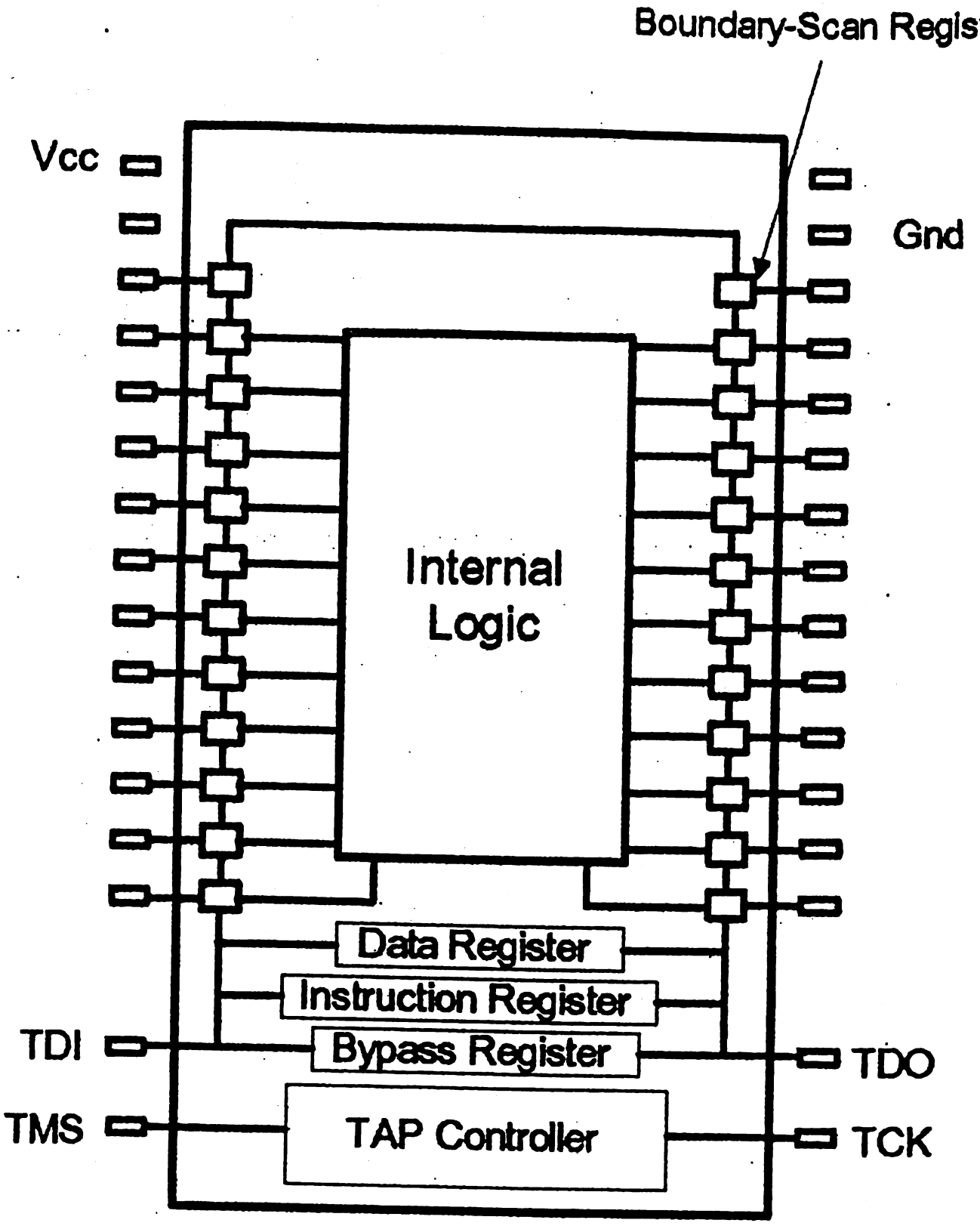

Boundary-Scan Register
Vcc
Gnd
Internal Logic
Data Register
Instruction Register
Bypass Register
TDI
TDO
TMS
TAP Controller
TCK

