SYNOPSYS®
Silicon to Software™

3D-IC Test Standard IEEE P1838

Overview

Adam Cron, Principal Engineer
ITC 2016
Special Session 2

Participants

- **Chair:** *Saman Adham,* TSMC
- **Discussant:** *Sandeep Goel,* TSMC

- **Overview:** *Adam Cron,* Synopsys
- **Serial Control Mechanism:** *Al Crouch, SiliconAid*
- **Die Wrapper Register:** *Teresa McLaurin,* ARM
- **Flexible Parallel Port:** *Adam Cron,* Synopsys
- **Description Language:** *Sandeep Bhatia,* Google

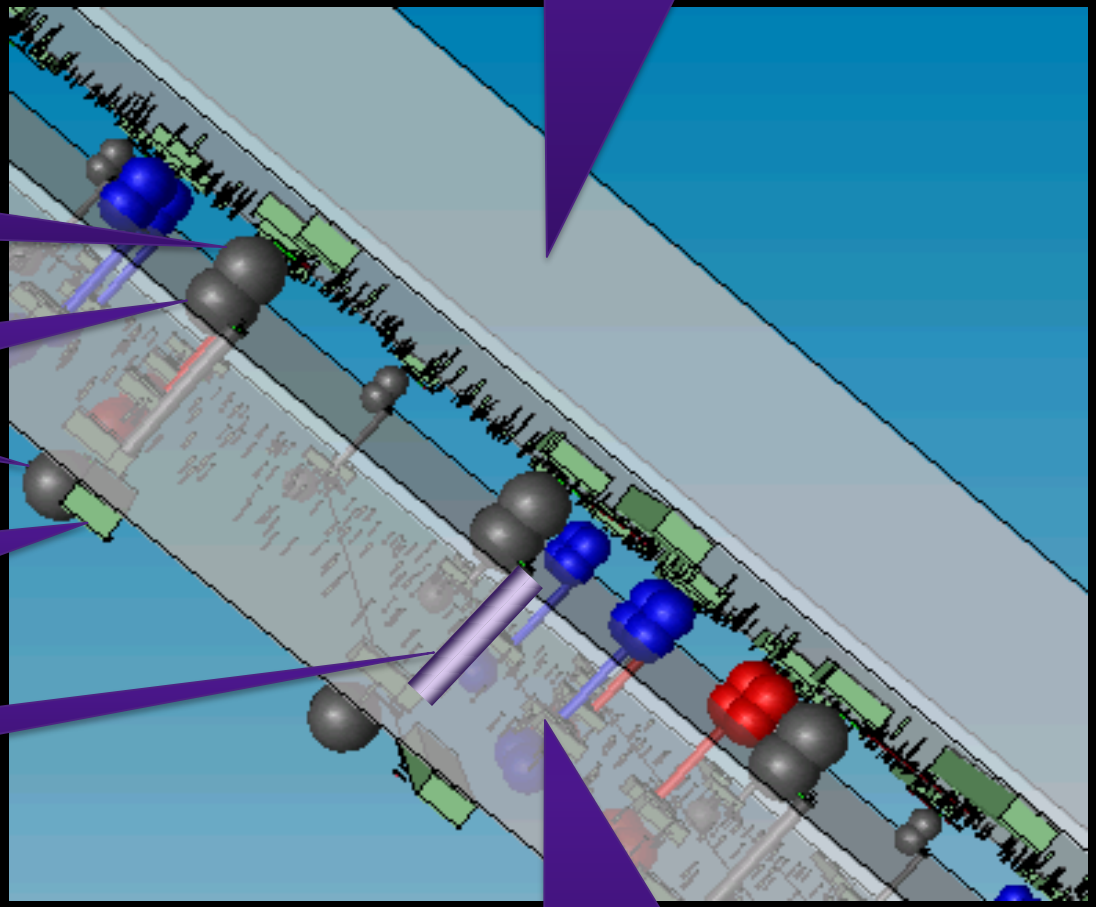
Chair, IEEE Std P1838

Erik Jan Marinissen
Principal Scientist, imec

Goals

Extending test access to a stack of die

3D-IC Stack


Top Die
Top Die Bumps
Bottom Die Bumps
Bottom Die Metal Layers
Bottom Die TSVs
Bottom Die

Package Possibilities

- 2.5D, 3D, 5.5D: many topologies
- Digital + memory is most prevalent
- Heterogeneous technology nodes
- Higher bandwidth
- Smaller footprint
- Lower power

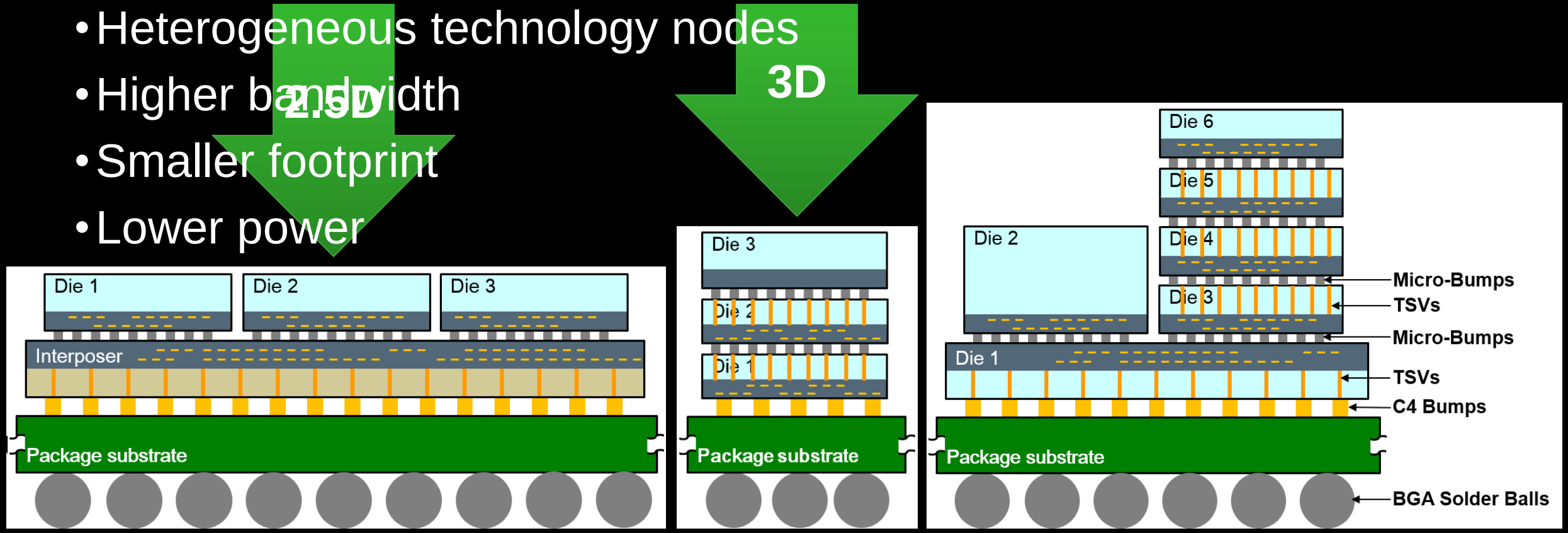
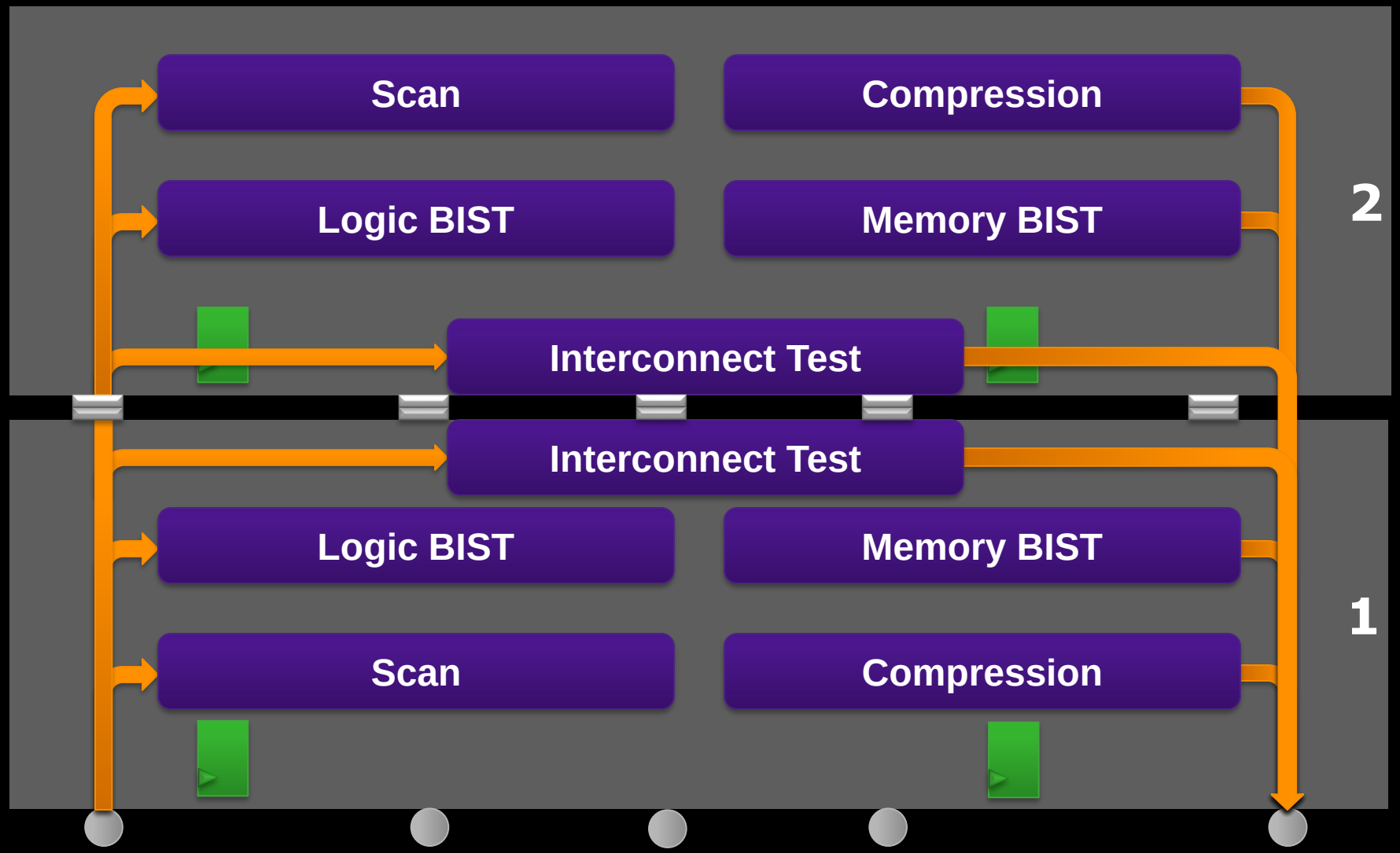


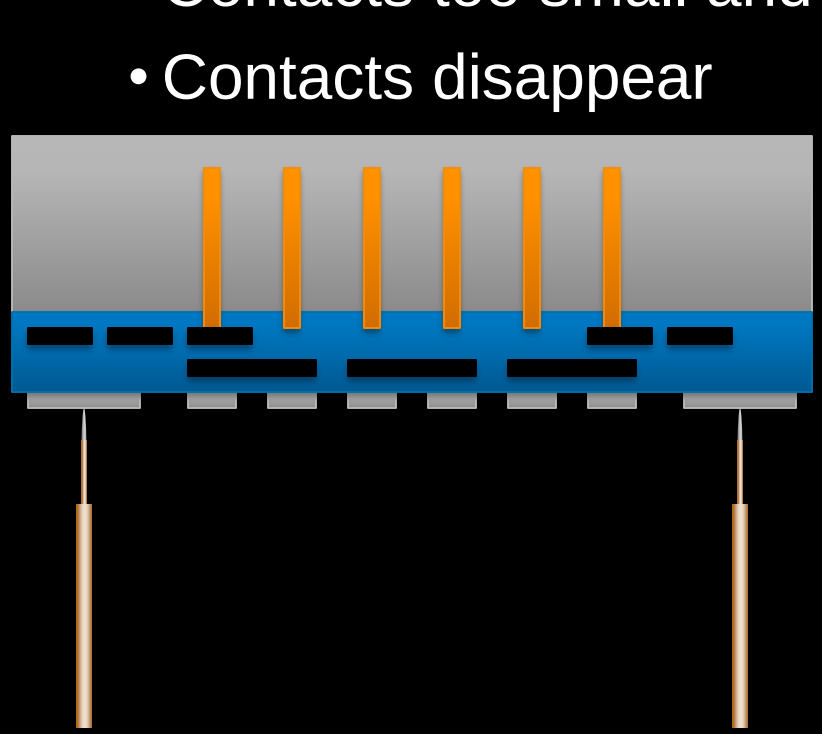
Image source: Erik Jan Marinissen - IEEE European Test Symposium, 2016

Test Context

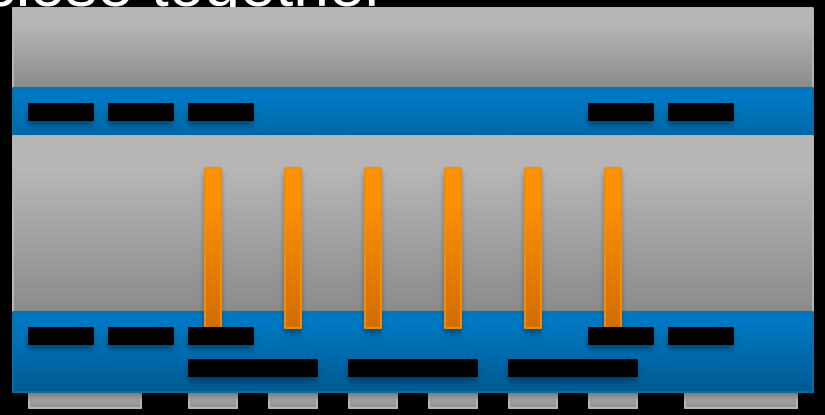


Access Issues and Solutions

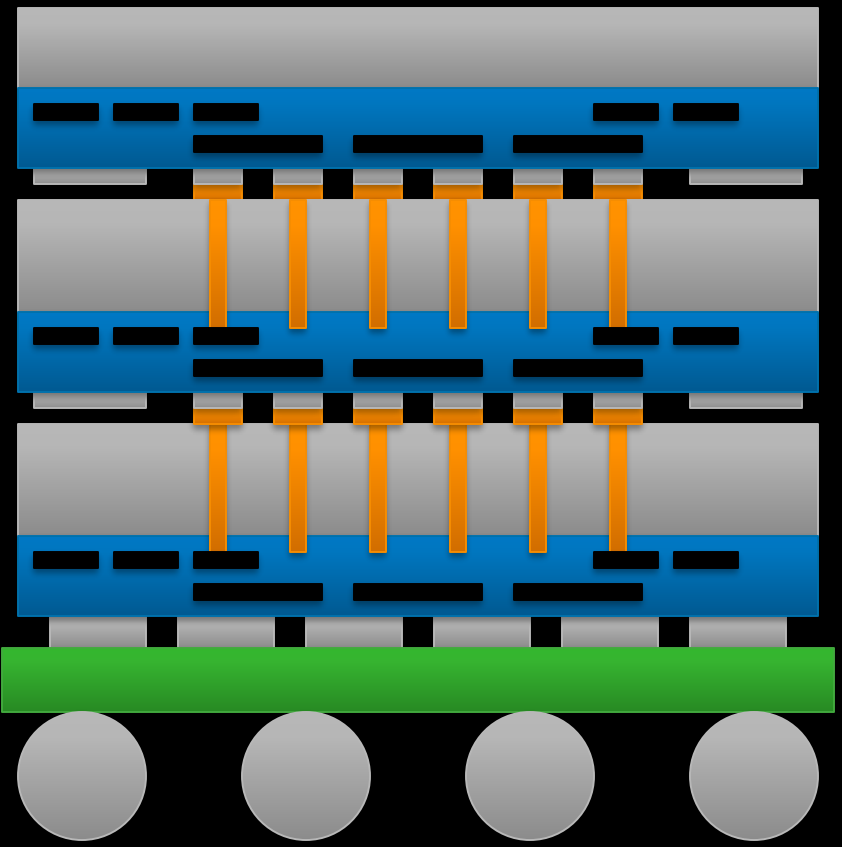
- TSVs not accessible
- Contacts too small and close together
- Contacts disappear



