



Extensible Framework for Debug, Diagnosis and Test

Jason Doege

2006 IEEE 7th International Workshop on
Microprocessor Test and Verification

December 5, 2006

Agenda

- Say what I'm going to say
- Say what I have to say
- Say what I said

But really...

- Quick review of current techniques (and their deficiencies)
- Description of this technique
- ???
- Profit!

Assessment Criteria

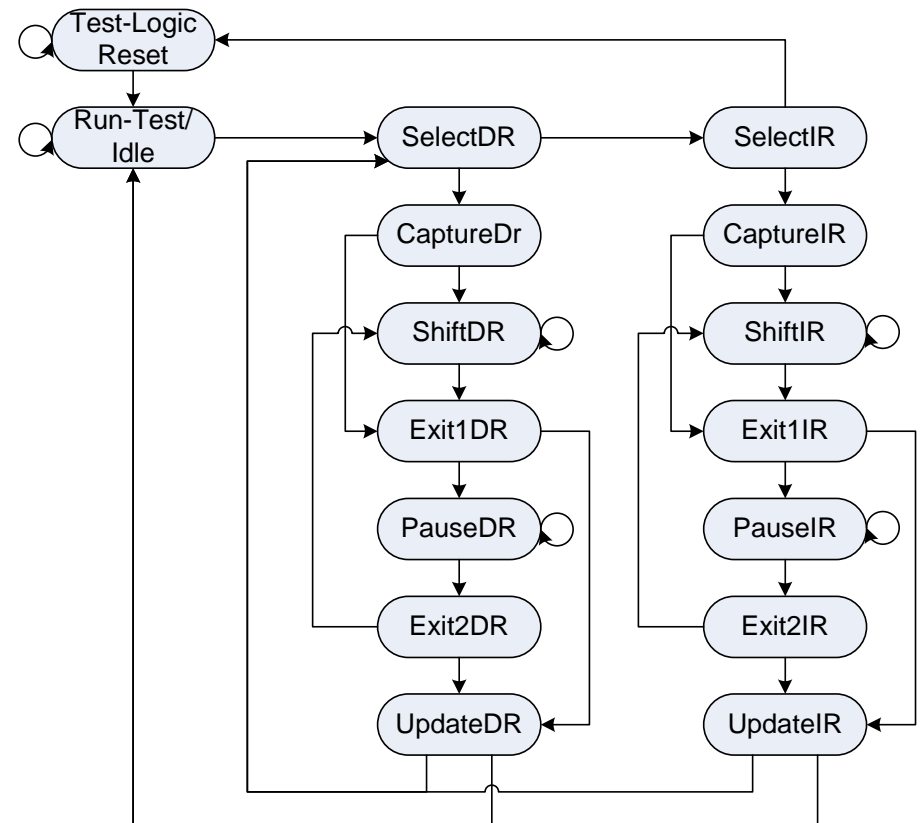
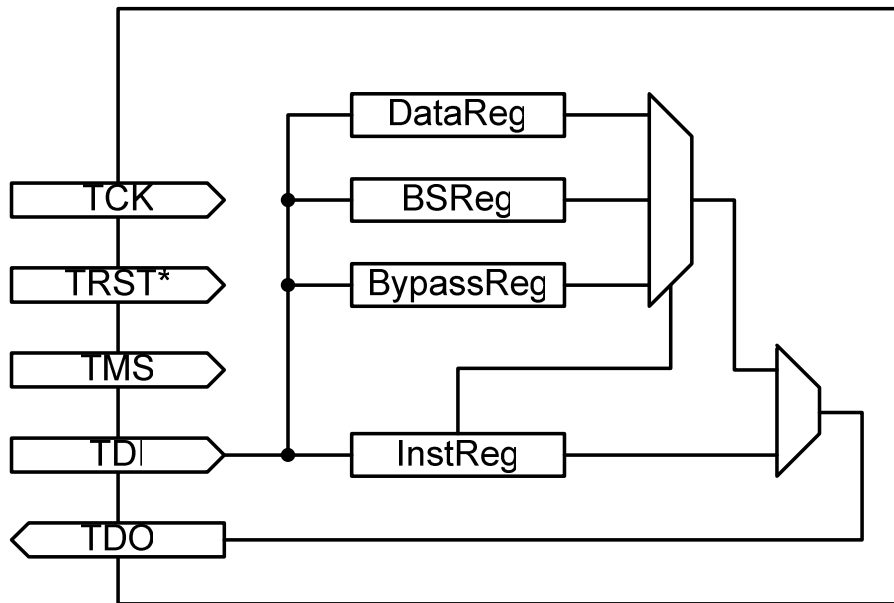
- ▶ Fixed Cost (engineering overhead)
 - Engineering
 - Tooling
- ▶ Variable Cost (manufacturing overhead)
 - Area
 - Route
 - Performance
- ▶ Scalability
 - System integration
 - Re-use

Adhoc

- ▶ Non-standard

Really, what more is there to say?

