Cascading ISR Devices

Introduction

This application note provides a detailed explanation of how to chain multiple programmable and non-programmable JTAG devices. Specifically covered herein will be the FLASH370i™ family of In System Reprogrammable (ISR™) CPLDs cascaded with themselves and with other programmable and non-programmable devices. Please refer to the application note titled "An Introduction to In System Reprogramming with FLASH370i™" for a preliminary understanding of these devices. This topic is broken into two categories: cascading FLASH370i with other FLASH370i devices, and cascading FLASH370i with other JTAG devices. The words manufacturer and vendor will be used interchangeably.

There are two methods which can be used to cascade devices with JTAG interfaces: parallel and serial. The parallel method has two variations: paralleled serial connection, and multiple independent paths. The following three figures illustrate the basics of each of these methods in block diagram form. *Figure 1a* shows a cascade of n devices in a serial connection. *Figure 1b* shows a cascade of n devices in a paralleled serial connection and *Figure 1c* shows a cascade of n devices in multiple independent paths. These are detailed where appropriate later in this application note.

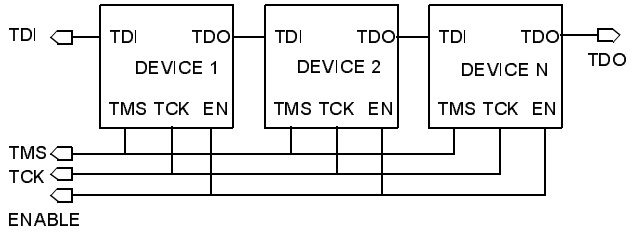

Figure 1a. Cascading N "JTAG Devices Using a Serial Connection"

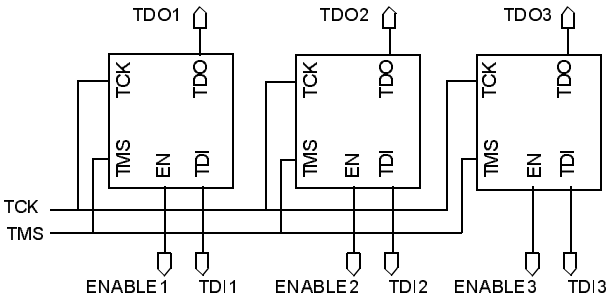

Figure 1c. Block Diagram: Cascading JTAG Devices Using Multiple Independent Paths

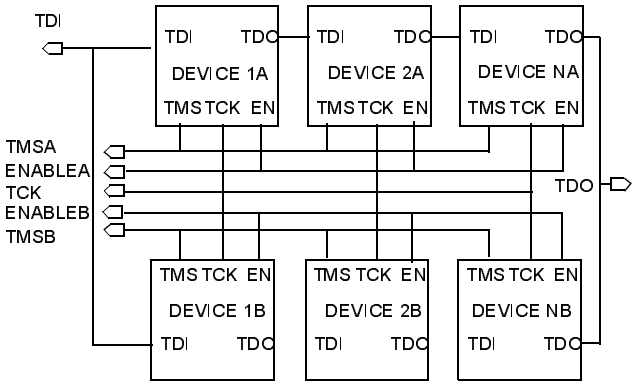

Figure 1b. Cascading N JTAG Devices Using Paralleled Serial Chains

The enable pin is typically present only on programmable devices. This will be addressed in detail in the following section.

Cascading FLASH370i Devices with Other FLASH370i Devices

Please refer to the application note "Designing with FLASH370i for PC Cable Programming" for details on single function and dual function pins.