Die-Level Pre-Bond Test

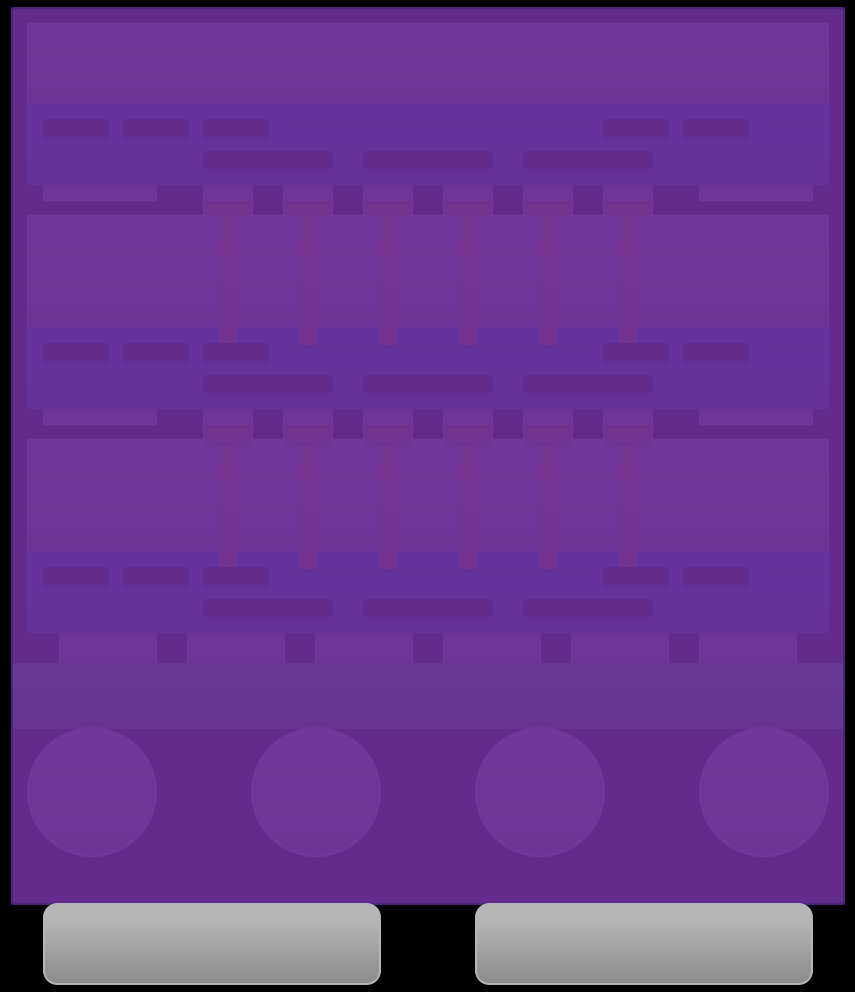


Partial Stack Mid-Bond Test

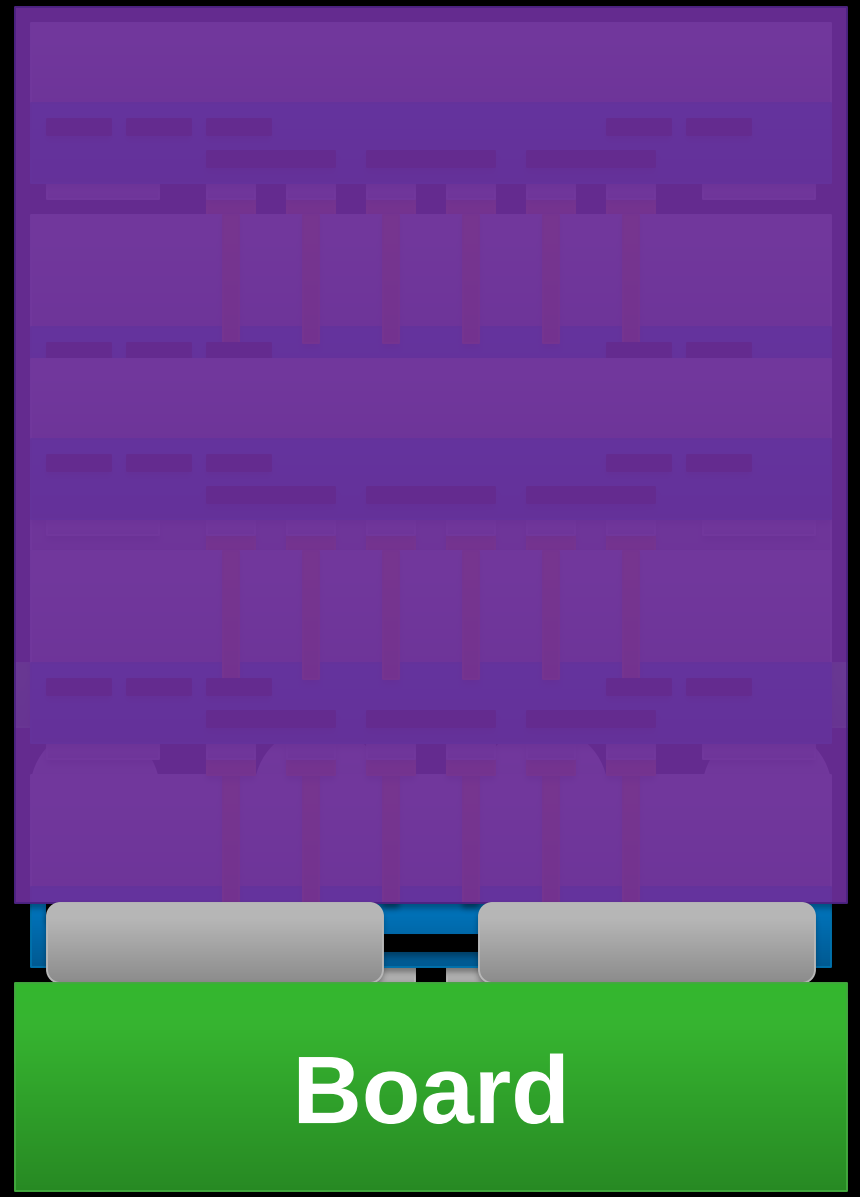


Complete Stack Post-Bond Test

Access Issues and Solutions



Final Package Test



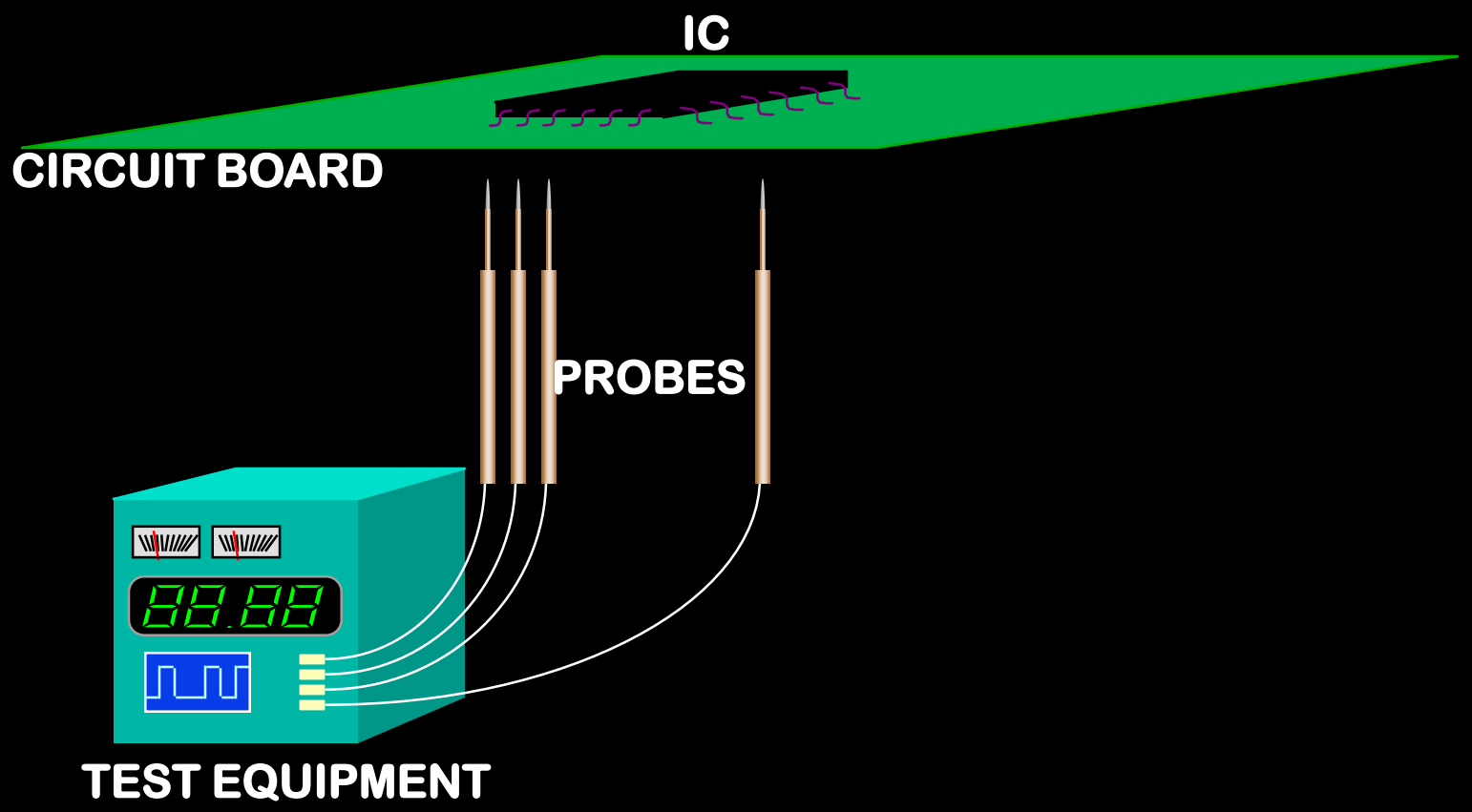
Complete System Post-Bond Test

Stacked Test

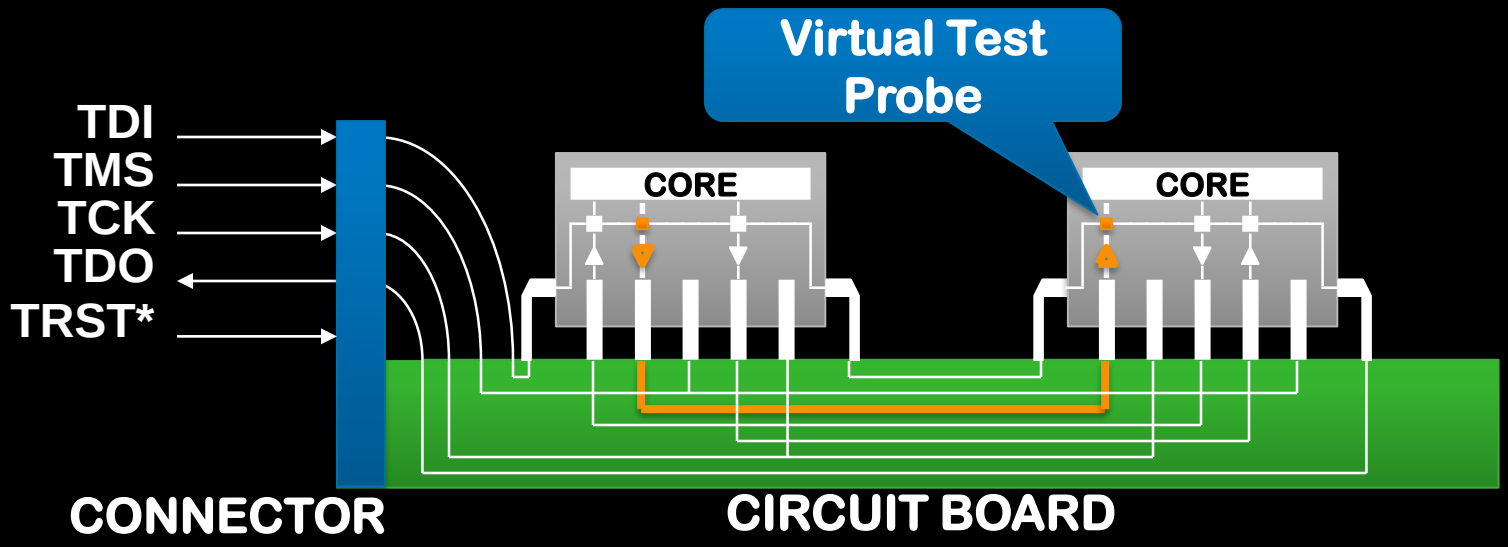
IEEE Std 1149.1

A little pertinent review

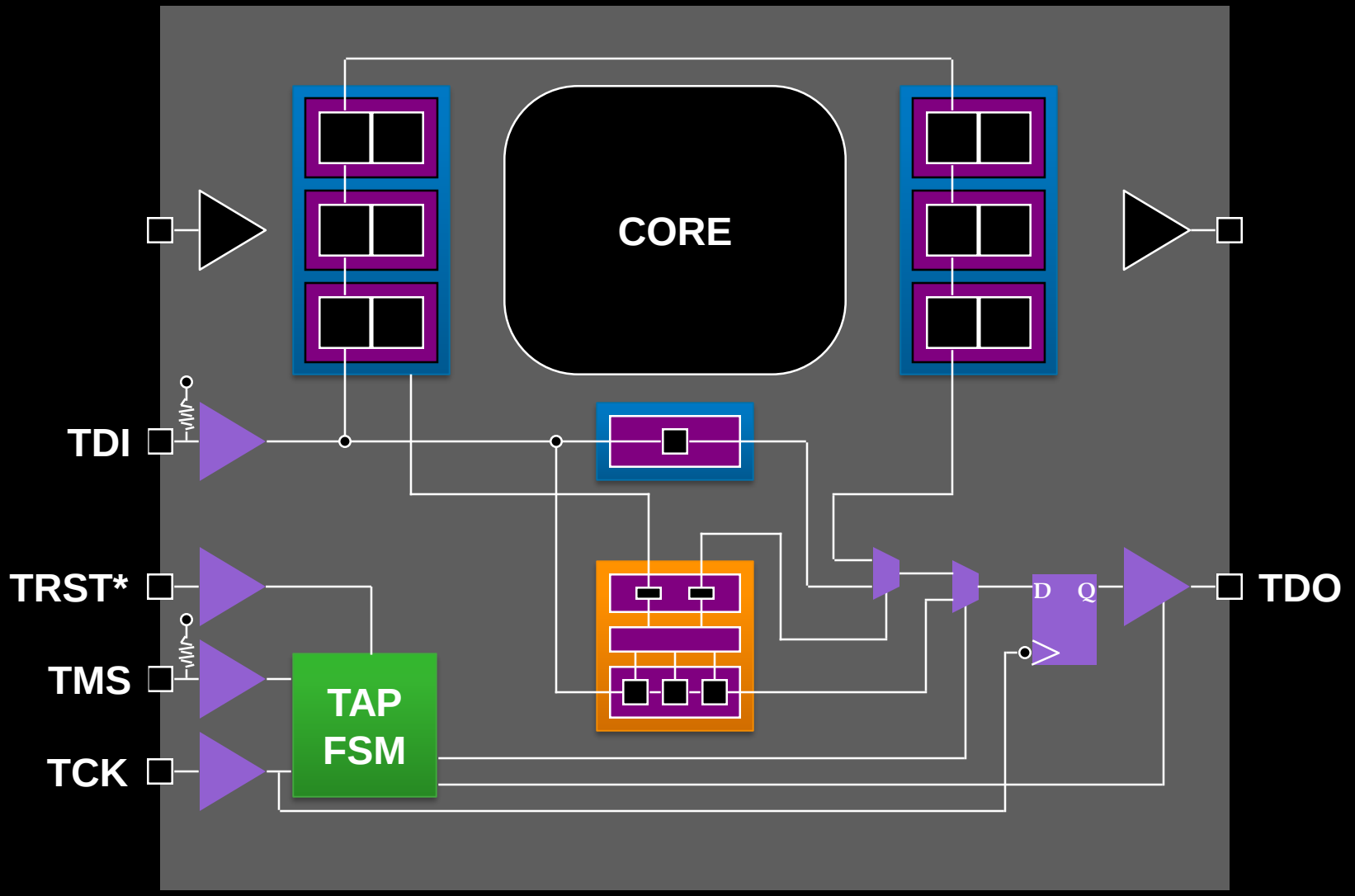
In-Circuit Test System



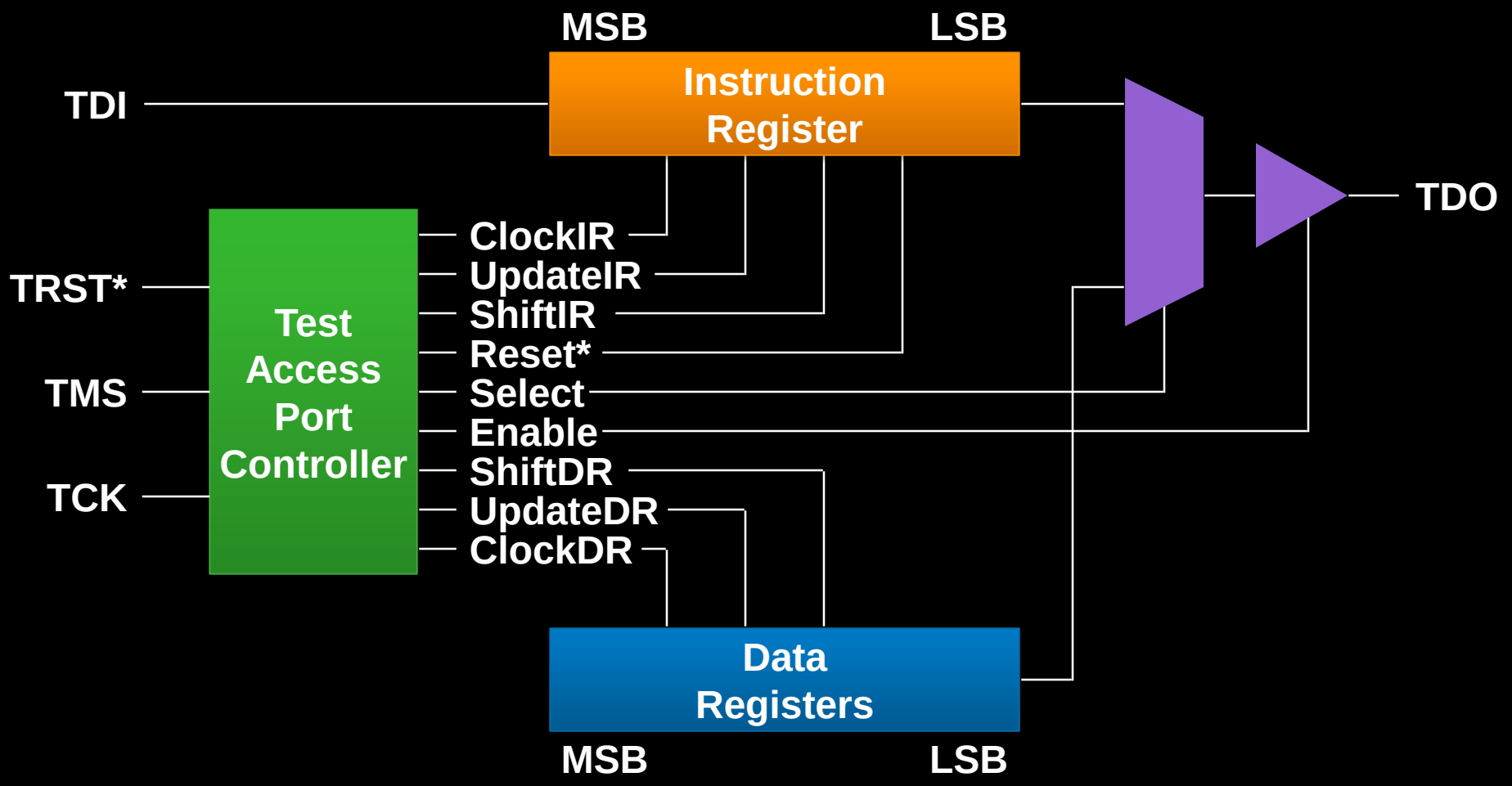
Boundary-Scan Architecture



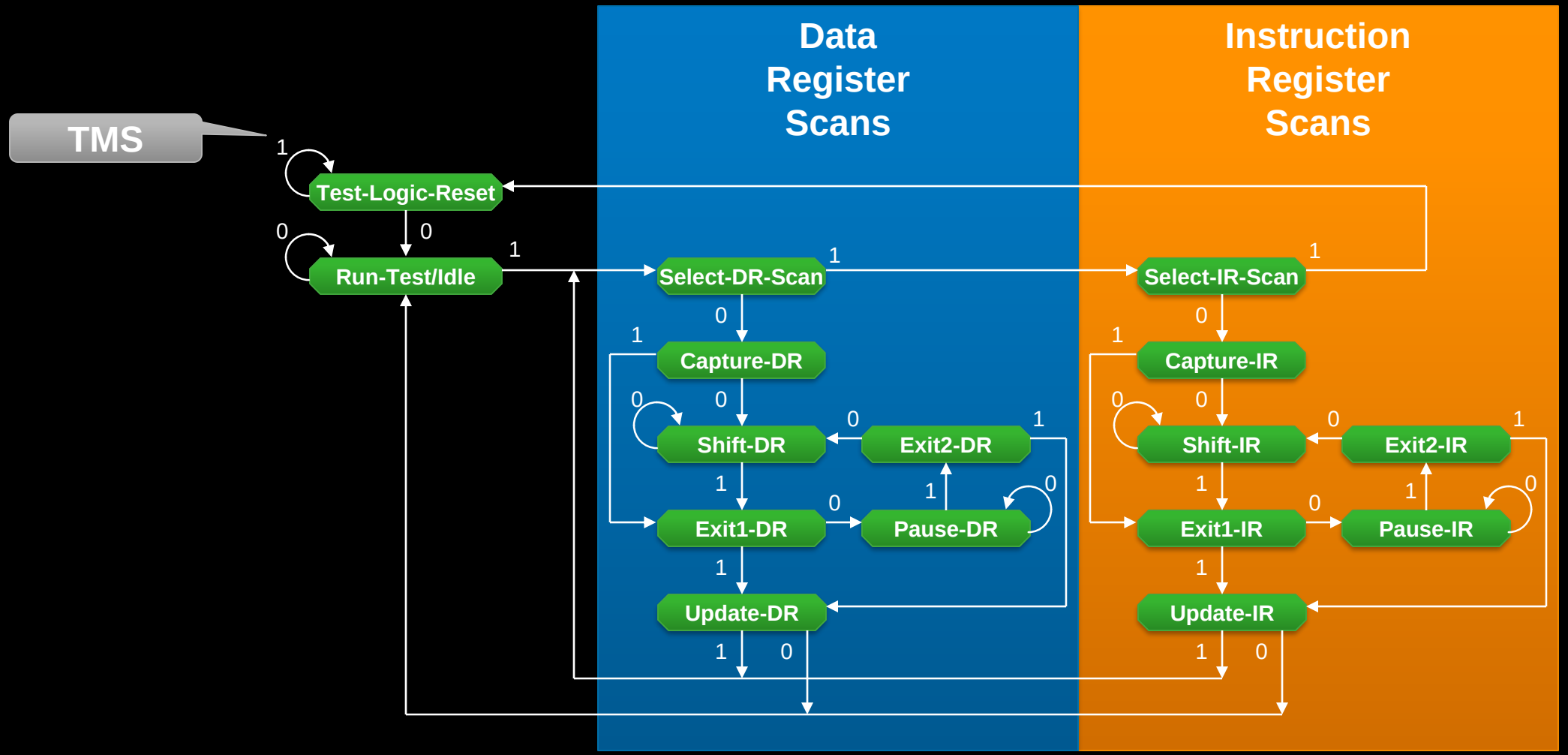
Basic IEEE Std 1149.1 Architecture



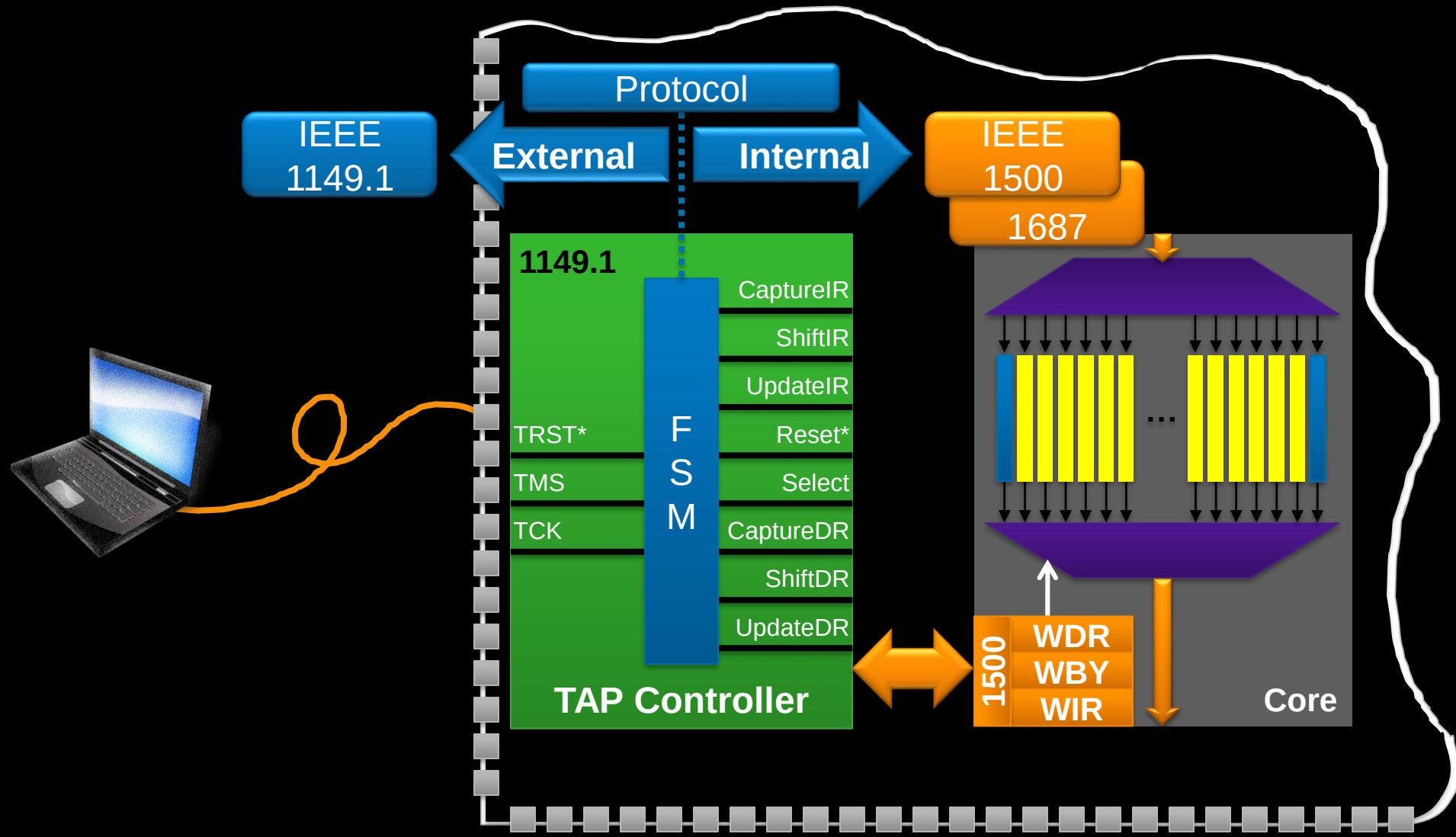
TAP Output Control Connections



TAP Controller State Machine



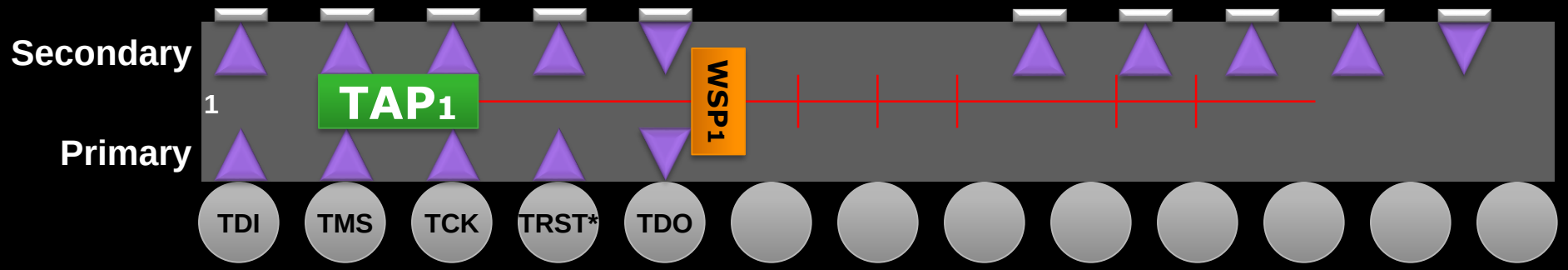
Single Chip View at the System Level



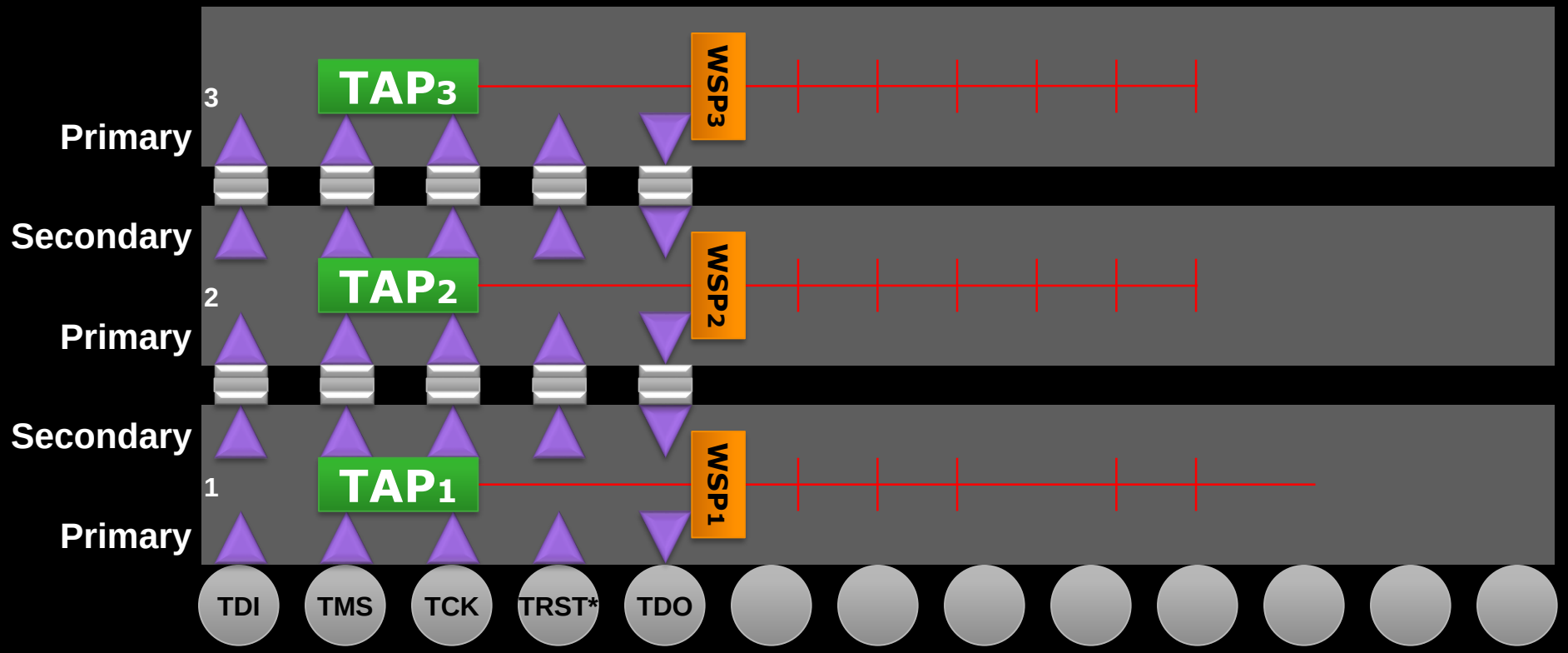
IEEE Std P1838

Technical overview

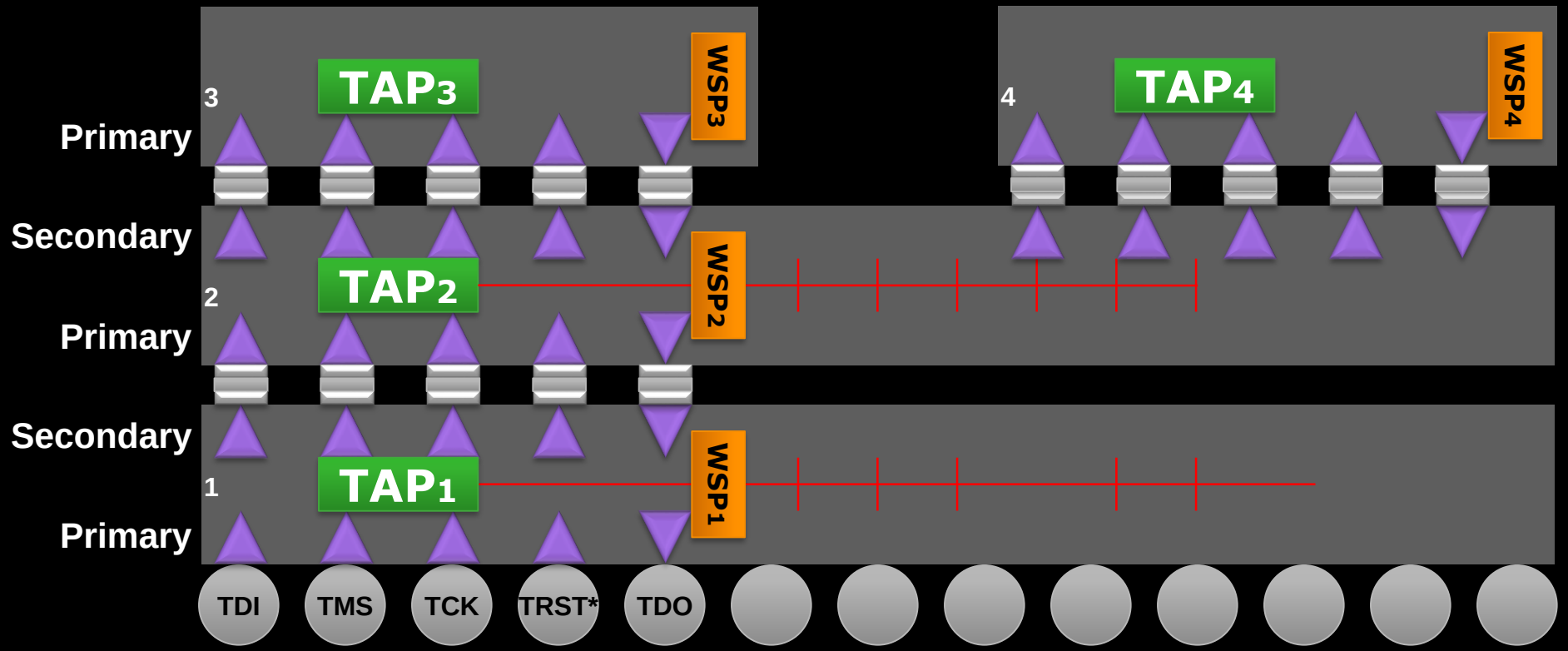
Current Thinking – Each Die Has a TAP



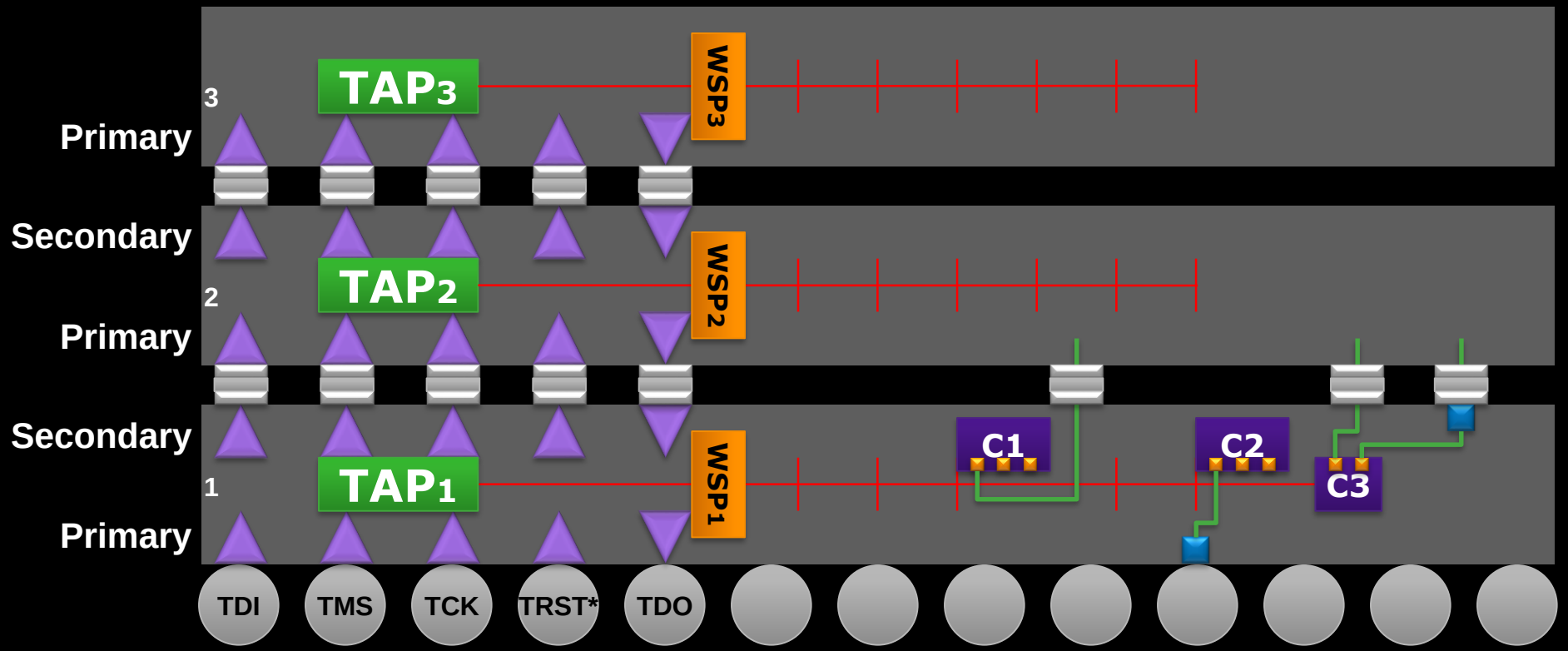
Current Thinking – Serial Access Up and Down



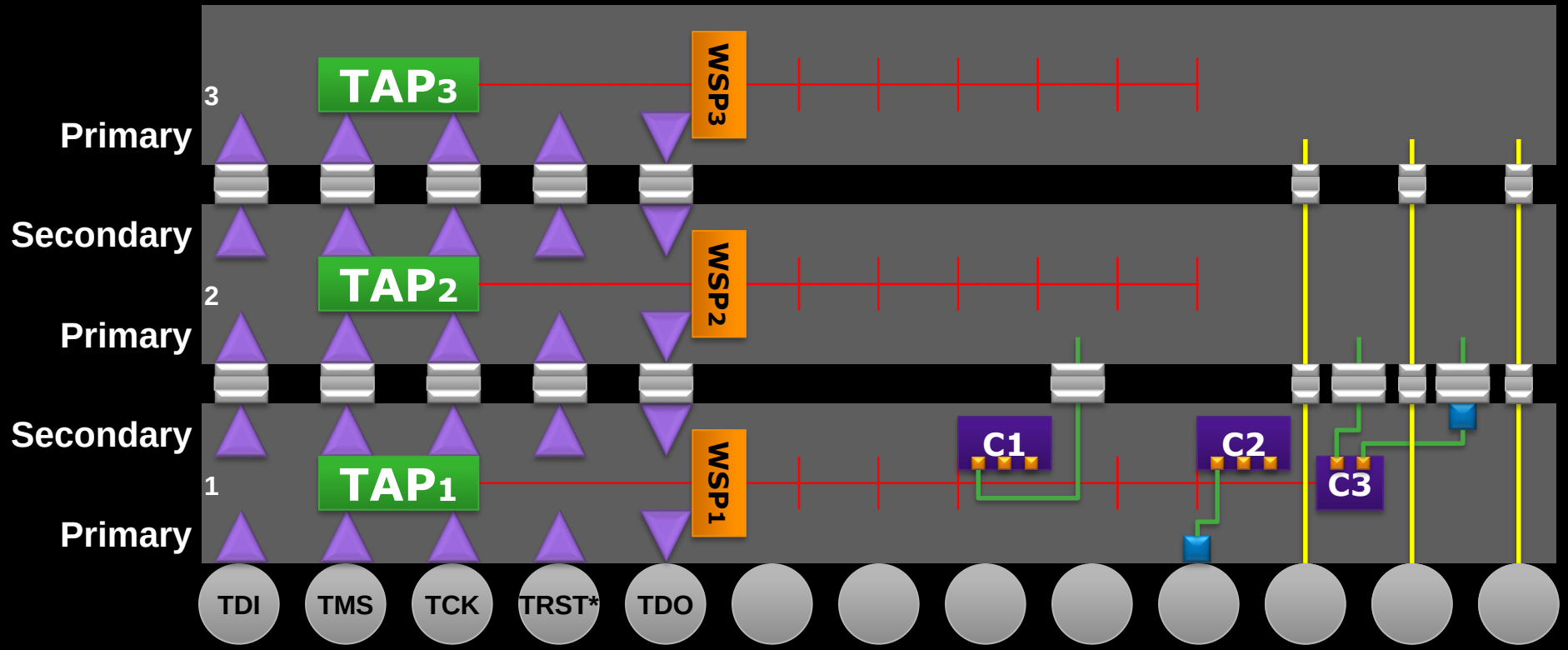
Current Thinking – Multi-Tower Support



Current Thinking – Serial Access



Current Thinking – Parallel Access



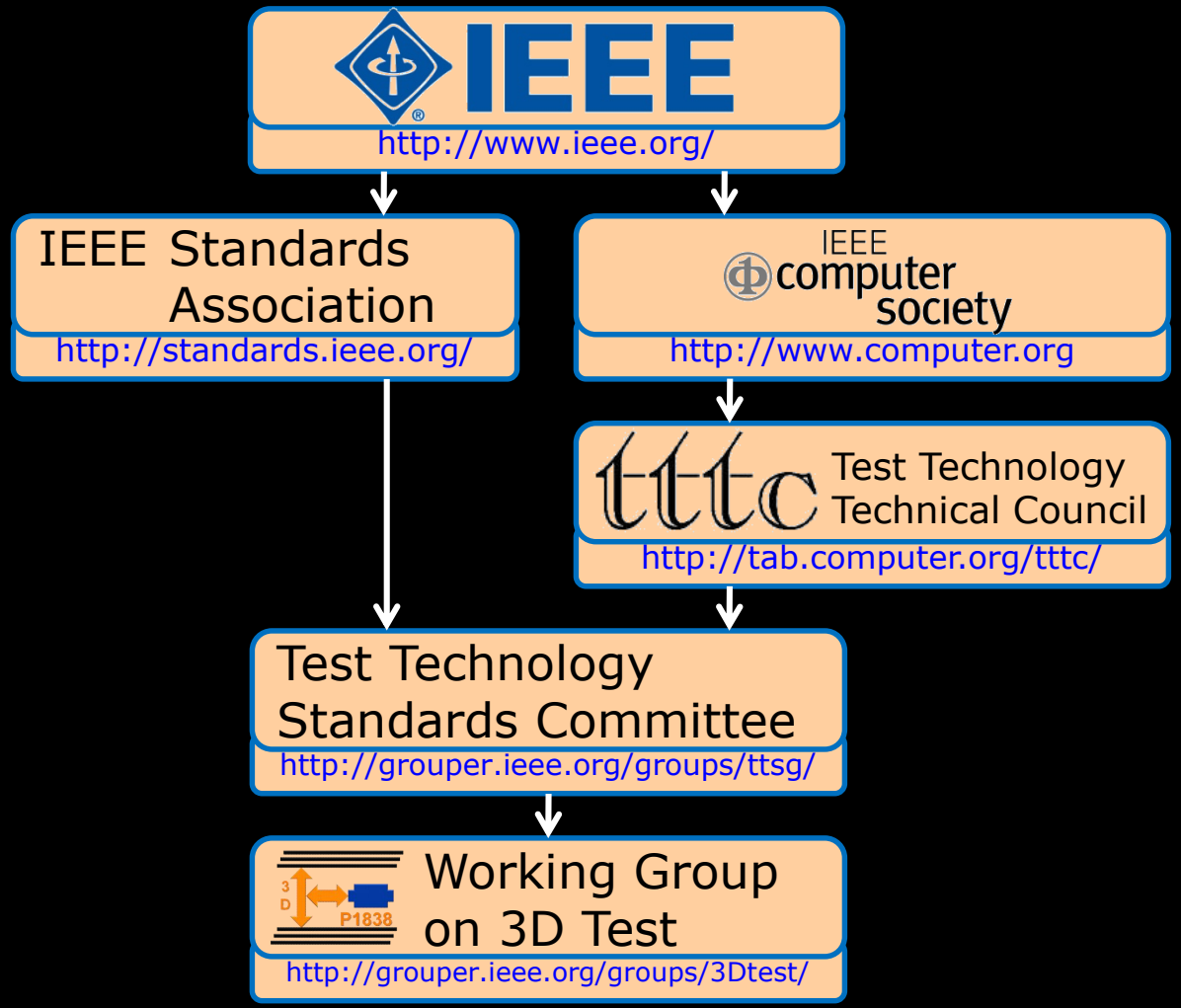
IEEE Std P1838

Working Group

IEEE P1838 History

- 3D-Test Standardization Study Group
  - January – November 2010
  - Submitted PAR to IEEE-SA NesCom in November 2010
- 3D-Test Standardization Working Group
  - Active since February 2011 after approval of PAR P1838 by IEEE-SA NesCom
- P1838: “Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits”
- Timeline
  - 2011-2013: Alignment on standardization direction
  - Since October 2013: “Tiger Teams” formed in focus areas
  - December 2015: Extension of PAR until December 2018

P1838 Working Group Organization



- **P1838 WG Officers**
  - Chair : Erik Jan Marinissen (imec)
  - Vice-Chair : Adam Cron (Synopsys)
  - Secretary : Sophocles Metsis (AMD)
  - Editor : Michael Wahl (Univ. Siegen)
- **WG Membership**
  - Open to professionals, no dues or fees
  - ~50 members across the industry
- **WG Meetings**
  - Bi-weekly conference calls:
    Thursdays 5-6pm Europe / 8-9am Pacific
  - Ad-hoc face-to-face meetings (e.g., @ITC)
- **Public Web-Site**
  - http://grouper.ieee.org/groups/3Dtest/

IEEE P1838 Sub-Groups

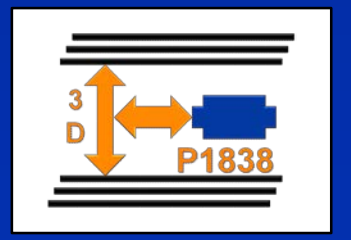
- Serial Control Mechanism – Al Crouch
  - Test access through serial IEEE 1149.1 TAP interface
- Die Wrapper Register (DWR) – Teresa McLaurin
  - Controllability and observability at die boundary
- Flexible Parallel Port (FPP) – Adam Cron
  - Multi-bit test access for high bandwidth
- Description Language – Sandeep Bhatia

Thank You

SiliconAid Solutions
Your DFT Partner...

