An Optical SpaceFibre Testbed for Transceiver Evaluation &

Validation of a Design-Time Configurable Router

Malte Bargholz, Wolfgang Rönninger, **Felix Siegle**

15/10/2022

Who we are

Malte Bargholz
Junior Professional in On-Board Data-Handling
European Space Agency

- Former trainee at ESA, now staff
- Interested in payload processing, high-speed SoC design, ML/AI acceleration
- Background in embedded system development, both HW and SW

Wolfgang Rönninger
Young Graduate Trainee in Microelectronics
European Space Agency

- Former trainee at ESA, now digital designer at Axelera AI, Netherlands
- Interested in digital and ASIC design, GNSS processing, AI/ML acceleration
- Background in electrical engineering with digital design specialization

Felix Siegle
Payload Data-Handling Engineer
European Space Agency

- Staff in the On-Board Computer and Data Handling Section, ESA-ESTEC
- R&D in the field of mass memory units, high-speed links, and future data handling architectures
- Project support (CO2M, Biomass, Solar Orbiter, Euclid…)

Motivation

- Current trends in on-board data-handling
  - More data from payloads: multi-spectral image sensors, SAR, … (multiple Gbit/s)
  - Faster downlinks (100 Mbit/s → >1 Gbit/s)
  - Harness reduction, higher integration density

→ **Future DH systems have to handle multiple multi-Gigabit data sources and sinks with minimal harness**

- Optical high-speed links are a promising option for future on-board communication
  - High-performance (> 10 Gbit/s) and good signal-integrity over long distances
  - Favorable SWaP parameters:
    - Low Power (typically ~100mW per transceiver channel)
    - Lightweight cabling and connectors
  - Built-in fault isolation
  - Wide-range of protocols due to SerDes: SpaceFibre, Serial RapidIO, WizardLink, many commercial protocols…

Agenda

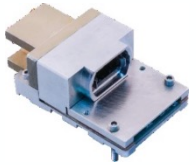
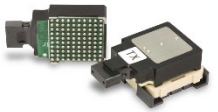
Goal: Evaluate optical high-speed links in the context of future on-board data-handling systems

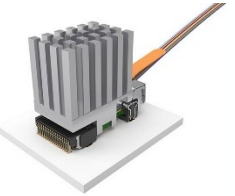
1. Create representative testbed that mimics future optical-based on-board data-handling architecture
    - Using optical high-speed transceivers from different manufacturers
    - Using modern protocol, currently used → SpaceFibre
    - Configurable data-sinks and sources with variable packet data rate
    - Configurable link parameters, traffic criticality (QoS), …
    - Observable link statistics, error-rate, signal-integrity

2. Evaluate the testbed by trying to emulate a realistic data handling unit
    - SpaceFibre router as example system providing high-speed data sources/sinks
    - Design-time configurable routing matrix further improves SWaP parameters

Optical transceivers (for space)

- Broad selection of transceivers viable for future missions, ranging from COTS to fully qualified components



Product	DataStar SPACE	LightAble / SpaceAble	OptoFire	FireFly
Manufacturer	Glenair	Smiths Interconnect (former: Reflex Photonics)	APITech	Samtec
Type	RHBD	COTS/RHBD	RT	COTS
No. channels	4 (full-duplex)	4 (full-duplex) 12 (half-duplex)	4 (full-duplex)	4 (full-duplex) 12 (half-duplex)
Link-rate	10Gbit/s	10 or 28 Gbit/s	10 Gbit/s	10 or 28 Gbit/s
Radiation Tests	Proton, TID, Heavy-Ion	Proton, TID, Heavy-Ion	?	TID

