



JTAG

(IEEE 1149.1/P1149.4)

Tutorial

Intermediate





Agenda

- The Boundary-Scan Architecture (35 minutes)
- The Boundary-Scan Description Language (BSDL) (20 minutes)
- Other Supporting Data Formats (15 minutes)
 - HSDL, SVF
- Future Directions (10 minutes)
 - Std 1149.1 Revision in Progress
 - P1149.4 Mixed Signal Test Bus
- Q & A (10 minutes)



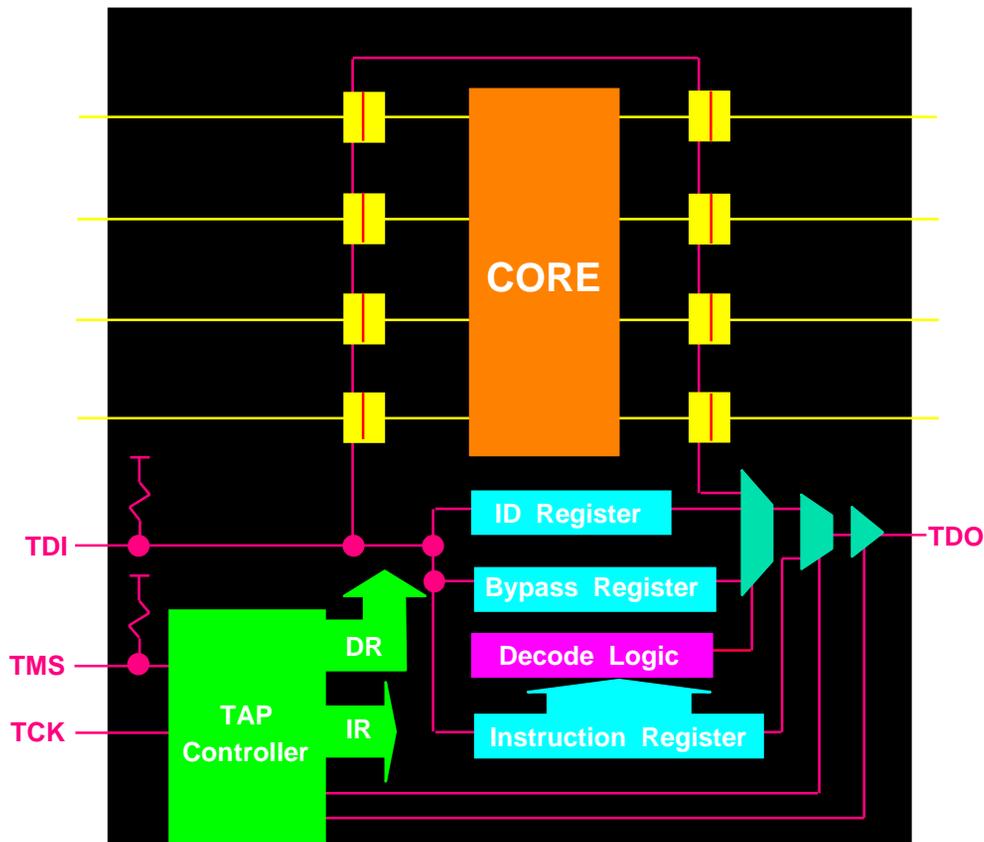


The Boundary-Scan Architecture





The Test Logic Architecture

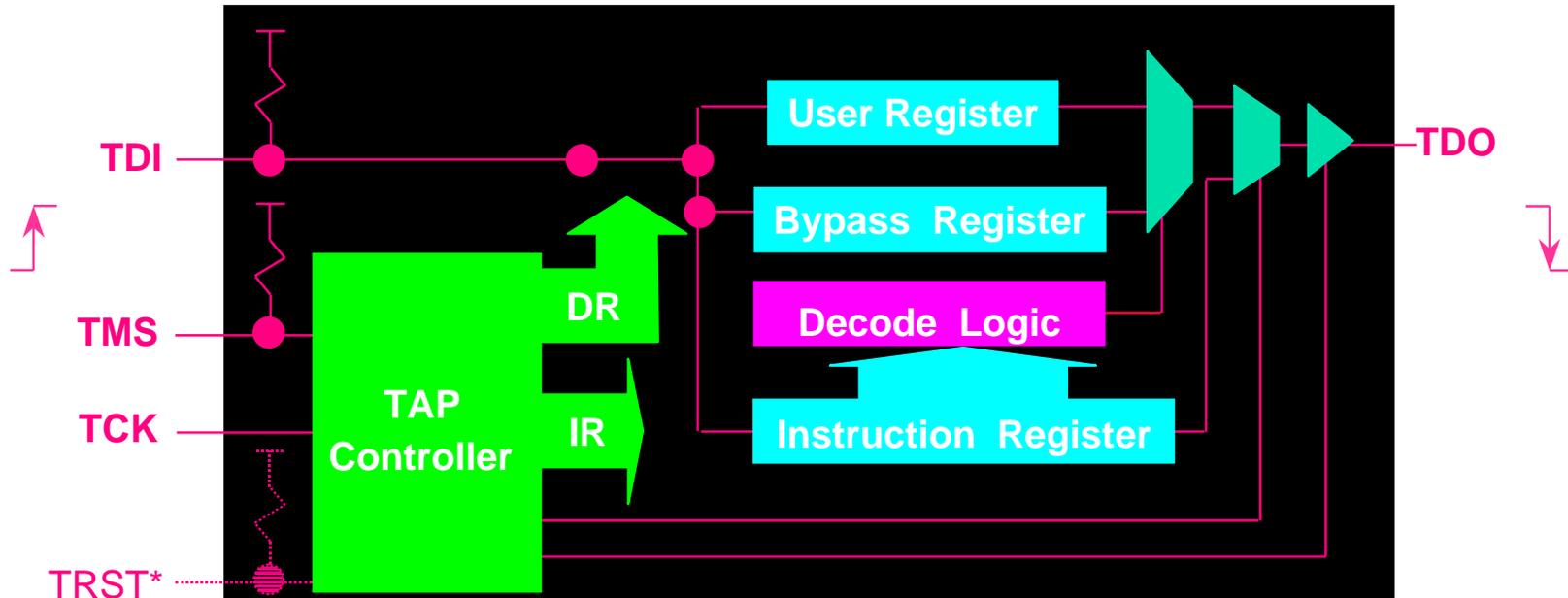


- The REQUIRED elements of the test logic architecture are:
 - Test Access Port (TAP)
 - TAP Controller
 - One Instruction Register
 - A Set of Data Registers
- All such elements are REQUIRED to be dedicated test logic (not used for any other purpose) with the exception of user-defined data registers





The Test Access Port

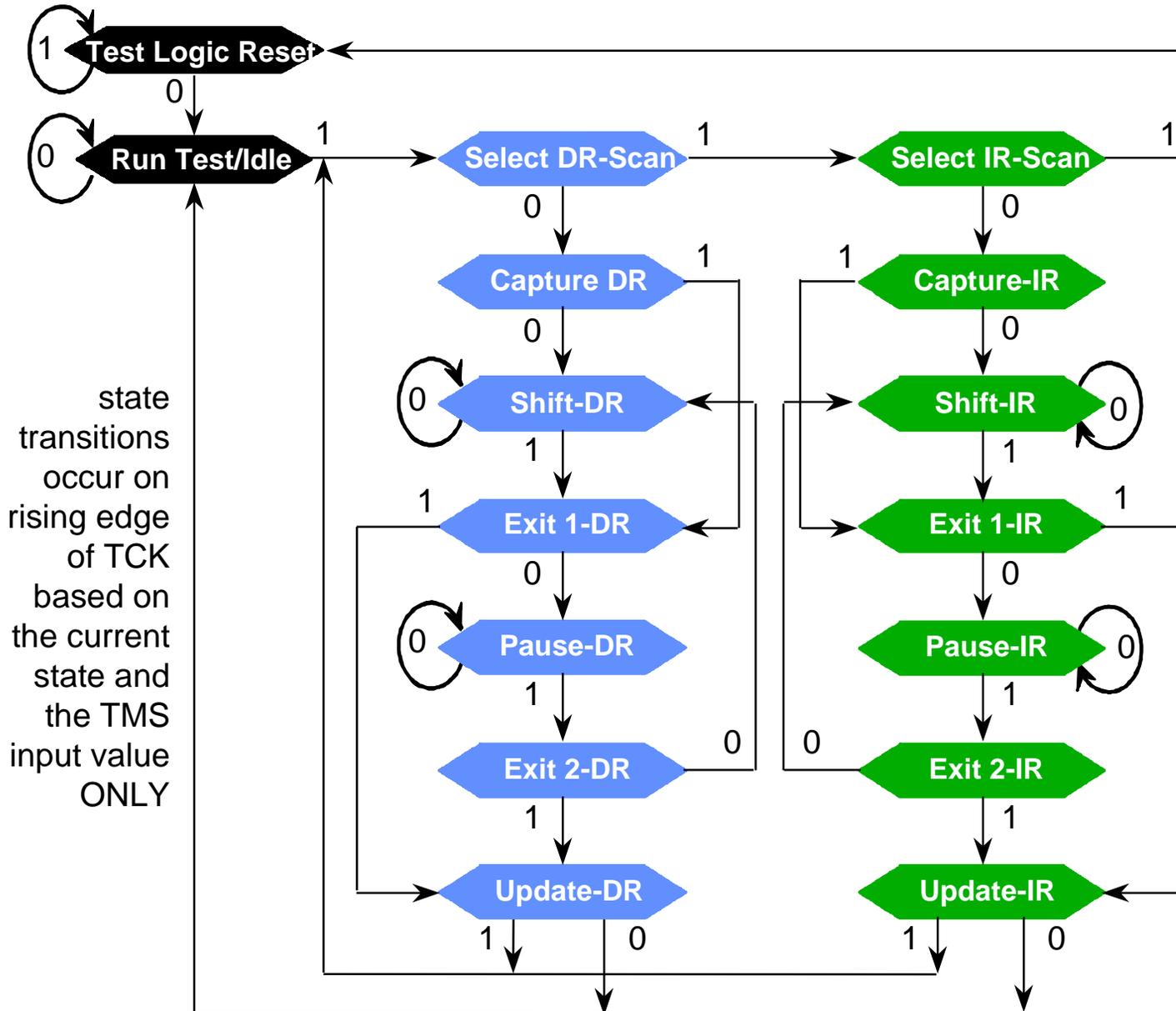


- 4 wire TAP interface REQUIRED
- either power-up reset or 5th wire, TRST*, is REQUIRED
- all TAP pins are REQUIRED to be dedicated (not used for any other purpose)
- pullups REQUIRED at TDI and TMS (also, TRST*, if implemented)
- all inputs (e.g., TDI) are REQUIRED to be sampled on TCK rising,
all outputs (e.g. TDO) are REQUIRED to be propagated on TCK falling