P1838
Serial Test Access Architecture

Al Crouch
SiliconAid Solutions
2016 International Test Conference



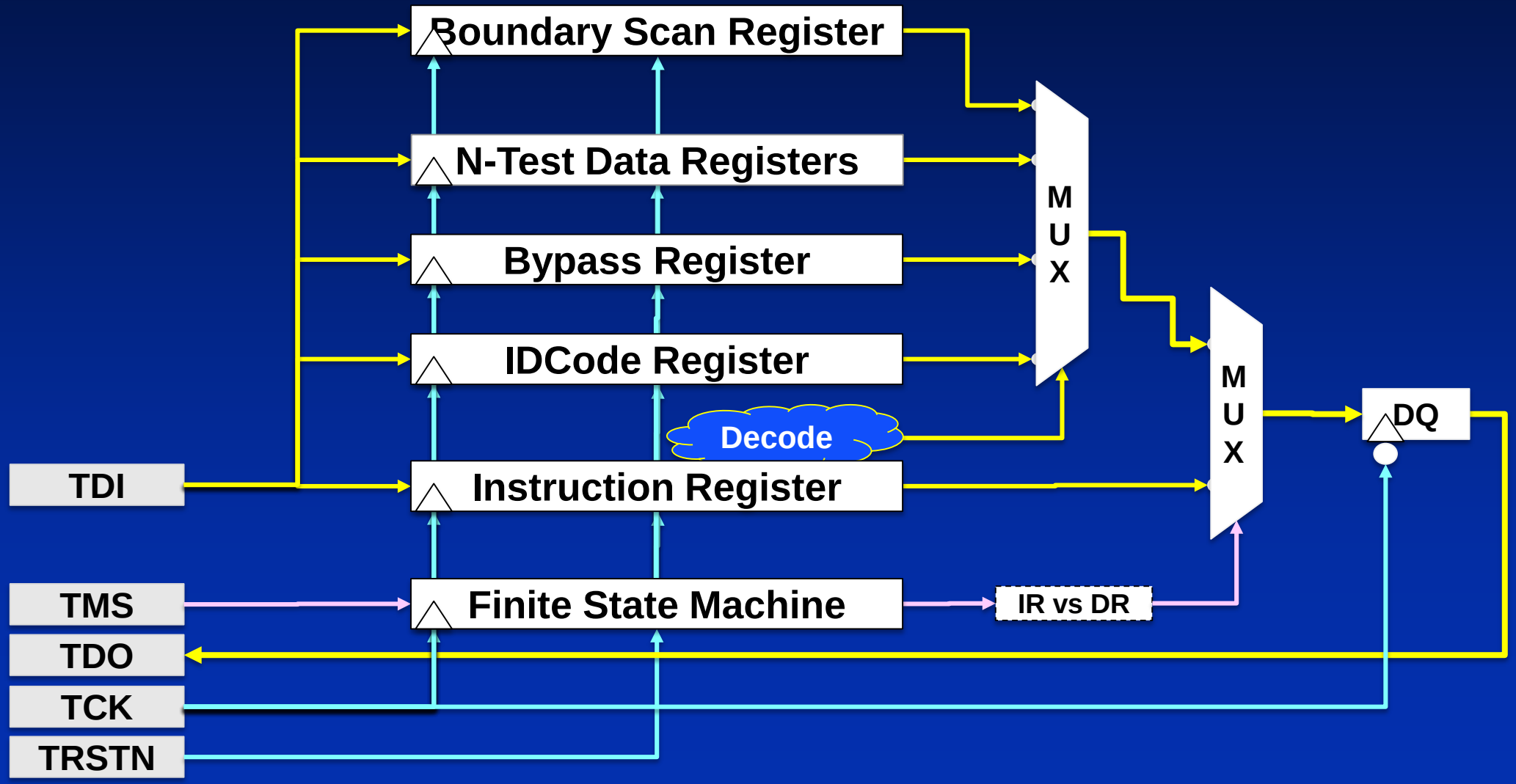
Outline

- Problem Statement
- Recap of a 2D solution
- The Pieces of the 2.5D and 3D Solution
- The Primary TAP Controller
- Secondary TAPs
- Feature Configuration Registers
- Instructions

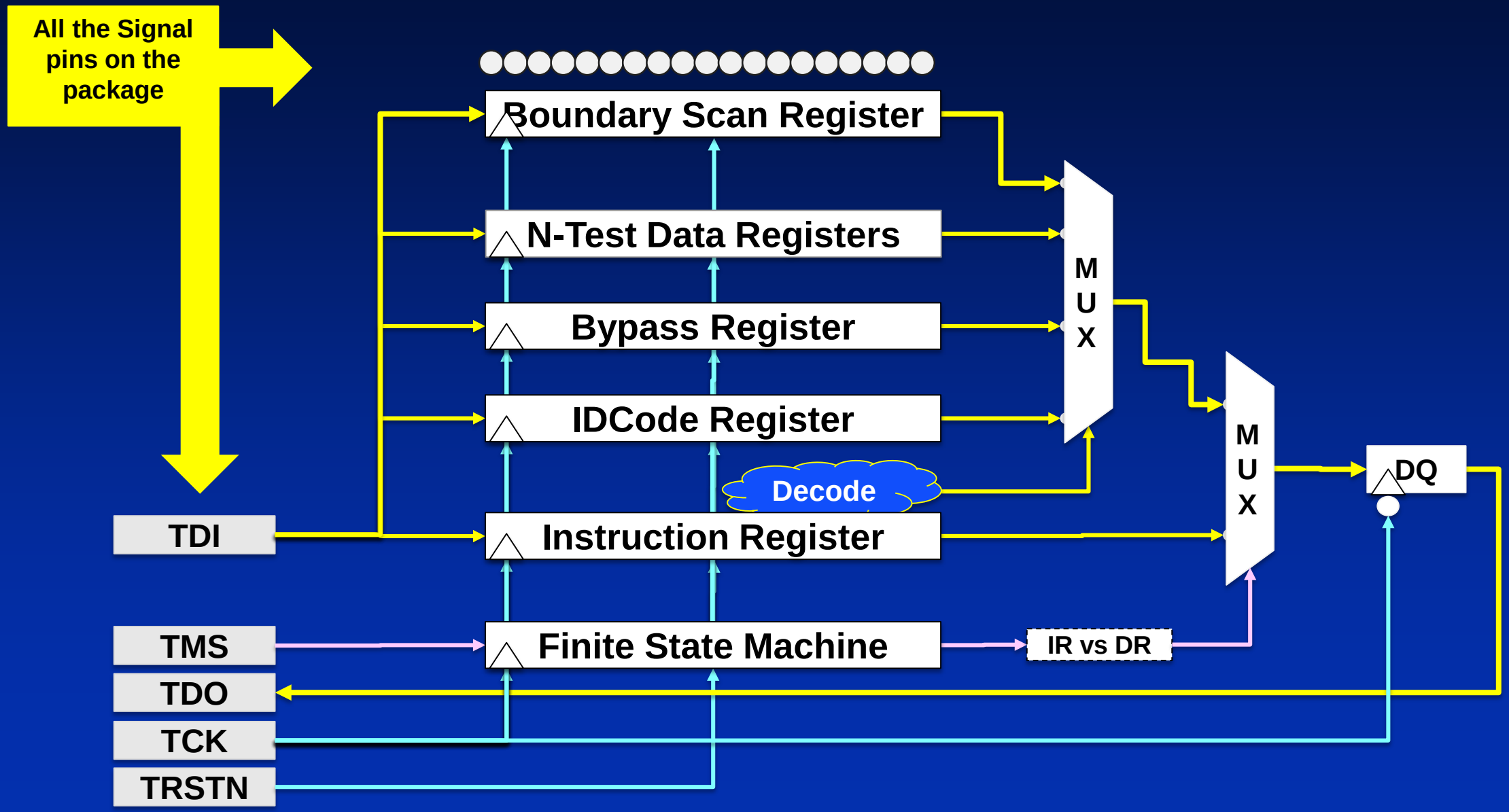
Problem Statement

The configuration of packaging such that die are connected to each other directly or through interposers creates an issue with the access to on-die test features, INTEST capabilities, and requires that die-to-die interconnect be verified in a manner similar to that of verifying the chip’s package connection being attached to a board. This drives the need for a 3D Serial Access solution.

Recap of 2D Solution – 1149.1 TAP Controller



Recap of 2D Solution – 1149.1 TAP Controller



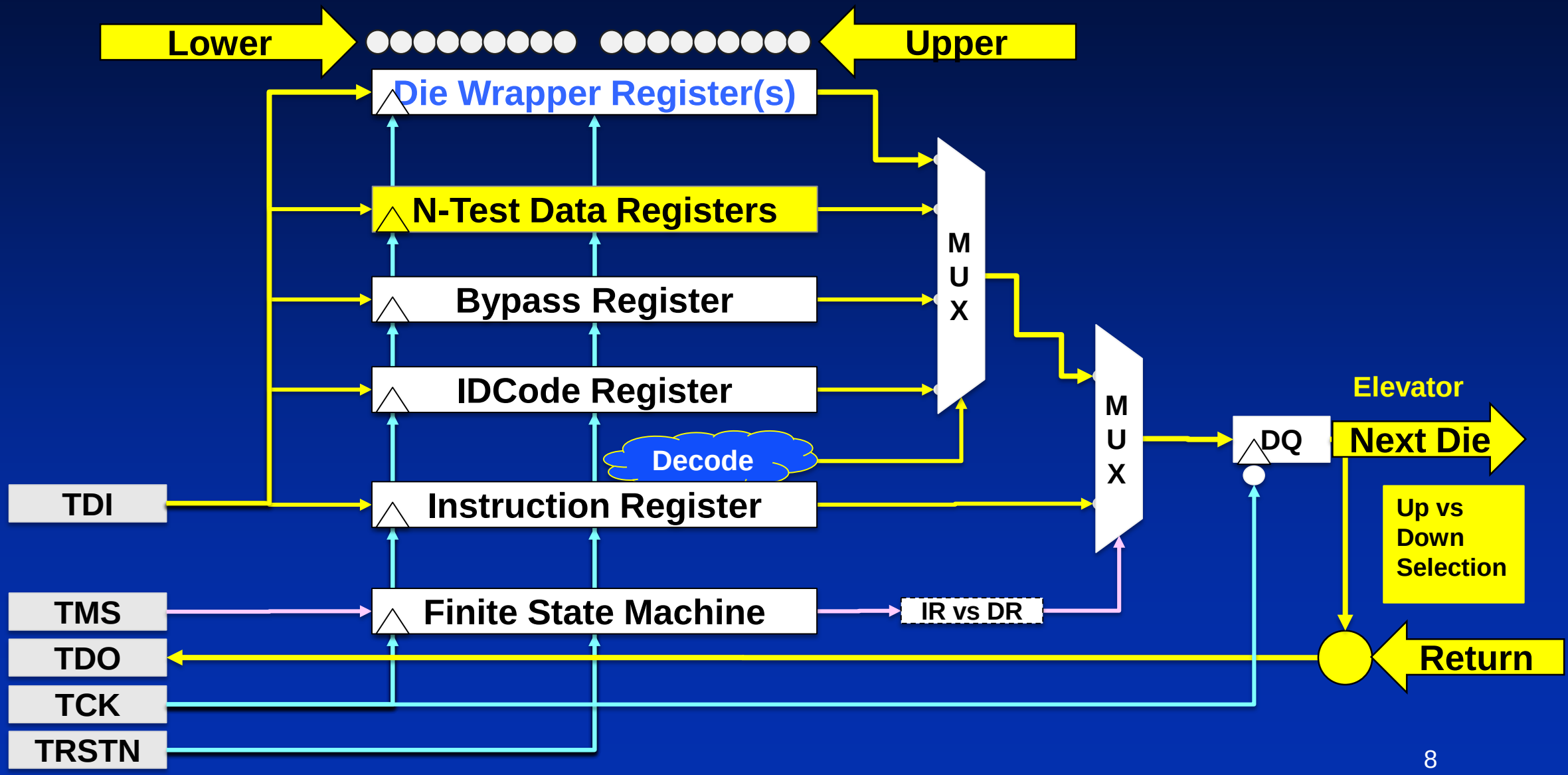
The 1149.1 TAP and TAP Controller

- Provides a defined signal interface
- Provides Instructions to select Registers
- Provides a Serial Access to Read/Write Registers
- Provides defined Registers for Test Features
  - Boundary Scan Register (interconnect, safety)
  - Bypass Register (reduction at board to 1-bit)
  - ID Code Register (identifies chip)

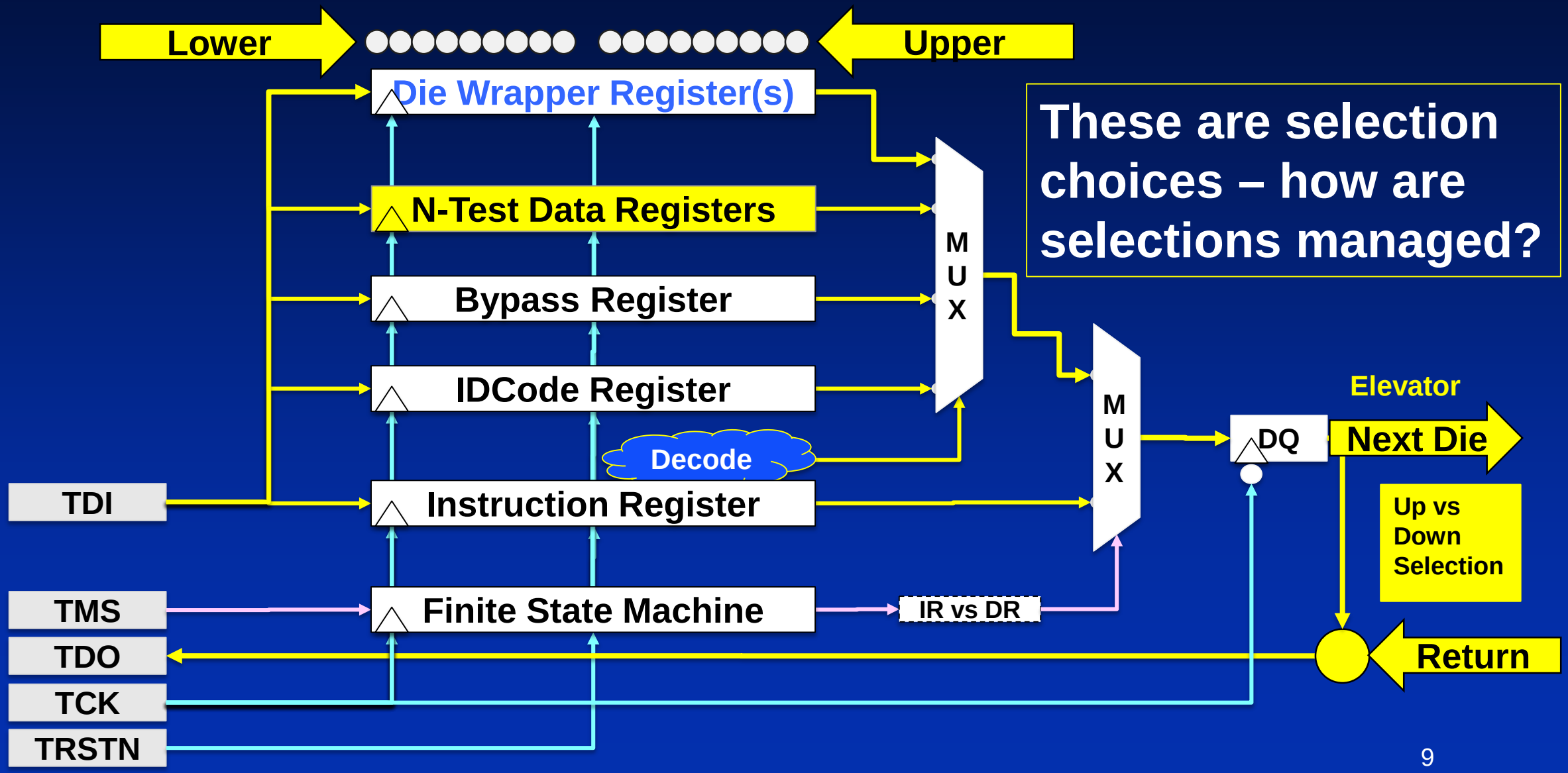
The P1838 3D Serial Access Requirements

- Must be Plug-and-Play (per die)
- Must provide a defined signal interface
- Must accommodate 2 or more signal connection interfaces
- Must extend vertically (direct die-to-die)
- Must provide defined Test Features
  - Die Wrapper Register (interconnect, safety)
  - Bypass Register (board compliance, stack management)
  - ID Code Register (identifies die)

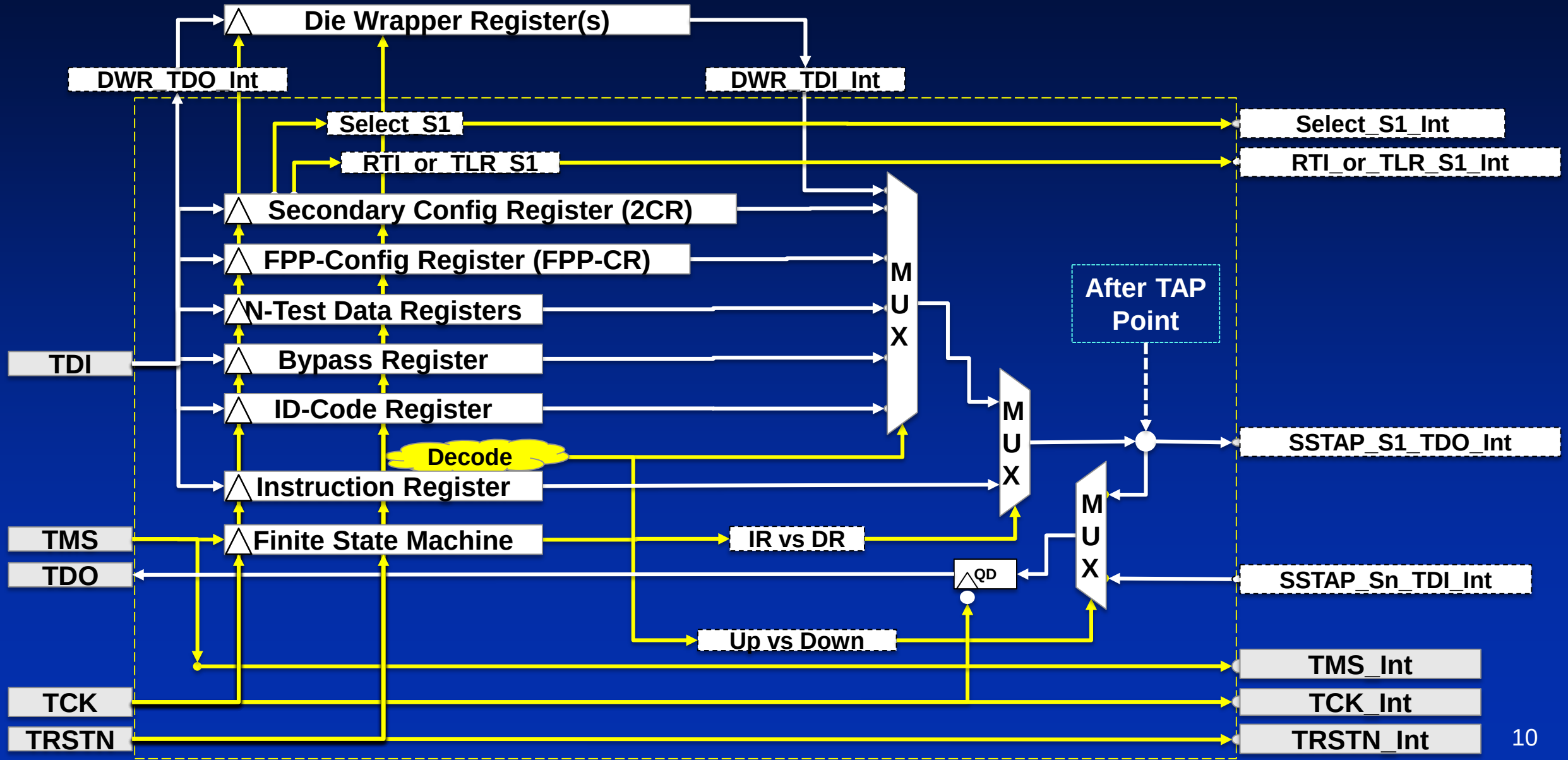
Example of 2.5D / 3D Added Requirements



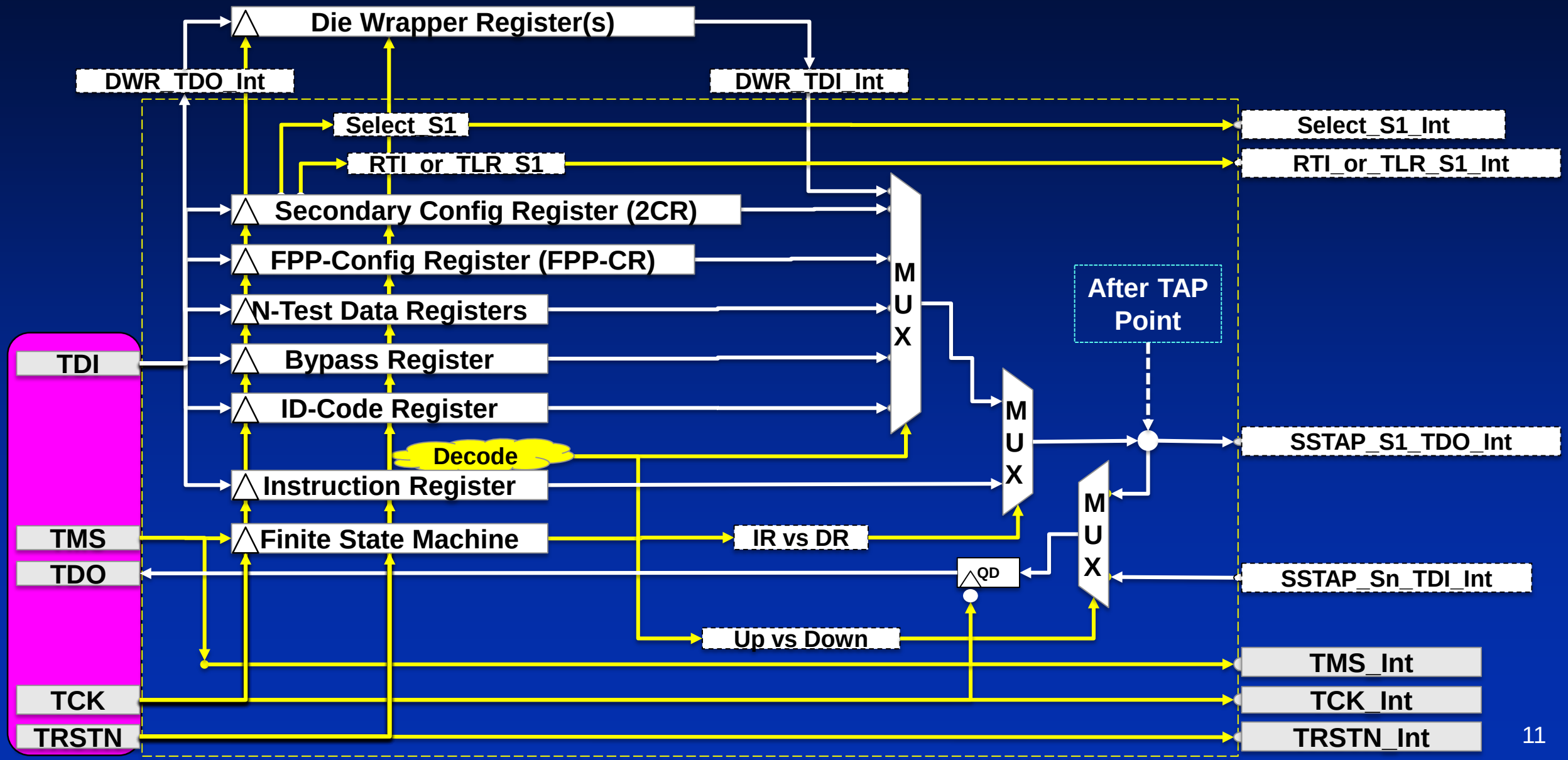
Example of 2.5D / 3D Added Requirements



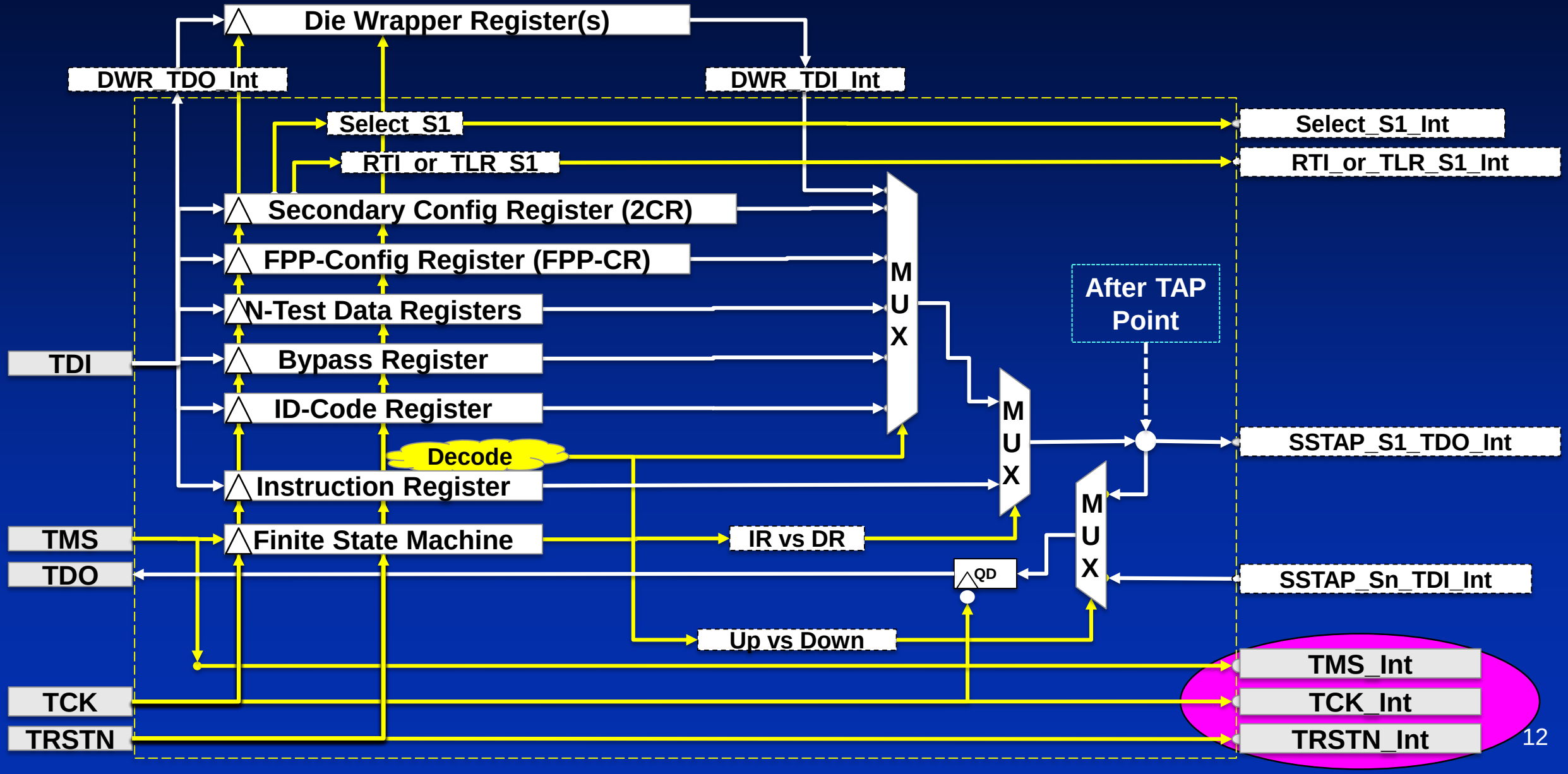
Example of 2.5D / 3D TAP Controller Solution



