PCB-level modularity exemplified by the Serenity-S1 ATCA board

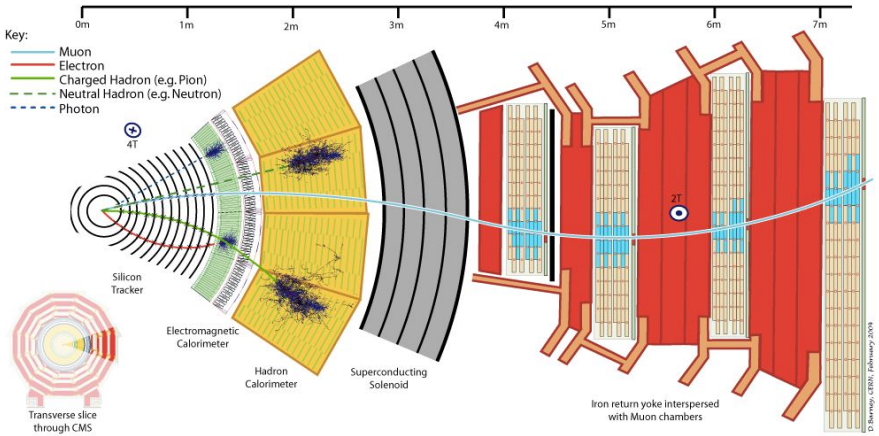
Torben Mehner on behalf of the Serenity collaboration

SEI Tagung 2023, KIT

17th April, 2023

High Luminosity LHC – CMS

- HL-LHC operation will start in 2029 with 5-7 times increased luminosity
- Two-level trigger
  - 40 MHz bunch crossing
  - 750 kHz after L1-trigger
  - 7.5 kHz after High Level Trigger (HLT)
- Necessary to include tracking information at first level of triggering



https://indico.desy.de/event/28053/contributions/97454/attachments/63827/78124/MT_DTS_2021_CMS_TT.pdf

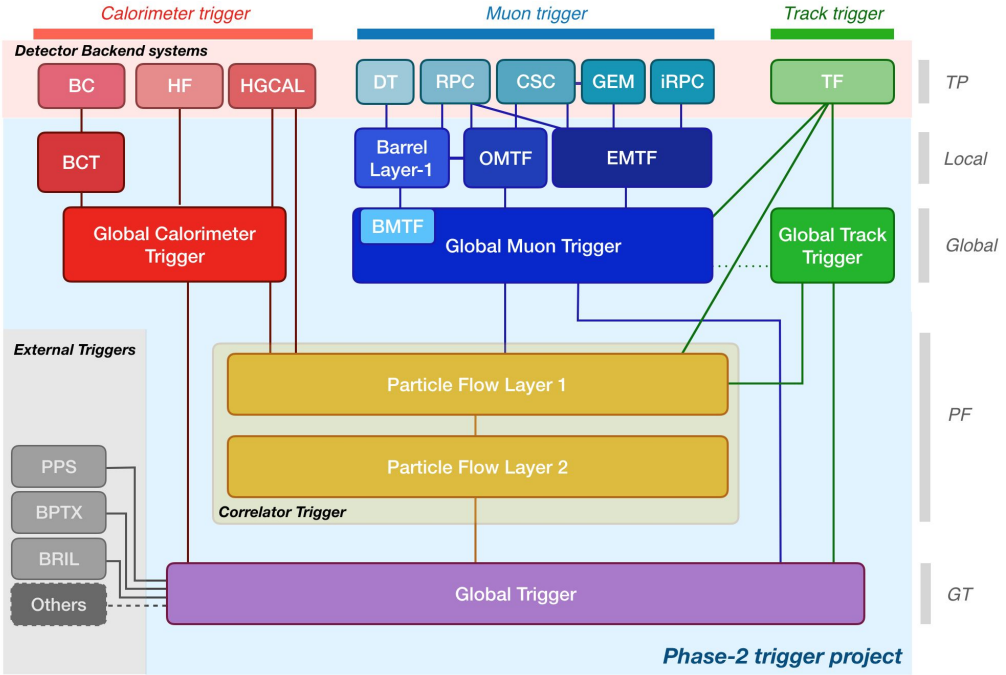
CMS Level-1 Trigger in HL-LHC

4 complementary trigger types

- Calorimeter Trigger
- Muon Trigger
- Track Trigger
- Particle Flow Trigger

Serenity boards intended to be used in

- HGCAL
- DTC, GTT
- Correlator L1, Correlator L2
- GT
- MTD, BRIL



https://lss.fnal.gov/archive/2020/pub/fermilab-pub-20-712-cms.pdf

Serenity Evaluation Cards

Generic data processing card for various CMS applications

Advanced Telecommunications Computing Architecture (ATCA) form factor

- Up to 16 cards in a shelf with dual-star topology
  - 2 hub cards (DTH)
  - 14 node cards (DTC)
- Backplane access to
  - LHC clocks
  - Trigger and Timing Control and Distribution System for Phase-2 (TCDS2)
  - Ethernet
- Intelligent Platform Management Controller (IPMC) for communication between card and shelf

https://indico.desy.de/event/28053/contributions/97454/attachments/63827/78124/MT_DTS_2021_CMS_TT.pdf