The TAP Controller

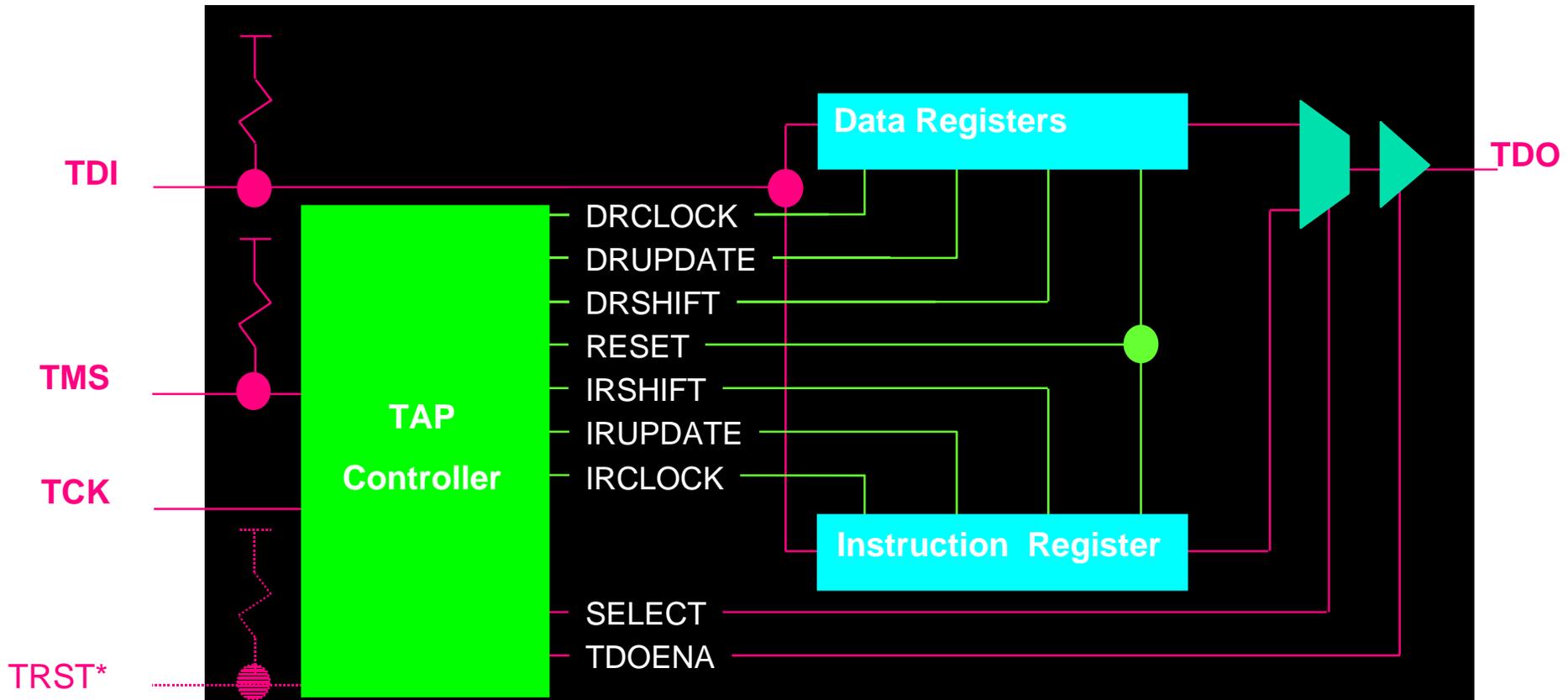


- The TAP controller is REQUIRED to conform exactly to this state diagram
- The instruction scan sequence is REQUIRED to access only the single instruction register
- The data scan sequence is REQUIRED to access only the data register selected by the current instruction
- TDO is REQUIRED to be active only in the Shift-IR and Shift-DR states





The TAP Controller Outputs



- IR scan and update clocks and shift/capture select
- DR scan and update clocks and shift/capture select
- TDO enable and TDO instruction/data select





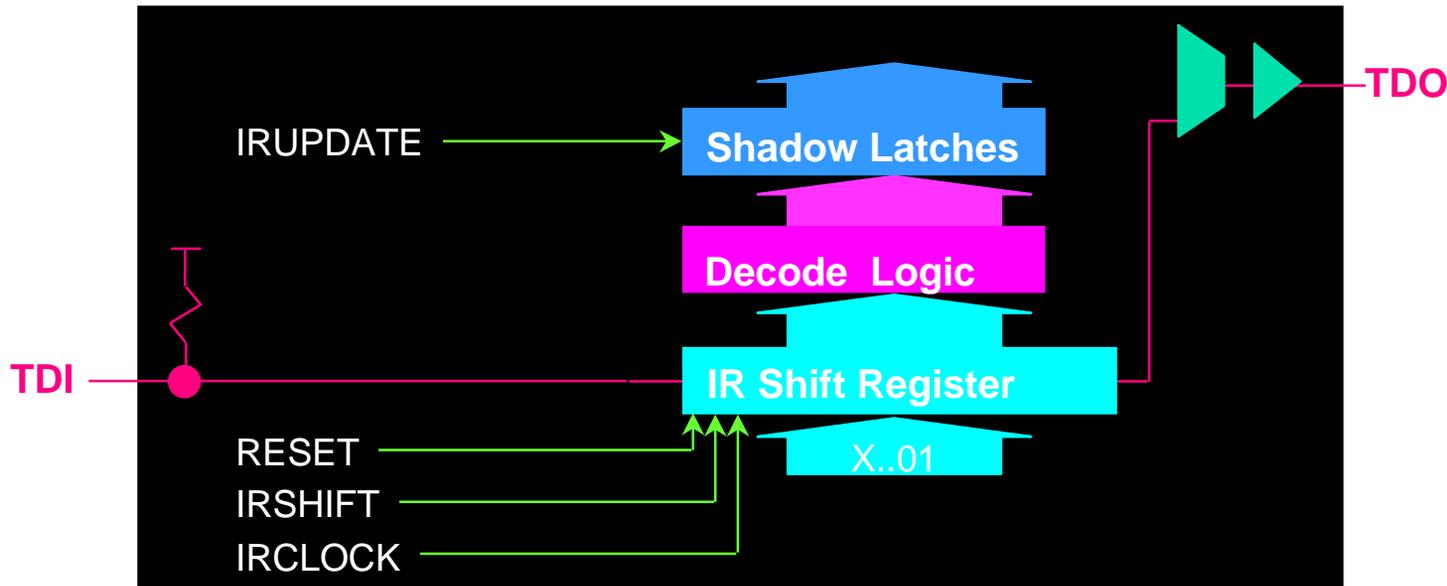
Register Summary

| Register (<u>REQUIRED</u> / Optional) | Length | Capture Value | Selected by Instruction(s) |
|--|----------------|----------------|--|
| <u>INSTRUCTION</u> | at least two | X..01 | n/a |
| <u>BOUNDARY</u> | user-specified | user-specified | Extest Sample/Preload Intest Clamp HighZ |
| <u>BYPASS</u> | exactly one | 0 | Bypass |
| Device ID | exactly 32 | X..1 | Idcode Usercode |
| user-defined | user-specified | user-specified | user-specified |





The Instruction Register

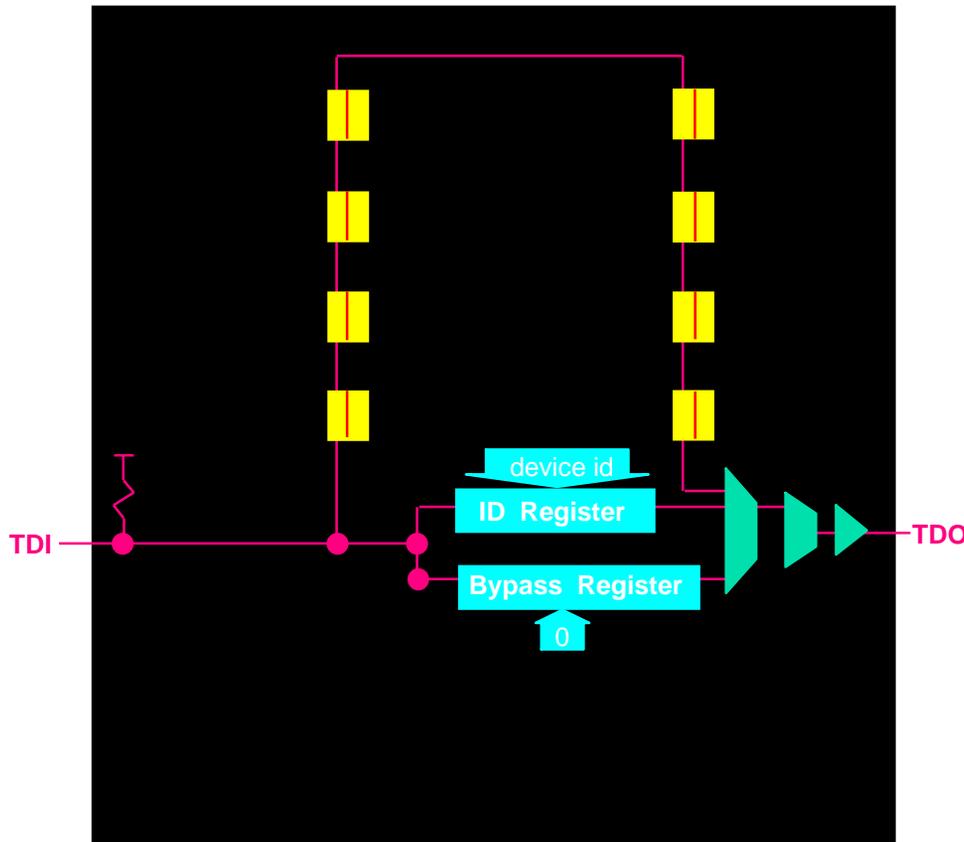


- The instruction register is REQUIRED to be at least 2 bits in length
- The instruction register is REQUIRED to capture the binary value 01 in its 2 least-significant bits on the rising edge of TCK in Capture-IR (other bits may capture user-specified data)
- The instruction register is REQUIRED to have a latched parallel output such that a new instruction only takes effect after shifting is complete (on falling edge of TCK in Update-IR)
- The instruction register is REQUIRED to reset to Idcode (if implemented), otherwise to Bypass