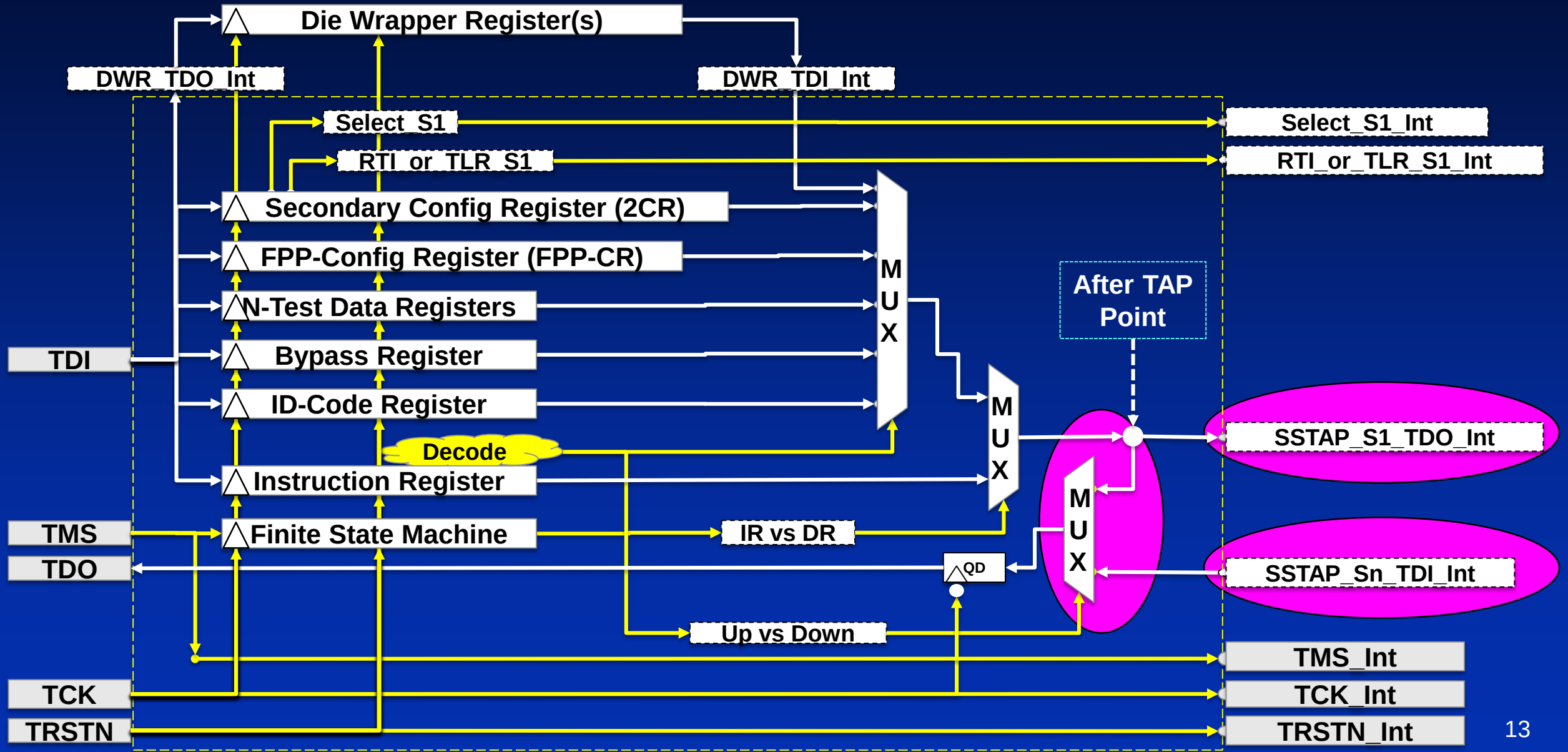
Example of 2.5D / 3D TAP Controller Solution



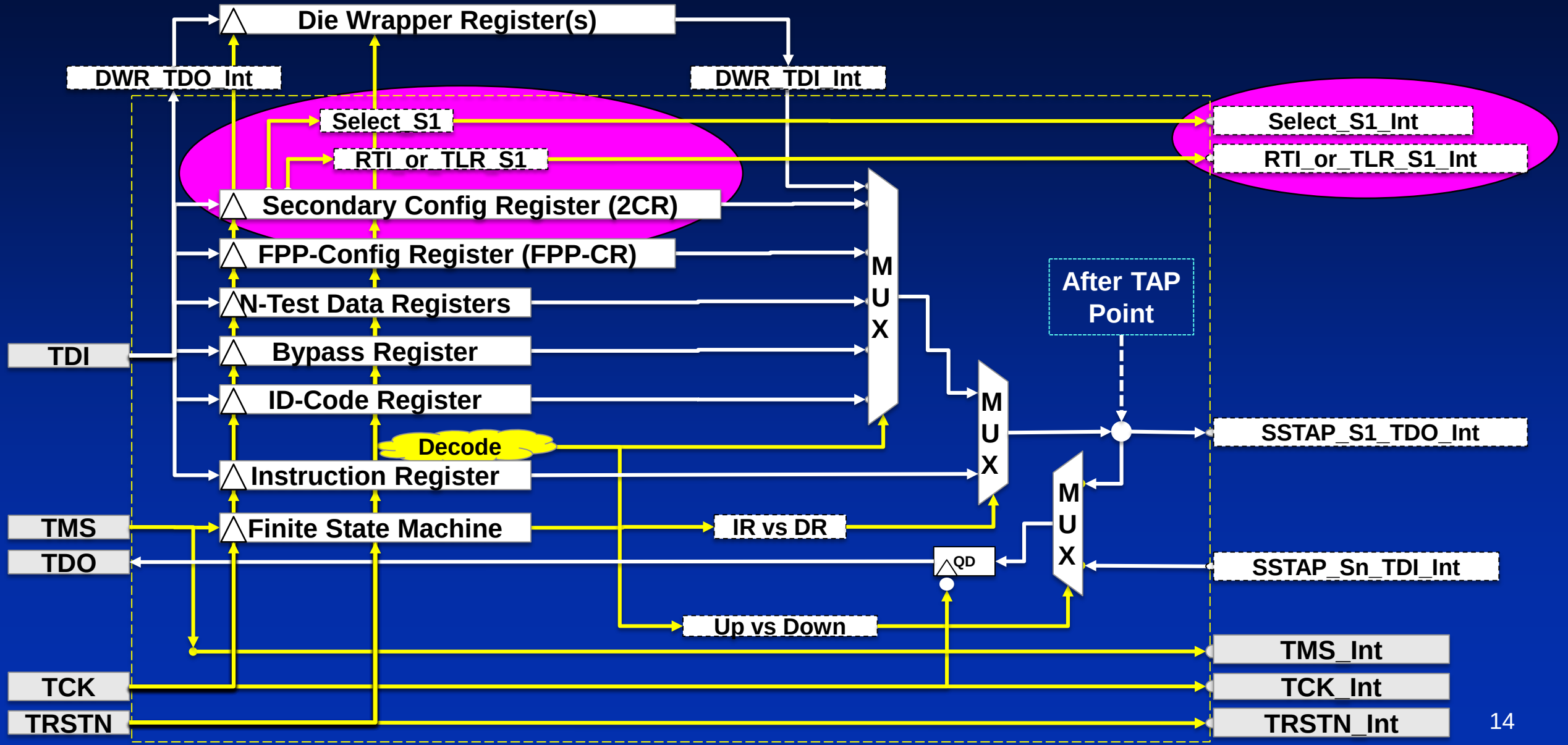
Example of 2.5D / 3D TAP Controller Solution



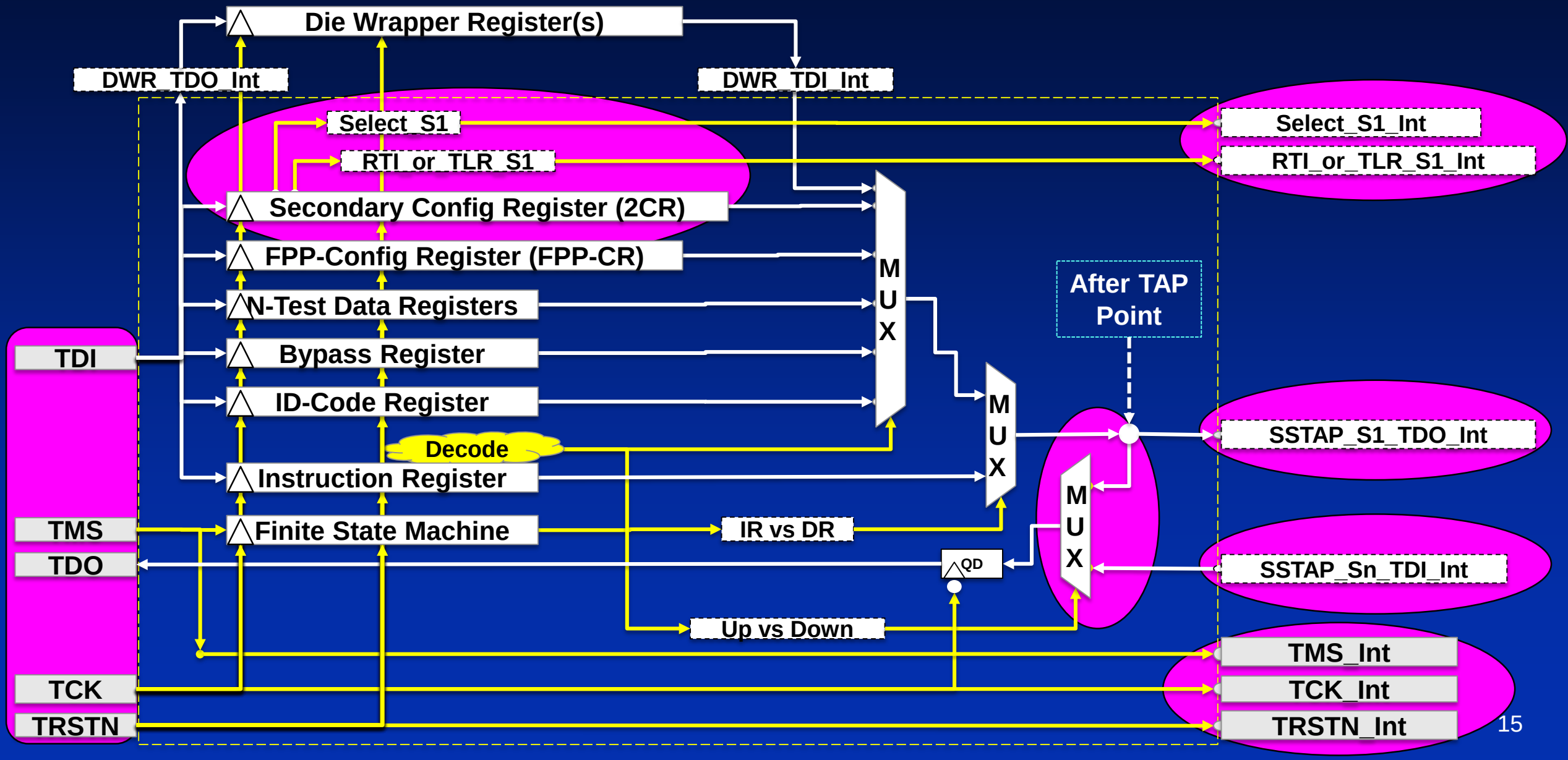
Example of 2.5D / 3D TAP Controller Solution



Example of 2.5D / 3D TAP Controller Solution



Example of 2.5D / 3D TAP Controller Solution

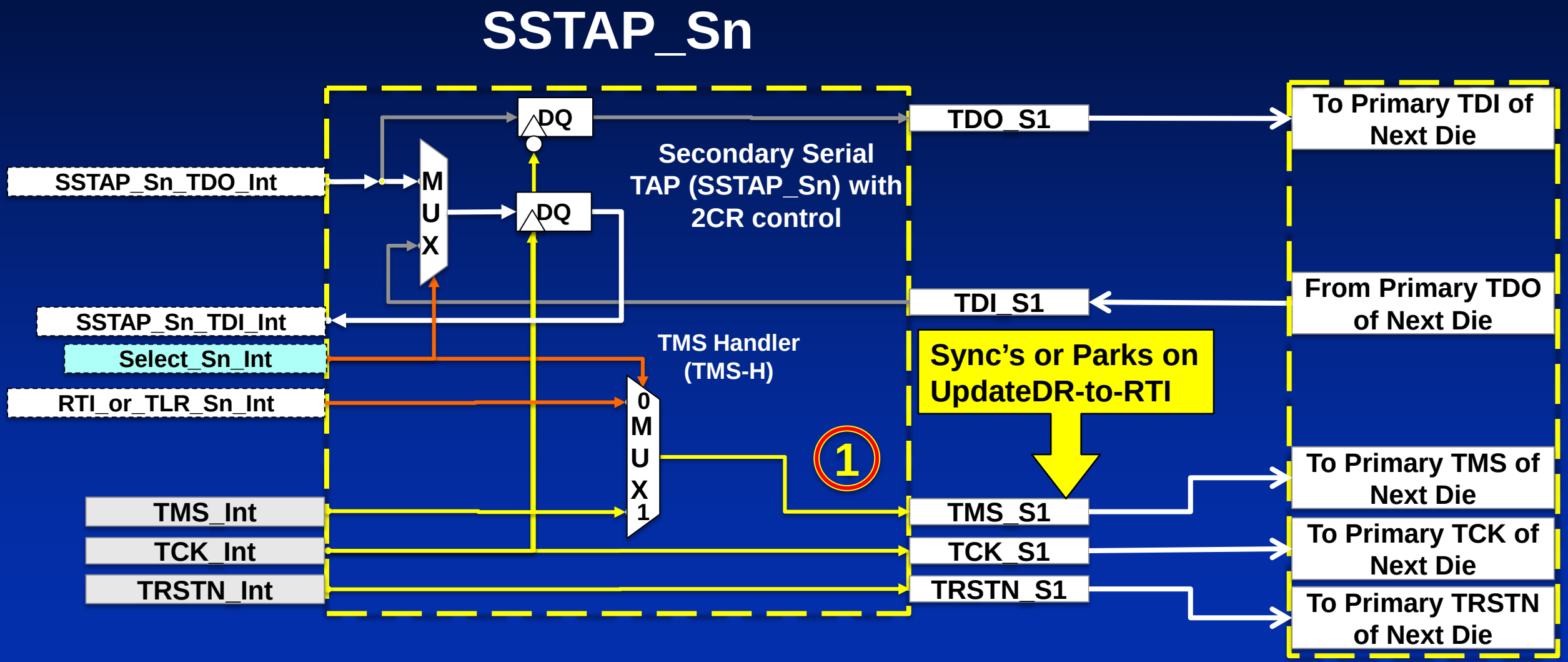


The P1838 3D Secondary TAP Requirements

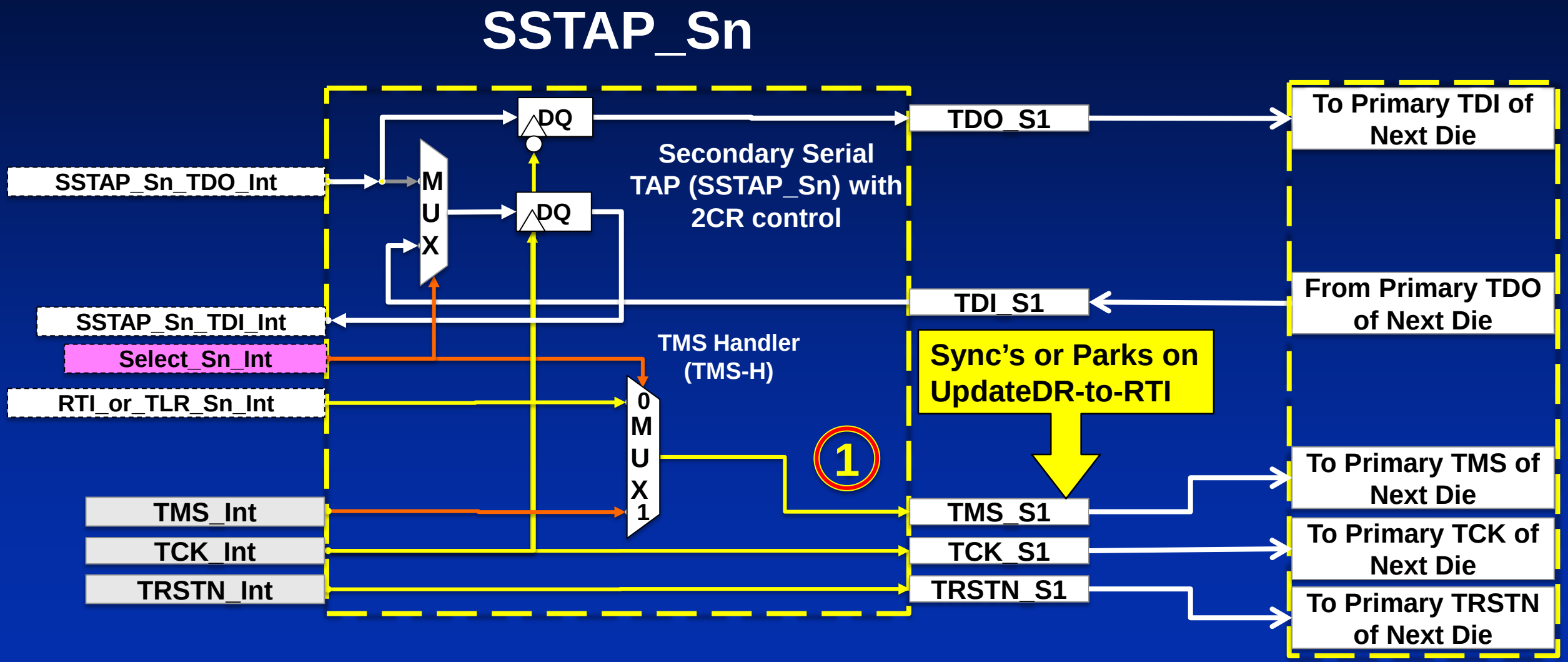
The Elevator

- Must be Plug-and-Play (per next die attach point)
- Must provide a defined secondary signal interface
- Must accommodate multiple towers/stacks
- Must allow selected/non-selected configuration
  - Synchronized TAPC on current die and next die
  - Parked in Run-Test-Idle (default)
  - Parked in Test-Logic-Reset

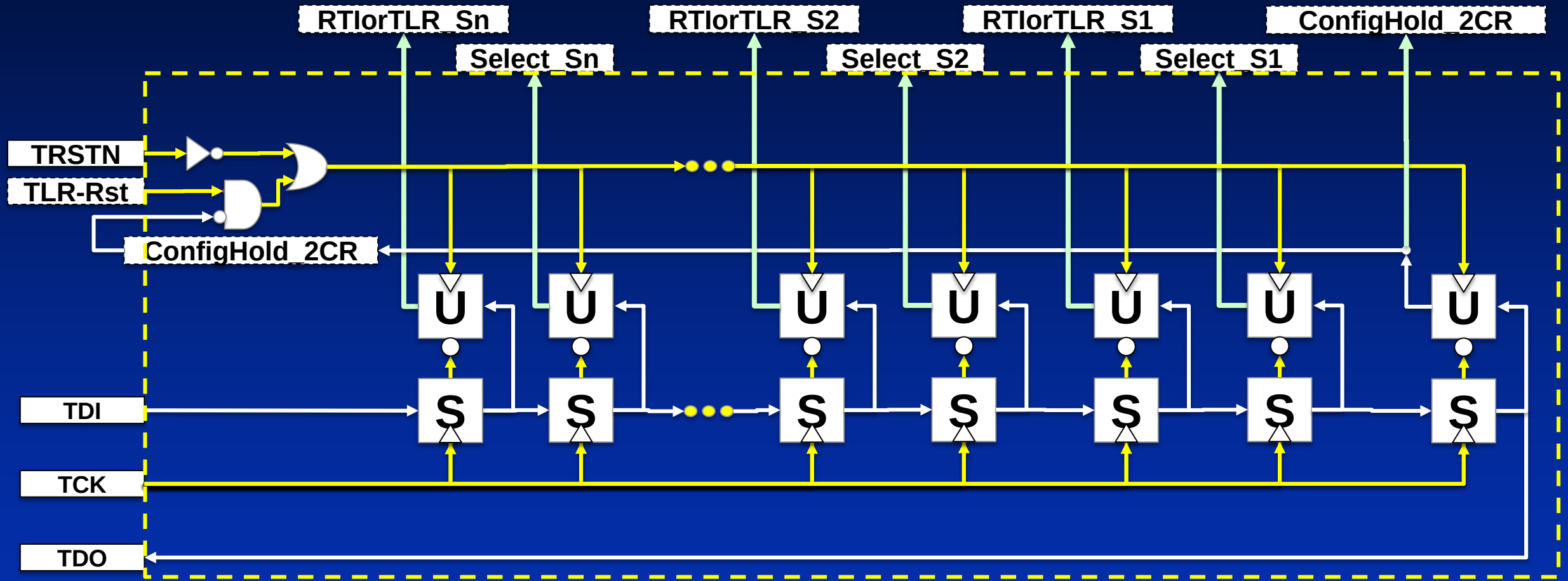
Example of 2.5D / 3D Secondary Serial TAP



Example of 2.5D / 3D Secondary Serial TAP



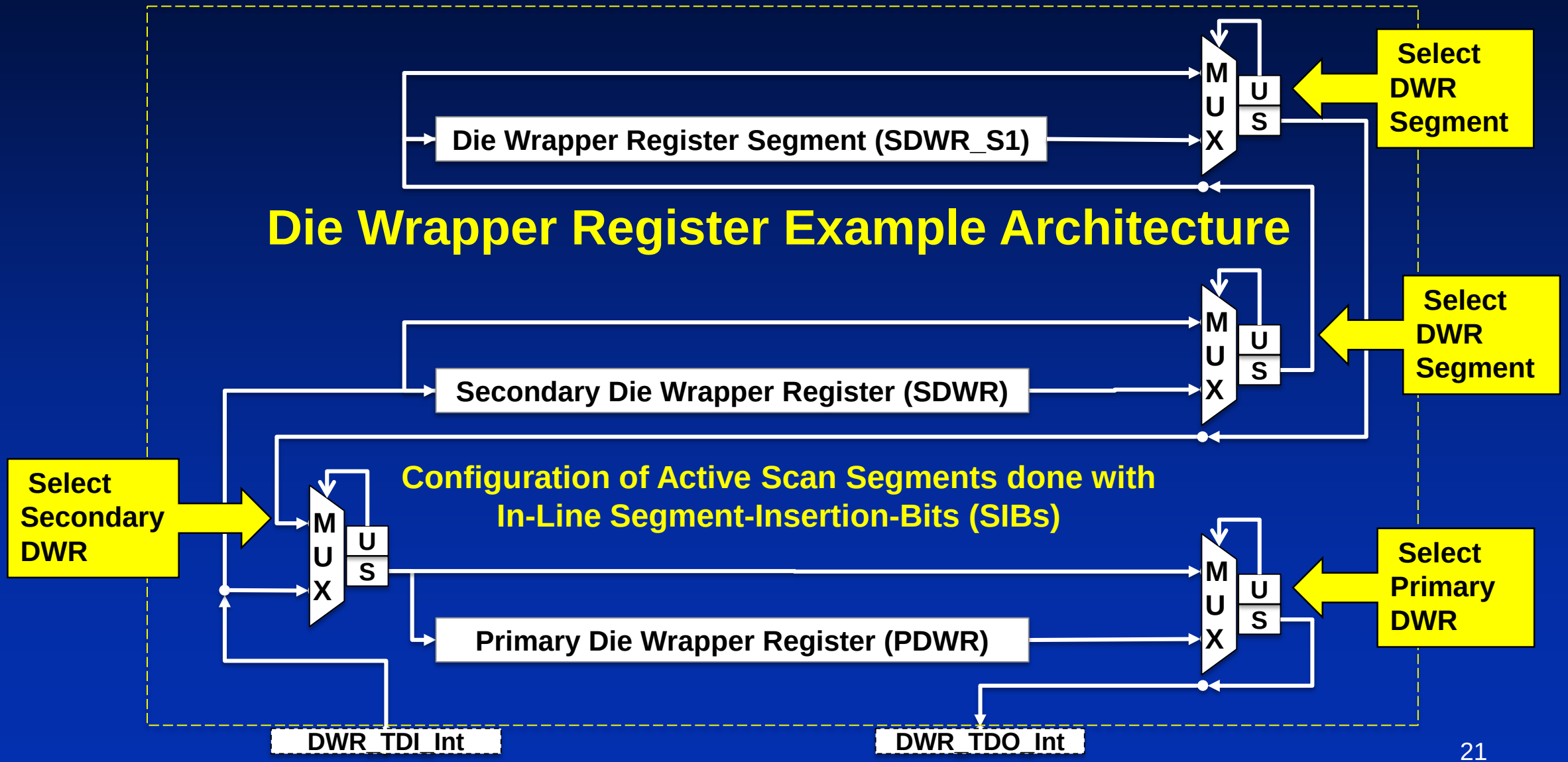
The Secondary Serial TAP Configuration Register (2CR)



The P1838 3D Die Wrapper Requirements

- Must be Plug-and-Play (per die)
- Must accommodate multiple surfaces/interfaces
- Must allow segmented configurations
- Must allow featured mode configurations
  - Inward-facing test
  - Outward-facing test
  - Functional configuation (transparent)

Example of 2.5D / 3D Die Wrapper Register



The P1838 3D Instruction Requirements

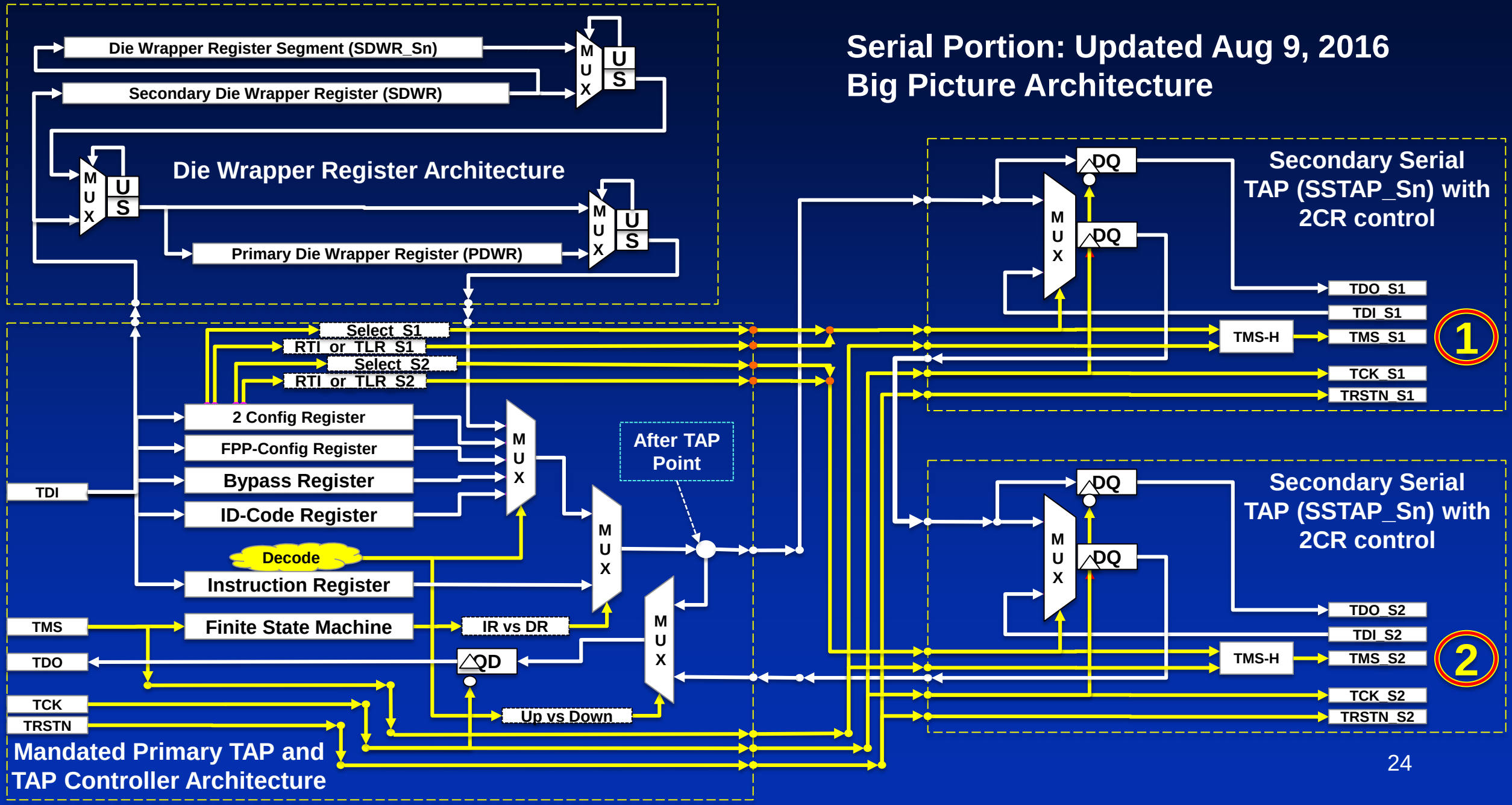
- Must allow 1149.1 Mandated/Optional Instructions
- Must manage Instruction Explosion
- Must allow P1838 Mandated/Optional Features
- Instructions are allowed to be represented in other ways
  - In-Line control and configuration bits
  - Configuration Registers (includes 1500 WIRs)
  - Feature Registers