Serenity Evaluation Cards

Serenity-A

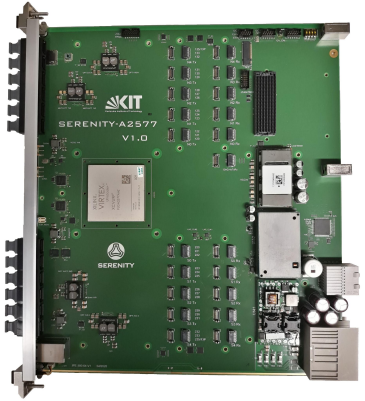
- Single FPGA Virtex Ultrascale+ VU9P or VU13P
- Up to 120 optical links with up to 25 Gbps

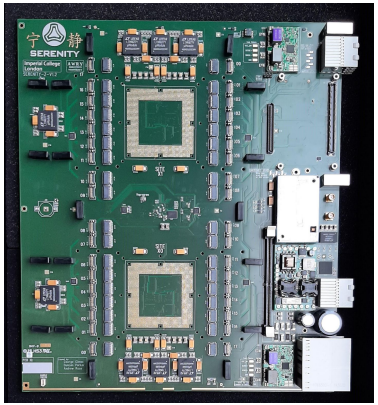
Serenity-Z

- Dual FPGA Kintex Ultrascale+ KU15P or Virtex Ultrascale+ VU7P, VU9P or VU13P
- Up to 96 optical links per FPGA with up to 25 Gbps

Objectives

- Develop production-ready cards from common infrastructure
- Improve maintainability by reducing variations

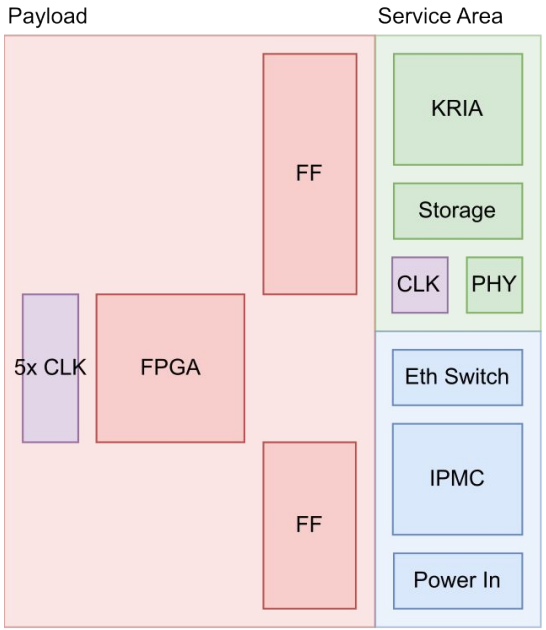




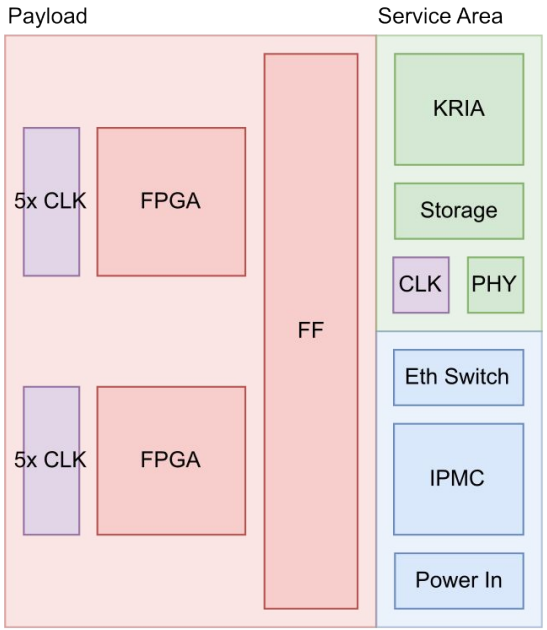
https://indico.desy.de/event/28053/contributions/97454/attachments/63827/78124/MT_DTS_2021_CMS_TT.pdf