Cascading in a serial chain is the most common method used when all devices in the chain are from the same manufacturer. This is the first case considered for this section. Cascading in a parallel connection can be accomplished in one of two ways: Paralleled Serial Chains or Multiple Independent Paths. Paralleled Serial Chaining is done by connecting SCLK(TCK) so that it is common to all devices and to have separate SMODE(TMS) signals connected to each device or group of devices. SDO(TDO) and SDI(TDI) are connected in serial fashion for each leg of the parallel chain so that SDI has one driver and SDO terminates in one single pin as shown in *Figure 1b*. Multiple Independent Paths are done by connecting both SCLK and SMODE so that each is common to all devices and just the SDI and SDO pins unique for each device or group of devices. SDI has a unique driver for each device or serial connection of devices. SDO likewise returns to a unique signal pin for each device or serial connection of devices. This method is illustrated in *Figure 1c*.

Cascading in a Serial Chain

Figure 2 shows n FLASH370i devices chained together in a serial fashion. This is a simple cascading example where the interface pins are either single function pins or dual function pins used only as programming pins. In this configuration, the SDO of one device is fed to the SDI of the next device. The SDI pin of the first device in the chain gets its signal from the interface used for programming. The SDO pin of the last device returns to the interface to complete the chain. The SCLK,

SMODE and ISRVPP pins are connected in parallel to all the devices. A large number of FLASH370i devices can be chained together in this manner. An example is if the chain consists of programmable devices where there might be a large current draw from each device during programming. In this case, the PC cable programming software included in the InSRkit package from Cypress lets the user choose how many devices are programmed at any given time. By limiting this, the load on the drivers in the cable is not exceeded. Since there is a limit on the number of devices being programmed at any given time, there can be a large number of these devices in the chain. To make this efficient, the PC cable programming software defaults to programming three devices at a time. Details on programming are discussed in later sections.

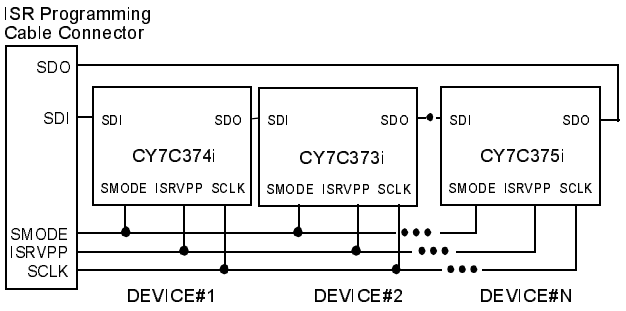

Figure 2. Block Diagram: Serial Connection of 'N' FLASH370i devices

Figure 1c shows n devices chained together in the “multiple independent paths” type of parallel connection. Each device in the figure could be a FLASH370i device. In this figure, the interface pins are either single function pins or dual function pins and are used only as programming pins. A large number of FLASH370i devices can be cascaded together in this configuration.

Cascading FLASH370i Devices with Other JTAG Devices

In the previous section, the various implementations of cascading several FLASH370i devices were discussed. This section will detail the implementation of FLASH370i devices with other JTAG compliant devices. When implementing a JTAG chain with multiple manufacturer's, the complexity of the chain increases and care should be taken. When cascading other JTAG compliant devices with Cypress's FLASH370i devices, it is recommended to use separate chains for each manufacturer. Implementing separate chains is easier because of the difference in connector sizes and pin configurations required by each manufacturer's programming scheme. The use of separate chains not only simplifies the hardware implementation but also the software tasks of communicating with each device. IEEE 1149.1 does not specify instruction register widths or standard file formats for software drivers. The remainder of this section may be skipped if the choice is to use separate chains. The remainder of this section will concern itself with the consideration of chaining multiple manufacturer's JTAG compliant devices in a single serial chain. Again, this is not the recommended method of utilizing the JTAG ports due to the difficulties that will become quite apparent.

Putting multiple manufacturers' JTAG-compliant devices into the same chain can be a very complex task. The first thing is to assure that all devices within the chain are JTAG compliant and can be put in bypass or 'transparent mode'. When in bypass mode there is a single register placed between the SDI and SDO pins per the IEEE 1149.1 specification. The single register delay is accounted for in the JTAG programming software to assure that the correct bit pattern is shifted into each device within the chain. If a device in the chain cannot be put into bypass mode then it should be put in a transparent mode instead. Transparent mode connects the SDI pin to the SDO within the CPLD, which adds a small combinatorial delay (insignificant, at the programming clock rate) but does not add any clocked delay. Devices in this configuration are added within the chain and can be ignored during the programming process.