Table of Instructions

Instruction Name	Selects Register	Selects Direction	Selects Persistence	Selects Segments	Configures DWR	Comments
BYPASS[1]	Bypass	2CR	NA	NA	NA	1149.1 = BypassDown
IDCODE[1]	ID Code	2CR	NA	NA	NA	1149.1 = JEDEC #s
EXTEST[1]	DWR Primary	2CR	TMP Reg	SIBs	Outfacing	1149.1 permission
INTEST[1]	DWR Primary	2CR	TMP Reg	SIBs	Infacing	1149.1 permission
PRELOAD[1]	DWR	2CR	TMP Reg	SIBs	Transparent	1149.1 permission
SAMPLE[1]	DWR	2CR	TMP Reg	SIBs	Transparent	1149.1 permission
CLAMP[1]	Bypass	2CR	TMP Reg	NA	Outfacing	1149.1 permission
SELDWR[2]	DWR	2CR	TMP Reg	SIBs	DWR-CR	P1838 DWR Management
SEL2CR[2]	2CR	NA	ConfigHold	NA	NA	P1838 Elevator Management
SELFPP[2]	FPP-CR	NA	ConfigHold	NA	NA	P1838 Parallel Port

1. **With the 2CR zero'ed out, these instructions match their 1149.1 definitions**
2. **The Select Config Register Instructions select only the target CR, no concatenations allowed**

Conclusion: P1838 Per Die Serial Test Architecture



Thank You

3D-IC Test Standard IEEE P1838 – Die Wrapper Register

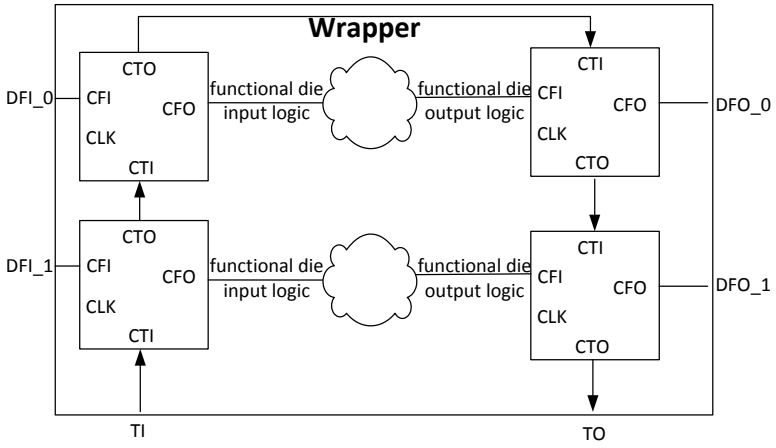
ARM

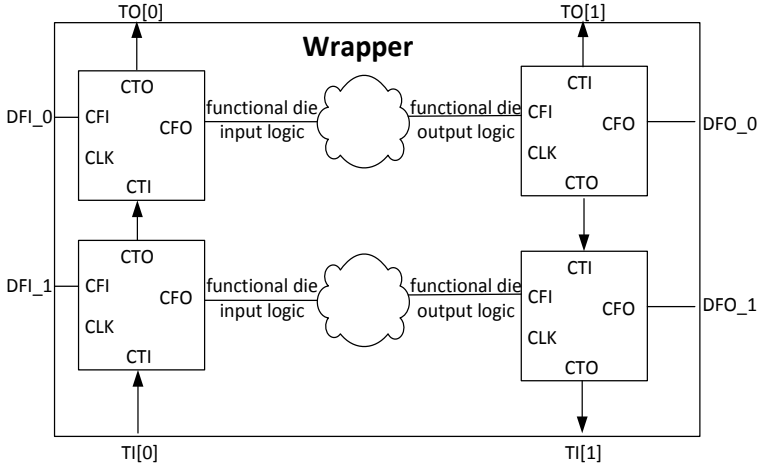
Teresa McLaurin
ARM Fellow and DFT Architect

ITC 2016
16 Nov 2016

What is a Die Wrapper Register (DWR)

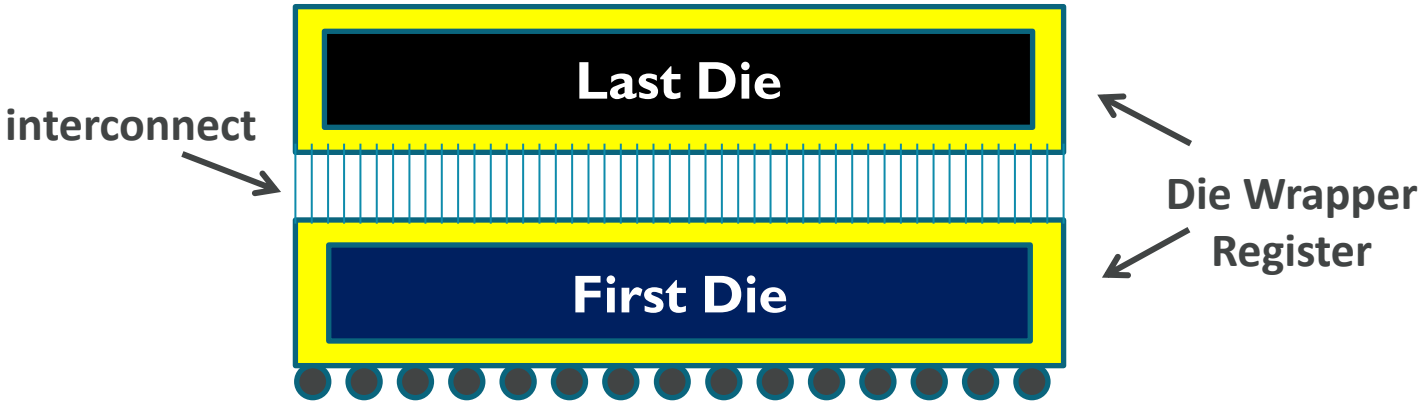
- Collection of P1838 compliant wrapper cells connected into one or more scan chains
  - Test data required at a die terminal is contained solely in the wrapper cell provided to this terminal





Die Wrapper Register

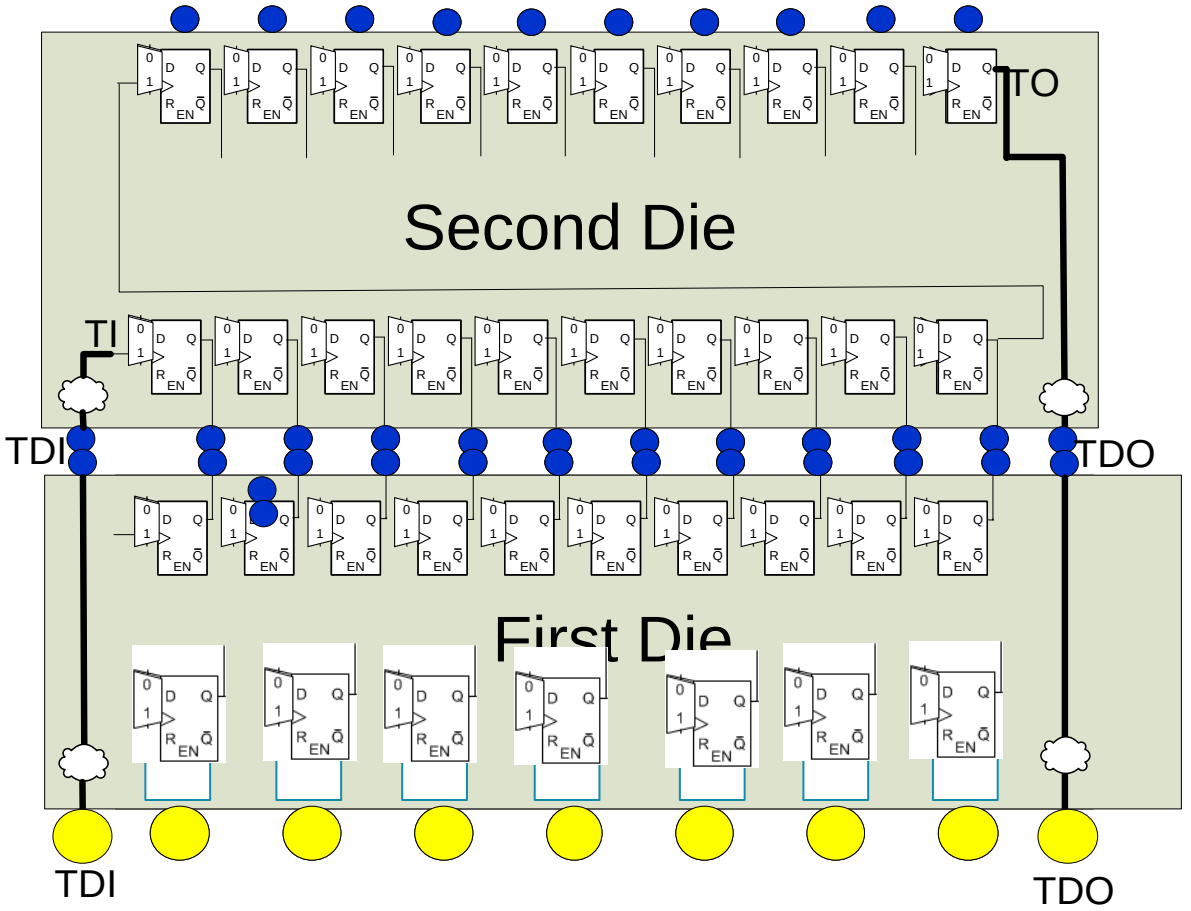
- Enables isolated INTEST of the die
- Enables EXTEST (using only the wrapper logic) of the TSV interconnect logic



- All digital die terminals, with some exceptions, must have a fully-provisioned wrapper cell. Non-digital signals under discussion
  - Example exceptions – clocks, asynchronous set or reset, specified test signals

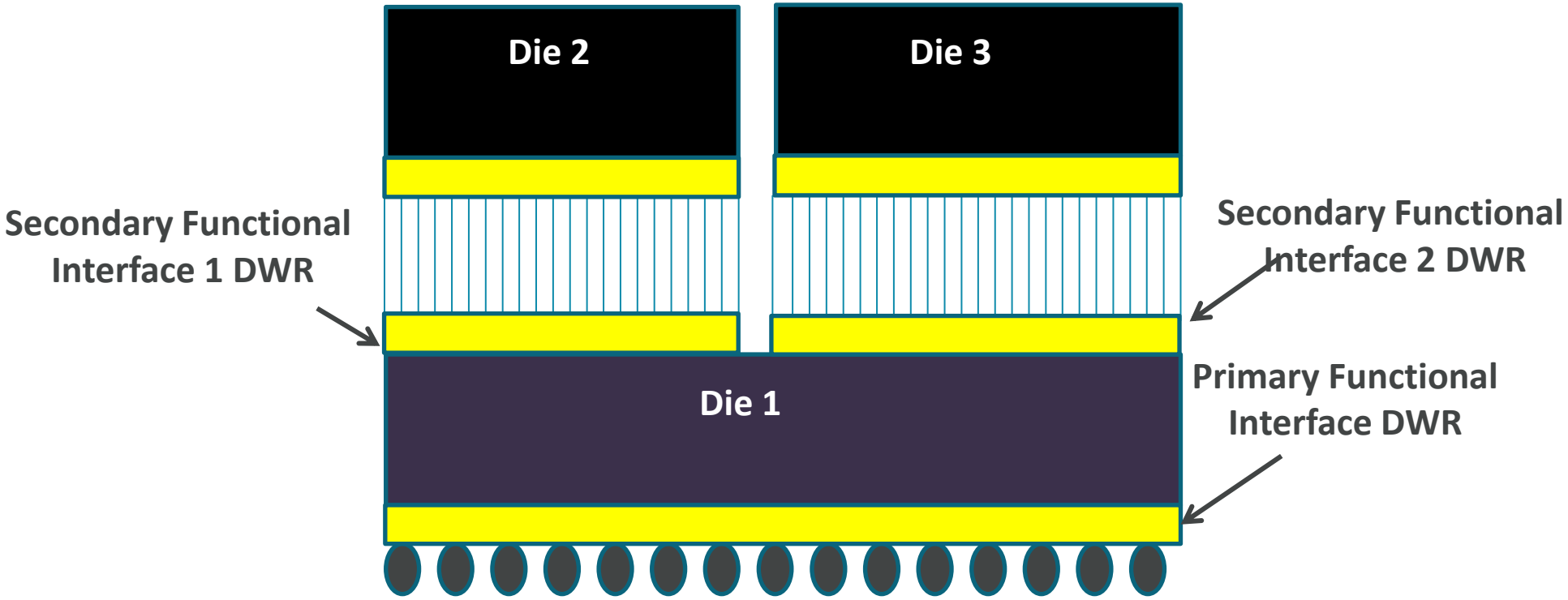
DWR

- One serial configuration in which all DWR cells are concatenated into a single chain
- At least one DWR cell between TI and TO when a segment is selected
- Can reuse 1149.1 bscan and most 1500 WBR cells



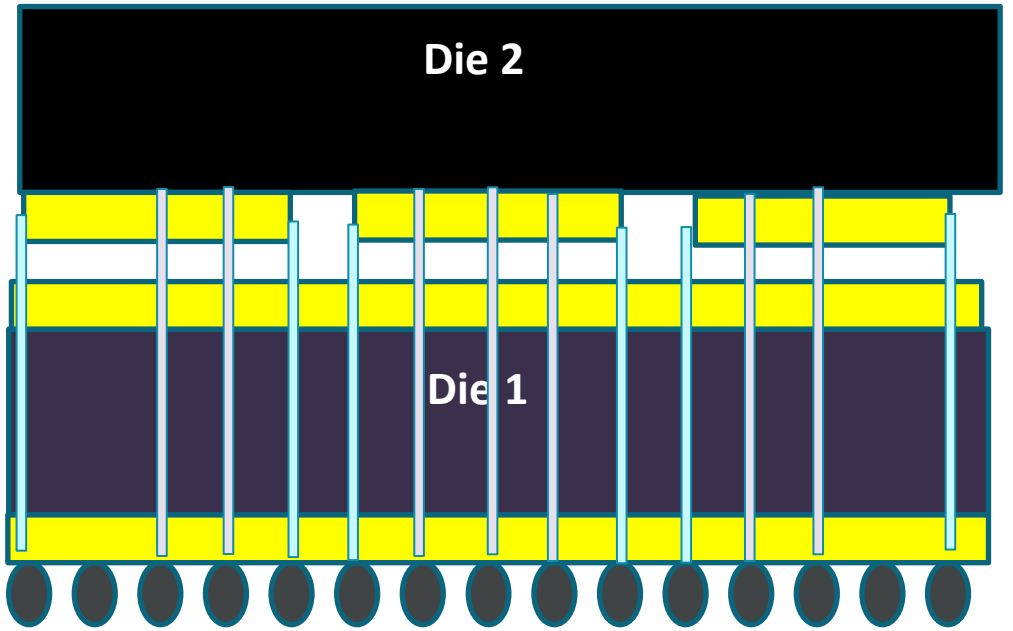
Die Wrapper Register

- Recommendation: each primary and secondary functional interface segment should have its own DWR serial chain



Die Wrapper Register

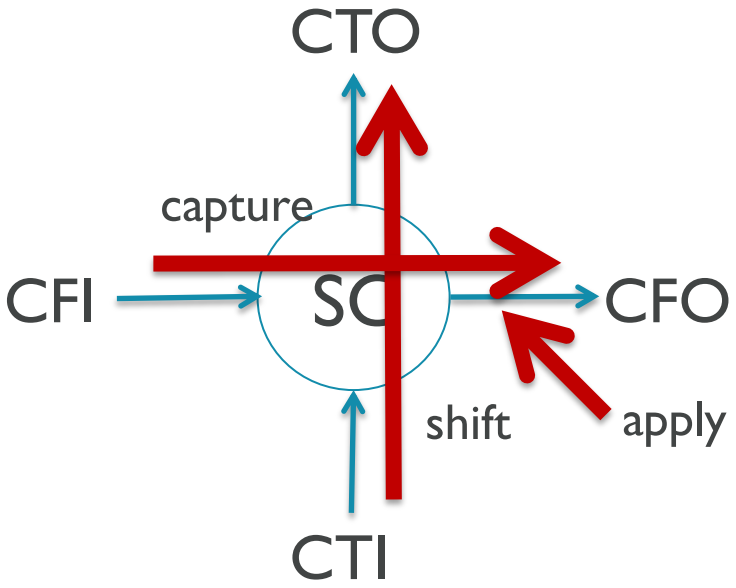
- More DWR segments permitted. When multiple segments (scan chains) must be run simultaneously the Flexible Parallel Port (FPP) is used.

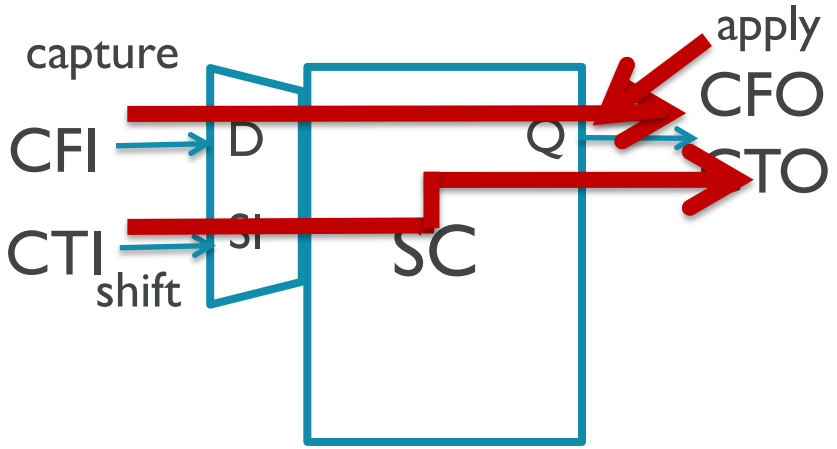


Die Wrapper Register (DWR) Cells

Two types of DWR cells

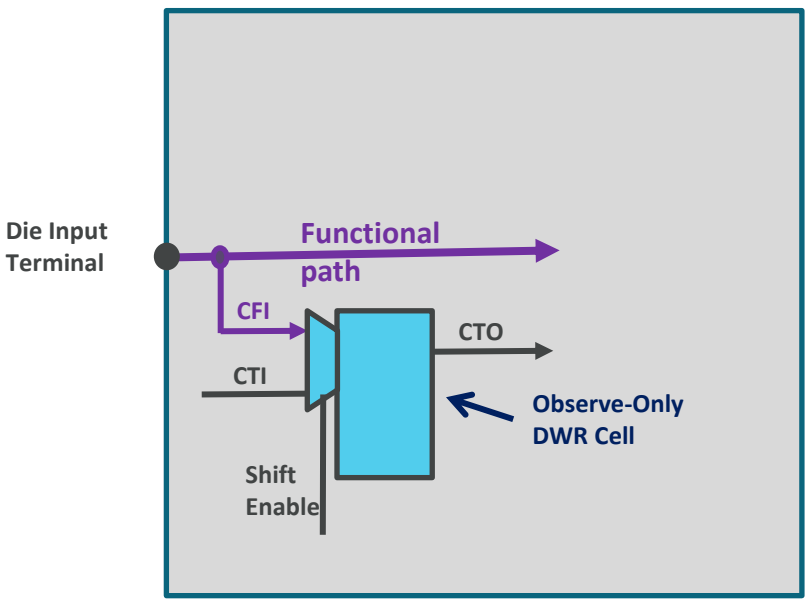
- Fully-provisioned – minimum requirements
  - One cell between the CFI and CFO
  - One cell between the CTI and CTO
  - Service Shift event – An event whereby the data is moved from CTI into the storage element
  - Service Capture event – An event whereby the data is moved from CFI into the storage element
  - Enable Apply event – An event where by the data is applied at CFO

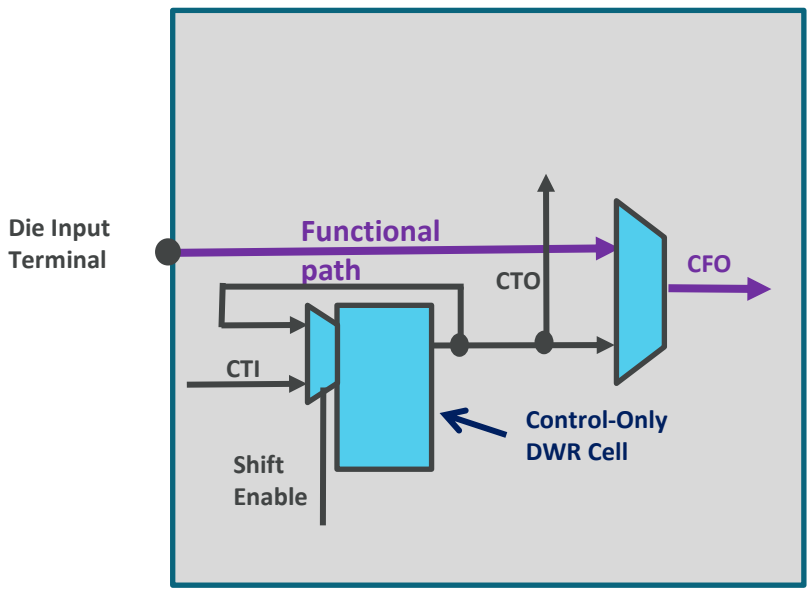




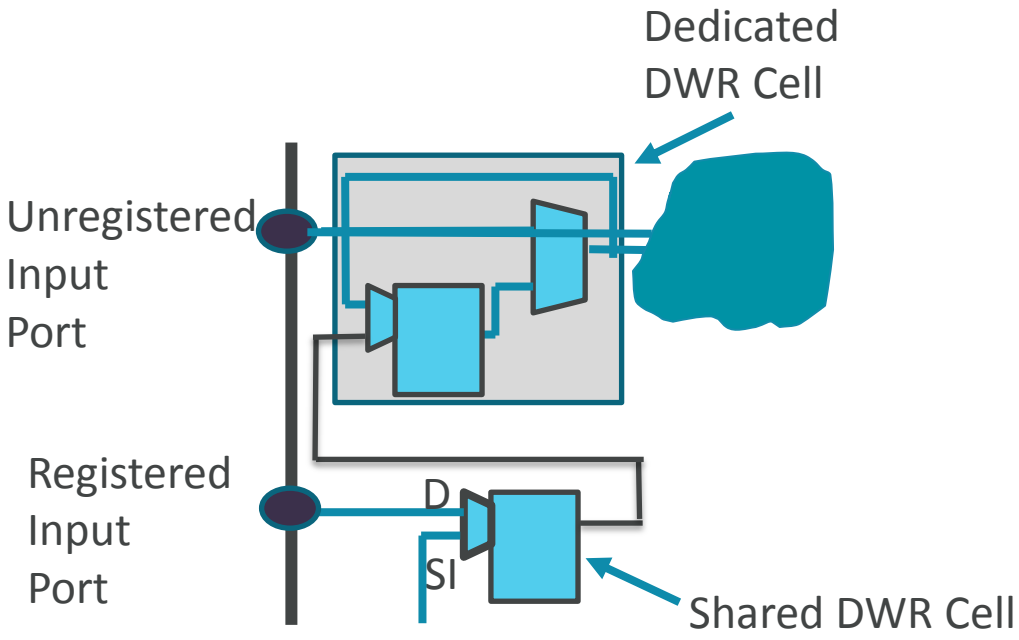
Two types of DWR cells

- Partially-provisioned
  - Shift
  - Capture or Apply





Shared and Dedicated Wrapper Cells



Options

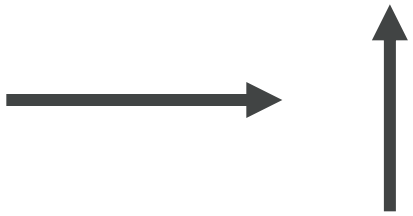
- DWR cells can have an update capability and it is required or recommended in some cases
  - For safety
  - For special control
  - Reuse of a boundary scan cell
- DWR cells can have safe gates
- DWR cells can be observe or control only
- DWR cells can have multiple sequential elements

Describing DWR cells

Bubble diagram

- Implementation independent representation
- Graphical building blocks:

Storage element
(e.g. Flipflop)



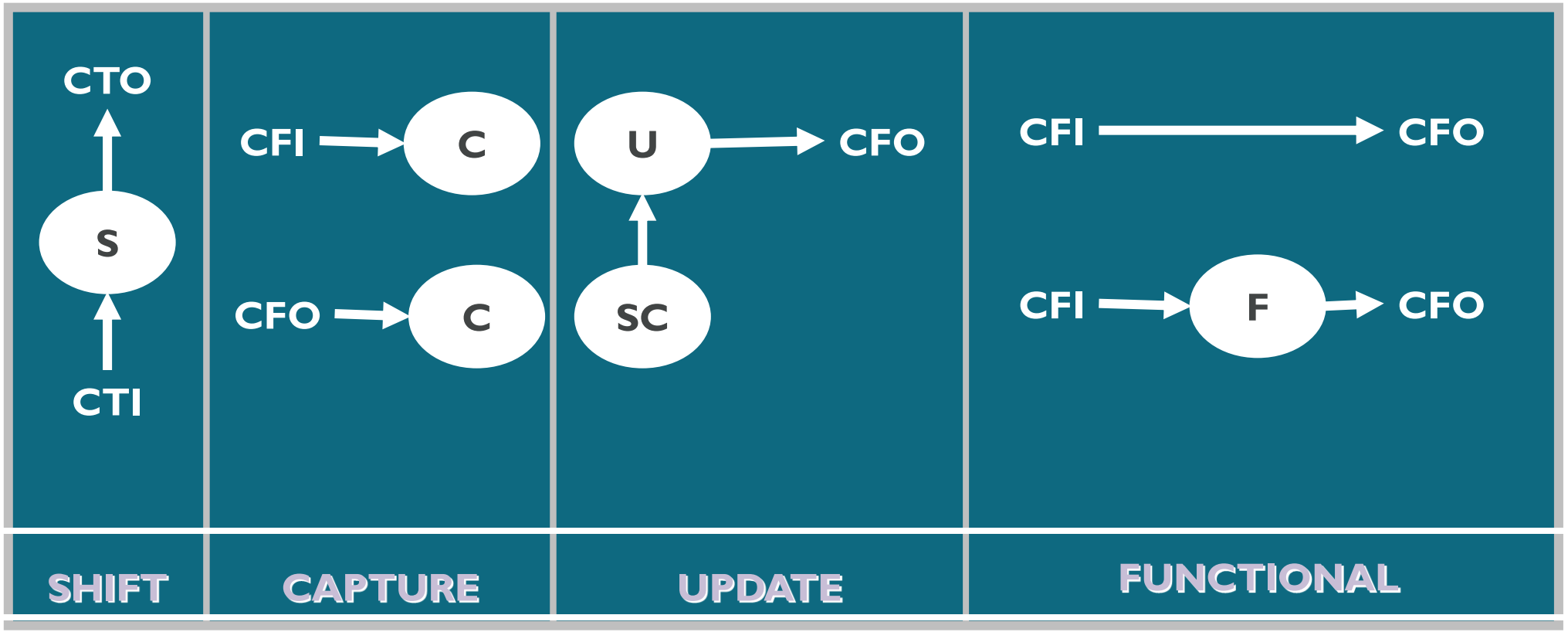
Data path

Data flow decision point
(e.g. mux)

