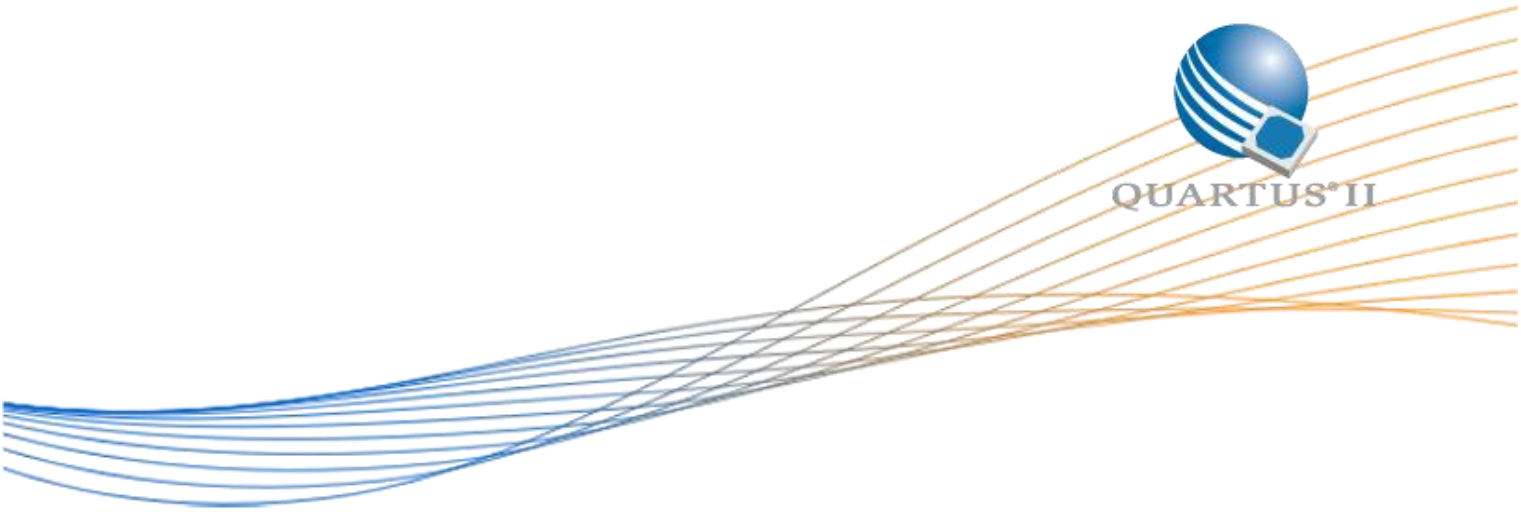


MAX10 JTAG Secure Unlock

Date: 30/3/2016

Revision:1.0

This design example shows how to use MAX10 JTAG core atom (Internal JTAG Interface) to unlock/lock external JTAG when the design is enable JTAG secure mode. To enable JTAG Secure mode, please contact Altera Support.

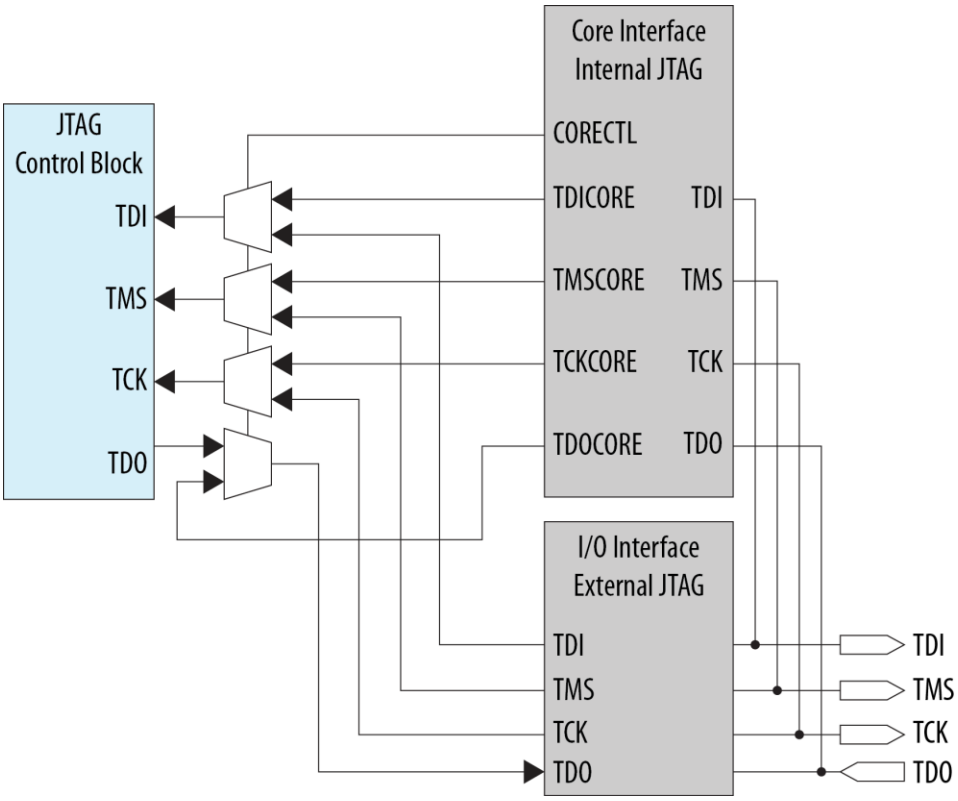
Internal JTAG Interface

There are two interfaces to access the JTAG control block in MAX10 FPGAs—external JTAG interface and internal JTAG interface.