Most devices that have single function pins can be placed directly into bypass mode by applying +5 volts to their V_{PP} pin. Bypass mode has been specified within the IEEE 1149.1 specification and is supported by all compliant devices. In the case of FLASH370i devices with dual function pins, the Vpp pin must have +12 volts applied in order to switch the dual function pins from 'normal use' to 'JTAG use'. No additional work need be performed if this can be achieved easily. Otherwise, a bypass register or transparent link must be formed within the part. Following is an example of VHDL code used to implement transparent mode in a CY7C371i device. The CY7C371i is a dual function pin device.

-- Transparent mode code for CY7C371i, change pin numbers for other device/packages

```
Library IEEE;

Use IEEE.Std_Logic_1164.all;

ENTITY trnsprnt IS

Port(

   sdi: IN   STD_LOGIC;

   sdo: OUT  STD_LOGIC

   );

ATTRIBUTE Pin_Numbers OF trnsprnt : EN-
TITY IS

   "sdo:27, "&-- JTAG serial data output

   "sdi:39";-- JTAG serial data input

ATTRIBUTE pin_avoid OF trnsprnt: ENTITY
IS "7"; --

ATTRIBUTE pin_avoid OF trnsprnt: ENTITY
IS "19";--

END trnsprnt;

ARCHITECTURE arch_trnsprnt OF trnsprnt
IS

BEGIN
```

```
sdo <= sdi;

END arch_trnsprnt;
```

Additional options exist that allow such devices to be used within a single daisy chain. For example, a signal can be provided that would pull V_{PP} high, enabling the JTAG pins, or a TAP controller can be implemented within the device. It is beyond the scope of this application note to detail the full design of a JTAG TAP controller. Additionally, this TAP controller would utilize a large number of macrocells within the device, allowing less room for the logic in the CPLD. It is recommended to use devices that have single function pins to simplify the design if a single JTAG chain is required.

IEEE 1149.1 Instructions

A brief interlude is taken here to introduce some of the terminology that will be used later. The IEEE 1149.1 specification defines three required instructions and two optional instructions (section 7.2). Two of the three required instructions are for boundary scan. The Cypress FLASH370i devices do not contain boundary scan registers, so the instructions for this are not supported. The BYPASS instruction and the optional IDCODE and USERCODE instructions are supported by all FLASH370i devices.

The bypass instruction is covered in minor detail next because of its inclusion in the following sections. Details for all of the diagnostic instructions supported are given in the Diagnostics section at the end.

Bypass Instruction

When executed, the bypass instruction inserts a single shift-register stage between the SDI and SDO pins of a device. This is so a minimum-length serial path can rapidly shift data through a device when no operations are required of that device. Upon issuing this instruction, the device is said to be in bypass mode.

Using the Cypress PC cable programming software, a Cypress device is put into bypass by including the following line in the configuration file.

CY7C###i n;

Where ### is the appropriate Cypress device number and 'n' is the command for putting a Cypress device in bypass mode.

The Cypress instruction register length is 4. So to put a Cypress device into bypass mode would require sending an instruction of '1111'.

It is important to note that bypass mode is supported in the dual function pin devices only when 12V is applied to the V_{PP} pin. It is supported in either mode (V_{PP} with 12V or with 0V) on the single function pin devices, the CY7C373i and CY7C374i in TQFP packages.

Optional Instructions

The optional IDCODE and USERCODE instructions are supported by the FLASH370i devices. Two devices in the family, the CY7C373i and CY7C374i in TQFP packages, support these instructions either 5V or 12V mode. These are the two single-function pin devices in the family. The dual function pin devices in the family only support these instructions in 12V mode.