S: Shift
C: Capture
U: Update
F: Functional element

DWR cell operations

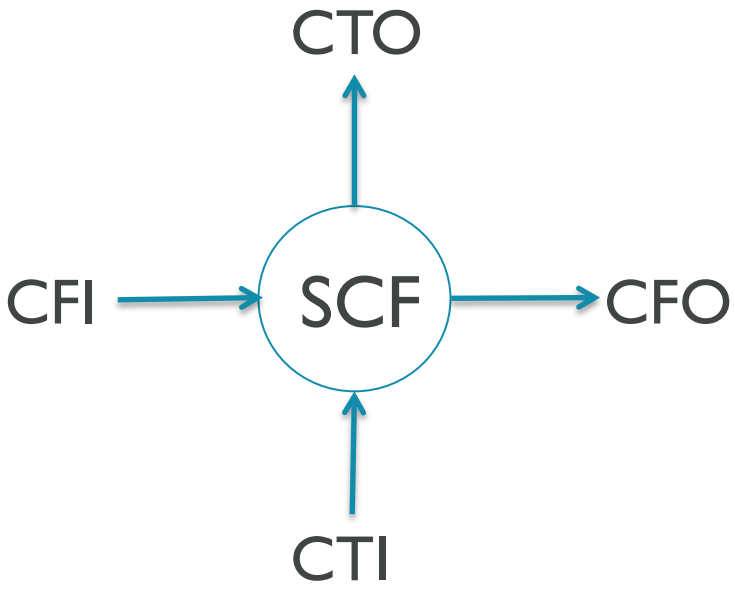
Bubble diagram representation



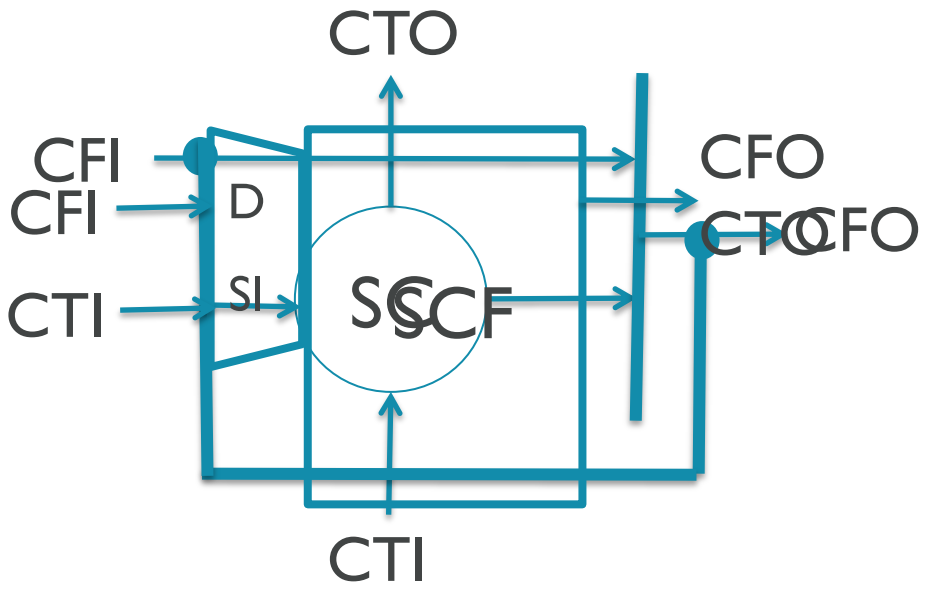
DWR Naming Convention

DWR Naming Convention

- /DC(_S[DF]\d+)(_C([IO\d+]))|N)?(_U)?(_O)?(_G[01]?)?/

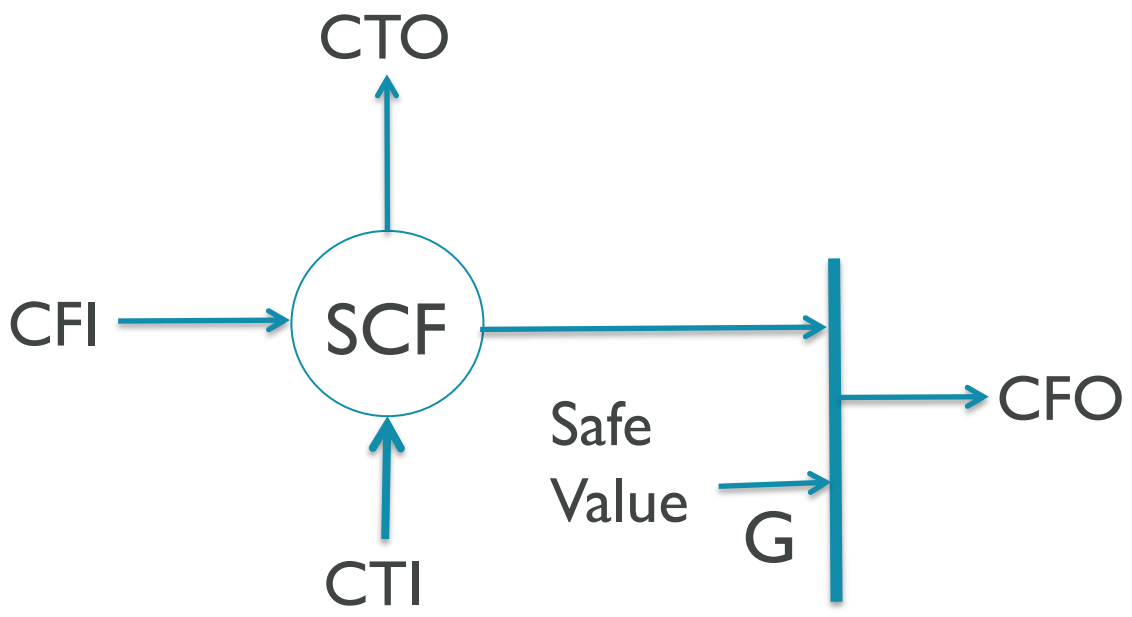


DC_SF1_CI1



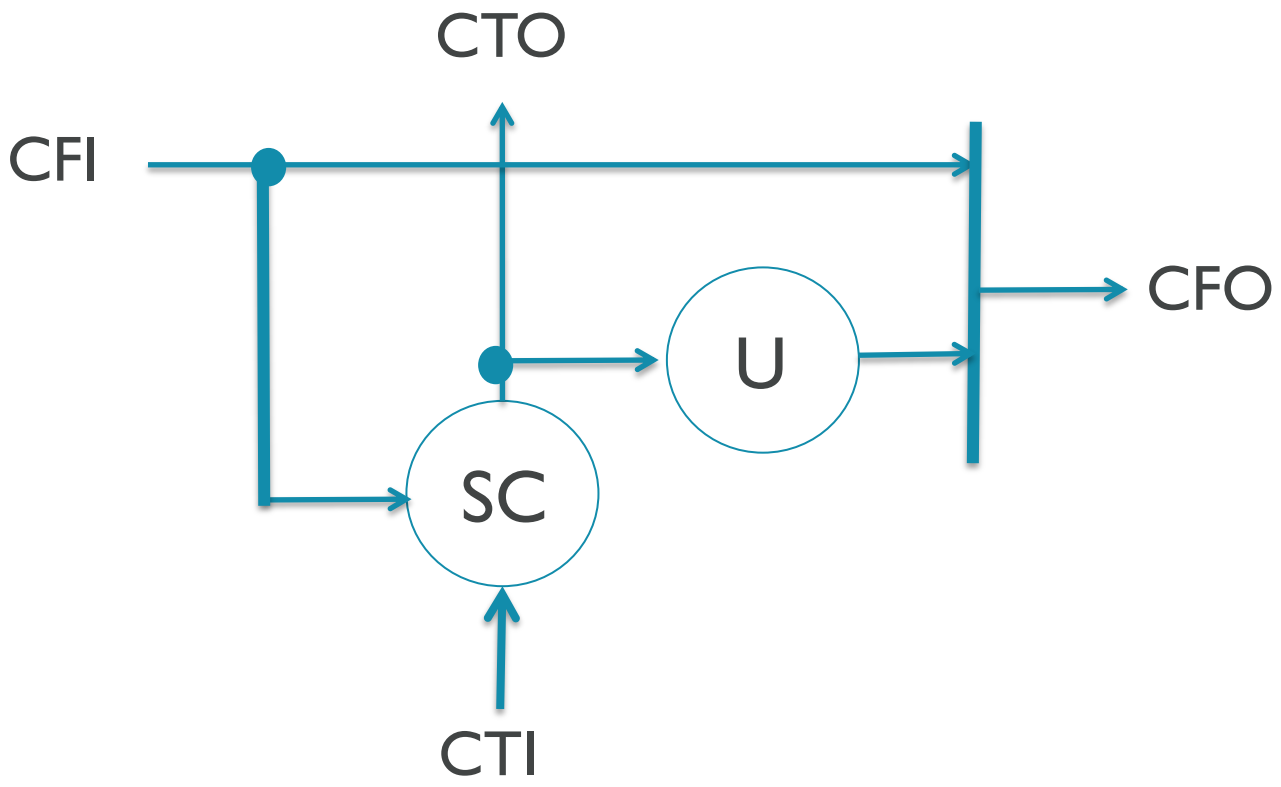
DC_SD1_CO1

Shared DWR cell with the optional Gate



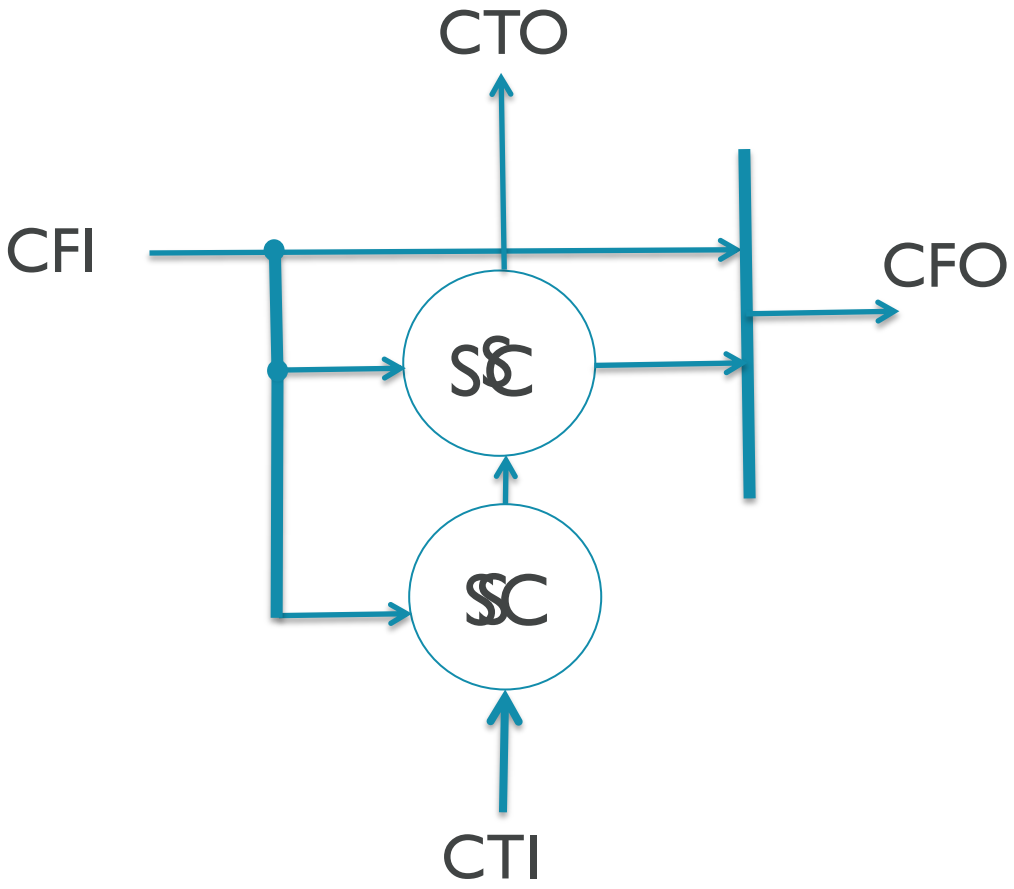
DC_SF1_CI1_G

DWR cell with the update option



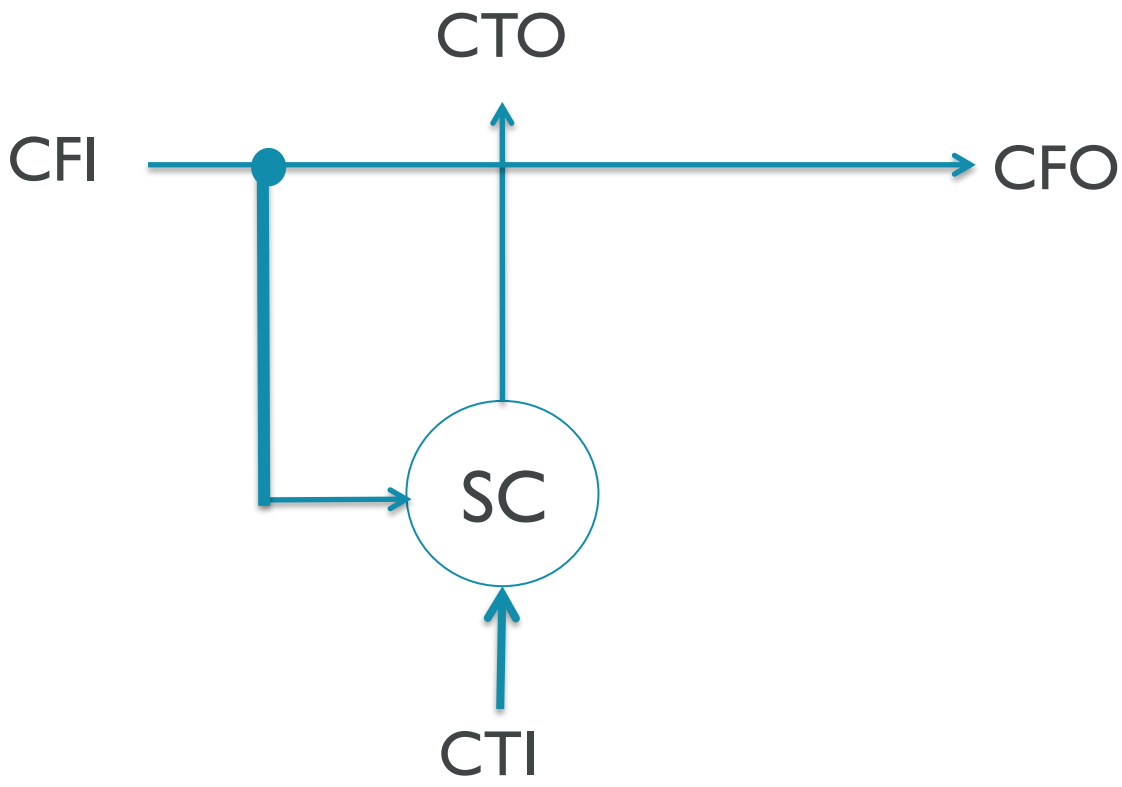
DC_SD1_CI1_U

DWR cell with multiple sequential elements



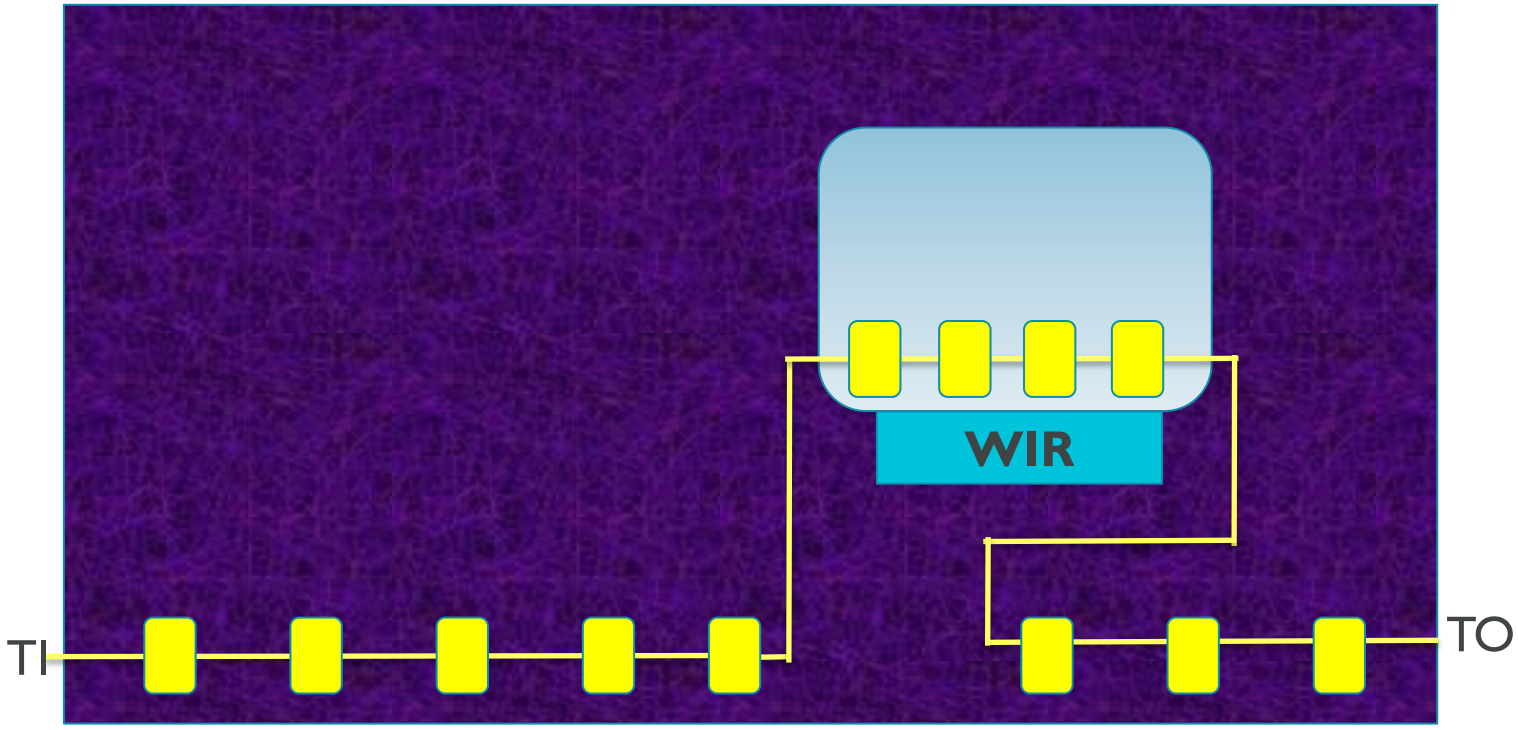
DC_SD2_CI2

Partially-provisioned DWR cell



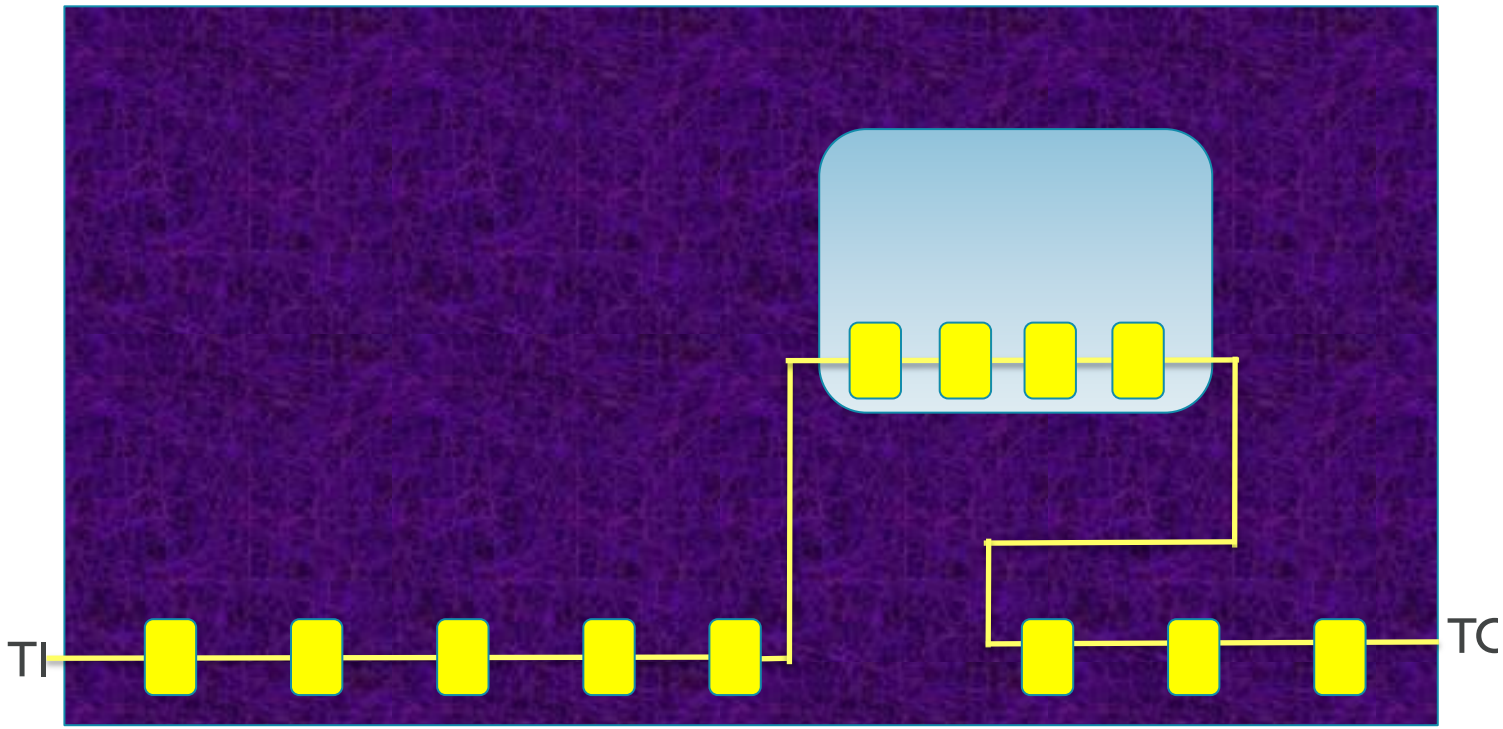
DC_SD1_CI1_O

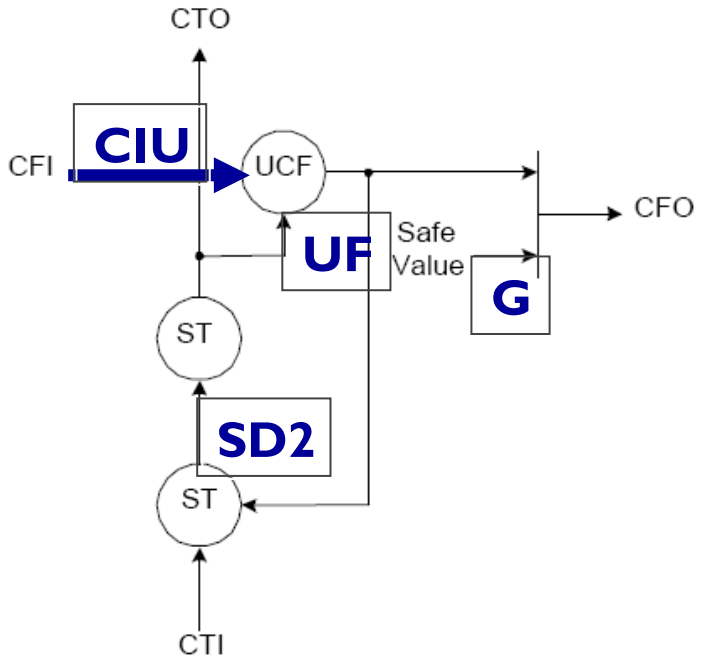
Reuse of IEEE 1500 wrapper cells in DWR



Reuse of IEEE 1500 wrapper cells in DWR

- /DC(_S[DF]\d+)(_C([IO\d+]))|N)?(_U)?(_O)?(_G[01]?)?/
- /WC(_S[DF]\d+)(_C([IO]))|N)?(_U)?(_O)?(_G[01]?)?/

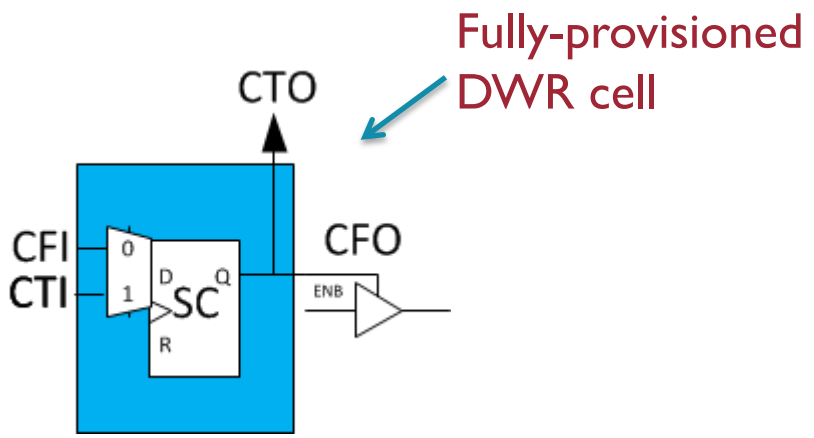


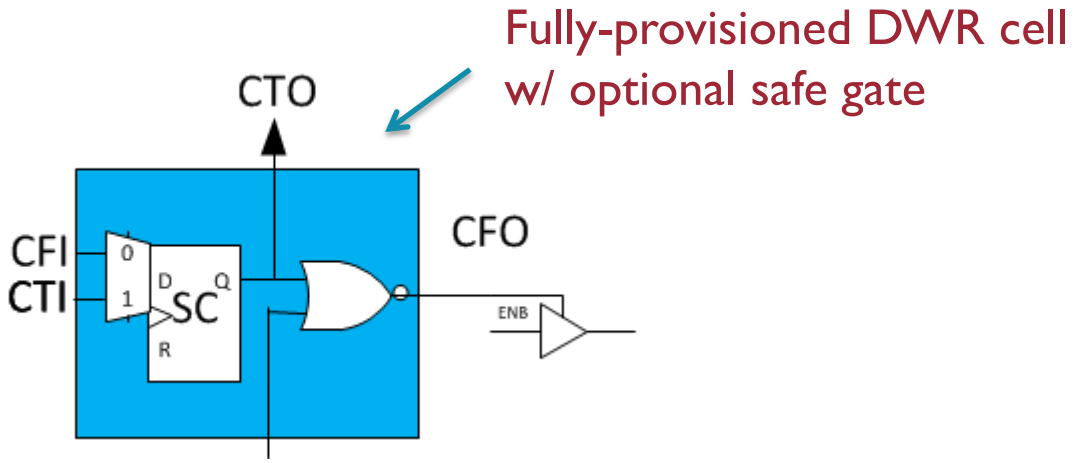


WC_SD2_CIU_UF_G

Safe Operation

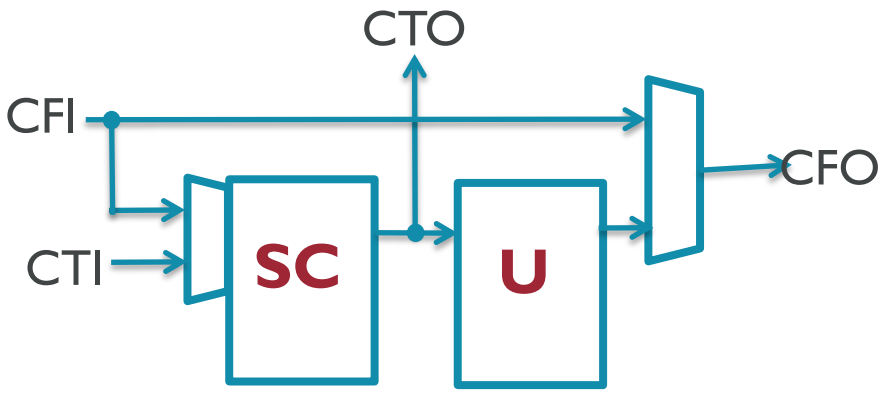
- Safe Operation pertains to moving into and out of a test mode safely
- In addition there are rules and recommendations around safe operation during test
  - The control signal on any three-state buffer on an output terminal of a die shall have a fully-provisioned wrapper cell.
  - Any three-state buffer on an output terminal of a die shall include control that is able to establish a persistent safe drive state on that output terminal during shift.

