Low-Latency High-Bandwidth Circuit and System Design for Trigger Systems in High Energy Physics

September 30th, 2020

Marcos Vinicius Silva Oliveira

Prof. Giovanni de Micheli, jury president

Prof. Yusuf Leblebici, thesis director

Dr. Alain Vachoux, thesis co-director

Prof. Sherenaz Al-Haj Baddar, examiner

Dr. Stefan Haas, examiner

Prof. Andreas Peter Burg, examiner

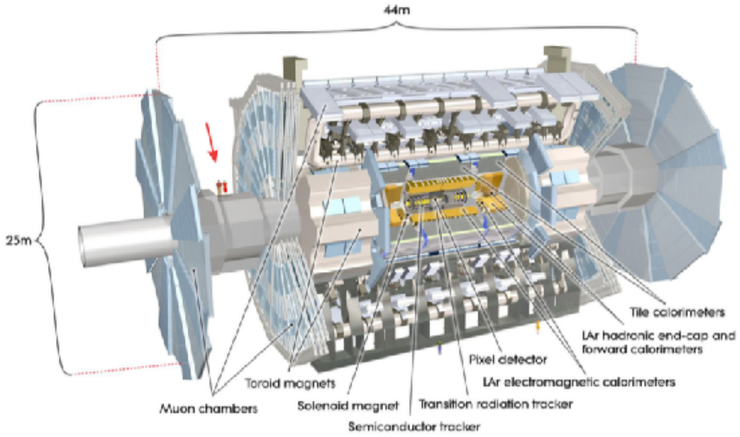
Agenda

- Introduction
- Part I: Data Transfer
- Part II: Data processing
- Summary
- Outlook

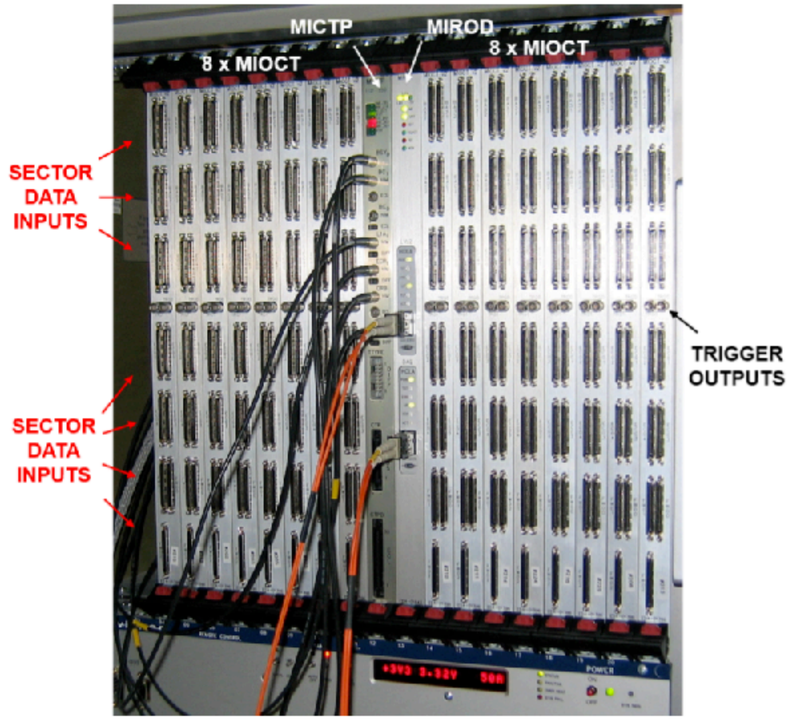
Low-Latency Applications

High-Frequency Trading
ms → $100M/year

High-definition image processing
us → low-latency



Trigger subsystems in ATLAS
ns → otherwise data are lost



- MUCTPI receives and synchronizes muon event data @ 40 MHz
- Processes overlap handling and multiplicity
- Coarse-topological information to L1Topo (M.Sc.)

LHC / MUCTPI Upgrade

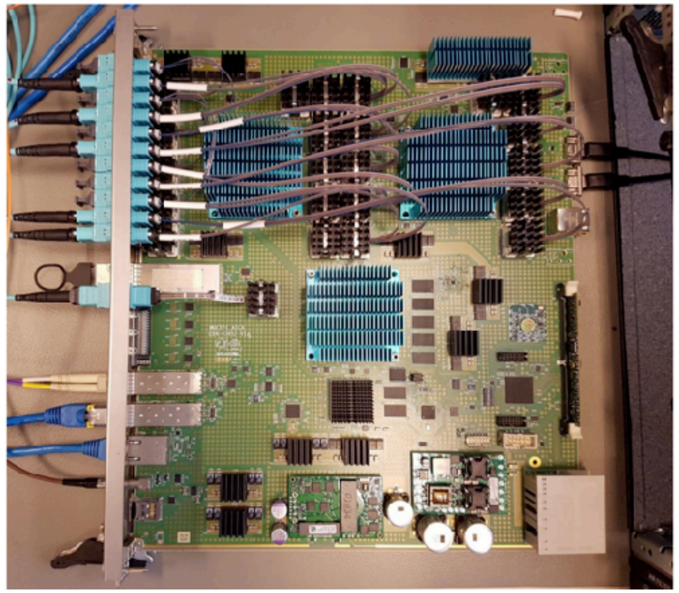
LHC upgrade → higher chances of rare events

Nominal luminosity in 2016
2x in 2021-2024
5-7.5x in 2027-2040

Trigger has to be more selective to keep output rates up to 100 kHz

More selective
↓
routing more detector information to the trigger
↓
Higher bandwidth
-
E.g. up to 4 instead of 2 muon candidates per trigger sector
&
sending full detector-granularity muon position to L1Topo

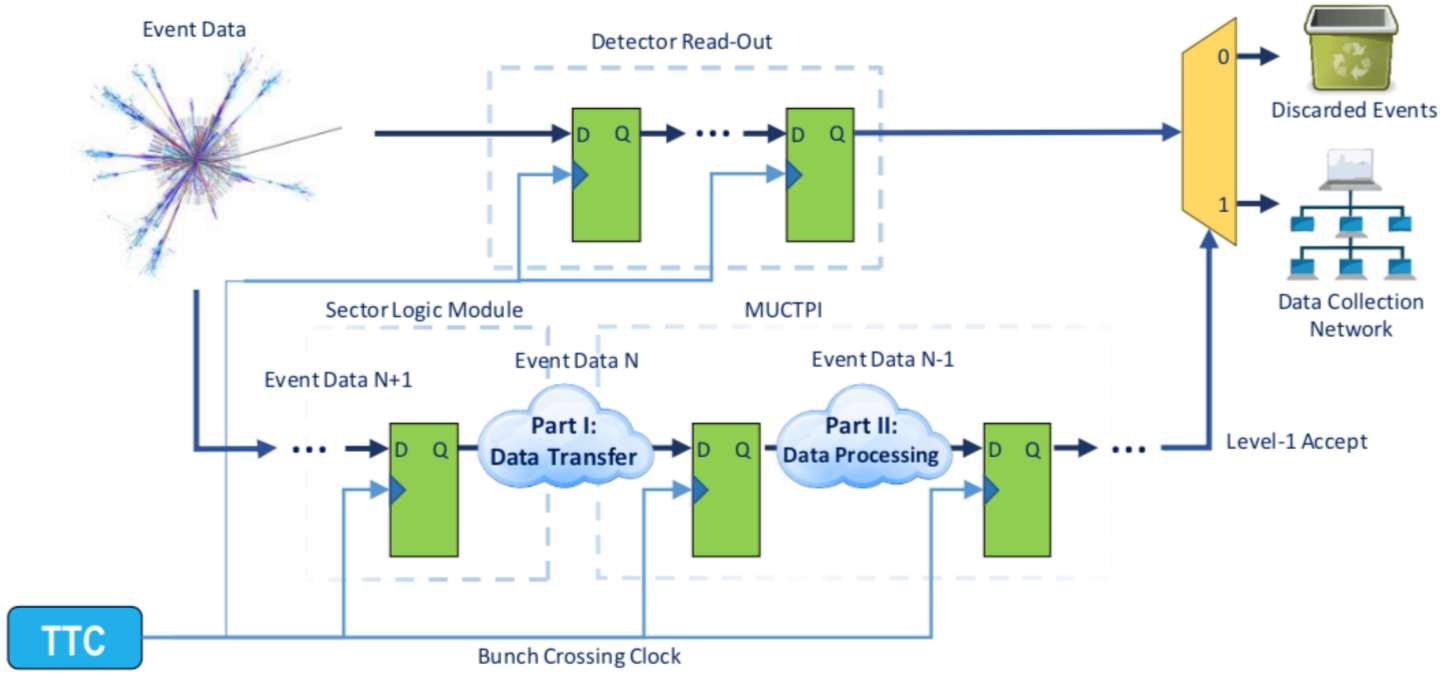
Upgraded MUCTPI



More selective
↓
processing larger portions of information together
↓
Higher integration
-
E.g. All MUCTPI data in one module instead of 18
&
Overlap handling in any region instead of 1/16 of the detector
&
Sorting muon candidates in 1/2 instead of 1/16 of the detector

Thesis Motivation

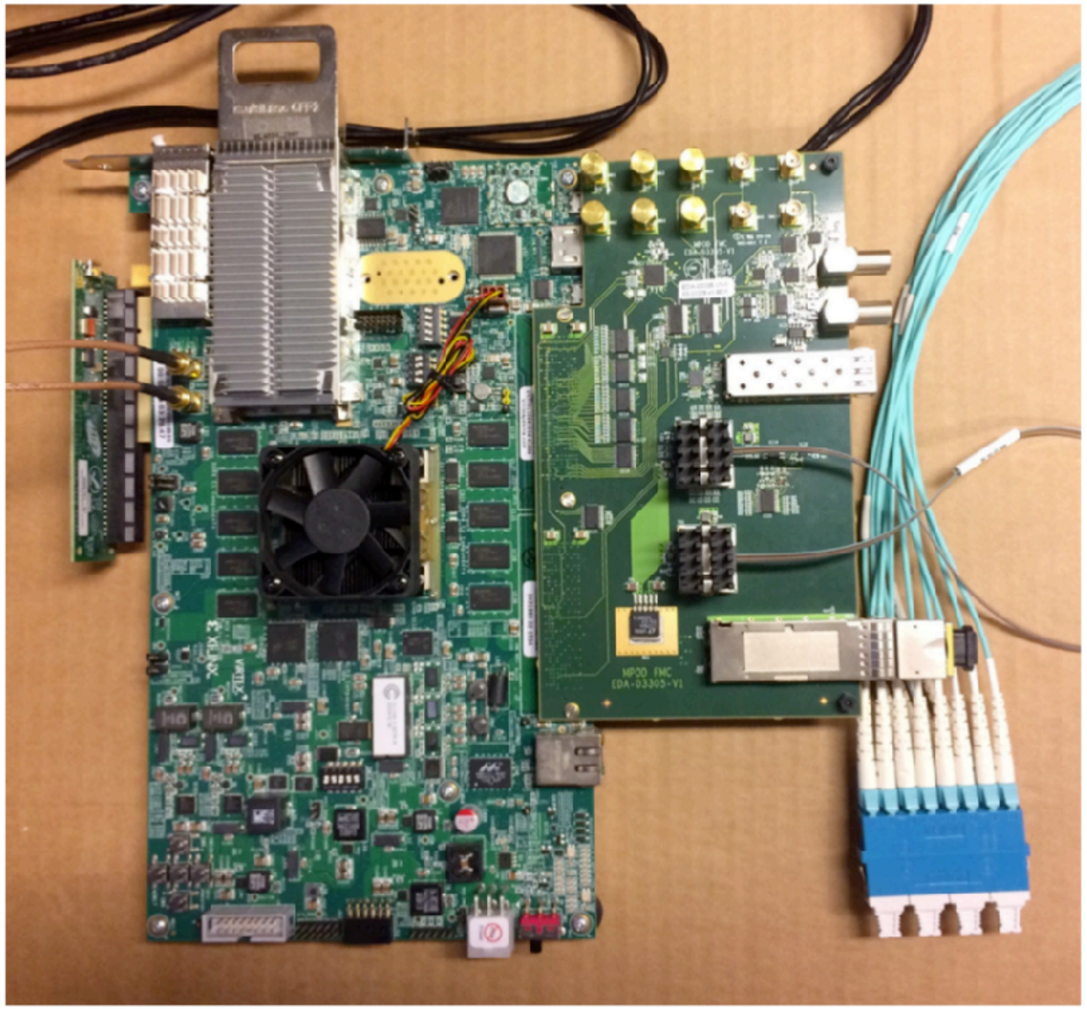
- Low-latency
  - → address limited storage
  - → data are lost
- Fixed-latency
  - → address pipelined processing
  - → wrong event accepted, right rejected
- Reliability
  - → high trigger efficiency
  - → fake triggers



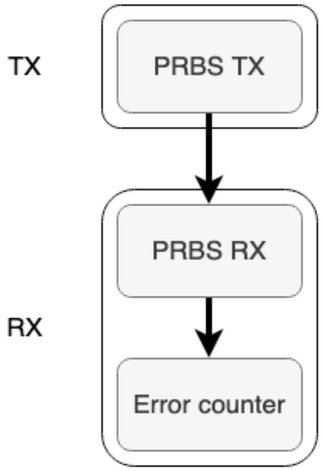
Part I : Data Transfer

MUCTPI Demonstrator

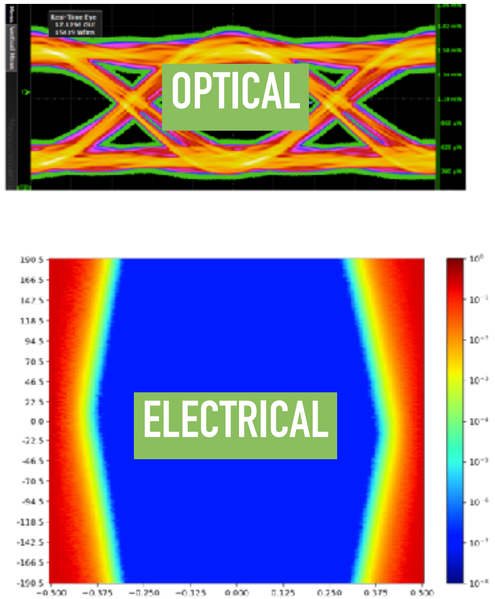
- Evaluation card + custom FMC
  - Demonstrate use of FPGA transceivers and optical modules
  - Latency measurements
  - Demonstrate optical TTC reception
  - Compare 2 jitter cleaner options
  - Enable early firmware development and integration tests



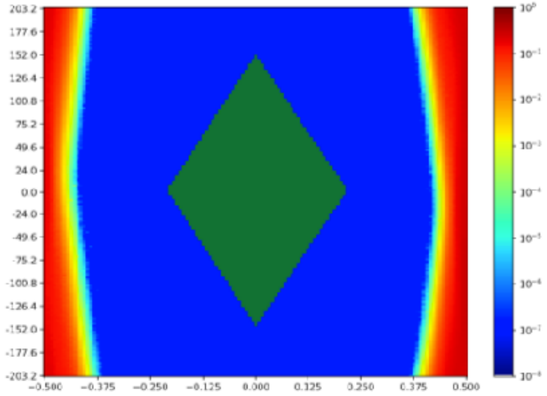
High-Speed Serial Link Testing Techniques



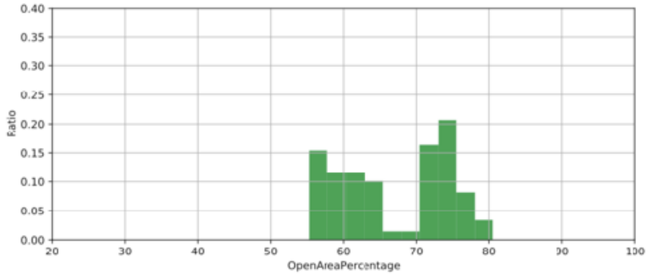
BER testing
↓
estimation of error probability < threshold with quantifiable CL
↓
No qualitative information on how link can be improved