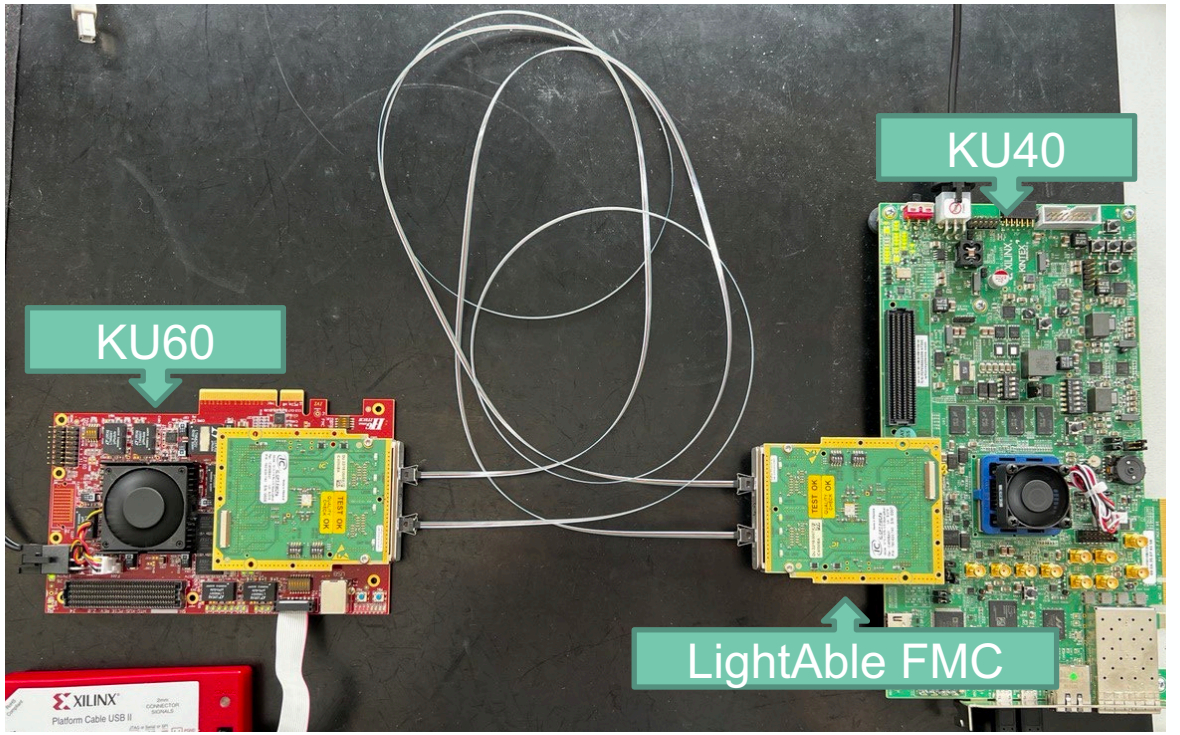
SpaceFibre Optical Transceiver Testbed

- Demonstrate SpaceFibre link(s) over optical transceivers and verify performance

Setup

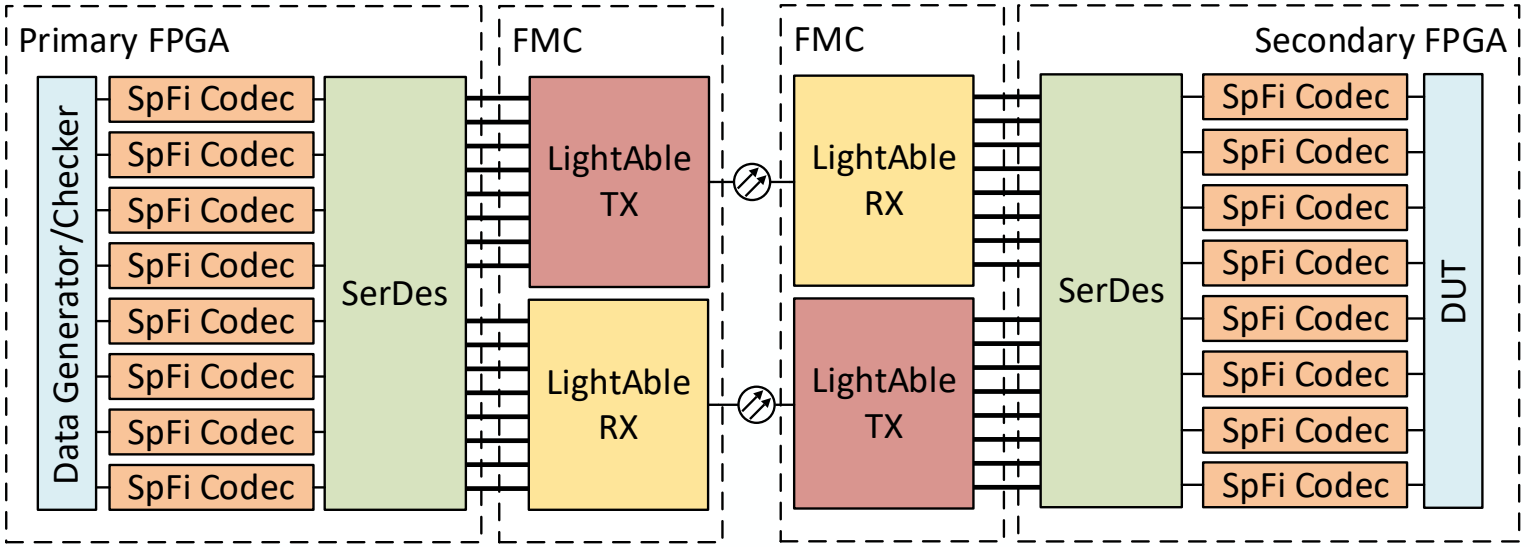
- Two FPGA board setup with Xilinx Kintex UltraScale development kits:
  - HTG-K800 (XCKU060): Data-Generator
  - KCU105 (XCKU040): Loopback

- Smiths Interconnect LightAble 10G LM series as FMC add-on card
  - Up-to 12 channel 10 GBit/s full-duplex configuration (separate TX/RX)
  - 1 meter of OM3 ribbon-fiber (up-to 300 meter is possible)