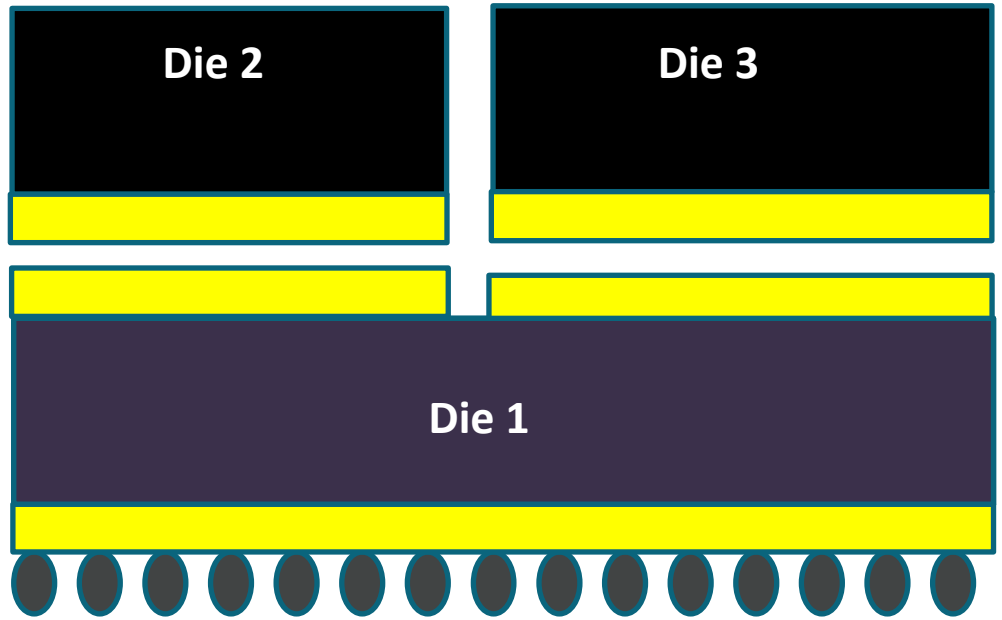




Safe Operation

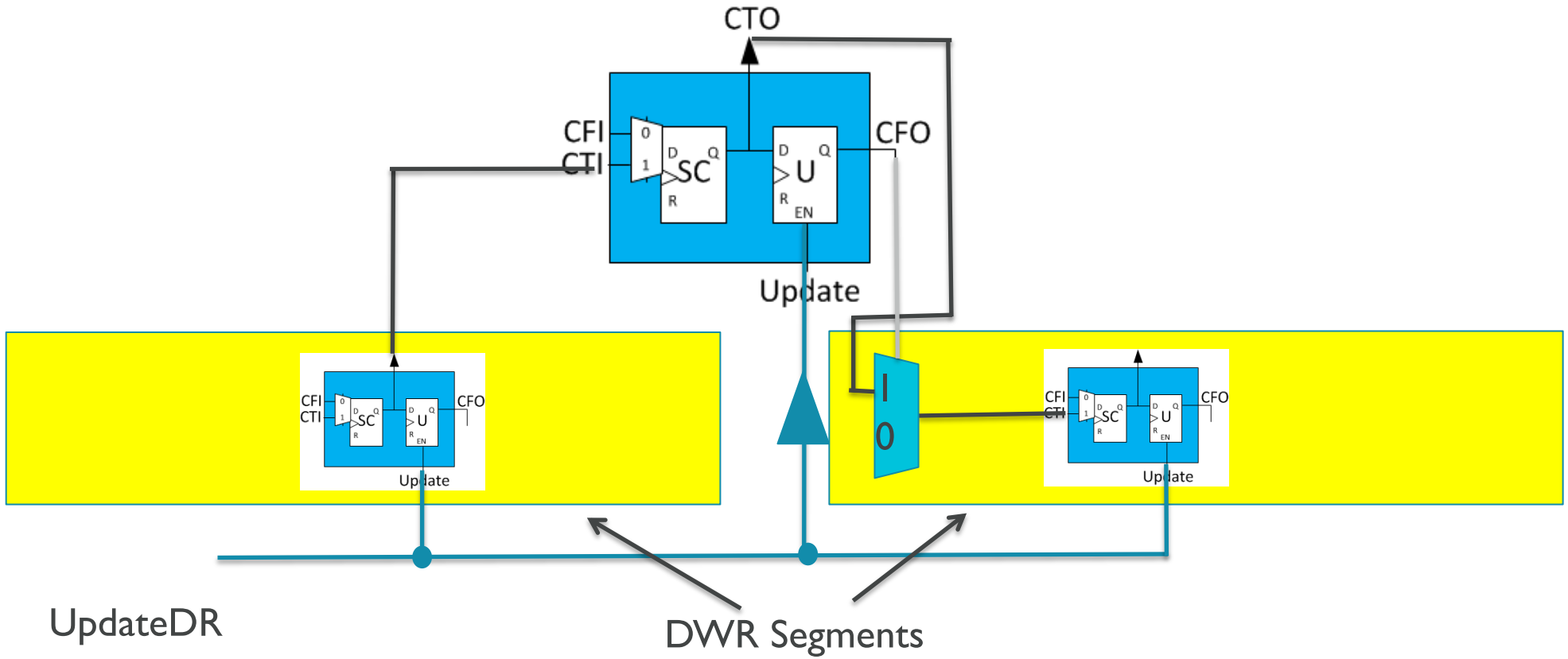
- The unselected segments can be in a test mode or functional mode
- If the DWR cell has update capability, the update must not occur if the segment in which the DWR cell resides is not selected



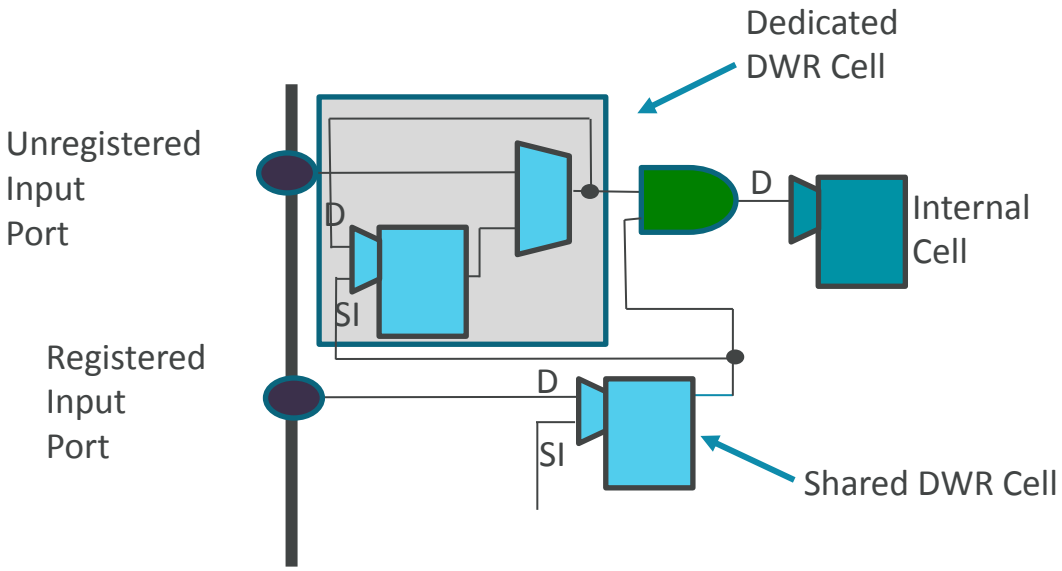


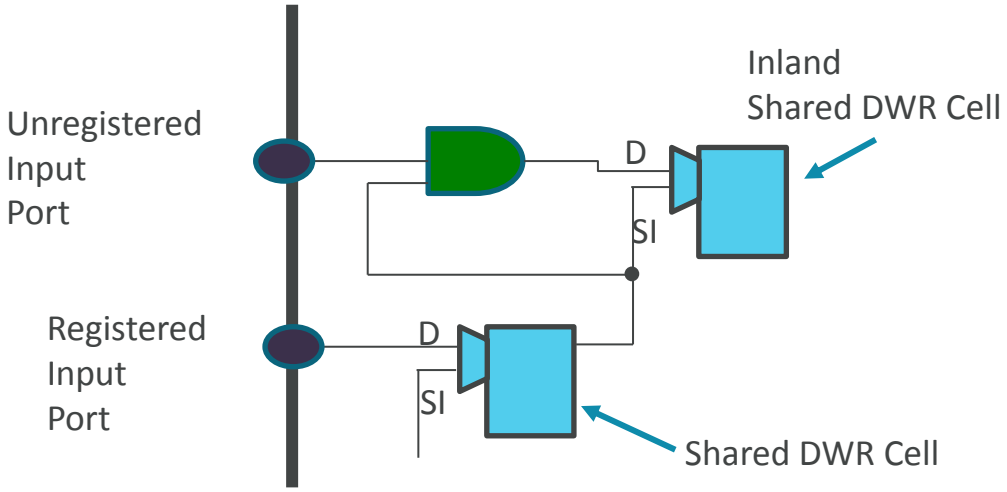
Safe Operation

- All die wrapper cells inline with other scan wrapper cells which control die wrapper segmentation multiplexers shall include a shift, capture, and update capability; the update shall be with delay such that the output of the inline segment select cell does not change until at least one half TCK cycle after UPDATE-DR.



Inland Wrapping



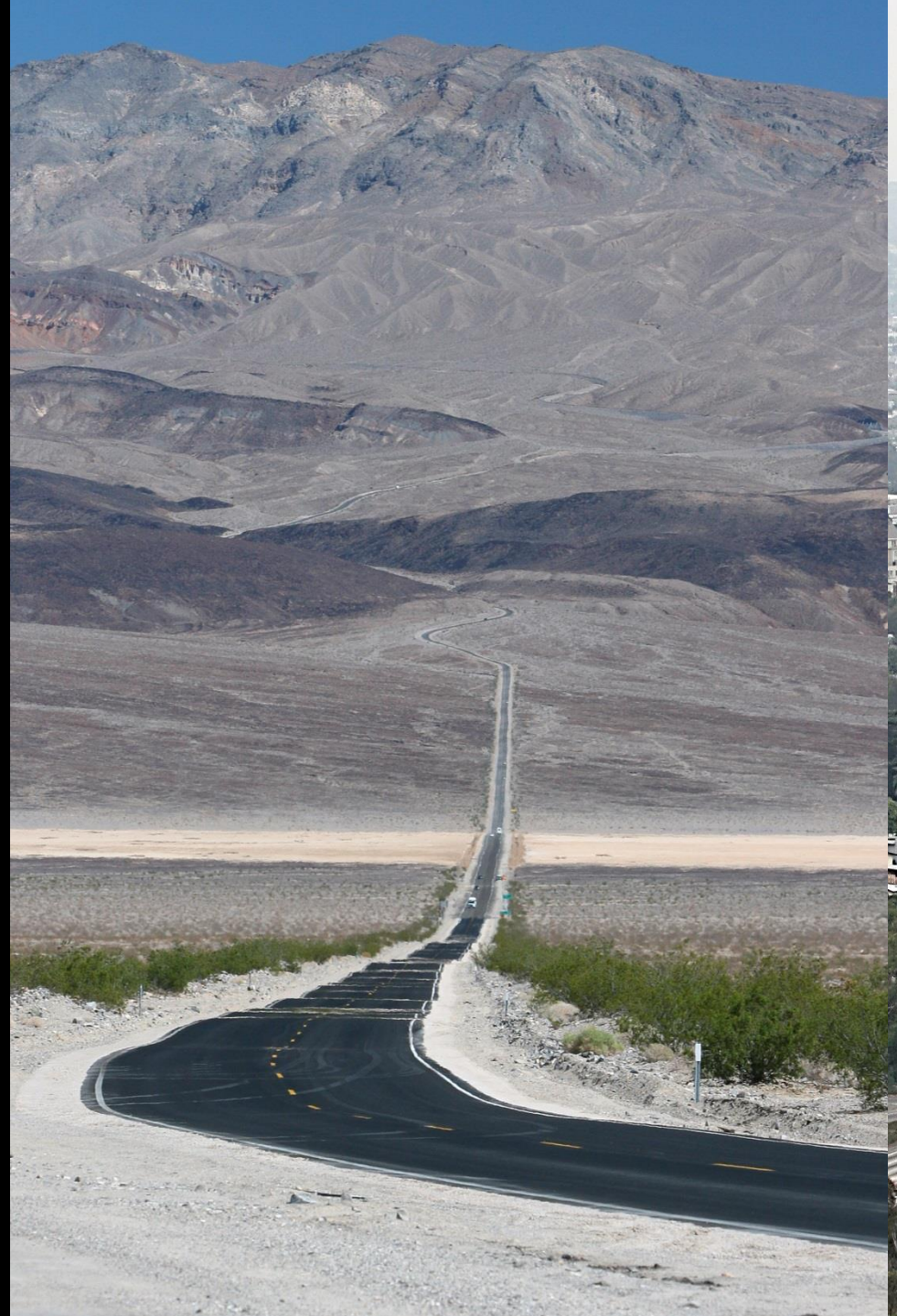


SYNOPSYS®
Silicon to Software™

3D-IC Test Standard IEEE P1838

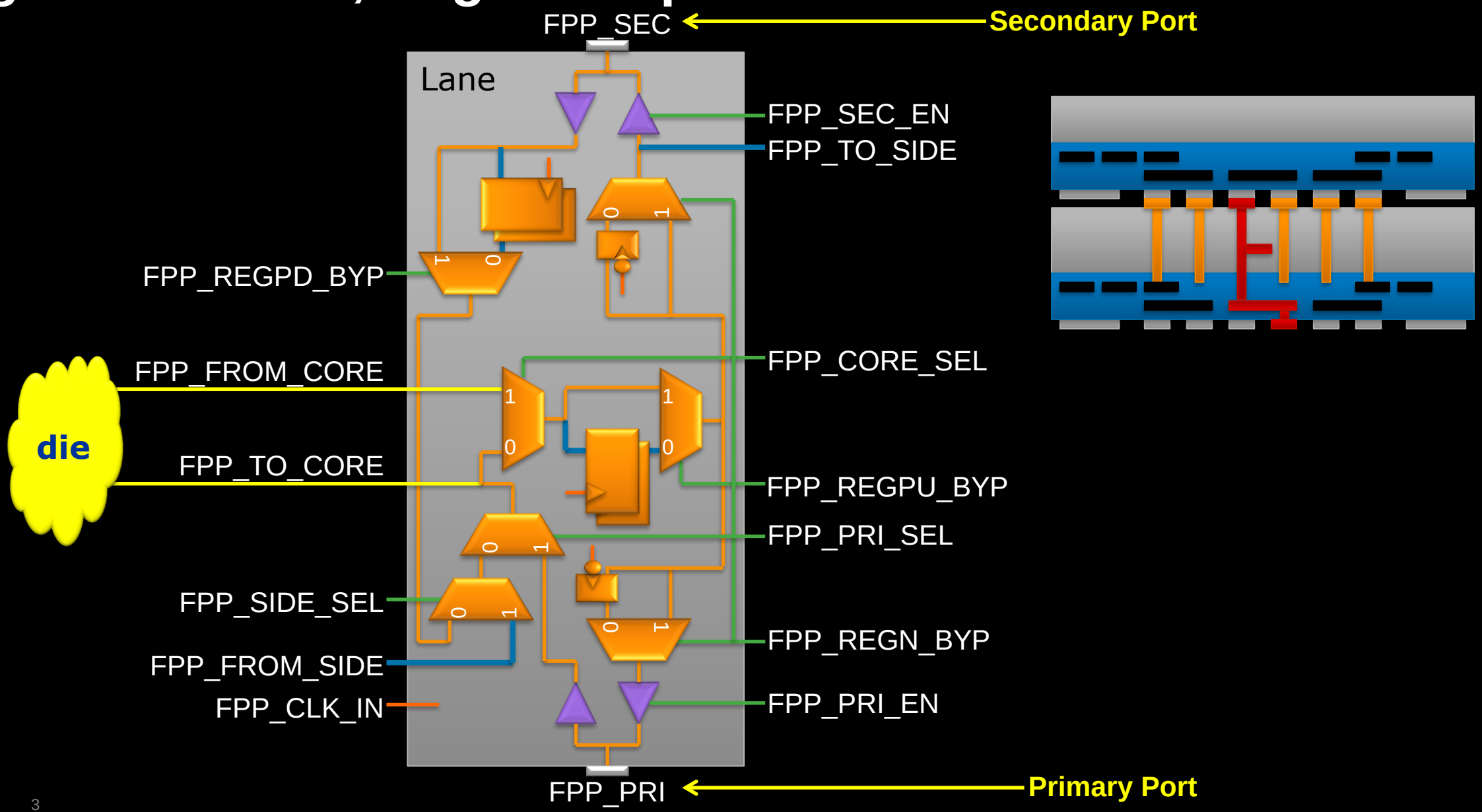
Flexible Parallel Port

Adam Cron, Principal Engineer
ITC 2016
Special Session 2

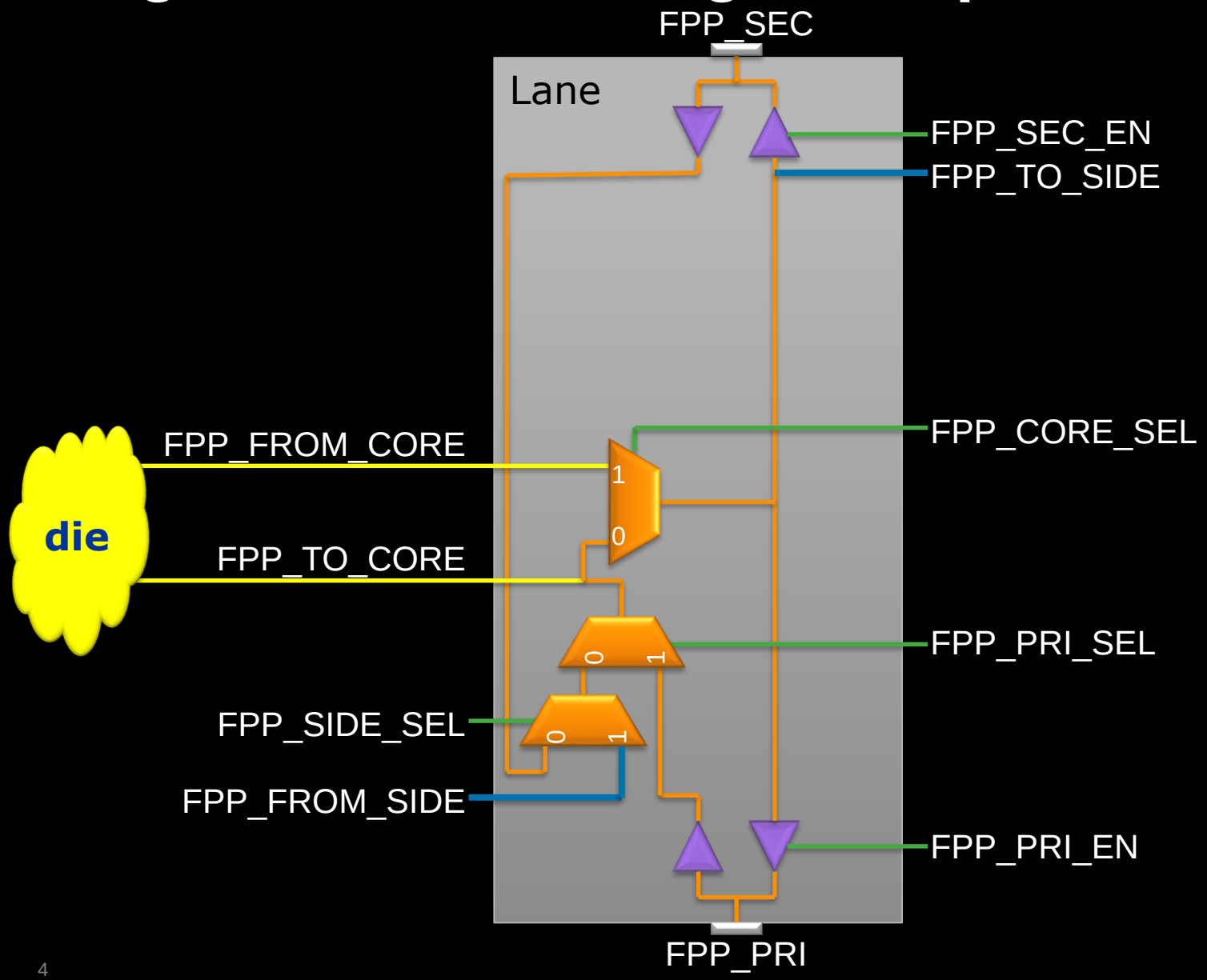



CAPITOL RECORDS

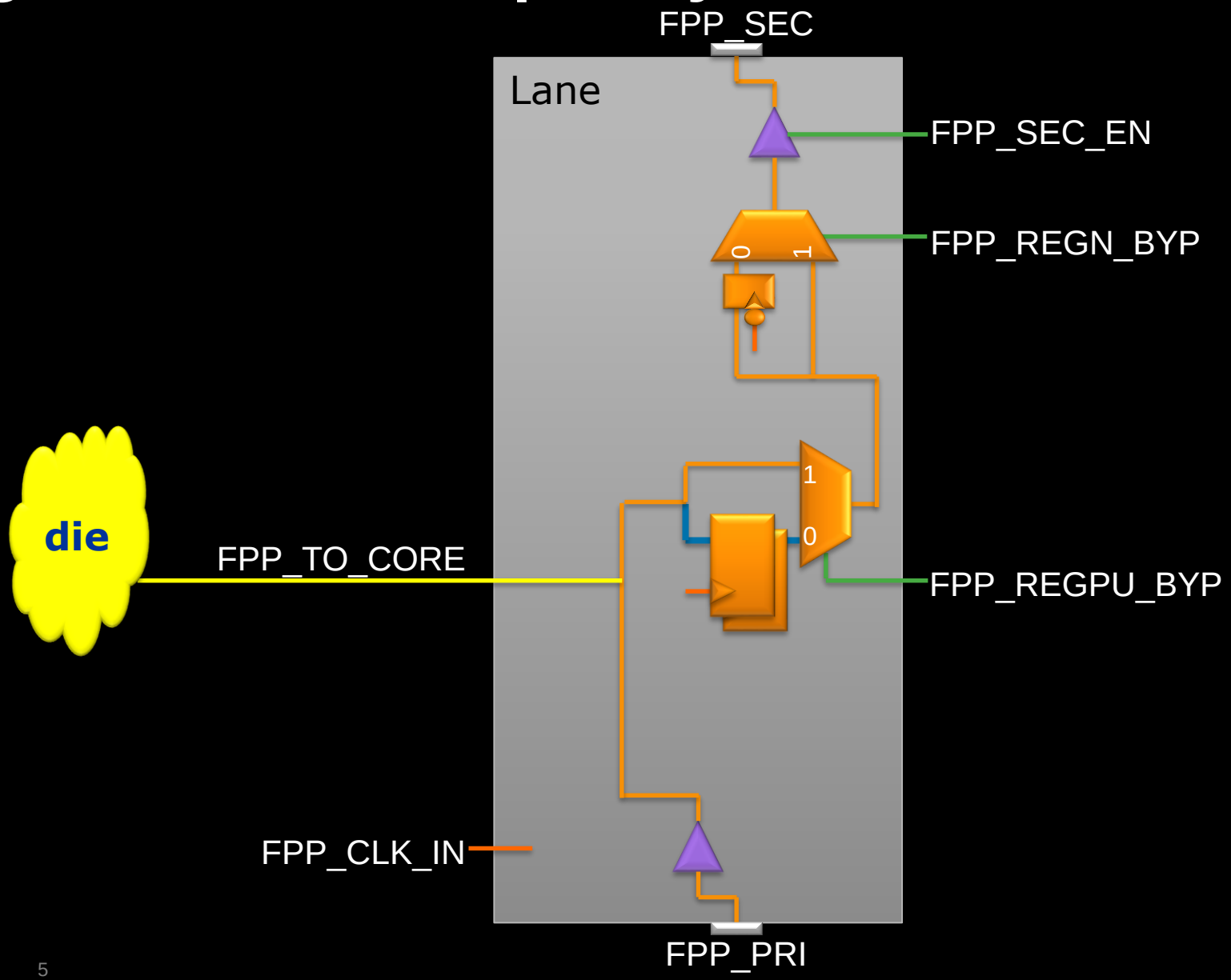
Registered Lane, Logical Implementation



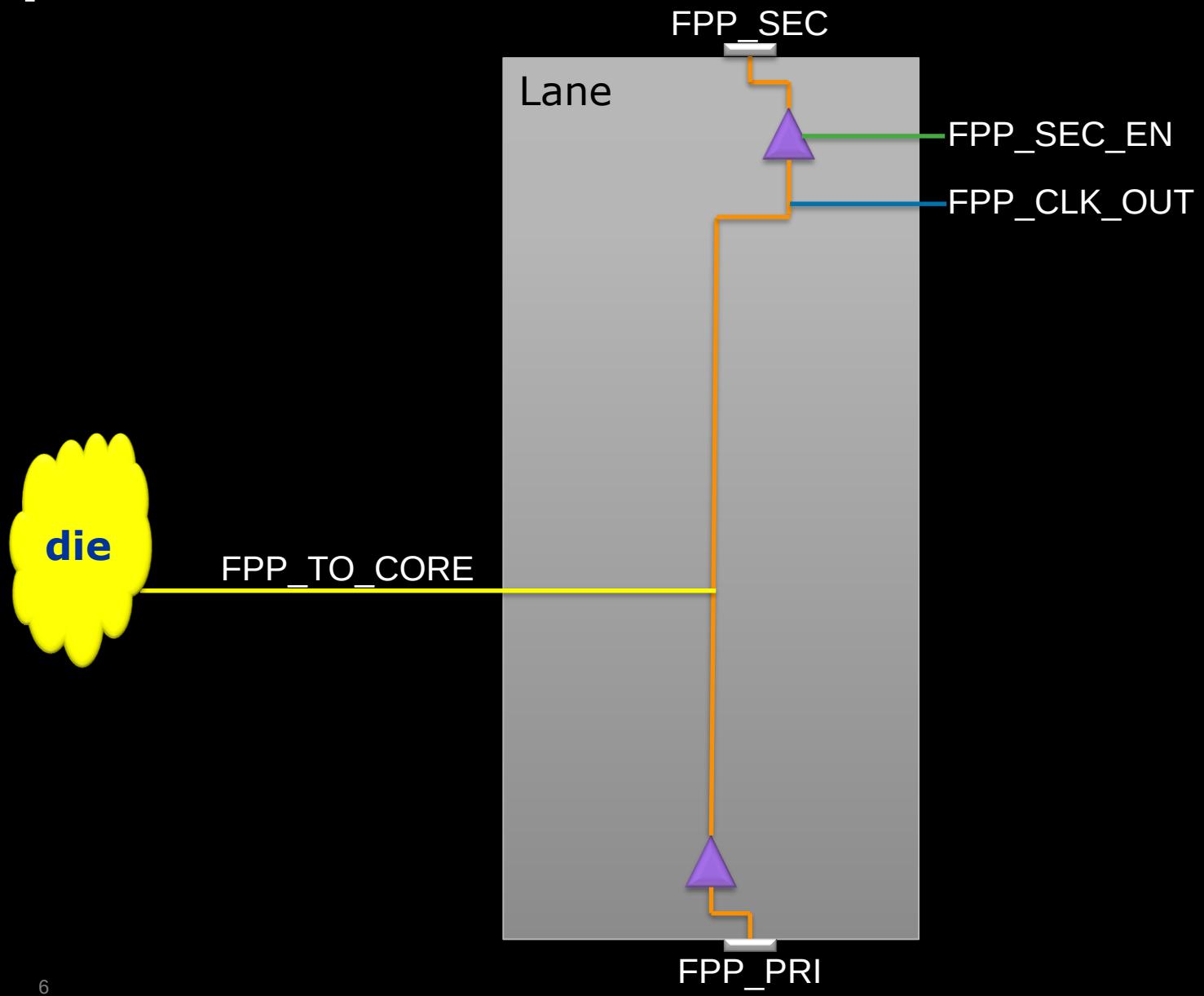
Non-Registered Lane, Logical Implementation