Eye diagrams
↓
qualitative information
↓
No pass/fail result

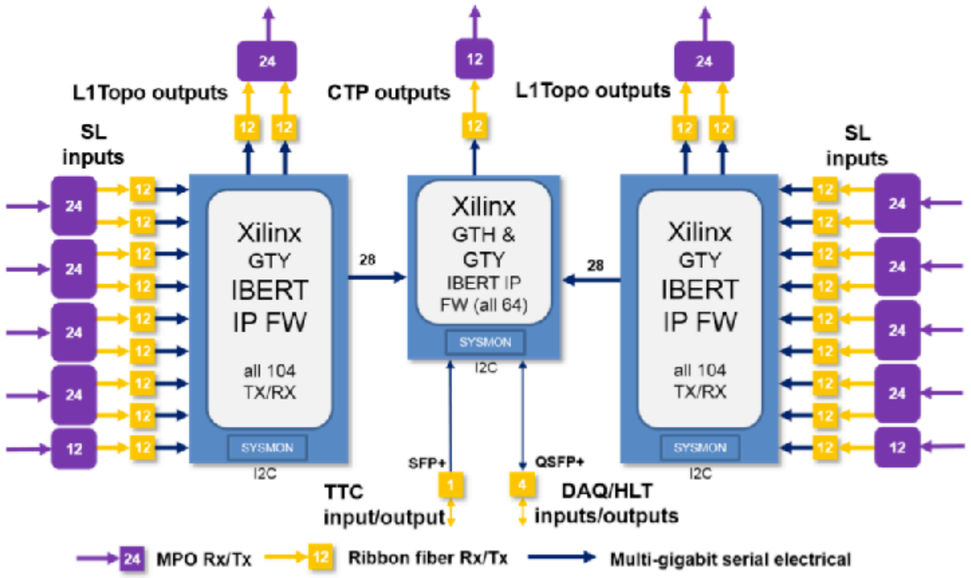


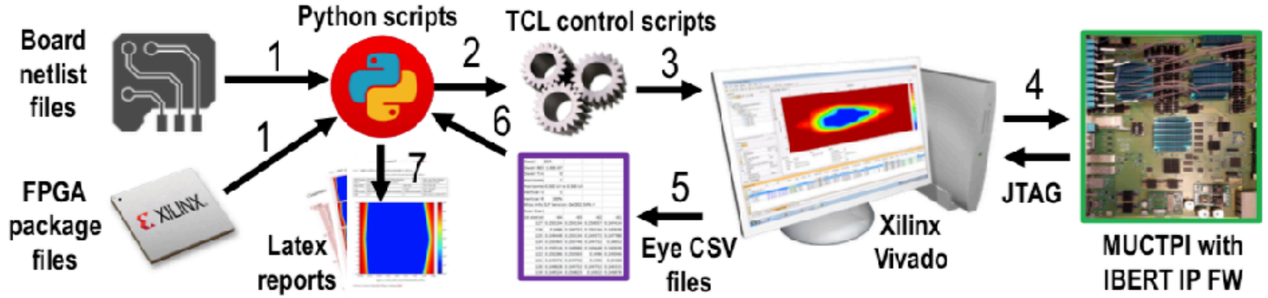
Mask compliance
↓
Introduces pass/fail result to eye-diagrams



Open area histograms
↓
Gives a performance overview, comparative qualitative information

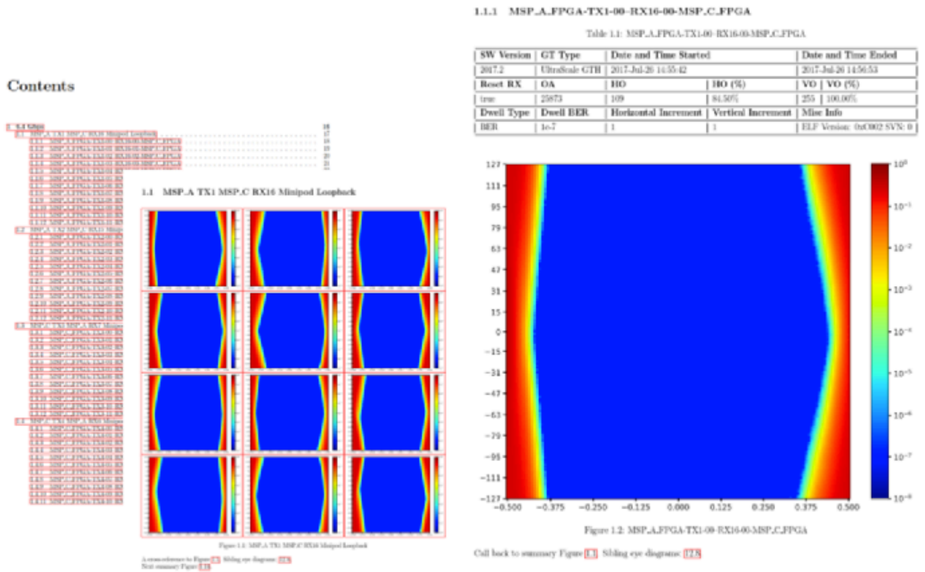
BERT Firmware / Software





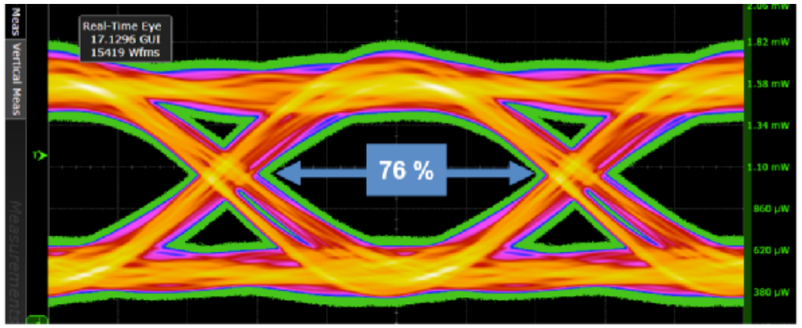
Software: Extracts interconnectivity from schematics, controls the hardware testing, and generate BERT tests, eye-diagrams, mask compliance, histograms, and compiles the result to a PDF report.

Firmware: Based on Xilinx IBERT IP core. Implements pattern generator and checker, error counter. Controls several transceiver settings such as voltage and time offsets, emphasis, differential swing, and equalization.

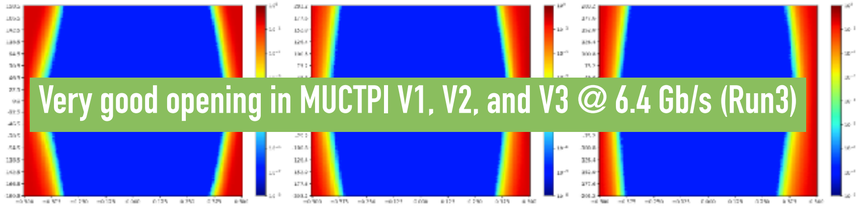


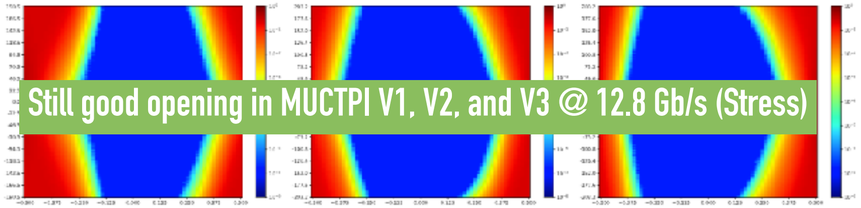
MUCTPI Loopback Test Results

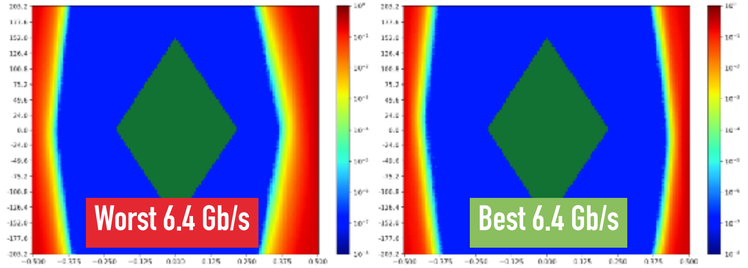
- BER testing: All links tested at 12.8 Gb/s
- $BER < 9 \times 10^{-16}$ with $CL = 95\,\%$
  equivalent to 1 bit error per day
  (Could cause 1 fake or lost trigger per day)

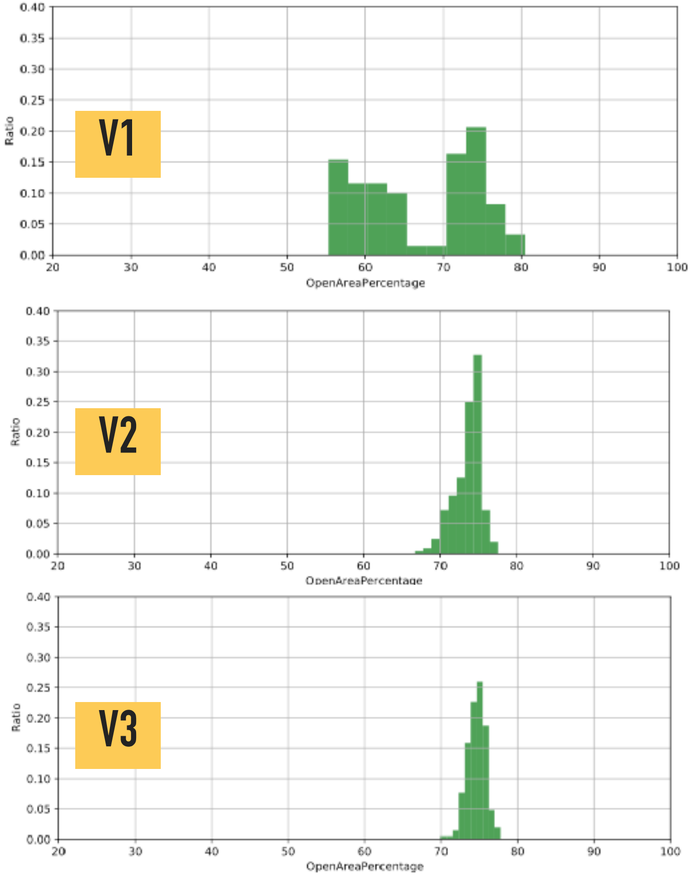


- Very good opening in optical output to L1topo



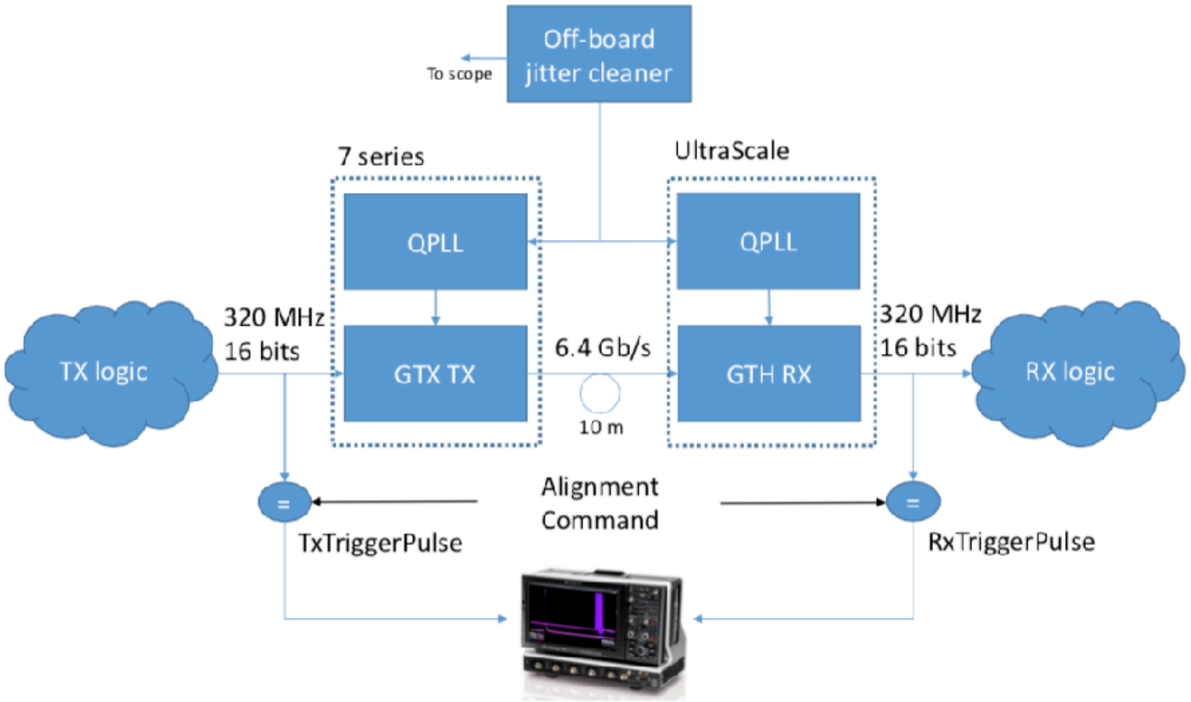


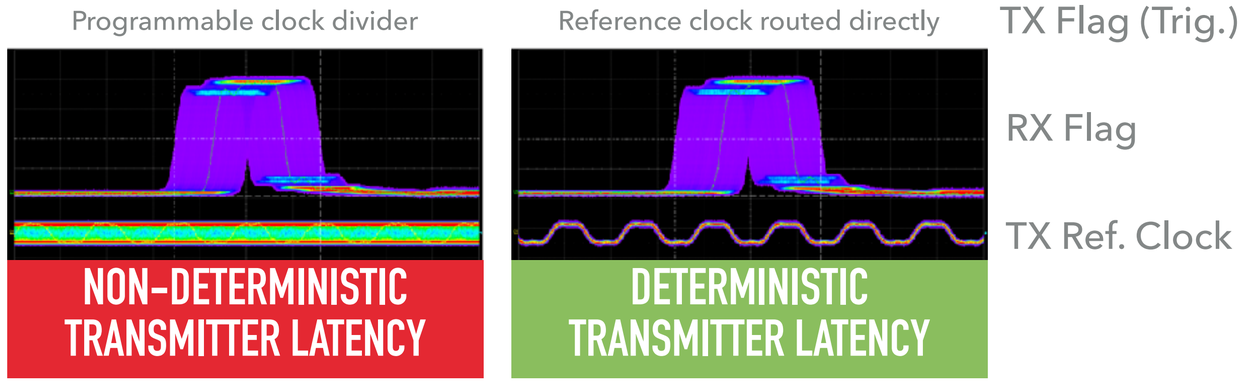




- All links passed mask compliance test
- Different transceiver performance in V1
- Similar performance for V2 and V3

Latency Evaluation Test System and Optimization Results

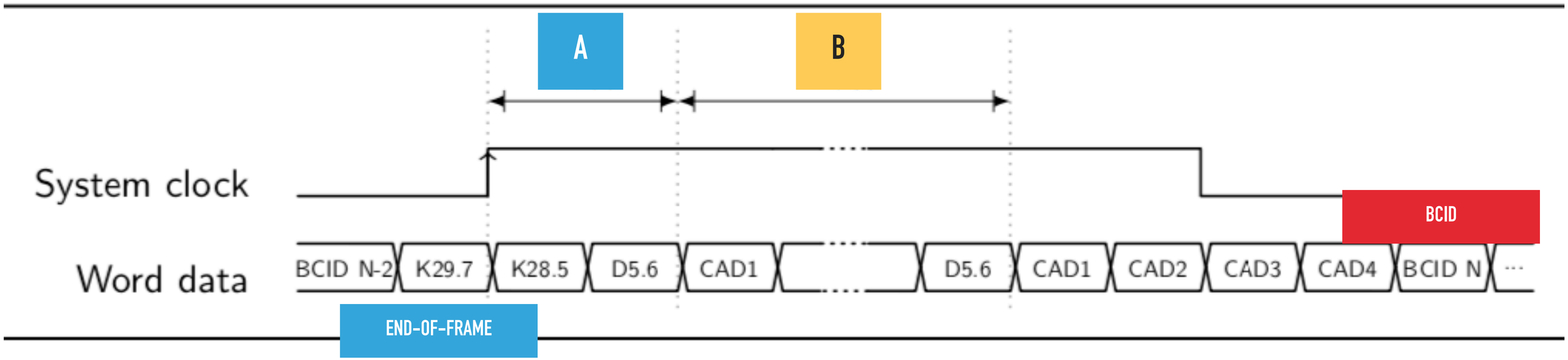




- Based on evaluation kits
- Shared reference clock driven by jitter cleaner
- Periodic sequence is transmitted and received
- Beginning of sequence is detected at TX and RX
  - To measure latency and latency-uncertainty

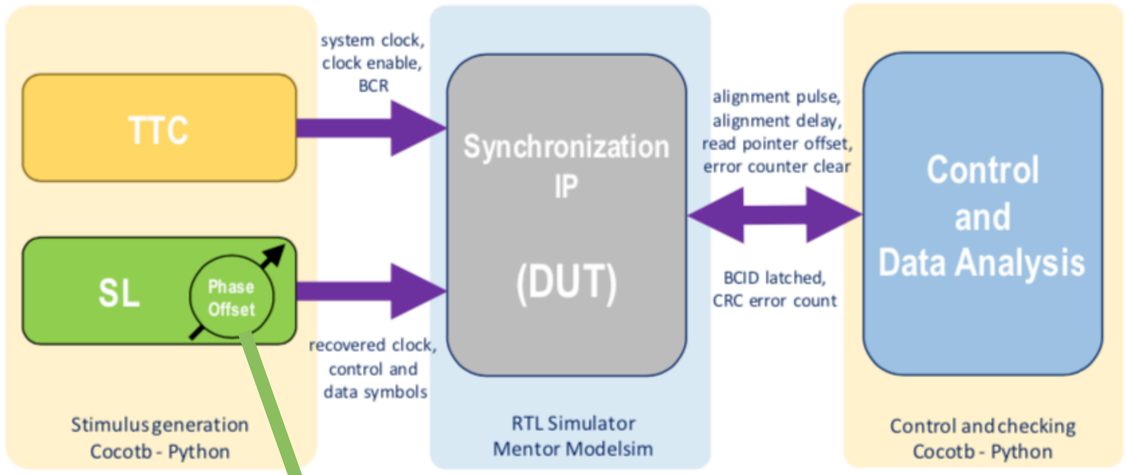
- Clock fabric and data-path MGT configurations optimized:
  - Clock-fabric → clock routed directly when possible
  - Data-path → memory buffers are bypassed
- Overall latency = 50 ns (200 ns budget)
- Overall latency uncertainty = 3.125 ns (absorbed by synchronizer)

