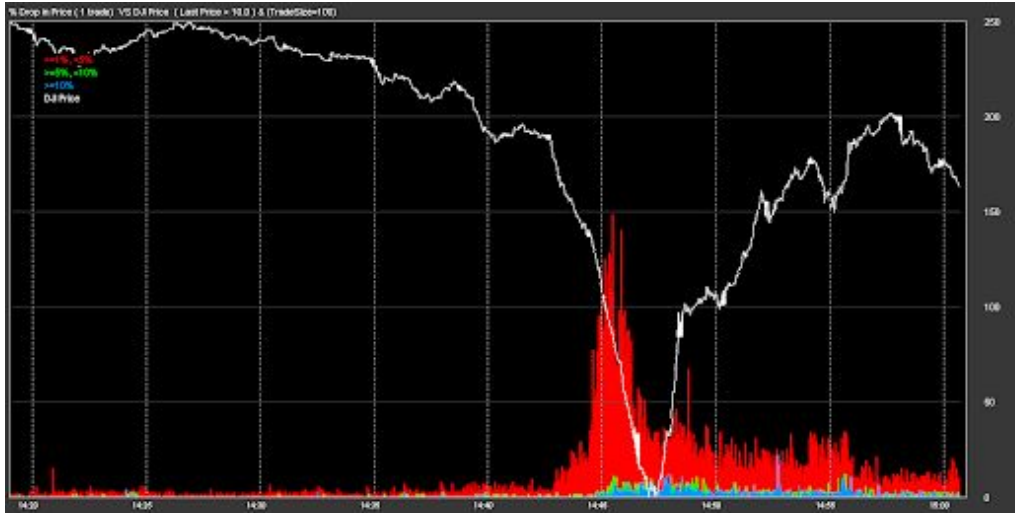
High Frequency Trading Acceleration using FPGA + SoC

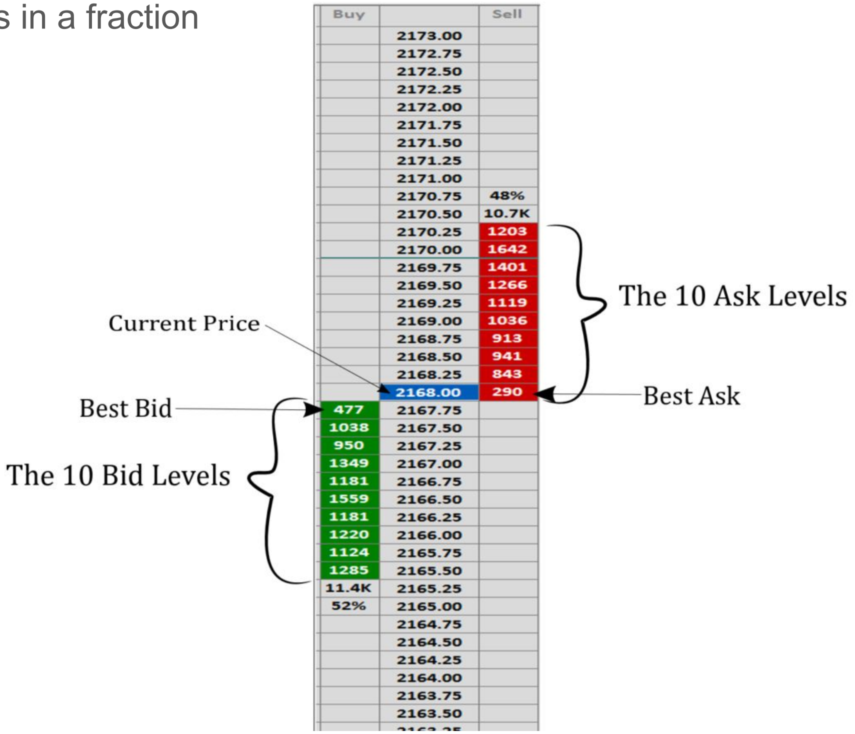
Brandon Reponte, Leeza Gutierrez-Ramirez, Rudy Osuna

But what is HFT?

