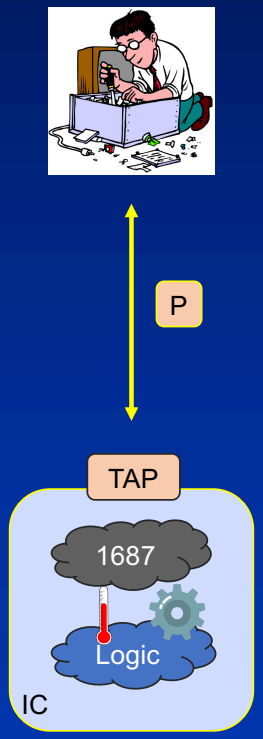
Accessing general IEEE Std. 1687 networks via functional ports

Erik Larsson, Prathamesh Murali, and Zilin Zhang

LUND UNIVERSITY

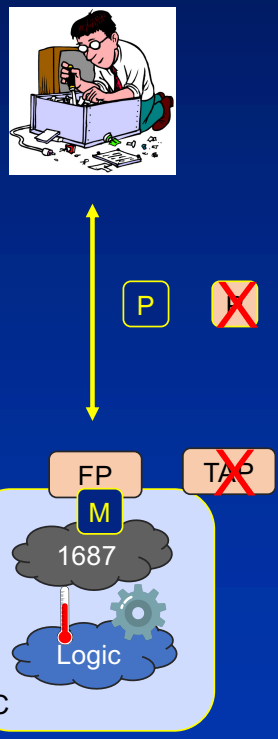
Purpose

- Modern ICs need on-chip instruments for testing, tuning, configuration, and so on
- IEEE Std. 1687 allows flexible and scalable access and IEEE Std. 1149.1 test access port (TAP) is the interface



Purpose

- Modern ICs need on-chip instruments for testing, tuning, configuration, and so on
- IEEE Std. 1687 allows flexible and scalable access and IEEE Std. 1149.1 test access port (TAP) is the interface
- As not all ICs have IEEE Std. 1149.1, IEEE Std. P1687.1 explores if functional ports (FP) can be used?
- We provide a hardware module (M) and protocol (P) with the aim to keep transported data and area low

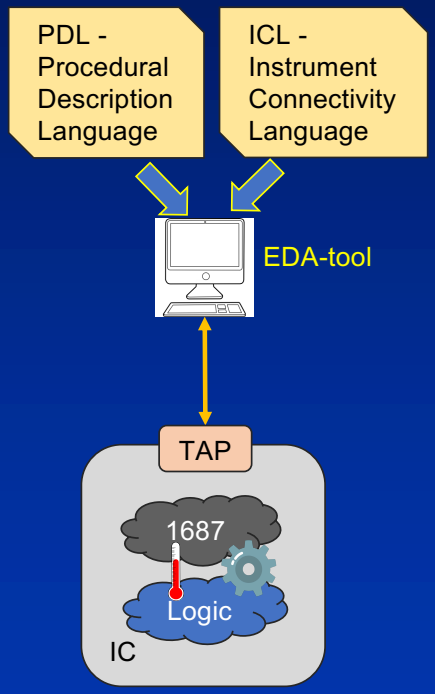


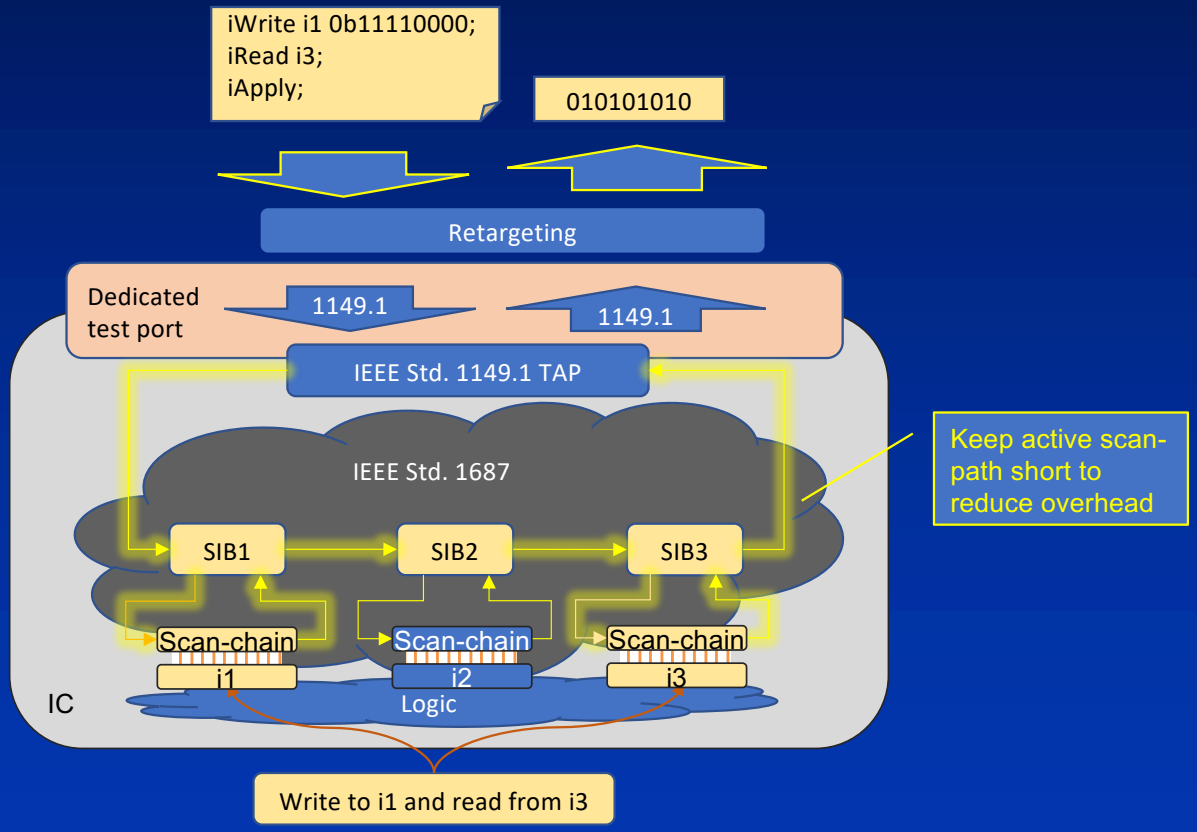
Disclaimer: The views expressed in this presentation are those of the presenter; they do not necessarily reflect the views of the IEEE Std. P1687.1 working group