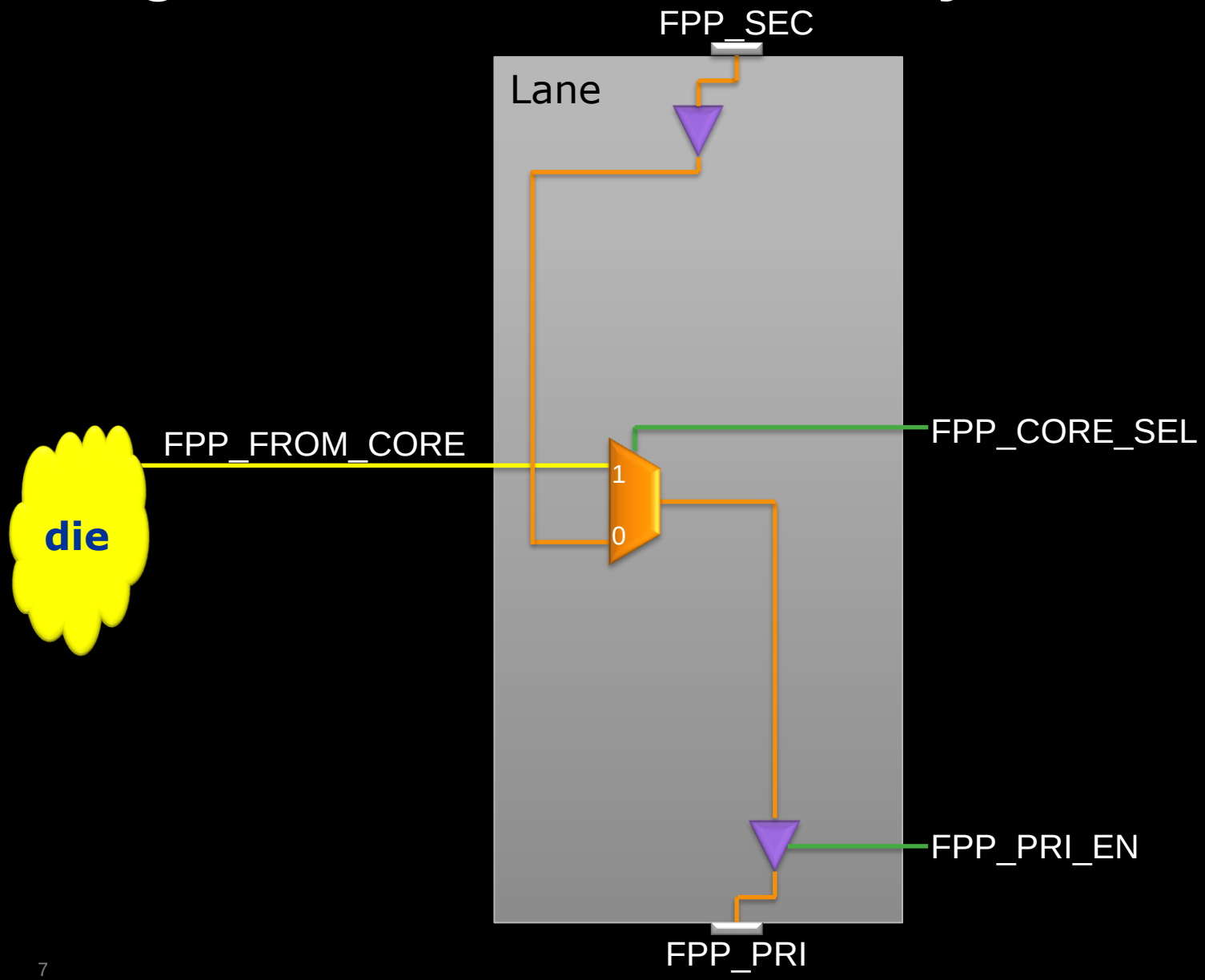
Registered Lane, Up Only



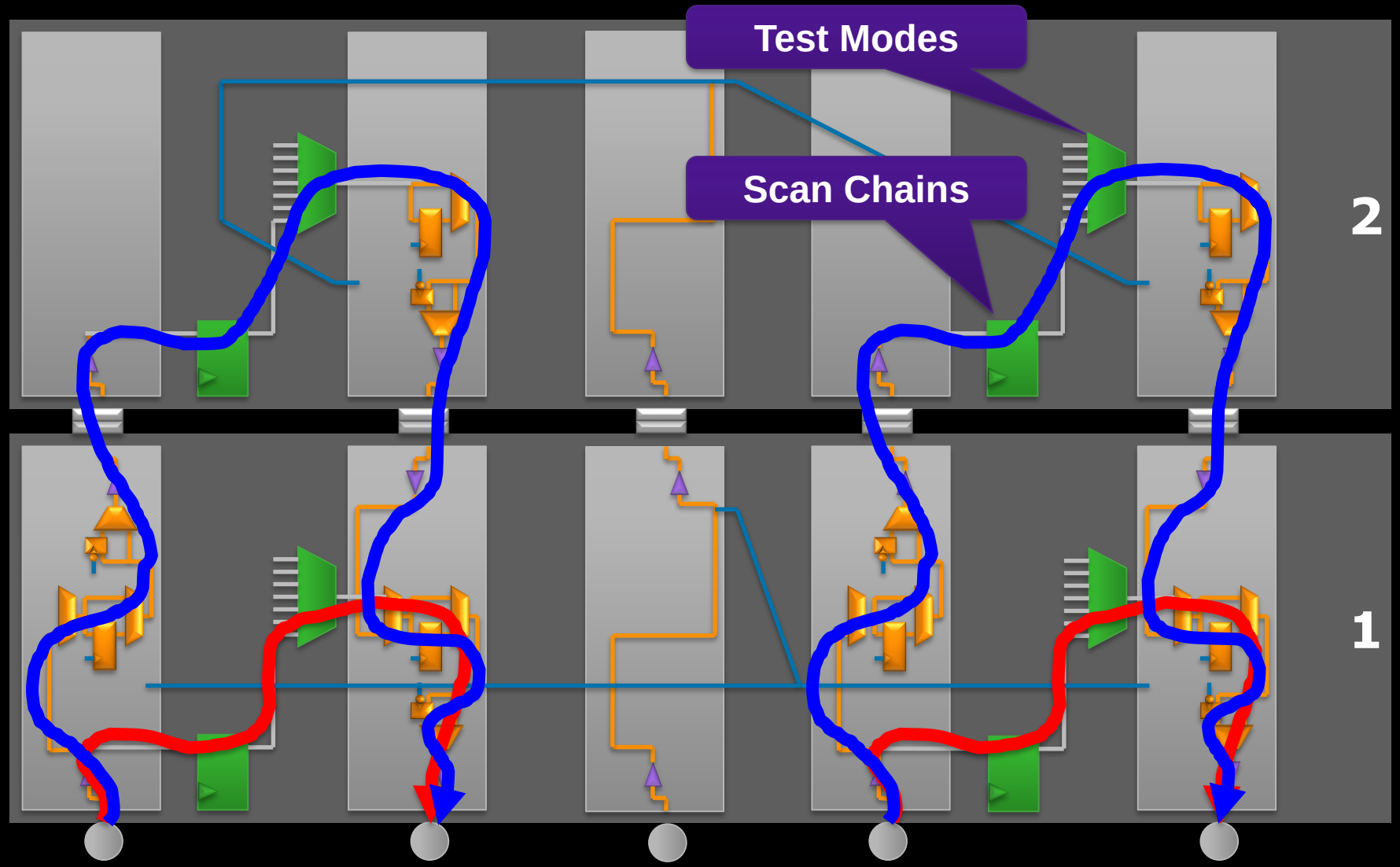
Simple Clock Path



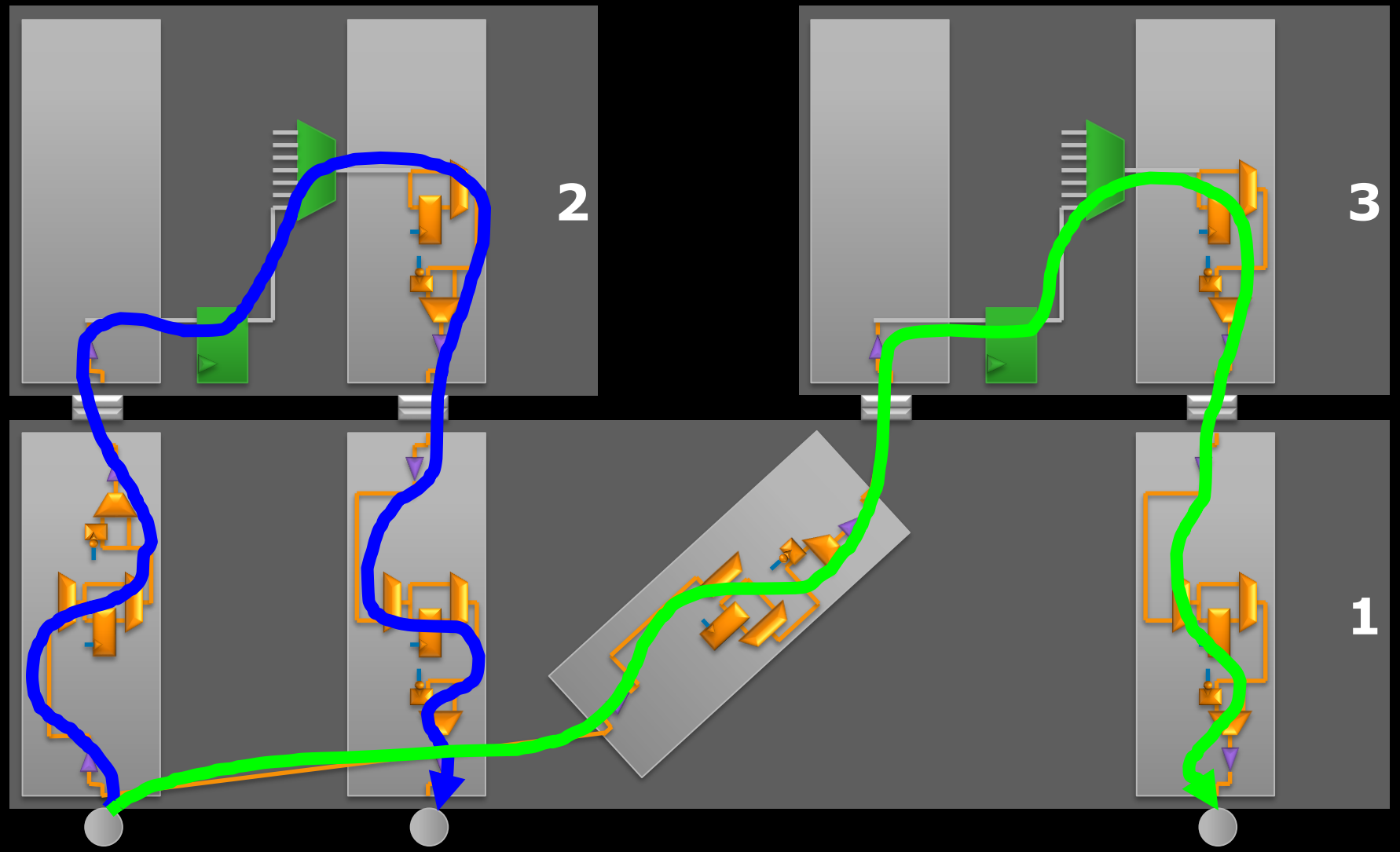
Non-Registered Lane, Down Only



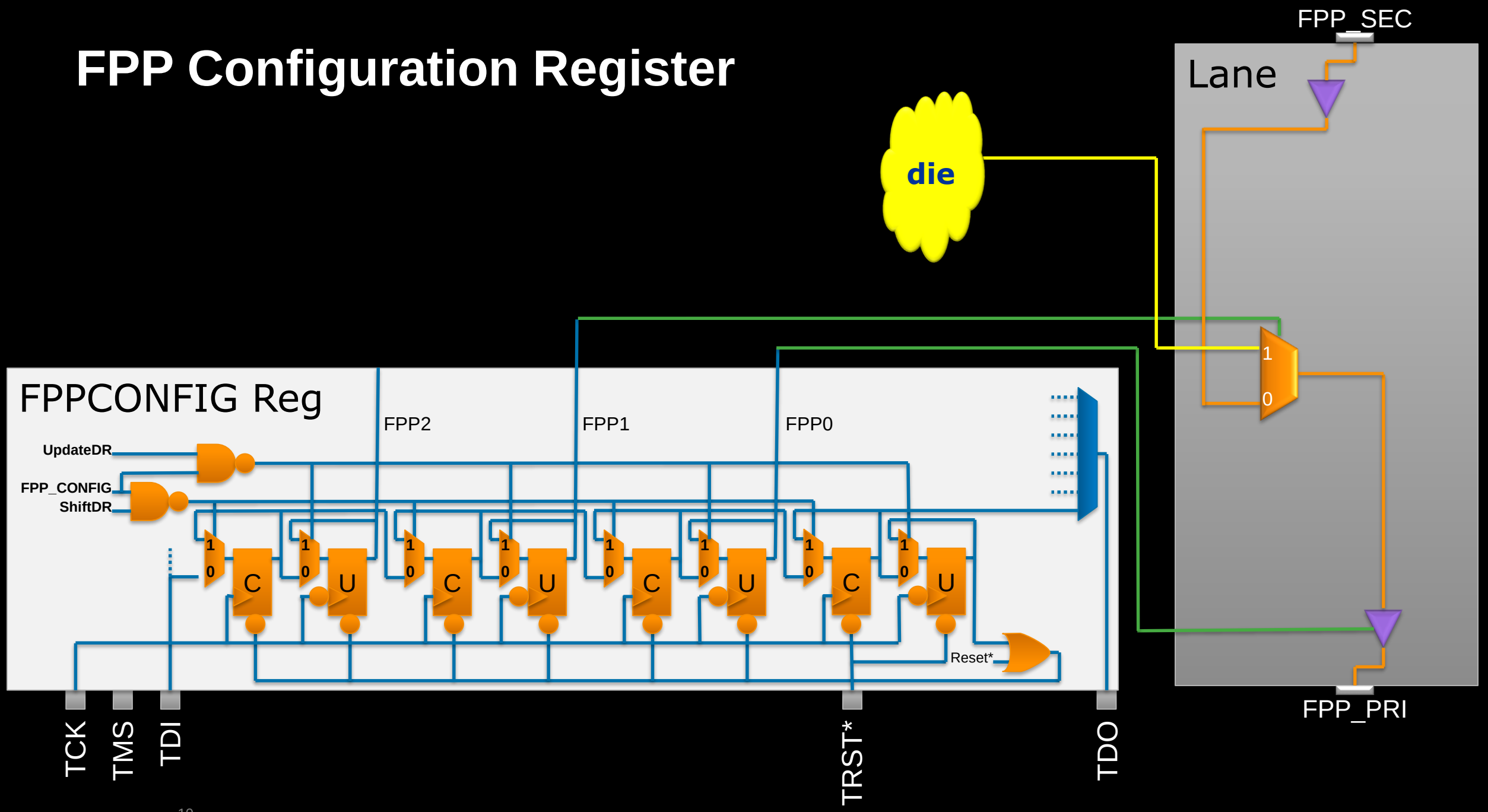
FPP Application



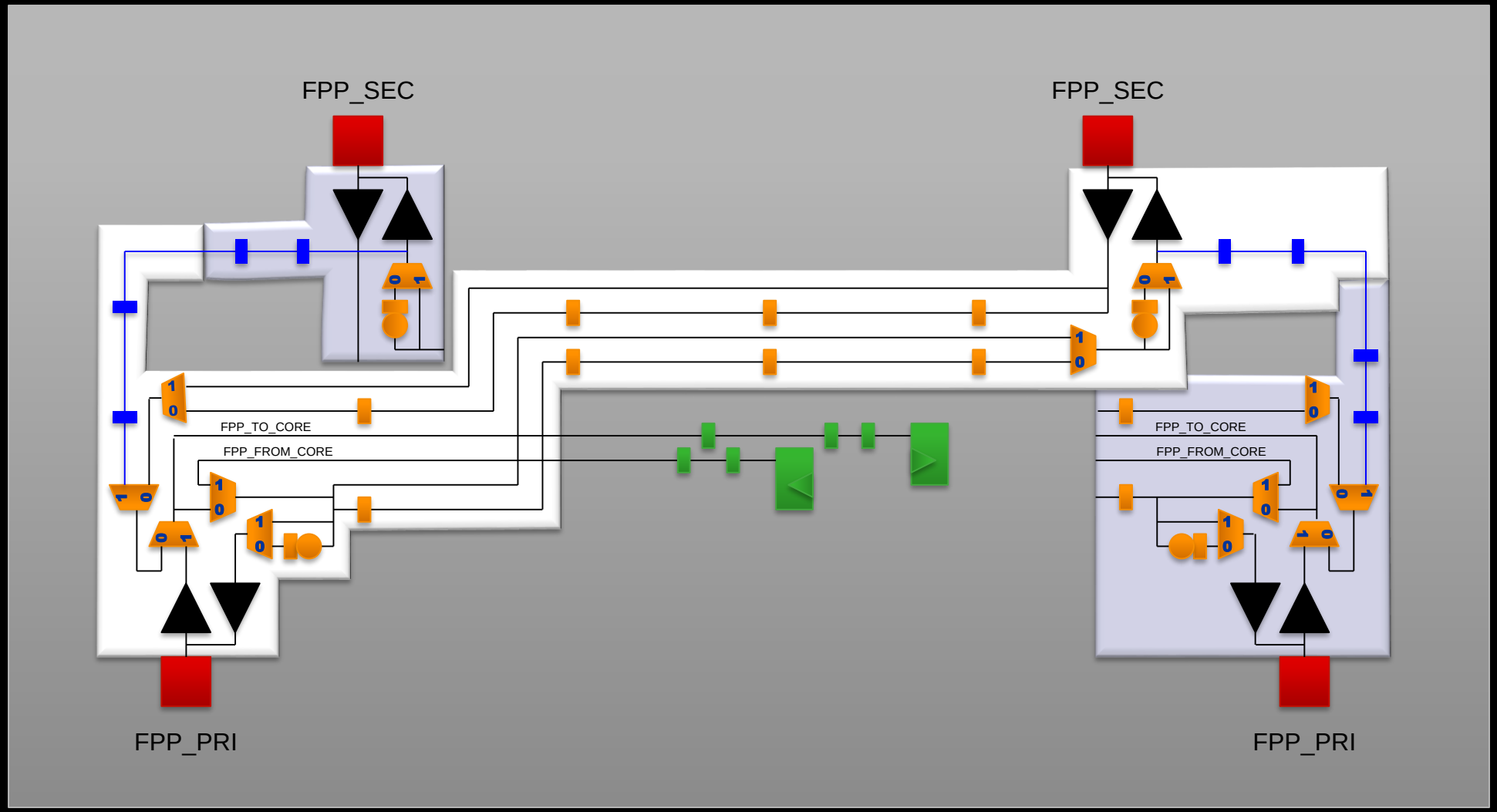
FPP Broadcast Application



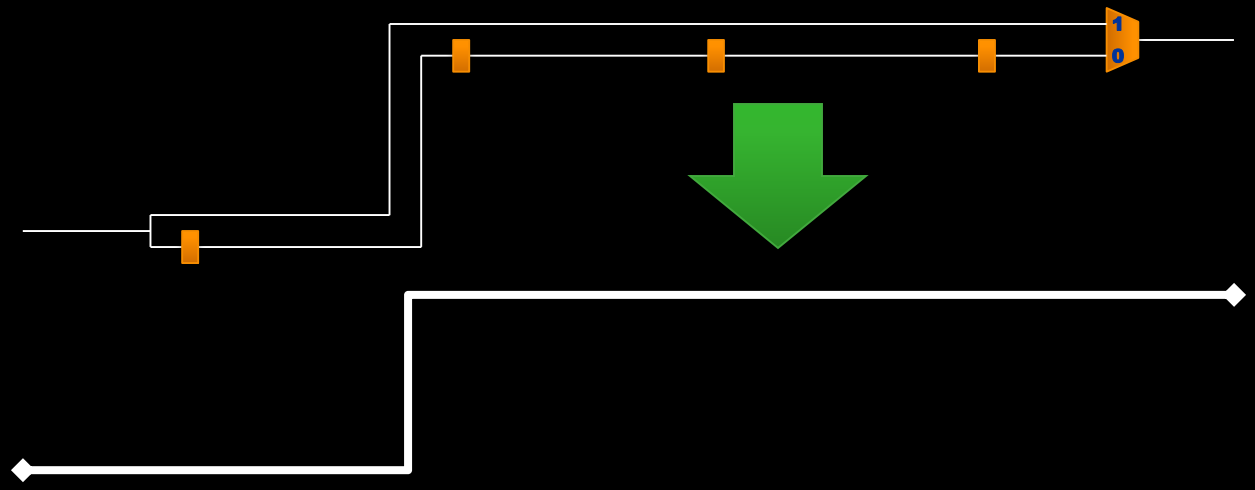
FPP Configuration Register



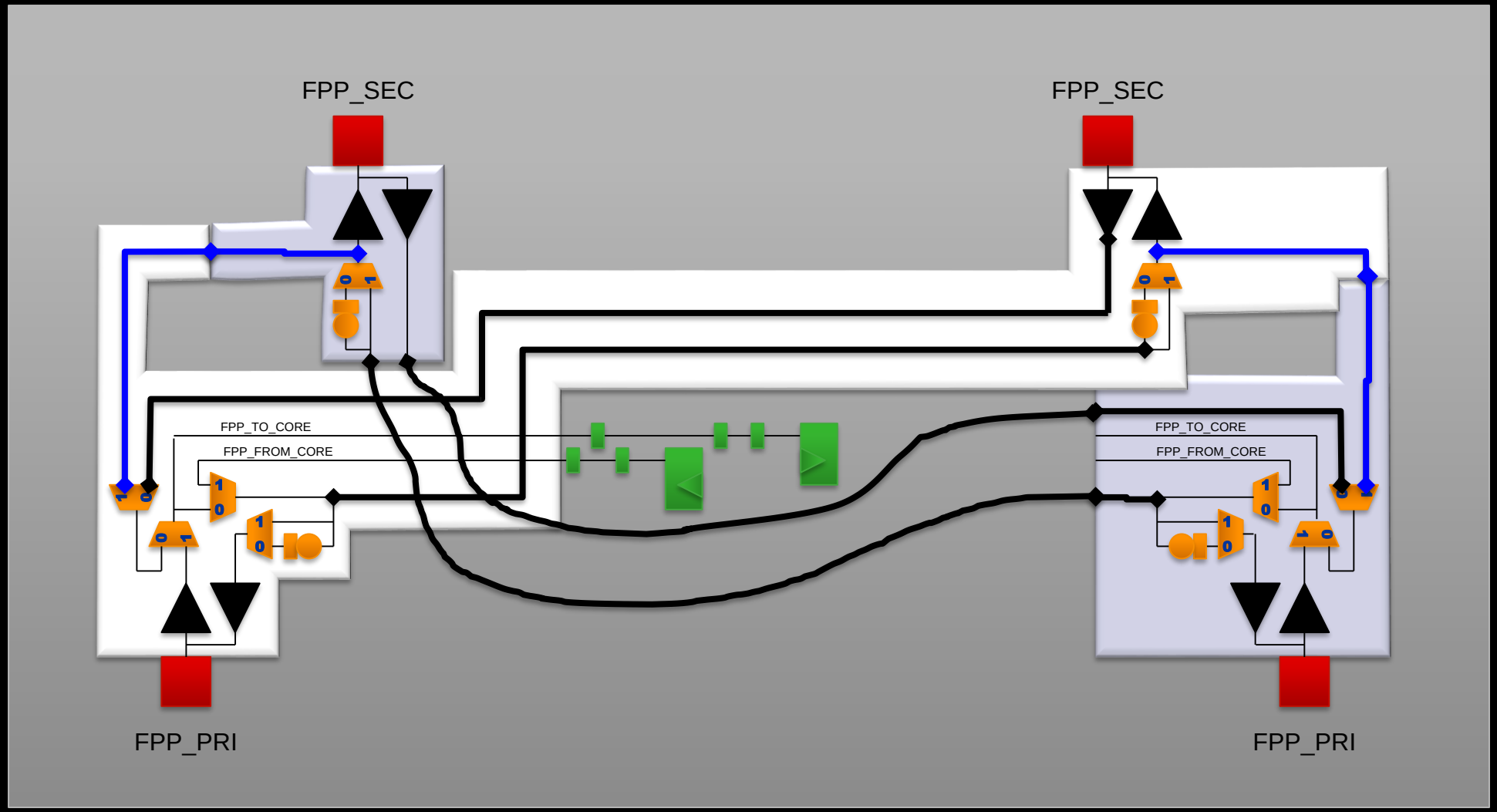
Bi-Directional “Layout”



Bypassable Pipe-Lined Entity



Bi-Directional “Layout” with Bypassable Elements



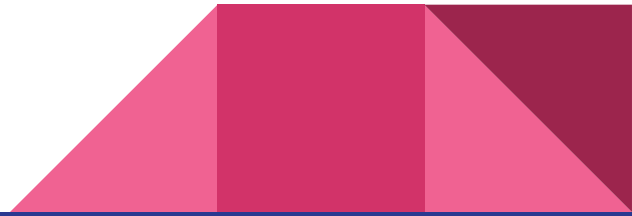
Thank You

P1838 Test Description Language(s)

Sandeep Bhatia
Google

P1838 Elements

- Serial Control
- Die Wrapper Register
- Flexible Parallel Port



Language Options

Extend and Reuse existing languages

- 1149.1 **BSDL** (Serial Control)
- 1450.6 **CTL** (Die Wrapper Register, FPP)
- 1687 **ICL** (FPP)

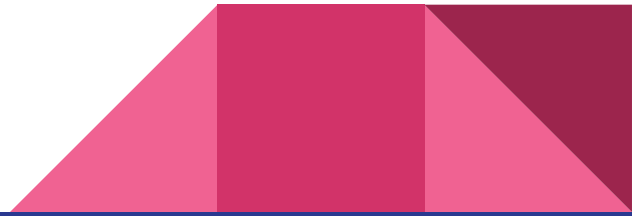
Or use a **New** Language

Challenges with Reuse

- Language **Extensions**
- **Mix** of languages to describe the complete standard
- **Complexity** - irrelevant language constructs

Test Data Languages

- **Key-Value** tuples
- **Bundling** of data
- **Nesting/Hierarchy** of data



CTL

```
Environment {
  CTL config1 {
    Internal {
      Ck { DataType
            TestClock {
              AssumedInitialState
              ForceDown;
            }
        }
      }
      SI { DataType
           TestData ScanDataIn;
      }
  }
}
```

CTL

Key Value tuples

```
Environment {
  CTL config1 {
    Internal {
      Ck { DataType
           TestClock {
             AssumedInitialState
             ForceDown;
           }
        }
      }
      SI { DataType
           TestData ScanDataIn;
      }
  }
}
```

BSDL

```
Attribute TAP_SCAN_IN of TDI :
  signal is True;
Attribute TAP_SCAN_OUT of TDO :
  signal is True;
Attribute TAP_SCAN_MODE of TMS :
  signal is True;
Attribute TAP_SCAN_CLOCK of TCK :
  signal is (50.0e6, BOTH);
Attribute INSTRUCTION_LENGTH of
  foo : entity is 2;
Attribute INSTRUCTION_OPCODE of
  foo: entity is “BYPASS (11), “&&
Attribute INSTRUCTION_OPCODE of
  foo: entity is “EXTEST (00), “&&
```

BSDL

Key Value tuples

```
Attribute TAP_SCAN_IN of TDI :
  signal is True;
Attribute TAP_SCAN_OUT of TDO :
  signal is True;
Attribute TAP_SCAN_MODE of TMS :
  signal is True;
Attribute TAP_SCAN_CLOCK of TCK :
  signal is (50.0e6, BOTH);
Attribute INSTRUCTION_LENGTH of
  foo : entity is 2;
Attribute INSTRUCTION_OPCODE of
  foo: entity is "BYPASS (11), "&&
Attribute INSTRUCTION_OPCODE of
  foo: entity is "EXTEST (00), "&&
```

ICL

```
Module SIB {
  ScanInPort si;
  ScanOutPort so {
    Source SIB;
    LaunchEdge Falling;
  }
  SelectPort en;
  ScanRegister sr {
    ScanInSource mux1;
    CaptureSource 1’b0;
    ResetValue 1’b0;
  }
}
```

ICL

Key Value tuples

```
Module SIB {
  ScanInPort si;
  ScanOutPort so {
    Source SIB;
    LaunchEdge Falling;
  }
  SelectPort en;
  ScanRegister sr {
    ScanInSource mux1;
    CaptureSource 1'b0;
    ResetValue 1'b0;
  }
}
```

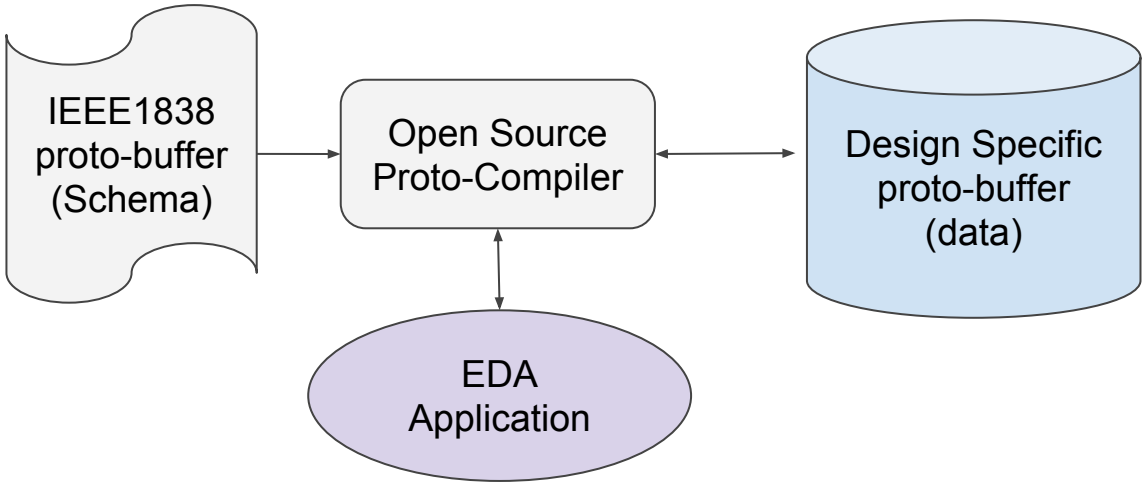
Protocol Buffers

- Structured data representation
- `Key:Value` tuple format with **data bundling** and **nesting**
- **Schema** definition for specific format
- Open source **Parser** with automatic **API** generation to access data
- Multiple platforms
  - *OSX, Linux, Windows*
- Multiple Programming language support
  - *Python, C++, Java, Go, Ruby, C#, Javanano*

[1] https://developers.google.com/protocol-buffers/

[2] Licensed under 3BSD License (https://github.com/google/protobuf/blob/master/LICENSE)

IEEE 1838 EDA Integration



FPP Proto Buffer Schema

```
message Fpp {
  int32 channel_count       = 1;
  repeated Channel channel = 2;
}
```

FPP Proto Buffer Schema

```
message Fpp {
  int32 channel_count       = 1;
  repeated Channel channel = 2;
}

message Channel {
  string name          = 1;
  ChannelType type     = 2;
  string clock         = 3;
  int32 lane_count     = 4;
  repeated Lane lane = 5;
}
```

FPP Proto Buffer Schema

```
message Fpp {
  int32 channel_count       = 1;
  repeated Channel channel = 2;
}

message Channel {
  string name          = 1;
  ChannelType type     = 2;
  string clock         = 3;
  int32 lane_count     = 4;
  repeated Lane lane = 5;
}
```

```
enum ChannelType {
  Registered   = 0;
  Unregistered = 1;
}
```

FPP Proto Buffer Schema

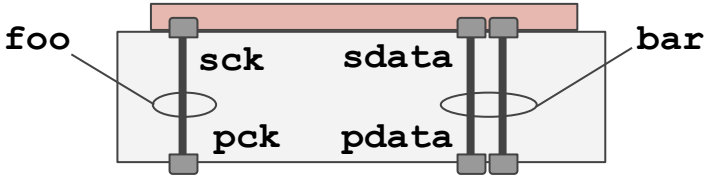
```
message Fpp {
  int32 channel_count       = 1;
  repeated Channel channel = 2;
}

message Channel {
  string name          = 1;
  ChannelType type     = 2;
  string clock         = 3;
  int32 lane_count     = 4;
  repeated Lane lane = 5;
}
```

```
enum ChannelType {
  Registered   = 0;
  Unregistered = 1;
}

message Lane {
  string name       = 1;
  string fpp_pri    = 2;
  string fpp_sec    = 3;
  string from_side = 4;
  string to_side    = 5;
}
```

FPP Proto Buffer Data



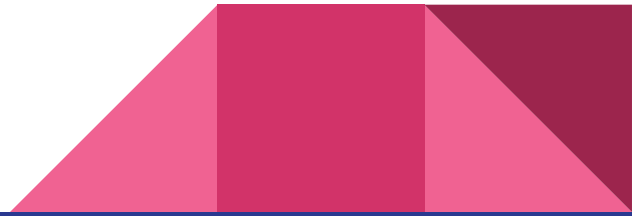

```
Fpp {
  channel_count: 2

  channel {
    name: "foo"
    type: Unregistered
    lane_count: 1
    lane {
      name: "lane_1"
      fpp_pri: "pck"
      fpp_sec: "sck"
    }
  }
```

```
  channel {
    name: "bar"
    type: Registered
    clock: "foo.lane_1"
    lane_count: 2
    lane {
      name: "lane_1"
      fpp_pri: "pdata[0]"
      fpp_sec: "sdata[0]"
    }
    lane {
      name: "lane_2"
      fpp_pri: "pdata[1]"
      fpp_sec: "sdata[1]"
    }
  }
}
```

Current Status

- Evaluating different options
  - Proto Buffer Schema design
  - Existing Language suitability, extensions, complexity
- Unify



Questions