Serenity Cards in the CMS Experiment

Serenity-S1



Serenity-D1

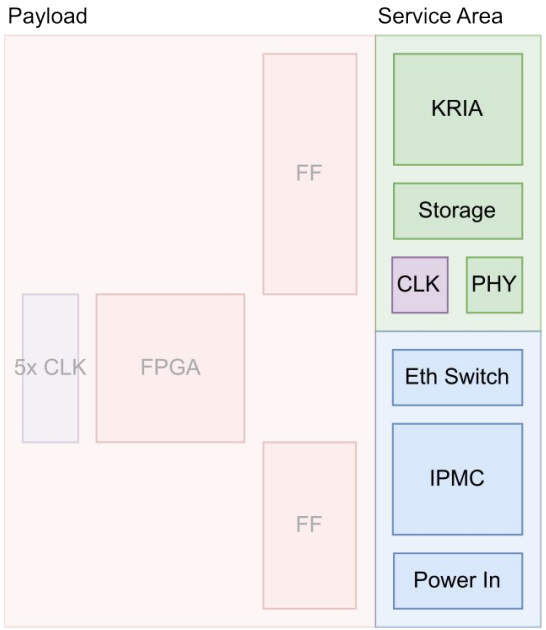


Partitioning of Serenity Cards

Karlsruher Institut für Technologie

SERENITY

Serenity-S1



- Board Infrastructure
  - Xilinx KRIA SoM
    - Zynq Ultrascale+ SoC on off-the-shelf module
  - Clock, power, PHY
  - SD, SSD
- ATCA Infrastructure
  - Backplane connectors
  - IPMC (OpenIPMC DIMM module)
  - Power input
  - Ethernet switch

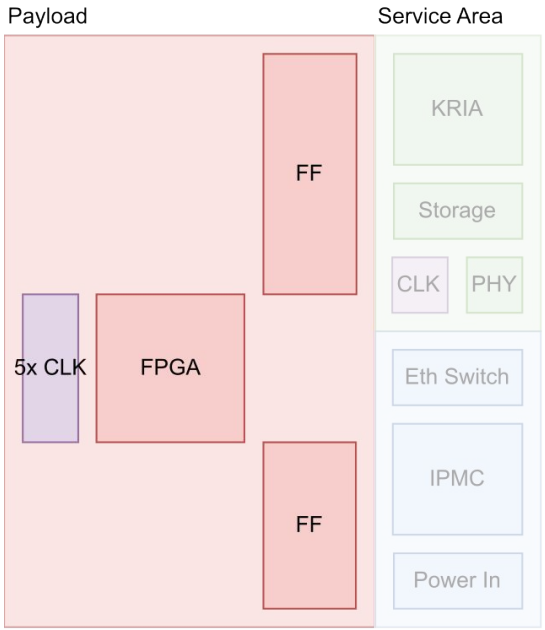
ZynqMP board management: https://iopscience.iop.org/article/10.1088/1748-0221/17/03/C03009
OpenIPMC: https://cds.cern.ch/record/2744530?ln=de

Partitioning of Serenity Cards

Karlsruher Institut für Technologie

SERENITY

Serenity-S1

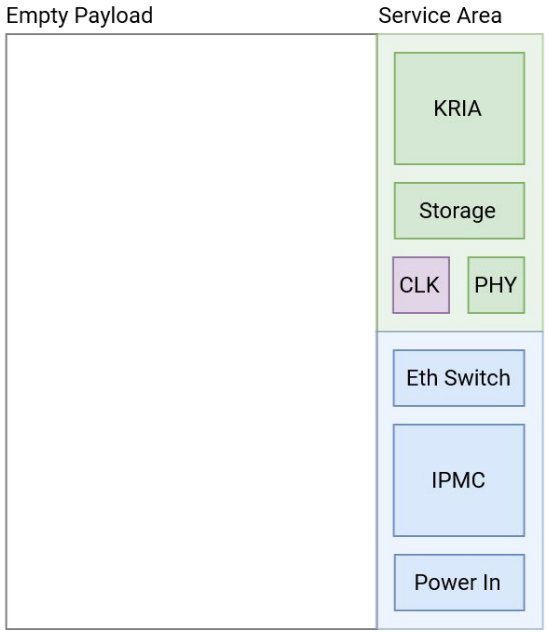


- Payload
  - Firefly + power supplies
  - FPGA + power supplies
    - Virtex Ultrascale+ VU9P or VU13P
  - Clocks
    - Microchip ZL30274

Modularity in Serenity-S1

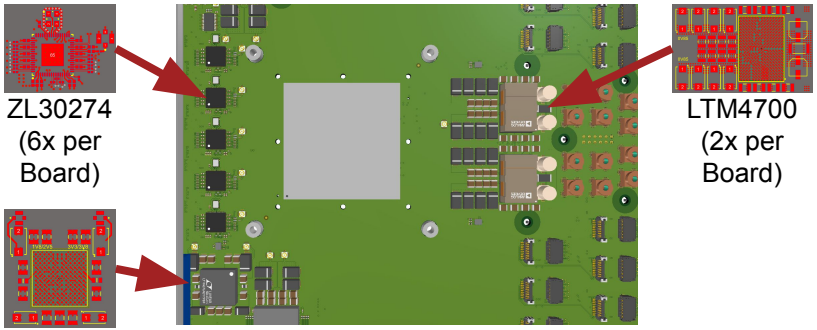
Serenity-Platform

- Large module fulfils repeating task in ATCA cards



Reuse Blocks

- Repeating small modules used across the board



Reuse repeating components within a project and in multiple projects

What is a Module?