Outline

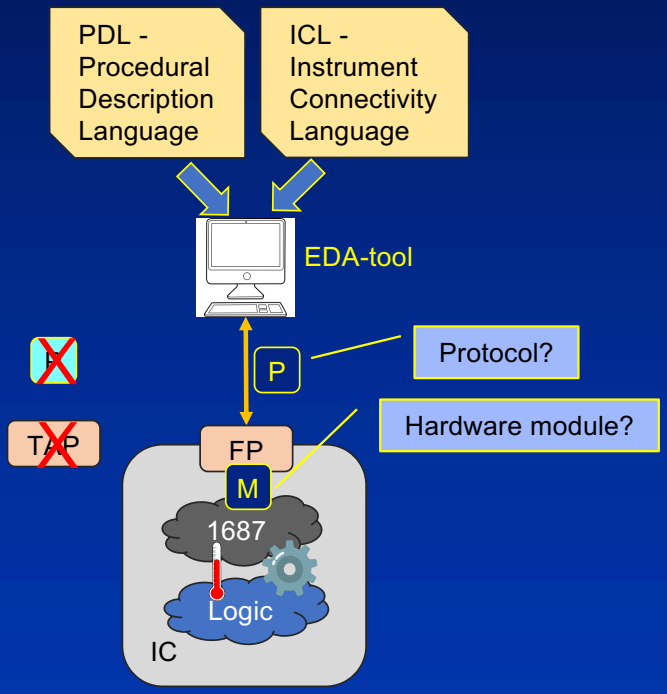
- Introduction: IEEE Std. 1149.1, 1687, 1687.1
- Prior work: Access to flat SIB-based networks
- Proposed technique: Access to general networks
- Experimental results: Three benchmarks with various number of instruments implemented on FPGA
- Conclusions

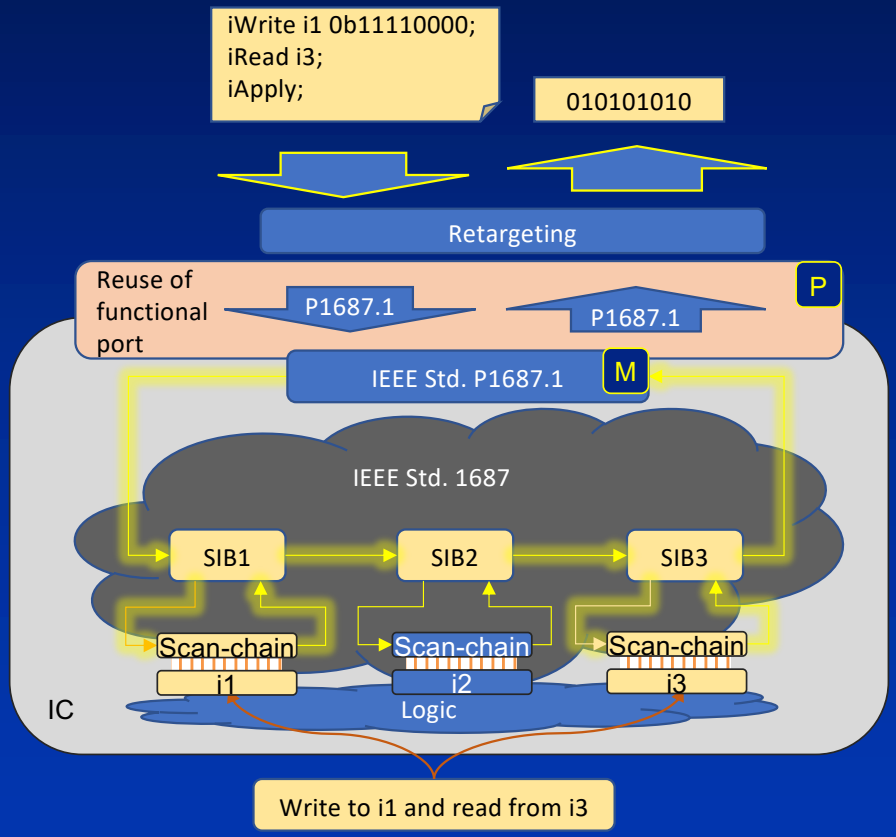
Introduction: IEEE Std. 1149.1 and 1687





Introduction: IEEE Std. P1687.1





Outline

- Introduction: IEEE Std. 1149.1, 1687, 1687.1
- Prior work: Access to flat SIB-based networks
- Proposed technique: Access to general networks
- Experimental results: Three benchmarks with various number of instruments implemented on FPGA
- Conclusions

Towards IEEE Std. P1687.1

E Larsson, P Murali, G Kumisbek,
IEEE Std. P1687. 1: translator and protocol,
2019 IEEE International Test Conference (ITC), 1-10

Data overhead

Control overhead

Order	Comment		Protocol bits
3	Data to be written in i1		1 1 1 1 0 0 0 0
	# data bytes		0 0 0 0 0 0 0 1
	Data command		1 0 0 0 0 0 0 0
2	Instrument i3		0 0 0 0 0 0 1 1
	Control	Read	0 0 0 0 0 0 0 0
1	Instrument i1		0 0 0 0 0 0 0 1
	Control	Write	0 1 0 0 0 0 0 1