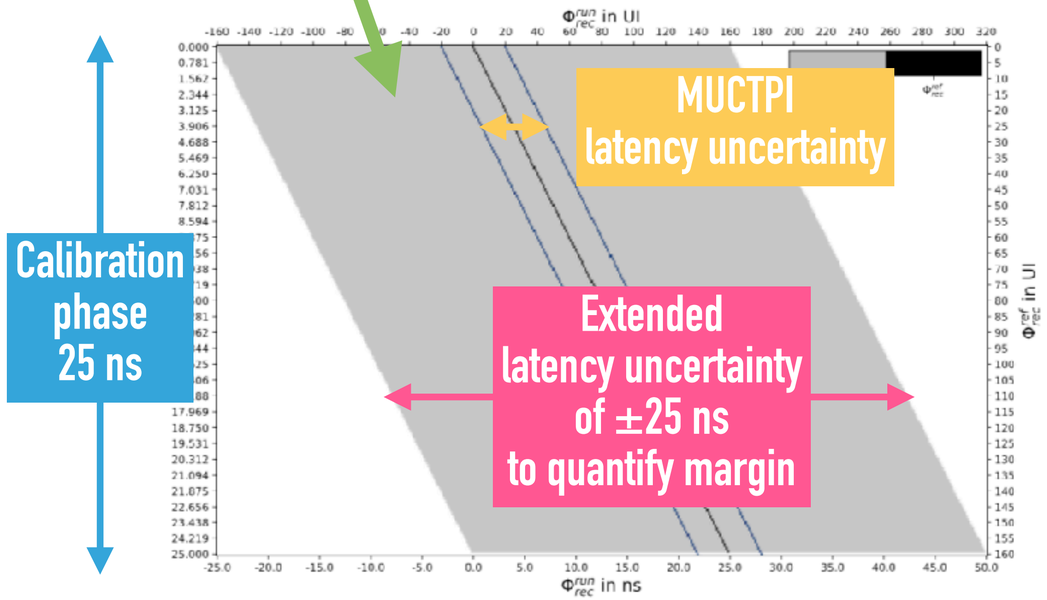
Synchronization Requirements

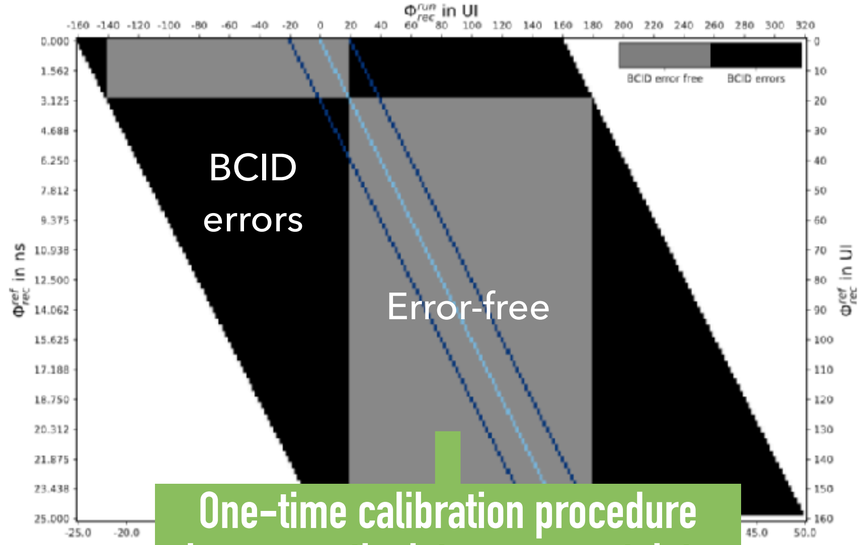


- A: Synchronization phase offset → < 25 ns, with uncertainty of 3.125 ns
- B: Alignment phase offset → k x 25 ns, k ≥ 0
- C: Guarantee deterministic latency
- Synchronization IP based on dual-port memories

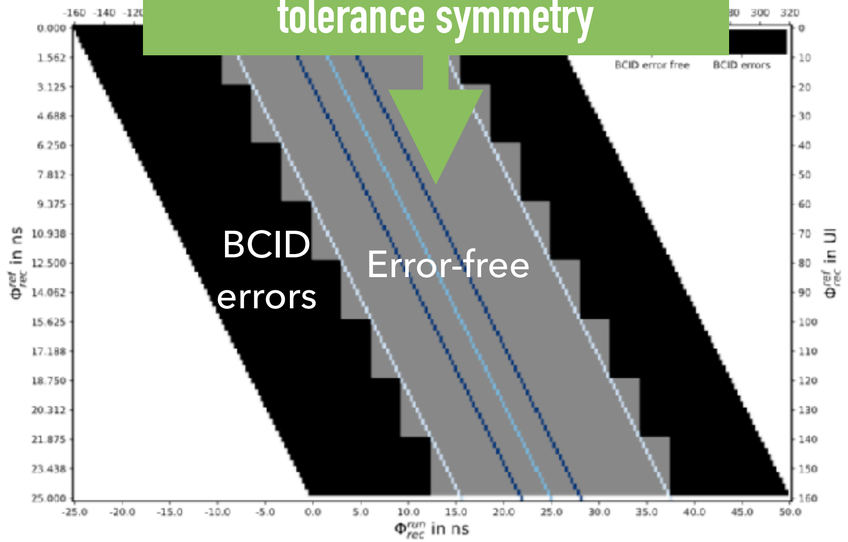
Functional Verification and Calibration



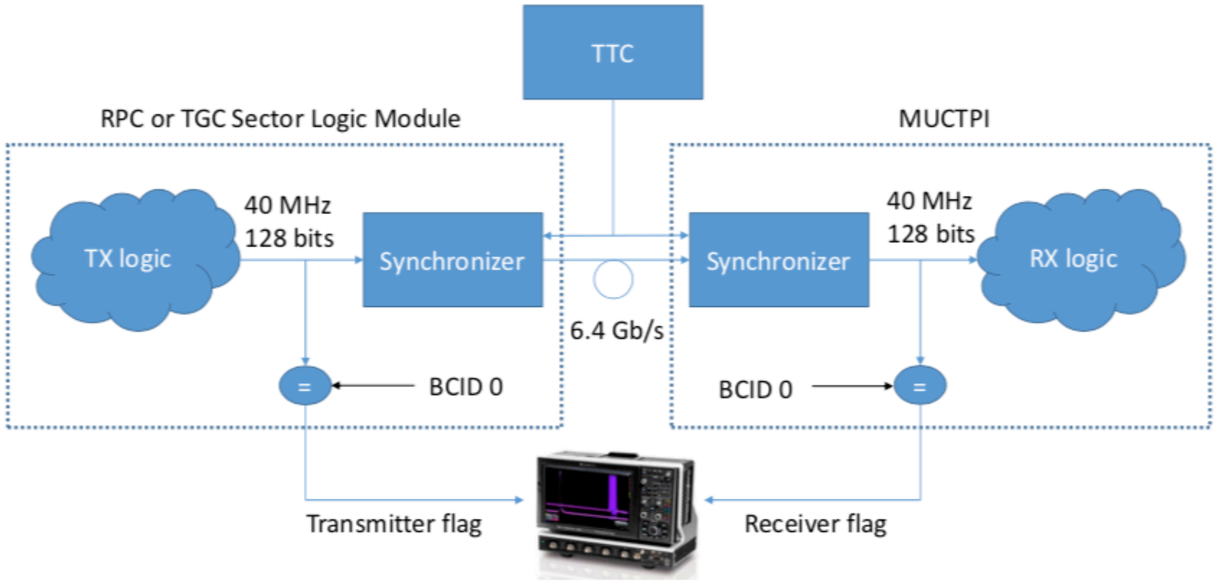


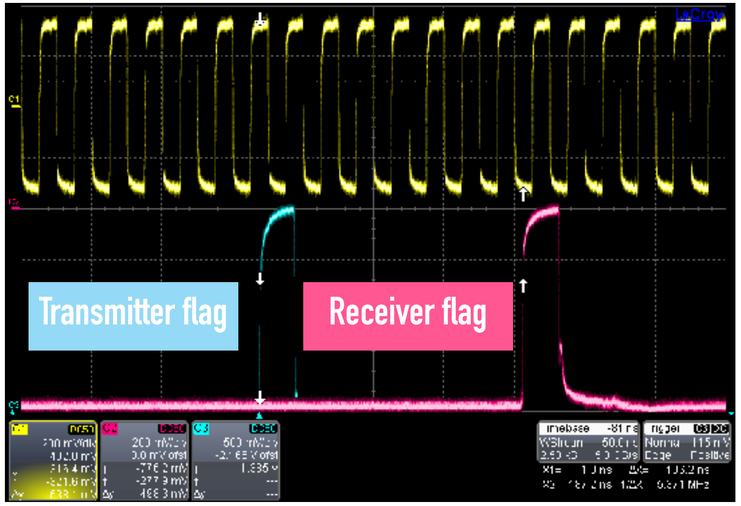


One-time calibration procedure increases the latency uncertainty tolerance symmetry



Integration Tests

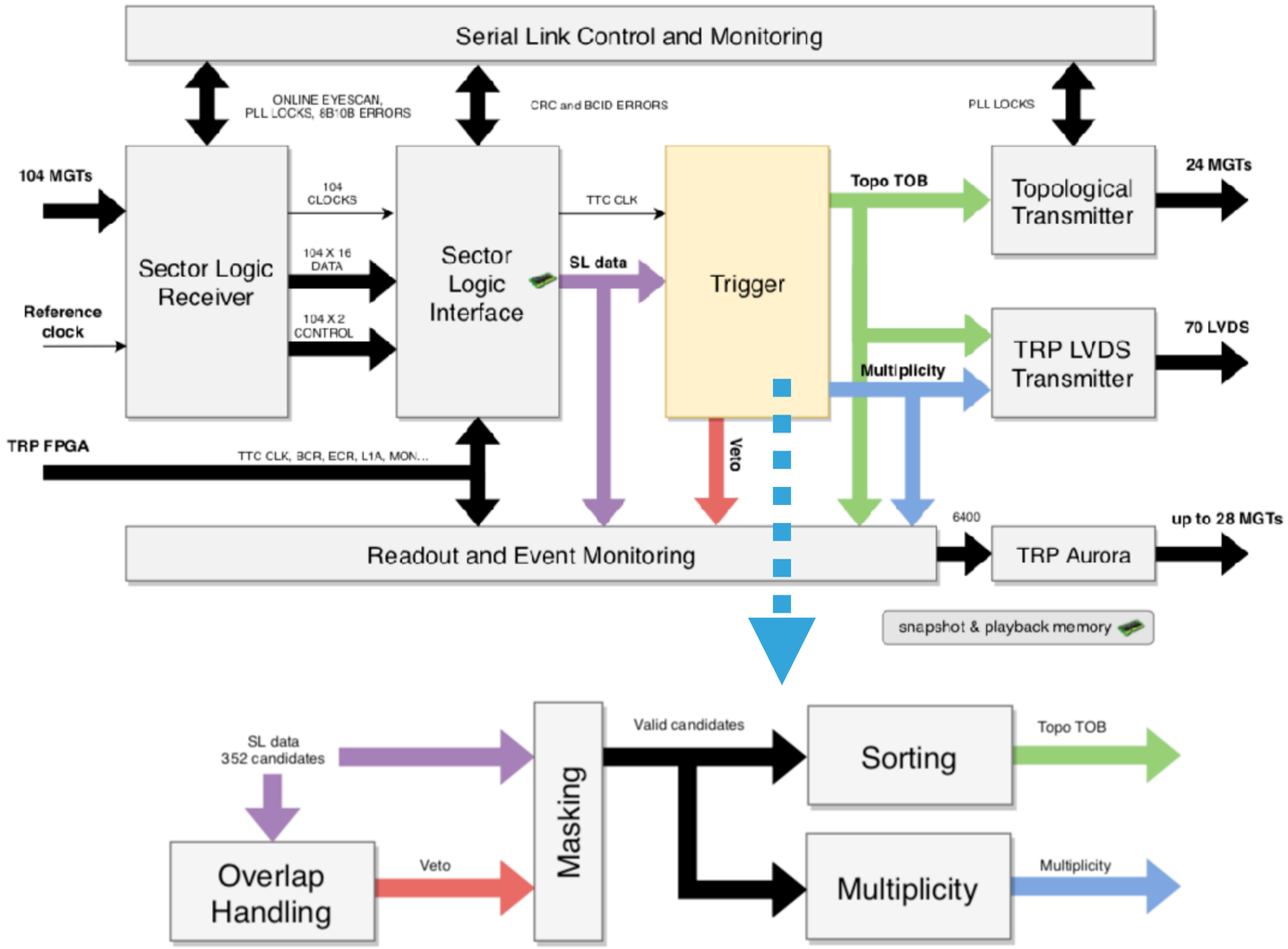


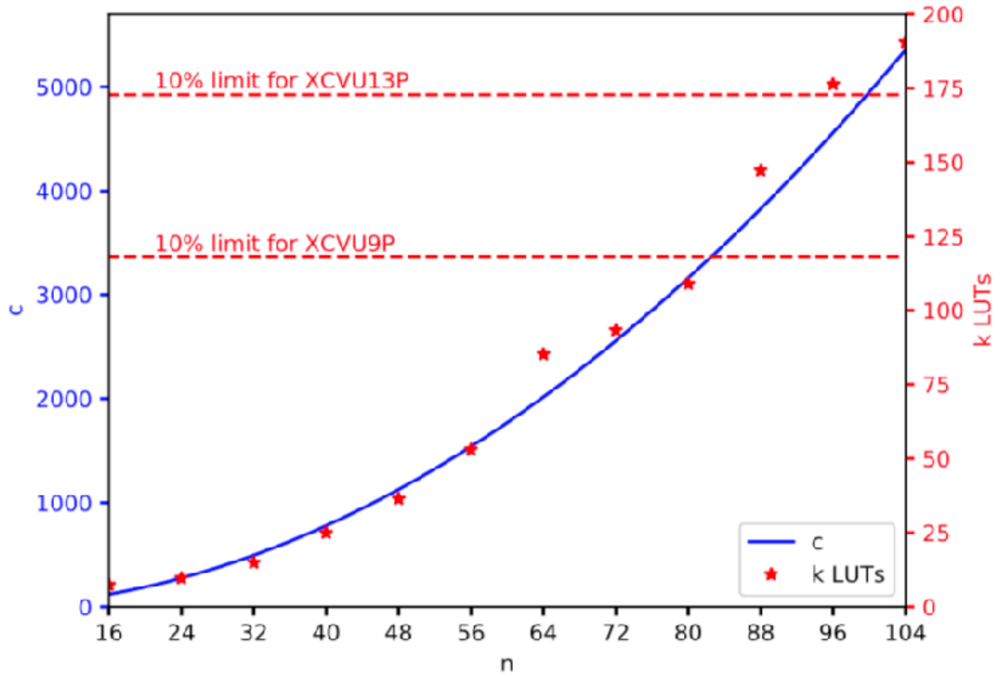


- Data transfer + TX/RX synchronization ≅ 110 ns
- Value is within functional simulation limits and the MUCTPI latency budget of 200 ns
- Fixed latency after MUCTPI synchronization (after reset and power cycle)