Smallest reusable group of components

- To fulfil a repeating task
- To make a part work (“Copy of” typical application diagram in the datasheet)

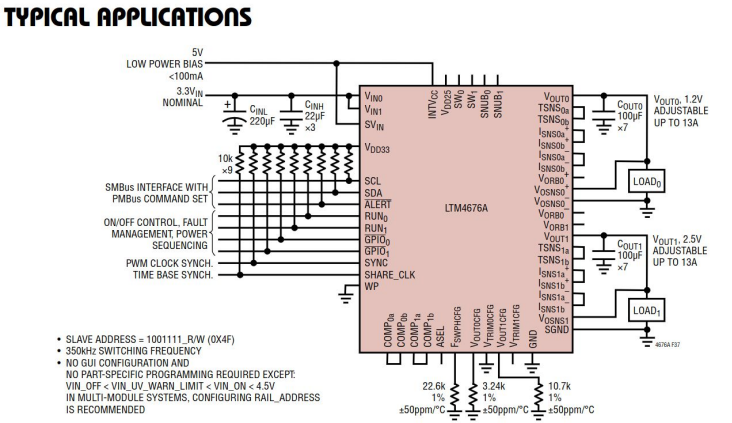
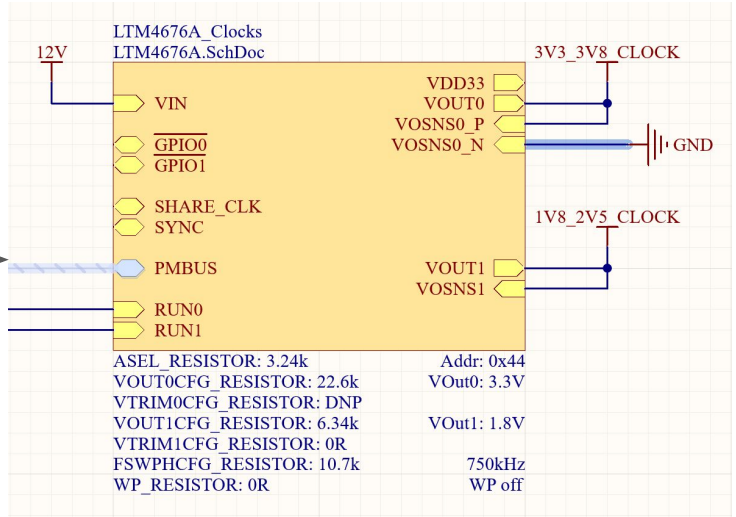


Figure 37. 13A, 1.2V and 2.5V Outputs Generated from 3.3V Power Input and Providing I²C/SMBus/PMBus Serial Interface



