



POLYTECH

Peter the Great
St. Petersburg Polytechnic
University



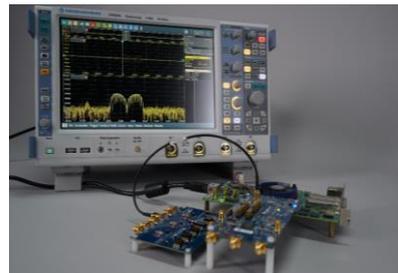
POLYTECH

Advanced Manufacturing
Technologies Center
National Technology Initiative



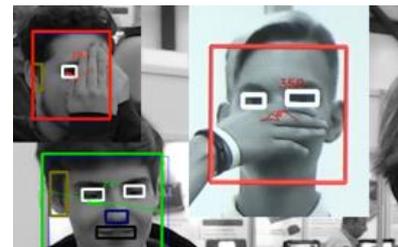
POLYTECH

Laboratory
Industrial Systems for
Streaming Data Processing



SPD collaboration meeting

October 3rd, 2022



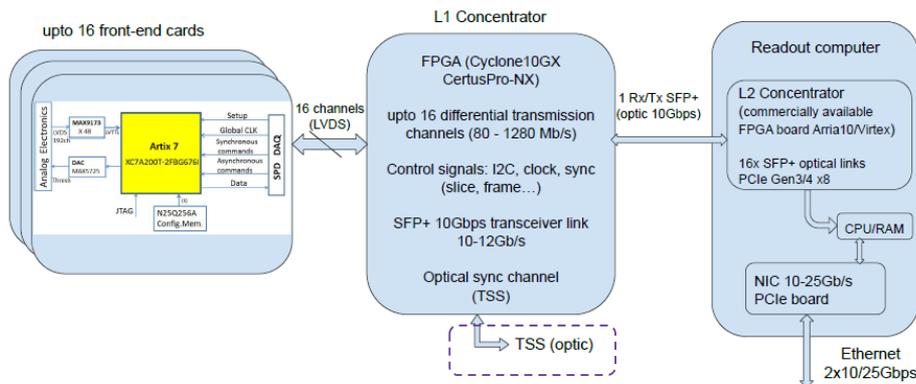
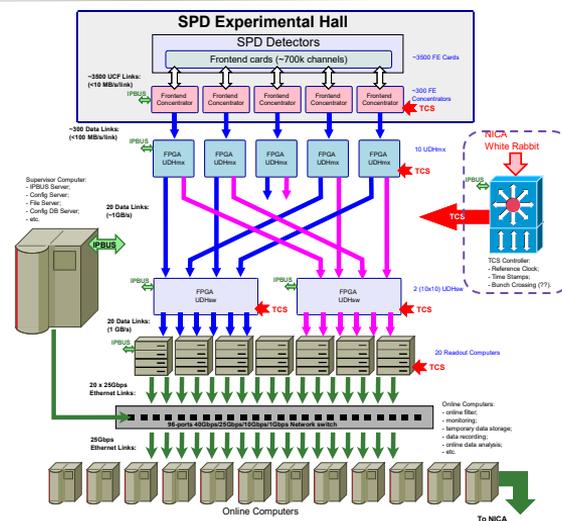
TSS hardware development platform