SpaceFibre Optical Transceiver Testbed: FPGA design

- FPGA design was developed based on previous SpFi lab activities
  - → re-uses SpFi Port IP core (ESA IP-Core library/Cobham Gaisler), data generator and configuration IPs
- Design in a nutshell
  - 8 separate SpFi ports with 3 virtual channels
  - Connected to SerDes macro configured with 8 TX/RX pairs in single-lane mode
  - Both boards run nearly identical designs, loop-back or data generator



SpaceFibre Optical Transceiver Testbed: **Evaluation**

- Synthesis and Implementation with Vivado 2020.1
  - Different virtual channel configurations (1, 2 and 3 virtual channels)
  - Recorded resource utilization, timing and power consumption estimation
- Bit-error rate calculation by enabling data generation and monitoring lane state through UART interface
  - Developed GRMON drivers and scripts to aid this process

```
grmon3> spfi::print_lane_error_summary
                  | Lane-04 | Lane-00 | Lane-05 | Lane-0
  --------------- | ------- | ------- | ------- | ------
  rx_err_cnt      |       0 |       0 |       0 |
  retry_cnt       |       0 |       0 |       0 |
  too_many_errors |       0 |       0 |       0 |
  rx_err_cnt_ov   |       0 |       0 |       0 |
  far_end_los     |       0 |       0 |       0 |
  timeout         |       0 |       0 |       0 |
  far_end_standby |       0 |       0 |       0 |
  far_end_reset   |       0 |       0 |       0 |
  seq_err         |       0 |       0 |       0 |
  crc8_err        |       0 |       0 |       0 |
  crc16_err       |       0 |       0 |       0 |
  frame_err       |       0 |       0 |       0 |
  protocol_error  |       0 |       0 |       0 |
  fe_cap          |       7 |       7 |       7 |
  fe_los_cause    |       0 |       0 |       0 |
```

```
grmon3> spfi::print_datagen_status
                  | VC 00 | VC 01 | VC 02 | VC 03 | VC 04 | VC 05 | VC
  --------------- | ----- | ----- | ----- | ----- | ----- | ----- | ---
  tx::bw_cnt_last | 95.5% |  0.0% |  0.0% |  0.0% |  0.0% |  0.0% |  0.
  rx::bw_cnt_last |  0.0% |  0.0% |  0.0% | 95.5% |  0.0% |  0.0% |  0.
  rx::err_cnt     |     0 |     0 |     0 |     0 |     0 |     0 |
  rx::eep_cnt     |     0 |     0 |     0 |     0 |     0 |     0 |

                  | BC    |
  --------------- | ----- |
  tx::bw_cnt_last |  0.0% |
  rx::bw_cnt_last |  0.0% |
  rx::late_cnt    |     0 |
  rx::err_cnt     |     0 |
  rx1::delayed_cnt |     0 |
```

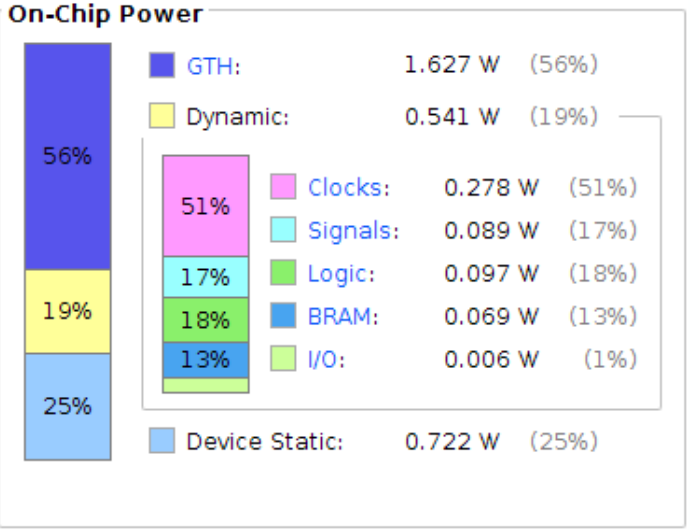
SpaceFibre Optical Transceiver Testbed: Results

- Recorded for the XCKU060
  - Utilization

	LUTs	FFs	BRAM
SpFi Port IP (1 VC)	3227 (0.97 %)	1931 (0.29 %)	3 (0.14 %)
SpFi Port IP (3 VC)	3645 (1.10 %)	2364 (0.36 %)	8 (0.37 %)
System Total	39023 (11.8 %)	30666 (4.62 %)	64 (2.96 %)

  - Power Consumption

[W]	Optical Engine	SpFi Codec	SERDES	Total
One Link	~0.1	0.047	0.203	0.353
System	~0.8	0.376	1.627	2.168 (excl. static)



