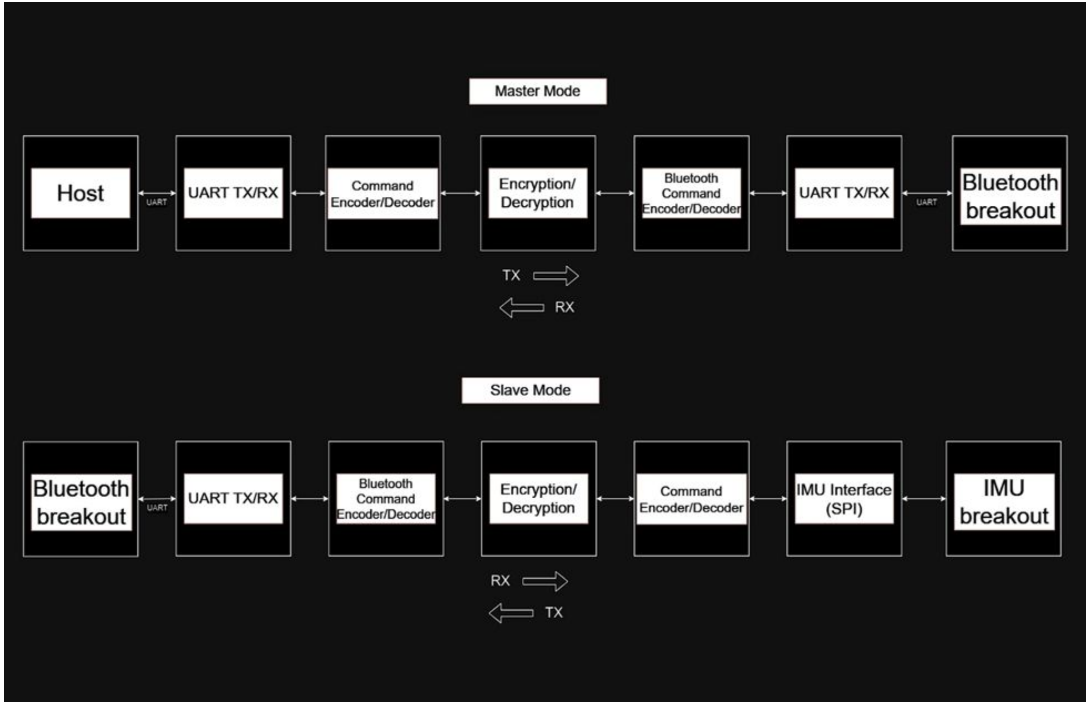
Wireless IMU Controller (WIC)

Noah Mecham, Mike Mercer, Sean Koo, Elmir Dzaka

Application

- Speed up data transfer and collection from the IMU
- All in one chip
- Offload Software

Master/Slave



Master Side

Slave Side

Sent From Host:

- **UART**
- **Command Encoder**
- **Encryption**
- **Bluetooth Command Encoder**
- **UART**

Sent to Host:

- **UART**
- **Bluetooth Command Decoder**
- **Decryption**
- **Command Decoder**
- **UART**

Sent to Slave:

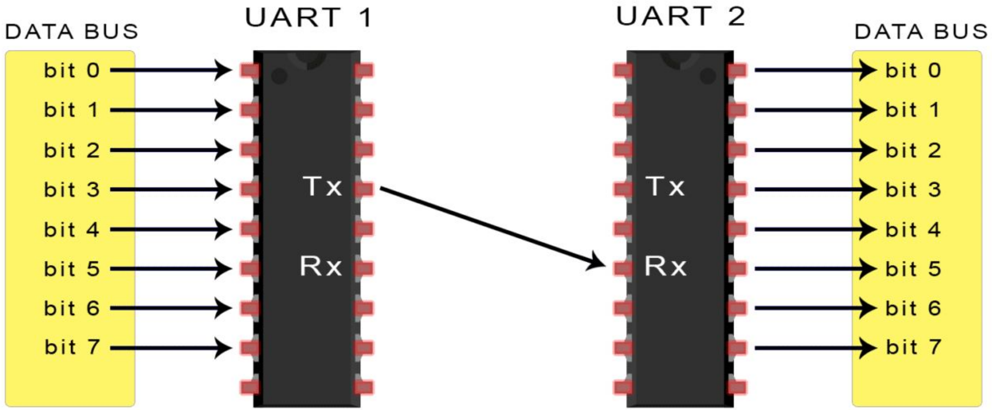
- **UART**
- **Bluetooth Command Decoder**
- **Decryption**
- **IMU Command Decoder**
- **IMU (SPI) Interface**

