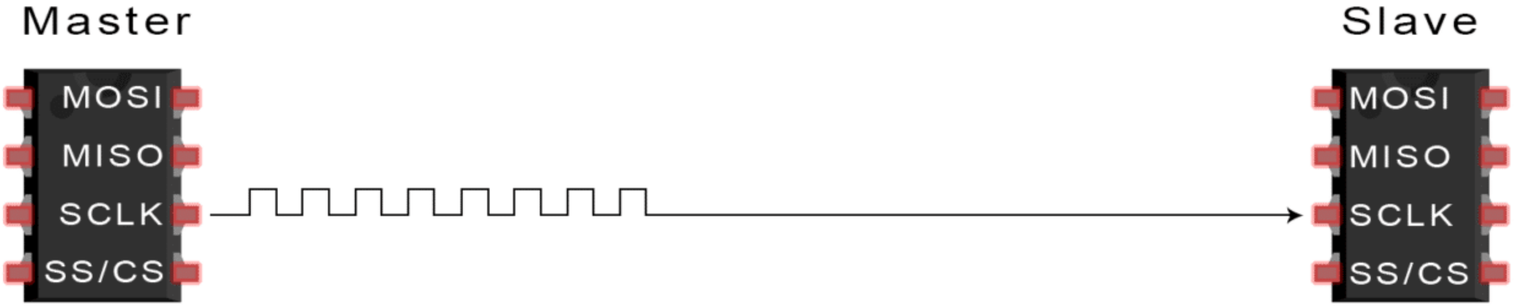
SPI (Serial Peripheral Interface)

- Master sends data on MOSI line
- Clocked by SCLK
- One bit per clock cycle
- Configurable clock
- Multiple slaves controlled by SS/CS (active low)
- Commonly used for: memory, memory cards, sensors, displays

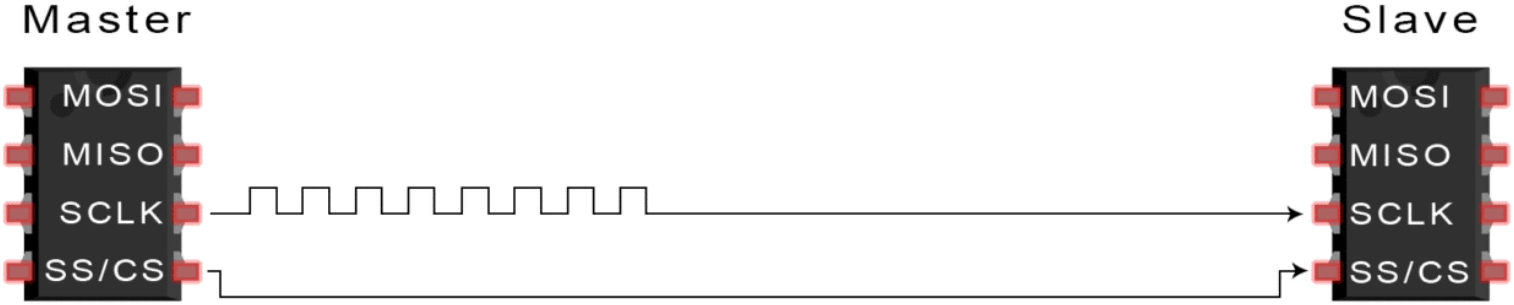


MOSI = Master Out/Slave In
MISO = Master In/Slave Out
SCLK = Serial Clock
SS/CS = Slave/Chip Select