Part II : Data Processing

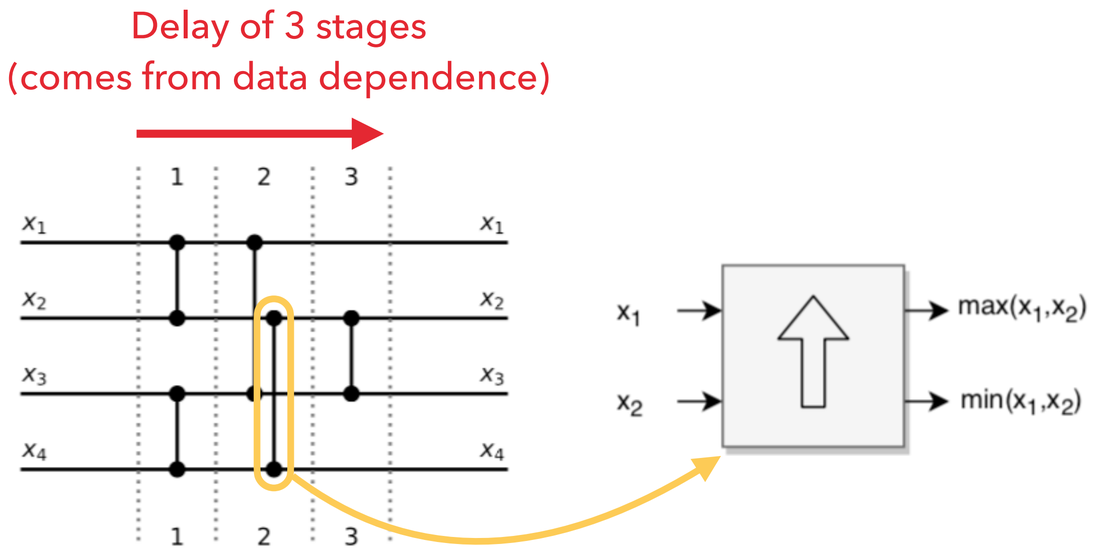
Muon Sector Data Processing

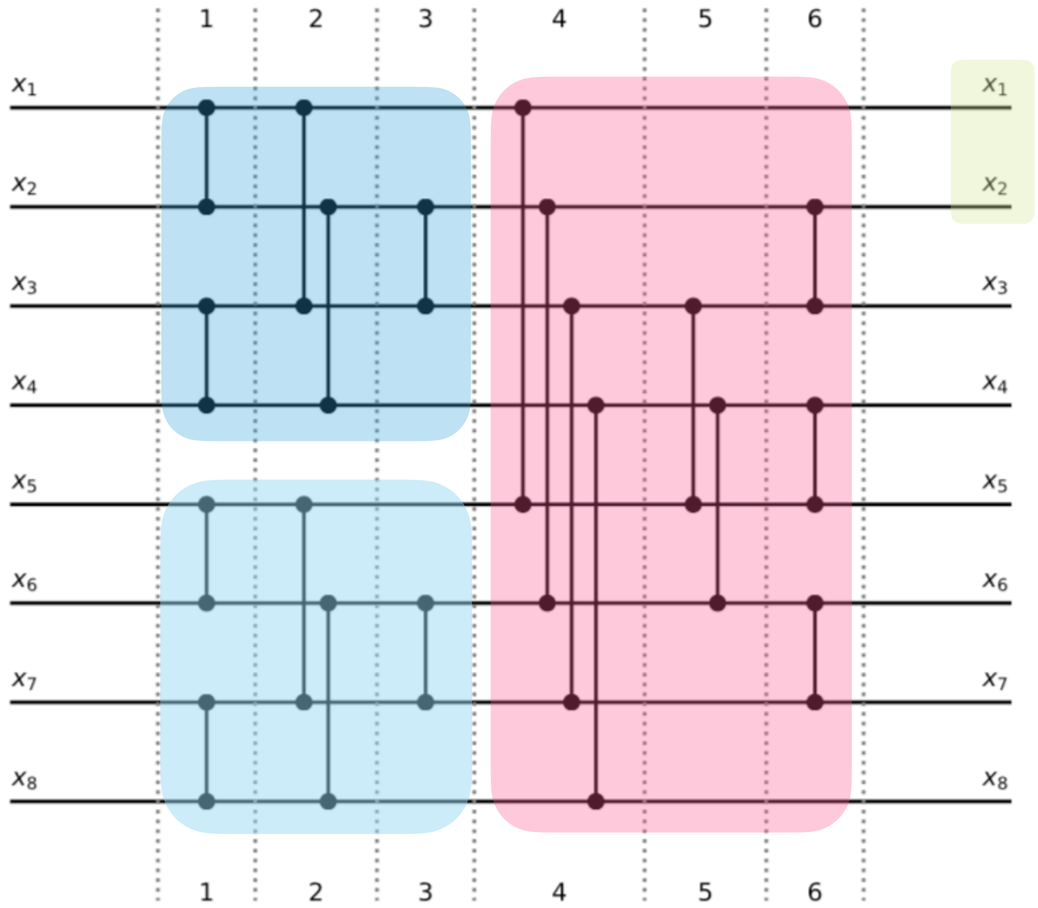




- SL data → Trigger → L1Topo
- Sorting is part of the Trigger
- Run 2 sorting algorithm (26-to-2) cannot be extended to Run 3 requirements (352-to-16)
- Sorting network is known as the fastest method to sort data in hardware

Sorting Networks



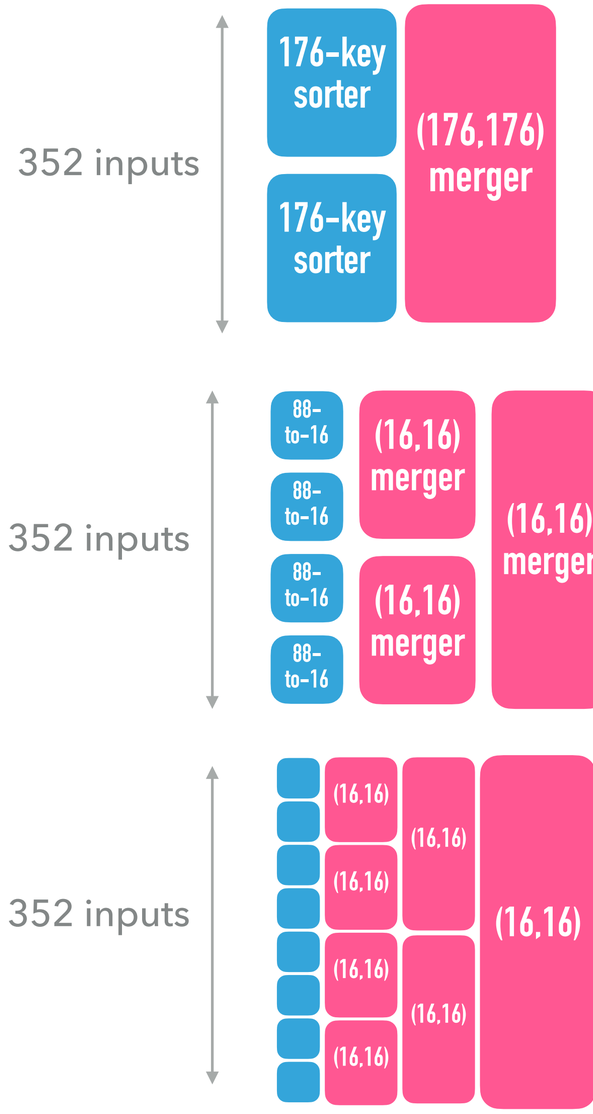


- Sorting networks made of comparison-exchange blocks
- Optimized in view of delay or size

Sorting Networks:
sorts any sequence

Merging Network:
merges two sorted sequences

Divide-and-Conquer Method



With mergesort, before the last merging step, 176 elements have to be sorted !!!!

Given that only the 16 highest outputs are needed, can we avoid sorting and merging hundreds of elements ?

Yes, but there are many options !
Can we select the best options before implementing them in hardware ?

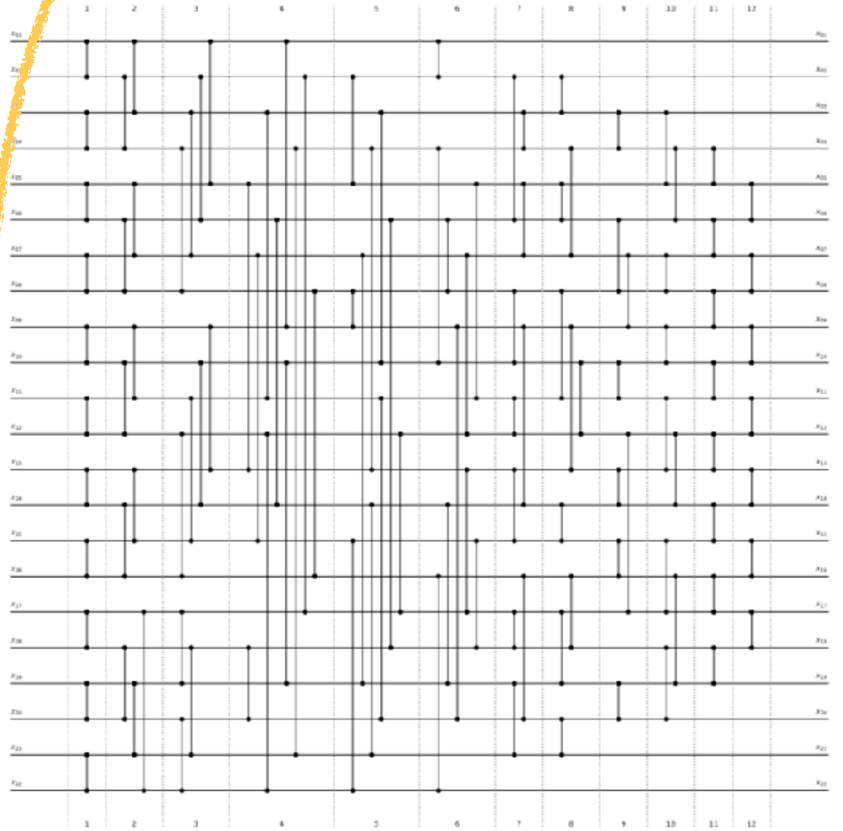
Merge-exchange Sorting Stages

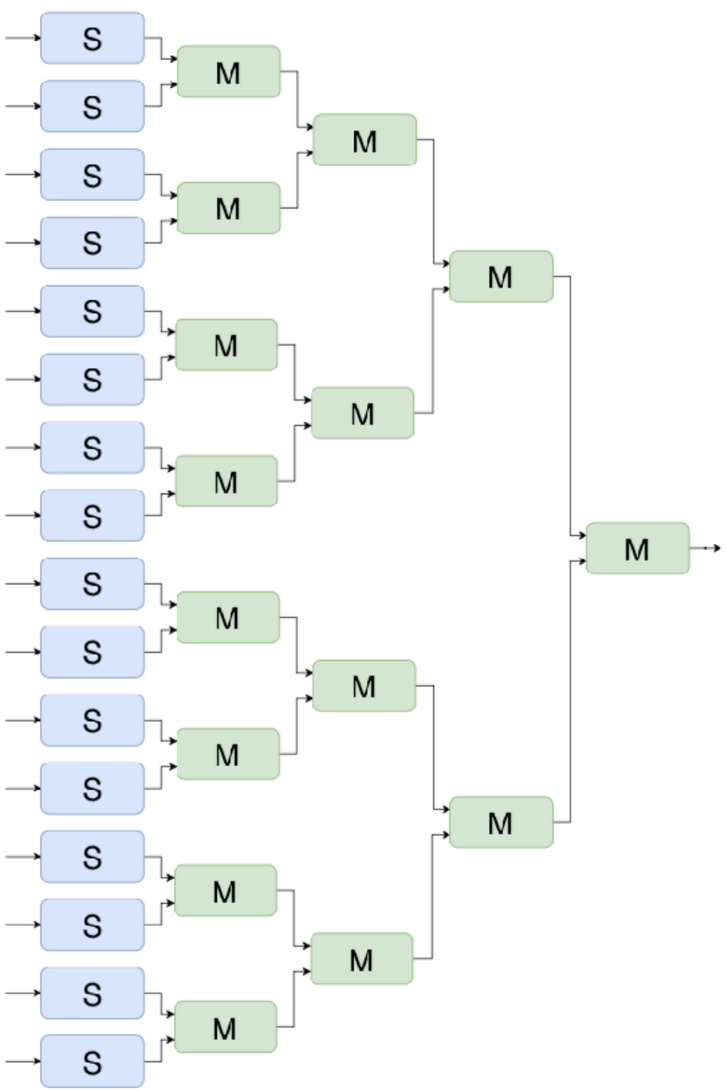
Odd-even Merging Stages

R	Sorting part				Merging part				Total	
	I_s	c_s	a_s	C_s	l_m	i_m	C_m	D_m	C	D
1	352	4446	45	4446	0	0	0	0	4446	45
2	176	1792	36	3584	1	1	48	5	3632	41
3	118	1014	28	3042	2	2	96	10	3138	38
4	88	726	28	2904	2	3	144	10	3048	38
5	71	534	26	2670	3	4	192	15	2862	41
6	59	407	21	2442	3	5	240	15	2682	36
7	51	348	21	2436	3	6	288	15	2724	36
8	44	288	21	2304	3	7	336	15	2640	36
9	40	250	20	2250	4	8	384	20	2634	40
10	36	216	19	2160	4	9	432	20	2592	39
11	32	174	15	1914	4	10	480	20	2394	35
12	30	164	15	1968	4	11	528	20	2496	35
13	28	150	15	1950	4	12	576	20	2526	35
14	26	138	15	1932	4	13	624	20	2556	35
15	24	122	15	1830	4	14	672	20	2502	35
16	22	111	15	1776	4	15	720	**20**	2496	35
17	21	104	15	1768	5	16	768	25	2536	40
18	20	96	14	1728	5	17	816	25	2544	39
19	19	90	14	1710	5	18	864	25	2574	39
20	18	82	13	1640	5	19	912	25	2552	38
21	17	74	12	1554	5	20	960	25	2514	37
22	16	63	**10**	1386	5	21	1008	25	2394	35

Divide-and-Conquer Method - Can We Improve Further?

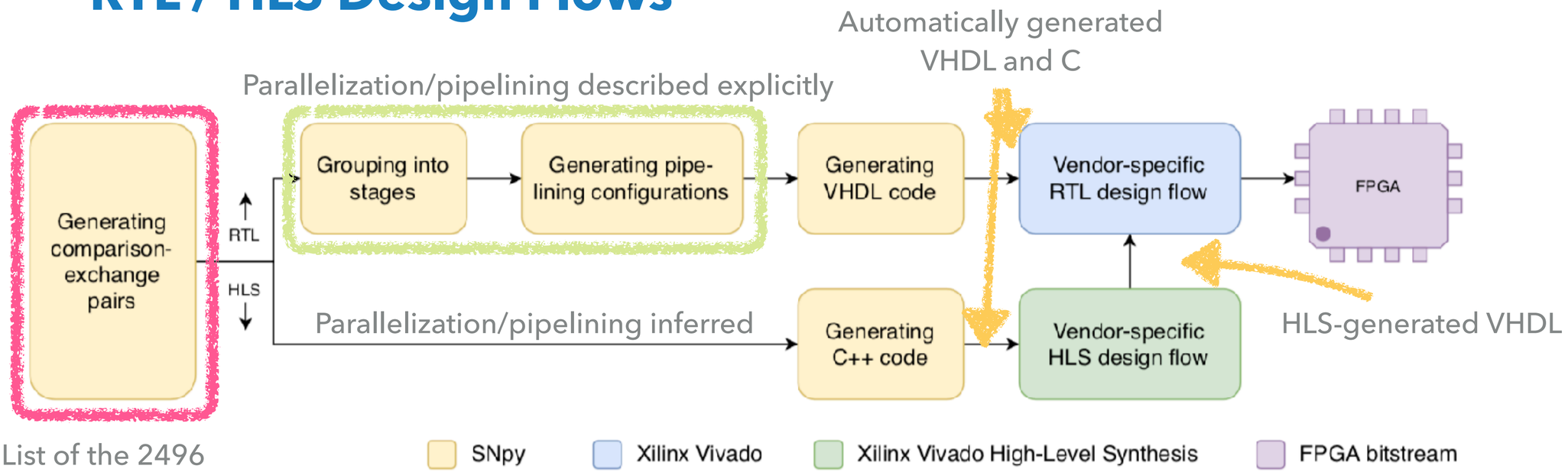
R	Sorting part					Merging part				Total	
	I_s	method	c_s	d_s	C_s	l_m	i_m	C_m	D_m	C	D
16	22	baddar22	113	12	1808	4	15	720	20	2528	**32**
22	1[illegible]	voorhis16	61	9	1342	5	21	1008	25	2350	34



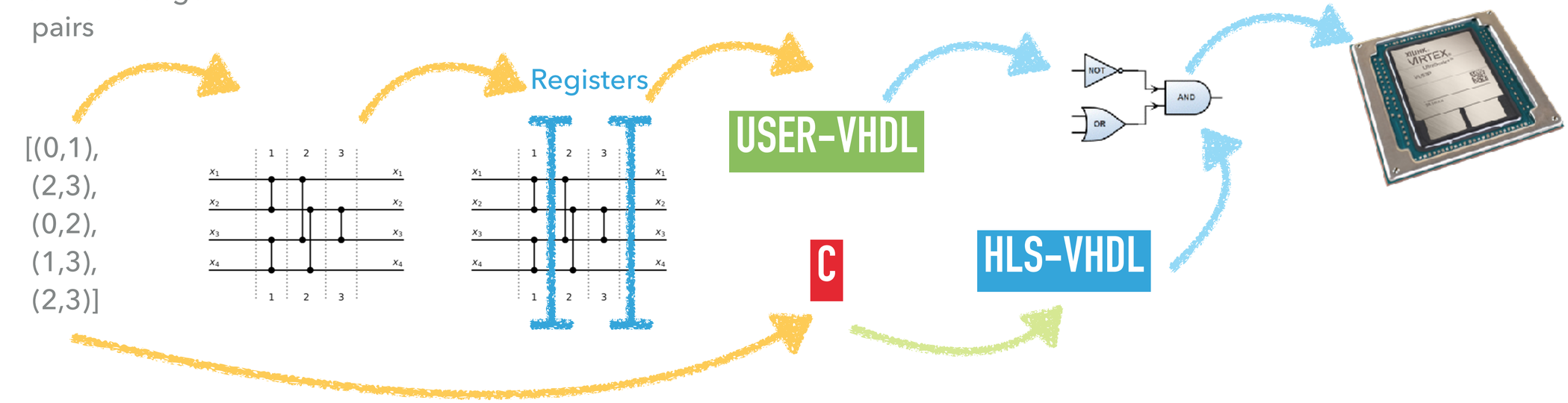


- What about using the fastest known networks ?
- 22-key Baddar sorting network (3 stages faster)
- (16,16) odd-even merging is optimal
- S and M are optimized to 16 outputs
- 32 instead of 45 stages (30 % faster, if compared to 352-key Batcher sorting network)
- 2496 instead of 4446 comparison-exchanges (–44 %)

RTL / HLS Design Flows



List of the 2496 comparison-exchange pairs



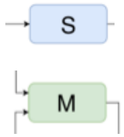
Sorting Network and RTL / HLS Verification

1

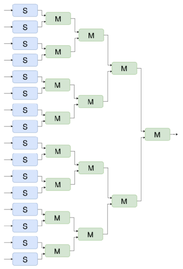
Testing list of comparison-exchanges

[(0,1), (2,3), (0,2), (1,3), (2,3)]

0/1 principle: If a network sorts 0s and 1s, it sorts any arbitrary number!