Antonov Andrei

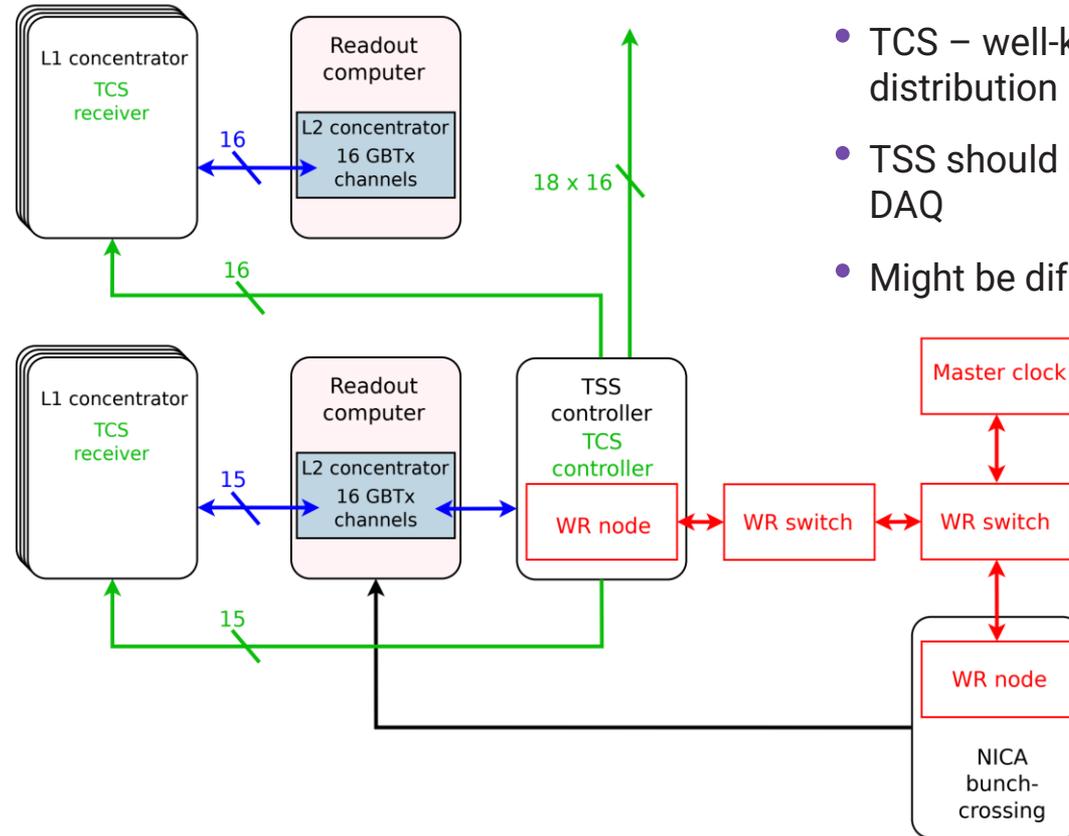
Laboratory "Industrial Systems for Streaming Data Processing", SPbPU NTI Center

TSS instead of TCS

- TCS – Trigger and Control System
- TSS – Time Synchronization System
- Triggerless
- Higher precision
- More complex and versatile
- Synchronizes with NICA
- Propagates bunch crossing signal into DAQ system
- Synchronizes all the DAQ units
- Controls data acquisition process