The Test Data Registers

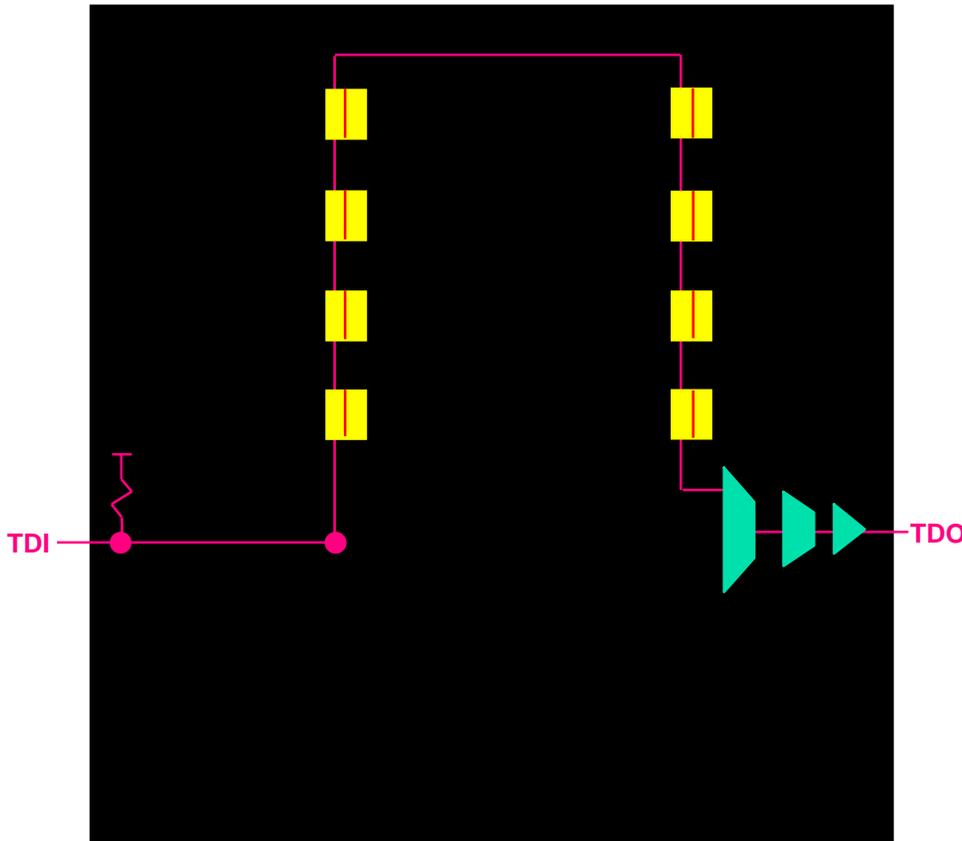


- Only 2 test data registers are **REQUIRED**:
 - The boundary-scan register - the serial concatenation of all boundary-scan cells, at least one for each digital signal pin
 - The bypass register
- These registers, and the device identification register, if implemented, are **REQUIRED** to be dedicated test logic (used for no other purpose)





The Boundary-Scan Register

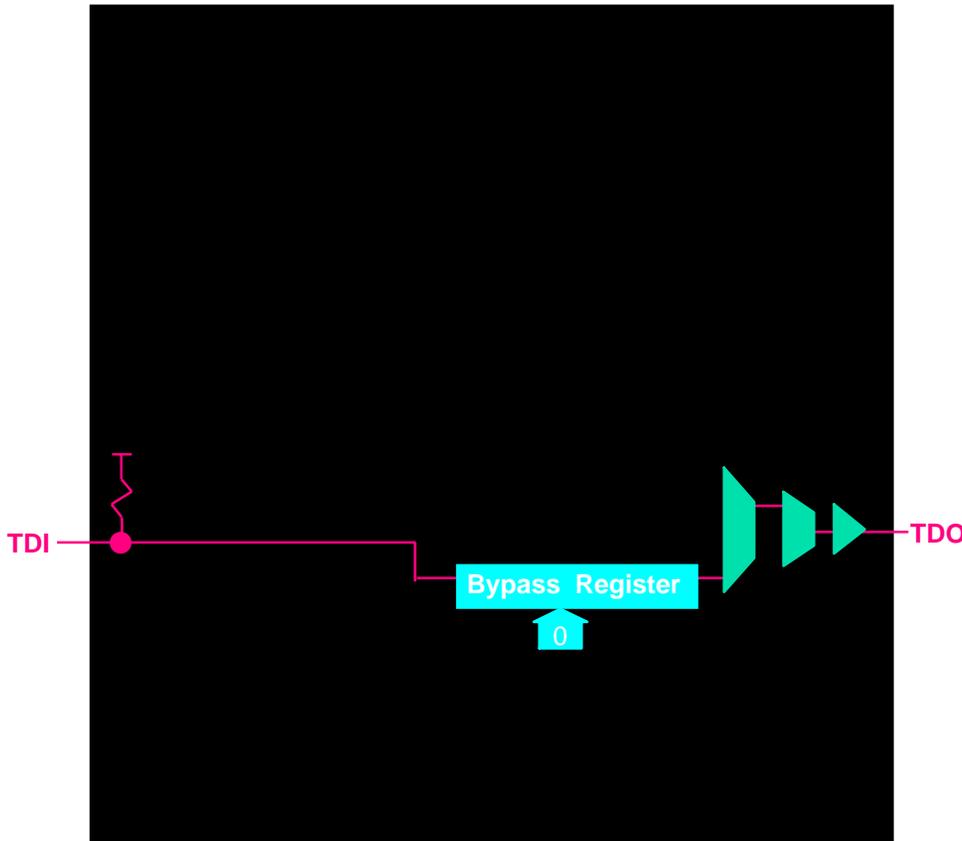


- The boundary-scan register is REQUIRED to be a concatenation of all boundary-scan cells
 - it is REQUIRED that there be at least one BSC for each digital input or output to the system logic (including on-chip outputs to analog circuitry)
 - it is REQUIRED that there NOT be BSCs at
 - » TAP pins
 - » compliance enables
 - » non-digital signals





The Bypass Register

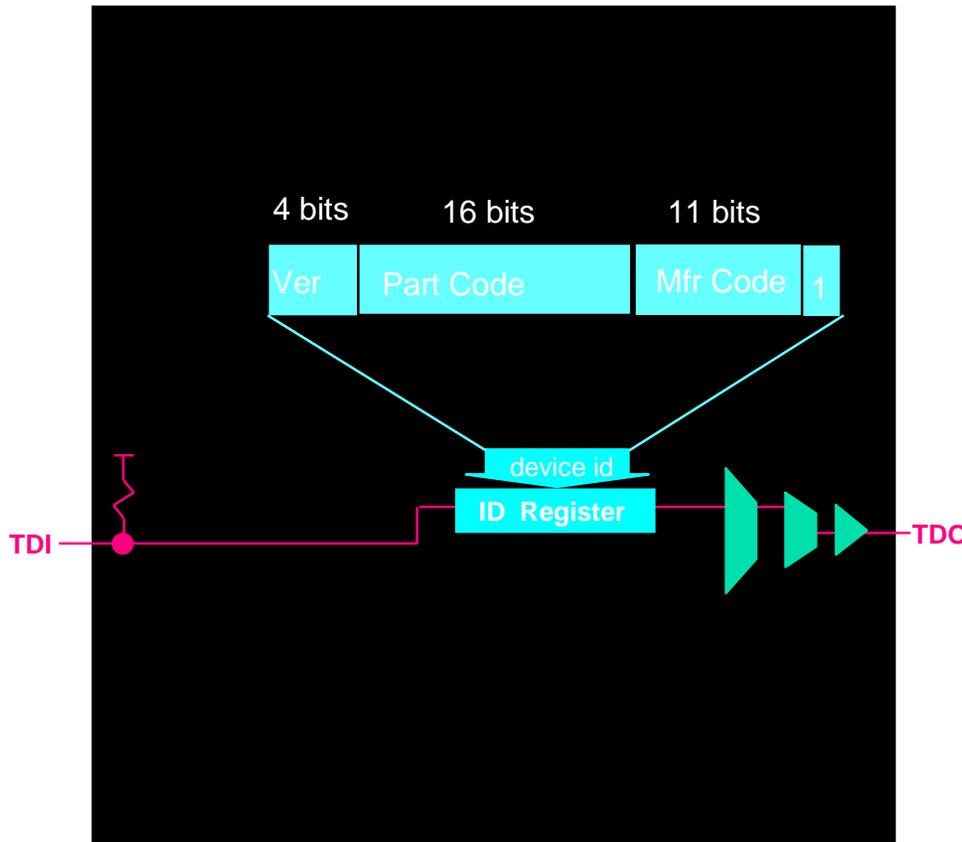


- The bypass register is REQUIRED to be one bit in length
- The bypass register is REQUIRED to capture a logic 0 value in Capture-DR state
- It is REQUIRED that any operation of the bypass register have no effect on the operation of the system logic





The Device Identification Register



Note: For TI components, the mfr code + LSB will always have value 02F
 Also, TI contact for part codes is Doug Kostlan, d-kostlan@ti.com.
 Version codes are managed by the originating product group.

- The device identification register is REQUIRED to be a 32-bit register
- It is REQUIRED to capture a device-specific value in Capture-DR state - this value has 4 fields
 - LSB = always logic 1
 - 1-11 = mfr code
 - 12-27 = part code
 - 28-31 = version code
- It is REQUIRED that any operation of the device identification register have no effect on the operation of the system logic





Instruction Summary

| Instruction (<u>REQUIRED/</u> Optional) | Opcode | Mode | Selected Data Register |
|--|----------------|----------------|---------------------------|
| <u>EXTEST</u> | 0..0* | Test | Boundary |
| <u>SAMPLE/PRELOAD</u> | user-specified | Normal | Boundary |
| <u>BYPASS</u> | 1..1 | Normal | Bypass |
| Intest | user-specified | Test | Boundary |
| Runbist | user-specified | Test | User-specified |
| Idcode | user-specified | Normal | Device ID |
| Usercode | user-specified | Normal | Device ID |
| Clamp | user-specified | Test | Bypass |
| HighZ | user-specified | Test | Bypass |
| user-defined | user-specified | user-specified | user-specified |

* - 1149.1 revision in progress will remove the requirement for EXTEST opcode of all zero