XAPP139 (v1.7) February 14, 2007

# Configuration and Readback of Virtex FPGAs Using JTAG Boundary-Scan


## Summary

This application note demonstrates using a Boundary-Scan (JTAG) interface to configure and read back Virtex™ FPGA devices. Virtex devices have Boundary-Scan features that are compatible with IEEE Standard 1149.1. This application note is a complement to the configuration section in the Virtex data sheet and application note **XAPP138 "Virtex FPGA Series Configuration and Readback."** Xilinx recommends reviewing both the data sheet and XAPP138 prior to reading this document.

***Note:*** The information in this application note also applies to the Virtex-E FPGA family.


## Introduction

The IEEE 1149.1 Test Access Port (TAP) and Boundary-Scan architecture, commonly referred to as JTAG, is a popular testing method. JTAG is an acronym for the Joint Test Action Group, the technical subcommittee initially responsible for developing the standard. This standard provides a means to ensure the integrity of individual board-level components and their interconnections. With increasingly dense multi-layer PC boards and more sophisticated surface mounting techniques, Boundary-Scan testing is becoming widely used as an important debugging standard.

Devices containing Boundary-Scan logic can send data out on I/O pins in order to test connections between devices at the board level. The circuitry can also be used to send signals internally to test the device specific behavior. These tests are commonly used to detect opens and shorts at both the board and device level.

In addition to testing, Boundary-Scan offers the flexibility for a device to have its own set of user-defined instructions. The added common vendor-specific instructions, such as configure and verify, have increased the popularity of Boundary-Scan testing and functionality.

## Boundary-Scan for Virtex Devices

The Virtex family is fully compliant with the IEEE Standard 1149.1 Test Access Port and Boundary-Scan architecture. The architecture includes all mandatory elements defined in the IEEE 1149.1 Standard. These elements include the TAP, the TAP controller, the instruction register, the instruction decoder, the Boundary-Scan register, and the bypass register. The Virtex family also supports some optional instructions – the 32-bit identification register and a configuration register in full compliance with the standard. Outlined in the following sections are the details of the JTAG architecture for Virtex devices.

## Test Access Port

The Virtex TAP contains four mandatory dedicated pins as specified by the protocol (Table 1).

*Table 1:* **Virtex TAP Controller Pins**

| Pin | Description |
|---|---|
| TDI | Test Data In |
| TDO | Test Data Out |
| TMS | Test Mode Select |
| TCK | Test Clock |

Three input pins and one output pin control the IEEE 1149.1 Boundary-Scan TAP controller. In addition to the required pins, there are optional control pins such as $\overline{\text{TRST}}$ (Test Reset) and enable pins, which can be found on devices from other manufacturers. Be aware of these optional signals when interfacing Xilinx devices with devices from different vendors because these signals can need to be driven. (To determine the set of signals that must be driven to enable IEEE 1149.1 compliance, see the vendor documentation for each device on the Boundary-Scan chain.)

The TAP controller is a 16-state state machine (Figure 1). Mandatory TAP pins are as follows:

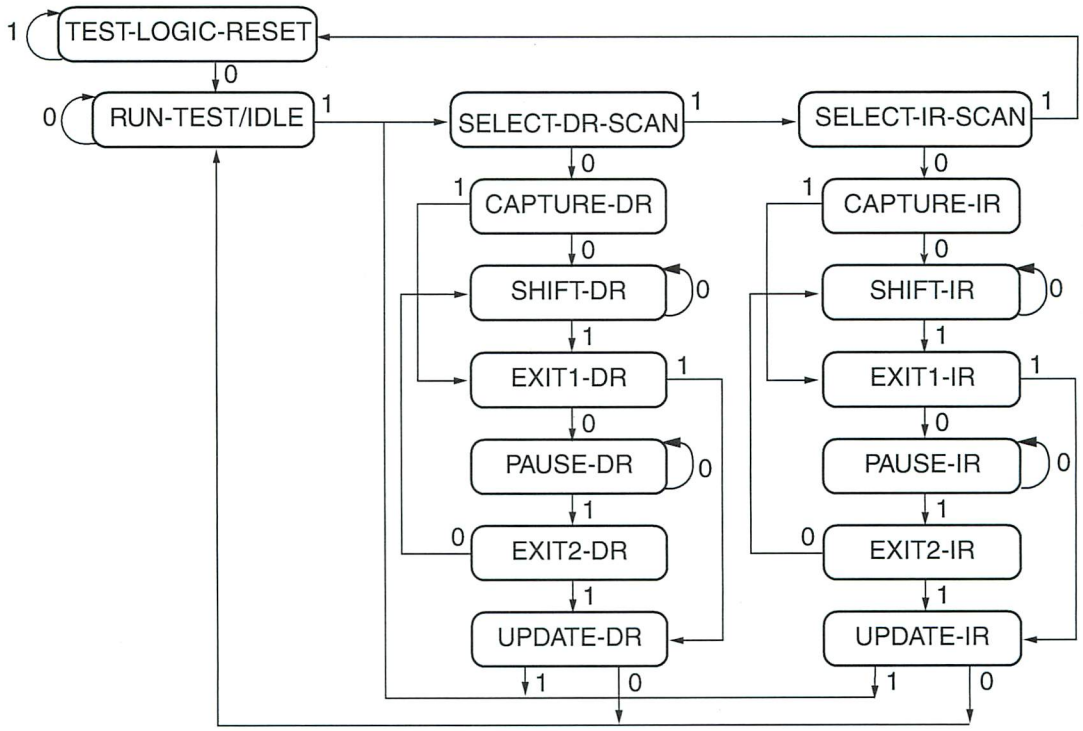
- TMS - The sequence of states through the TAP controller is determined by the state of the TMS pin on the rising edge of TCK. TMS has an internal resistive pull-up to provide a logic High if the pin is not driven.
- TCK - This pin is the JTAG test clock. It sequences the TAP controller and the JTAG registers in the Virtex devices.
- TDI - This pin is the serial input to all JTAG instruction and data registers. The state of the TAP controller and the current instruction held in the instruction register determine which register is fed by the TDI pin for a specific operation. TDI has an internal resistive pull-up to provide a logic High to the system if the pin is not driven. TDI is applied into the JTAG registers on the rising edge of TCK.
- TDO - This pin is the serial output for all JTAG instruction and data registers. The state of the TAP controller and the current instruction held in the instruction register determine which register (instruction or data) feeds TDO for a specific operation. TDO changes state on the falling edge of TCK and is active only during the shifting of instructions or data through the device. This pin is placed in a 3-state condition at all other times.

***Note:*** As specified by the IEEE Standard, the TMS and TDI pins all have internal pull-ups. These internal pull-ups of 50-150 kΩ are active regardless of the mode selection.

When using the Boundary-Scan operations in Virtex devices, the $V_{CCO}$ for Bank 2 must be at 3.3V for the TDO pin to operate at the required LVTTL level.

## TAP Controller

Figure 1 diagrams a 16-state finite state machine. The four TAP pins control how the data is scanned into the various registers. The state of the TMS pin at the rising edge of the TCK determines the sequence of state transitions. There are two main sequences, one for shifting data into the data register and the other for shifting an instruction into the instruction register.




Note: The value shown adjacent to each state transition in this figure represents the signal present at TMS at the time of a rising edge at TCK.

x139_01_011207

*Figure 1:* **State Diagram for the TAP Controller**

## Boundary-Scan Instruction Set

To determine the operation to be invoked, a 5-bit instruction is loaded into the instruction register. Table 2 lists the available Boundary-Scan instructions for Virtex devices.