Verified using the 0/1 principle



2^{30} out of 2^{352} combinations verified using the 0/1 principle

2

C

Testing network description with no implementation details such as parallelism and pipelining

100,000 randomly selected inputs

Available only in HLS (C Simulation)

3

Functional Verification of user-generated and HLS-generated VHDL

USER-VHDL

HLS-VHDL

User VHDL: Mentor Modelsim + Cocotb that enables using a Python testbench

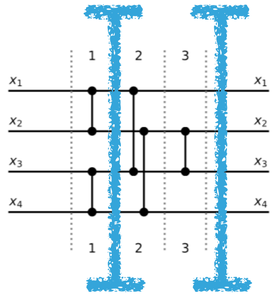
HLS VHDL: Integrated Vivado simulator, same testbench used in C simulation

100,000 randomly selected inputs

RTL Results

L

Number of registered stages (1 to 8)

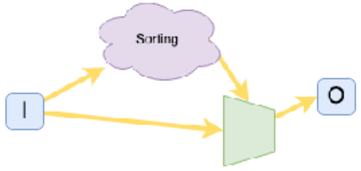


M

Multiplexer

0

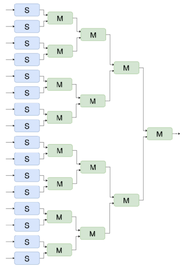
1



H

Abstraction level

3



2

L	M	H	R	WNS	TNS	WHS	Power	LUT	FF	LUTR	Δ_{TS}	Δ_{TI}
5	0	3	0	0.37	0	0.08	6.3	63593	20281	0	00:21:12	00:35:18
			1	0.36	0	0.06	7.25	62280	20277	1	00:22:55	00:35:22
		2	0	0.16	0	0.05	6.32	64401	20311	1	15:48:26	00:36:54
			1	0.04	0	0.05	7.22	61425	20186	49	16:33:34	00:41:42
	1	3	0	0.06	0	0.04	5.02	56818	15591	4224	00:17:53	00:37:28
			1	-0.42	-220.17	0.04	5.61	53696	15889	4225	00:17:50	00:42:44
		2	0	0.01	0	0.04	4.96	57958	15583	4225	23:31:01	00:38:21
			1	0.03	0	0.04	5.68	59333	15896	4237	23:32:48	00:42:50
6	0	3	0	0.9	0	0.05	6.13	56395	24147	0	00:22:31	00:34:22
			1	0.7	0	0.04	6.84	53216	24142	1	00:21:22	00:33:57
		2	0	0.54	0	0.05	6.16	56567	24223	1	13:22:25	00:36:19
			1	0.66	0	0.05	6.93	59281	23950	97	13:07:54	00:35:52
	1	3	0	0.46	0	0.04	4.93	50741	17251	4224	00:18:59	00:32:54
			1	0.02	0	0.05	5.58	54394	17581	4225	00:18:22	00:37:28
		2	0	0.67	0	0.04	4.87	50597	17316	4225	15:46:06	00:33:54
			1	0.45	0	0.04	5.59	54490	17550	4250	16:23:50	00:38:51
7	0	3	0	1.14	0	0.05	6.17	56342	28644	0	00:22:35	00:33:51
			1	0.79	0	0.05	6.97	56359	28638	1	00:22:42	00:33:16
		2	0	1.08	0	0.04	6.07	56336	28704	1	13:12:33	00:36:24
			1	0.55	0	0.05	6.76	65765	28192	191	13:28:43	00:40:00
	1	3	0	0.63	0	0.04	4.86	48350	18964	4224	00:18:55	00:32:09
			1	0.61	0	0.04	5.35	49684	19262	4225	00:18:50	00:40:22
		2	0	0.66	0	0.04	4.87	48303	19022	4225	13:26:41	00:38:50
			1	0.44	0	0.04	5.35	52831	19202	4274	13:43:12	00:37:51
8	0	3	0	1.61	0	0.04	6.12	56335	31984	1	00:23:59	00:33:45
			1	1.29	0	0.05	6.63	57272	31979	1	00:23:37	00:32:31
		2	0	1.36	0	0.05	6.03	56336	32103	1	11:26:42	00:39:59
			1	1.28	0	0.04	6.66	64134	31590	191	10:14:37	00:37:38
	1	3	0	0.89	0	0.04	4.82	48283	20999	4224	00:20:08	00:32:33
			1	0.75	0	0.04	5.36	50649	21262	4225	00:20:06	00:32:09
		2	0	0.8	0	0.04	4.83	48309	21021	4225	13:29:24	00:39:47
			1	0.76	0	0.04	5.28	53886	21187	4274	12:22:54	00:39:40

R Rebuilt

0: Keeps hierarchy in synthesis
1: Flattens hierarchy

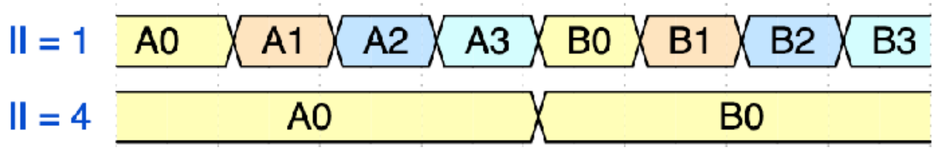
- Clock frequency: 160 MHz (6.25 ns)
- H=2 → increases in up to 100X in synthesis time and have negative timing impact
- M=0; H=3 → highest WNS with L=5 (370 ps)

HLS Results

Options				HLS				HLS-driven RTL								
L	M	II	R	II'	WNS	LUT	FF	WNS	TNS	WHS	Power	LUT	FF	LUTR	Δ_{TS}	Δ_{TI}
5	0	1	0	1	-0.85	134521	17568	0.40	0.00	0.04	6.37	54291	23144	0	00:17:26	00:24:50
			1	1	-0.85	134521	17568	0.54	0.00	0.04	6.33	54216	23144	0	00:17:22	00:24:32
		4	0	4	-0.85	134568	10073	0.13	0.00	0.04	4.04	56348	15675	0	00:19:28	00:26:44
			1	4	-0.85	134568	10073	0.24	0.00	0.04	3.96	54017	15675	0	00:16:25	00:26:07
	1	1	0	1	-0.84	160985	57012	-2.26	-651.56	0.04	5.66	51354	17504	4224	00:17:02	02:20:03
			1	1	-0.84	160985	57012	-2.39	-675.54	0.04	5.67	52423	17498	4224	00:20:55	07:42:51
		4	0	4	-0.84	138504	11692	-0.63	-107.41	0.05	4.45	48141	17236	0	00:16:22	00:49:01
			1	4	-0.84	138504	11692	-1.00	-182.66	0.04	4.45	48648	17235	0	00:16:33	01:13:16
6	0	1	0	1	0.83	134521	18392	0.50	0.00	0.04	6.27	52301	23968	0	00:19:23	00:24:22
			1	1	0.83	134521	18392	0.51	0.00	0.04	6.33	52815	23968	0	00:19:59	00:24:41
		4	0	4	0.81	134568	11020	0.35	0.00	0.05	4.47	51219	16621	0	00:19:10	00:25:30
			1	4	0.81	134568	11020	0.33	0.00	0.04	4.44	51457	16621	0	00:16:58	00:26:11
	1	1	0	1	0.79	160985	57604	-0.53	-44.08	0.04	5.61	51194	18124	4224	00:14:53	00:54:10
			1	1	0.79	160985	57604	0.04	0.00	0.04	5.53	51228	18124	4224	00:14:57	00:35:35
		4	0	4	0.79	138504	12164	-0.27	-2.88	0.05	4.40	46335	17745	0	00:15:29	00:55:11
			1	4	0.79	138504	12164	0.04	0.00	0.05	4.41	46864	17745	0	00:15:46	00:26:13
7	0	1	0	1	0.83	134521	18796	0.64	0.00	0.04	6.32	53134	24374	0	00:22:25	00:25:46
			1	1	0.83	134521	18796	0.54	0.00	0.04	6.29	53069	24374	0	00:19:27	00:26:14
		4	0	4	0.81	134559	11000	0.51	0.00	0.05	4.64	52489	16603	0	00:16:13	00:27:01
			1	4	0.81	134559	11000	0.29	0.00	0.05	4.64	52811	16603	0	00:16:31	00:26:33
	1	1	0	1	0.83	160985	57797	0.14	0.00	0.04	5.50	50419	18330	4224	00:16:54	01:50:51
			1	1	0.83	160985	57797	0.20	0.00	0.04	5.53	50947	18330	4224	00:17:10	00:33:30
		4	0	4	0.83	138495	12245	0.12	0.00	0.04	4.41	46432	17839	0	00:17:17	02:12:08
			1	4	0.83	138495	12245	0.52	0.00	0.04	4.39	46818	17865	0	00:16:31	00:31:57
8	0	1	0	1	0.83	135033	20120	0.41	0.00	0.05	6.42	52949	24773	0	00:18:00	00:26:34
			1	1	0.83	135033	20120	0.72	0.00	0.03	6.30	52688	24773	0	00:17:35	00:25:34
		4	0	4	0.81	134566	11487	0.31	0.00	0.05	4.68	54463	17090	0	00:17:38	00:26:42
			1	4	0.81	134566	11487	0.38	0.00	0.04	4.68	53873	17090	0	00:18:11	00:26:30
	1	1	0	1	0.83	161049	58315	0.46	0.00	0.04	5.59	50499	18734	4224	00:18:17	00:28:58
			1	1	0.83	161049	58315	0.37	0.00	0.04	5.61	50595	18760	4224	00:18:01	00:26:29
		4	0	4	0.83	138502	12636	0.62	0.00	0.04	4.36	45869	18262	0	00:17:04	00:26:16
			1	4	0.83	138502	12636	0.62	0.00	0.04	3.88	45896	18259	0	00:17:29	00:26:41

II

Iteration Interval



- Clock frequency: 160 MHz (6.25 ns)
- M=0; II=1 → highest WNS with L=5 (540 ps)
- II=4 → reduces dissipated power but also WNS
- Synthesis and Implementation time is low for any option

Comparative Study

HLS → 12% ↑ FFs

Option	WNS	TNS	WHS	Power	LUT	FF	LUTR
RTL $\{L=5, M=0, H=3, R=0\}$	0.37	0	0.08	6.3	63593	20281	0
HLS $\{L=5, M=0, II=1, R=1\}$	0.54	0	0.04	6.3	54216	23144	0

HLS → 50% ↑ WNS

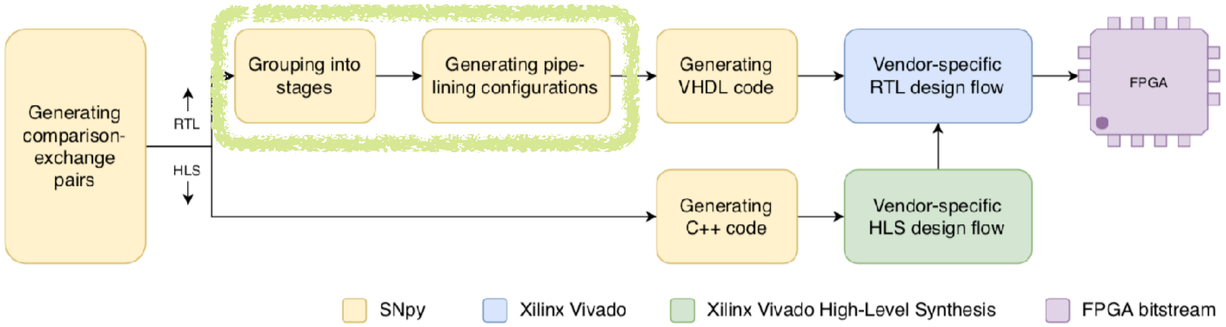
HLS → 15% ↓ LUTs

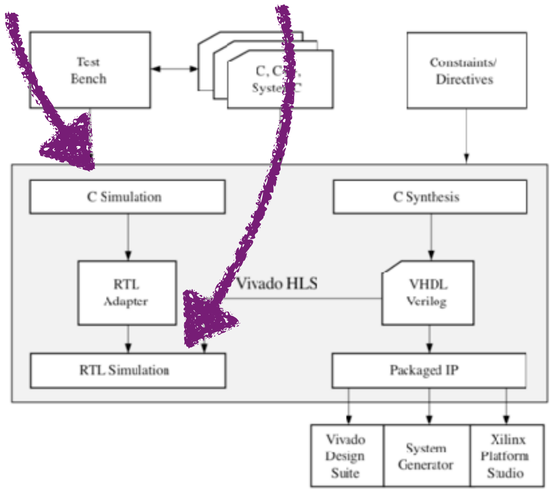
Parallelization and pipelining is not required to be described in HLS (lower design time)

C code has higher abstraction, more time to design instead of mechanical RTL tasks

Early verification, already available when only the functionality is expressed in C code.

Lower synthesis and implementation time (up to 100X)





Δ_{TS}	Δ_{TI}	Δ_{TS}	Δ_{TI}
00:21:12	00:35:18	00:17:26	00:24:50
00:22:55	00:35:22	00:17:22	00:24:32
15:48:26	00:36:54	00:19:28	00:26:44
16:33:34	00:41:42	00:16:25	00:26:07
00:17:53	00:37:28	00:17:02	02:20:03
00:17:50	00:42:44	00:20:55	07:42:51
23:31:01	00:38:21	00:16:22	00:49:01
23:32:48	00:42:50	00:16:33	01:13:16

Summary

Summary

- Part I
  - Development of MUCTPI demonstrator
  - Software packages to automate the testing of hundreds of high-speed serial links
  - $BER < 9 \times 10^{-16}$ with $CL = 95\,\%$ equivalent to 1 bit error per day → 1 fake or lost trigger per day
  - FPGA MGT latency of ≈ 50 ns and latency uncertainty of 3.125ns
  - Synchronizer IP, 208 SL inputs with low and fixed latency.
    Total data transfer latency 110 ns (200 ns total latency budget)

Summary

- Part II
  - Sorting Networks Python Package
  - MUCTPI sorting network, 13 fewer steps than the 45-step 352-key Batcher
    - Based on Baddar 22-key sorting network and divide-and-conquer method
  - FPGA implementation MUCTPI sorting network, 31.25 ns, RTL and HLS

- Part I & II already integrated to MUCTPI firmware and tested

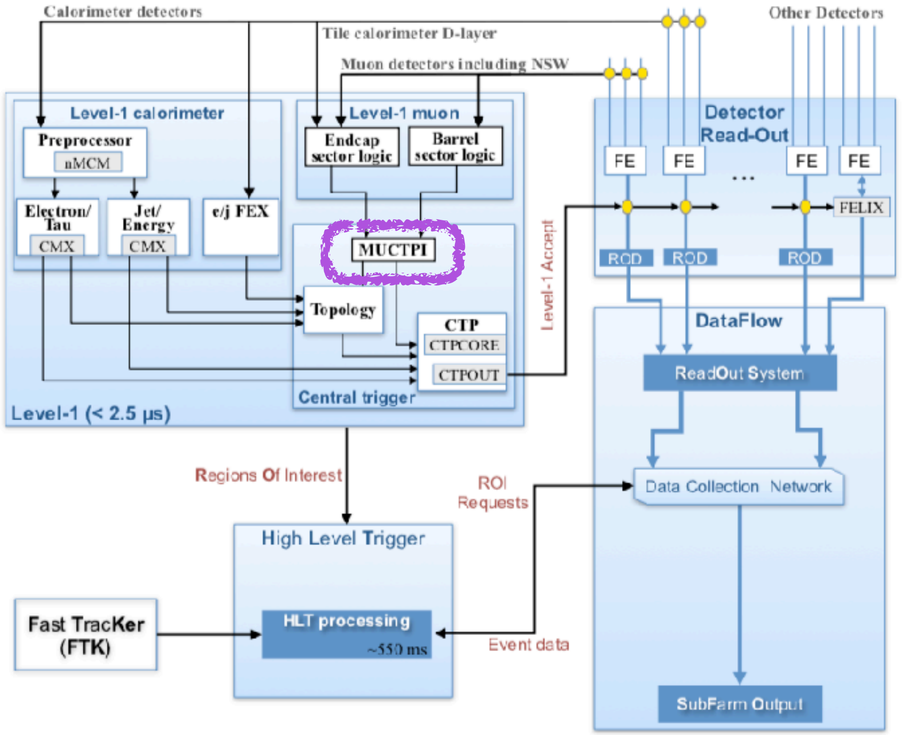
Outlook

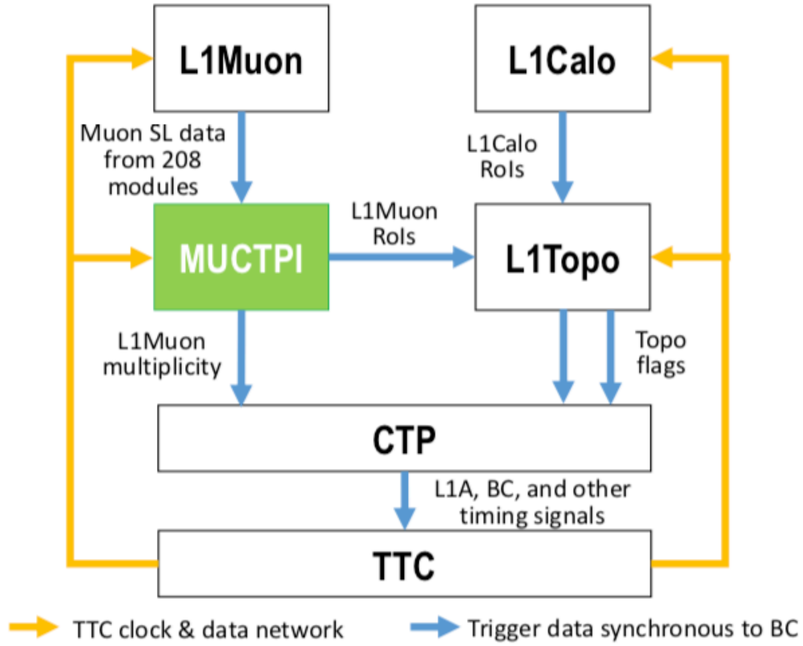
- Automated MGT testing reused to other projects with hundreds of links
- Latency-optimized MGT configurations
- Synchronization IP design and testing
- MUCTPI sorting network experience in low-latency applications
- HLS experience in low-latency applications

Thank You Very Much !