Master Outputs Clock Signal

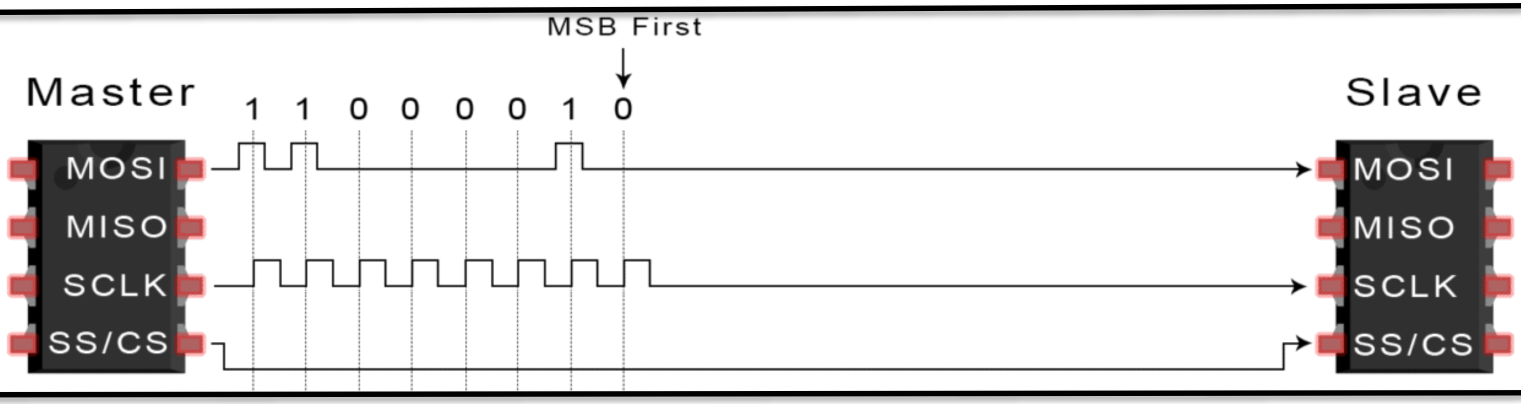


Master Switches SS/CS Low



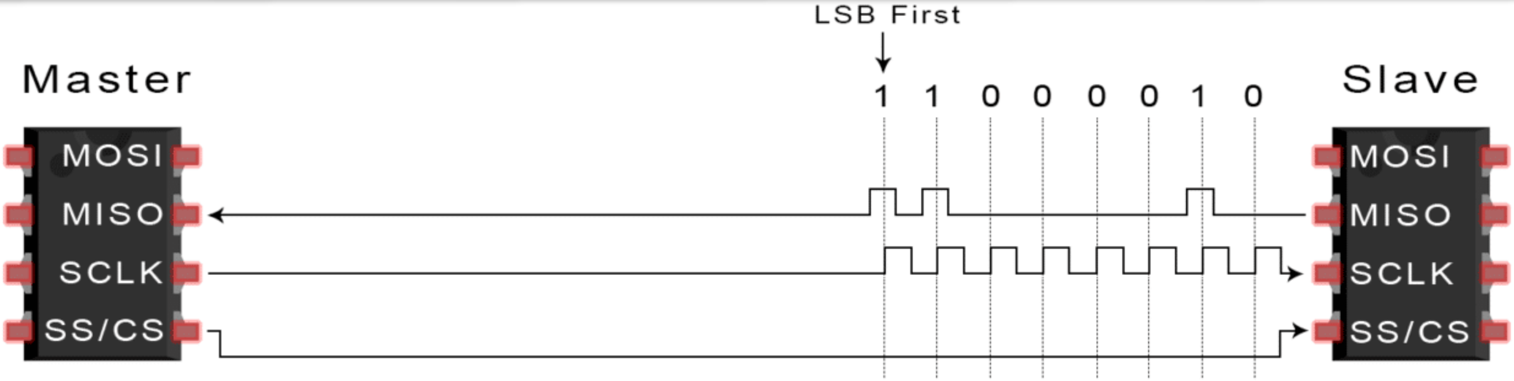
Slave Is Activated

Master Sends Data One Bit At A Time



Slave Reads Bits As They Are Received

Master Reads One Bit At a Time

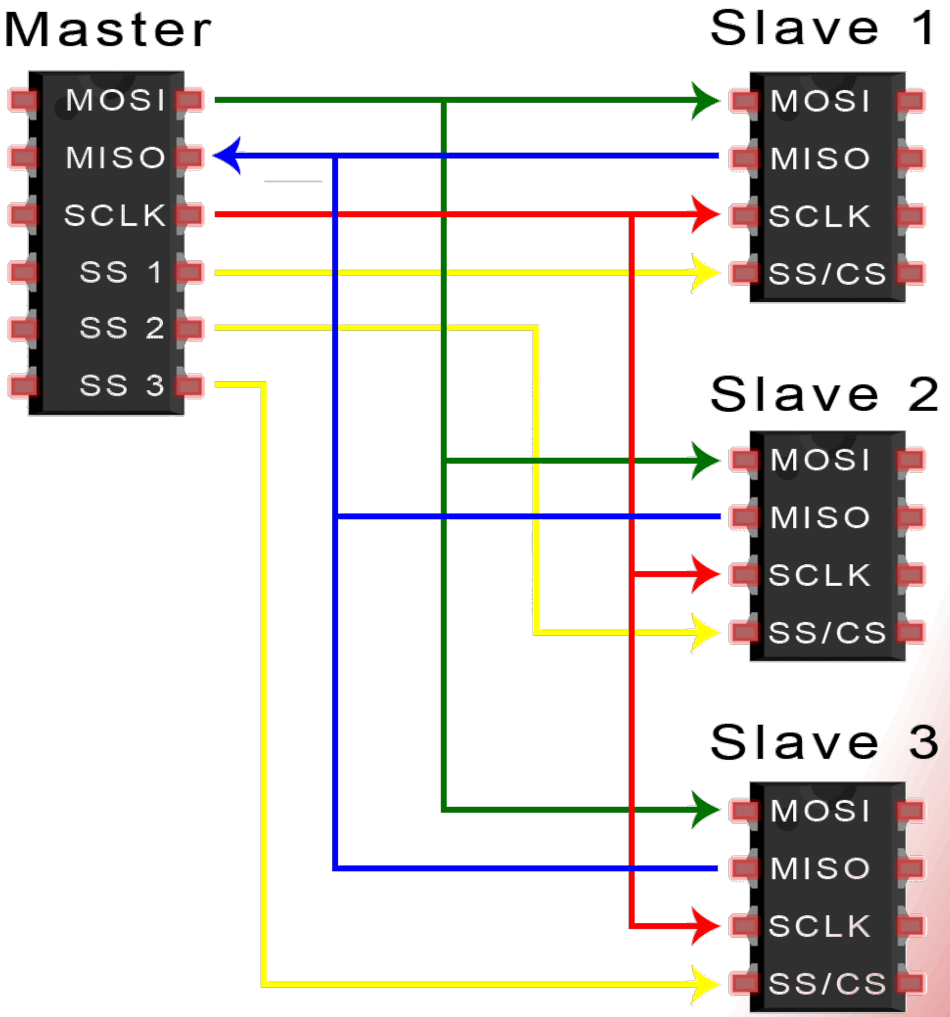


Slave Responds Along MISO

Collective Embodied Intelligence Lab

SPI

- Advantages
  - Stream bits (unlimited frame)
  - High speed
  - Send & receive lines
- Disadvantages
  - No error checking
  - No acknowledgement
  - Requires four wires
  - Single master

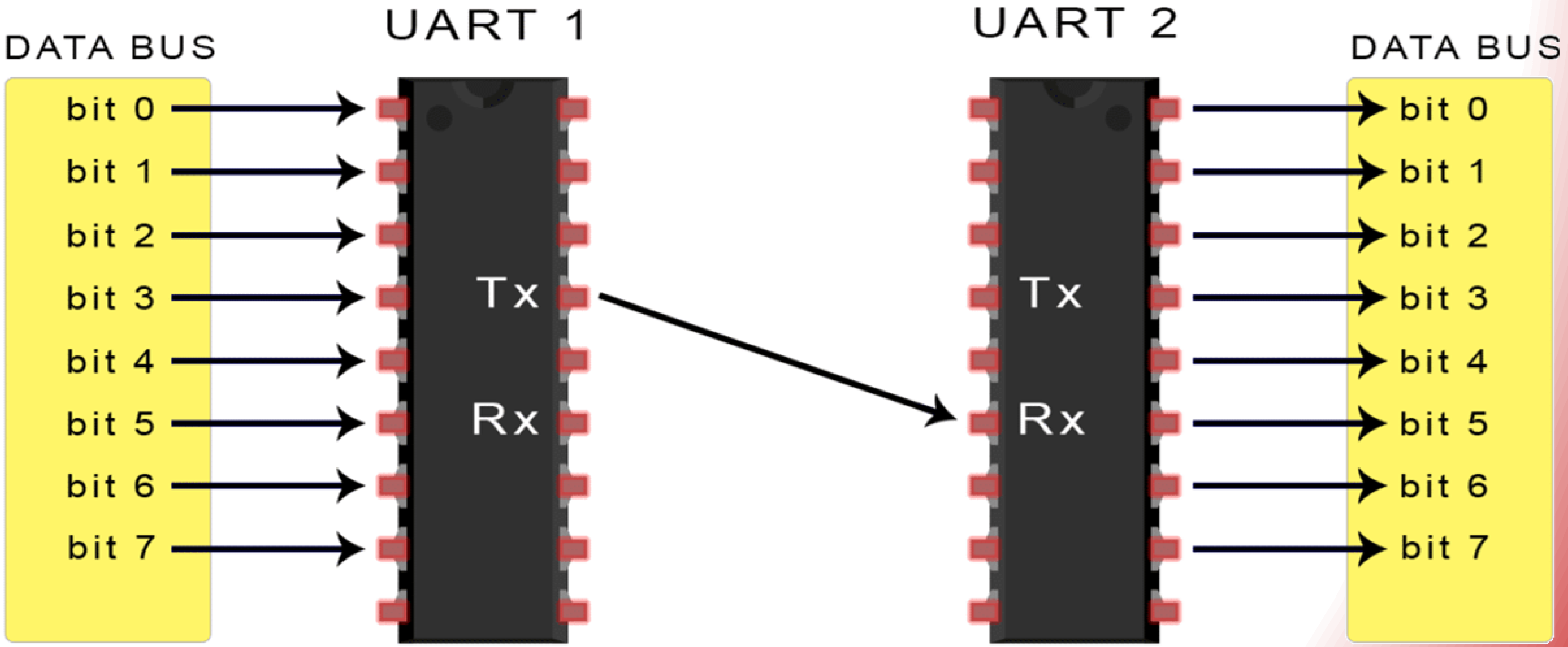


Collective Embodied Intelligence Lab

SPI

- Team Alpha implemented SPI communication to their FPGA
- Check out their website for Lab 4
- https://cei-lab.github.io/ECE3400-2017-teamAlpha/lab4.html

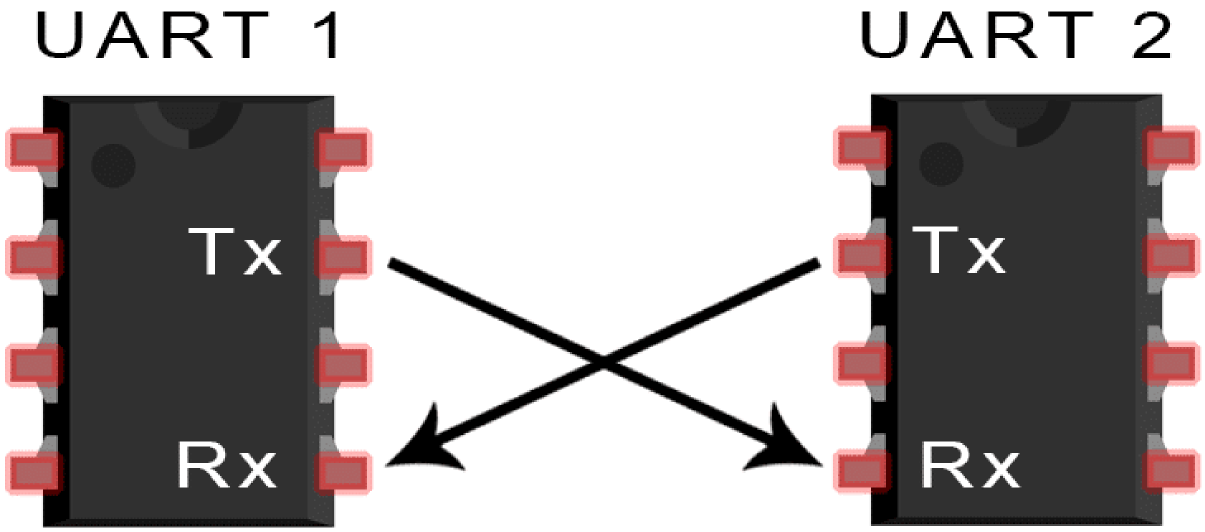
UART (Universal Asynchronous Receiver/Transmitter)