- Sending Large numbers of orders to buy and sell assets in a fraction of a second
  - Only possible with optimized infrastructure
- Pros: Provides market liquidity
- Cons: Order Spoofing (market manipulation)



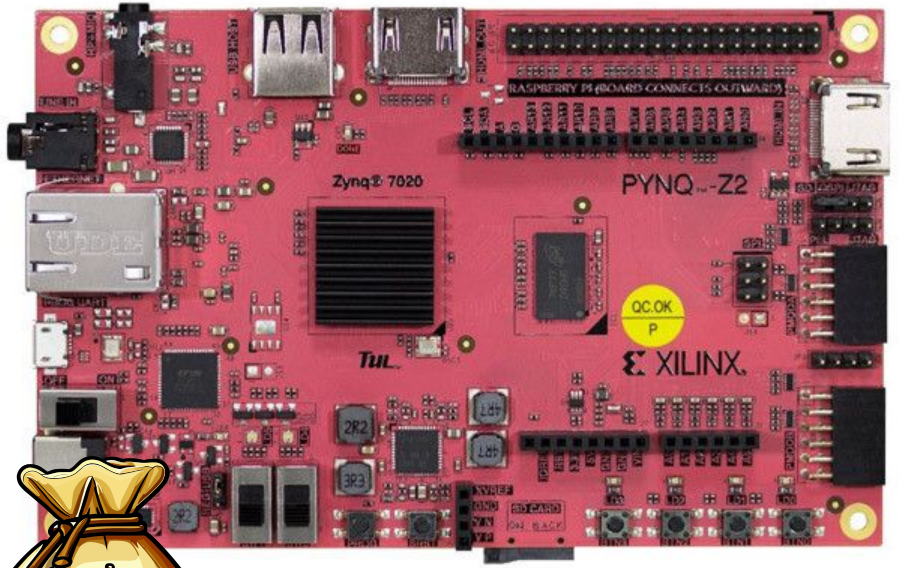
Source Nanex

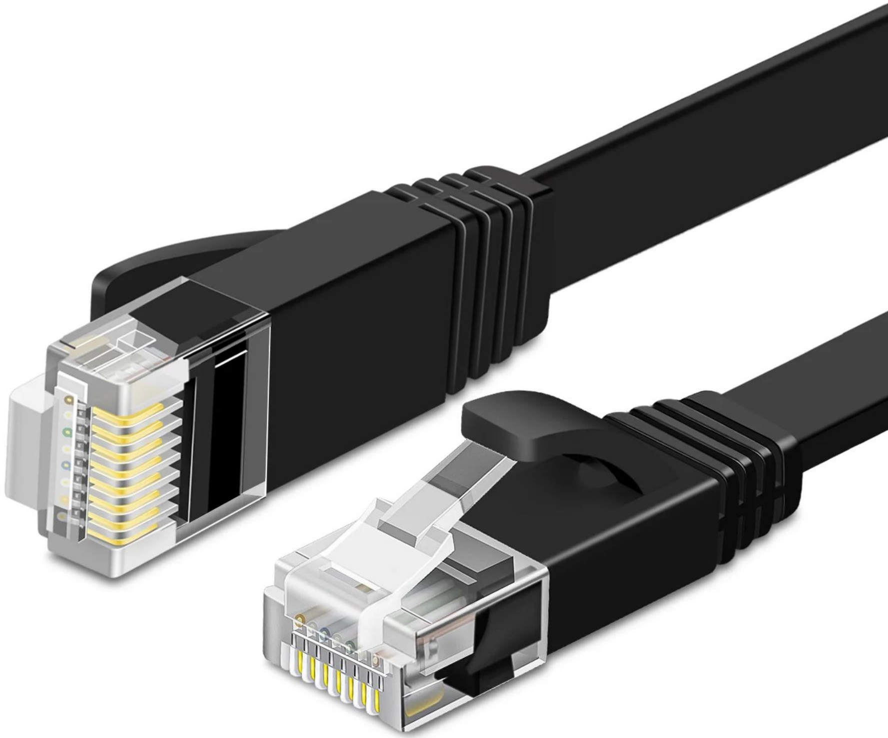


Buy		Sell
	2173.00	
	2172.75	
	2172.50	
	2172.25	
	2172.00	
	2171.75	
	2171.50	
	2171.25	
	2171.00	
	2170.75	48%
	2170.50	10.7K
	2170.25	1203
	2170.00	1642
	2169.75	1401
	2169.50	1266
	2169.25	1119
	2169.00	1036
	2168.75	913
	2168.50	941
	2168.25	843
	2168.00	290
477	2167.75	
1038	2167.50	
950	2167.25	
1349	2167.00	
1181	2166.75	
1559	2166.50	
1181	2166.25	
1220	2166.00	
1124	2165.75	
1285	2165.50	
11.4K	2165.25	
52%	2165.00	
	2164.75	
	2164.50	
	2164.25	
	2164.00	
	2163.75	
	2163.50	