The external JTAG interface refers to access from the physical JTAG pins—TCK, TDI, TDO, and TMS—to the JTAG control block. You use the external JTAG interface for FPGA configuration when using JTAG configuration scheme via programming cables.

The internal JTAG interface refers to the connection of TCK, TDI, TDO, and TMS signals from the internal FPGA core fabric to the JTAG control block. You can only access the JTAG control block using either external or internal JTAG interface one at a time. For example, when you use the internal JTAG interface, the external JTAG interface to the JTAG control block is disabled. To access the internal JTAG interface, you must include the WYSIWYG atom in your Quartus Prime design.

Figure 1: Internal and External JTAG Interface Connection



Note: To make sure the internal JTAG of MAX10 FPGA function correctly, all four JTAG signals (TCK, TDI, TMS and TDO) in the JTAG WYSIWYG atom need to be routed out and let Quartus Prime software to auto assign the ports to their corresponding dedicated JTAG pins.

Accessing Internal JTAG Block for JTAG Secure Unlock

The following example shows how the input and output ports of a WYSIWYG atom are defined in the MAX 10 Device.

```
fiftyfivenm_jtag <name>
(
    .tms(),
    .tck(),
    .tdi(),
    .tdoutap(),
    .tdouser(),
    .tdicore(),
    .tmscore(),
    .tckcore(),
    .corectl(),
    .tdo(),
    .tmsutap(),
    .tckutap(),
    .tdiutap(),
    .shiftuser(),
    .clkdruser(),
    .updateuser(),
    .runidleuser(),
    .usr1user(),
    .tdocore(),
    .ntdopinena()
);
```

Table 1: Ports Description

Ports	Input/Output	Functions
<name>	-	Identifier for the MAX10 JTAG WYSIWYG atom and represents any identifier name that is legal for the given description language, such as Verilog HDL, VHDL, and AHDL.
.corectl()	Input	Active high input to the JTAG control block to enable the internal JTAG access from core interface. When the FPGA enters user mode after configuration, this port is low by default. Pulling this port to logic high will enable the internal JTAG interface (with external JTAG interface disabled at the same time) and pulling this port to logic low will disable the internal JTAG interface (with external JTAG interface enabled at the same time).
.tckcore()	Input	Core tck signal
.tdicore()	Input	Core tdi signal
.tmscore()	Input	Core tms signal
.tdocore()	Output	Core tdo signal
.tck()	Input	Pin tck signal
.tdi()	Input	Pin tdi signal
.tms()	Input	Pin tms signal

.tdo()	Output	Pin tdo signal
.clkdruser()	Input/Output	These ports are not used for enabling the JTAG secure mode using the internal JTAG interface, hence you can leave them unconnected.
.runidleuser()		
.shiftuser()		
.tckutap()		
.tdiutap()		
.tdouser()		
.tdoutap()		
.tmsutap()		
.updateuser()		
.usr1user()		
.ntdopinena()		

Design Example for MAX10 JTAG Secure Mode

This design example demonstrates the instantiation of internal JTAG WYSIWYG atom and shows the example of user logic implementation in the Quartus Prime software to execute the LOCK and UNLOCK JTAG instructions. This reference design is targeted on the MAX10 device with the JTAG Secure Mode enabled.

LOCK and UNLOCK JTAG Instructions

When you configure this reference design into a MAX10 device with the JTAG Secure Mode enabled, the MAX10 device is in JTAG secure mode after power up and configuration, whereby you can only execute mandatory JTAG instructions.

To disable the JTAG secure mode, you can trigger the start_unlock port of the user logic to issue the UNLOCK JTAG instruction. After the start_unlock port goes high, the UNLOCK JTAG instruction is issued. After the UNLOCK JTAG instruction is issued, the device exits from JTAG secure mode, whereby both mandatory and non-mandatory JTAG instructions are allowed. After UNLOCK JTAG, you can choose to full chip erase the internal flash of MAX10 device to disable JTAG secure mode permanently, since JTAG secure mode is just one of the ICB settings.

The start_lock port in the user logic triggers the execution of the LOCK JTAG instruction. The function of the LOCK JTAG instruction is to put the device back into JTAG secure mode.