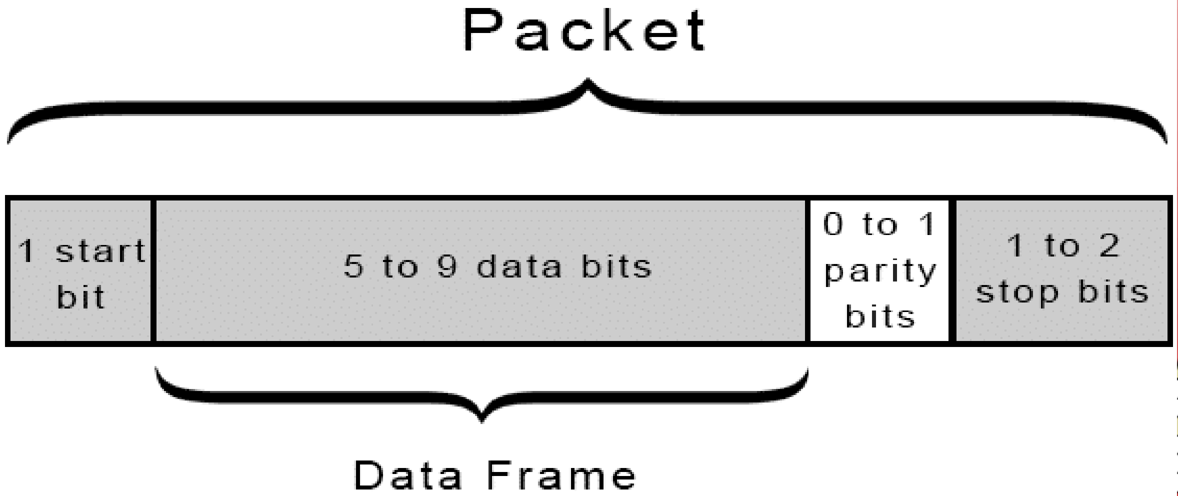


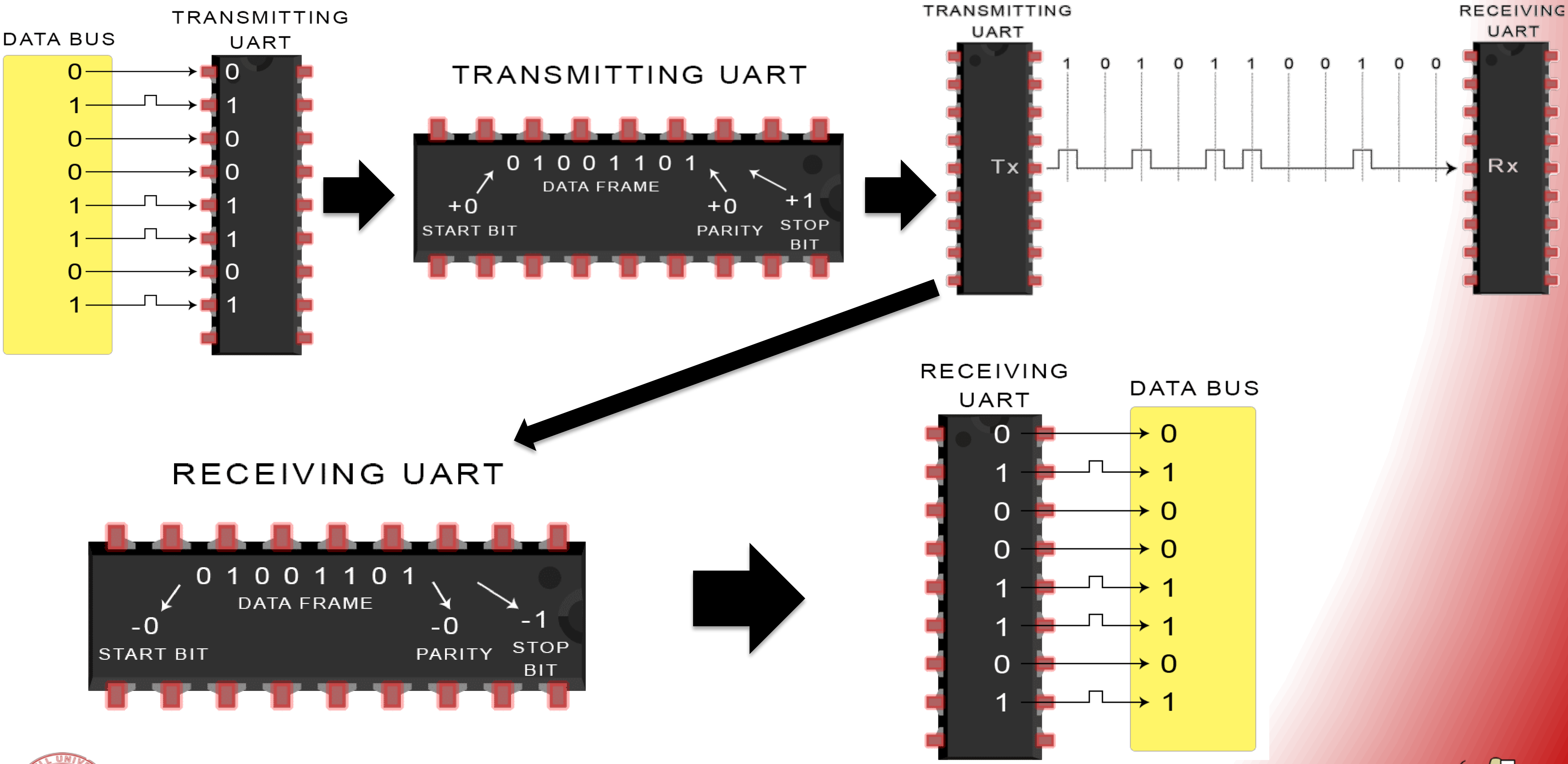
Collective Embodied Intelligence Lab

UART

- A UART is a physical IC (sometimes contained as part of an MCU)
- Two wire communication
- Converts incoming parallel data to serial, transmits, converts back
- Asynchronous
  - But there is an agreed upon baud rate
  - Start bit signals when reading begins
  - Use a particular "packet" format





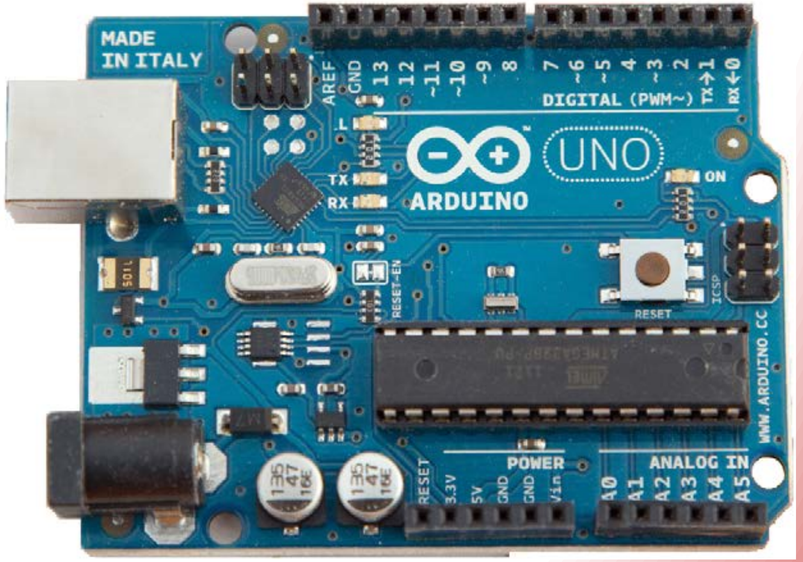


Collective Embodied Intelligence Lab

UART

- Advantages
  - Widely used
  - Requires only two wires
  - Asynchronous
  - Parity bit for error checking
- Disadvantages
  - Fixed packet size
  - Single Master/Slave
  - Baud rates must be within 10%