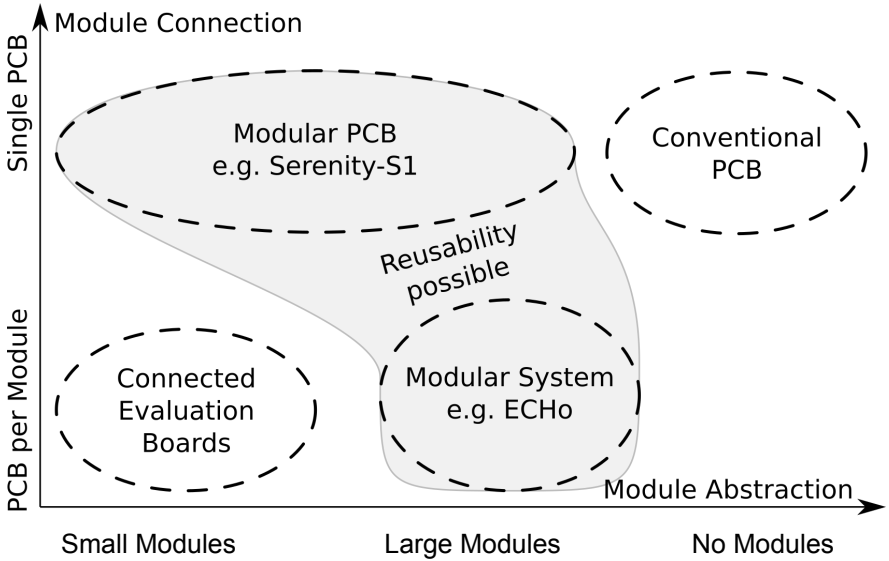
Modularity

Modular PCB

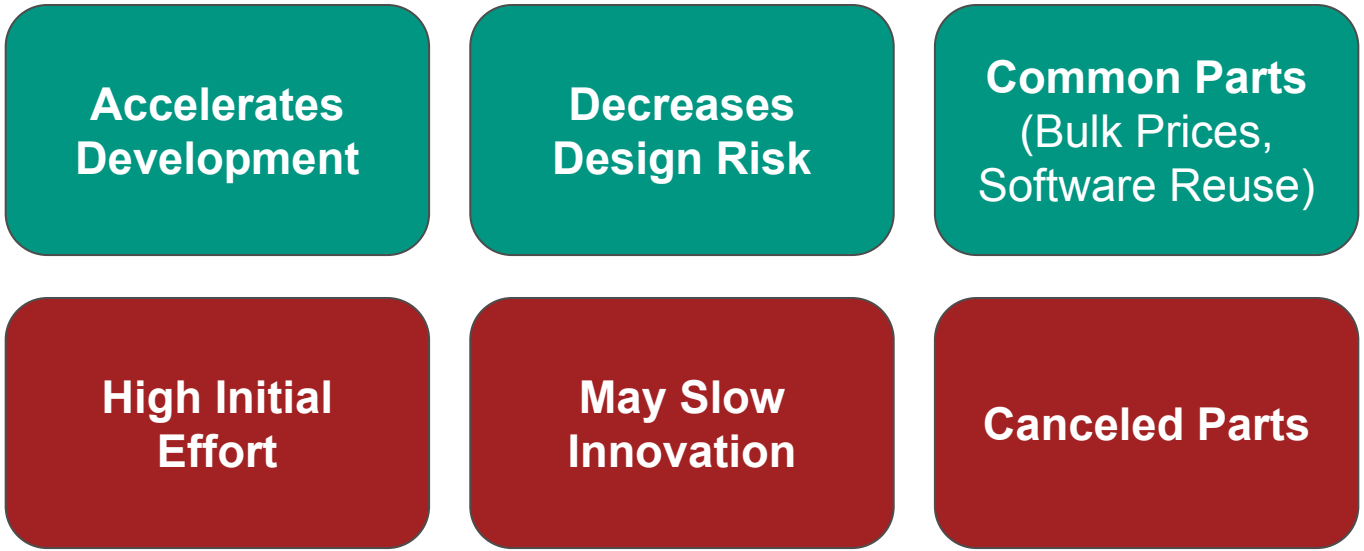
- Relies on EDA functionality
- Best connections and flexibility
- Large PCBs are harder to fabricate

Modular System

- Multiple PCBs with single function
- Signal attenuation and reduced flexibility through connectors
- Design parallelization possible
- Production for off-the-shelf reuse



Reasons for Design Reuse



Modularity in other Fields

Software uses programming languages and IC design uses hardware description languages (HDLs)

- Hierarchy to reduce complexity
- Reuse from libraries

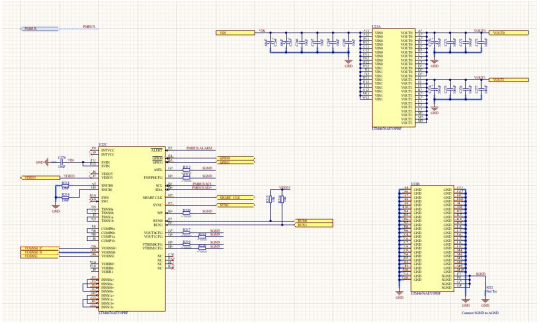
There are PCB-Level HDLs that replace schematic entry, but they tend to be very short lived:

- Polymorphic Blocks (for KiCAD)
- jitX (commercial, for KiCAD, Altium)

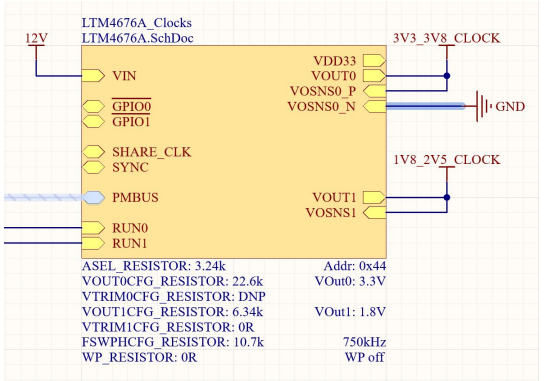
Solution:

Use software concepts in hardware design

Why do this?



When you can do this?



