

Blackhawk™ Emulation

TECHNICAL ARTICLE

Using the Adaptive Clocking Feature of the TI OMAP™ Platform

Adding OMAP Adaptive Clocking support to TI JTAG Emulators

BHadaptiveClocking-TA-01
June 2005



Using the Adaptive Clocking Feature of the TI OMAP™ Platform

By Fadata Ltd. and Blackhawk – EWA Technologies, Inc.

Adding Adaptive Clocking Support to TI JTAG Emulators

What is Adaptive Clocking?

Adaptive clocking is a feature of synthesizable cores, introduced by ARM® Ltd. and adopted by TI in their OMAP platform, wherein the input test clock (TCK) is delayed (synchronized) before producing the resulting output clock (RTCK). During this synchronization period the target core samples Test Data In (TDI), Test Mode Select (TMS), and TCK with the core clock.

Emulators can adapt to the RTCK output of these adaptive clocking cores using a simple scheme. This scheme requires the emulator to generate the next TCK edge only after receiving an “acknowledge” signal from the target, indicating that it has received and processed the previous edge. The target will “acknowledge” these TCK edges by repeating them (after some delay) on its RTCK output. This protocol creates a natural, target-controlled “throttle” for the TCK rate. A positive side-effect to this mechanism is that any delays introduced by the JTAG cable are automatically taken into consideration.

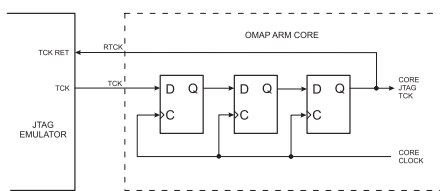


Figure 1: Adaptive Clocking Mechanism

Internally, the adaptive clocking mechanism is implemented as a multi-stage synchronizer consisting of several D-Triggers (Figure 1) synchronizing the TCK to the core clock of the target (as well as transferring all other JTAG signals to the core clock domain). The number of synchronizers, N_s , is device-dependent, but has shown to be a value of 3 or 4 in OMAP cores at this time. RTCK is the output from the last D-trigger and is typically connected to the TCK_RET pin of the JTAG connector.

The theoretical maximum TCK rate can be easily calculated as the frequency of the core clock divided by 2 times N_s (Equation 1).

$$f_{TCK}^{max} = \frac{f_{core}}{2N_s}$$

Equation 1: Maximum TCK

Handling adaptive clocking is not obligatory, so emulators that do not support it should still operate. They will, in general, just disregard the information on the RTCK pin.

The drawback of not using adaptive clocking will be the inability of the emulator to work at the optimum TCK rate. If the emulator supports a variable TCK, the user will have to manually set the maximum and sustainable TCK frequency through experimentation. And if the device’s core has a variable clock rate, this RTCK can change during operation.

Another disadvantage for fixed TCK emulators is that they will have trouble synchronizing the data with the adaptive clocking cores. For example, if the emulator has a 12 MHz TCK and the OMAP core is 24MHz, you’ll likely have data corruption.

Utilizing this feature adds convenience, stability, and increased performance, and is therefore strongly desired.

The Inverter Approach

The simplest solution is to have the emulator invert each RTCK edge to produce the next TCK edge (Figure 2).

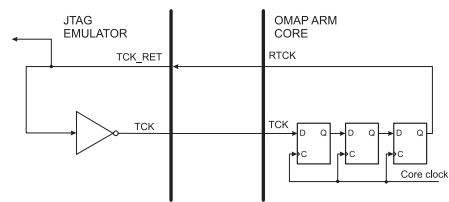


Figure 2: The Inverter Design

The main advantage of this approach is its ability to reach a TCK rate very close to the theoretical maximum. The downside,

however, is that the design is fully dependant on the target return clock and cannot function with non-adaptive clocking targets. This is because such targets usually have TCK_RET shorted to TCK on their JTAG connector, forcing the inverter to produce an unusable TCK (“infinite TCK problem”).

The Trigger-Inverter Approach

A solution to the “infinite TCK problem” is to have the emulator send the target an inverted version of RTCK as done in the inverter approach, but only after sampling it with its own clock, referred to here as ETCK (Figure 3). This method will make the emulator function with non-adaptive clocking targets, but the maximum output TCK will be determined by ETCK.

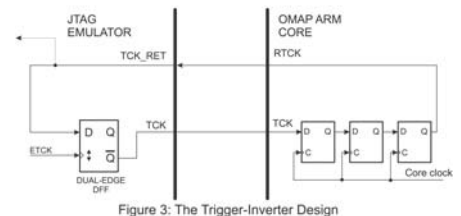


Figure 3: The Trigger-Inverter Design

In this approach, the sampling of RTCK is best performed by a dual-edge D-Flip Flop (DFF) to avoid halving the frequency of the resulting TCK, which would confuse those emulators that measure it. Due to the added delay imposed by the DFF, the maximum TCK rate will be somewhat lower, but will still be compatible with non-adaptive clocking targets, which is its major advantage.

The asynchronous relationship between ETCK and the target core clock introduces an inevitable jitter on the produced TCK. This jitter (with a maximum value of $0.5TETCK$) is harmless, and has the added benefit of introducing spread-spectrum modulation to TCK, lowering EMI levels and improving EMC.

JTAG Emulator Implementation

We implemented both designs using Blackhawk™ JTAG emulators (models USB510 and USB560) and tested them on a

TI OMAP5912 processor with an ARM926EJ-S core running at 192, 96, 48 and 12 MHz as the adaptive clocking target.