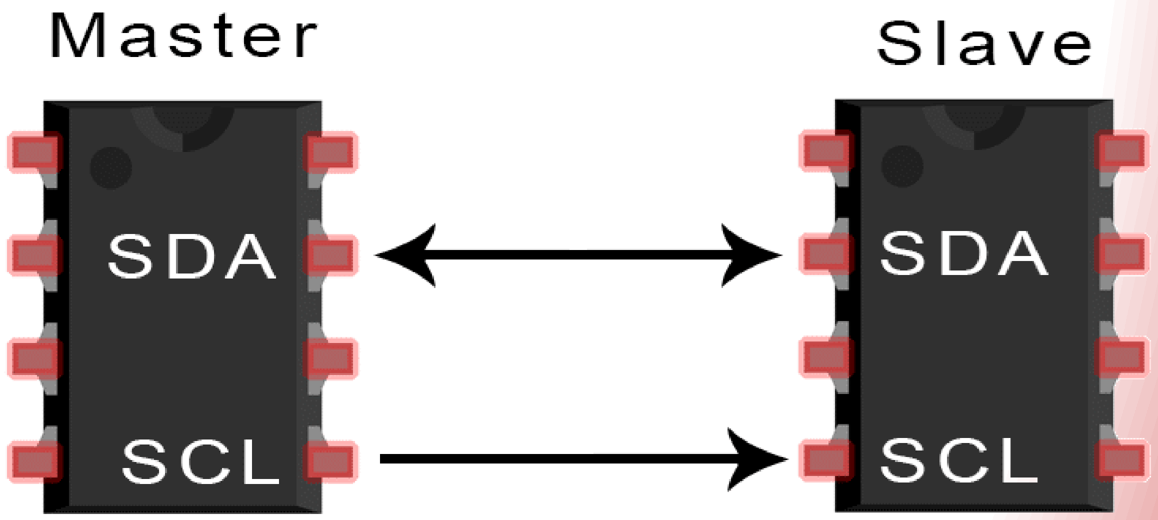
- Commonly Used For
  - GPS Modules
  - Bluetooth Modules
  - RFID Card Reader Modules

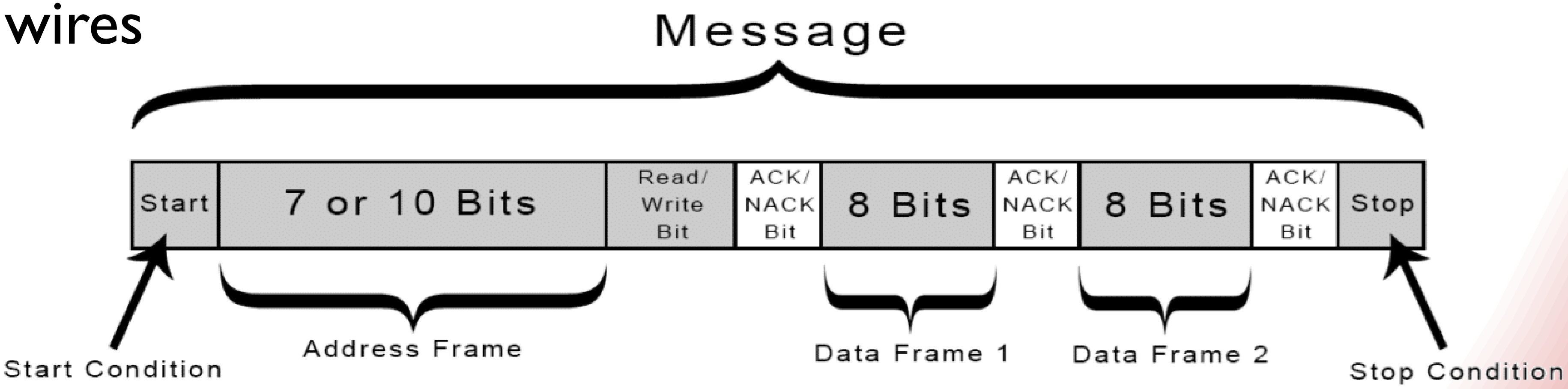


Collective Embodied Intelligence Lab

I2C (Inter Integrated Circuit)

- Inter Integrated Circuit combines SPI and UART benefits
  - Multiple masters and multiple slaves
  - 2 wires