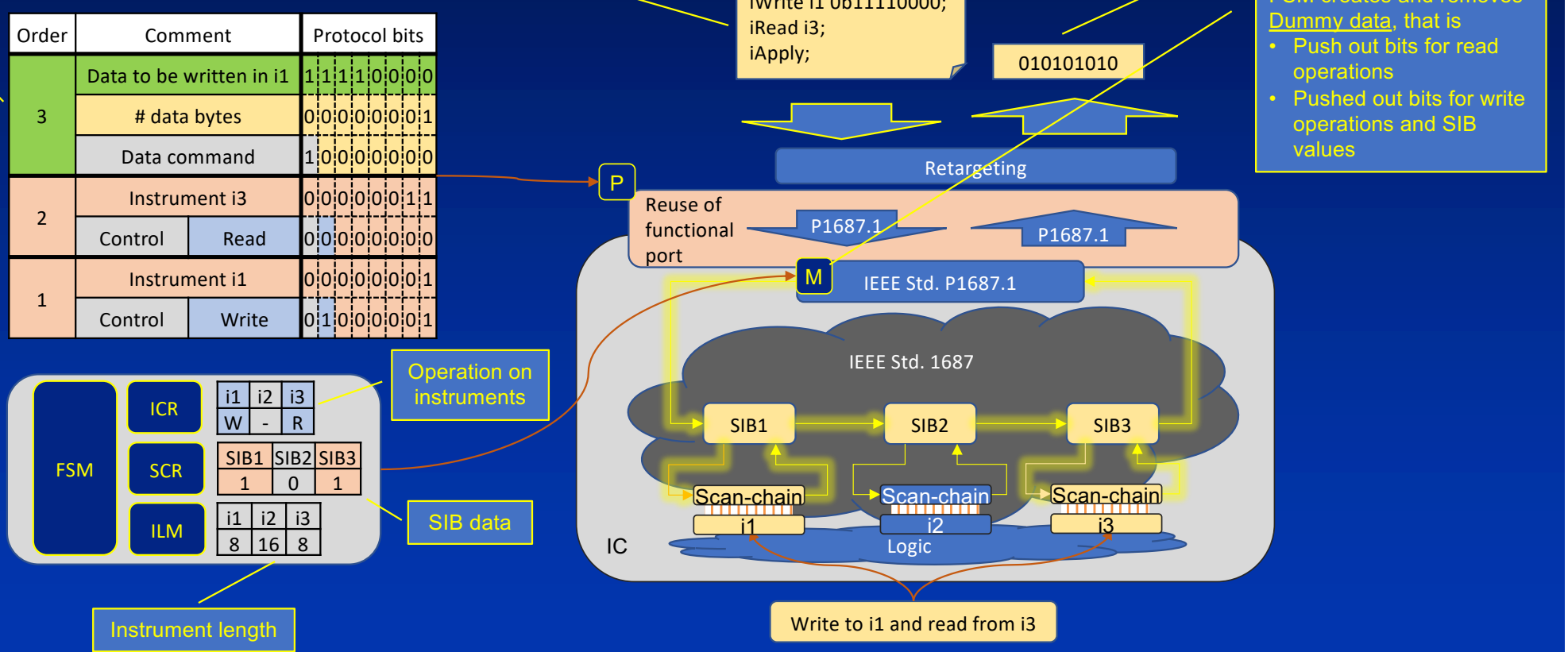
Useful data to be written to i1

iWrite i1 0b11110000;
iRead i3;
iApply;

Useful data read from i3

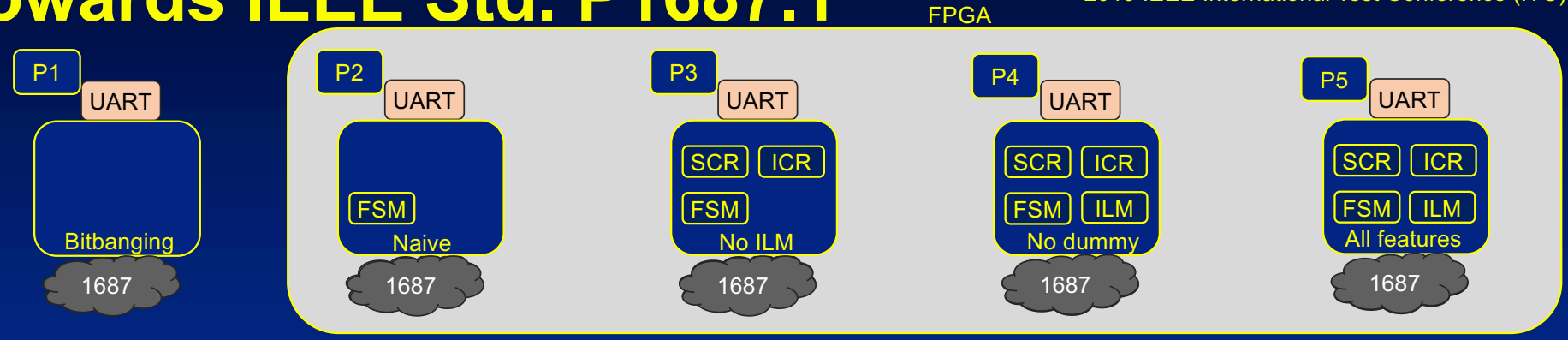
010101010

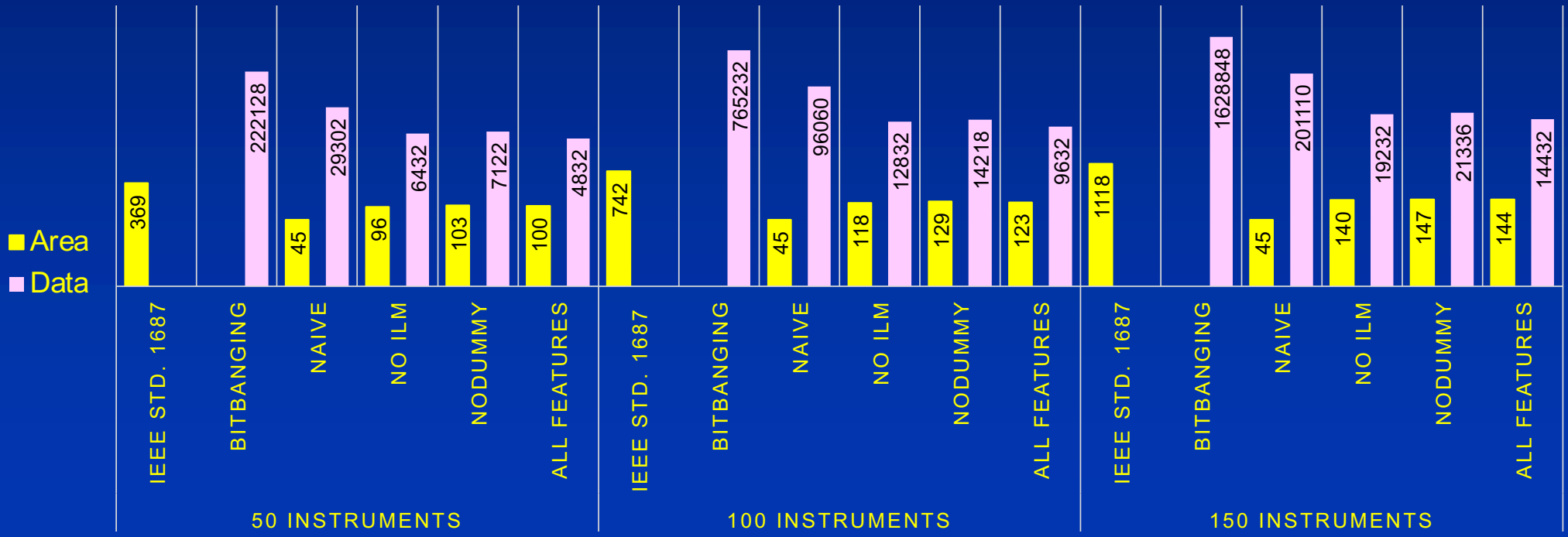
FSM creates and removes Dummy data, that is
- Push out bits for read operations
- Pushed out bits for write operations and SIB values



Towards IEEE Std. P1687.1

E Larsson, P Murali, G Kumisbek,
IEEE Std. P1687. 1: translator and protocol,
2019 IEEE International Test Conference (ITC), 1-10