*Table 2:* **Virtex Boundary-Scan Instructions**

| Boundary-Scan Command | Binary Code (4:0) | Description |
|---|---|---|
| EXTEST | 00000 | Enables Boundary-Scan EXTEST operation |
| SAMPLE | 00001 | Enables Boundary-Scan SAMPLE operation |
| USER1 | 00010 | Access user-defined register 1 |
| USER2 | 00011 | Access user-defined register 2 |
| CFG_OUT | 00100 | Access the configuration bus for readback |
| CFG_IN | 00101 | Access the configuration bus for configuration |
| INTEST | 00111 | Enables Boundary-Scan INTEST operation |

*Table 2:* **Virtex Boundary-Scan Instructions *(Continued)***

| Boundary-Scan Command | Binary Code (4:0) | Description |
|---|---|---|
| USERCODE | 01000 | Enables shifting out user code |
| IDCODE | 01001 | Enables shifting out of ID code |
| HIGHZ | 01010 | Places output pins in a 3-states condition while enabling the bypass register |
| JSTART | 01100 | Clocks the start-up sequence when StartClk is TCK |
| BYPASS | 11111 | Enables BYPASS |
| RESERVED | All other codes | Xilinx reserved instructions |

The mandatory IEEE 1149.1 commands are supported in Virtex devices along with several Xilinx vendor-specific commands. Virtex devices have a powerful command set. The EXTEST, INTEST, SAMPLE/PRELOAD, BYPASS, IDCODE, USERCODE, and HIGHZ instructions are all included. The TAP also supports two internal user-defined registers (USER1 and USER2) and configuration/readback of the device. The Virtex Boundary-Scan operations are independent of the mode selection. The Boundary-Scan mode in Virtex devices overrides the other mode selections. For this reason, Boundary-Scan instructions using the Boundary-Scan register (SAMPLE/PRELOAD, INTEST, EXTEST) must not be performed during configuration. All instructions except USER1 and USER2 are available before the Virtex device is configured. After configuration, all instructions are available.

JSTART is an instruction specific to the Virtex architecture and configuration flow. As described in Table 2, the JSTART instruction clocks the startup sequence when the appropriate bitgen option is selected. The instruction does not work correctly without the correct bitgen option selected.

```
bitgen -g startupclk:jtagclk designName.ncd
```

For details on the standard Boundary-Scan instructions, EXTEST, INTEST, and BYPASS, refer to the IEEE Standard. The user-defined registers (USER1/USER2) are described in a later section of this application note.

## Boundary-Scan Architecture

Virtex devices have several registers including all registers required by the IEEE 1149.1. In addition to the standard registers, the family contains optional registers for simplified testing and verification (Table 3).

*Table 3:* **Virtex JTAG Registers**

| Register Name | Register Length | Description |
|---|---|---|
| Instruction register | 5 bits | Holds current instruction OPCODE and captures internal device status |
| Boundary-Scan register | 3 bits per I/O | Controls and observes input, output, and output enable |
| Bypass register | 1 bit | Device bypass |
| Identification register | 32 bits | Captures device ID |
| JTAG configuration register | 32 bits | Allows access to the configuration bus when using the CFG_IN or CFG_OUT instructions |
| USERCODE register | 32 bits | Captures user-programmable code |

### Single Device Configuration

***Note:*** Refer to **XAPP058** for the recommended embedded solution.

To configure a Virtex part as a single device through Boundary-Scan operations, use the steps listed in Table 8, which lists and describes the TAP controller commands required to configure a Virtex device. Ensure the bitstream is generated with the JTAG clock option:

```
bitgen -g startupclk:jtagclk designName.ncd
```

Also, when programming with iMPACT software, verify that the most current version of software is used. Refer to Figure 1 for the TAP controller states. These TAP controller commands are issued automatically if configuring the part with the iMPACT software.