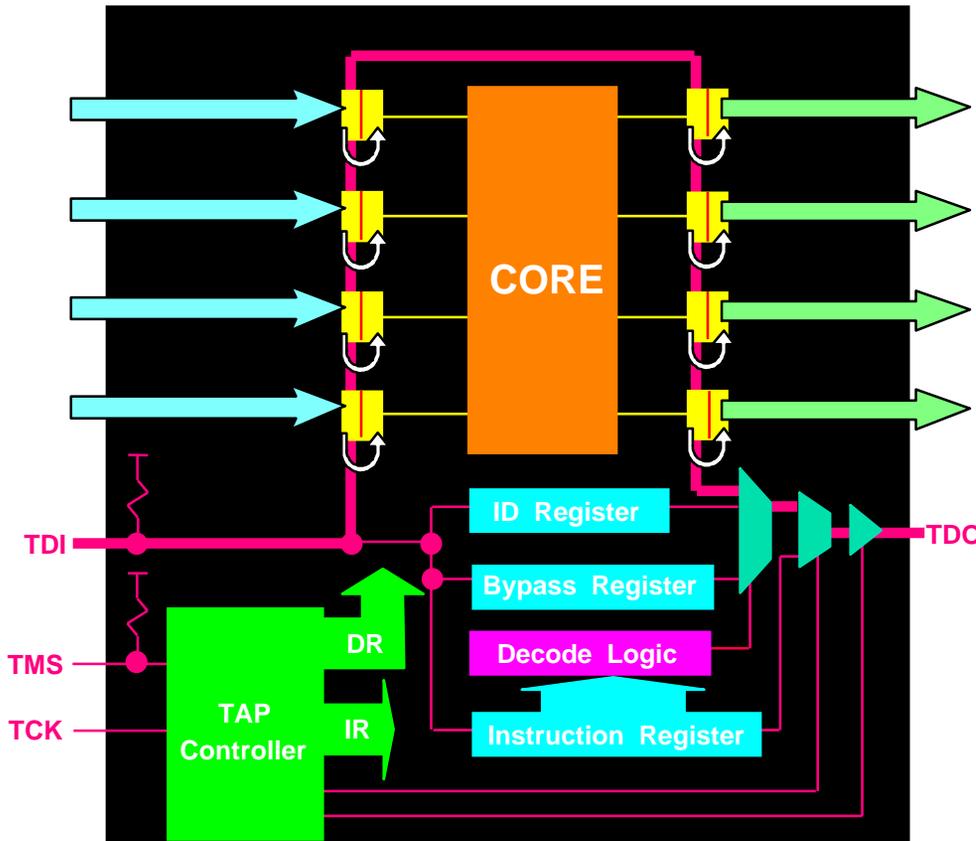




The Exttest Instruction

(REQUIRED)

- Provides for test external to chip, such as interconnect test



➡ Output pins operate in test mode, driven from contents of BSC update latch

➡ Input data captured in BSC scan latches prior to shift operation

- Shift operation allows test response to be observed at TDO while next test stimulus inserted at TDI

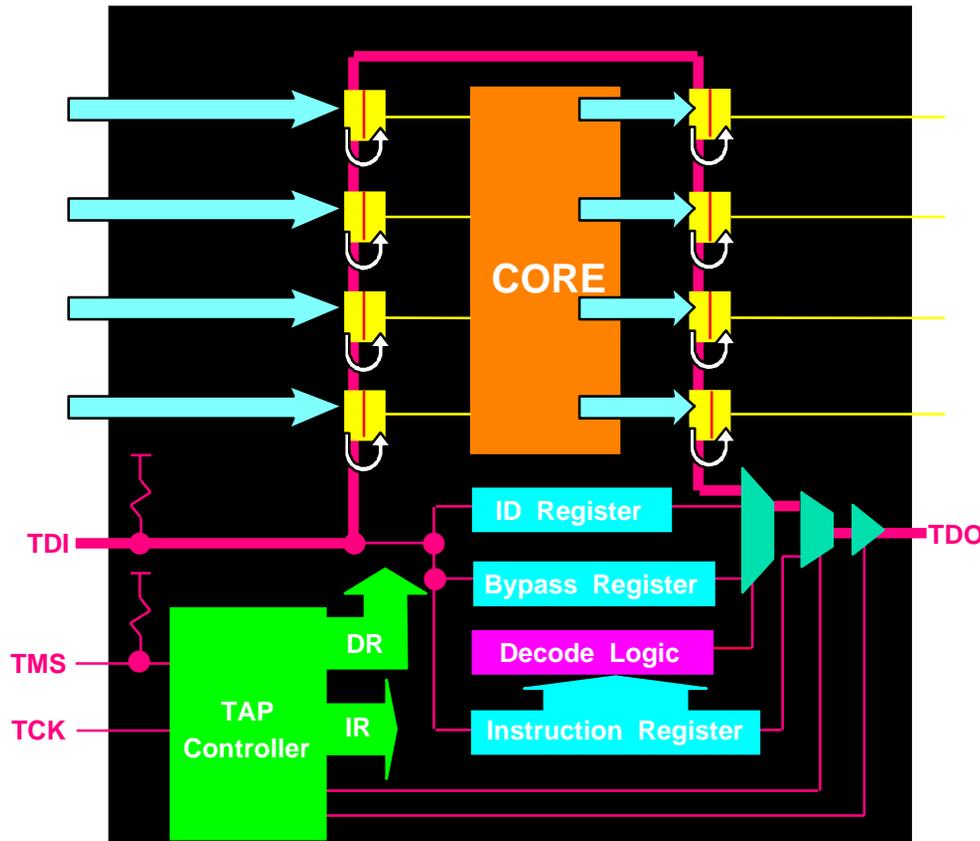
Following shift operation, new test stimulus transferred to BSC update latches





The Sample/Preload Instruction

(REQUIRED)



- Provides means to preload boundary before entry to test mode
- Output and input pins operate in normal mode
- ➡ Input pin data and core logic output data captured in BSC scan latches
- Shift operation allows test response to be observed while next test stimulus inserted at TDI
- ↻ Following shift operation, new stimulus transferred to BSC update latches

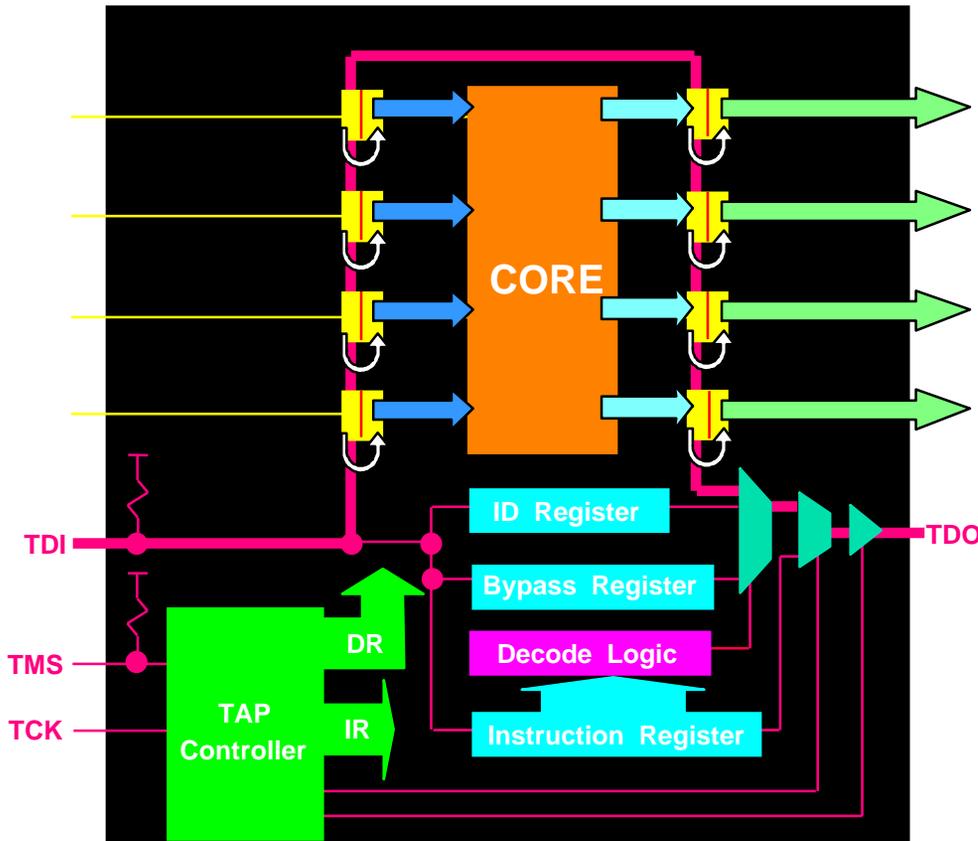




The Intest Instruction

(optional)

- Provides for test internal to chip - core functional test



- ➔ Output pins operate in test mode, driven from contents of BSC update latch or HighZ
- ➔ Core inputs operate in test mode, driven from contents of BSC update latch
- ➔ Core output data captured in BSC scan latches prior to shift operation

- Shift operation allows test response to be observed at TDO while next test stimulus inserted at TDI

➔ Following shift operation, new test stimulus transferred to BSC update latches





The Runbist Instruction

(optional)

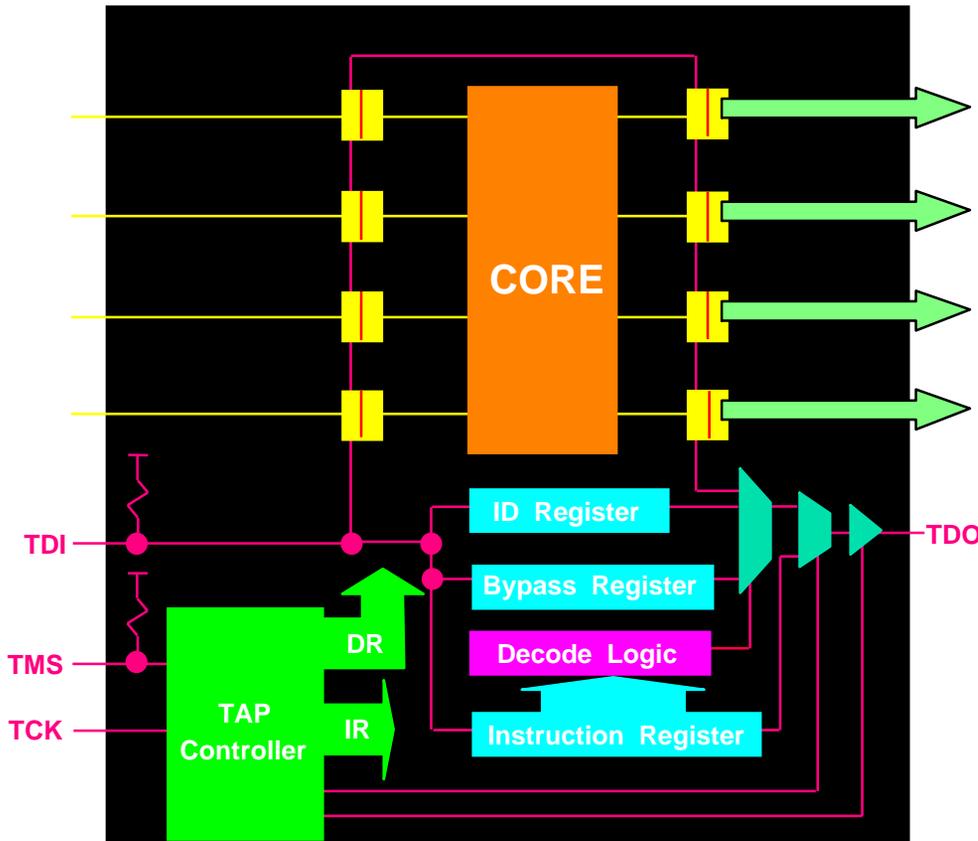
- Provides for test internal to chip - core built-in self test (BIST)