JTAG



Basic JTAG

- ▶ Uses Shift-DR, Capture-DR, Update-DR and Run-Test/Idle to control activities.
- ▶ Useful for setting test mode states, initiating events, sampling results
- ▶ Good system integration
- ▶ Not very useful for data transfer (state machine gets in the way)

“Clever” JTAG

- ▶ Use of “compliance enable” pins to create ad-hoc modes
- ▶ Uses other states of the JTAG state machine in creative ways to solve various problems
- ▶ Often non-compliant
- ▶ Usually non-standard so poor EDA support
- ▶ Standard solutions solve tightly constrained problems

Just About Any JTAG

- ▶ Chip level solution
- ▶ Intrinsically not scalable

This proposal

- ▶ Adhoc
- ▶ Basic JTAG
- ▶ “Clever” JTAG

This proposal

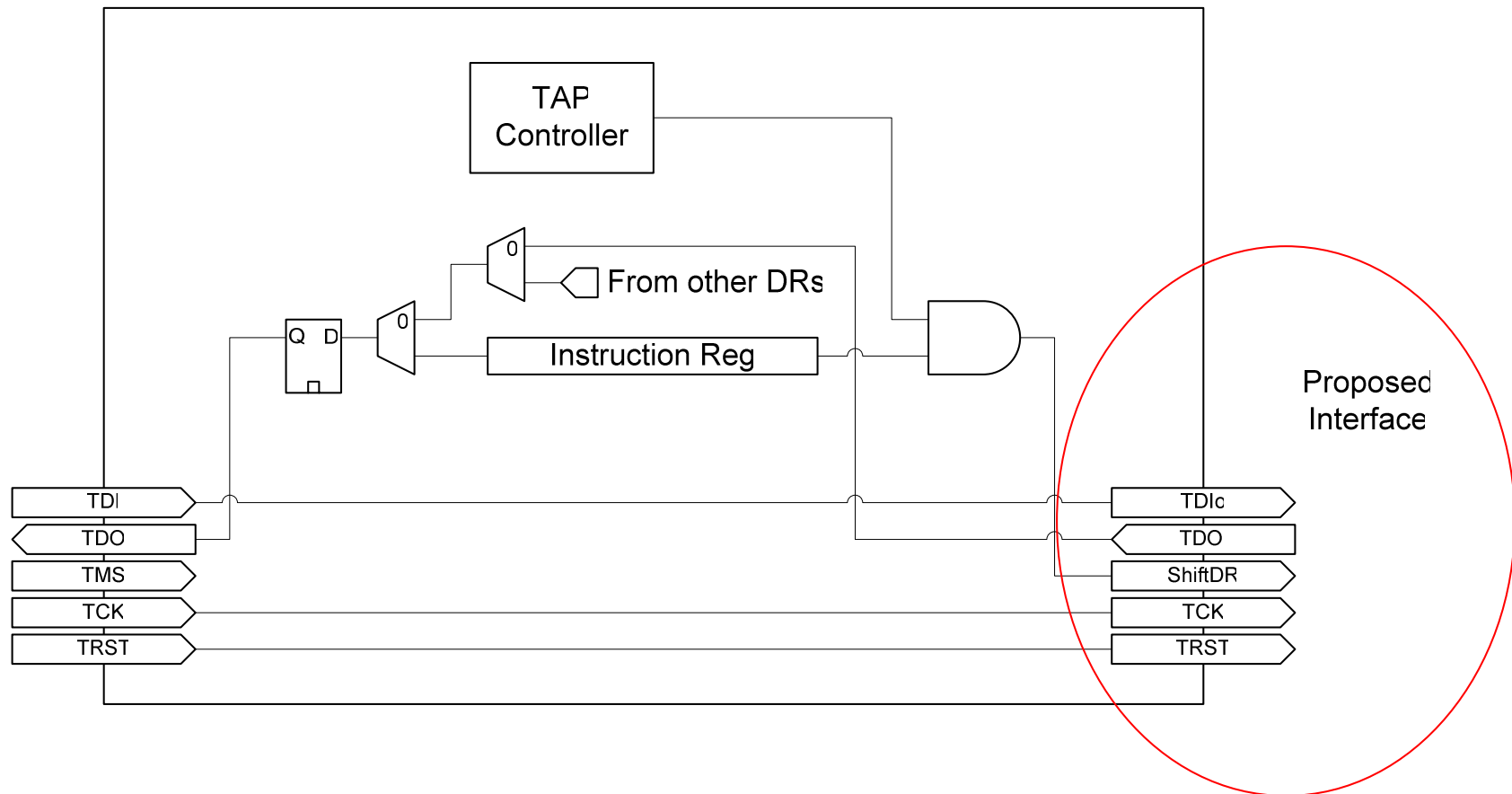
- ▶ Adhoc
- ▶ **Basic JTAG** with some compliant cleverness
- ▶ “Clever” JTAG