Sent From Slave:

- **IMU (SPI) Interface**
- **IMU Command Encoder**
- **Encryption**
- **Bluetooth Command Encoder**
- **UART**

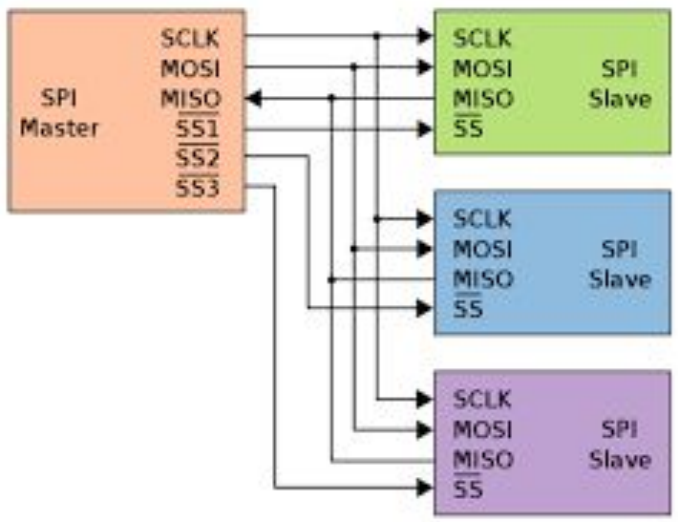
Design Choices - UART

- UART provides a simple communication protocol
  - Send data back and forth
- Straightforward implementation
  - Rx and Tx data lines
- Simple
  - Less area needed, increase in performance



Design Choices - SPI

- We needed to incorporate a SPI interface since our IMU module requires a SPI communication
  - IMU gathers flight data from unmanned craft
  - Data is modified to send back to host via bluetooth and UART