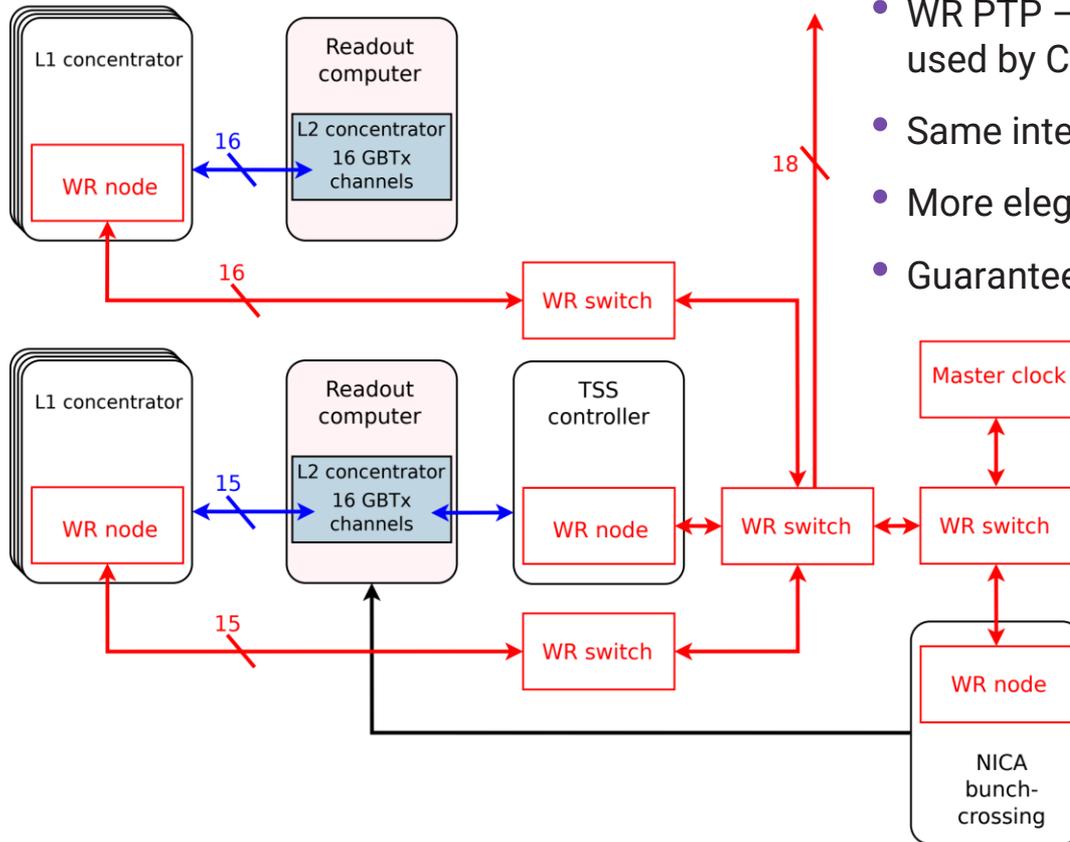


TSS architecture options. TCS-based approach



- TCS – well-known method of clock and control distribution
- TSS should have different interfaces for NICA and for DAQ
- Might be difficult to achieve the required 1 ns accuracy

TSS architecture options. WR-based approach



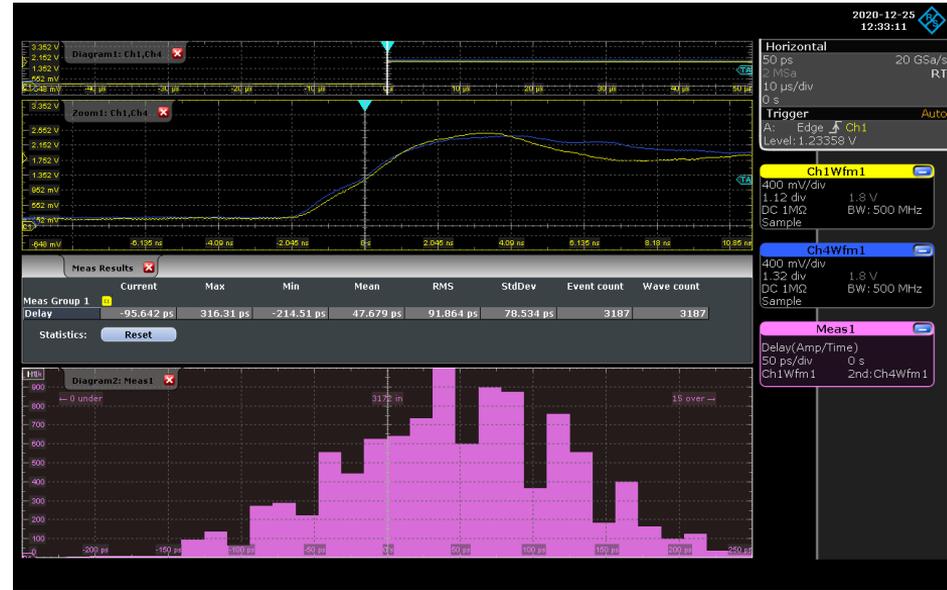
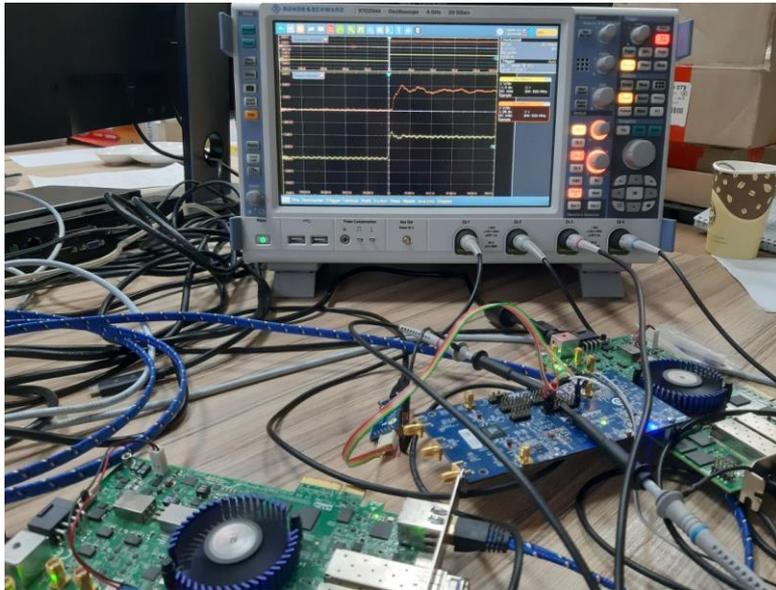
- WR PTP – standard protocol developed and widely used by CERN
- Same interfaces for all connections
- More elegant but expensive and sophisticated solution
- Guaranteed accuracy is much better than 1 ns