SpaceFibre Optical Transceiver Testbed: **Results (2)**

- Timing
  - SpFi port IP (and system AHB bus) is clocked with SerDes TX clock
    → 10 Gb/s / 40 bit = **250 MHz**
  - Timing achieved with 0.002 ns WNS, 0.004 ns WHS
  - Critical path in data-generator

- Bit-Error Rate (BER)
  - Test-setup ran full-bandwidth test for approx. 24 hours (TBC)
    → **0 errors**
  - 8 lanes * 10 Gbit/s * 24h * 60 min * 60 sec = **6912 Tbit** transferred
  - System BER < 1e-15 with 99.9% confidence level

$$CL = 1 - e^{-N \times BER_S} \times \sum_{k=0}^{E} \frac{(N \times BER_S)^k}{k!}$$

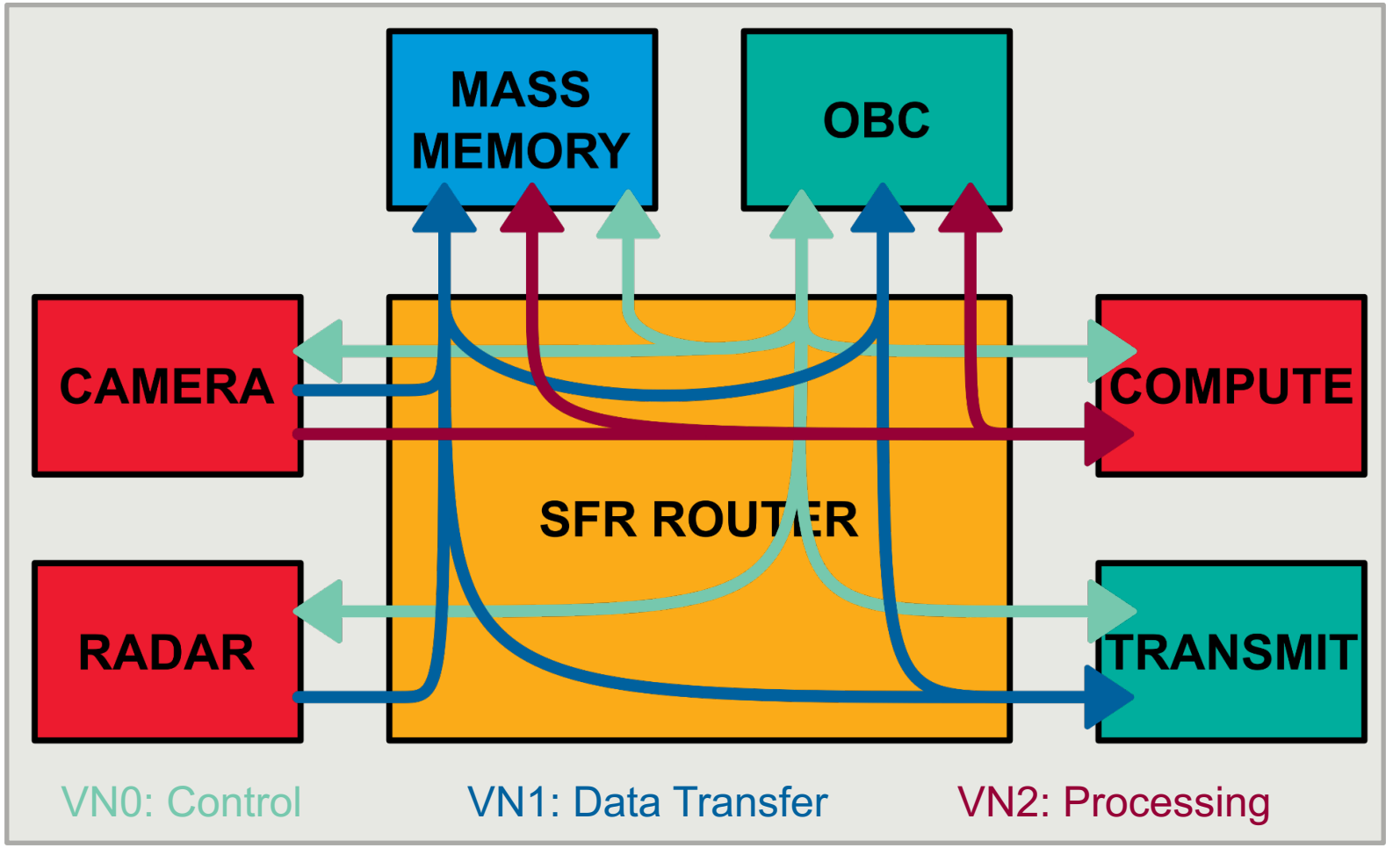
Design-Time Configurable SpaceFibre Router

Design-Time Configurable SpaceFibre Router

Goal: Emulate a realistic on-board data-handling unit with high-speed data sinks/sources

