Router (ICEPick-D) Debug Configuration Information

Texas Instruments, Inc.

Version 0.1

Introduction

The following document provides enough information to enable customers to use a third party debugger on the processor JTAG test access ports (TAPs) connected to a TI router called ICEPick-D.

JTAG port description

The target debug interface uses the five standard IEEE 1149.1 (JTAG) signals (nTRST, TCK, TMS, TDI, and TDO), a return clock (RTCK) due to the clocking requirements of the ARM926 processor and the two TI extensions (EMU0, EMU1).

Pin	Type	Name	Description
nTRST	I	Test Logic Reset	*When asserted (active low) causes all test and debug logic in OMAP3430 to be reset along with the IEEE 1149.1 interface*
TCK	I	Test Clock	*This is the test clock used to drive an IEEE 1149.1 TAP state machine and logic. Depending on the emulator attached to OMAP3430 this is a free running clock or a gated clock depending on RTCK monitoring.*
RTCK	O	Returned Test Clock	*Synchronized TCK. Depending on the emulator attached to OMAP3430 the JTAG signals are clocked from RTCK or RTCK is monitored by the emulator to gate TCK.*
TMS	I	Test Mode Select	*Directs the next state of the IEEE 1149.1 test access port state machine*
TDI	I	Test Data Input	*Scan data input to the device*
TDO	O	Test Data Output	*Scan data output of the device*
EMU0	I/O	Emulation 0	*Channel 0 trigger – Boot mode*
EMU1	I/O	Emulation 1	*Channel 1 trigger – Boot mode*

The power domain containing the debug logic can be switched off in normal operating mode to reduce active power consumption. **Before starting the debugger, the debug power domain must be activated by applying a minimum of 100 (free running) TCK pulses to the device after nTRST is pulled high.**

The debug power domain will be activated when there is a return RTCK.

Initial Scan Chain Configuration

The first level of debug interface that sees the scan controller is the TAP router module. The debugger can configure the TAP router for serially linking up to 16 TAP controllers or individually scanning one of the TAP controllers without disrupting the IR state of the other TAPs.

The initial scan chain configuration of the device is determined from the level of the EMU0 and EMU1 pins upon the power-on-reset release. At power-on-reset, EMU0 and EMU1 are automatically configured as inputs.

The EMU0 and EMU1 pins should be pulled high at power-on-reset to configure the initial scan chain of the device to the TAP Router-only mode. In the TAP Router-only configuration, none of

the secondary TAPs are selected. The TAP router is the only TAP between the device level TDI and TDO. The Router TAP has an IR length of 6 bits. This is the recommended boot mode. The 3rd party debugger must assume that the TAP router is the only TAP between TDI and TDO at boot.

Adding a single TAP to the Scan Chain

The TAP router must be programmed to add additional TAPs to the scan chain. The following JTAG scans must be completed before the ARM926 (or DAP) is added to the scan chain.

Note: P is the 4-bit TAP port address.

Note: Q is the 4-bit core ID. It should be set to 0 for Netra/Centaurus/OMAP4 to communicate with the Cortex-A8/A9.
It is important that the JTAG run test idle state is not entered between Send only JTAG IR/DR scan to send data value '0xaP002108' and the JTAG IR/DR scan to send data value 'eQ002008'.

```
## Function  : Update the JTAG preamble and post-amble counts.
   Parameter : The IR pre-amble count is '0'.
   Parameter : The IR post-amble count is '0'.
   Parameter : The DR pre-amble count is '0'.
   Parameter : The DR post-amble count is '0'.
   Parameter : The IR main count is '6'.
   Parameter : The DR main count is '1'.

## Function  : Do a send-only JTAG IR/DR scan.
   Parameter : The route to JTAG shift state is 'shortest transition'.
   Parameter : The JTAG shift state is 'shift-ir'.
   Parameter : The JTAG destination state is 'pause-ir'.
   Parameter : The bit length of the command is '6'.
   Parameter : The send data value is '0x00000007'.
   Parameter : The actual receive data is 'discarded'.

## Function  : Do a send-only JTAG IR/DR scan.
   Parameter : The route to JTAG shift state is 'shortest transition'.
   Parameter : The JTAG shift state is 'shift-dr'.
   Parameter : The JTAG destination state is 'pause-dr'.
   Parameter : The bit length of the command is '8'.
   Parameter : The send data value is '0x00000089'.
   Parameter : The actual receive data is 'discarded'.

## Function  : Do a send-only JTAG IR/DR scan.
   Parameter : The route to JTAG shift state is 'shortest transition'.
   Parameter : The JTAG shift state is 'shift-ir'.
   Parameter : The JTAG destination state is 'pause-ir'.
   Parameter : The bit length of the command is '6'.
   Parameter : The send data value is '0x00000002'.
   Parameter : The actual receive data is 'discarded'.

## Function  : Do a send-only JTAG IR/DR scan.
   Parameter : The route to JTAG shift state is 'shortest transition'.
```

```
   Parameter : The JTAG shift state is 'shift-dr'.
   Parameter : The JTAG destination state is 'pause-dr'.
   Parameter : The bit length of the command is '32'.
   Parameter : The send data value is '0xaPP002108'.
   Parameter : The actual receive data is 'discarded'.

## Function   : Do a send-only JTAG IR/DR scan
   Parameter : The route to JTAG shift state is ‘shortest transition’
   Parameter : The JTAG shift state is ‘shift-dr’
   Parameter : The JTAG destination state is ‘pause-dr’
   Parameter : The bit length of the command is ‘32’
   Parameter : The send data value is ‘0xeQQ002008’
   Parameter : The actual receive data is ‘disgarded’

## Function  : Do a send-only all-ones JTAG IR/DR scan.
   Parameter : The JTAG shift state is 'shift-ir'.
   Parameter : The JTAG destination state is 'run-test/idle'.
   Parameter : The bit length of the command is '6'.
   Parameter : The send data value is 'all-ones'.
   Parameter : The actual receive data is 'discarded'.

## Function  : Wait for a minimum number of TCLK pulses.
   Parameter : The count of TCLK pulses is '10'.

## Function  : Update the JTAG preamble and post-amble counts.
   Parameter : The IR pre-amble count is '0'.
   Parameter : The IR post-amble count is '6'.
   Parameter : The DR pre-amble count is '0'.
   Parameter : The DR post-amble count is '1'.
   Parameter : The IR main count is '4'.
   Parameter : The DR main count is '1'.
```