*Table 8:* **Single Device Configuration Sequence**

| | TAP Controller Step Description | Set and Hold | | Number of Clocks |
|---|---|---|---|---|
| | | TDI | TMS | TCK |
| 1 | On power-up, place a "1" on the TMS and clock the TCK five times. (This ensures starting in the TLR (Test-Logic-Reset) state.) | X | 1 | 5 |
| 2 | Move into the RTI state. | X | 0 | 1 |
| 3 | Move into the SELECT-IR state. | X | 1 | 2 |
| 4 | Enter the SHIFT-IR state. | X | 0 | 2 |
| 5 | Start loading the CFG_IN instruction .(1) | 0101 | 0 | 4 |
| 6 | Load the last bit of CFG_IN instruction when exiting SHIFT-IR (defined in the IEEE standard). | 0 | 1 | 1 |
| 7 | Enter the SELECT-DR state. | X | 1 | 2 |
| 8 | Enter the SHIFT-DR state. | X | 0 | 2 |
| 9 | Shift in the Virtex bitstream. ($bit_N$ (MSB) is the first bit in the bitstream (1)) | $bit_1$... $bit_N$ | 0 | (Number of bits in bitstream) -1 |
| 10 | Shift in the last bit of the bitstream. ($bit_0$ (LSB) is shifted on the transition to EXIT1-DR) | $bit_0$ | 1 | 1 |
| 11 | Enter UPDATE-DR state. | X | 1 | 1 |
| 12 | Enter the SELECT-IR state. | X | 1 | 2 |
| 13 | Move to the SHIFT-IR state. | X | 0 | 2 |
| 14 | Start loading the JSTART instruction.(1) (The JSTART instruction initializes the startup sequence.) | 1100 | 0 | 4 |
| 15 | Load the last bit of the JSTART instruction. | 0 | 1 | 1 |
| 16 | Move to the SELECT-DR state. | X | 1 | 2 |
| 17 | Move to SHIFT-DR and clock the STARTUP sequence. (by applying a minimum of 12 clock cycles to the TCK). | X | 0 | ≥14 |
| 18 | Move to the UPDATE-DR state. | X | 1 | 2 |
| 19 | Return to the RTI state. (The device is now functional). | X | 0 | 1 |

***Note:***

1. In the TDI column, the right-most bit is shifted in first.

# ispDOWNLOAD® Cables

For Programming Lattice ISP™ Devices

November 2007

## Features

- **Support for All Lattice Programmable Products**
  - 1.2V to 5V programming
  - Ideal for design prototyping and debugging
- **Connect to Multiple PC Interfaces**
  - USB (v.1.0, v.2.0)
  - PC Parallel Port
- **Easy-to-Use Programming Connectors**
  - Versatile flywire, 2 x 5 (.100") or 1 x 8 (.100") connectors
  - 6 feet (2 meters) or more of programming cable length (PC to DUT)
- **Lead-Free/RoHS Compliant Construction**

## ispDOWNLOAD Cables

Lattice ispDOWNLOAD cables are designed to facilitate in-system programming for all Lattice Semiconductor ISP™ devices directly from a PC. With in-system programmability, hardware functions can be programmed and modified in real-time on the system board to give additional product features, shorten system design and debug cycle time, enhance product manufacturability and simplify field upgrades.

After you complete your logic design and create a programming file with the ispLEVER® development tools, you can use ispVM® System software to program devices on your board. The ispVM System software automatically generates the appropriate ISP commands, programming addresses and programming data based on information stored in the programming file and parameters you set in ispVM. Programming signals are then generated from the USB or parallel port of a PC and directed through the ispDOWNLOAD Cable to the device, no additional components are required for programming.

ispVM System software is included with all Lattice design tool products and available for download from the Lattice web site at www.latticesemi.com.