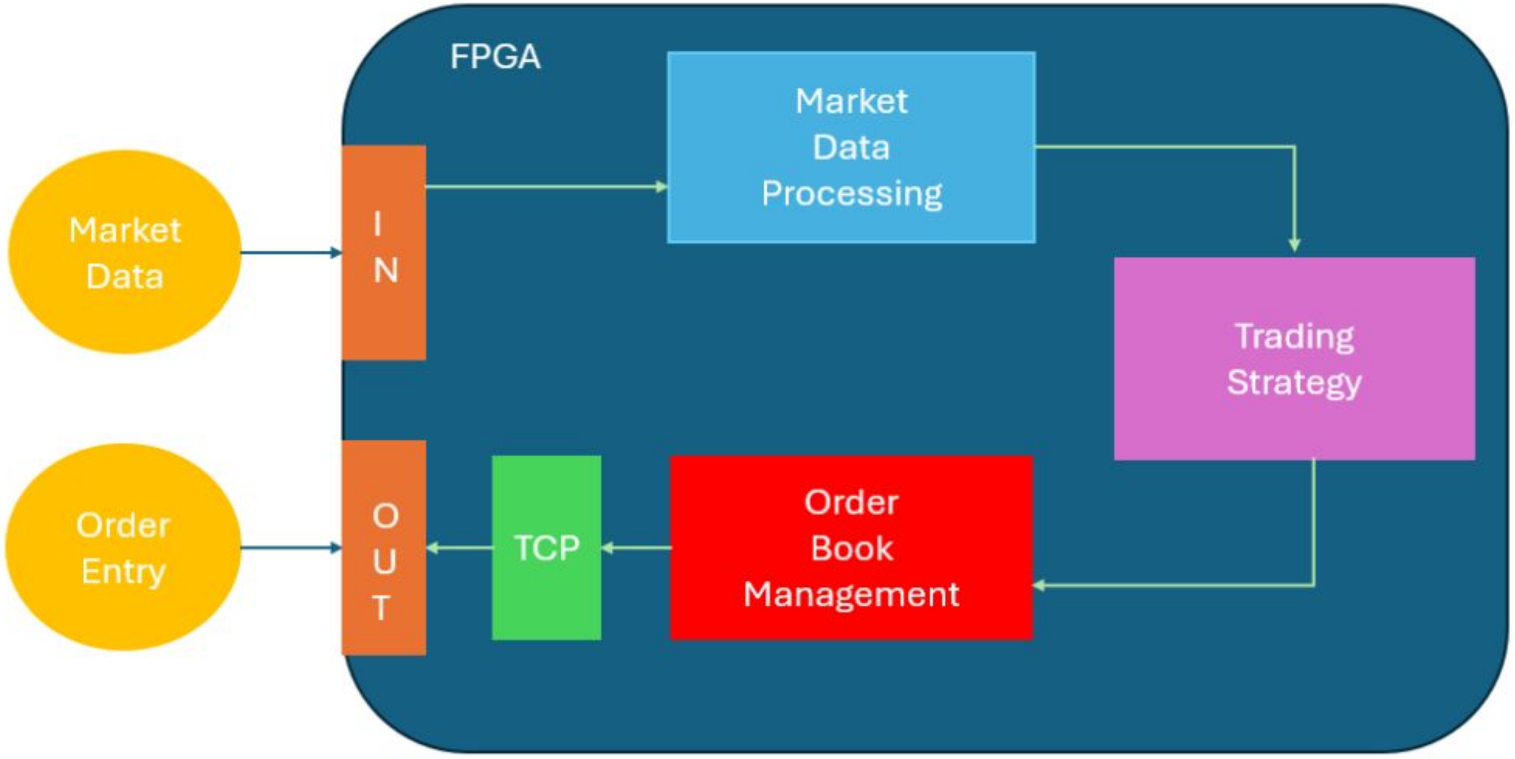
Source optimusfutures

Why? How?