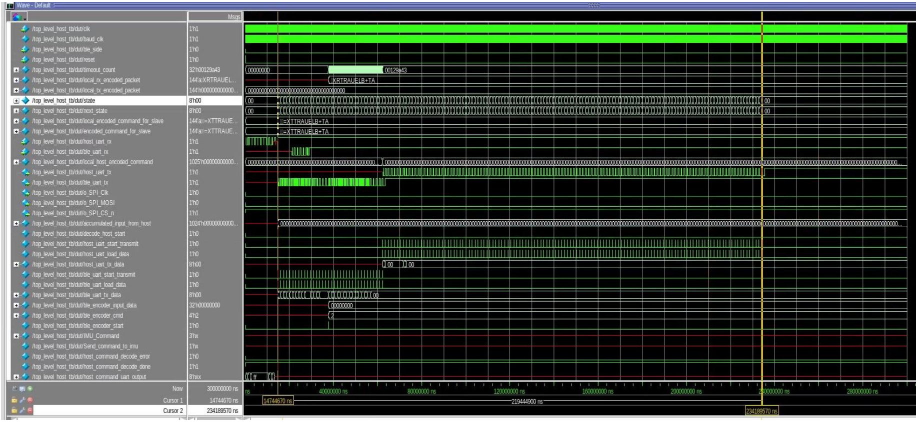
Design Comparison

- Designed to off-load software.
- Similar design typically implemented with mcu.
- Clock speed increase is big advantage.
- Our design takes ~24ms to deliver a packet to BLE module, ~25ms on slave to retrieve data from IMU and send packet back, and ~172ms to deliver packet back to host (50 MHz 9600 baud).
- With ~99% of the clock time used for uart. Only 220ns left for processing.

```
Analysis View: wc
Other End Arrival Time          0.124
- Setup                         0.121
+ Phase Shift                  20.000
= Required Time                20.002
- Arrival Time                 17.535
= Slack Time                    2.468
     Clock Rise Edge                     0.000
     + Input Delay                       0.000
     = Beginpoint Arrival Time           0.000
     +------------------------------------------------------------------------------------------+
     |          Instance            |          Arc          |  Cell   | Delay | Arrival | Required |
     |                              |                       |         |       |  Time   |   Time   |
     |------------------------------+-----------------------+---------+-------+---------+----------|
     |                              | reset ^               |         |       |   0.000 |    2.468 |
     | reset_pad                    | pad ^ -> DataIn ^     | pad_in  | 0.225 |   0.225 |    2.692 |
     | soc/g168216                  | A ^ -> Z v            | INVX2   | 7.289 |   7.514 |    9.981 |
     | soc/g167145                  | C v -> Z ^            | NAND3X1 | 6.462 |  13.976 |   16.444 |
     | soc/g167013                  | B ^ -> Z v            | NOR2X1  | 0.563 |  14.539 |   17.007 |
     | soc/g166609__1617            | A v -> Z ^            | NAND2X1 | 0.477 |  15.016 |   17.483 |
     | soc/g166426__2398            | B ^ -> Z v            | NOR2X1  | 0.261 |  15.277 |   17.745 |
     | soc/g166220__2398            | A v -> Z v            | AND2X1  | 0.524 |  15.801 |   18.269 |
     | soc/g166089__1666            | A v -> Z ^            | NAND2X1 | 0.242 |  16.043 |   18.511 |
     | soc/g165995__4733            | A ^ -> Z ^            | AND2X1  | 0.208 |  16.251 |   18.719 |
     | soc/g165898__5122            | B ^ -> Z v            | NAND3X1 | 0.110 |  16.361 |   18.829 |
     | soc/g165882__1666            | B v -> Z ^            | NOR2X1  | 0.108 |  16.469 |   18.937 |
     | soc/g165855__7098            | A ^ -> Z v            | NAND3X1 | 0.132 |  16.601 |   19.069 |
     | soc/g165840__5107            | B v -> Z ^            | NOR2X1  | 0.224 |  16.826 |   19.293 |
     | soc/g165829__9315            | A ^ -> Z v            | NAND3X1 | 0.111 |  16.937 |   19.404 |
     | soc/g165816__1705            | B v -> Z ^            | NOR2X1  | 0.249 |  17.186 |   19.654 |
     | soc/g165796__9945            | A ^ -> Z v            | NAND3X1 | 0.118 |  17.304 |   19.772 |
     | soc/g165794                  | A v -> Z ^            | INVX2   | 0.103 |  17.407 |   19.874 |
     | soc/g165785__5115            | A ^ -> Z v            | NAND3X1 | 0.128 |  17.534 |   20.002 |
     | soc/ble_uart_tx_data_reg[1]  | D v                   | DFFQX1  | 0.000 |  17.535 |   20.002 |
     +------------------------------------------------------------------------------------------+
```

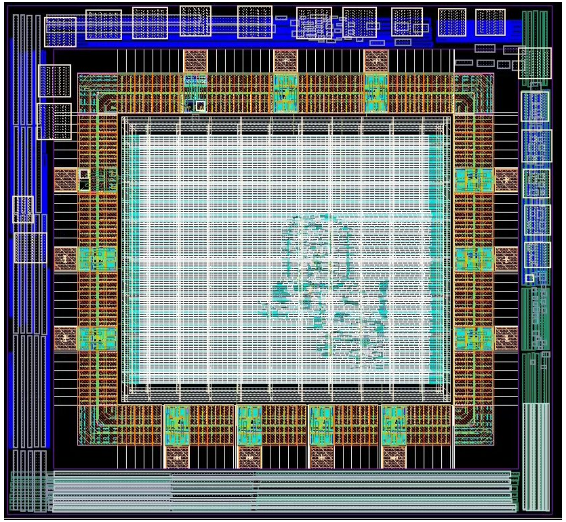
Design Comparison (continued)

- A similar design using mcu would require a mcu on both the host and slave side. Our design only requires one on the host side.
- Applications with constrained processors or high amounts of processing would be able to offload processing to this chip to free up instruction cycles.
- Mcu has advantage of software libraries to help reduce bring up time and allow work on multiple platforms.



Conclusion

- **What does this chip do?**
  - Transmits IMU data
  - Scalable to multiple slaves
- **Why does our chip matter?**
  - Faster and efficient
- **How to utilize this chip?**
  - Military field
  - Drone controller



Questions??