➔ Output pins operate in test mode, driven from contents of BSC update latch or HighZ

- Autonomous BIST is run in the core during Run-Test/Idle TAP state

- After completion, the BIST result/signature is captured into a user-specified register

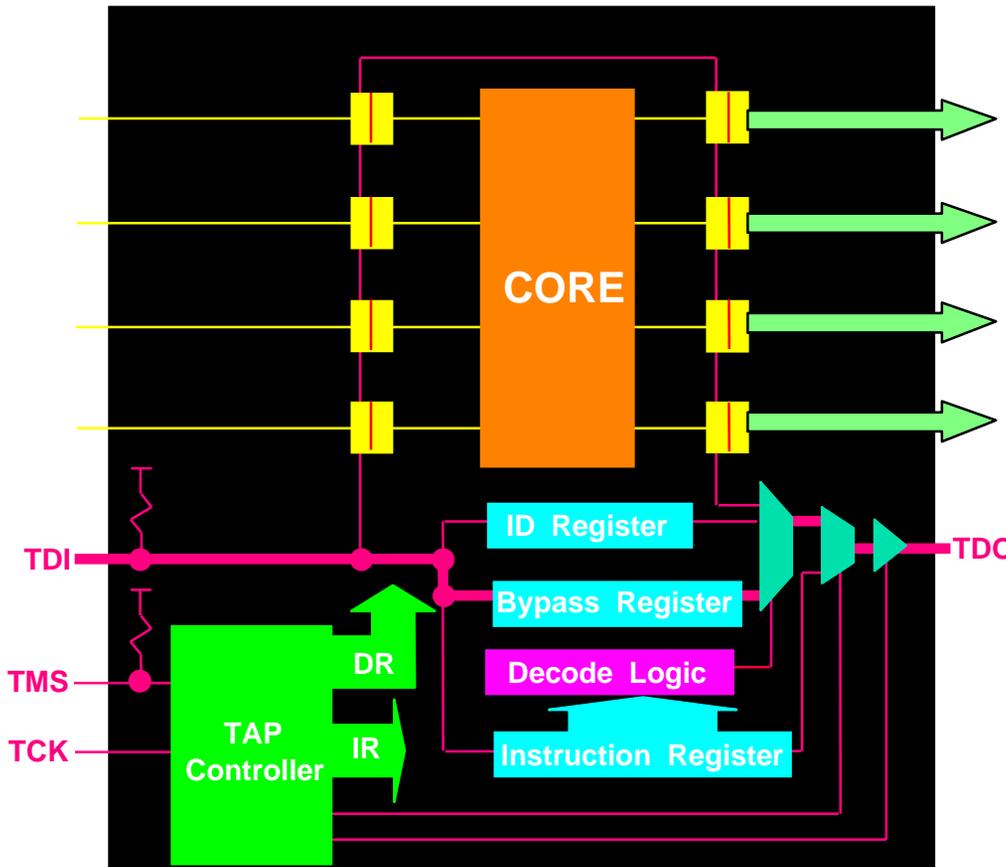
- Shift operation allows BIST result/signature to be observed at TDO





The Clamp Instruction

(optional)



■ Provides for “guarding” chip outputs during in-circuit test or boundary-scan functional test

➔ Output pins operate in test mode, driven from contents of BSC update latch

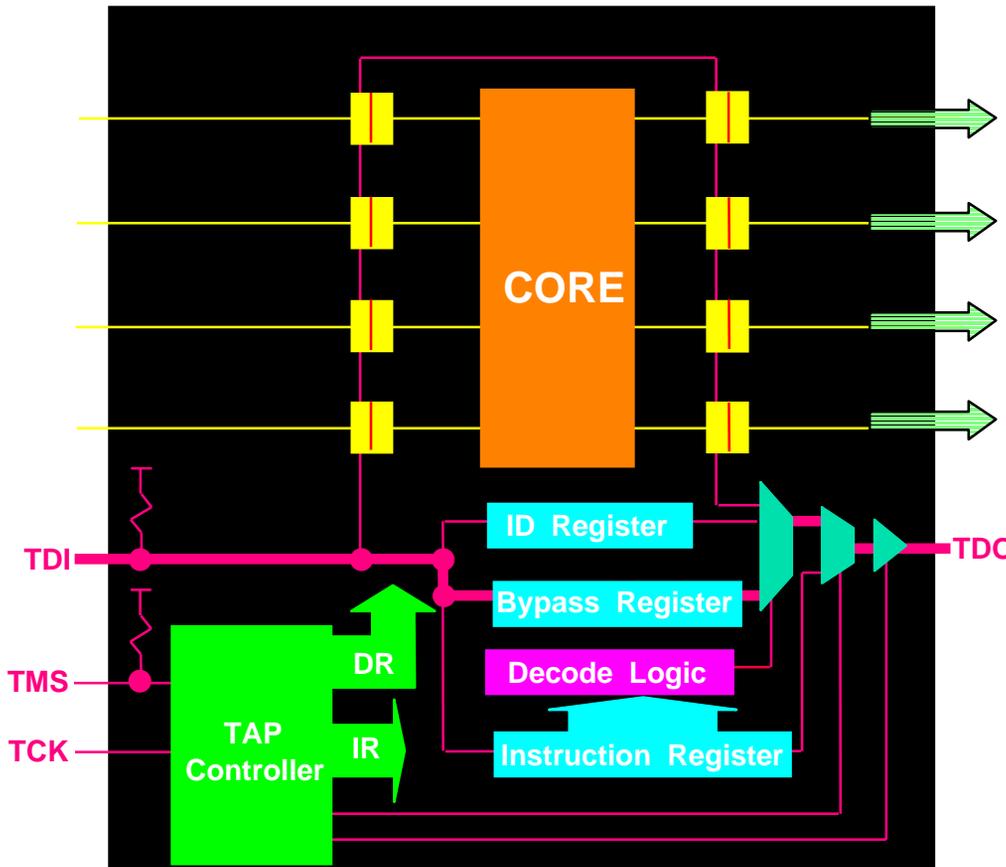
■ The one-bit bypass register is selected for scans





The Highz Instruction

(optional)



- Provides for disabling chip outputs during in-circuit test or boundary-scan functional test

→ Output pins are placed at high-impedance, INCLUDING pins which are 2-state for normal function

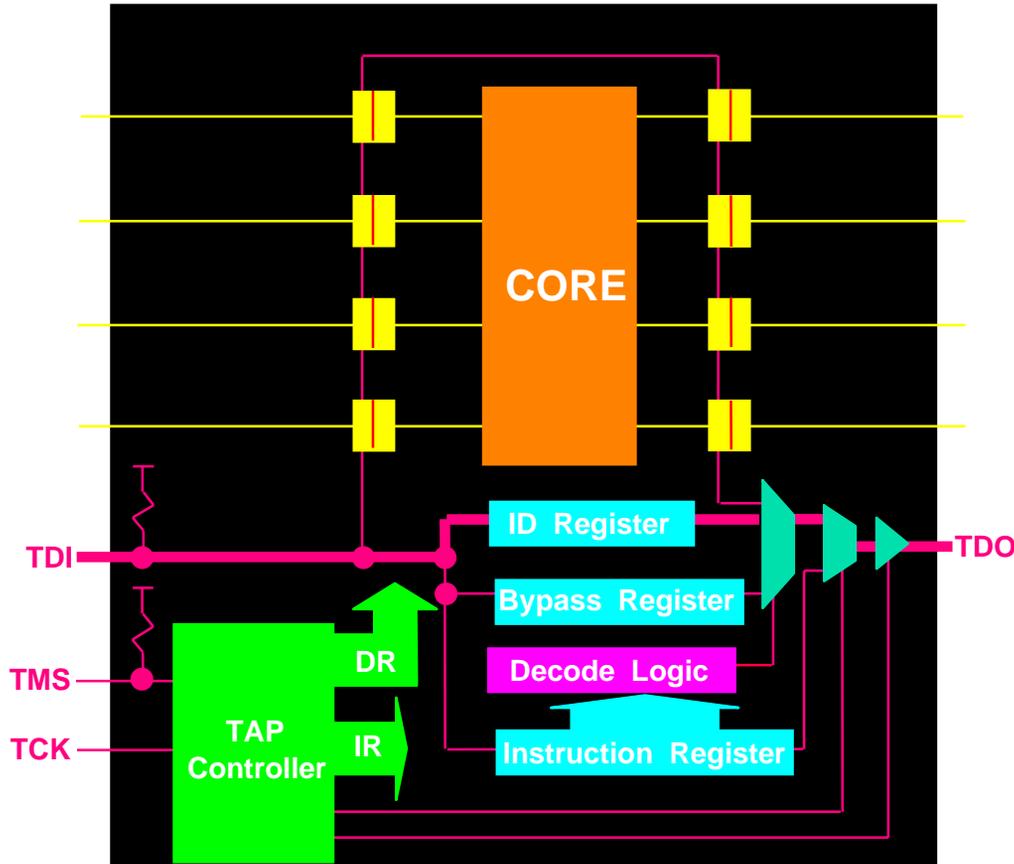
- The one-bit bypass register is selected for scans





The Idcode Instruction

(optional)



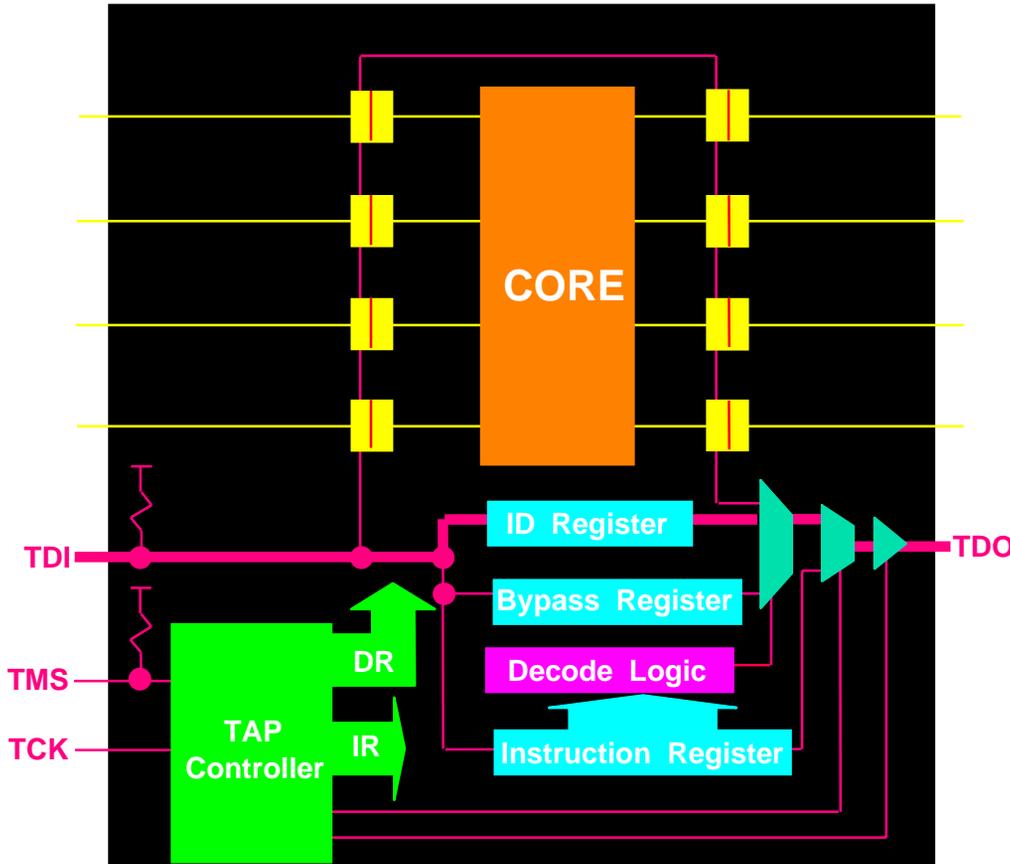
- Provides for device identification
- Output and input pins operate in normal mode
- The 32-bit device identification register is selected for scans - capture value is hardwired for the chip manufacturer, part type, and version