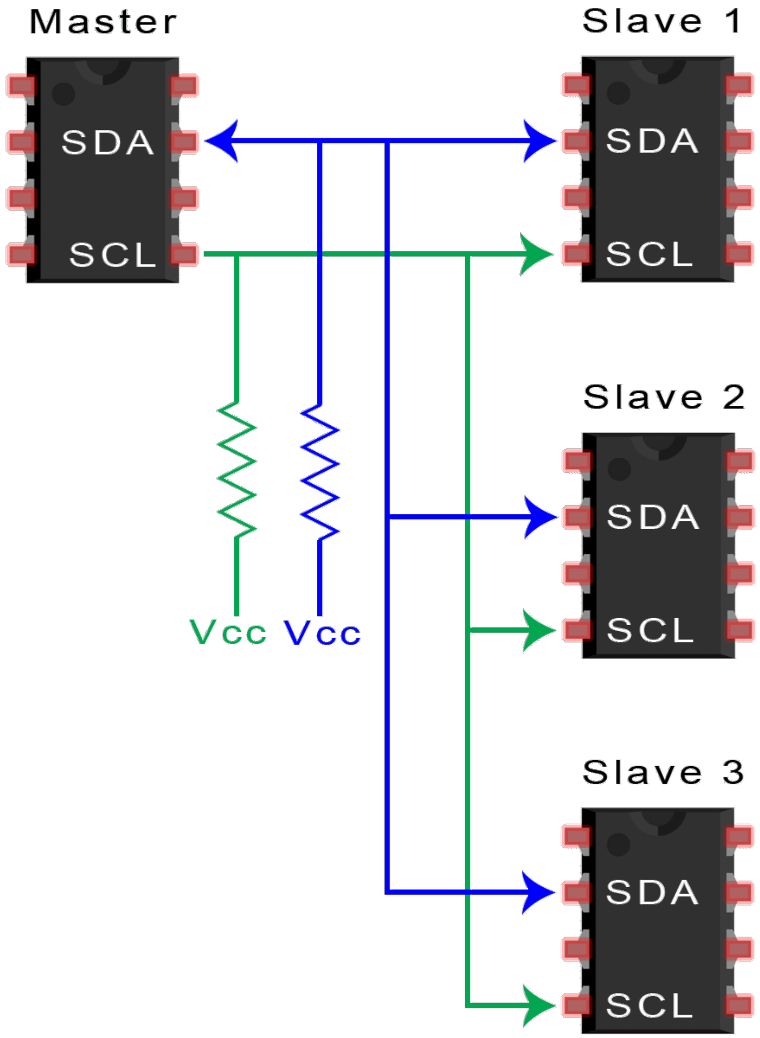


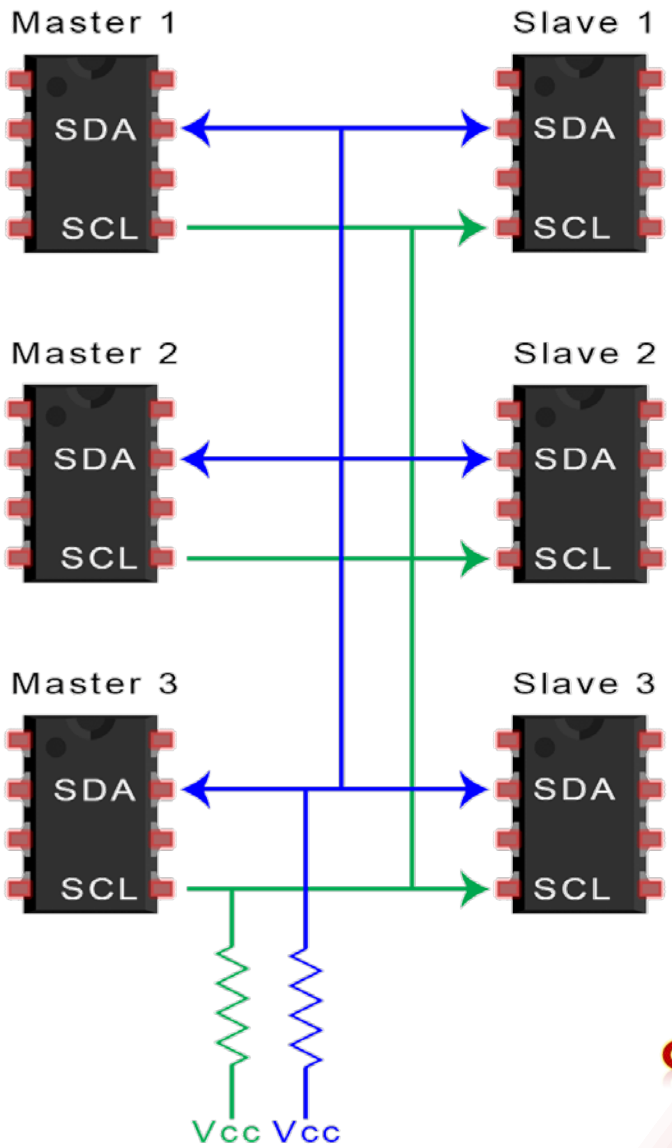


Start Condition: SDA High to Low before SCL High to Low
Stop Condition: SDA Low to High after SCL High to Low