Define a simple internal port

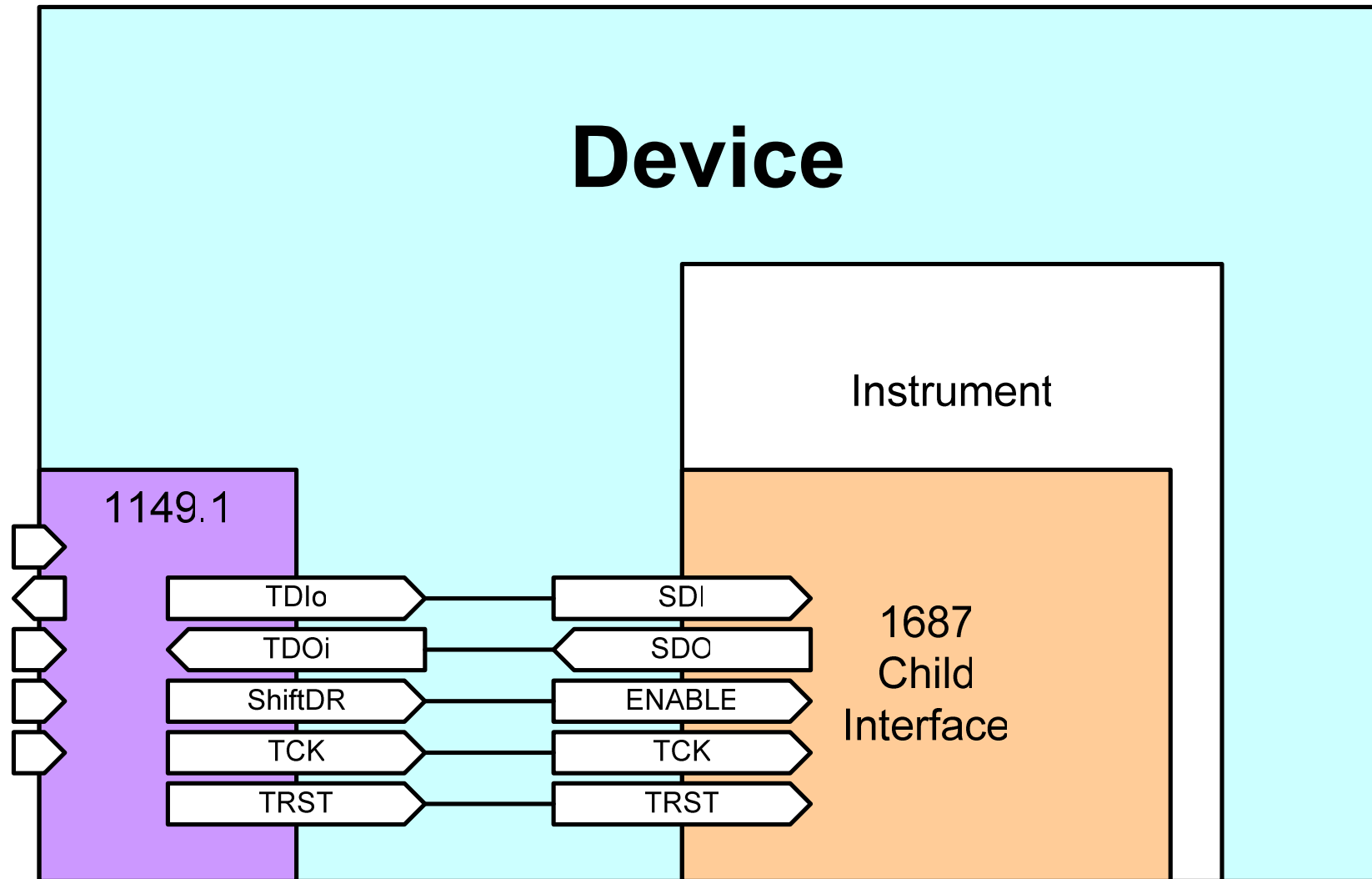
- ▶ 5 or 6 signals
 - TCK
 - TRST
 - Enable (derived from ShiftDR)
 - TDI
 - TDO
 - UpdateDR?
- ▶ Make child and parent ports look the same
 - This is what differentiates this proposal from conventional JTAG techniques.