The Usercode Instruction

(optional)

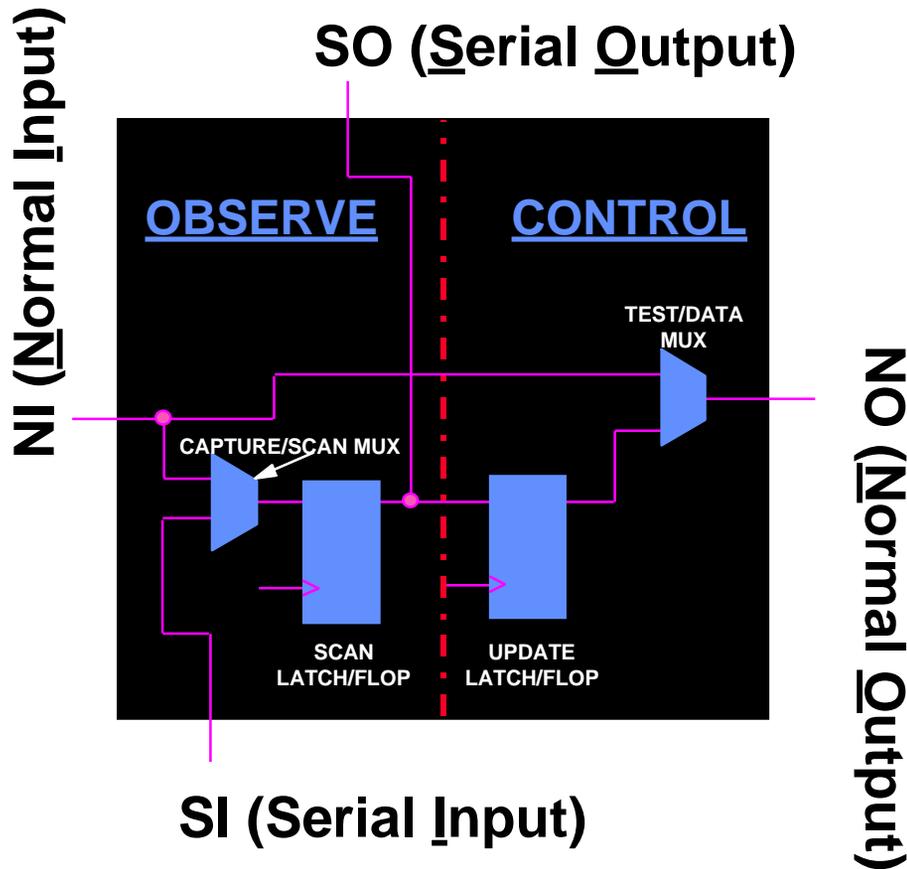


- Provides for extended device identification (programming) for PLDs
- Output and input pins operate in normal mode
- The 32-bit device identification register is selected for scans - capture value is fixed at time of device programming





The Latched-Output Observe/Control Boundary Scan Cell

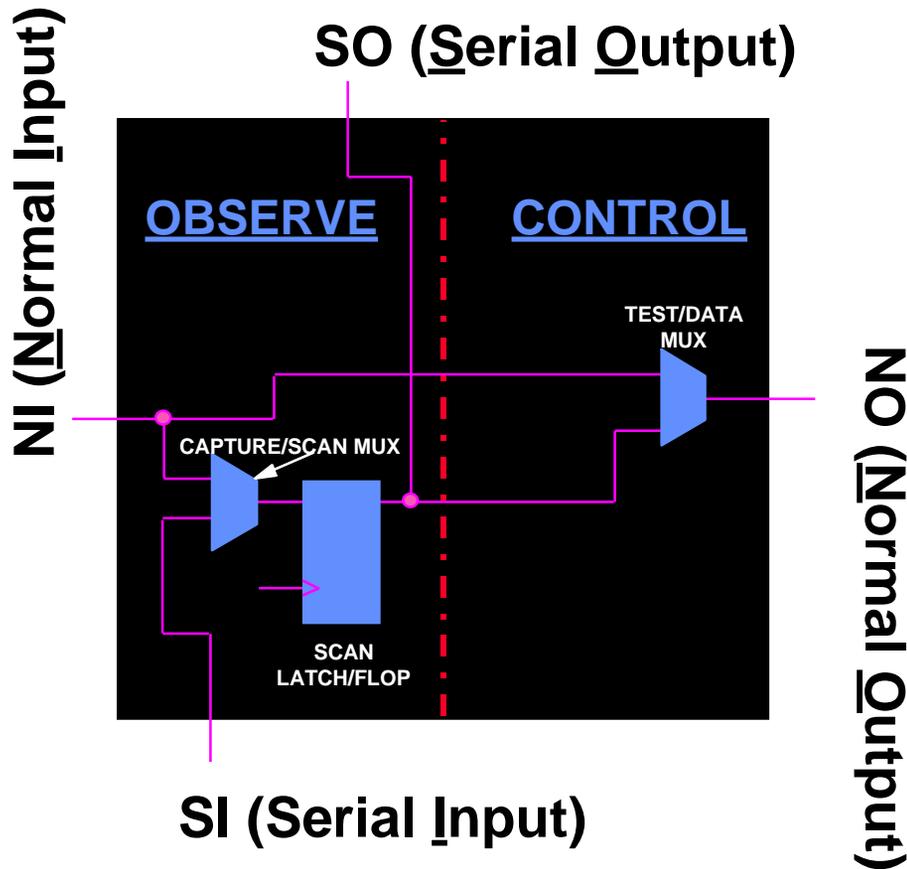


- Capture/scan mux selects parallel load vs serial scan
- Test/data mux selects normal data vs test data
- Latched output prevents “ripple” at chip pins while scan is in progress
- REQUIRED at all output pins (for both data and control signals)
- Suitable for use at ANY input, output data, or output control





The Observe/Control Boundary Scan Cell

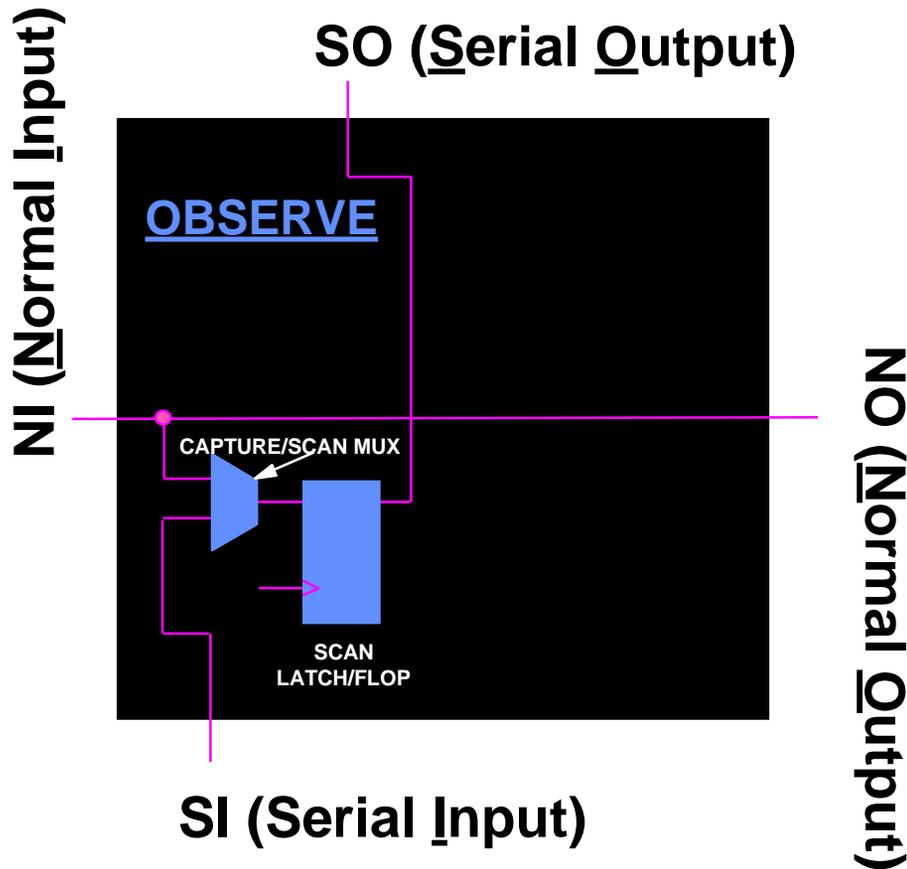


- Capture/scan mux selects parallel load vs serial scan
- Test/data mux selects normal data vs test data
- Unlatched output allows “ripple” at chip pins while scan is in progress
- Suitable for use at inputs which can tolerate “rippling”
- NOT SUITABLE at ANY output pins (for both data and control signals)