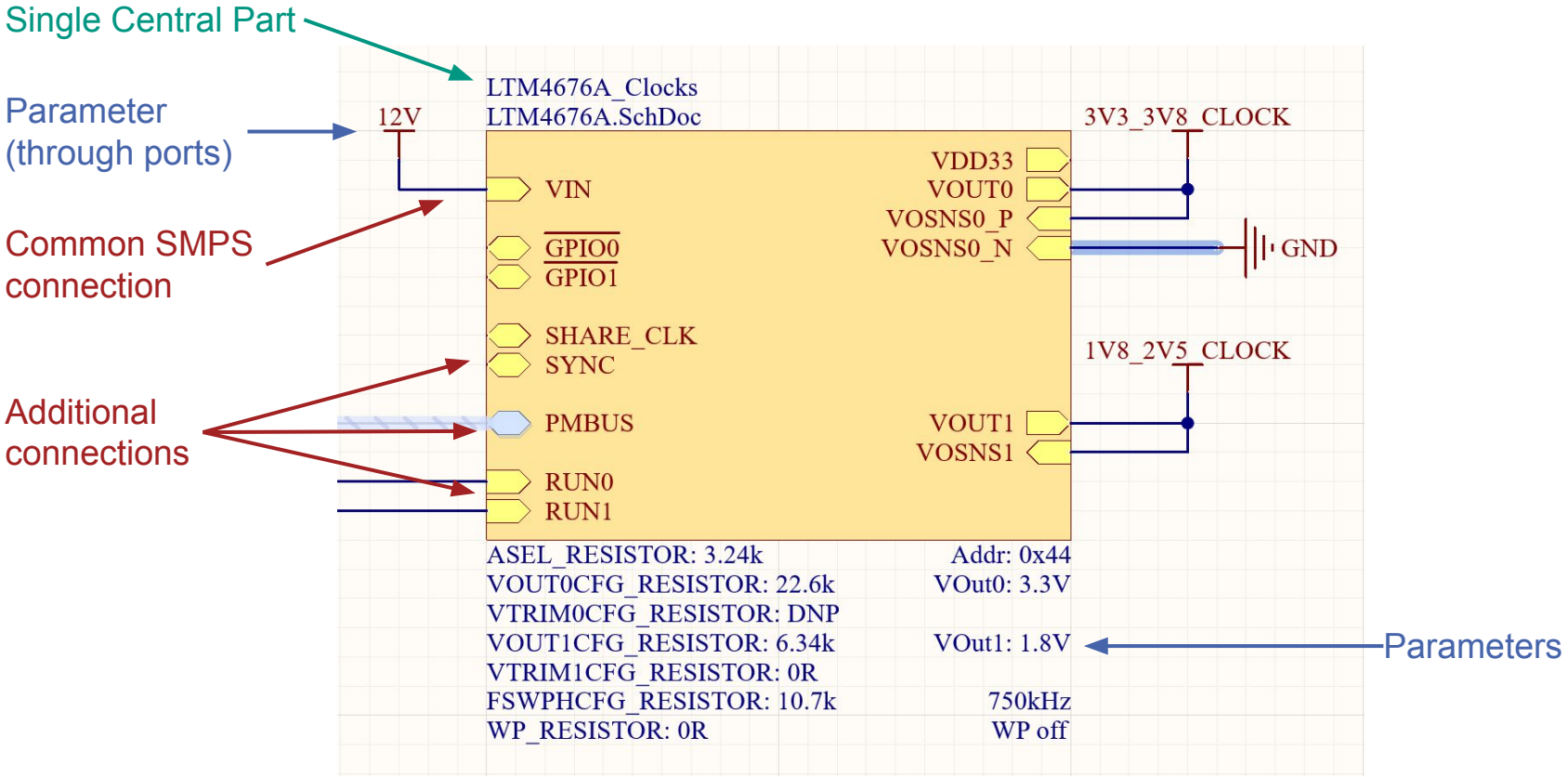
Polymorphic Blocks: https://dl.acm.org/doi/10.1145/3379337.3415860
jitX: https://www.jitx.com/

Guidelines for Modularity in EDA

- Hierarchical Design
- SOLID design principles
  - Single Responsibility
    - Single central part or
    - Single functionality by multiple parts
  - Open for additions - Closed for changes
    - Simple “Placeholder” module for e.g. LDOs, SMPS
    - Extendable with additional connections
  - Interface Segregation Principle
    - Only have absolutely needed functionality in a module
    - Use parameters for potentially changing values
- GIT integration for collaboration