- Design-time configurable SpaceFibre Router
  - Configurable network topology → minimal, generated implementation
  - Improved isolation and security guarantee

- **Implement design-time configurable SpaceFibre router for realistic example system with optical transceiver testbed**



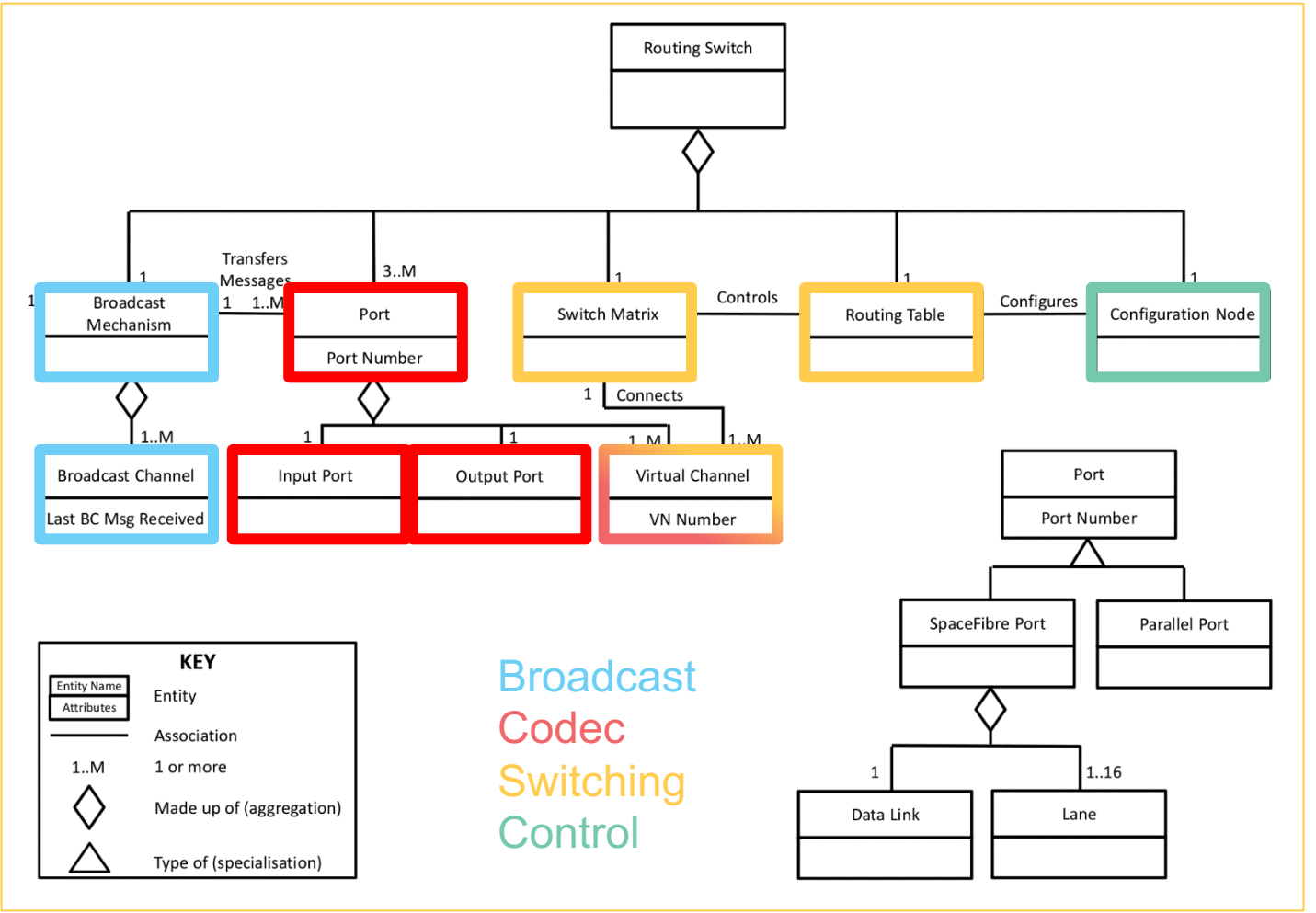
Example system

SpaceFibre Router: Routing Switch Specification

- Standard specifies routing switch components
- General implementation: Switch matrix, routing table, configuration node
  - Often, not all features are used
  - Overhead in implementation
  - Easy to introduce errors

Idea: Use the configurability of an FPGA to bake the switching configuration into the fabric