A (few) Picture(s) speak a thousand words...

1149.1 Tap Controller with Proposed Interface



A (few) Picture(s) speak a thousand words...



Enabling Hierarchy

- ▶ Define a simple protocol
- ▶ The proposed port enables a serial data stream so... serial protocol
- ▶ Every “message” composed of a simple header followed by the message payload all delivered via “ShiftDR”

Define a Simple Protocol

Each shift operation is a “message”

Each message will be composed of a header and payload.

The header will set the Child interface’ instruction which will select the data path and the payload will be passed down the selected data path to whatever will consume it (data registers, scan chains, etc.)

Using this structure, an instrument may have one or more Parent interfaces, thereby enabling an extremely simple and relatively efficient hierarchical access mechanism.

Payload destined for top level instrument directly attached to TAP Controller



Payload destined for a 2nd level instrument connected to the parent interface of a top level instrument



An implementation

Grey box is the device

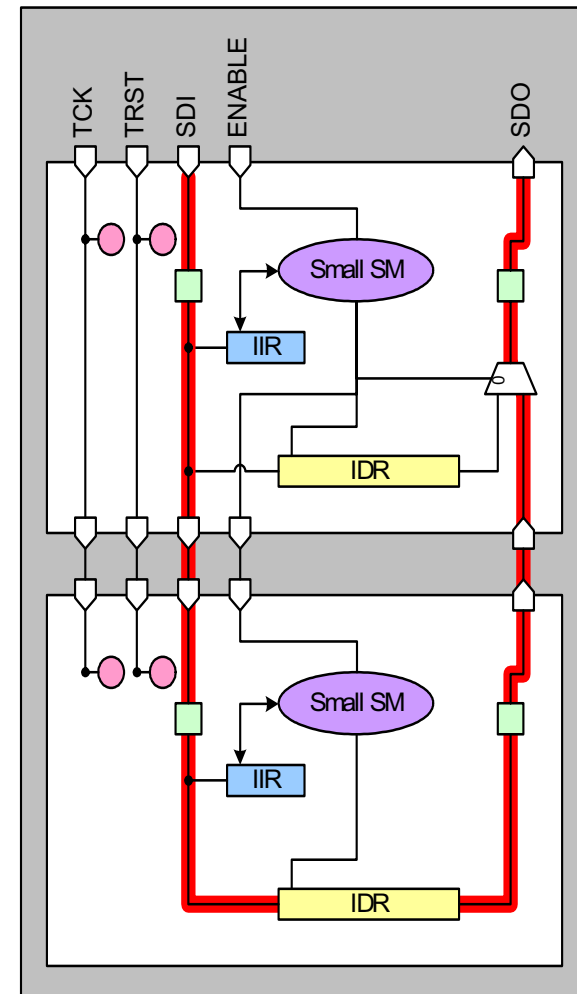
White boxes are instrument

Red line is the datapath selected by a message

Pink circles indicate that TCK and TRST are connected through the interface as needed.

Note:

Parent and child interfaces have a one to one, pass-through relationship and no special handling is needed to cascade the interfaces. **Each child is simply a selected DR of the parent.**



Method for use

- ▶ Select the JTAG instruction that enables the network
- ▶ For each message
 - Construct message (header + payload)
 - Go to ShiftDR
 - Shift message
 - Leave ShiftDR
 - Update???

What does this support?

Features

- ▶ Dynamic triggers
- ▶ Static mode bits
- ▶ Static status bits
- ▶ High volume data in
- ▶ High volume data out

With instrument support:

- ▶ Asynchronous event capture

Instrument Types

- ▶ MBIST
- ▶ Memory loading
- ▶ IEEE 1500
- ▶ Embedded JTAG controllers
- ▶ Embedded Logic Analyzers
- ▶ Full Scan and Compressed Scan
- ▶ Logic BIST
- ▶ Etc.

Complex instrument types are supported through the use of serial to parallel converters that decompose the serial data stream to parallel control and data.

Benefits

- ▶ **Scalability**
 - Easy intergration because of defined hierarchy support
 - Simple interconnect and protocol for easy automation
- ▶ **Fixed Costs**
 - High re-use because of hierarchical support
 - Ideally broad knowledge due to standardization
- ▶ **Variable Costs**
 - Minimal route requirements
 - Very low gate count

More work needed

- ▶ Propose these as an interface and protocol for the IEEE P1687 standards body

- ▶ Resolve synchronous vs. asynchronous JTAG control issue

- ▶ Satisfy two more requirements
 - Parent-directed multidrop communication
 - Support for asynchronous signals

Comments? Questions?

This presentation will be made available on the IEEE P1687 Website
<http://grouper.ieee.org/groups/1687/>