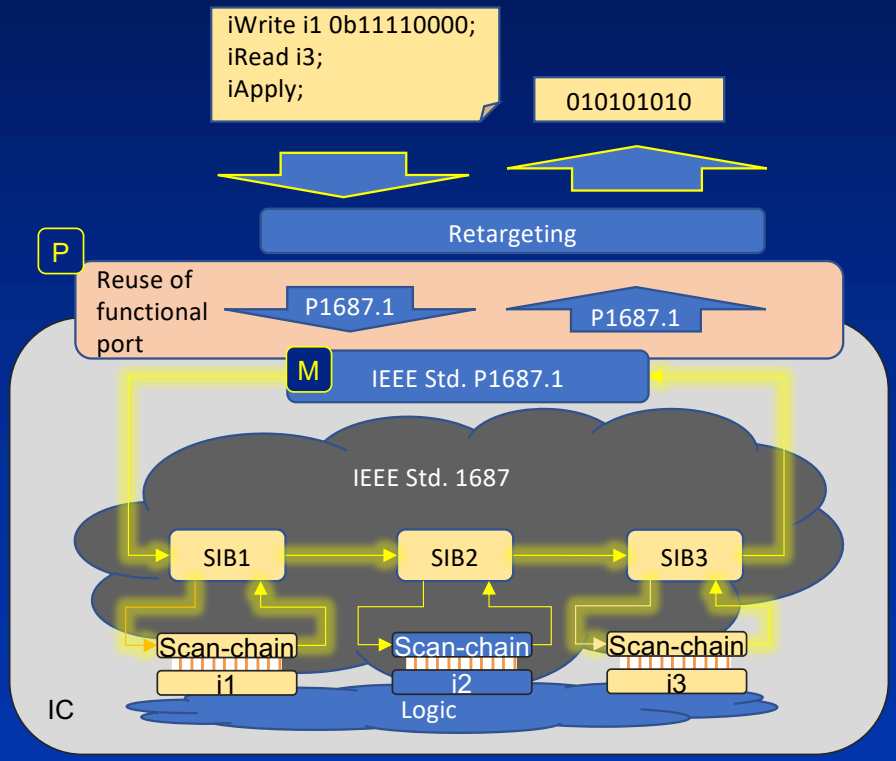
Outline

- Introduction: IEEE Std. 1149.1, 1687, 1687.1
- Prior work: Access to flat SIB-based networks
- Proposed technique: Access to general networks
- Experimental results: Three benchmarks with various number of instruments implemented on FPGA
- Conclusions

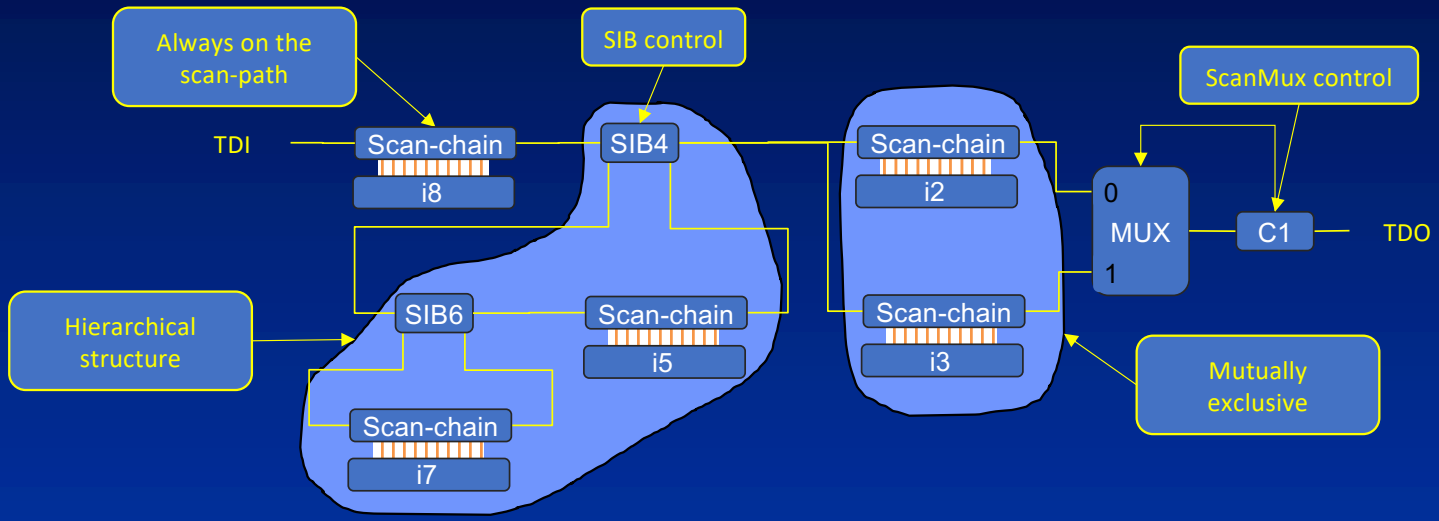
Observations

E Larsson, P Murali, G Kumisbek,
IEEE Std. P1687. 1: translator and protocol,
2019 IEEE International Test Conference (ITC), 1-10

- The IEEE Std. 1687 network:
  - Flat design
  - One SIB per instrument
- Consequences:
  - Each iApply group can be handled independently
  - The FSM makes:
    - A reset
    - Sets the active path
    - Operates on path

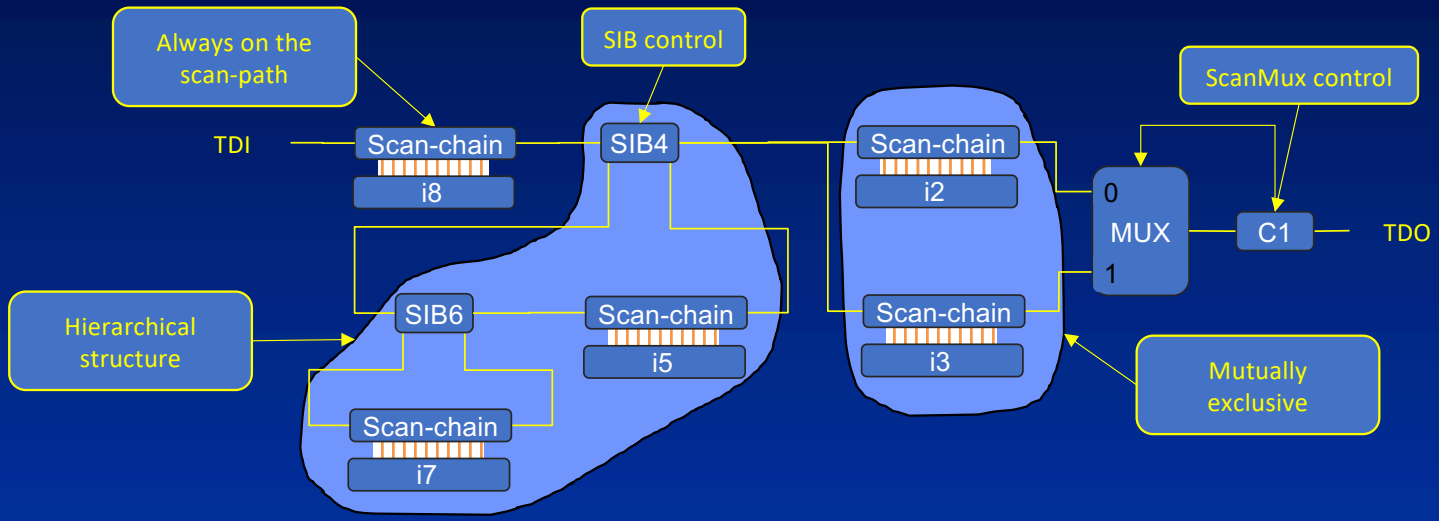


Access and operation



Name	i8	i7	SIB6	i5	SIB4	i3	i2	C1
Number (as they can appear on scan path)	8	7	6	5	4	3	2	1
Length (number of shifts)	10	8	1	10	1	5	8	1
Operation	NOP	W	1	NOP	0	R	NOP	1
Active	1	0	0	0	1	1	0	1