ECSS-E-ST-50-11C: Fig 5-55: Components of a SpaceFibre Routing Switch



SpaceFibre Router: Switch configuration file

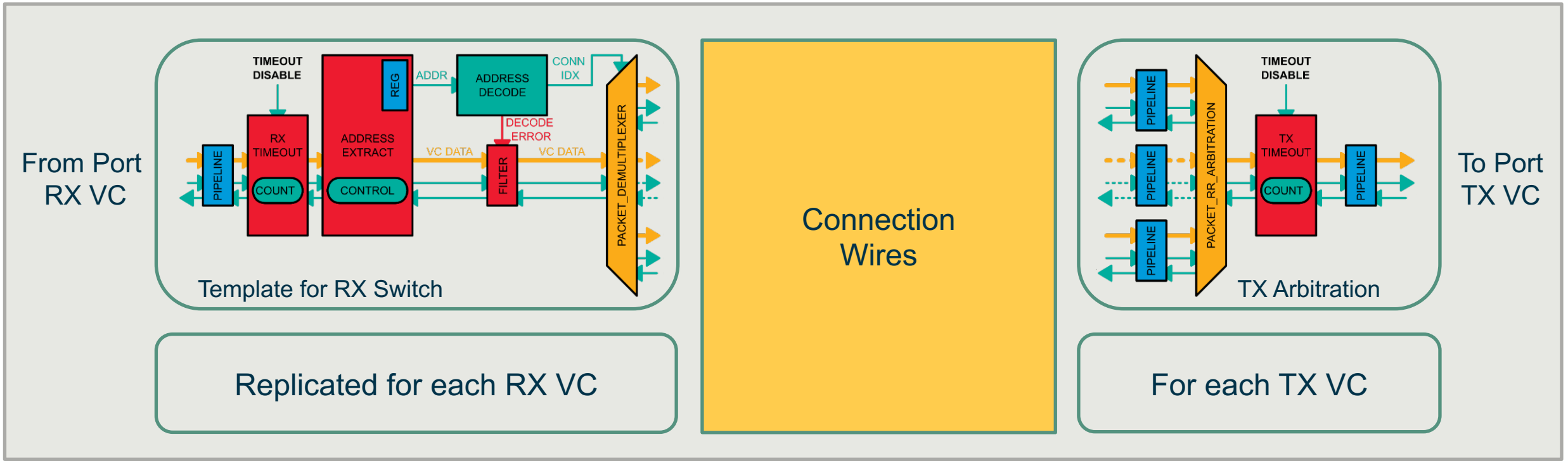
Model routing switch configuration using a description file (YAML)

```yaml
name: sfr_sample_system

ports:
  # The first port in the list is reserved for configuration of the router
  - name: config
    params:
      NumVc: 1
  - name: obc
    params:
      NumVc: 2
  - name: mmu
    params:
      NumVc: 3
  - name: processing
    params:
      NumVc: 2
  - name: instr_1
    params:
      NumVc: 2
  - name: instr_2
    params:
      NumVc: 3
  - name: transmit
    params:
      NumVc: 2

# This is the global address map used for generating the address decoding
# Path addressing is generated by port index if the vc has it enabled
addr_map_logic:
  32  : obc
  33  : obc
  34  : obc
  35  : obc
  36  : obc
  40  : obc
  41  : processing
  42  : processing
```

```yaml
# Virtual Network description
virtual_nets:

  - name: command
    glob_index: 0
    port_vc_map:
      config: 0
      obc: 0
      mmu: 0
      processing: 0
      instr_1: 0
      instr_2: 0
      transmit: 0
    connections:
      - from_port: config
        to_ports: [obc]
        addr_type: ['path']
      - from_port: obc
        to_ports: [config, mmu, processing, instr_1, instr_2, transmit]
        addr_type: ['path']
      - from_port: mmu
        to_ports: [obc]
        addr_type: ['path']
      - from_port: processing
        to_ports: [obc]
        addr_type: ['path']
      - from_port: instr_1
        to_ports: [obc]
        addr_type: ['path']
      - from_port: instr_2
        to_ports: [obc]
        addr_type: ['path']
      - from_port: transmit
        to_ports: [obc]
```

