#### ••••

#### **McKay Brothers**

# Moving beyond the false dichotomy of FPGA solutions

April 30, 2021

Mike Schonberg

#### Definitions

- Buy: pure commercial off-the-shelf (COTS)
  - Purchase, configure, deploy
  - No custom HDL required (VHDL, HLS, etc.)
- Build: any solution that requires writing HDL

. . .

- In-house
- 3<sup>rd</sup> party

#### What constitutes "build"







# FPGA at McKay/Quincy

#### **Evolution of McKay/Quincy**

- Startup : building products and services
- Rapid growth and competition: focus on expansion and optimization
- Steady growth: continuing optimization

#### **FPGA Approach**

- Off-the-shelf (COTS): rapid deployment with minimal effort
- Custom 3<sup>rd</sup> Party FPGA: slow development requiring minimal inhouse expertise
- In-house development: requires building an FPGA team



#### Start Up Phase

#### Commercial off-the-shelf FPGA solutions



#### **Off the Shelf FPGA Solutions**

- Quincy: FPGA for QED Feed handlers
  - Build vs Buy?
  - Purchase FPGA where possible.
  - Software for backup for most exchanges

#### Pros and Cons of COTS

- Advantages
  - Allows us to focus on our core competencies
  - Time to market
  - Vendor support and QA
  - Frequent improvements and new offerings
- Disadvantages
  - No competitive advantage
  - One size does not always fit all
  - Often less performant that built-to-purpose solutions

# Rapid Growth and Competition

#### **Outsourced custom FPGA solutions**

8



#### *Necessity Competition is the Mother of Invention*

- Competition demands continuous improvement
- Options:
  - Pressure vendors to improve their offerings
  - Purchase faster hardware
  - Outsource or build custom FPGA solutions

#### **Outsourcing Requirements**

- Existing FPGA platform
  - At the time (circa 2013) PCIE FPGA boards
  - Today FPGA enabled L1 switches
- Significant latency reduction
- Improve quality of service/new features

# Example: Custom FPGA Traffic Policer/Shaper

- Off-the-shelf options (at the time):
  - Source of significant latency
  - Shaping based on average utilization not policing based on IFD
  - Deep fixed-sized buffers
- Custom FPGA solution

11

- Much lower latency
- Include proprietary IP
  - Policing to a more exacting standard
  - Shallow, configurable buffering
  - Tight coupling with other components

#### **Custom FPGA Solutions Pros and Cons**

- Advantages
  - We continue to focus on improving our products
  - Free to select best partner
  - Integrate proprietary IP
- Disadvantages
  - Contracts are challenging
  - Communications overhead
    - Writing good specifications is hard
    - Change orders and misunderstandings are costly
  - Acceptance: QA is a cooperative effort
  - Evolution and support is difficult

#### **Best Practices for Custom FPGA**

- Choose a commercial product as a starting point
- Write complete and detailed specifications
- Keep the design simple
- Plan for integration and deployment in advance
  - Diagnostics and counters
  - Management interface(s)

## **Evolution of Custom Solutions**

- As McKay commissioned custom devices...
- Vendors developed similar products
- Hybrid solution



. . .

## **Hybrid Solutions**

- Challenges
  - Requires compromise on both sides
  - May require hardware upgrades
  - Migration/integration
- Benefits
  - Immediate access to improvements
  - Better support
  - Retain advantages of proprietary IP
  - Reduces vendor's overhead

#### **Building an Internal Team**

#### Building our own

16



## Why Build an Internal Team

- Agility
  - Streamlined communication
  - Address rapidly evolving requirements
  - Iterative development
  - Shorter development time
- Protect IP
- More effective interface with 3<sup>rd</sup> party developers
  - Customize off-the-shelf solutions
  - Collaborative custom/hybrid solutions

#### Hybrid Solutions: Best of Both Worlds

- Integrate in-house IP with existing commercial solutions
- Better IP protection
- Easy communication
- Continue to benefit from vendor improvements
- We focus on our core competencies and benefit from 3<sup>rd</sup> party expertise



## Example Project: NASDAQ ITCH Splitter

- Based on a client's requirements
  - Split the NASDAQ Feed into multiple streams for CPU processes
    - Throughput and determinism more important than latency
  - Alphabetic splits

19

- Some instruments in all streams (stock lists)
- Include QED data

#### **ITCH Splitter**

- Exablaze Fusion
- Up to 47 output ports
  - Can be grouped
- Up to 16 splits based on alphabetic ranges
- Stock lists: lists of names that can be directed to any group of ports
- QED (or any other ethernet stream) can be injected into any port group.
- Latency: 170 ns\* port to port:

\* Not a STAC benchmark

#### Lessons Learned How I learned to stop worrying and love FPGA Development

- Software prototypes are very helpful
- Keep the FPGA component simple
  - ITCH splitter implements generic packet steering
  - Configuration dictates the precise behavior
- Plan early for the supporting software
  - Monitoring, configuration, logging, etc.



Contact us to learn more:

Mike.Schonberg@mckay-brothers.com