Once a debug TAP is added to the scan chain, the debugger can communicate with that debug TAP by bypassing the ROUTER.

Adding multiple TAPs to the Scan Chain

The TAP router must be programmed to add additional TAPs to the scan chain. The following JTAG scans must be completed before multiple debug TAPs can be added to the scan chain.

Note: P is the 4-bit port address

```
## Function  : Update the JTAG preamble and post-amble counts.
   Parameter : The IR pre-amble count is '0'.
   Parameter : The IR post-amble count is '0'.
   Parameter : The DR pre-amble count is '0'.
   Parameter : The DR post-amble count is '0'.
   Parameter : The IR main count is '6'.
   Parameter : The DR main count is '1'.

## Function  : Do a send-only JTAG IR/DR scan.
   Parameter : The route to JTAG shift state is 'shortest transition'.
   Parameter : The JTAG shift state is 'shift-ir'.
   Parameter : The JTAG destination state is 'pause-ir'.
```

```
   Parameter : The bit length of the command is '6'.
   Parameter : The send data value is '0x00000007'.
   Parameter : The actual receive data is 'discarded'.

## Function  : Do a send-only JTAG IR/DR scan.
   Parameter : The route to JTAG shift state is 'shortest transition'.
   Parameter : The JTAG shift state is 'shift-dr'.
   Parameter : The JTAG destination state is 'pause-dr'.
   Parameter : The bit length of the command is '8'.
   Parameter : The send data value is '0x00000089'.
   Parameter : The actual receive data is 'discarded'.

## Function  : Do a send-only JTAG IR/DR scan.
   Parameter : The route to JTAG shift state is 'shortest transition'.
   Parameter : The JTAG shift state is 'shift-ir'.
   Parameter : The JTAG destination state is 'pause-ir'.
   Parameter : The bit length of the command is '6'.
   Parameter : The send data value is '0x00000002'.
   Parameter : The actual receive data is 'discarded'.
```

Repeat for each TAP:

```
## Function  : Do a send-only JTAG IR/DR scan.
   Parameter : The route to JTAG shift state is 'shortest transition'.
   Parameter : The JTAG shift state is 'shift-dr'.
   Parameter : The JTAG destination state is 'pause-dr'.
   Parameter : The bit length of the command is '32'.
   Parameter : The send data value is '0xaP002108'.
   Parameter : The actual receive data is 'discarded'.

// Comment: The Run Test IDLE state should not be entered
//          in this routine. The Run Test IDLE state entry
//          causes IR/DR to be updated.
```

End Repeat

```
## Function  : Do a send-only all-ones JTAG IR/DR scan.
   Parameter : The JTAG shift state is 'shift-ir'.
   Parameter : The JTAG destination state is 'run-test/idle'.
   Parameter : The bit length of the command is '6'.
   Parameter : The send data value is 'all-ones'.
   Parameter : The actual receive data is 'discarded'.

## Function  : Wait for a minimum number of TCLK pulses.
   Parameter : The count of TCLK pulses is '10'.

## Function  : Update the JTAG preamble and post-amble counts.
   Parameter : The IR pre-amble count is '0'.
   Parameter : The IR post-amble count is '6'.
   Parameter : The DR pre-amble count is '0'.
   Parameter : The DR post-amble count is '1'.
```

Example: Coresight DAP (or ARM9xx) added to the scan chain.

```
   Parameter : The IR main count is '4'.
```

```
    Parameter : The DR main count is '1'.

## When multiple debug TAPs are added to the scan chain, the one with the highest
## port address is closest to the device TDO.
## The router is closest to the device TDI
```

Once multiple debug TAPs are added to the scan chain, the debugger can communicate with those debug TAP by bypassing the ROUTER.

When does the tap insertion sequence need to be re-executed?

The tap insertion sequence needs to be re-executed whenever the JTAG tap controller enters the Test Logic Reset state either synchronously (based upon TAP controller state transitions) or asynchronously (based upon an assertion of nTRST).