White Rabbit in TSS

- No matter what option we will choose – anyway TSS will contain at least one WR node
- NICA uses White Rabbit protocol, so it is the most convenient way for SPD-DAQ to obtain synchronization with NICA
- Decided to start from developing WR-compatible hardware platform
- It should have enough resources and performance

White Rabbit experience

- Implemented WR node on Cyclone10GX DevKit
- Achieved <100ps accuracy

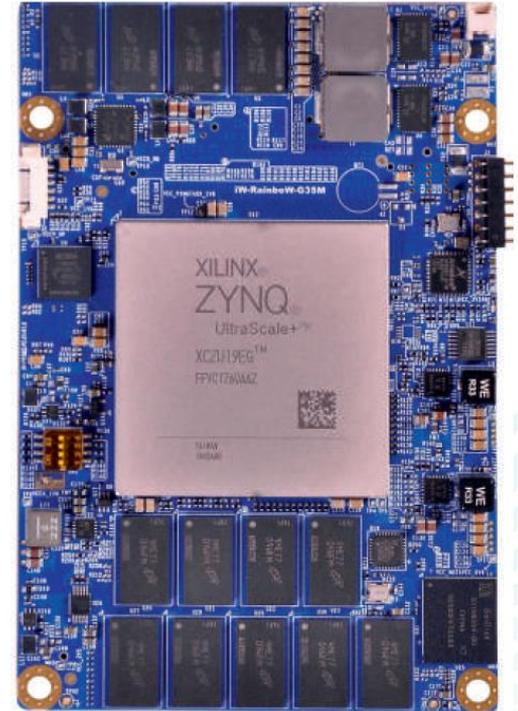


TSS development roadmap

- **HW platform: Schematic and PCB** 2022
- **White Rabbit node project porting to the HW platform**
- TSS design option choosing: TCS-based or WR-based
- NICA interface implementation
- Concentrators interface implementation
- TSS control protocol implementation
- In-system debugging and testing 2024

