# JTAG Boundary Scan & TAP

Jesús Arias

## 1 JTAG description

JTAG is an standard serial interface for hardware debug, or TAP (Test Access Port), that includes at least the following signals:

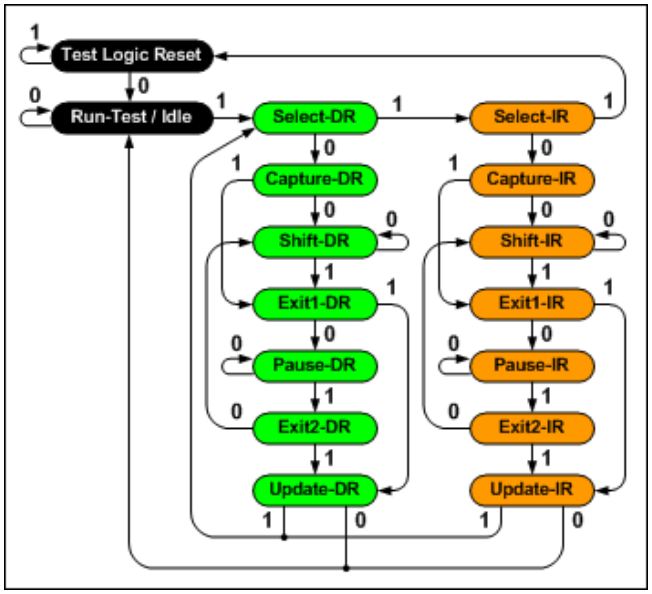
- TCK: Test Clock. Inputs are sampled on its rising edges and outputs change state on the falling edges (as in a mode #0 SPI bus)
- TMS: Test Machine State. An input that is used to change the internal state of the controller.
- TDI: Test Data Input. The serial input for registers.
- TDO: Test Data Output. The serial output of registers.

In addition to these signals there is also specified a reset signal, TRST, that is redundant because we can force a reset state by just keeping TMS at high for 5 TCK pulses. Thus, this reset signal is often removed in order to save a pin.

Inside the TAP there is an state machine controller and several registers like:

- IR: Instruction Register. It has a vendor defined length and meaning, with its main use being the enabling of test conditions and the selection of the current data register (any of the following registers).
- BSR: Boundary Scan register. This is the main TAP register for hardware tests, and its bits are directly related with the package pins of the device under test.
- DIR: Device Identification Register. It holds a 32-bit constant with information about the manufacturer, device, and revision of the chip under test.
- Bypass Register. A single bit register that can be selected instead of BSR if we want to skip this chip during tests. This allows a faster access to the rest of the boundary-scan chain.

The state machine diagram is shown in the following figure. It is a Moore machine with 16 states: 7 states for accessing the current data register (green), 7 states for accessing the instruction register (orange), one reset state, and an idle state. Each state has two possible next states that are selected by the level in the TMS signal:

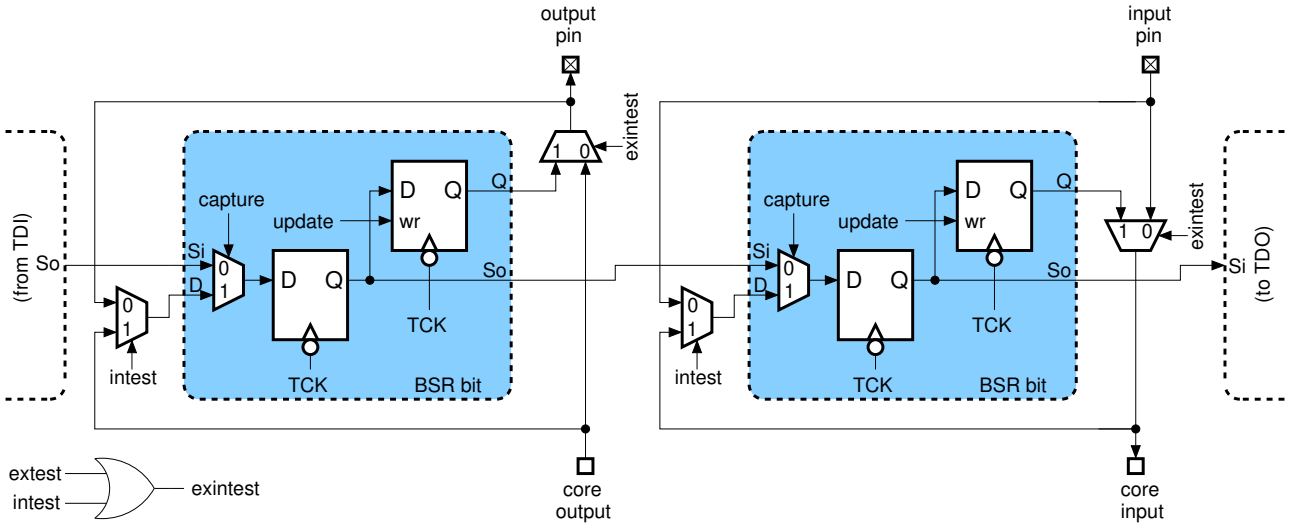



Usually, the controller is sitting in the IDLE state until some data has to be moved to or from a register. Then, the state has to be changed to one of the SHIFT-DR of SHIFT-IR states. The external JTAG interface can then shift the bits of the register through TDI/TDO in the same way as in an SPI bus, but the new data isn't presented at the register outputs until the UPDATE-DR or UPDATE-IR state is reached (this implies two flip-flops for each bit, more or less the same as in the 74xx595 serial to parallel shift register). Also, before reaching the shift state a CAPTURE-DR or CAPTURE-IR state was entered, resulting in the parallel load of the shift register chain with the input data of the selected register that will be then shifted out.

