Accessing Embedded Instruments through the Life Cycle Using P1687

Farrokh Ghani Zadegan, Gunnar Carlsson, Erik Larsson

Linköping University

ERICSSON

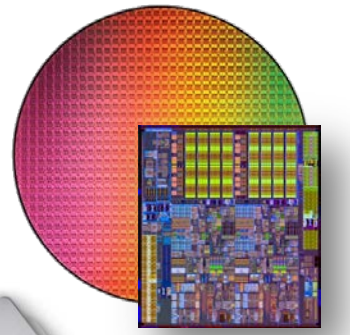
LUNDS UNIVERSITET
Lunds Tekniska Högskola

Motivation

Life Cycle of a Chip

On-chip instruments used in:

- **Chip-level:**
  - **Debugging**
  - **Characterization**
  - **Burn-in**
  - **Manufacturing test**
- **Board-level**
  - **Bring-up**
  - **Manufacturing test**
- **System-level**
  - **Monitoring & fault management**
  - **In-field test**



Motivation

- **An instrument might be used in several scenarios**
- **Example: A Memory Built-In-Self-Test might be used in**
  - **Characterization ⇨ choosing most suitable algorithms**
  - **Manufacturing test ⇨ detecting & repairing defective ICs**
  - **Burn-in ⇨ causing activity in the chip and detecting infant mortality**
  - **PCB debugging ⇨ taking a suspected memory error out of the equation**
  - **PCBA manufacturing test ⇨ detecting defective ICs**
  - **Power-on-self-test & in-field test ⇨ detecting defective ICs**

Motivation

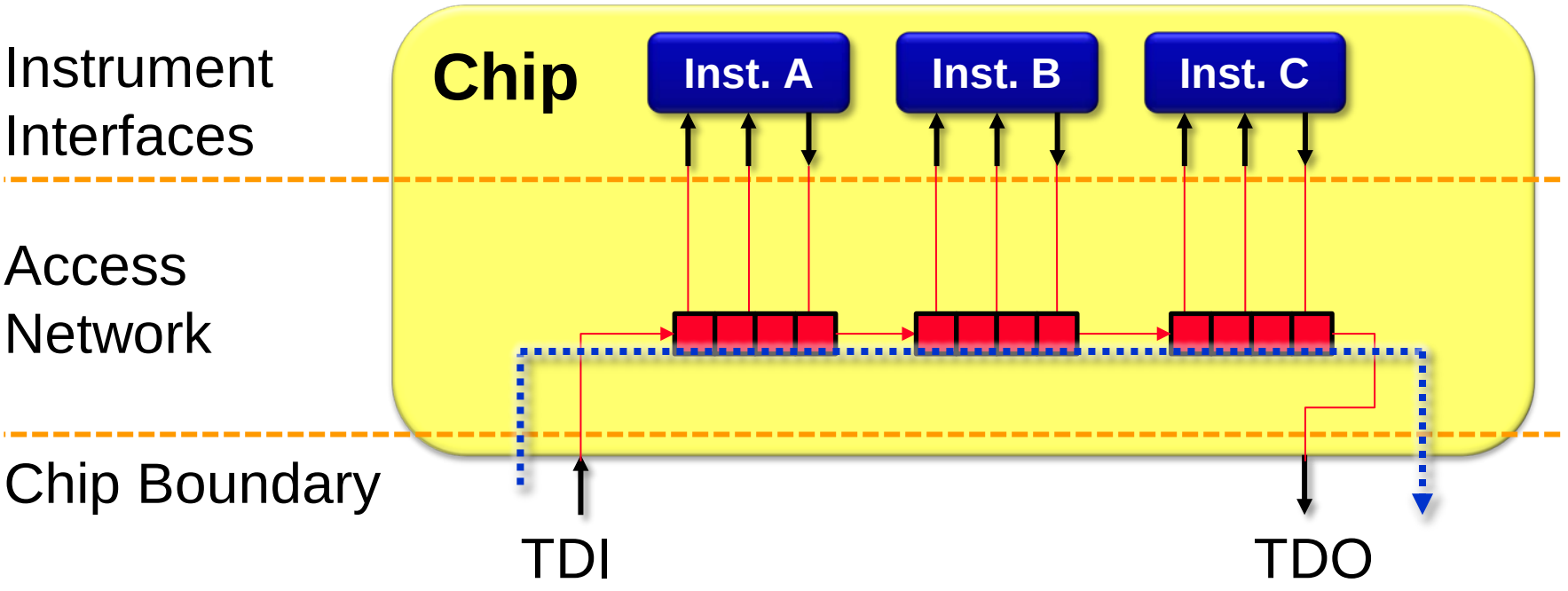
- How to access the instruments?
- IEEE P1687 standardizes the access through the JTAG TAP
  - On-chip instruments become accessible through the whole life cycle
- Many ways to build the on-chip instrument access infrastructure (network)
  - How to do it efficiently?
  - In this work we describe and compare six network design approaches w.r.t
    - Overall access time
    - Hardware overhead
    - Robustness toward new scenarios
- We believe the work is also applicable to IEEE 1149.1-2013