Access and operation

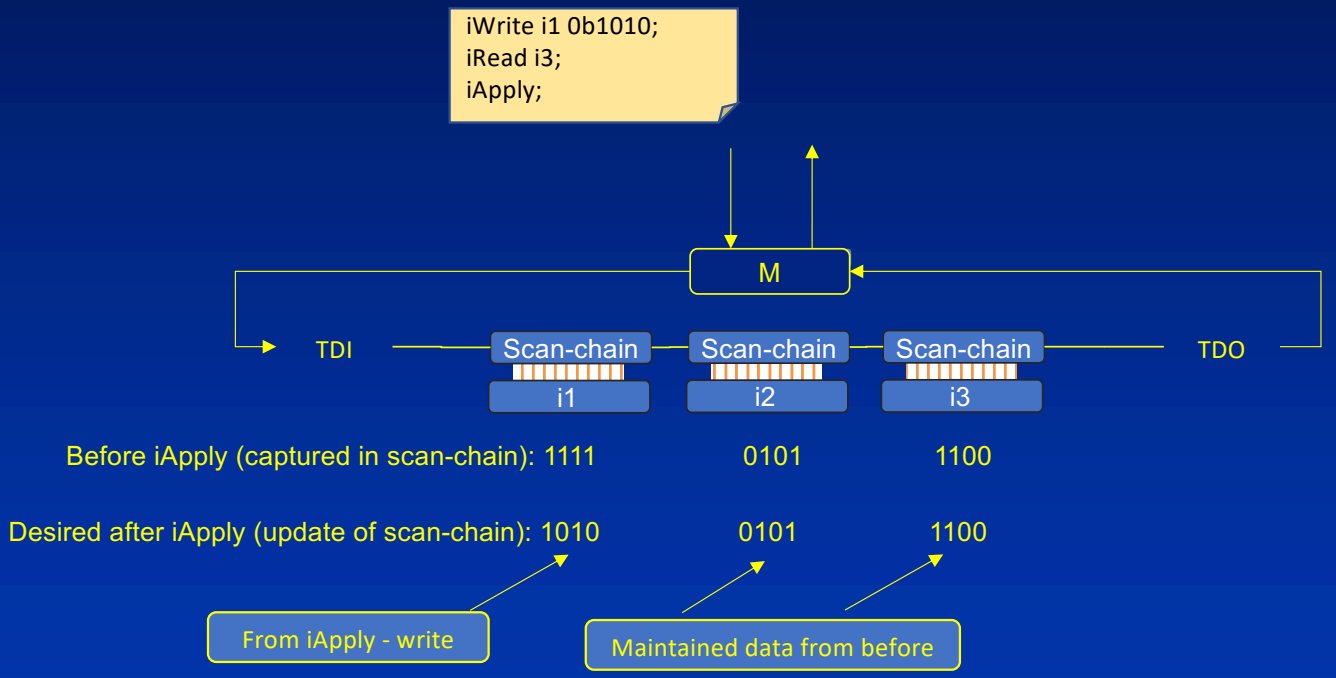


FSM

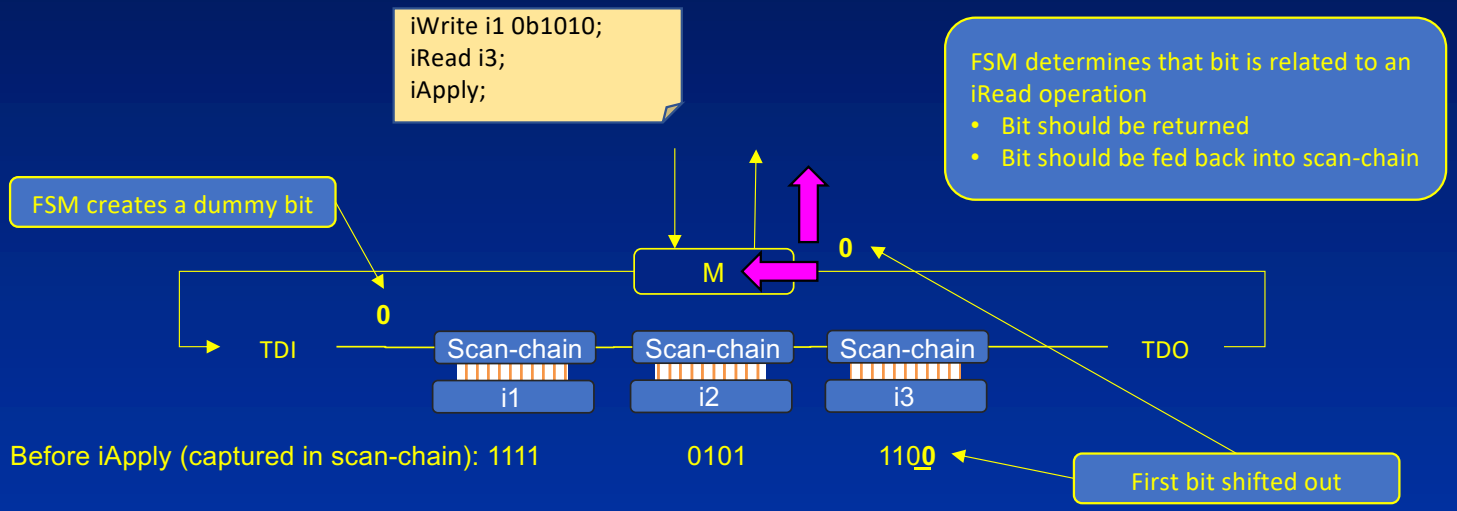
FSM

Name	i8	i7	SIB6	i5	SIB4	i3	i2	C1
Number (as they can appear on scan path)	8	7	6	5	4	3	2	1
Length (number of shifts)	10	8	1	10	1	5	8	1
Operation	NOP	W	1	NOP	0	R	NOP	1
Active	1	0	0	0	1	1	0	1
	??????????	-	-	-	0	?????	-	1