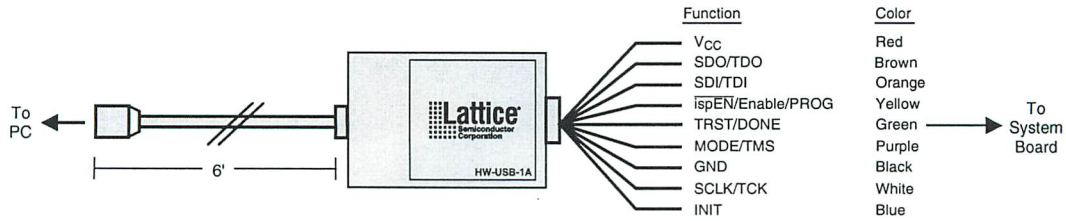
## ispDOWNLOAD Cable Pin Definitions

The functions provided by of the ispDOWNLOAD cables correspond with available functions on Lattice programmable devices. Since some devices contain different programming features, the specific functions provided by the ispDOWNLOAD cable may depend on the selected target device. ispVM System software will automatically generate the appropriate functions based on the selected device. See Table 1 for an overview of the ispDOWNLOAD cable functions.

***Table 1. ispDOWNLOAD Cable Pin Definitions***

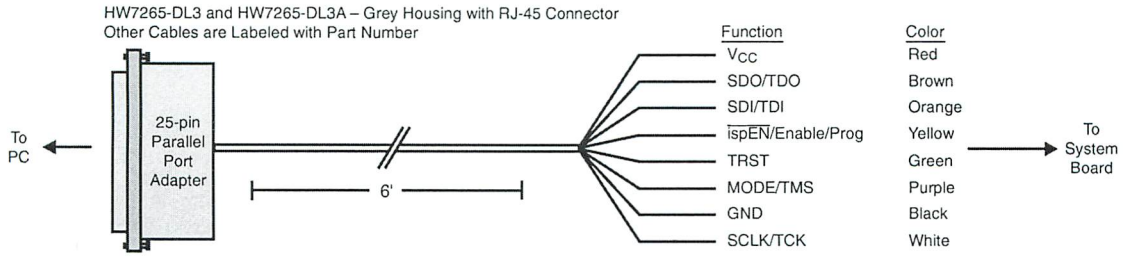
| ispDOWNLOAD Cable Pin | Name | ispDOWNLOAD Cable Input/Output | Description |
|---|---|---|---|
| VCC | Programming Voltage | Input | Connect to $V_{CC}$ or $V_{CCJ}$ plane of the target device. Typical $I_{CC}$ = 10mA. (Note: this may not be the same as a target device's $V_{CCO}$ plane). |
| SDO/TDO | Test Data Output | Input | Used to shift data out via the IEEE1149.1 (JTAG) programming standard. |
| SDI/TDI | Test Data Input | Output | Used to shift data in via the IEEE1149.1 programming standard. |
| $\overline{\text{ispEN}}$/Enable/ PROG | Enable | Output | Enable device to be programmed. |
| TRST | Test Reset | Output | Optional IEEE 1149.1 state machine reset. |
| DONE | DONE | Input | Done indicates status of configuration |
| MODE/TMS | Test Mode Select Input | Output | Used to control the IEEE1149.1 state machine. |
| GND | Ground | Input | Connect to ground plane of the target device |
| SCLK/TCK | Test Clock Input | Output | Used to clock the IEEE1149.1 state machine |
| INIT | Initialize | Input | Indicates that ORCA device is ready for configuration. |

***Figure 1. ispDOWNLOAD Cable In-System Programming Interface for the PC (HW-USB-1A or HW-USB-2A)[1]***




1. Lattice PAC-Designer® software does not support programming with USB cables. To program ispPAC devices with these cables, use the ispVM System software.