Outline

- **Introduction & background**
- **Design approaches**
- **Experiments**
- **Conclusion**

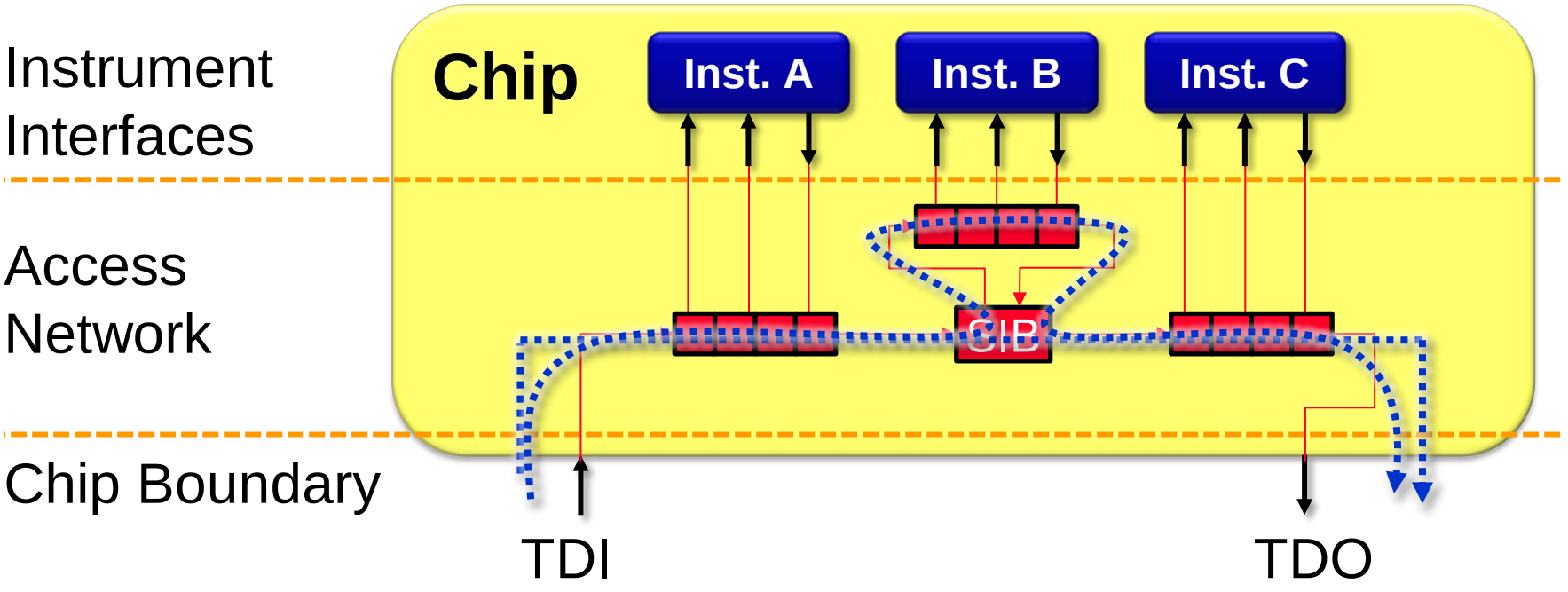
Introduction & Background

Overview of P1687



Introduction & Background

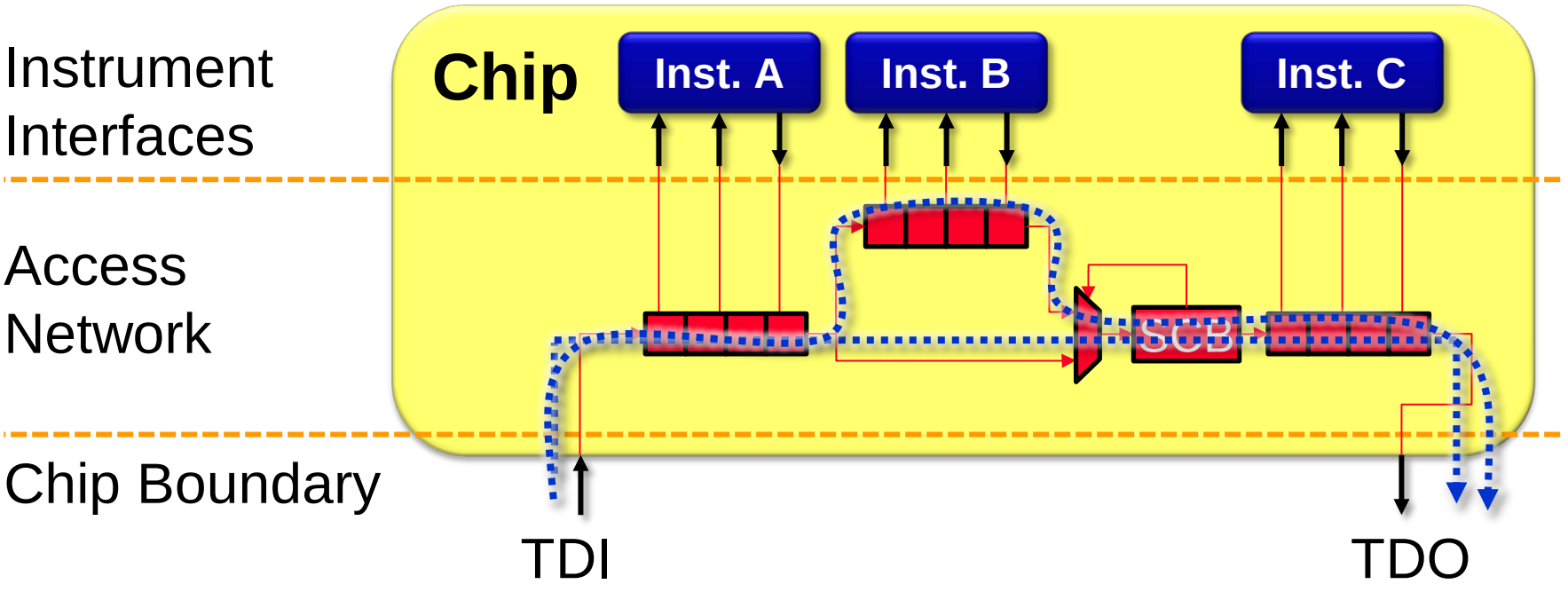
Overview of P1687



SIB: Segment Insertion Bit

Introduction & Background

Overview of P1687



SCB: ScanMux Control Bit

Introduction & Background

- Prior work on network design for P1687 [Zadegan et al, DATE'11]
  - SIB-based
  - One scenario
    - sequential access (one instrument at a time)
    - concurrent access (all instruments at the same time)
  - Overhead
    - Not affected by length of shift-registers
    - Affected by number of accesses and schedule
    - Both vary by scenario
- Networks optimized for one scenario, not optimal for others

Introduction & Background

- In this work,
  - we assume given is
    - A number of scenarios, each assigned a weight W_s by the designer
    - The number of accesses for each instrument per scenario $A_{i,s}$
    - For each scenario, sequential/concurrent access
  - we compare six design approaches w.r.t.
    - Overall access time (OAT)
    - Hardware overhead
    - Robustness against new scenarios
  - each design approaches is either
    - SIB-based, or
    - SCB-based (daisy-chained)

