JTAG Instruction Register and Instruction Decoder

➢ Instruction Register (IR)

The purpose of Instruction Register is to shift in instruction through TDI and having the provision to store the instruction till a new instruction is fully shifted in.

Typically, an IR has two registers inside it as shown below. The Hold Register stores the previous instruction, and the Shift Register is used to shift-in the next instruction without affecting the previous instruction's execution.

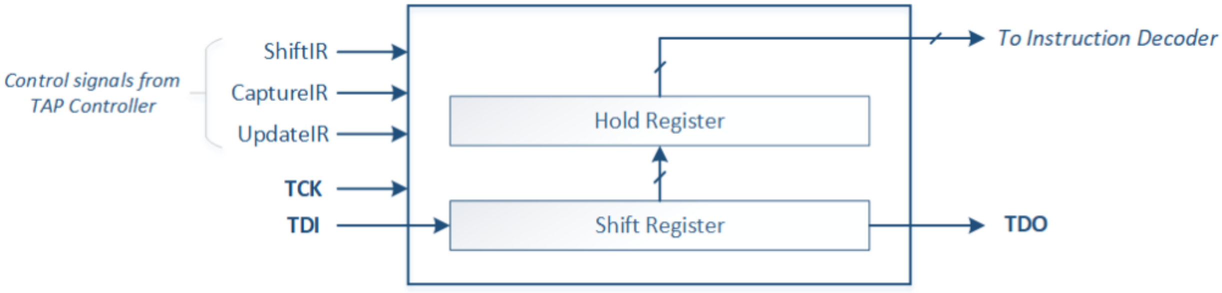

Figure 1: A top level view of Instruction Register (IR)

The control signals to the Instruction register originates from the TAP controller and depending upon the FSM state can either cause a shift-in/shift-out through the Shift Register (serial update operation in Shift-IR state) or cause the contents of the Shift Register to be passed across to the Hold Register (parallel update operation in Update-IR state).

Suppose we have a 4-bit instruction register and we want to load a value, say 1100 to it. Then we have to follow the steps below :

1. **Move to "Shift-IR" state from the initial "Test Logic Reset" state by applying:**

   TMS = 0 (1st clock)

   TMS = 1 (2nd clock)

   TMS = 1 (3rd clock)

   TMS = 0 (4th clock)

   TMS = 0 (5th clock)

2. **While in "Shift-IR" state, keep TMS = 0 for the next 3 clock cycles to serially shift-in the data (1100) through TDI.**

NOTE: 3 clock cycles to shift 4 bits data because the LSB (logic 0) of data 1100 was made available in the TDI port after 4th clock, thus in the 5th clock itself the LSB was loaded when the FSM entered the "Shift-IR" state. The MSB (logic 1) of data 1100 was shifted in 8th clock.

3. **Move to "Update-IR" state from the "Shift-IR" state by applying:**

TMS = 1 (9th clock)

TMS = 1 (10th clock)

When the FSM entered the "Update-IR" state, the data (1100) in the Shift Register was loaded to the Hold Register in parallel. Thus, the instruction is now made available for the instruction decoder.

➢ Instruction Decoder

The instruction from the Instruction Register (IR) is fed to a decoder logic, which selects the Data Register for JTAG operation. We assign a unique value (or opcode) to each and every Data Register in the JTAG. In order to select a Data Register, we load the IR with the corresponding opcode and the Instruction Decoder decodes the value and establishes an access path between the TDI/TDO and the required Data Register.

Suppose we have two Data Registers DR-1 and DR-2, and 4 bits Instruction Register in our JTAG. So, we will assign opcodes to the Data Registers and specify the same to the IR:

DR-1 (say opcode = 4'b0010)

DR-2 (say opcode = 4'b0011)

NOTE: 4 bits opcode, because the IR is 4 bits.

If we want to access DR-2 for any JTAG operation, then we have to load the instruction register with a value 4'b0011 and the instruction decoder will be able to decode this value to select the DR-2.