SpaceFibre Router: HDL Code Generation

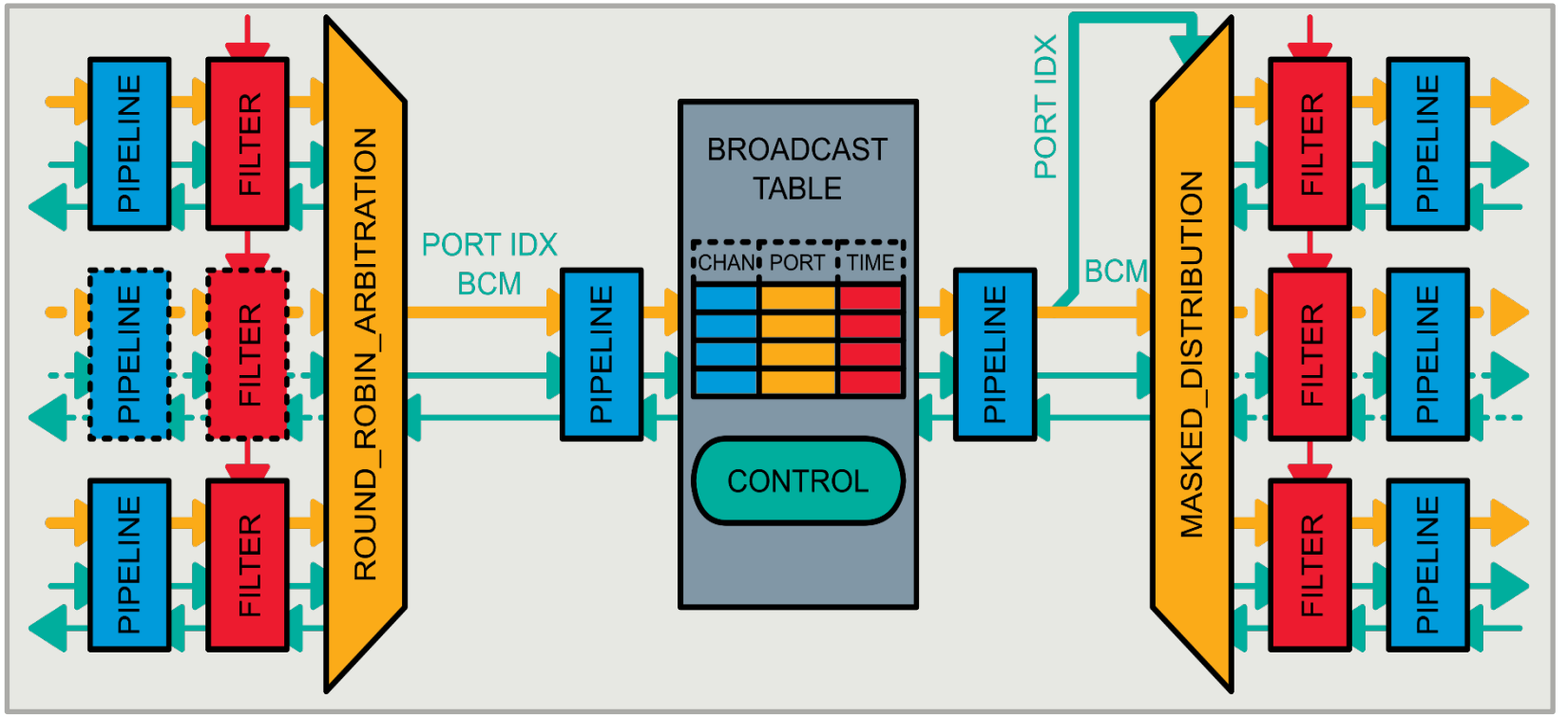
Generate with Python the HDL instantiation and connections between modules



SpaceFibre Router: Broadcast Mechanism

System to quickly get small messages to all nodes in a SpaceFibre network

- Transfer broadcast messages (BCM)
- Prevent broadcast storms → Timeout
- Drop outdated BCM
- Enable/disable individual ports (filter)



SpaceFibre Router: **Validation**

- UVM testbench with random packet stimuli generation
  - Virtual channel switch matrix: **generated, post simulation validated**
  - Broadcast mechanism: **custom, scoreboard validation during simulation**

- Hardware implementation in optical transceiver-testbed
  - Vivado 2020.1 for XCKU040 FPGA (loop-back side)
  - Two configurations implemented, designed with example system in mind
    - 5x5 fully-connected router with 3 virtual networks
    - 5x5 minimal router with 3 virtual networks
  - Implementation frequency **250MHz → 10GB/s** for each virtual channel per port

SpaceFibre Router: **Evaluation**

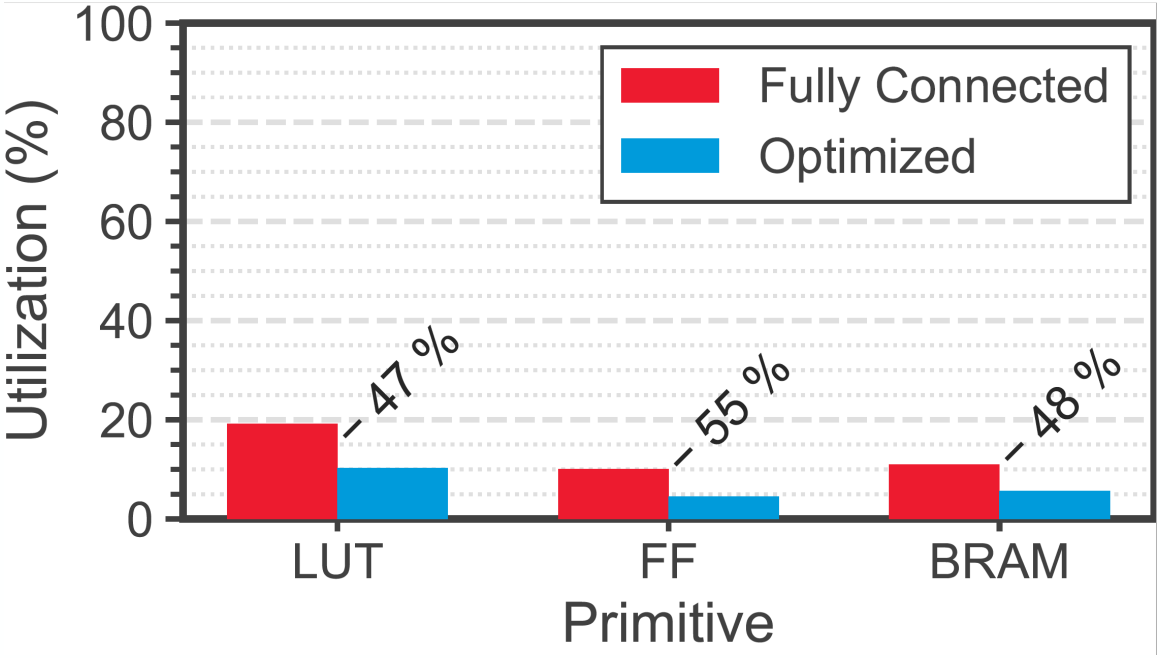
- Utilization shows clear advantage of optimized switch-matrix