Backup Slides

Level-1 Trigger System / MUCTPI

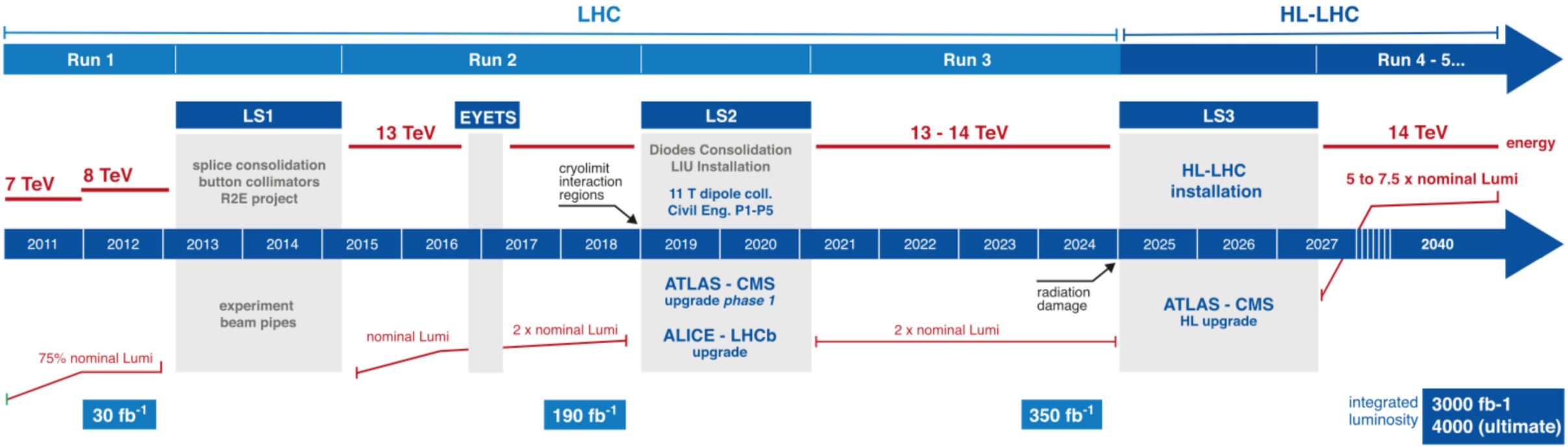




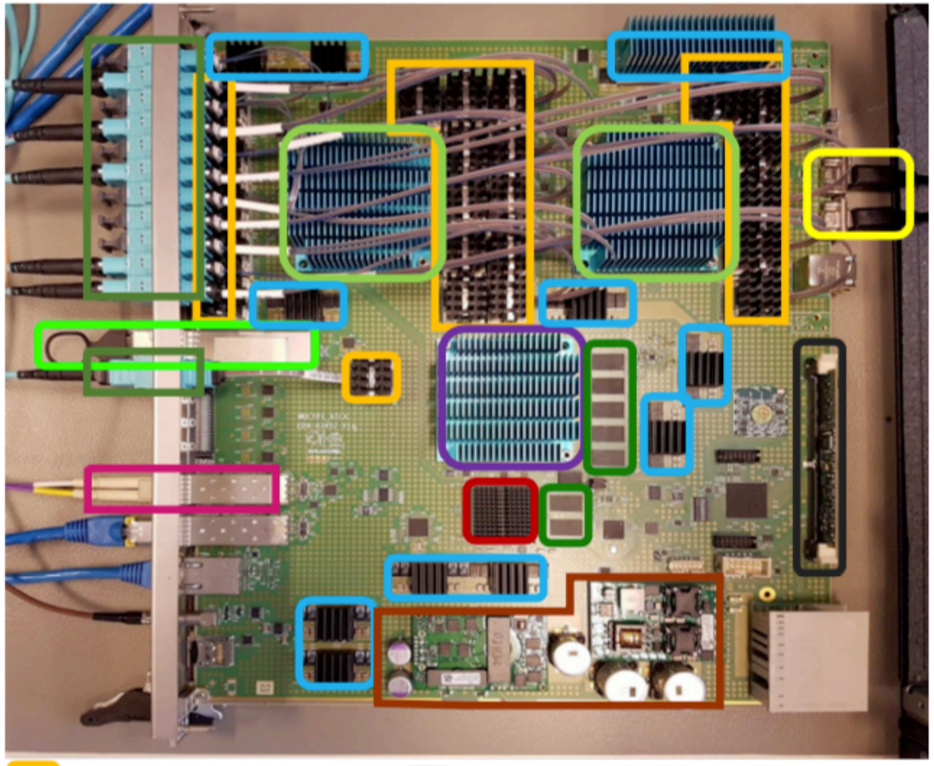
- Trigger / Data Acquisition
- Event rate 40 MHz → 100 kHz (L1) → 1 kHz
- Level-1: subset-detector data, custom, 2.5 us
- MUCTPI: 200 ns

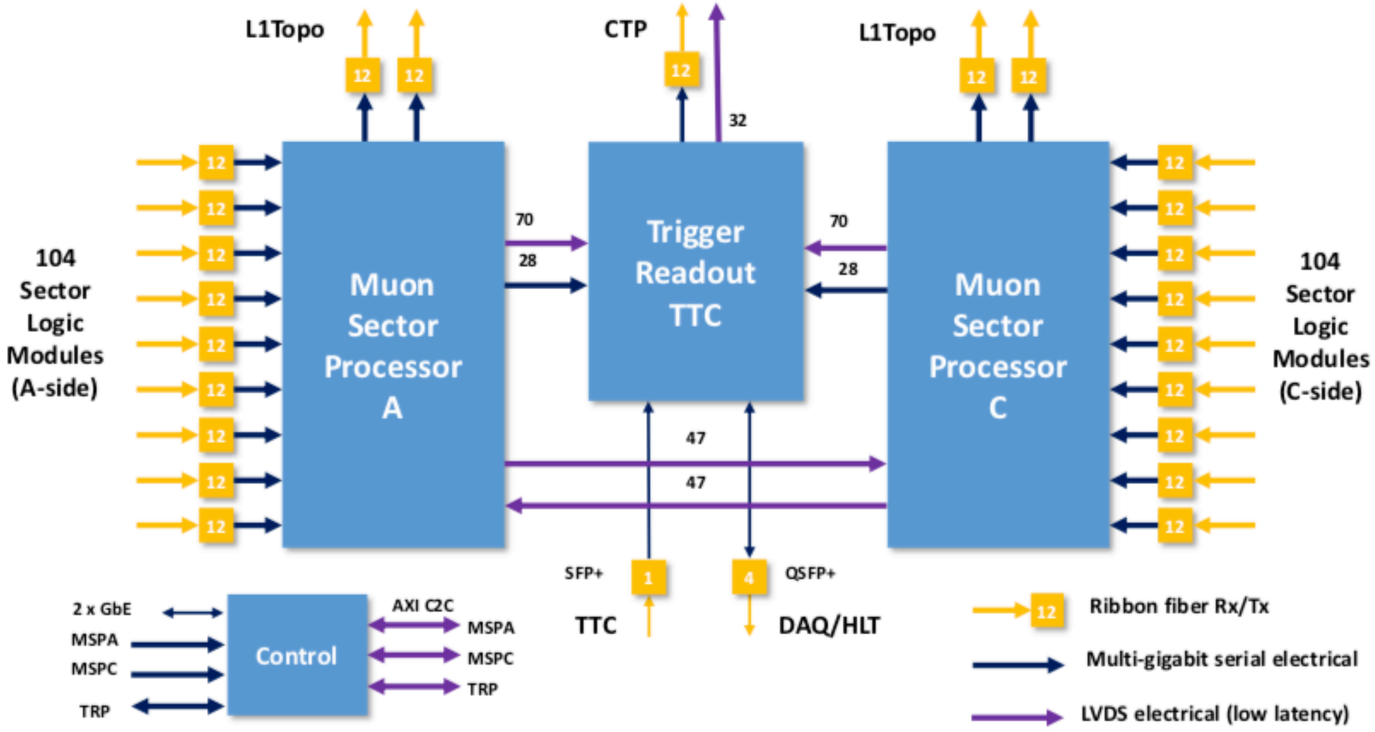
- Real-time / low-latency / high-bandwidth event selection system
- Based on Muon and Calorimeter information
- CTP takes final decision

LHC Upgrade


LHC
HL-LHC
Run 1
Run 2
Run 3
Run 4 - 5...
LS1
EYETS
LS2
LS3
7 TeV
8 TeV
13 TeV
13 - 14 TeV
14 TeV
energy
splice consolidation
button collimators
R2E project
cryolimit
interaction
regions
Diodes Consolidation
LIU Installation
11 T dipole coll.
Civil Eng. P1-P5
HL-LHC
installation
5 to 7.5 x nominal Lumi
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2040
experiment
beam pipes
ATLAS - CMS
upgrade phase 1
ALICE - LHCb
upgrade
radiation
damage
ATLAS - CMS
HL upgrade
75% nominal Lumi
nominal Lumi
2 x nominal Lumi
2 x nominal Lumi
30 fb-1
190 fb-1
350 fb-1
integrated
luminosity
3000 fb-1
4000 (ultimate)

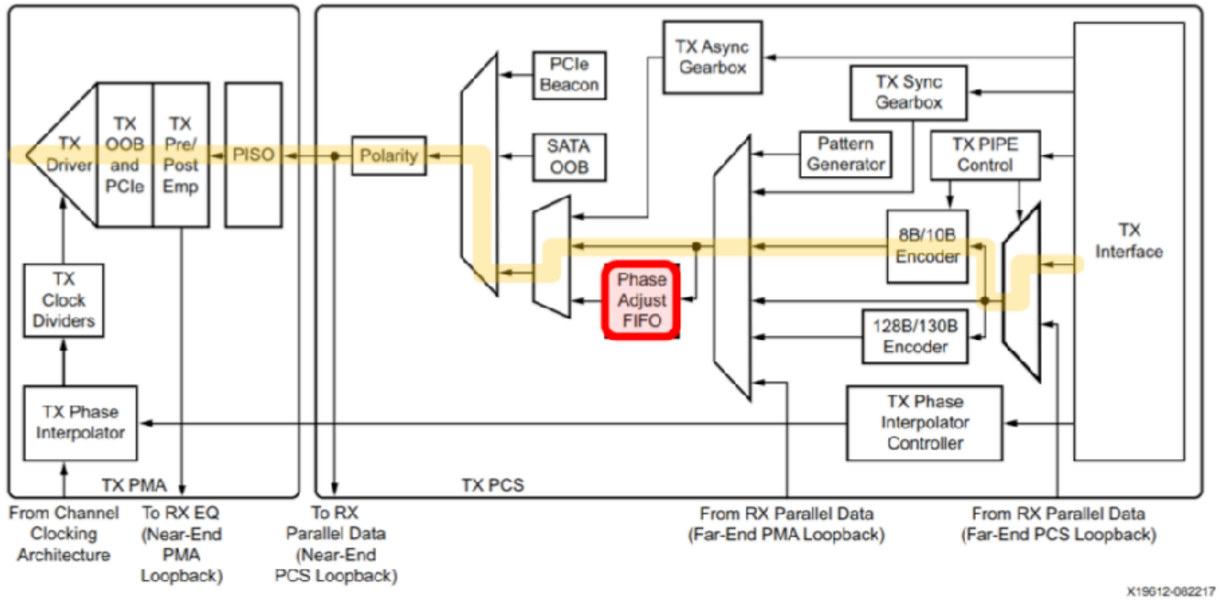
MUCTPI Prototype

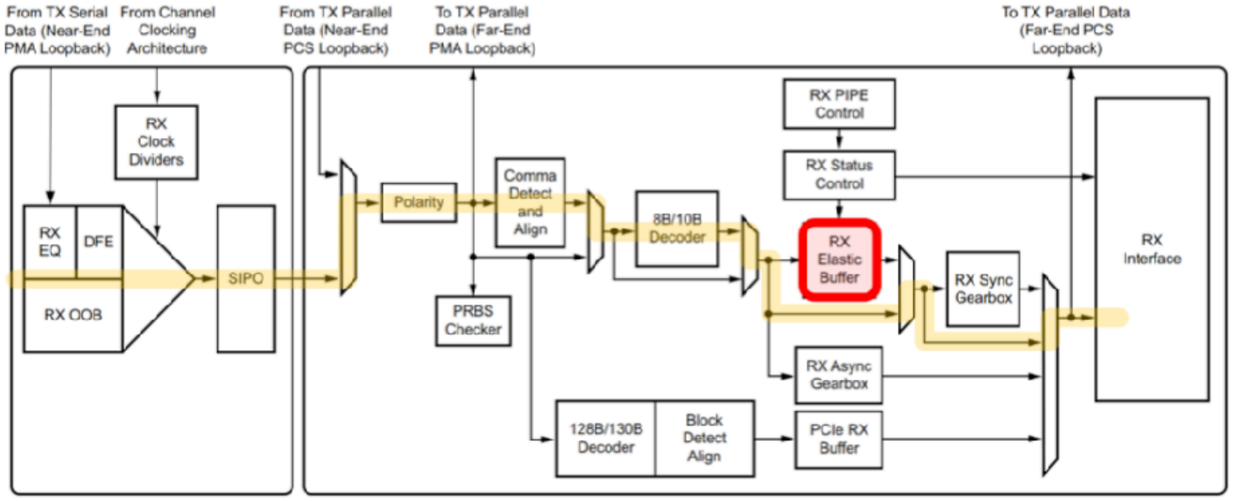




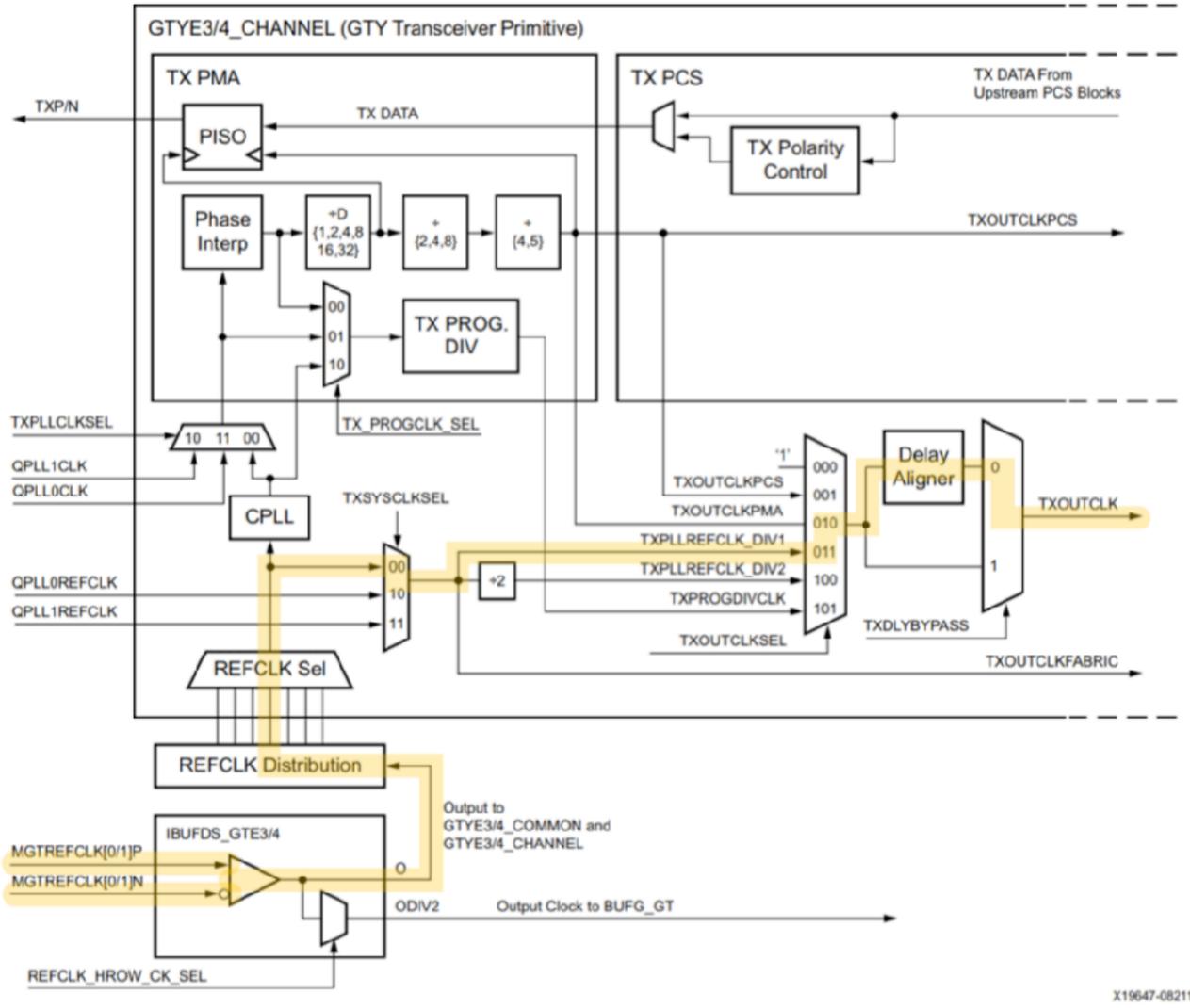
FPGA	Version 1	Version 2	Version 3
MSP	Ultrascale VU160	Ultrascale+ VU9P	
TRP	Ultrascale KU095		
SoC	Zynq-7000 7Z030 SoC		Zynq Ultrascale+ ZU3EG MPSoC