Figure 2: LOCK and UNLOCK JTAG Instruction Execution

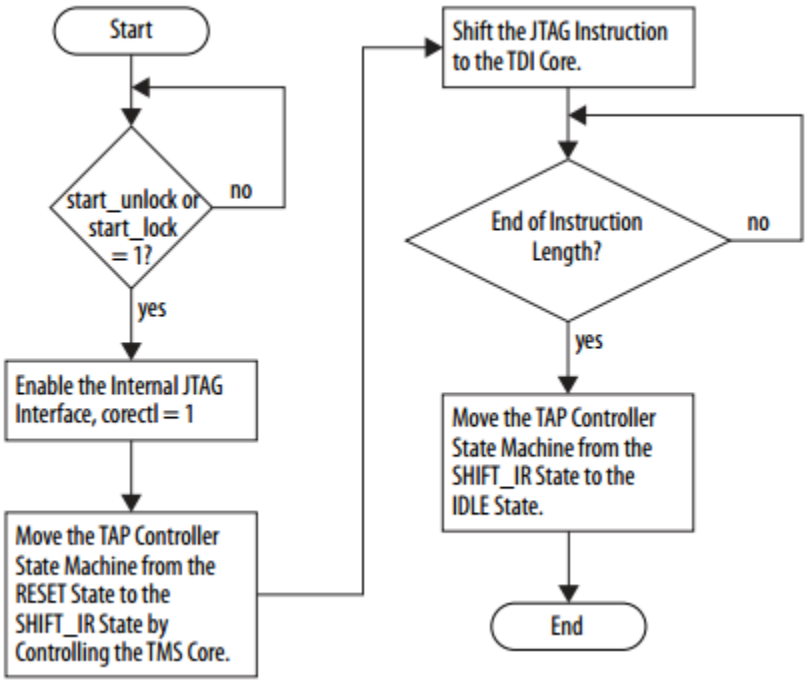


Table 2: Input and Output Port of the User Logic

Port	Input/Output	Function
clk_in	Input	Clock source for the user logics. The fMAX of the user logic depends on the timing closure analysis. You need to apply timing constraint and perform timing analysis on the path to determine the fMAX.
start_lock	Input	Logic high to trigger the execution of the LOCK JTAG instruction to the internal JTAG interface.
start_unlock	Input	Logic high to trigger the execution of the UNLOCK JTAG instruction to the internal JTAG interface.
jtag_core_en_out	Output	Output to the JTAG WYSIWYG atom. This port is connected to the corectl port of the JTAG WYSIWYG atom to enable the internal JTAG interface.
tck_out	Output	Output to the JTAG WYSIWYG atom. This port is connected to the tck_core port of the JTAG WYSIWYG atom.
tdi_out	Output	Output to the JTAG WYSIWYG atom. This port is connected to the tdi_core port of the JTAG WYSIWYG atom.
tms_out	Output	Output to the JTAG WYSIWYG atom. This port is connected to the tms_core port of the JTAG WYSIWYG atom.
indicator	Output	Logic high of this output pin indicates the completion of the LOCK or UNLOCK JTAG instruction execution.

Verify JTAG Secure Mode

Altera recommends that you verify whether your device has successfully entered or exit JTAG secure mode by executing the non-mandatory JTAG instructions. To validate the JTAG secure mode with the reference design, follow these steps:

1. FPGA power up
   After the FPGA is powered up, the FPGA is in the JTAG secure mode because the JTAG Secure Mode has been set in ICB settings.
2. FPGA configuration
   Configure the reference design into the FPGA. To ensure the device enters user mode successfully, you can check the CONFDONE pin or observe the counter_output pin. If the device enters user mode successfully, the CONFDONE pin goes high and the counter_output pin should toggle.
3. Verify the JTAG secure mode
   After the device enters user mode, issue the PULSE_NCONFIG JTAG instruction using the external JTAG pins. You can use the pulse_ncfg.jam file attached in the design example. To execute the pulse_ncfg.jam file, you can use the quartus_jli or the JAM player. The PULSE_NCONFIG JTAG instruction triggers device reconfiguration. If your device is in the JTAG secure mode, reconfiguration is not taking place because the PULSE_NCONFIG JTAG instruction is a non-mandatory JTAG instruction. You can confirm this by observing the CONFDONE pin and the counter_output pin. If reconfiguration did not take place, the CONFDONE pin stays high and the counter_output pin continues to toggle.
4. Execute the UNLOCK JTAG instruction
   Pull the start_unlock port of the user logic to logic high. After the UNLOCK JTAG instruction is complete, the indicator port goes high.
5. Verify the JTAG secure mode
   After the UNLOCK JTAG instruction is completed, issue the PULSE_NCONFIG JTAG instruction again using the external JTAG pins. If your device is not in the JTAG secure mode, the PULSE_NCONFIG JTAG instruction triggers device reconfiguration. You can observe the CONFDONE pin and the counter_output pin to monitor the device reconfiguration. The CONFDONE pin goes from high to low and the counter_output pin stops toggling during device reconfiguration.