***Figure 2. ispDOWNLOAD Cable In-System Programming Interface for the PC[1]***




1. HW7265-DL3, HW7265-DL3A, HW-DL-3B, HW-DL-3C and HW-DLN-3C are functionally equivalent products.

***Figure 3. ispDOWNLOAD Cable In-System Programming Interface for the PC (pDS4102-DL2 or pDS4102-DL2A)***

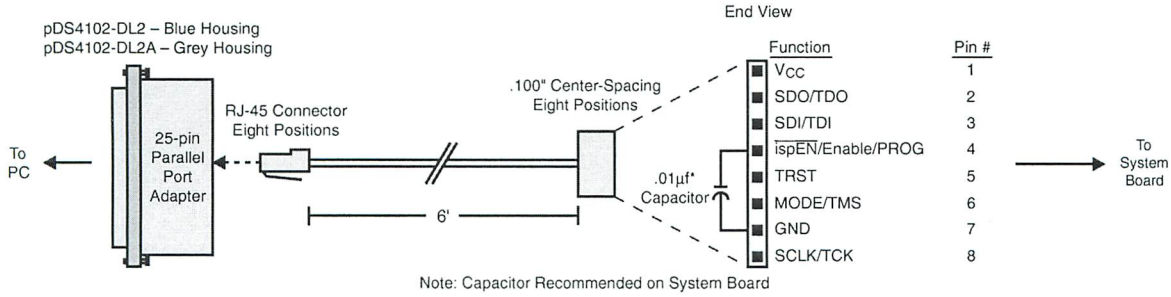
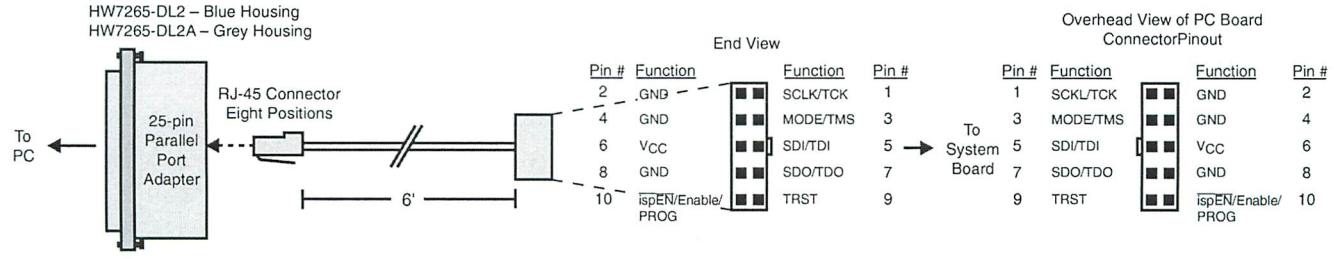



***Figure 4. ispDOWNLOAD Cable In-System Programming Interface for the PC (HW7265-DL2 or HW7265-DL2A)***




## Programming Software

ispVM System is the preferred programming management software tool for all Lattice devices and download cables. The latest version of ispVM System is always available for download from the Lattice web site at www.latticesemi.com/software.

PAC-Designer is the design tool for Lattice ispPAC and ispCLOCK devices. PAC-Designer can also be used to program these devices. If using PAC-Designer for programming, pay special attention to the notes in this document, and the PAC-Designer system help.