Outline

- Introduction & background
- Design approaches
- Experiments
- Conclusion

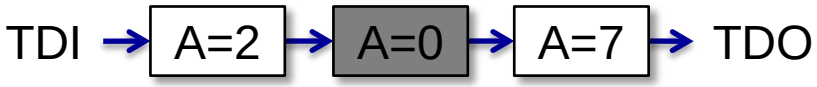
Design Approaches

- We consider the following approaches:
  - A regular scan-chain
  - Flat network
    - SIB-based
    - SCB-based (daisy-chained)
  - Hierarchical network
    - SIB-based
    - SCB-based (daisy-chained)
  - Multiple networks
    - SIB-based
    - SCB-based (daisy-chained)

Design Approaches

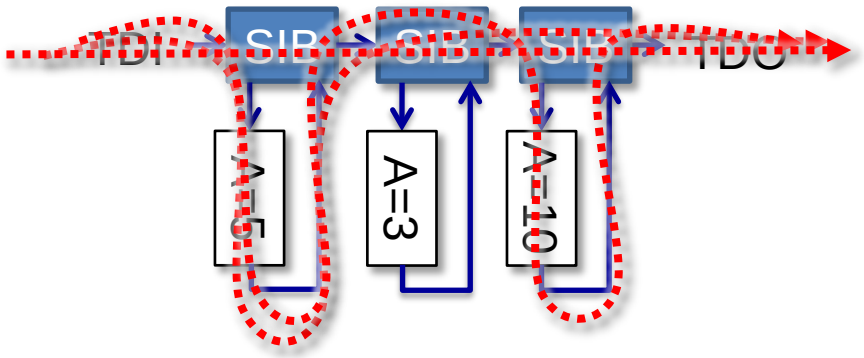
- **A regular scan-chain**

- Assume three instruments with 5, 3, and 10 accesses
- Instruments always on the scan-path
  - Overhead: dummy-bits shifted for inactive instruments



Design Approaches

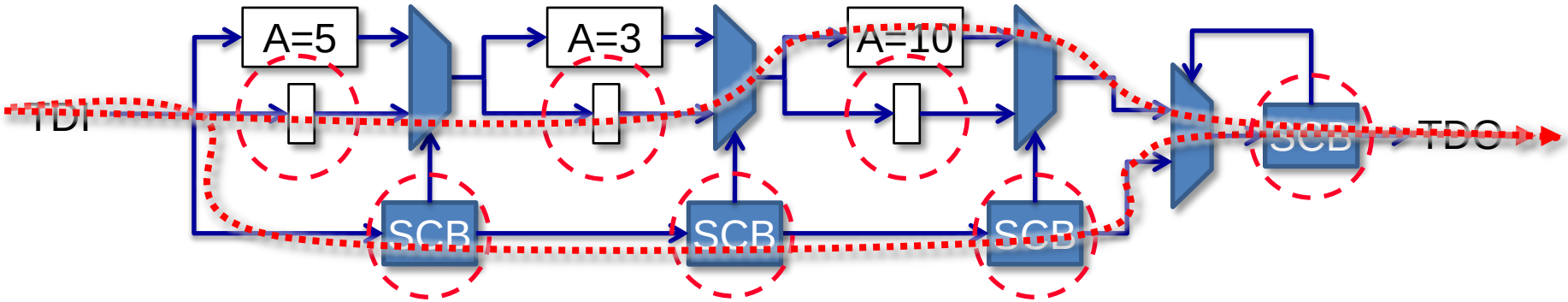
- **Flat network (SIB-based)**



- Instruments included in the scan-path when needed
  - Time overhead: SIBs on the scan-path
  - Hardware overhead: SIBs
- Straightforward to design