The Observe-Only Boundary Scan Cell

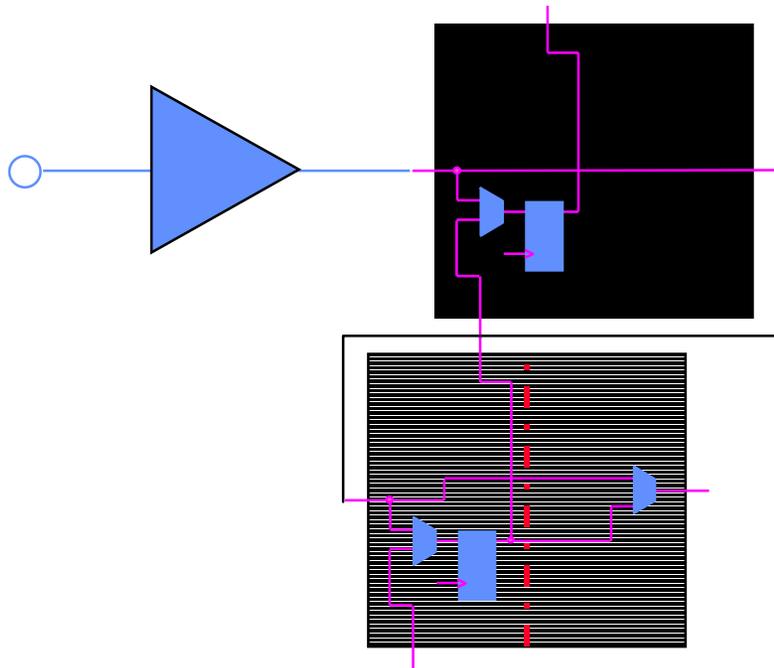


- Capture/scan mux selects parallel load vs serial scan
- Lack of test/data mux means test data may not be selected (only normal data)
- Suitable for use at ANY input, unless INTEST is implemented
- NOT SUITABLE at nonclock inputs if INTEST is implemented
- NOT SUITABLE at ANY output pins (for both data and control signals)





Provisioning of BSCs at Inputs

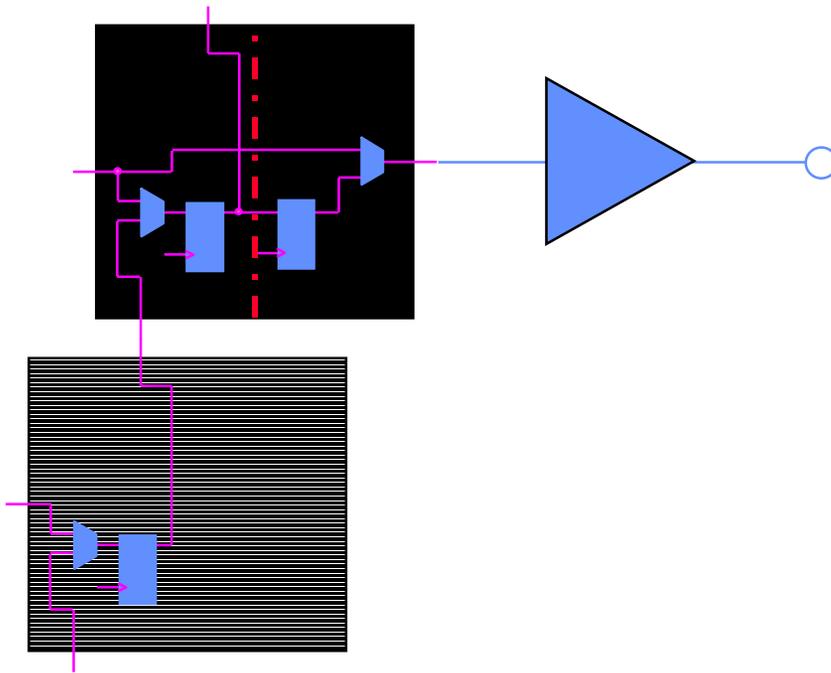


- Each chip input is REQUIRED to have at least one associated BSC with observe capability and also MAY have one associated BSC with control capability
- If the optional INTEST instruction is supported, then each chip input which is not a clock is REQUIRED to have one associated BSC with control capability (either latched or unlatched) - clocks MAY also have one associated BSC with control capability





Provisioning of BSCs at 2-state Outputs

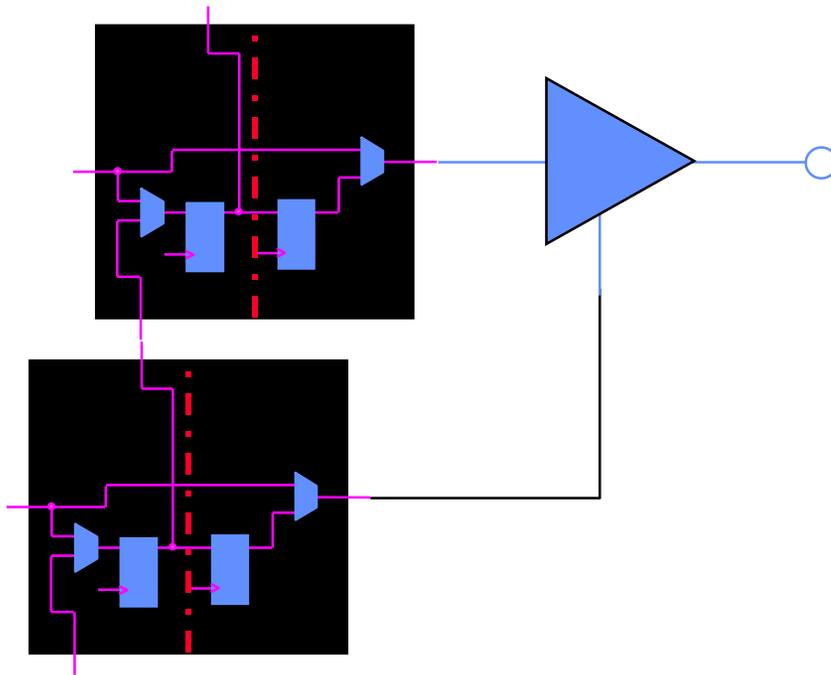


- Each chip output is REQUIRED to have at least one associated BSC with latched-output observe/control capability and also MAY have any number of associated BSCs with observe-only capability
- If the optional HighZ instruction is supported then all outputs which have 2-state system function actually must be implemented with 3-state buffers and means must be provided to force them to high impedance state during HighZ





Provisioning of BSCs at 3-state Outputs

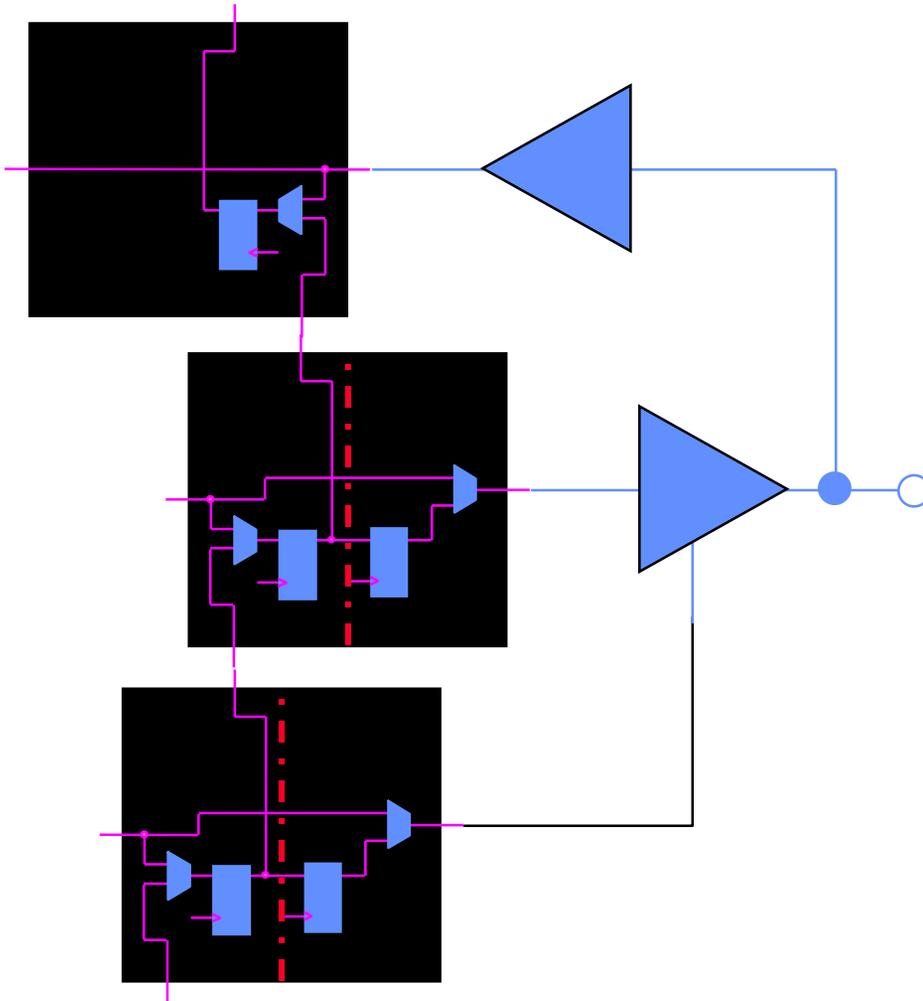


- Each chip 3-state output is REQUIRED to have at least one associated BSC with latched-output observe/control capability at each data input and control input and also MAY have any number of associated BSCs with observe-only capability
- If the optional HighZ instruction is supported then means must be provided to force outputs to high impedance state during HighZ





Provisioning of BSCs at Bi-directionals



- Each chip bi-directional is REQUIRED to have associated BSCs such that when it operates as an input it meets input rules and when it operates as an output it meets output rules - in practice, this means control signal must have latched O/C BSC and there must be at least 1 BSC for data (input/output)
- If the optional HighZ instruction is supported then means must be provided to force bi-directionals to high impedance state during HighZ





Boundary-Scan Description Language (BSDL)





Purpose/Use of BSDL

- BSDL provides a standard machine- and human-readable format for describing how IEEE Std 1149.1 boundary-scan architecture is implemented in a device.
- BSDL has been formally adopted as part of IEEE Std 1149.1 in supplement 1149.1b-1994.
- BSDL is a subset and standard practice of VHDL (VHSIC Hardware Description Language)
- BSDL is used as a data input format for test tools, such as ATPG and ATE, and for automated generation of boundary-scan test logic.
- BSDL is generated as an output from many Design-for-Test automation tool sets.