Zynq® 7020
PYNQ-Z2
XILINX.
RASPBERRY PI (BOARD CONNECTS OUTWARD)
QC.OK

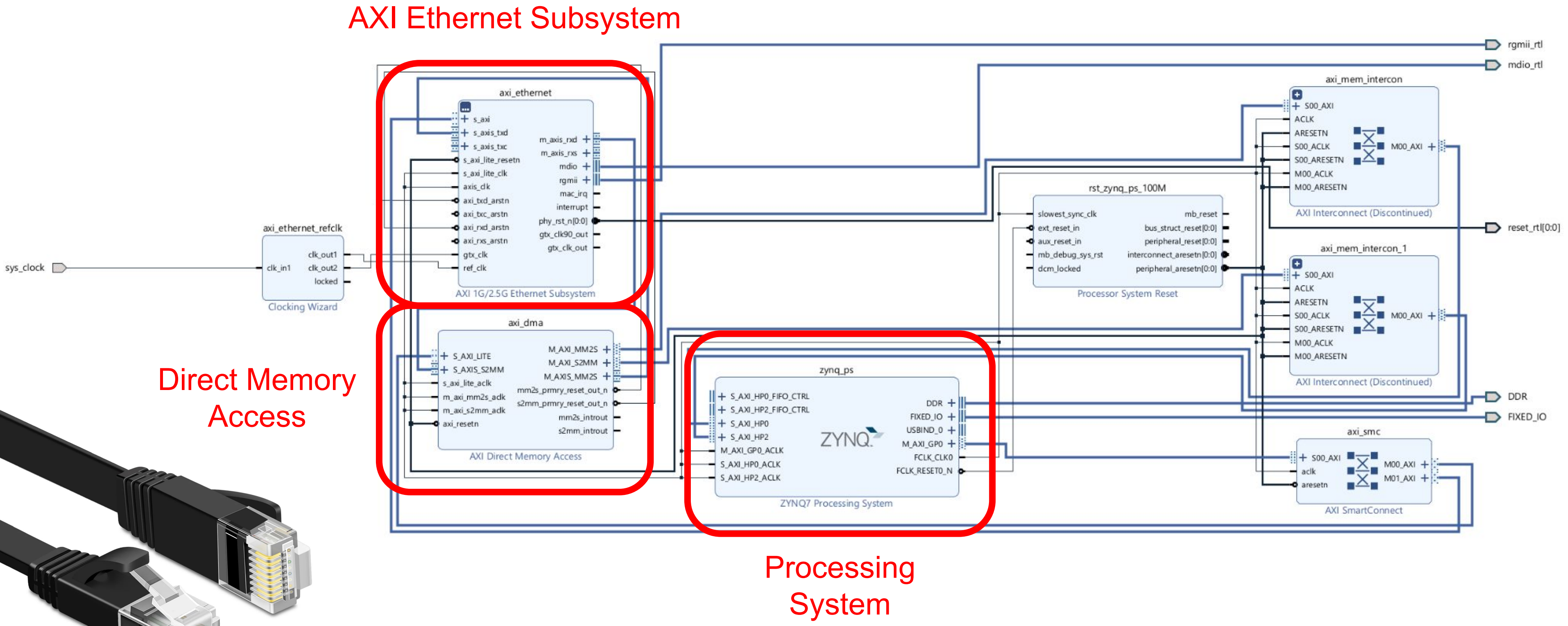


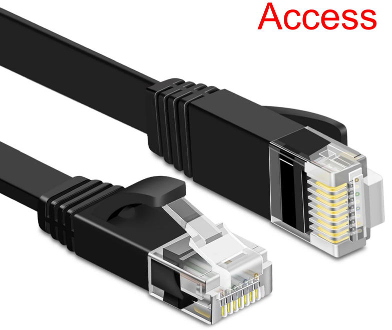
Objectives


FPGA
Market Data
I N
Market Data Processing
Trading Strategy
Order Entry
O U T
TCP
Order Book Management

So Far… Literature Review

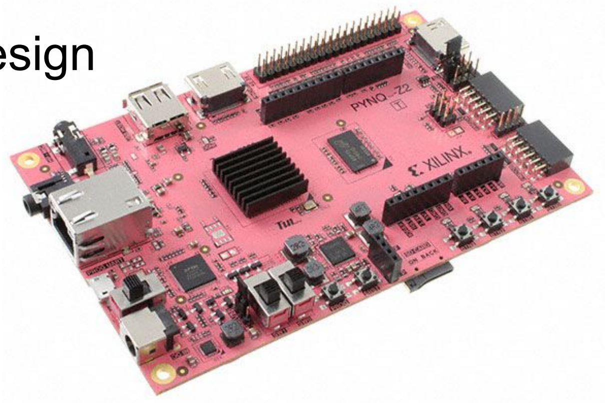
So Far… Ethernet





So Far… Ethernet

- Used a repository of a Taxi transport library that facilitates data transport and interfacing via Ethernet.
- Was able to generate the .bit file using the example design for ZCU106.