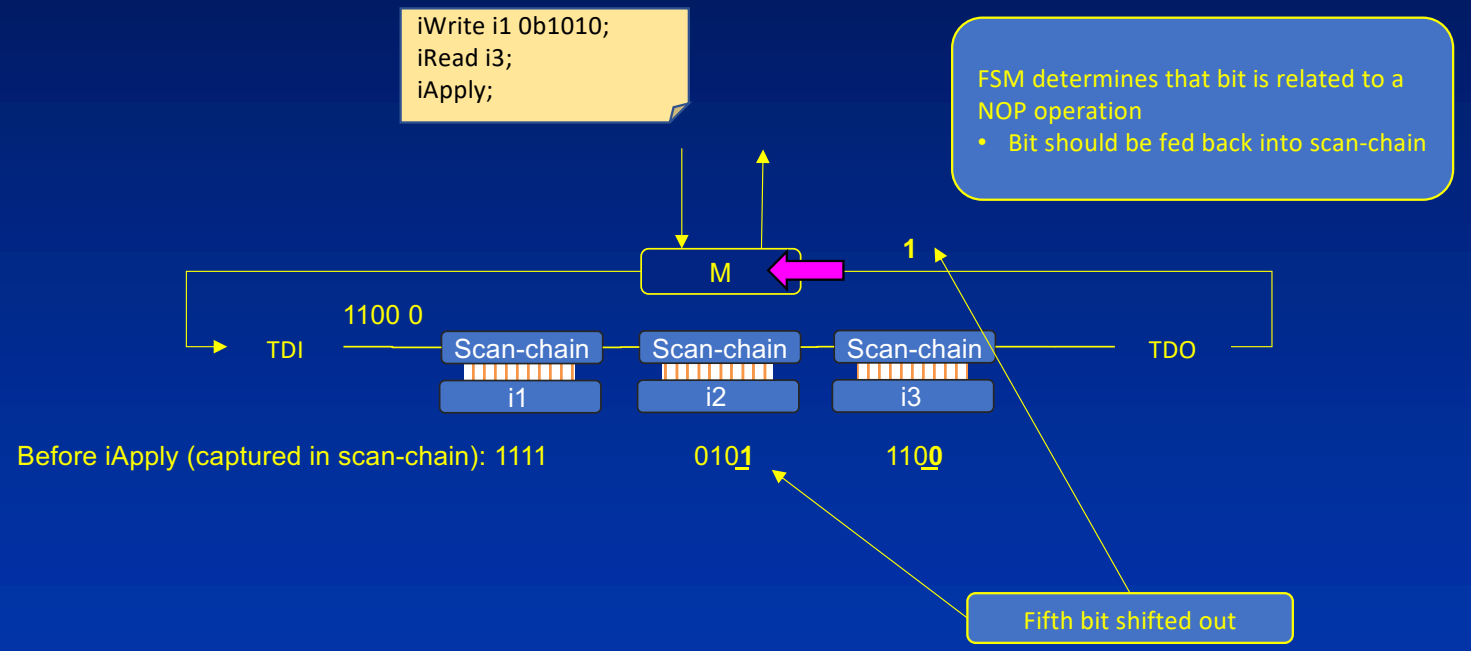
Active scan-path



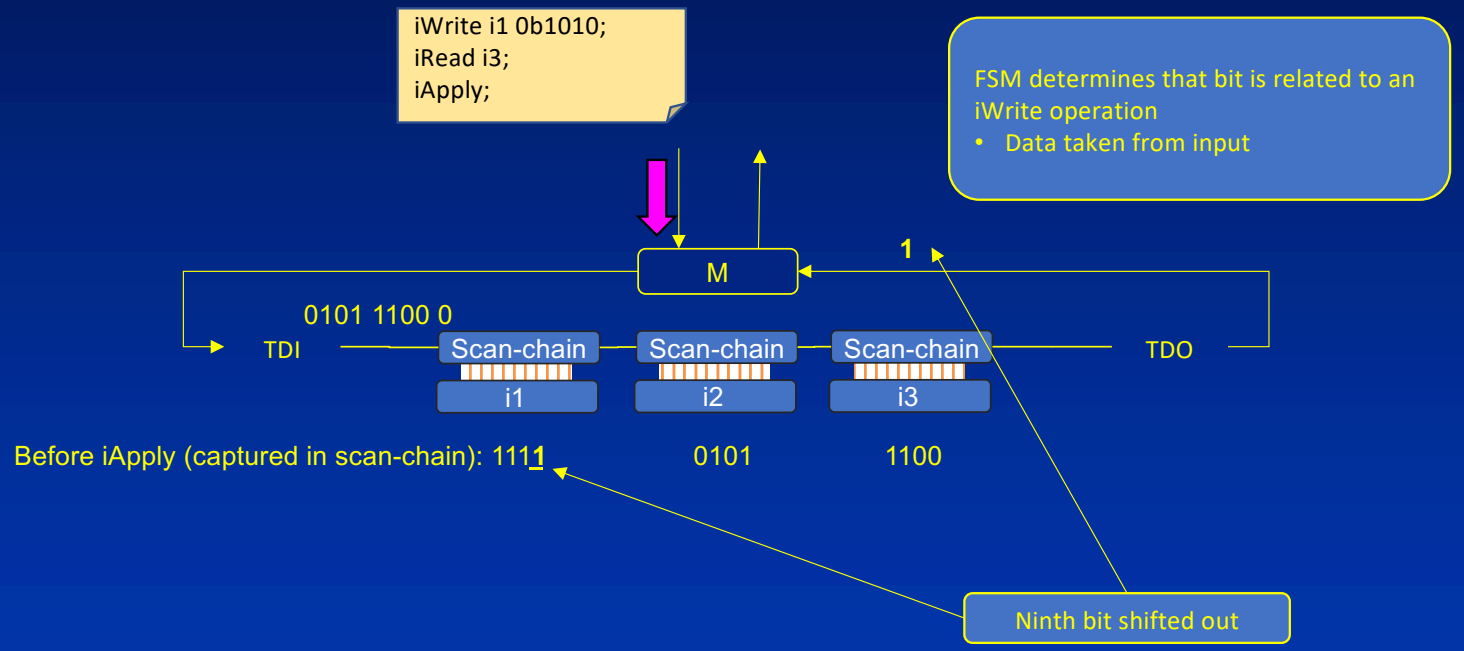
Active scan-path



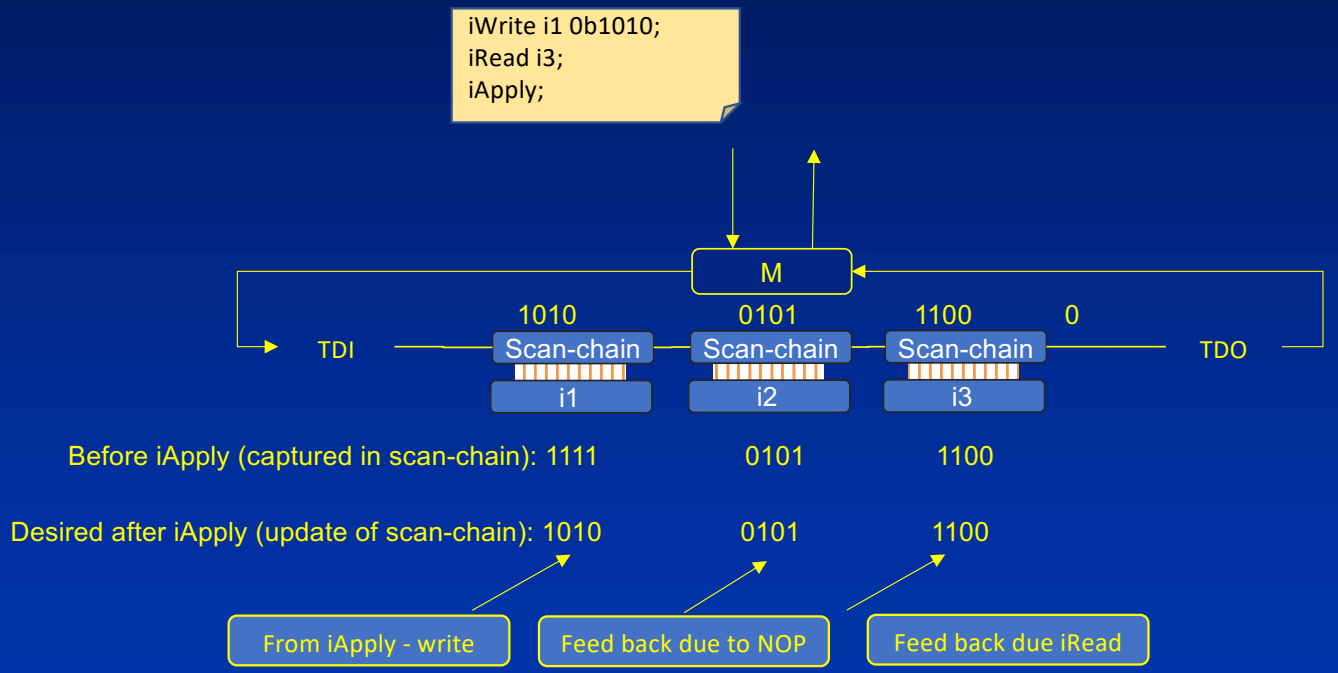
Active scan-path



Active scan-path



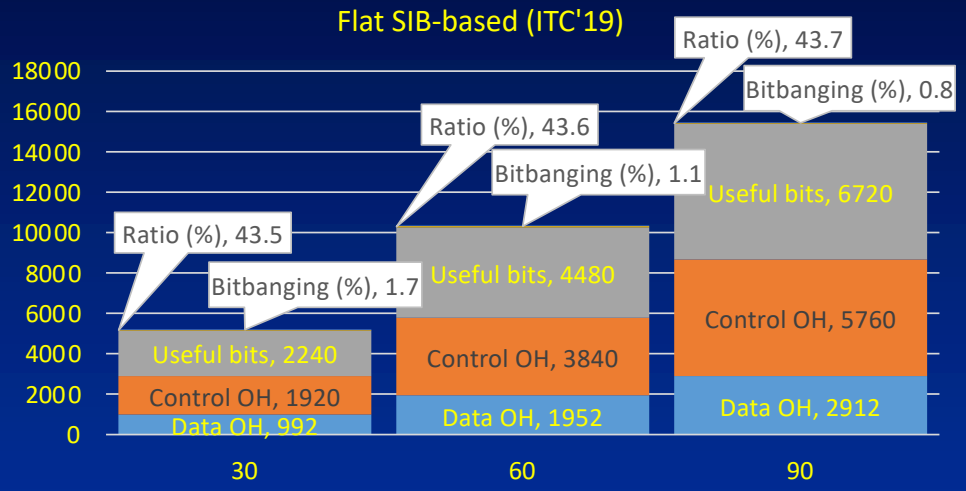
Active scan-path

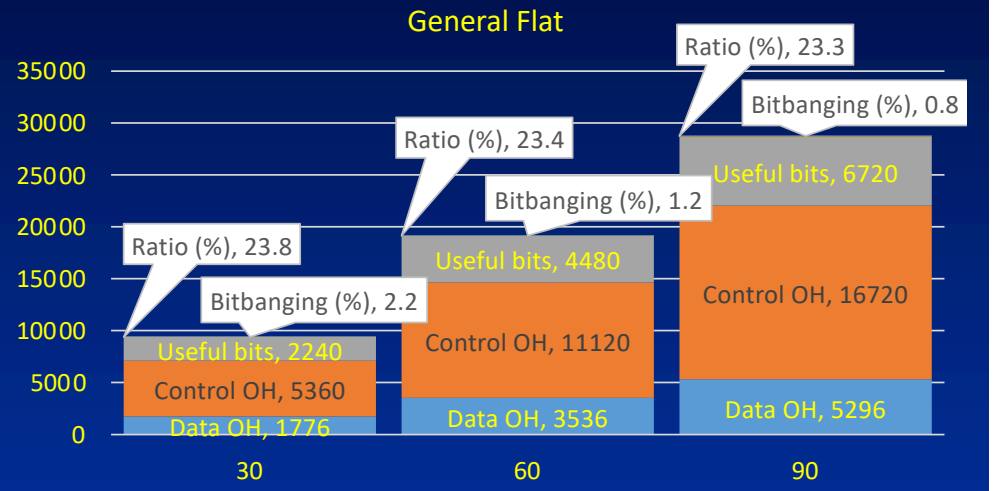


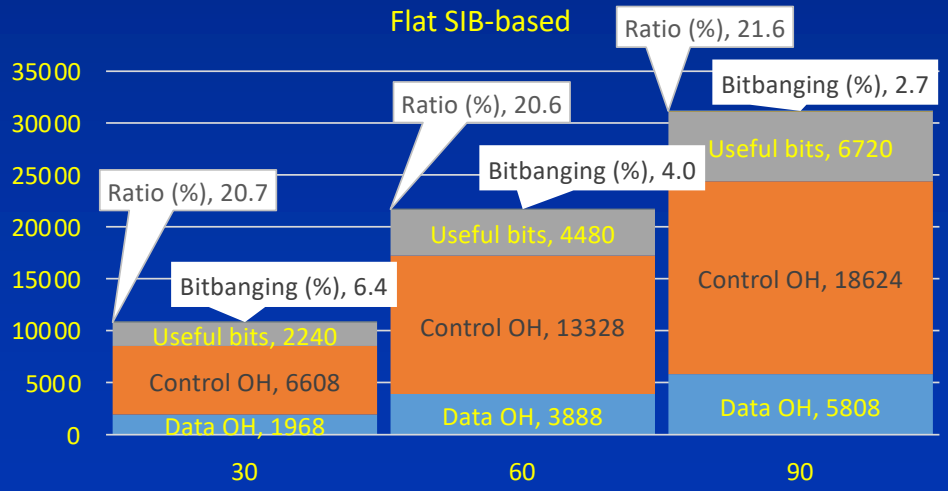
Outline

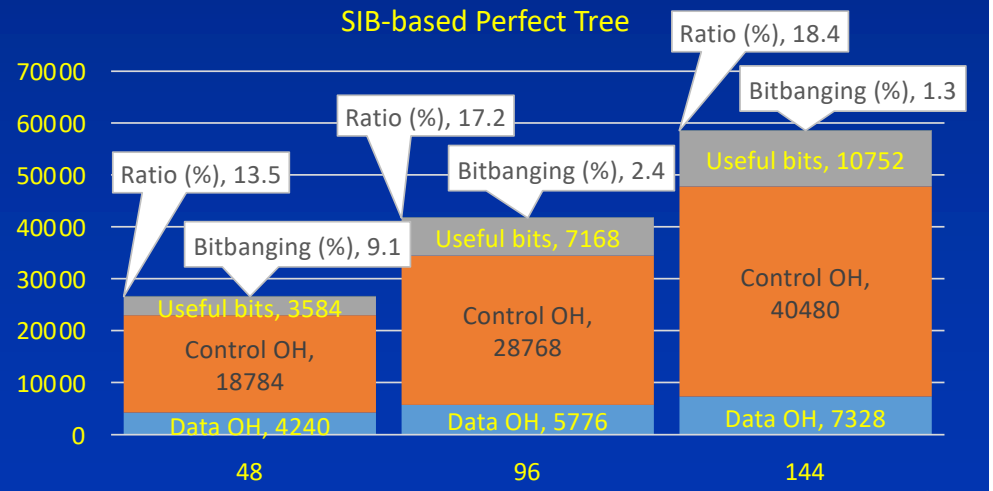
- Introduction: IEEE Std. 1149.1, 1687, 1687.1
- Prior work: Access to flat SIB-based networks
- Proposed technique: Access to general networks