Collective Embodied Intelligence Lab

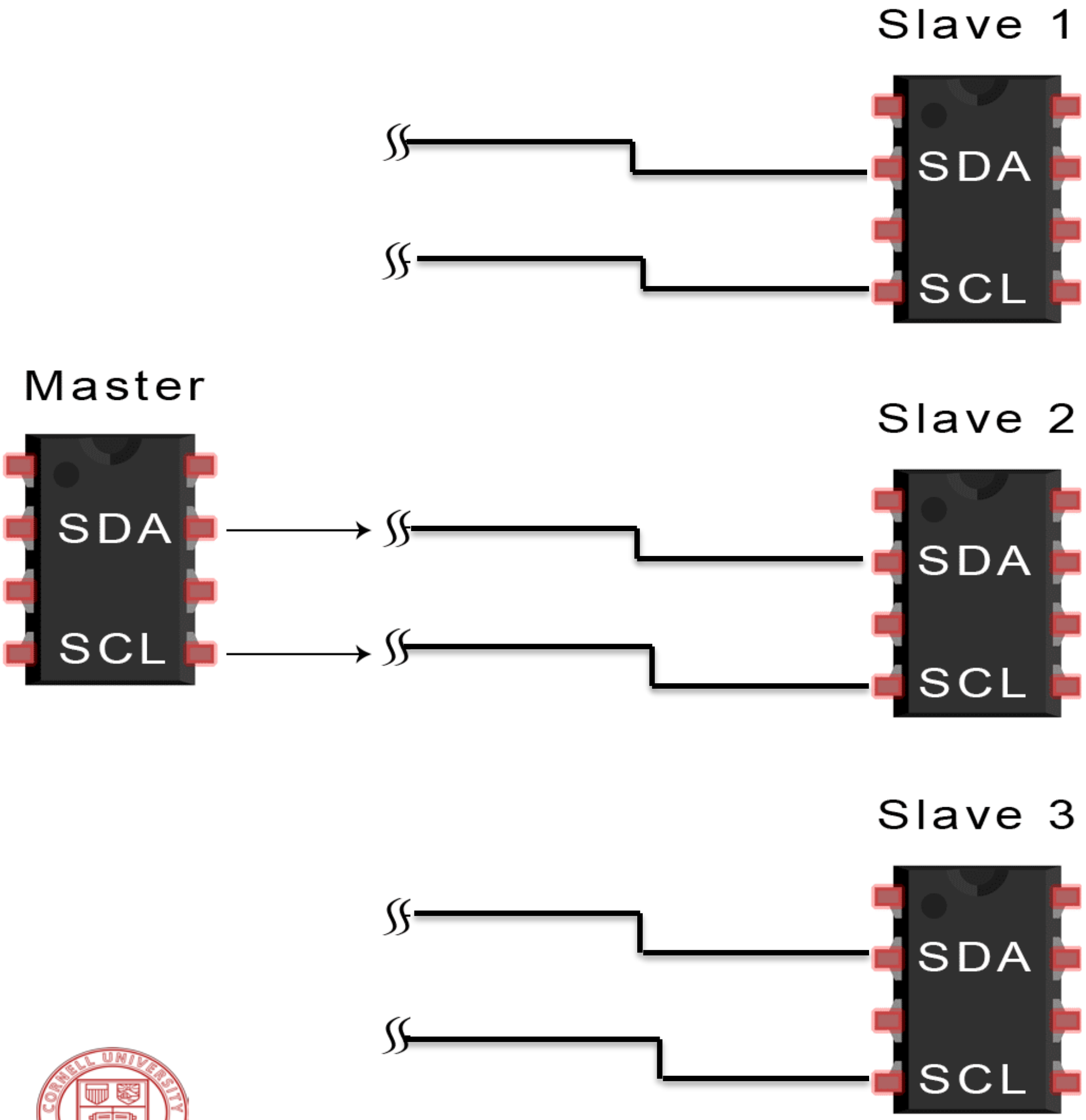
I2C Configurations



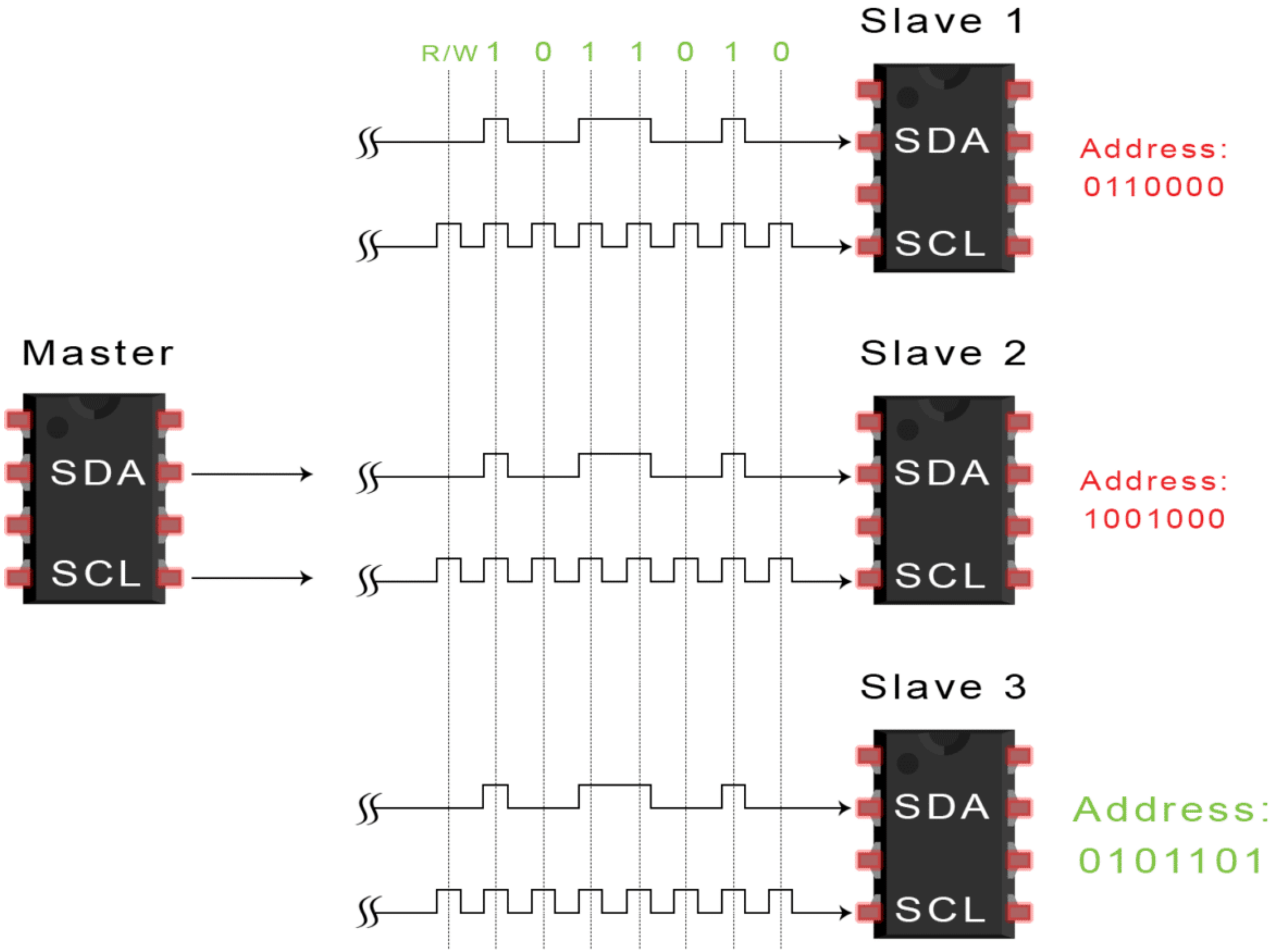


Collective Embodied Intelligence Lab

Start Condition Is Sent