As we can see, the IR register has its own states and it can be always written without a previous selection. The ugly thing about it is the lack of standardization about its width and meaning (every vendor defines it as it likes), but the set has to support at least some basic instructions:

- IDCODE. This IR value is automatically loaded after entering the RESET state, meaning we can read the 32 bits of the DIR register and check a database about the current device under test.
- BYPASS. Often an all-ones value. Selects the bypass register as the data register.
- SAMPLE/PRELOAD. Selects the BSR register as the DR, allowing the sampling of the pin levels on the CAPTURE-DR state, and then their shifting towards the TDO pin. AT the same time another values can be loaded into the BSR from TDI, but these values are hold internally without disturbing the operation of the device.
- EXTEST. External Test. BSR is selected as DR and its contents are presented to the output pins of the device instead of the usual internal values. The input pins can also be sampled. That means the BSR is replacing the device when viewed from outside the chip.
- INTEST. Internal Test. BSR is selected as DR and its contents are presented to the input signals of the device core. The output signals of the core can also be sampled. That means the BSR is replacing the outside world when viewed from the inside of the chip.

So, taking into account these instructions the interfacing of the BSR bits can follow this schematic:




Here, the core signals are routed to the external pins until either an EXTEST or INTEST instruction is stored in IR. Then, the output of the BSR bit replaces the core signal for outputs or the external pin for inputs. The BSR samples the external pins for all instructions but INTEST, where the internal signals of the core are routed to the BSR inputs instead.

This schematic presents a simple and general case for the BSR bits but there are other variants, like:

- Control bits. They are usually intended for tristate outputs or bidirectional pins. They modify the behavior of a pin already controlled by another BSR bit. These bits lack the input multiplexers because they don't sample anything.
- Input-only. These bits lack the output flip-flop. They can replace a general purpose BSR bit if we want to sample an input signal and we don't mind about the value presented to the core (or to the pin for INTEST)

## 2 Details of the laRVa_tinytapeout JTAG

In this device the IDCODE value is: 0x00047FAB, meaning :

| | | |
|---|---|---|
| Manufacturer | 1111.1010.101.1 | Generic manufacturer |
| device | 0x0047 | Tiny Tapeout laRVa prototype |
| Stepping | 0x0 | First prototype |

The IR register is only 3-bit wide and has the following instructions:

| | |
|---|---|
| 000 | IDCODE |
| 001 | SAMPLE/PRELOAD |
| 010 | EXTEST |
| 011 | INTEST |
| — | reserved |
| 111 | BYPASS |

The BSR register is 30-bit wide, and its bits corresponds to the device pins in the following way:

| bit(s) | signal(s) | pin(s) | I/O | Comments |
|---|---|---|---|---|
| BSR[0] | rst_n | rst_n | I | Reset, active low |
| BSR[1] | clk | clk | I | Clock, active on rising edges |
| BSR[2] | rxd | ui_in[3] | I | UART input |
| BSR[6:3] | gpin[3:0] | ui_in[7:4] | I | General purpose inputs |
| BSR[14:7] | xdi[7:0] | uio_in[7:0] | I | Data bus when input (xoeb low) |
| BSR[22:15] | xdo[7:0] | uio_out[7:0] | O | Data bus when output (xoeb high) |
| BSR[23] | xbh | uo_out[0] | O | Byte high output (A[0]) |
| BSR[24] | xlal | uo_out[1] | O | Latch address low (A[9:2] at xdo[7:0]) |
| BSR[25] | xlah | uo_out[2] | O | Latch address high (A[17:10] at xdo[7:0]) |
| BSR[26] | txd | uo_out[4] | O | UART output |
| BSR[27] | xhh | uo_out[5] | O | Halfword high output (A[1]) |
| BSR[28] | xoeb | uo_out[6] | O | Output Enable, active low. Control for data bus direction |
| BSR[29] | xweb | uo_out[7] | O | Write Enable, active low |

Notice that the direction control for the bidirectional data bus is an already routed output pin (xoeb), and no extra control bits are thus required. But each bidirectional pin takes two BSR bits: one for input and another for output.

Also, the JTAG pins are excluded from the BSR. These are:

| Signal | I/O | Pin | Comments |
|---|---|---|---|
| TCK | I | ui_in[0] | |
| TMS | I | ui_in[1] | |
| TDI | I | ui_in[2] | |
| TDO | O | uo_out[3] | Signal shared with PWM output |

As we were short of output pins the TDO and PWMOUT signals use the same pin. PWMOUT is selected as long as the TAP is in the RESET or IDLE states, with TDO being switched in otherwise. This arrangement results in some noise in the PWM output during JTAG use, not a really big problem.