Design Approaches

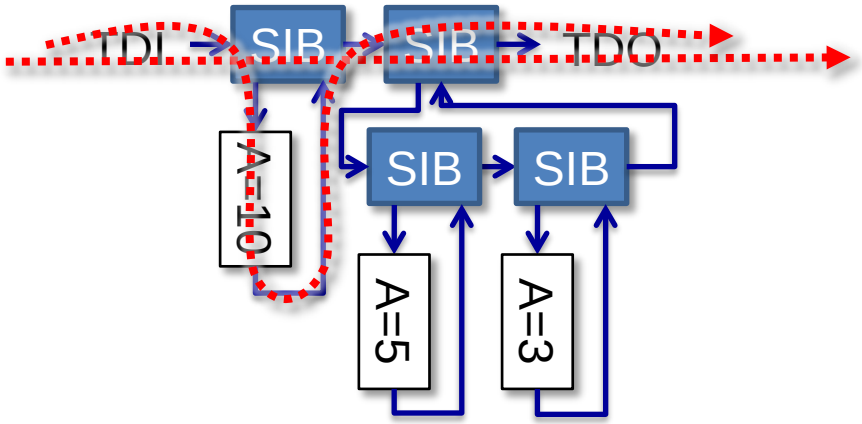
- **Flat network (daisy-chained)**



- Instruments included in the scan-path when needed
  - Time and hardware overhead:
    - SCBs on the scan-path
    - Bypass registers
- Straightforward to design

Design Approaches

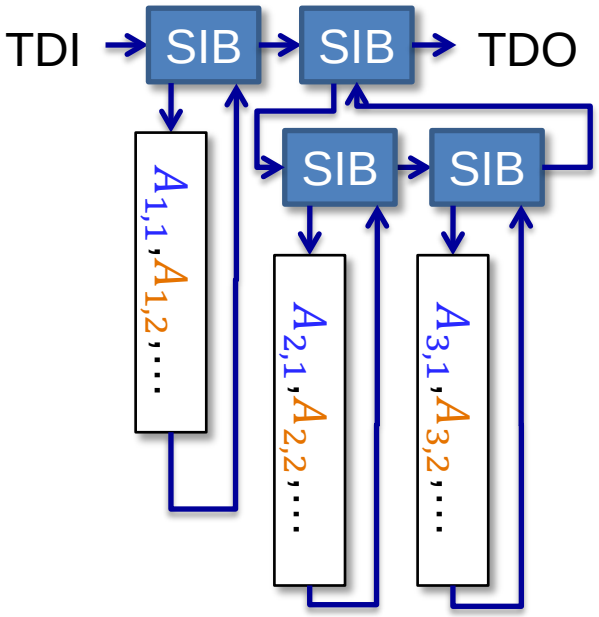
- **Hierarchical network (SIB-based)**



- Instruments included in the scan-path when needed
  - Overhead: SIBs on the scan-path