The emulators had their TCK-related logic (contained in a 7-ns Xilinx® Coolrunner™ or Coolrunner II CPLD) enhanced to accommodate the adaptive TCK support. These units have total delay, $t_D = 15$ ns (logic delay of 13 ns and line delay of 2 ns).

The maximum TCK for the OMAP5912 running at 192 MHz should have been $f_{CORE}/6$, or 32 MHz, but on accounting for the logic and line delays, we recalculated the actual maximum f_{TCK} of OMAP5912 to be 19.2 MHz.

The Inverter Approach produced a jitter-free TCK at 12 MHz.

The Trigger-Inverter Approach also produced a 12-MHz TCK along with the expected jitter. However, this design functioned correctly with non-adaptive clocking targets, achieving similar maximum TCK frequencies as non-adaptive clocking emulators.

Adapter Board Implementation

We also implemented the trigger-inverter approach in a commercially available add-on adapter board (Blackhawk Adaptive TCK) that enables adaptive clocking for any TI JTAG emulator lacking this capability (Figure 4).

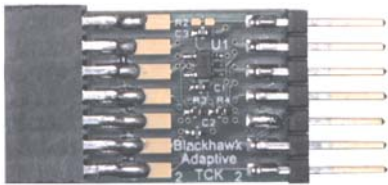


Figure 4: Blackhawk Adaptive TCK Adapter

The adapter board is simply inserted between the standard 14-pin connection of the emulator and target board and its operation is completely transparent to the emulator. Figure 5 shows the fundamental logic of the adapter.

Since fast low-voltage CMOS (LVC) logic was used for this implementation, the emulator reaches TCK frequency very close to the OMAP5912's theoretical maximum of 19.2 MHz. The characteristic jitter was also present. The adapter board also functions correctly with non-Adaptive TCK targets at frequencies as high as 64 MHz.

As seen in Figure 5, the add-on adapter board also addresses a test reset (TRST) issue found with XDS510™-class emulators built around the SN74ACT8990 Test Bus Controller (TBC) and certain OMAP targets. This includes the TI XDS510 ISA card emulator.

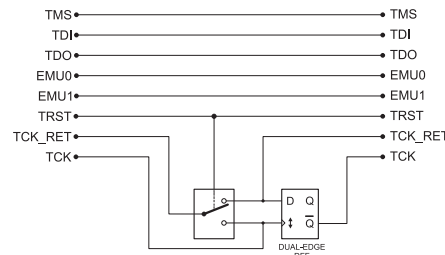


Figure 5: Blackhawk Adaptive TCK Architecture

The issue appears in emulators that are dependent on the TBC device and use one of its pins to handle the TRST signal. The problem occurs when some OMAP targets (e.g. OMAP DM320) stop their RTCK signal the moment TRST is asserted. Since the TBC is clocked by RTCK, it will halt, keeping TRST asserted indefinitely and producing a deadlock. The adapter board solves this condition by connecting the emulator's TCK to TCK_RET during assertion of TRST. This feature of the adapter board does not affect emulators that do not use a TBC pin to handle the TRST signal, such as the XDS560™.

Timing Analysis

Figure 6 shows TCK (upper waveform) and RTCK (lower waveform) on an adaptive clocking target being debugged by an emulator that does not support adaptive clocking. As the diagram shows, the emulator-supplied TCK is not adapted to the target RTCK speed requirements, forcing the latter to miss some of its edges.

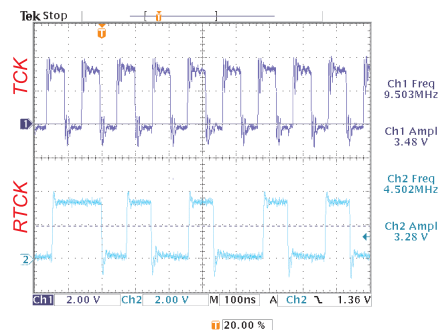


Figure 6: Adaptive TCK Target using a non-adaptive clocking emulator

Figure 7 shows the correct operation, which was achieved by attaching the same emulator

used in Figure 6 to the same target, but using an adaptive clocking add-on board. Now the TCK is throttled by RTCK to meet the requirements of the target.

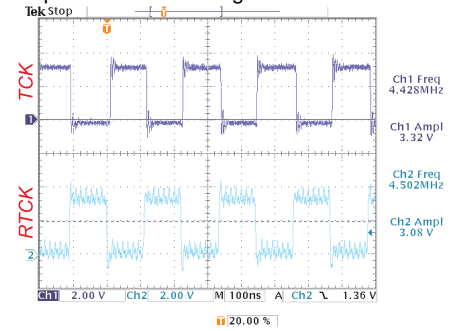


Figure 7: Adaptive TCK Target using an adaptive clocking emulator

Conclusion

Maximum JTAG performance and data integrity on today's TI OMAP platform is subject to handling the adaptive clocking feature of these cores.

The adaptive clocking feature allows for automatic setting of the optimal TCK rate in the system under test. However, support for this highly useful feature is lacking in JTAG emulators for TI DSPs currently on the market.

Our goal was to correct this deficiency, both by directly incorporating the adaptive clocking logic into Blackhawk™ JTAG emulators and by developing an add-on board that allows non-adaptive clocking emulators to handle adaptive clocking, thereby preserving current investments.

Written by:

Boycho Kostadinov, Ivailo Kassamakov, and Nenko Lazarov, Fadata Ltd.

Contributions by:

Andrey Kostov, Fadata Ltd.
Andrew Ferrari, EWA Technologies, Inc.



Blackhawk
123 Gaither Drive
Mount Laurel, NJ 08054-1701
Web: www.blackhawk-dsp.com