Introduction to JTAG

apertus°
open source cinema

Swaraj Hota

Contents

Boundary Scan
JTAG Intro
JTAG Architecture
Relationship with BSDL
An Example

What is Boundary Scan?

- Allows you to debug individual IO pins of a chip in an assembled PCB (without physical access)
- Uses extra circuitary (called *wrapper*), one “test cell” attached to each pin
- These test cells provide a way to capture/update data
- All test cells are “daisy chained” (think a single big shift register)
- We essentially need a serial data i/p pin and an o/p pin to shift-in and shift-out bits from all the pins
- The *wrapper* lies dormant (invisible) during normal operation

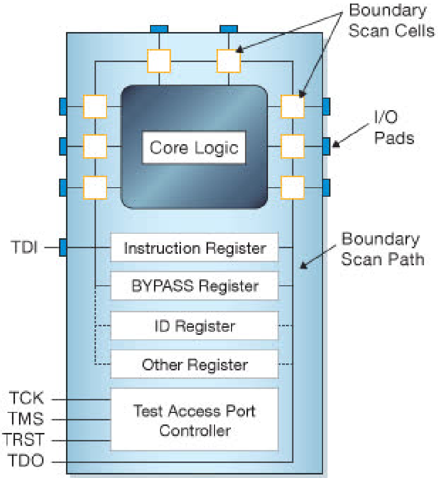
Enter... JTAG!

- JTAG, the IEEE 1149.1 standard, was formalized
- Also called the *boundary scan standard*
- A standardized way to do *boundary scan*

So... JTAG is just for Boundary Scan?

- Originally that was its purpose, but it turned out to be much more versatile!
- Not just single chips, any number of chips can be debugged in a board (again, daisy chained)
- And not just the “boundary”, interal system logic can also be debugged
- Further, can also be used for programming chips! (like FPGAs)

Schematic of a JTAG enabled device



Test Access Port (TAP)

The JTAG port or Test Access Port (TAP) requires only 5 pins:

- TDI: Test Data Input (Serial Data in)
- TDO: Test Data Output (Serial Data out)
- TCK: Test Clock
- TMS: Test Mode Select (Change states in JTAG state machine)
- TRST: Test Reset (Optional, to reset JTAG state machine)

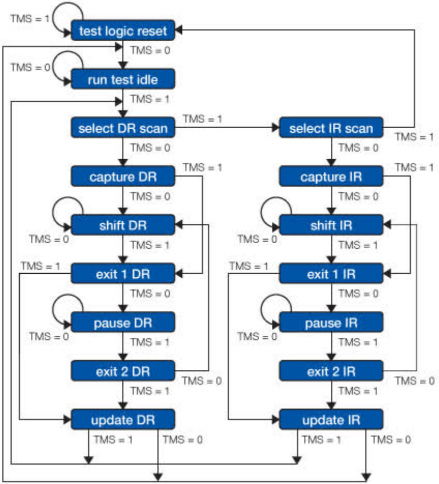
JTAG Registers

- Standard specifies one Instruction Register (IR) and various Data Registers (DR) to select from
- Mandatory Data registers:
  - Boundary Scan Register (BSR) - the *wrapper*
  - Bypass Register (BYPASS) - one bit register
- Optional Data Registers:
  - Device ID Register (IDCODE)
  - Various Device specific registers
- The register which is selected will shift-in bits from TDI and shift-out bits from TDO

TAP Controller

- Contains the *JTAG/TAP State Machine*
- Also the Instruction Register and Instruction Decoder
- According to the current state in the state machine, one of the data register or instruction register will be selected/scanned/captured/updated
- Which data register is selected will depend on the instruction passed

TAP State Machine



JTAG Instructions

- Mandatory instructions:
  - BYPASS (111...) - Use BYPASS Register (Passthrough)
  - EXTEST (000...) - Use BSR (Capture-DR -¿ Shift-DR -¿ Update-DR)
  - SAMPLE/PRELOAD - Use BSR (when device in normal mode)
- Optional instructions:
  - IDCODE - Use IDCODE register
  - INTEST - Use BSR (test internal core-logic)
  - Various other design specific instructions

Relationship with BSDL

- Boundary Scan Description Language (BSDL) is a subset of VHDL
- A BSDL file is needed for each JTAG enabled chip
- Contains all implementation details, like size of registers, number of registers, associated pins etc.
- A JTAG Debugger software can read this file to understand how to communicate with the device through JTAG

Example: EXTEST

- Load instruction:
  - TMS sequence: 11111 (Reset)
  - TMS sequence: 01100 {0...0} 110 (instruction)
  - While TMS in {0...0}, send via TDI: 000... (EXTEST)
- Shift BSR (for EXTEST)
  - TMS sequence: 100 {0...0} 111
  - While TMS in {0...0}, send via TDI: xxx... (pin states to be written)
  - Length of {0...0} = Total length of Data register (BSR, here)
- While TMS in {0...0}, receive captured pin states via TDO