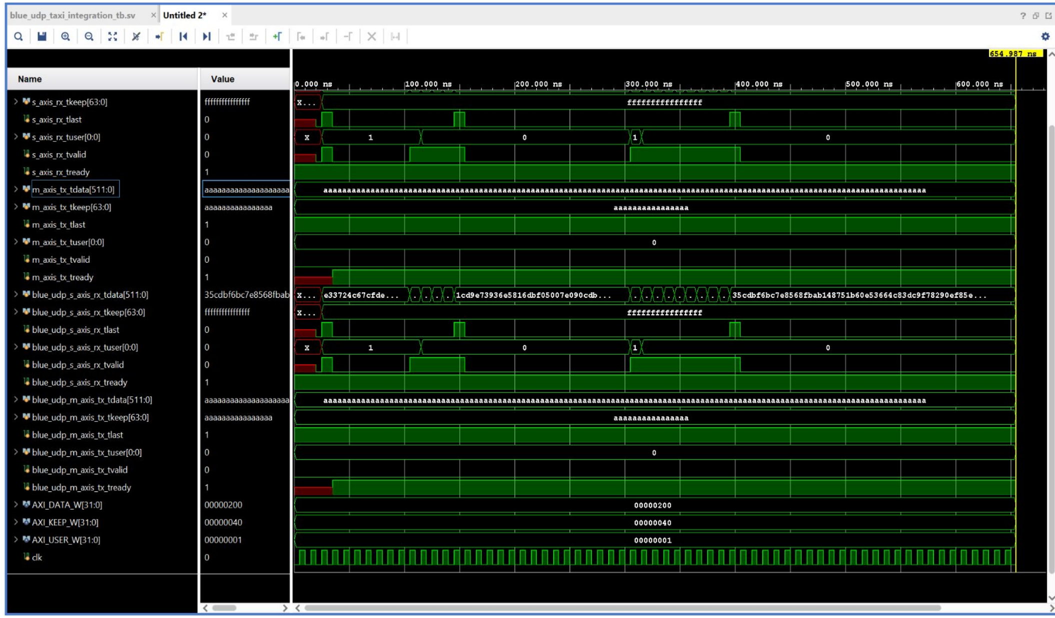


What's next:

- Adapt the example design for ZCU106 for the PYNQ-Z2 board.
- Use that .bit file to program the FPGA on the PYNQ-Z2 board.

Integrating BlueUDP core + Taxi-like AXI-Stream interface

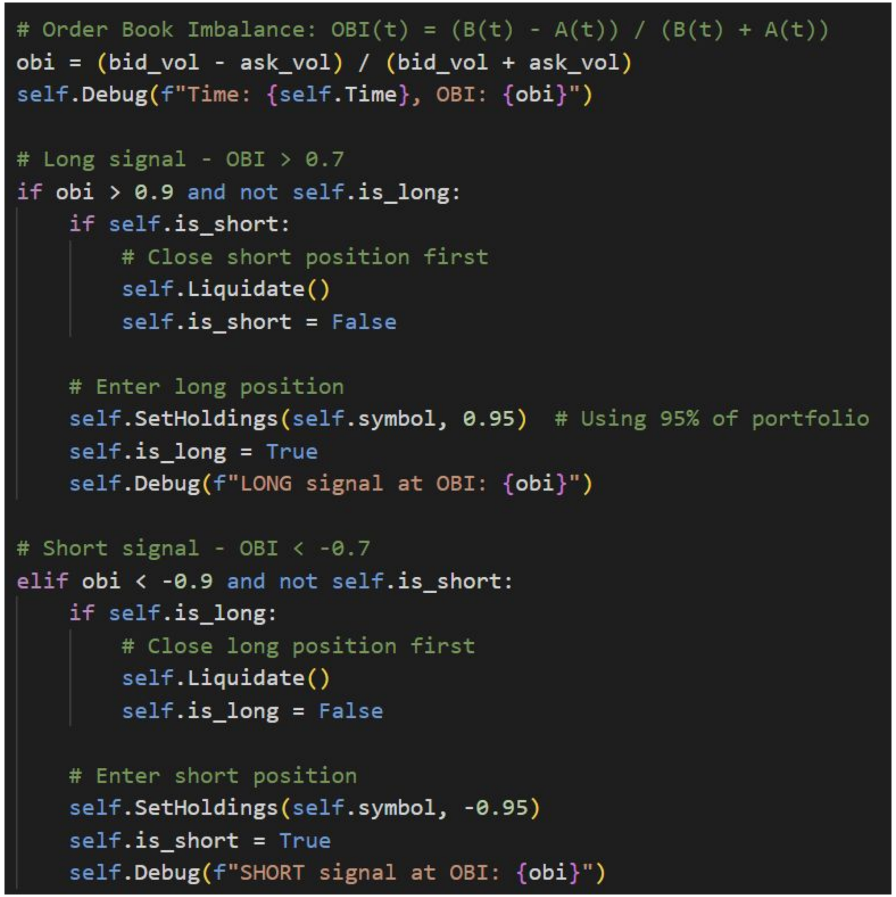
- Compiled Bluespec SystemVerilog (BSV) source code from the blue-udp repository
  - Represents the core UDP/IP/Ethernet network processing unit.
- Merged the AXI-Stream interface (BlueUDP core) to a generic AXI-Stream interface
  - Simulating how it would connect to an Ethernet MAC (Taxi)
- Sends AXI-Stream packets into the adapter (simulating data from a MAC)
  - Monitors for AXI-Stream packets coming out of the adapter (simulating data going to a MAC)