Design Approaches

- **How to design the hierarchy for multiple scenarios?**



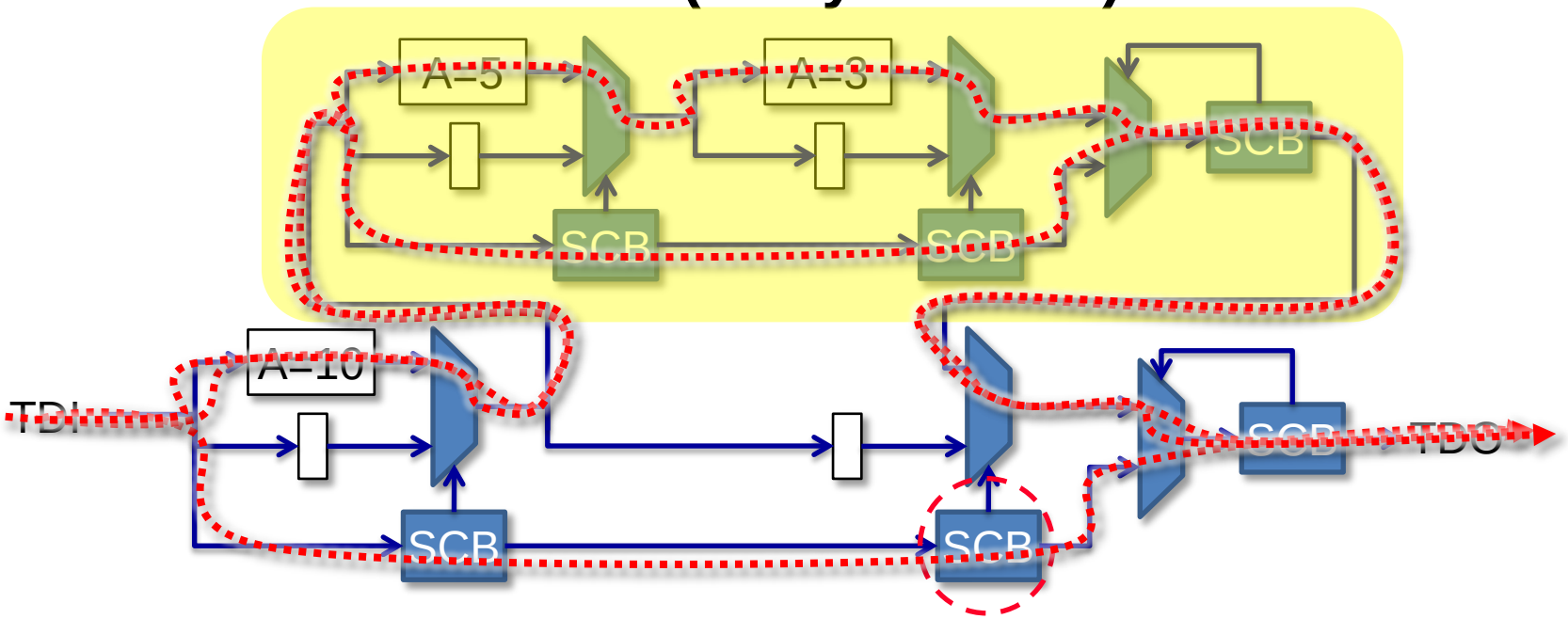
Accesses for scenario 1: $A_{1,1}$, $A_{2,1}$, $A_{3,1}$

Accesses for scenario 2: $A_{1,2}$, $A_{2,2}$, $A_{3,2}$

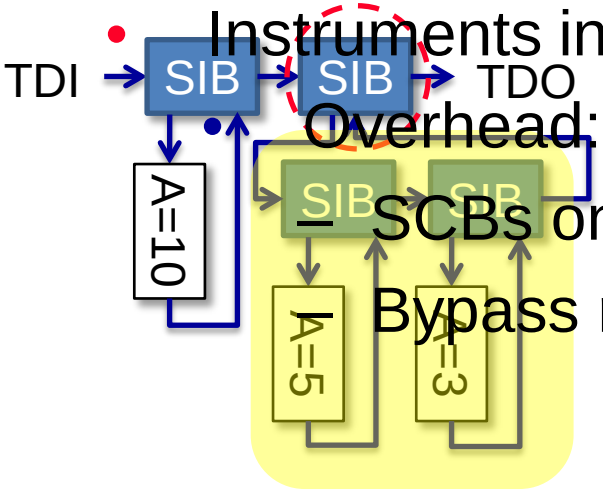
- For each instrument $A_{i,total} = \sum_{s \in S}(A_{i,s} \times W_s)$
- Use methods in [Zadegan et al, DATE'11]

Design Approaches

- **Hierarchical network (daisy-chained)**



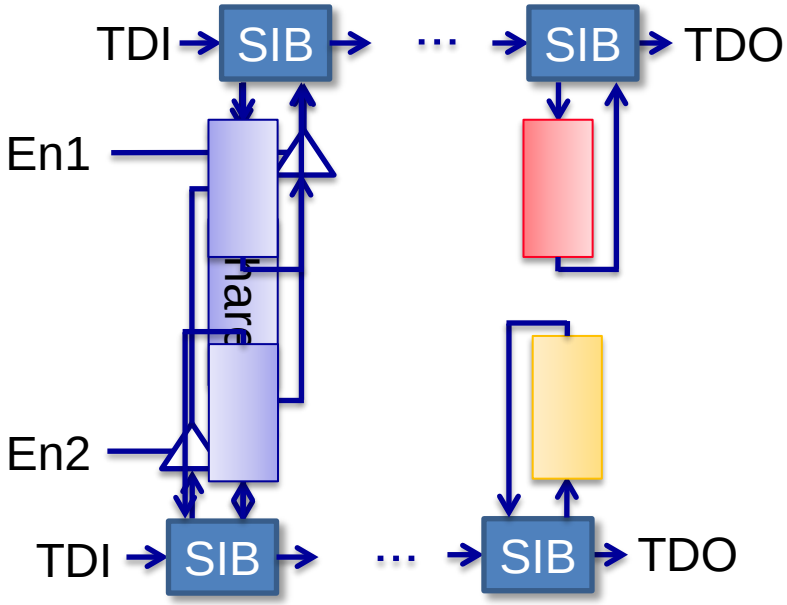
- Instruments included in the scan-path when needed