(a) Optimized

Block	LUT	FF	BRAM
Switch	3622 (1.5 %)	4749 (1.0 %)	0 (0.0 %)
Broadcast	1187 (0.5 %)	2501 (0.5 %)	0 (0.0 %)
Codecs	19710 (8.1 %)	13932 (2.9 %)	34 (5.7 %)
Total	24802 (10.2 %)	22110 (4.6 %)	34 (5.7 %)

(b) Fully connected

Block	LUT	FF	BRAM
Switch	17740 (7.3 %)	24900 (5.1 %)	0 (0.0 %)
Broadcast	1158 (0.5 %)	2501 (0.5 %)	0 (0.0 %)
Codecs	27272 (11.3 %)	20487 (4.2 %)	66 (11.0 %)
Total	46492 (19.2 %)	48816 (10.1 %)	66 (11.0 %)



- Power estimation [W]

Variant	Block			
	Switch Matrix	Broadcast	Codecs	Total
Fully Connected	0.18	0.02	0.78	0.98
Optimized	0.05	0.02	0.41	0.49

Summary

- Optical transceiver testbed in a nutshell
  - 10 Gbit/s data-rate achieved with up-to 3 virtual channels, 8 links → 80 Gbit/s system data-rate
    - Utilizes ~1% of LUTs, 0.36 % of FFs and 0.37% of BRAM per channel
    - Consumes ~0.35W of power including SerDes and transceiver per link
  - System BER < 1e-15 with 99.9 % confidence level

- Design-time configurable SpaceFibre Router
  - Successfully demonstrated a realistic high-speed data-handling unit in optical transceiver testbed
  - 10 Gbit/s data-rate achieved with 5x5 fully-connected/optimized network
    - Utilizes ~10% of LUTs, ~5% of FFs and BRAMs overall (optimized)
    - Consumes <1 W of power (w/o SerDes) (optimized)
  - Validation and evaluation of design-time tailoring shows significant advantage
    - Utilizes 0.5x the resources and power on average when compared to fully connected switch

Backup Slides

SpaceFibre Router: Summary & Outlook

Done:

- HDL generator for the switching fabric
- Broadcast mechanism
- RTL functional tests
- KU040 implementation

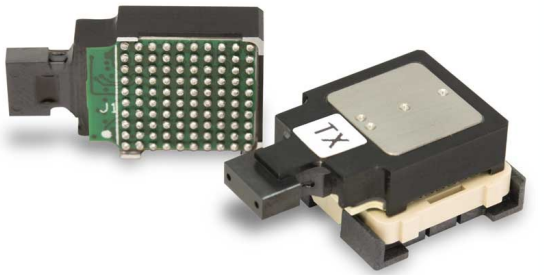
Next:

- Integration of the broadcast mechanism into the KU040 demonstrator
- End-to-End system HDL generation
- Optional feature implementation
  - Dedicated router configuration node for status
  - Group adaptive routing
  - Packet multicast

SpaceFibre Optical Transceiver Testbed - Background

- SpaceFibre
  - Successor protocol to SpaceWire, packet-level compatible
  - Built-in FDIR and QoS on protocol level
  - Variable data-rates up-to 10 Gbit/s depending on cabling used even more with multi-lane implementations

- Smith Interconnect (former Reflex Photonics) LightAble series Transceivers
  - Industrial optical transceiver based on 850 nm multi-mode laser
  - Up-to 10 Gbit/s per channel, available in 4+4 ch. full-duplex and 12 ch. half-duplex configurations
  - Low-power consumption: <100 mW per channel
  - Space-qualified versions (SpaceAble series) available (>100 kRad TID, heavy-ion and proton results available)



SpaceFibreRouter: Broadcast Table