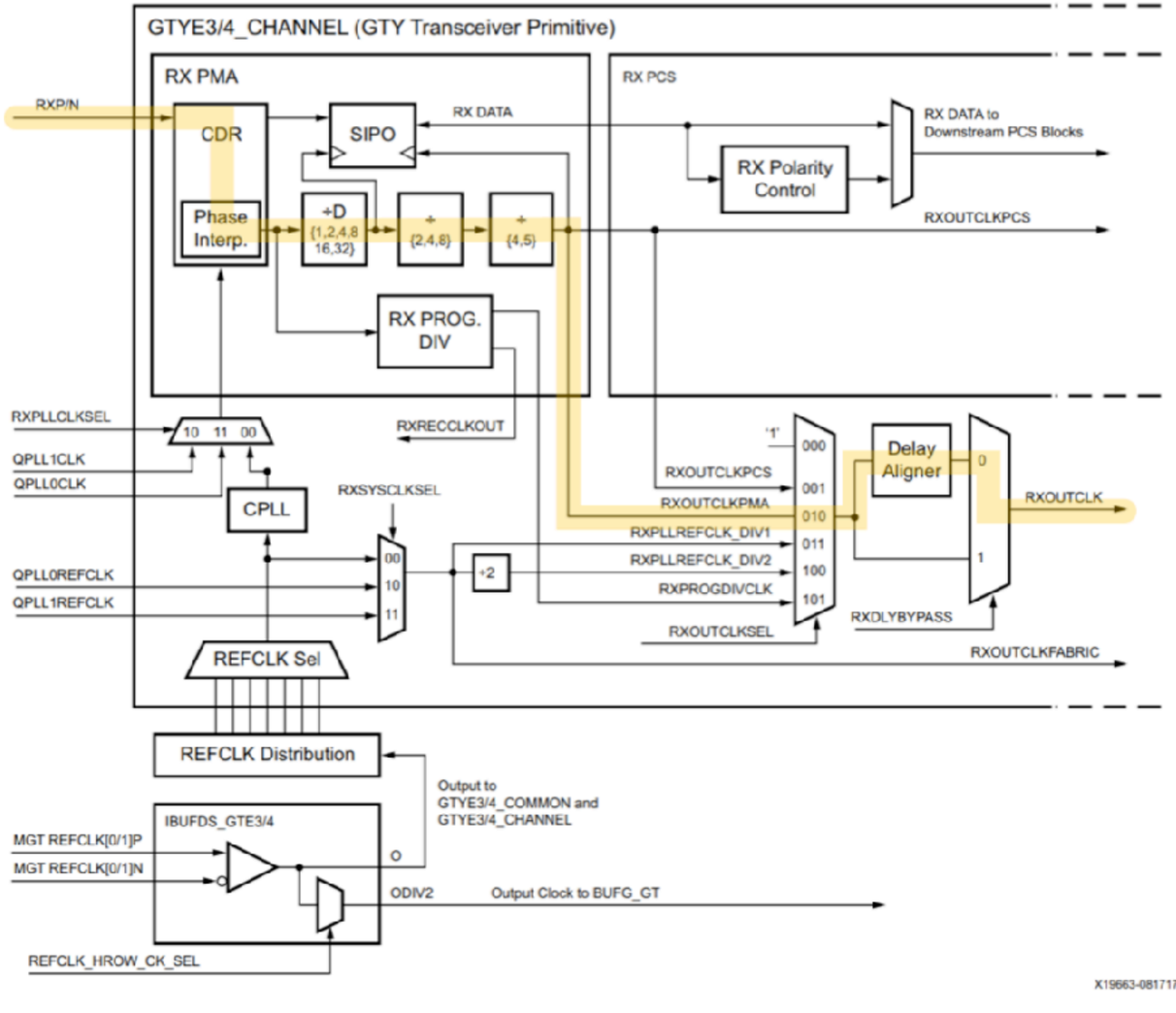
Data-Path Optimization





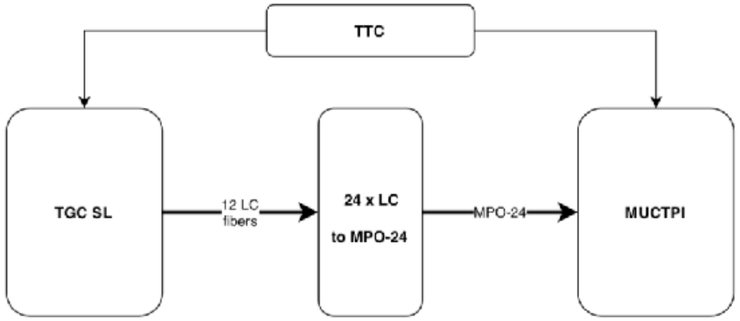
Clock-Fabric Optimization

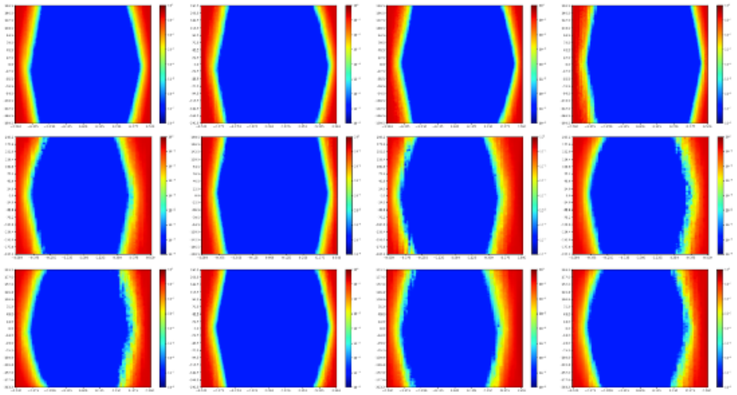




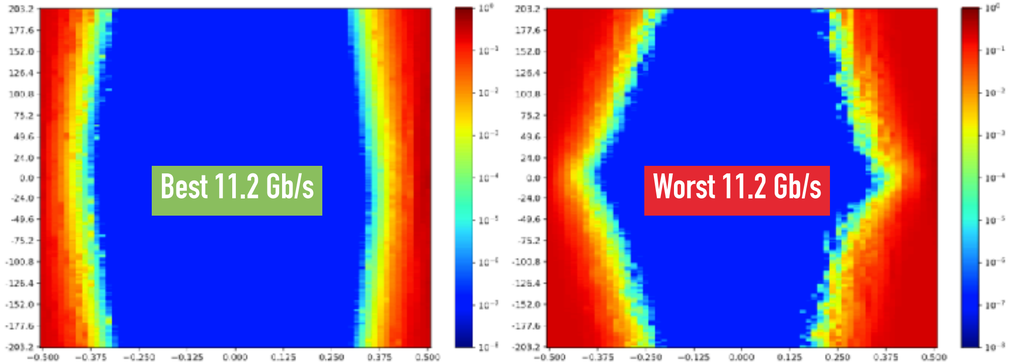
Integration Test Results

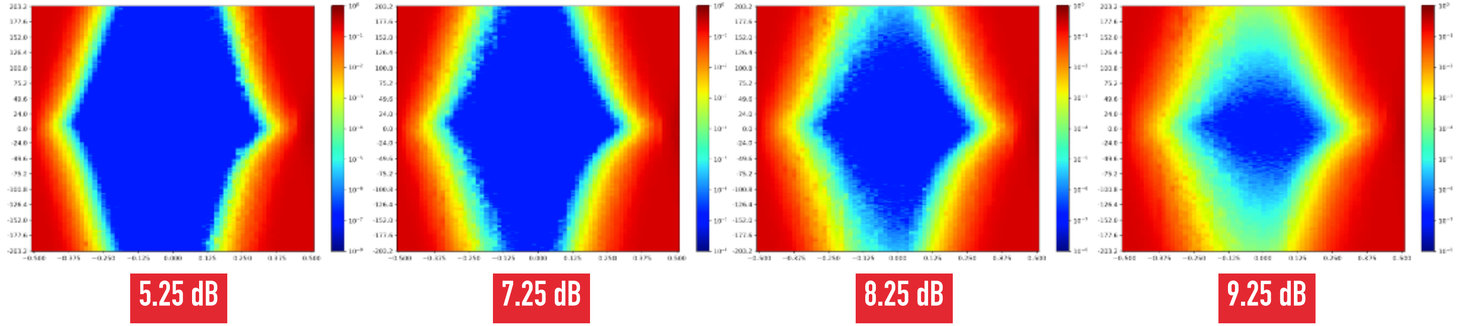
- SL module tests @ 6.4 Gb/s
  - No bit errors after overnight run
  - Very good area eye opening (58%-74%)



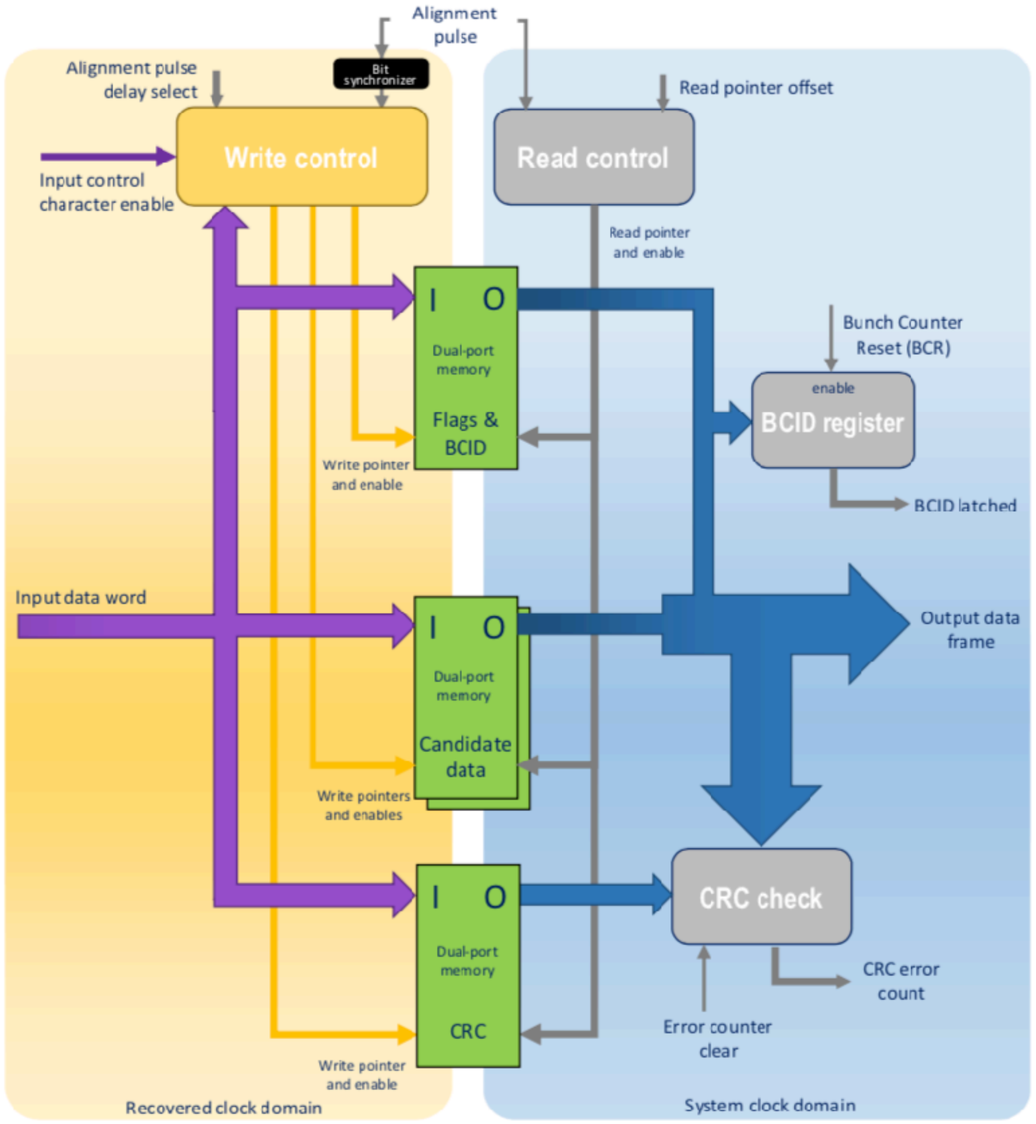


- L1Topo module tests @ 11.2 Gb/s
  - No bit errors after 40 h run
  - Very good area eye opening (50-64%)
- 7 dB power margin for SL modules and L1Topo

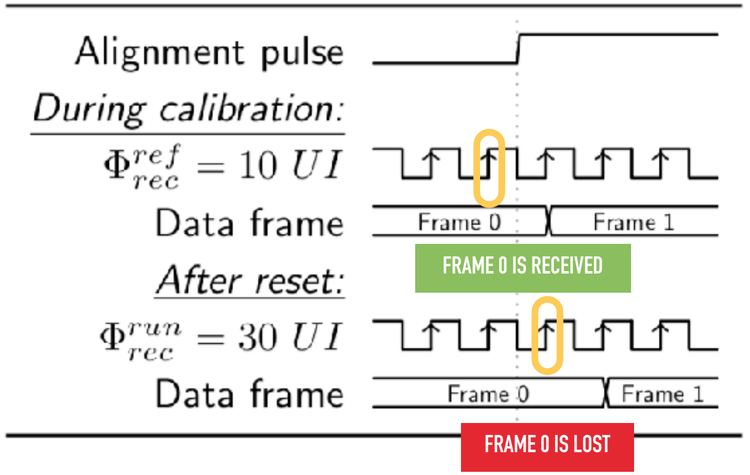




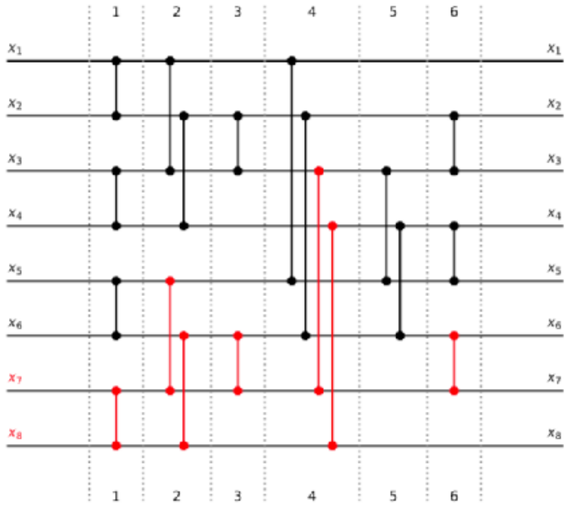
Synchronization Firmware



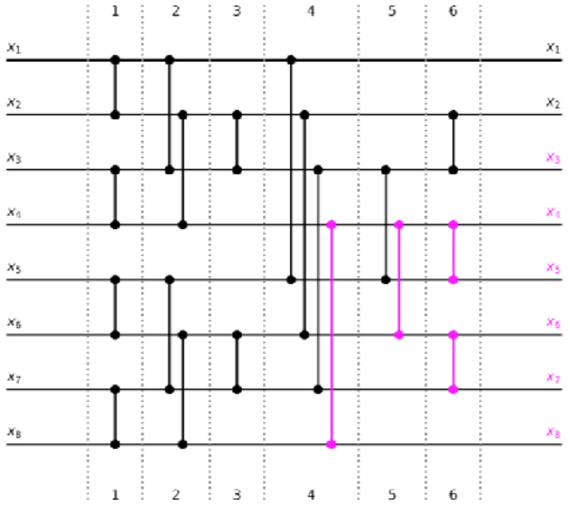
- Dual-port memories:
  - Transfer Rec Clk → Sys clk addressing synchronization & alignment
  - For fixed phase relationship: Write and read pointers engine are enabled simultaneously
  - Write side: End-of-frame character feeds write control to increment write pointer
  - Read side: Increments every 25 ns (40 MHz)
  - What happens when end-of-frame arrives at the same time the engine is enabled?
  - Write control adds configurable delay to alignment pulse to address latency uncertainty
    - Delay is computed from an one-time calibration test
  - Configurable read pointer offsets to address alignment
  - CRC and BCID error are computed at the read side for monitoring and calibration