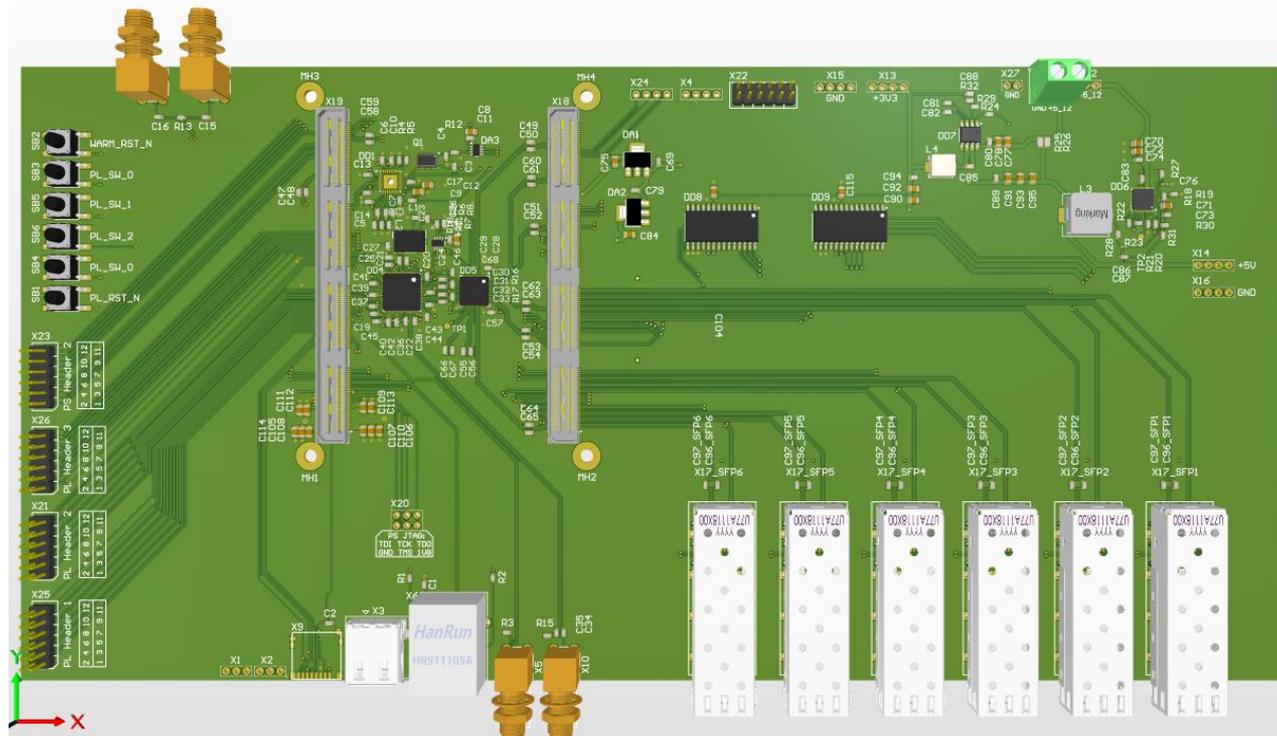
Hardware development platform. FPGA

- Xilinx Zynq UltraScale+
- iWave iW-RainboW-G30M SoM with everything needed on-board
- Up to 504K Logic cells & 230K LUTs
- PL GTH High Speed Transceivers x 16 @ 16.3 Gbps
- DDR4 RAM: 4GB on PS, 2GB on PL
- Flash: 8GB eMMC
- ARM Cortex-A53 hard processor system on-board – some TSS functions could be implemented in software



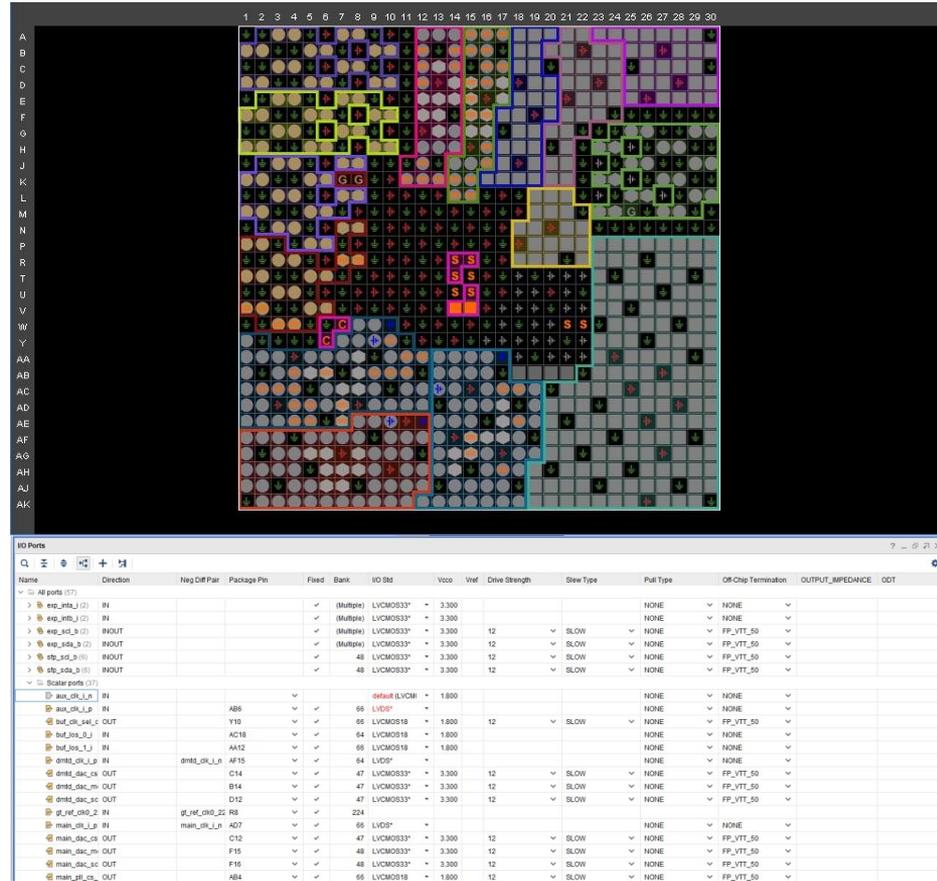
Hardware development platform. PCB

- Custom board based on CERN White Rabbit reference design
- Multi-port baseboard with WR switch functionality – could be a multipurpose device