Elements of BSDL

- A BSDL description consists of the following elements:
 - Entity descriptions
 - Generic parameter
 - Logical port description
 - Use statement(s)
 - Component conformance statement
 - Pin mapping(s)
 - Scan port identification
 - Instruction register description
 - Optional register description
 - Register access description
 - Boundary register description





Elements of BSDL

- Entity Descriptions — The entity statement names the entity, such as the device name (e.g., SN74BCT8245A). An entity description begins with an entity statement and terminates with an end statement.

```
entity XYZ is  
  {statements to describe the entity go here}  
end XYZ;
```

- Generic Parameter — A generic parameter is a parameter that can come from outside the entity, or it can be defaulted, such as a package type (e.g., “DW”).

```
generic (PHYSICAL_PIN_MAP : string := "DW");
```





Elements of BSDL

- Logical Port Description — The port description gives logical names to the I/O pins (system and TAP pins), and denotes their nature, such as input, output, bi-directional, linkage (analog or power supply/return) and so on.

```
port (OE:in bit;
      Y:out bit_vector(1 to 3);
      A:in bit_vector(1 to 3);
      GND, VCC, NC:linkage bit;
      TDO:out bit;
      TMS, TDI, TCK:in bit);
```

- Use Statement(s) — The use statement refers to external definitions found in packages and package bodies.

```
use STD_1149_1_1994.all;
```





Elements of BSDL

- Component Conformance Statement — The component conformance statement indicates the latest issue of IEEE Std 1149.1 to which the device conforms.

```
attribute COMPONENT_CONFORMANCE of XYZ : entity is
  "STD_1149_1_1993" ;
```

- Pin Mapping(s) — The pin mapping provides a mapping of logical signals onto the physical pins of a particular device package.

```
attribute PIN_MAP of XYZ : entity is
  PHYSICAL_PIN_MAP ;
constant DW:PIN_MAP_STRING:=
  "OE:1, Y:(2,3,4), A:(5,6,7), GND:8, VCC:9, " &
  "TDO:10, TDI:11, TMS:12, TCK:13, NC:14" ;
```





Elements of BSDL

- Scan Port Identification — The scan port identification statements define the device's TAP.

```
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);
```





Elements of BSDL

- Instruction Register Description — The instruction register description identifies the device-dependent characteristics of the instruction register.

```
attribute INSTRUCTION_LENGTH of XYZ : entity is 2;
```

```
attribute INSTRUCTION_OPCODE of XYZ : entity is
```

```
"BYPASS (11), "&
```

```
"EXTEST (00), "&
```

```
"SAMPLE (10), "&
```

```
"IDCODE (01)"
```

```
attribute INSTRUCTION_CAPTURE of XYZ : entity is
```

```
"01";
```





Elements of BSDL

- Optional Register Description — The optional register description identifies the values captured in the device identification register for the optional IDCODE and USERCODE instructions, if supported.

```
attribute IDCODE_REGISTER of XYZ : entity is  
    "01010100000011111100000000101111";
```

- Register Access Description — The register access description defines which register is placed between TDI and TDO for each instruction.

```
attribute REGISTER_ACCESS of XYZ : entity is  
    "BOUNDARY (EXTEST, SAMPLE), "&  
    "BYPASS (BYPASS)";
```





Elements of BSDL

- **Boundary Register Description** — The boundary register description contains a list of boundary-scan cells, along with information regarding the cell type and associated control.

```

attribute BOUNDARY_LENGTH of XYZ : entity is 7;
attribute BOUNDARY_REGISTER of XYZ : entity is
"0 (BC_1, Y(1), output3, X, 6, 0, Z),"&
"1 (BC_1, Y(2), output3, X, 6, 0, Z),"&
"2 (BC_1, Y(3), output3, X, 6, 0, Z),"&
"3 (BC_1, A(1), input , X),"&
"4 (BC_1, A(2), input , X),"&
"5 (BC_1, A(3), input , X),"&
"6 (BC_1, OE , input , X),"&
"6 (BC_1, * , control, 0)";

```





Verifying BSDL Accuracy

- BSDL should be validated against silicon in addition to being checked for syntax and semantics.
- Even after passing syntax and semantics, the following BSDL errors could potentially exist:
 - Wrong pinout
 - Wrong cell types
 - Wrong boundary-scan register order
 - Wrong boundary-scan register length
 - Wrong instruction register opcodes
 - Wrong control cell locations
 - Wrong control cell disable value
 - Wrong I/O pin control cell
 - Wrong identification code value
 - Wrong capture-IR value





Other Supporting Data Formats





Hierarchical Scan Description Language (HSDL)

- HSDL describes how IEEE Std 1149.1 devices are connected at the board and system levels.
- HSDL describes additional attributes of IEEE Std 1149.1 devices which BSDL does not.
- Originally developed by Texas Instruments primarily in support of its ASSET business, HSDL is now supported by ASSET InterTech, which acquired ASSET in 1995.





Elements of HSDL

- HSDL adds new statements to BSDL to describe the members and scan paths of a module and to simplify interactive use:

Entity description

Generic parameter

Logical port description

Use statement(s)

[Optional module description(s)]

[Optional port description(s)]

Pin mapping(s)

Scan port identification

[Optional members description(s)]

[Optional bus composition(s)]

Path description

[Optional member connections]

[Optional constraint description(s)]

[Optional design warning]





Elements of HSDL

- Members Description (Optional) — Members represent devices or other modules that are on the module. Usually members represent components, but some boards may contain scannable daughter cards, card slots, etc. that require member modules to describe them.

```
attribute MEMBERS of BOARD : entity is  
  "U1 (XYZ1, DW), "&  
  "U2 (XYZ2, DW)";
```





Elements of HSDL

- Bus Composition (Optional) — Buses in an HSDL module can be built of module buses, member module buses, member device buses, and member device test registers.

```
attribute BUS_COMPOSITION of BOARD : entity is
"bus1[4] (U1.Boundary[3,0]),"&
"bus2[4] (U2.Boundary[3,0])";
```

- Path Description — Module paths are intended to describe the netlist of TAP signals (scan paths) on the board.

```
constant boardpath1 : STATIC_PATH :=
"U1, U2";
end BOARD;
```





Serial Vector Format (SVF)

- SVF is a standard ASCII format for expressing test patterns that represent the stimulus, expected response, and mask data for IEEE Std 1149.1-based tests.
- The need for SVF arose from the desire to have vendor-independent IEEE Std 1149.1 test patterns that are transportable across a wide selection of simulation software and test equipment from design verification through field diagnostics.
- SVF was jointly developed by Texas Instruments and Teradyne in 1991.





Serial Vector Format (SVF)

- SVF controls the IEEE Std 1149.1 test bus using commands that transition the TAP from one steady state to another.
- Rather than describe the explicit state of the IEEE Std 1149.1 bus on every TCK cycle, SVF describes it in terms of transactions conducted between stable states.
- In addition to supporting higher level depictions of scan operations, SVF also supports combined serial and parallel operations.





SVF Commands

■ State commands:

SDR Scan data register

SIR Scan instruction register

ENDDR Define end state of DR scan

ENDIR Define end state of IR scan

RUNTEST Enter Run-Test/Idle state

STATE Go to specified stable state

TRST Drive the TRST line to the designated level

■ Offset commands:

HDR Header data for data bits

HIR Header data for instruction bits

TDR Trailer data for data bits

TIR Trailer data for instruction bits

■ Parallel commands:

PIO — Specifies a parallel test pattern

PIOMAP — Designates the mapping of bits in the PIO command to logical pin names





SVF Example

```

! Begin Test Program : Disable Test Reset line
TRST OFF;
! Initialize UUT
STATE RESET;
! End IR scans in DRPAUSE
ENDIR DRPAUSE;
! End DR scans in DRPAUSE
ENDDR DRPAUSE;
! 24 bit IR header
HIR 24 TDI (FFFFFF);
! 3 bit DR header
HDR 3 TDI (7);
! 16 bit IR trailer
TIR 16 TDI (FFFF);
! 2 bit DR trailer
TDR 2 TDI (3);
! 8 bit IR scan, load BIST opcode
SIR 8 TDI (41) TDO (81) MASK (FF);
! 16 bit DR scan, load BIST seed
SDR 16 TDI (ABCD);
! RUNBIST for 95 TCK Clocks
RUNTEST 95 TCK ENDSTATE IRPAUSE;
! 16 bit DR scan, check BIST status
SDR 16 TDI (0000) TDO(1234) MASK(FFFF);
! Enter Test-Logic-Reset
STATE RESET;
! End Test Program

```





Future Directions





A Brief History of 1149.1

- Std 1149.1 was initially ratified by the IEEE in February of 1990 culminating 5 years of technical and editorial work which began in JTAG.
- Std 1149.1 has been supplemented twice in the interim:
 - IEEE Std 1149.1a-1993 was issued as a major clarification of the original standard - specifications for CLAMP and HIGHZ instructions were added
 - IEEE Std 1149.1b-1994 was issued as a major addition to the standard - specifically, the BSDL
- IEEE Standards policies and procedures require that standards are reviewed on a 5-year periodic basis. Std 1149.1 came due in 1995.





Std 1149.1 Revision in Progress

- Since somewhat before BSDL was ratified in Sept 1994, the 1149.1 working group has been actively engaged in a process of review and revision. Following are some changes which are being incorporated into a new draft:
 - Permit “extension standards” to supplement the TAP with additional test-dedicated pins (P1149.4 enabler)
 - Add provisions for boundary cells which provide digital test facilities at analog signal pins (“Whet-cell”)
 - Remove the requirement for Extest to have the all zeros opcode (better fail-safe design)
 - Other clarifications (differential, device ID code, SAMPLE)





P1149.4 Mixed Signal Test Bus

- The P1149.4 project of the IEEE seeks to define a test bus which facilitates continuous signal testing of mixed signal ICs, boards, and systems - their first draft is currently out for ballot.
- The current draft calls for compliant ICs to firstly be fully compliant to Std 1149.1, then specifies additional requirements as follows:
 - digital test cells at analog signal pins
 - two bussed test pins for analog stimulus/response
 - on-chip switching matrices to route analog stimulus/response to the pin of interest
 - the draft also documents a metrology to be used with the specified test circuitry





Q and A Notes





Notes





Abbreviations/Acronyms

| | | | |
|------|---|------|---|
| ASIC | Application-Specific Integrated Circuit | IEEE | Institute of Electrical & Electronics Engineers |
| ASP | Addressable Scan Port | | IR Instruction Register |
| ATE | Automatic Test Equipment | ISP | In-System Programming |
| ATPG | Automatic Test Pattern Generation | JTAG | Joint Test Action Group |
| BIST | Built-In Self-Test | MCM | Multi-Chip Module |
| B/S | Boundary-Scan | Mfg | Manufacturing |
| BSC | Boundary-Scan Cell | PCB | Printed Circuit Board |
| BSDL | Boundary-Scan Description Language | PLD | Programmable Logic Device |
| BSR | Boundary-Scan Register | PRPG | Pseudo-Random Pattern Generation |
| BST | Boundary-Scan Test | PSA | Parallel Signature Analysis |
| CAE | Computer-Aided Engineering | PWB | Printed Wiring Board |
| DFT | Design-for-Test | SPL | Scan Path Linker |
| DR | Data Register | SVF | Serial Vector Format |
| DSP | Digital Signal Processing/Processor | TAP | Test Access Port |
| EDA | Electronic Design Automation | TBC | Test Bus Controller |
| eTBC | Embedded Test Bus Controller | TCK | Test Clock |
| FPGA | Field-Programmable Gate Array | TDI | Test Data Input |
| HSDL | Hierarchical Scan Description Language | TDO | Test Data Output |
| ICE | In-Circuit Emulation | TMS | Test Mode Select |
| ICT | In-Circuit Test | TRST | Test Reset |
| | | UUT | Unit Under Test |