SOLID design principles: https://en.wikipedia.org/wiki/SOLID

Guidelines for Modularity in EDA



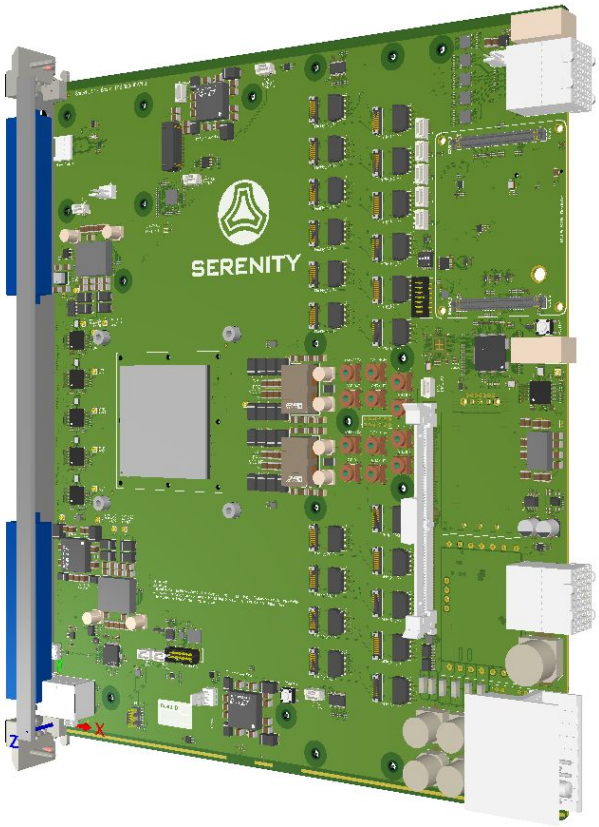
Conclusion

Conclusion

- Serenity production cards converge to common, unified structure with little variation
- Modularity can increase development speed of both hardware and software by using principles from software engineering

Outlook

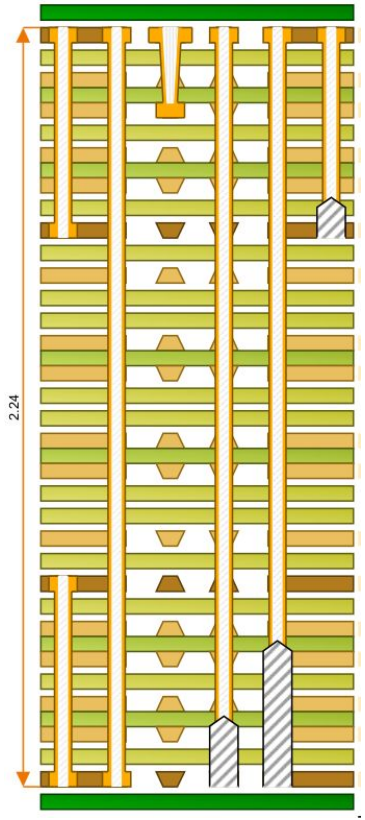
- First 12 Serenity-S1 boards are now being fabricated
- Results are expected in September
- Modularity can ease the design of new systems by providing a framework and reusable common modules

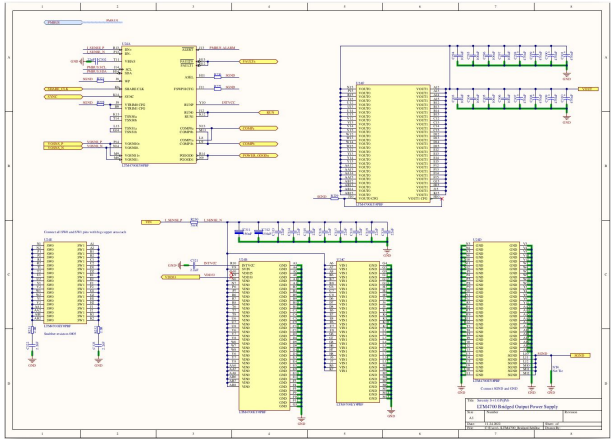


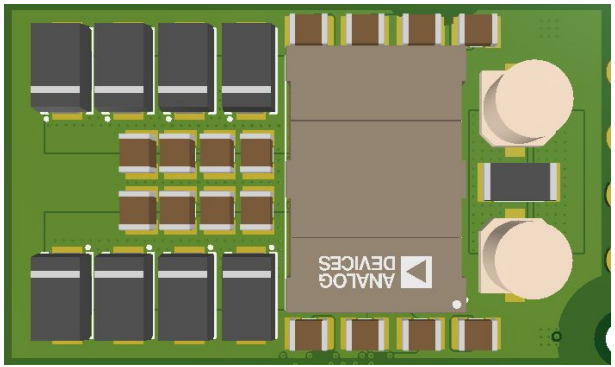
Backup slide section

Reusable Assets

- Library Parts
- Schematics
  - Frame Template
  - Project Rules
  - Modules
- Layout
  - Stack-Up & Design Rules
  - Modules
  - Outlines (e.g. PCIe, ATCA)