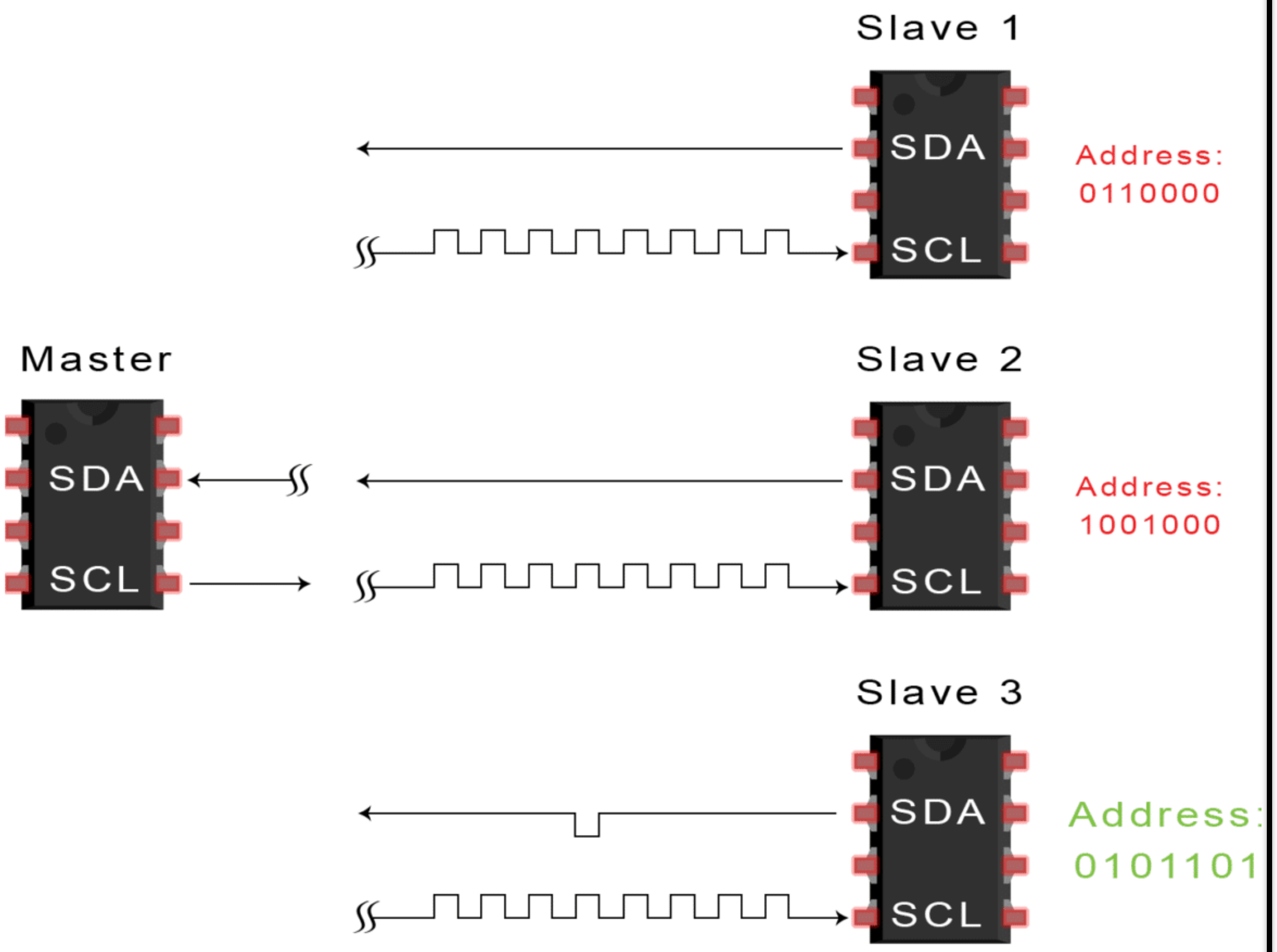


Address is Sent

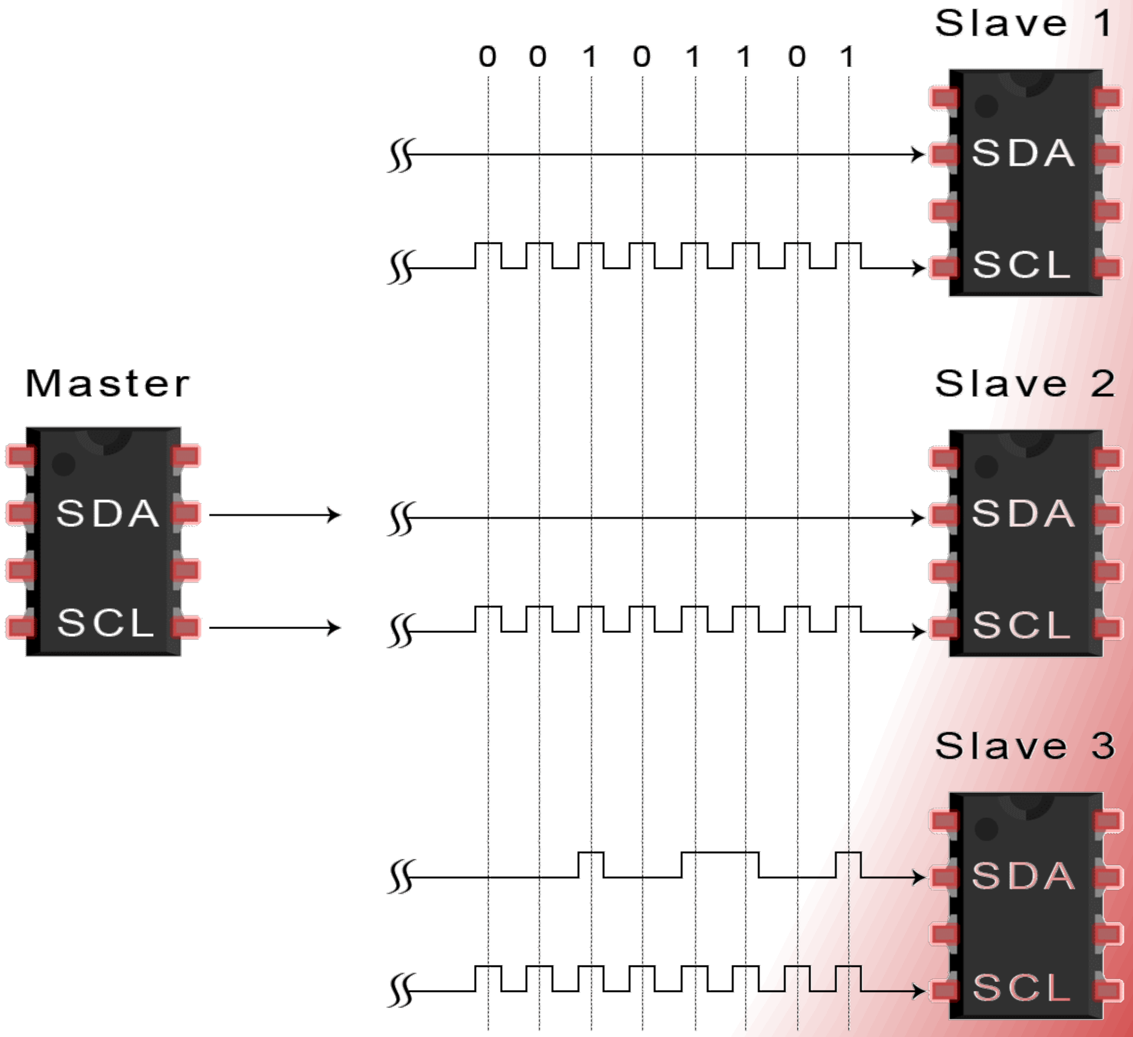


Collective Embodied Intelligence Lab

ACKnowledgement Sent (Pull SDA Line Low)