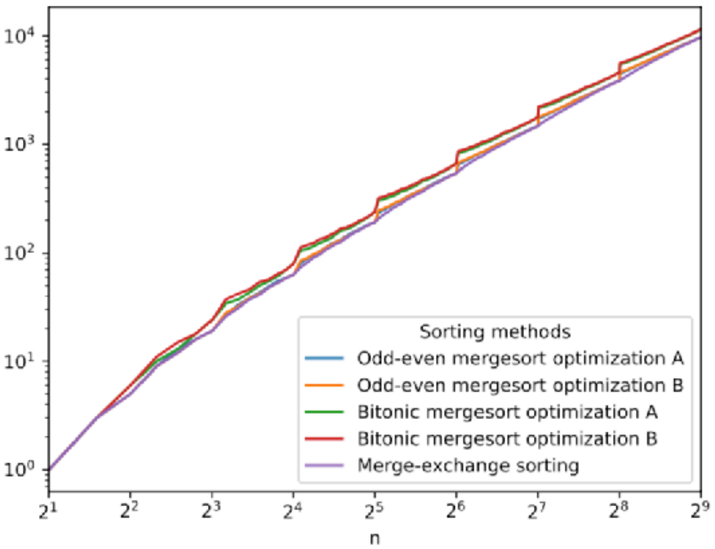
Sorting Networks



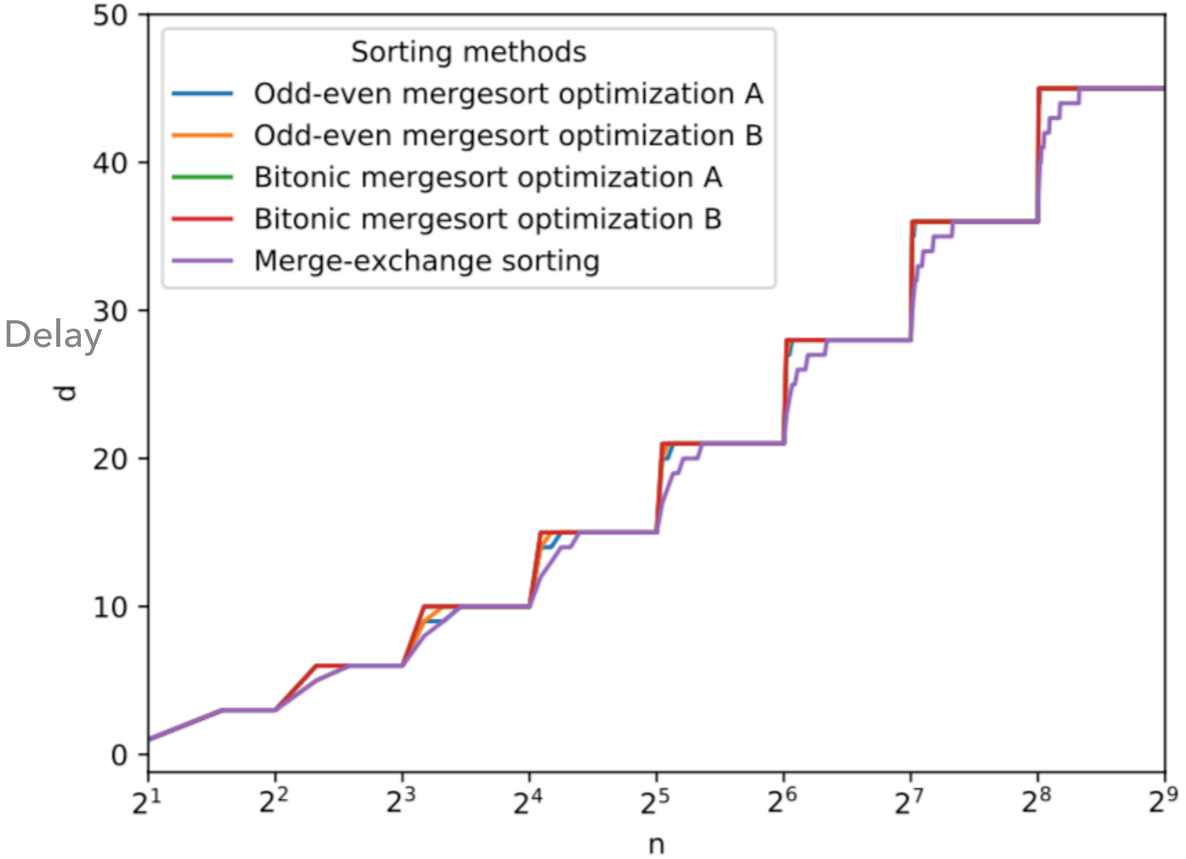
Input optimization



Output optimization



comparisons C



Delay

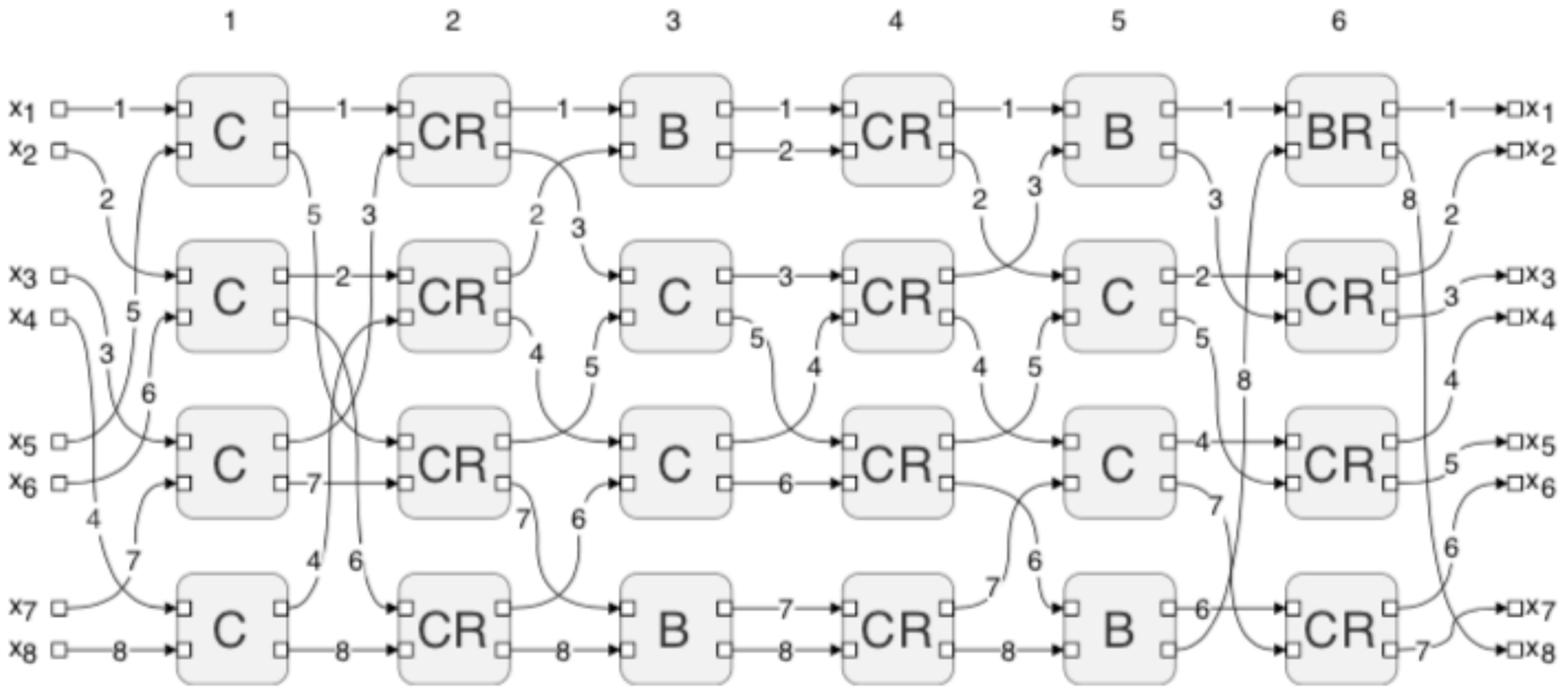
Within Batcher methods, merge-exchange has minimum delay without requiring more comparison-exchanges

0/1 Principle - Merging

Combination	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
9	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
11	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
12	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
13	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
14	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

- Only sorted input combinations
- (16,16) merging:
  - Total combinations: $17^2 = 289$

RTL Implementation



D	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0																																
1																																
2																																
3																																
4																																
5																																
6																																
7																																
8																																

RTL/HLS L1-4

L	M	H	R	WNS	TNS	WHS	Power	LUT	FF	LUTR	Δ_{TS}	Δ_{TI}
1	0	3	0	-15.02	-5574.77	0.09	7.01	100855	6034	0	00:21:01	00:58:31
			1	-17.34	-6547.35	0.05	7.91	60378	6034	0	00:21:19	00:46:00
		2	0	-	-	-	-	-	-	-	-	-
			1	-	-	-	-	-	-	-	-	-
	1	3	0	-21.1	-7396.16	0.09	5.49	60652	6034	0	00:15:32	02:02:50
			1	-21.86	-7841.56	0.05	6.22	55060	6034	0	00:16:44	02:22:25
		2	0	-21.57	-7535.19	0.21	5.49	60699	6034	0	00:29:09	12:30:44
			1	-28.14	-9649.24	0.05	6.49	60455	6034	0	00:28:38	24:09:38
2	0	3	0	-5.79	-16178.55	0.09	6.96	98301	9146	0	00:20:46	00:53:17
			1	-6.52	-18950.93	0.1	7.65	61231	9146	0	00:19:52	00:45:47
		2	0	-5.53	-15961.54	0.05	6.93	98462	9157	0	72:48:29	00:54:46
			1	-6.18	-18225.33	0.05	7.44	72399	9157	0	72:35:57	00:53:00
	1	3	0	-14.41	-5496.41	0.04	5.05	63030	10656	0	00:15:55	00:55:54
			1	-15.88	-10374.03	0.05	5.81	55055	10947	0	00:15:44	00:47:40
		2	0	-	-	-	-	-	-	-	-	-
			1	-	-	-	-	-	-	-	-	-
3	0	3	0	-1.92	-9469.74	0.06	6.62	73507	13567	0	00:19:22	00:47:45
			1	-2.57	-12087.92	0.07	7.47	63163	13565	1	00:20:28	00:46:11
		2	0	-1.78	-8113.61	0.05	6.51	73680	13616	1	38:47:49	00:55:25
			1	-2.15	-11337.58	0.05	7.37	74694	13616	1	40:50:12	00:55:34
	1	3	0	-4.5	-6689.91	0.04	5.02	62277	16331	0	00:17:55	00:46:57
			1	-5.71	-10548.47	0.04	5.78	55063	16649	1	00:17:17	00:46:49
		2	0	-4.6	-7043.31	0.04	5.08	62585	16332	0	72:42:47	00:48:28
			1	-5.57	-10217.15	0.04	5.73	56460	16652	5	72:11:06	00:47:43
4	0	3	0	0.01	0	0.05	6.39	69663	16740	0	00:22:25	00:39:58
			1	-0.52	-1609.48	0.06	7.21	59326	16737	1	00:22:00	00:44:29
		2	0	0.02	0	0.04	6.42	67995	16774	1	26:01:10	00:45:42
			1	-0.47	-975.85	0.05	7.34	67138	16724	25	27:17:57	00:50:41
	1	3	0	-1.38	-2661.11	0.04	5.02	59492	14136	4224	00:17:26	00:45:20
			1	-1.64	-4558.7	0.04	5.71	58139	14397	4225	00:18:21	00:47:41
		2	0	-0.98	-1954.83	0.04	5.03	59553	14149	4225	39:52:58	00:49:50
			1	-1.98	-5498.06	0.05	5.8	61097	14418	4237	40:31:22	00:43:55

Options				HLS				HLS-driven RTL								
L	M	II	R	II'	WNS	LUT	FF	WNS	TNS	WHS	Power	LUT	FF	LUTR	Δ_{TS}	Δ_{TI}
1	0	1	0	1	-23.12	134521	402	-21.12	-7909.24	0.08	8.11	73329	6036	0	00:23:59	00:50:44
			1	1	-23.12	134521	402	-22.51	-8472.54	0.13	8.28	73599	6038	0	00:25:40	00:46:12
		4	0	2	-23.12	134532	402	-20.23	-7522.05	0.05	8.01	73575	6052	0	00:24:21	00:53:34
			1	2	-23.12	134532	402	-21.02	-7913.68	0.06	8.07	72861	6042	0	00:26:43	00:48:25
	1	1	0	1	-24.53	138457	402	-28.25	-10357.58	0.23	6.24	65504	6046	0	00:19:30	08:15:06
			1	1	-24.53	138457	402	-23.36	-8328.31	0.15	6.47	65195	6045	0	00:22:53	00:53:55
		4	0	2	-24.53	138468	402	-36.57	-12005.19	0.06	6.67	67691	6041	0	00:21:30	08:09:28
			1	2	-24.53	138468	402	-26.06	-9253.09	0.06	6.76	69415	6043	0	00:22:19	00:49:54
2	0	1	0	1	-11.27	134521	9066	-8.78	-11507.93	0.06	7.46	73013	14570	0	00:27:02	00:35:02
			1	1	-11.27	134521	9066	-7.83	-13639.70	0.06	7.30	69597	14557	0	00:22:22	00:43:00
		4	0	3	-11.27	134538	9066	-7.39	-11226.14	0.05	6.05	66662	14558	0	00:20:50	00:42:29
			1	3	-11.27	134538	9066	-6.89	-11390.01	0.05	6.17	67907	14555	0	00:23:26	00:48:05
	1	1	0	1	-12.69	138457	8168	-17.74	-19462.54	0.05	6.23	53674	13676	0	00:16:36	08:39:20
			1	1	-12.69	138457	8168	-11.51	-14718.08	0.04	6.06	57962	13686	0	00:20:17	00:48:48
		4	0	3	-12.69	138474	3944	-14.17	-16477.29	0.06	5.57	55487	9454	0	00:23:30	09:02:42
			1	3	-12.69	138474	3944	-13.27	-16152.58	0.08	5.64	55951	9455	0	00:26:37	00:39:42
3	0	1	0	1	-5.08	134521	12151	-2.39	-1869.97	0.05	6.51	57130	17721	0	00:19:11	00:37:36
			1	1	-5.08	134521	12151	-2.19	-1924.01	0.04	6.54	57263	17720	0	00:21:42	00:45:01
		4	0	4	-5.08	134544	9289	-3.13	-2378.44	0.05	4.48	56203	14884	0	00:17:06	00:40:50
			1	4	-5.08	134544	9289	-2.75	-2792.73	0.04	4.58	57212	14876	0	00:18:19	00:40:41
	1	1	0	1	-5.59	138457	14076	-10.05	-5595.34	0.02	5.66	50444	19578	0	00:17:20	08:26:37
			1	1	-5.59	138457	14076	-7.03	-5403.16	0.04	5.57	50895	19568	0	00:18:25	08:00:30
		4	0	4	-5.59	138480	5628	-7.38	-4895.43	0.04	4.71	53736	11123	0	00:26:20	03:59:41
			1	4	-5.59	138480	5628	-5.85	-4292.17	0.04	4.79	52378	11127	0	00:20:48	01:19:19
4	0	1	0	1	-2.86	134521	15378	-1.04	-410.06	0.04	6.41	54430	20911	0	00:19:20	00:30:54
			1	1	-2.86	134521	15378	-0.25	-11.52	0.04	6.52	57197	20911	0	00:25:41	00:41:22
		4	0	4	-2.86	134566	11155	-0.46	-62.50	0.04	5.10	55110	16693	0	00:24:41	00:32:01
			1	4	-2.86	134566	11155	-0.52	-133.72	0.05	5.11	55804	16694	0	00:20:21	00:38:34
	1	1	0	1	-2.86	160985	55891	-3.54	-1807.18	0.05	5.56	51798	16376	4224	00:18:12	02:32:32
			1	1	-2.86	160985	55891	-3.41	-1545.52	0.05	5.66	52788	16381	4224	00:21:07	00:51:04
		4	0	4	-2.86	138502	6597	-3.35	-1573.54	0.05	4.88	49026	12149	0	00:20:08	01:09:17
			1	4	-2.86	138502	6597	-2.79	-1210.14	0.04	4.84	49950	12146	0	00:16:59	00:47:46