Name	Value
s_axis_rx_tkeep[63:0]	ffffffffffffffff
s_axis_rx_tlast	0
s_axis_rx_tuser[0:0]	0
s_axis_rx_tvalid	0
s_axis_rx_tready	1
m_axis_tx_tdata[511:0]	aaaaaaaaaaaaaaaaaaaa
m_axis_tx_tkeep[63:0]	aaaaaaaaaaaaaaaa
m_axis_tx_tlast	1
m_axis_tx_tuser[0:0]	0
m_axis_tx_tvalid	0
m_axis_tx_tready	1
blue_udp_s_axis_rx_tdata[511:0]	35cdbf6bc7e8568fbab
blue_udp_s_axis_rx_tkeep[63:0]	ffffffffffffffff
blue_udp_s_axis_rx_tlast	0
blue_udp_s_axis_rx_tuser[0:0]	0
blue_udp_s_axis_rx_tvalid	0
blue_udp_s_axis_rx_tready	1
blue_udp_m_axis_tx_tdata[511:0]	aaaaaaaaaaaaaaaaaaaa
blue_udp_m_axis_tx_tkeep[63:0]	aaaaaaaaaaaaaaaa
blue_udp_m_axis_tx_tlast	1
blue_udp_m_axis_tx_tuser[0:0]	0
blue_udp_m_axis_tx_tvalid	0
blue_udp_m_axis_tx_tready	1
AXI_DATA_W[31:0]	00000200
AXI_KEEP_W[31:0]	00000040
AXI_USER_W[31:0]	00000001
clk	0

Trading Logic :D

- Basic prototyping the Order Book Imbalance (OBI) core idea on Quantconnect
  - `TickType.QUOTE` Using Tick Quote data
- Problems?
  - Low Portfolio Turnover >100%
  - Low order amount for HFT strategy
- Will refine strategy after Ethernet is finalized
  - Will switch sources to synthetic data to simulate a full **depth 5** order book



```python
# Order Book Imbalance: OBI(t) = (B(t) - A(t)) / (B(t) + A(t))
obi = (bid_vol - ask_vol) / (bid_vol + ask_vol)
self.Debug(f"Time: {self.Time}, OBI: {obi}")

# Long signal - OBI > 0.7
if obi > 0.9 and not self.is_long:
    if self.is_short:
        # Close short position first
        self.Liquidate()
        self.is_short = False

    # Enter long position
    self.SetHoldings(self.symbol, 0.95)  # Using 95% of portfolio
    self.is_long = True
    self.Debug(f"LONG signal at OBI: {obi}")

# Short signal - OBI < -0.7
elif obi < -0.9 and not self.is_short:
    if self.is_long:
        # Close long position first
        self.Liquidate()
        self.is_long = False

    # Enter short position
    self.SetHoldings(self.symbol, -0.95)
    self.is_short = True
    self.Debug(f"SHORT signal at OBI: {obi}")
```

What Now?

- ❑ Ethernet Receive and Transmission (In Progress)
- ~~✓ Trading Strategy Logic~~
- ❑ FAST UDP data processing
- ❑ Order book keeping
- ❑ Integration to SystemVerilog
- ❑ Simulation in Vivado
- ❑ Deployment onto PYNQ-Z2 board (If Time)

Conclusion

- Ethernet connection is a BIG hurdle and a bottleneck in the project.
- It’s why we approached the problem from various perspectives.