FPGA project

- WR node FPGA project implementation – done.
- Taken the most relevant WR reference design for Xilinx Zynq UltraScale+ FPGA from CERN's open hardware repository
- Ported it to our hardware platform
- Revised physical constraints
- Successfully compiled the project
- Ready for hardware debug – waiting for the PCB



The screenshot displays the Xilinx Vivado IDE. The top portion shows the FPGA floorplan, a grid of logic resources with various components highlighted in different colors (green, blue, red, yellow). The bottom portion shows the 'IO Pins' table, which lists the configuration for each pin, including direction, package pin, fixed status, bank, I/O standard, voltage, width, drive strength, slew type, pull type, on-chip termination, and output impedance.

Name	Direction	Req Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	On-Chip Termination	OUTPUT_IMPEDANCE	OOT
exp_intn_j (2)	IN			✓	(Multiplex)	LVCMS033*	3.300				NONE	NONE		
exp_intn_j (2)	IN			✓	(Multiplex)	LVCMS033*	3.300				NONE	NONE		
exp_intn_b (2)	INOUT			✓	(Multiplex)	LVCMS033*	3.300	12		SLOW	NONE	NONE	FP_VTT_50	
exp_intn_b (2)	INOUT			✓	(Multiplex)	LVCMS033*	3.300	12		SLOW	NONE	NONE	FP_VTT_50	
stp_intn_b (8)	INOUT			✓	48	LVCMS033*	3.300	12		SLOW	NONE	NONE	FP_VTT_50	
stp_intn_b (8)	INOUT			✓	48	LVCMS033*	3.300	12		SLOW	NONE	NONE	FP_VTT_50	
Scale ports (17)														
dm8_intn_c	IN			✓		dm8n8 LVCM	1.800				NONE	NONE		
dm8_intn_c	IN			✓		LVCM033*					NONE	NONE		
dm8_intn_c	IN		A86	✓	✓	66 LVCM033*	1.800				NONE	NONE		
dm8_intn_c	OUT		Y10	✓	✓	66 LVCM033*	1.800	12		SLOW	NONE	NONE	FP_VTT_50	
dm8_intn_c	IN		AC18	✓	✓	64 LVCM033*	1.800				NONE	NONE		
dm8_intn_c	IN		AA12	✓	✓	66 LVCM033*	1.800				NONE	NONE		
dm8_intn_c	IN	dm8n8	AF15	✓	✓	64 LVCM033*					NONE	NONE		
dm8_intn_c	OUT		C14	✓	✓	47 LVCM033*	3.300	12		SLOW	NONE	NONE	FP_VTT_50	
dm8_intn_c	OUT		B14	✓	✓	47 LVCM033*	3.300	12		SLOW	NONE	NONE	FP_VTT_50	
dm8_intn_c	OUT		D12	✓	✓	47 LVCM033*	3.300	12		SLOW	NONE	NONE	FP_VTT_50	
dm8_intn_c	IN			✓	✓	224					NONE	NONE		
dm8_intn_c	IN	dm8n8	A07	✓	✓	66 LVCM033*					NONE	NONE		
dm8_intn_c	OUT		C12	✓	✓	47 LVCM033*	3.300	12		SLOW	NONE	NONE	FP_VTT_50	
dm8_intn_c	OUT		F15	✓	✓	48 LVCM033*	3.300	12		SLOW	NONE	NONE	FP_VTT_50	
dm8_intn_c	OUT		F16	✓	✓	48 LVCM033*	3.300	12		SLOW	NONE	NONE	FP_VTT_50	
dm8_intn_c	OUT		AB4	✓	✓	66 LVCM033*	1.800	12		SLOW	NONE	NONE	FP_VTT_50	

TSS hardware development. Further steps

- PCB manufacturing (outsource) and assembly (on site)
- Standalone FPGA project debugging
- Testing connection of two implemented WR nodes
- Testing with the original WR equipment
- TSS-dedicated WR node signoff
- Enquiring into the possibility of custom WR-switch implementation on the HW platform

Thank you for your attention

Contacts



Andrey Antonov

Lead Engineer

Tel: +7 (921) 322-47-67

E-mail: andrey.antonov@spbpu.com