Position Statement: TAPs All Over My Chips

Steven F. Oakland
IBM Microelectronics Division
1000 River Street MS 863B
Essex Junction, Vermont 05452, USA

An increasing number of system-on-chip (SoC) application-specific integrated circuits (ASICs) have more than one embedded processor with test access port (TAP). A processor’s TAP facilitates a hardware interface to a software development/debug tool. Such tools exist for several different processors and embedded controllers such as the IBM PowerPC family of processors, the ARM family of processors, and digital signal processors (DSPs). Processor control features typical of such tools include run, start, step, set breakpoints, reset, and initialize. Typical low-level processor watch functions include displaying and modifying memory, registers, and cache.

When a hardware/software development tool has direct access to the TAP pins of the processor, the interface is straightforward. The development tool merely sequences the processor’s state machine to the Shift-IR and Update-IR states to load a debug instruction and then to the Capture-DR, Shift-DR, and Update-DR states to observe and/or control a specific debug register within the processor. However, when multiple processors are embedded in a single chip, challenges exist for the chip integrator and for developers and users of hardware/software debug tools.

The chip integrator needs a consistent method for providing access to all embedded TAPs. Few can afford to provide additional pins for accessing embedded TAPs.

Using component pins for selecting an embedded TAP also presents a challenge to providers of debug tools. The hardware interface to such tools includes connections to four or five TAP pins, but cannot be expected to include an unknown number of TAP selection pins.

On the other hand, users of debug tools are not willing to accept the performance degradation associated with a serial (daisy-chain) connection of embedded TAPs. Even though all embedded TAPs other than the one being debugged can be set up such that their bypass registers are selected, reducing the impact on data-shift operations, the time required for each instruction-shift operation increases proportionally to the number of TAPs in the daisy-chain.

Thus, users of debug tools are driving for a mechanism whereby a single TAP at a time can be accessed. Chip integrators and providers of debug tools require a standard, predictable hardware interface consisting of as few pins as possible -- preferably just the five TAP pins. Is there a solution? Yes there is.

A solution will be presented whereby embedded TAPs can be accessed one at a time via a single TAP consisting of four or five pins. No TAP selection pins are required, and the runtime performance of debug software is relatively independent of the number of embedded processors. A brief description of the solution follows:

The component has a chip-level TAP (for accessing the component’s boundary-scan and optional device identification registers) in addition to any number of embedded TAPs. The number and instruction-register lengths of all embedded TAPs are considered when determining the instruction-register length of the chip-level TAP. Specifically, the length of the chip-level instruction register is $K + L$, where $K$ is greater than or equal to the length of the longest instruction register among the embedded TAPs, and where $2^L$ is greater than or equal to the number of embedded TAPs. The $L$ low-order bits of the chip-level instruction register are used for selecting a TAP -- either the chip-level TAP or an embedded TAP -- and the remaining bits are shared with the selected TAP and used for decoding the instruction within that TAP. All unselected TAPs (except the chip-level TAP) are held in the Test-Logic-Reset state. Immediately after selecting an embedded TAP by loading the chip’s instruction register (when the chip-level TAP is in the Update-IR state), the chip’s TMS pin must either be held low for one TCK cycle or high for three TCK cycles, allowing the selected TAP to get into the same state as the chip-level TAP. As an embedded TAP is selected, its device identification (or bypass) register is selected, regardless of the contents of the chip-level instruction register. All subsequent instruction-register loads will be loaded into the selected TAP and the chip-level TAP.

This solution maintains compliance with IEEE Standard 1149.1 and enables one-at-a-time access to any embedded TAP without using TAP selection pins.