- Overhead:
  - SCBs on the scan-path
  - Bypass registers

Design Approaches

- **Multiple networks**



- One optimized network per scenario
- Two alternatives:
  - SIB-based
  - SCB-based (daisy-chained)

Outline

- Introduction & background
- Design approaches
- Experiments
- Conclusion

Experiments

- **Comparing design approaches:**
  - **Overall access time (OAT)**
    - **Weighted sum of the access time for each scenario**
  - **Hardware overhead**
  - **Robustness for new scenarios**

Experiments

Instruments and Access Scenarios

Instruments		Scenarios (weights)						
Type	Count	S1(1) Seq.	S2(100) Con.	S3(1) Con.	S4(1) Seq.	S5(100) Con.	S6 Seq.	S7 Con.
1	20	100	10	10	10	10	10	10
2	20	10000	0	0	0	0	0	0
3	10	100	10	10	10	10	10	10
4	40	100000	10	10	100000	100	10	10
5	10	10	100	10	10	10	100000	100000

Experiments

Overall Access Time (for known scenarios)

Approach	Weighted overall access time (x10^6)					Sum (x10^6)
	S1(1) Seq.	S2(100) Con.	S3(1) Con.	S4(1) Seq.	S5(100) Con.	
Regular scan-chain	8427	20.3	0.02	8021	20.1	16489
Flat Network	525	4.7	0.02	500	10.1	1040
Flat daisy-chained	525	4.5	0.02	500	9.7	1040
Hierarchical network	152	4.1	0.02	143	9.7	309
Hierarchical daisy-chain	152	3.9	0.02	143	9.3	308
Multiple networks	151	3.8	0.02	143	9.5	307
Multiple daisy-chains	151	3.7	0.02	143	9.1	307

Experiments

Hardware Overhead

Approach	Flip-flops	Muxes	Buffers
Regular scan-chain	0	0	0
Flat Network	200	100	0
Flat daisy-chained	302	101	0
Hierarchical network	286	143	0
Hierarchical daisy-chain	517	187	0
Multiple networks	1172	586	400
Multiple daisy-chains	1940	677	400

Experiments

Overall Access Time (for new scenarios)

Approach	S6 (seq.)		S7 (con.)	
	OAT ($x10^6$)	Potential improvement	OAT ($x10^6$)	Potential improvement
Regular scan-chain	2006	0%	2005	0%
Flat Network	125	0%	31	0%
Flat daisy-chained	125	0%	30	0%
Hierarchical network	40	17.56%	23	3.84%
Hierarchical daisy-chain	40	17.55%	23	4.00%
Multiple networks	36	11.78%	22	0%
Multiple daisy-chains	36	11.79%	21	0%

Conclusion

- P1687 to access the on-chip instruments
- Instruments used in multiple scenarios
  - Different schedules
  - Different number of accesses
- Network optimized for one scenario, not optimal for others
- We compared six approaches, regarding
  - Overall access time
  - Hardware overhead
  - Robustness against new scenarios

Conclusion

Observations from the comparison

- Use of multiple networks results in the least OAT
  - at the cost of extra hardware
- Daisy-chains perform better for concurrent access
  - Slightly higher hardware overhead compared with their SIB-based counterparts
- A single hierarchical network resulted in low OAT
  - with a relatively low hardware overhead

Accessing Embedded Instruments through the Life Cycle Using P1687

Farrokh Ghani Zadegan, Gunnar Carlsson, Erik Larsson

Linköping University

ERICSSON

LUNDS UNIVERSITET
Lunds Tekniska Högskola