- Experimental results: Three benchmarks with various number of instruments implemented on FPGA
- Conclusions

Experimental results

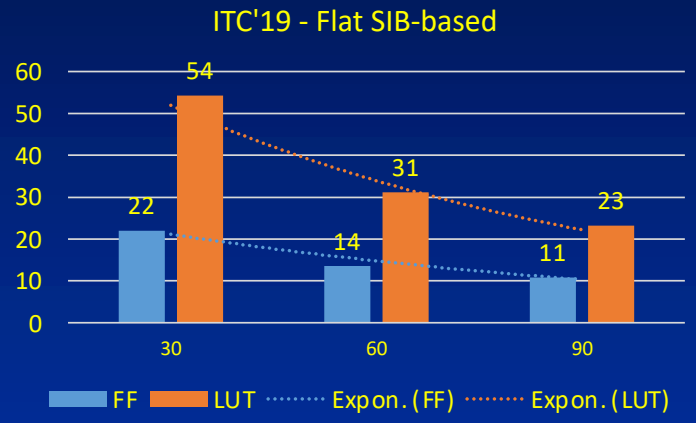


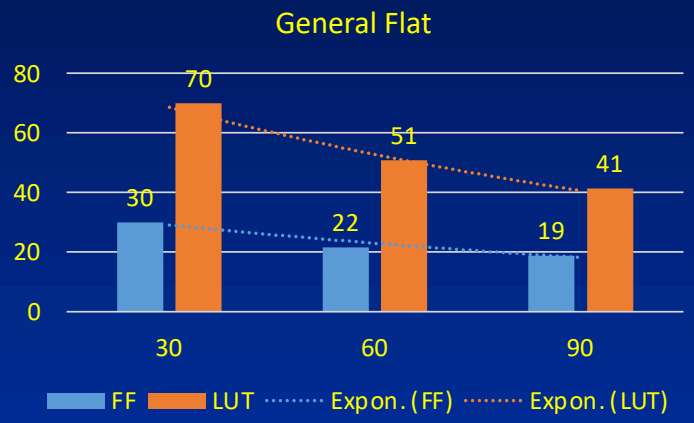


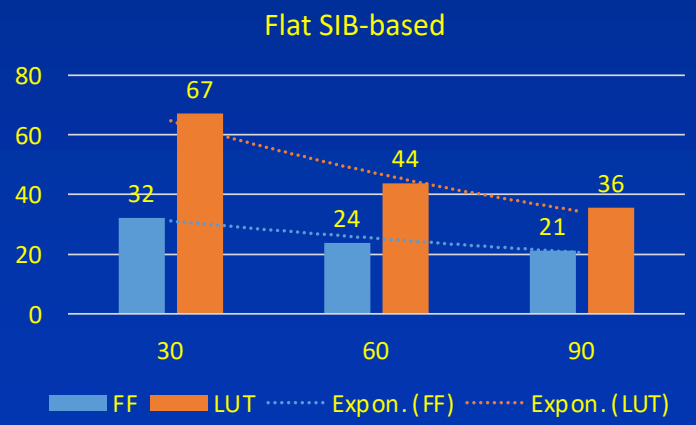


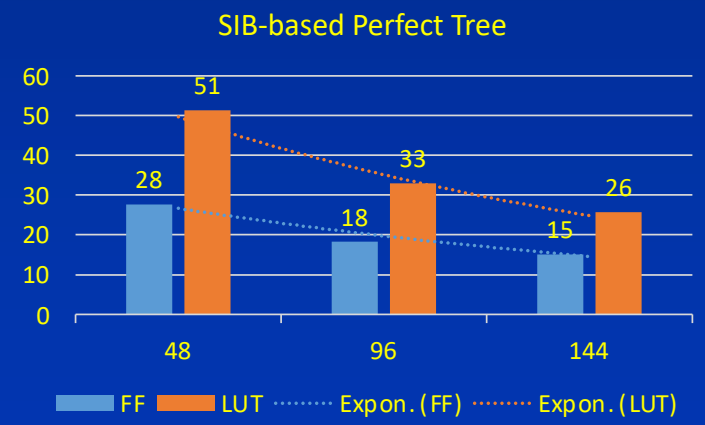


Area: Hardware module (%) of 1687 network



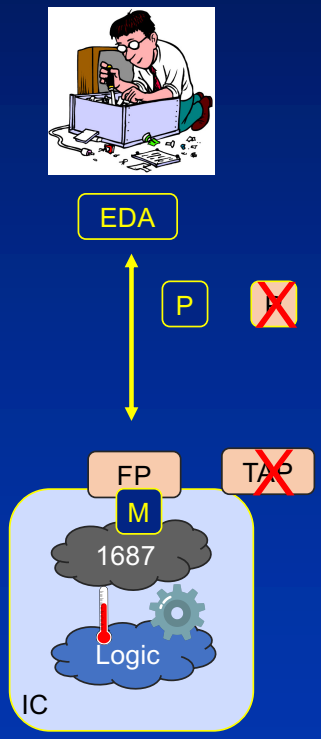






Conclusions

- Enabling access to general IEEE Std. 1687 networks using a functional port leads to a need to understand the interplay between the hardware module and the protocol, but also the EDA-tool
- We demonstrated a **hardware module** and a **protocol** that keep **key information** such that
  - Low need of transported data
  - Area need scales in a favourable way with increasing 1687 networks
- The approach is validated by FPGA implementations



International
Test Conference
November 12-14, 2019
Marriott Washington Wardman Park
Washington, D.C.

Accessing general IEEE Std. 1687 networks via functional ports

Erik Larsson, Prathamesh Murali, and Zilin Zhang

LUND
UNIVERSITY