Serenity Platform

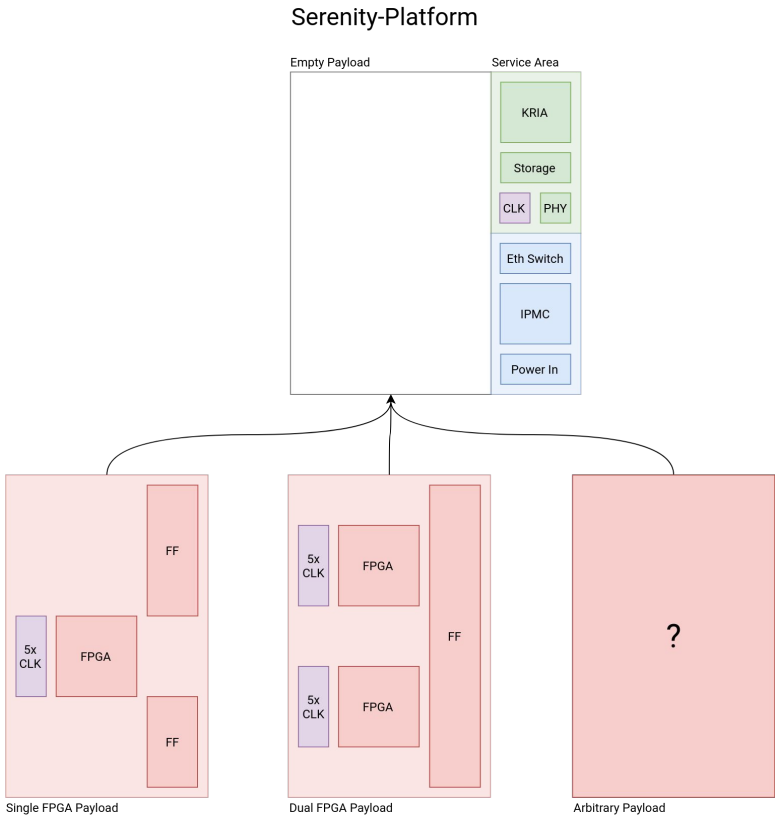
Partitioning leads to a platform framework with interchangeable payload area

Benefits

- Reuse service area to accelerate development of ATCA card with different payloads (hardware & software)
- No connectors and therefore no restrictions or attenuation from connectors

Drawbacks

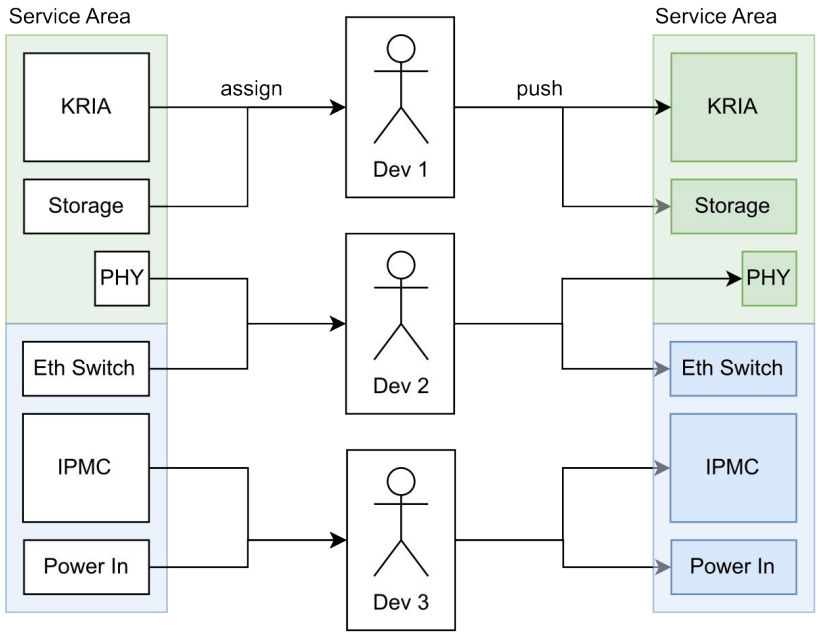
- Initial design effort is higher
- IO-limits due to platform framework (limited to node usage in dual-star backplane)



Cooperative Schematic Design Process

- Initial design
  - Build empty “skeleton” with interfaces
- In-Depth design
  - Distribute design sheets to multiple engineers or reuse sheets
  - Each engineer can work independently on sheets
  - Have steady chat and weekly meetups to be able to change structure if necessary
- Review

Cooperative Schematic Design Process



Cooperative Layout Design Process

Cooperative layout design process with single git-controlled layout file

1. Assign work packets
2. Get local copy
3. Work on local, self-contained design area
4. Update main branch
   a. Pull latest version from repository
   b. Copy progress from local copy to main branch
   c. Push main branch