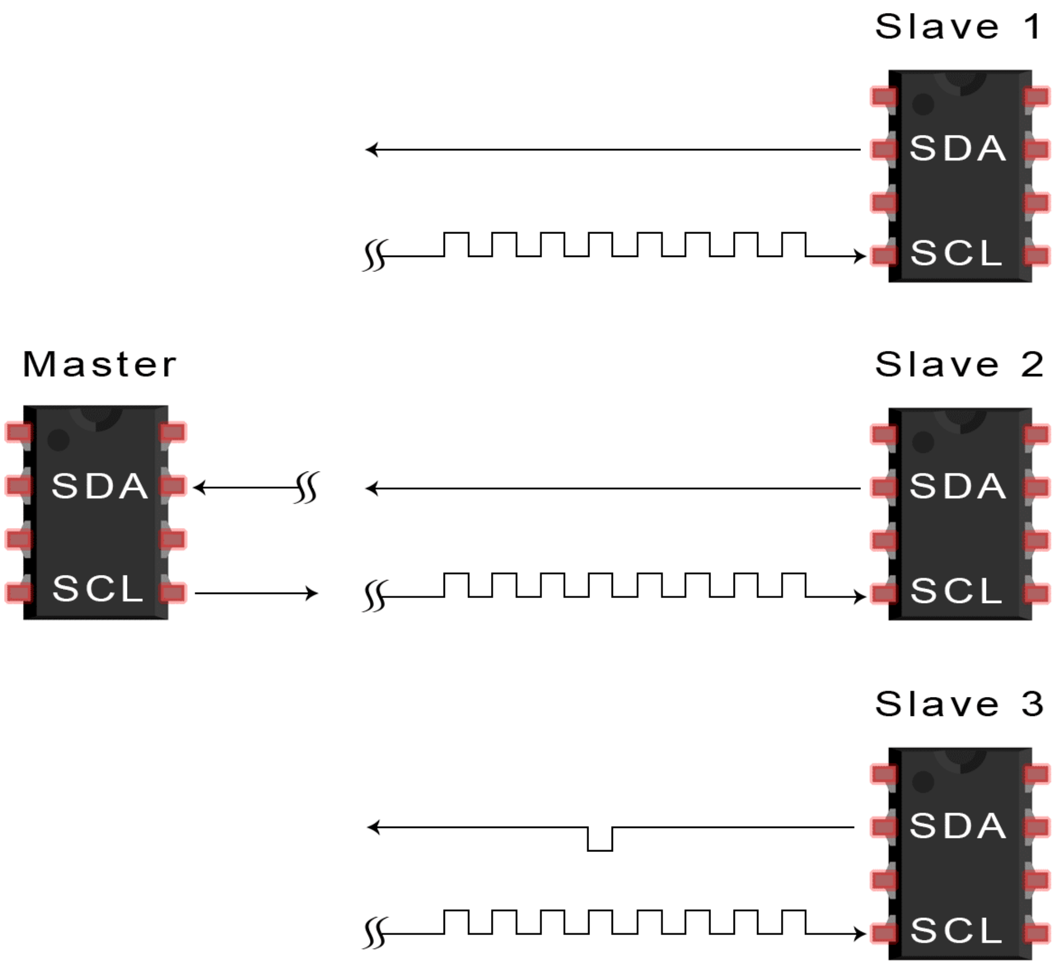


Master Sends or Receives Data

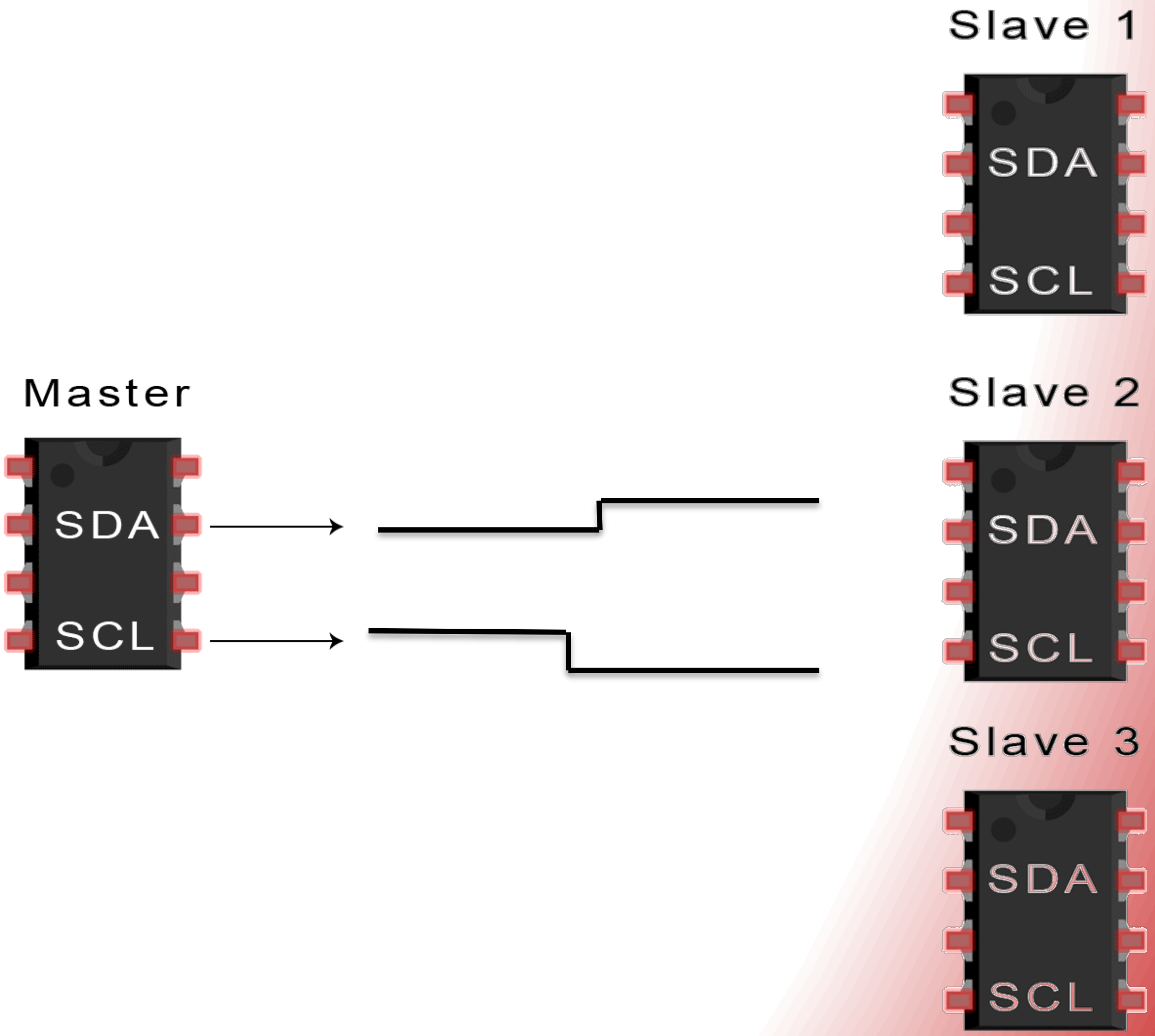


Collective Embodied Intelligence Lab

After Each Data Frame An ACK/NACK is Sent



Master Sends Stop Condition



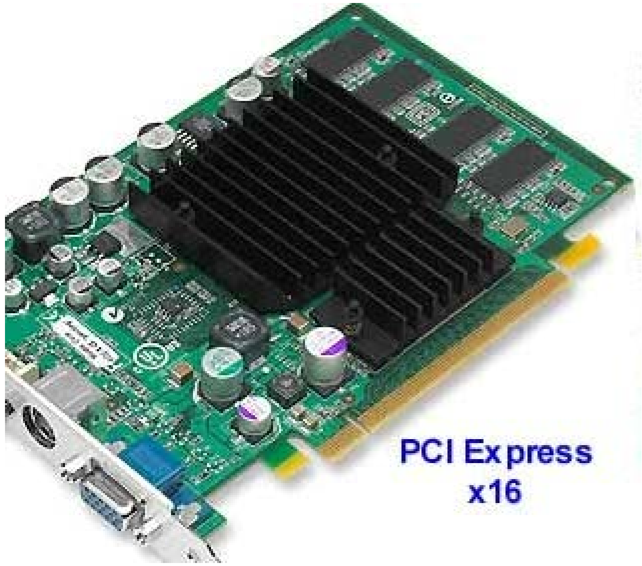
Collective Embodied Intelligence Lab

I2C

- Advantages
  - Widely used
  - Requires only two wires
  - Ack/Nack
  - Multiple master/multiple slave
- Disadvantages
  - Fixed frame size
  - Slower than SPI
  - More complicated than SPI
- Commonly Used For
  - Sensors
  - Control of multiple devices
  - OLEDs
  - Accelerometers

Other Serial Devices

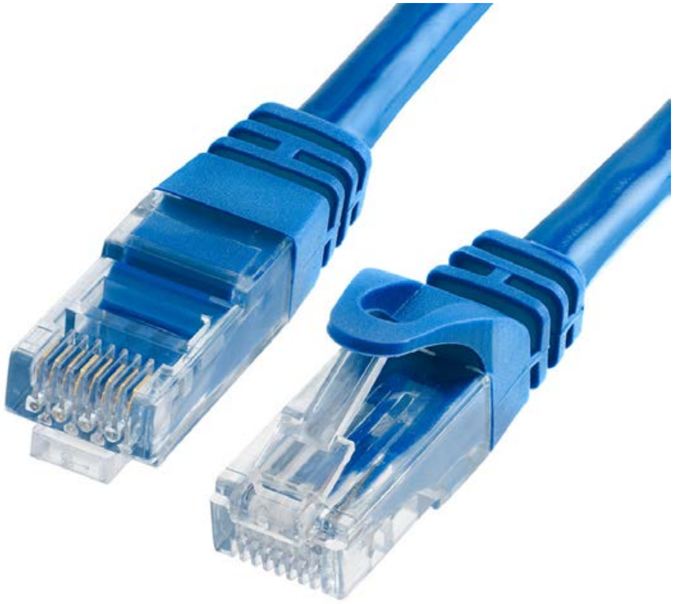
- USB ,PCIe, SATA, CAN



Collective Embodied Intelligence Lab

Even More Serial Devices

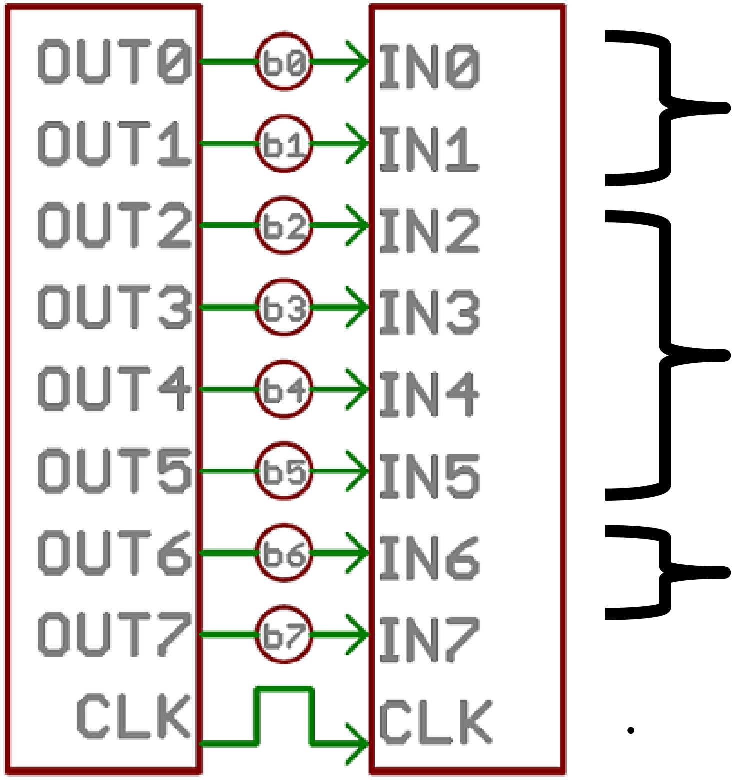
- Ethernet, HDMI, Coax (Digital Video), Display Port, Firewire



Parallel Communication

- Examples are PCI, PATA
  - No longer favorable because of limits in speed
  - Cable length
  - Hardware complexity
- However, is easy to implement on FPGA

Parallel Communication & FPGAs



<u>For example:</u>

2-bits for message type/header
(eg. moved to new location; found treasure at square, etc.)

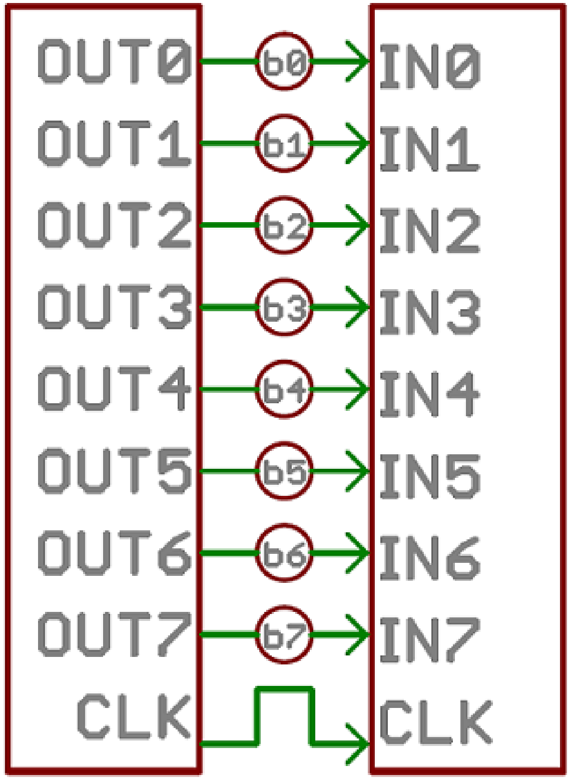
3-bits for message body
(eg. moved to square 4,5; square 7,7 is type *wall*, etc.)

1-bit for started, stopped mapping…

and so on …

Collective Embodied Intelligence Lab

Parallel Communication & FPGAs



Drawback: Consumes a lot of pins

Plenty of ways to implement (pseudocode):

```
reg [2:0] grid_array [n-1:0][n-1:0];
wire [2:0] square_color;
assign square_color = GPIO_0_D[k:k-2];
…
always @ (*) begin
  if (square_color == 3'd0) begin
    grid_array[x][y] = square_color;
  end
…
```

Arduino sends color directly. Reserved GPIO pins specifically for color.

```
wire inputs [n:0];
assign inputs = GPIO_0_D[m:0];
…
always @ (*) begin
  if (inputs{1:0} == 2'd3) begin
    grid_array[x][y] = inputs{5:2};
  end
…
```

msg header
msg body

Arduino sends msg header {1:0} + body {5:2} indicating color change should take place at square x,y.

Choosing a Protocol

- Heavily dependent on what type of robot you want to build
- Considerations
  - Cost
  - Usable wire length
  - Device pin count
  - Speed
  - Etc.
- You can also make your own communication protocol!

References

- Much of this presentation material is based on material discussed in www.circuitbasics.com
  - SPI, UART, & I2C
- Sparkfun also has excellent discussions on these topics