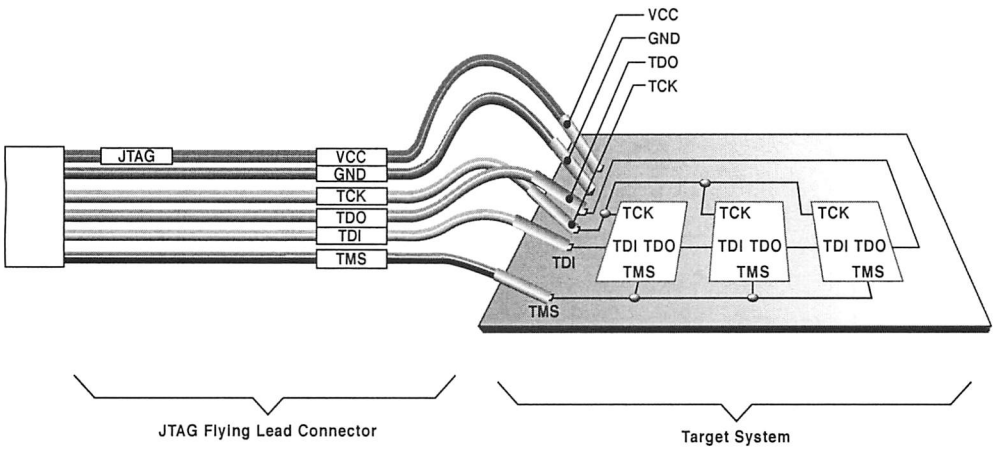



**Figure 2-6 Parallel Download Cable Connection to JTAG Boundary-scan TAP**

Appendix B contains schematic diagrams of the Parallel Download Cable.

### Configuring the Parallel Download Cable

On PCs you can connect the parallel cable to your system's parallel printer port. The JTAG Programmer software will automatically identify the cable when correctly connected to your PC. If you choose to, you may also select this connection manually. To set up a parallel port manually:

**`Output → Cable Setup`**

Select the **`Parallel`** box and match to the port you are using, then click on **`OK`**.

## Flying Lead Connectors

The flying lead connector has a 9-pin (6 signals, 3 keys) header connector that fits onto the cable's JTAG header. The pin order is listed in Table 2-3. These header connectors are keyed to assure

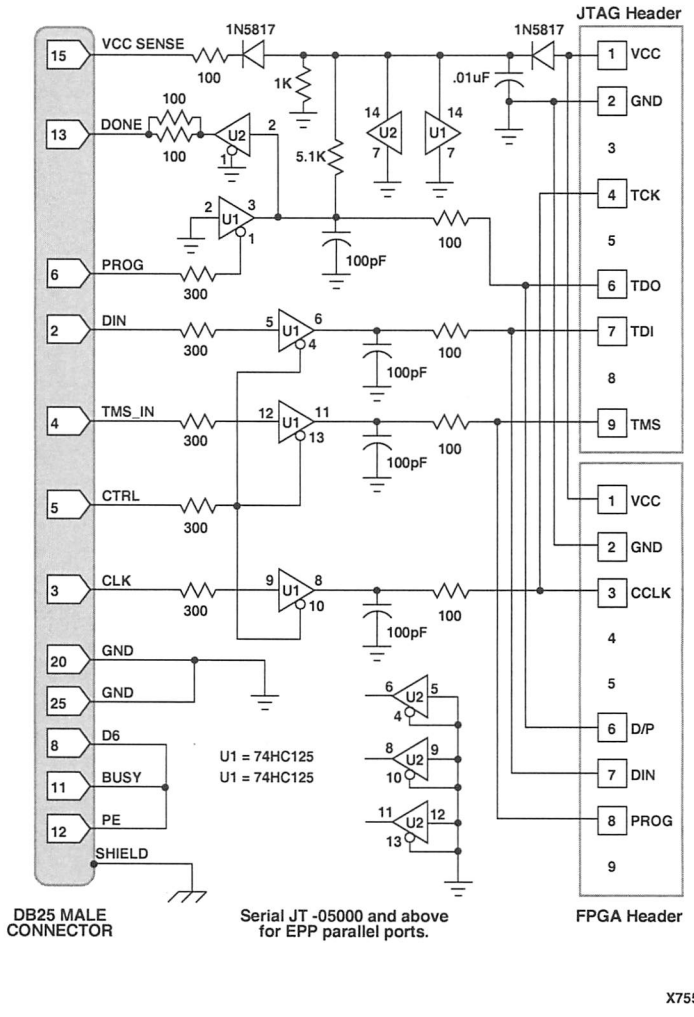



**Figure B-1 Parallel Download Cable**