Diagnostics with Multiple Single-Function Pin Devices (374/5 TQFP)

The single function pin devices can perform the supported JTAG instructions in either 5V or 12V mode.

The Cypress ISR software can be used to communicate with the Cypress devices in the chain provided all other devices are bypassed. This is easily accomplished as explained in the previous section.

The user can perform bypass, get user code and get silicon ID instructions on the various Cypress devices in the chain.

Diagnostics with Multiple Dual Function Pin Devices

The dual function pin devices can only perform the JTAG instructions in 12V or programming mode. Cypress devices require 12V to program. When 12V is applied to the Vpp pin, the device goes into programming mode. When in programming mode, the dual function pins have JTAG functionality. When 12V is removed from the Vpp pin, the device returns from programming mode to operational mode. In this mode, the dual function pin devices no longer have JTAG capabilities and return to regular I//O pin functions.

Bypass mode can be supported two ways for the dual function pin devices as explained above. The dual function pin devices have to be either in 12V mode to be instructed to go into bypass mode and then can only remain in bypass mode with 12V on V_{PP} or must be coded to have a bypass register implemented in hardware for 5V operation.

Programming

Refer to the one of the following application notes for details on programming the FLASH370i devices

Designing with FLASH370i for PC Cable Programming

FLASH370i 5V to 12V DC-DC Converter Solutions

Programming Sequences with MixedVendor's Devices

When using a single vendor per serial chain, the use of single and dual function devices is taken into account in the specific vendor's programming software. If multiple vendor's are used, however, the task of which devices to program in which order is left to the designer. A brief overview of the possibilities that will be encountered are given below.

When programming a chain with Cypress single function pin devices with:

1. Other vendor's devices that all have single function JTAG compliant pins - In this case, the JTAG pins/functions on all devices are always available. Therefore the Cypress ISR programming software only needs to know the relative location of the other vendor's devices in the chain and their instruction register lengths.
2. Other vendor's devices that have dual function JTAG compliant pins - In this case, the JTAG pins/functions may not always available.
   a. If the other vendor's dual function devices pins can be forced to perform their JTAG function, then this should be done. Once in JTAG mode, the Cypress ISR programming software only needs to know the relative location of the other vendor's devices and their instruction register length. OR

b. If the other vendor's dual function pins cannot be forced to perform their JTAG function, then the other vendor's device must be programmed first and put in the 'transparent mode'. The Cypress part is then programmed as usual, and the 'transparent' part is not listed in the chain.

When programming a chain that contains Cypress dual function pin devices with:

1. Other vendor's devices that all have single function JTAG compliant pins - In this case, the JTAG pins/functions on all non-Cypress devices are always available. Therefore the Cypress ISR programming software only needs to know the relative location of the other vendor's devices in the chain and their instruction register lengths.
2. Other vendor's devices that have dual function JTAG compliant pins - In this case, the JTAG pins/functions may not always available.
   a. If all vendor's parts can be forced to be in the JTAG mode, then force them all and proceed with programming. Once in JTAG mode, the Cypress ISR programming software, as well as the other vendor's programming software only needs to know the relative location of the other vendor's devices in the chain and their instruction register lengths. OR
   b. If the other vendor's dual function devices pins can be forced to be in JTAG mode, then this should be done. Once in JTAG mode, the Cypress ISR programming software only needs to know the relative location of the other vendor's devices in the chain and their instruction register lengths. Each dual function device should also be left in transparent mode, to facilitate programming the non-Cypress parts afterwards. OR
   c. If the Cypress dual function parts can be forced to be in JTAG mode, then the other vendor's parts may be programmed and left in transparent mode. The Cypress ISR programming software could then program the Cypress parts, not listing the transparent parts in the chain. OR
   d. If neither vendor's dual function pins cannot be forced to be in JTAG mode, then the chain is broken and no devices within the chain may be programmed

Using Cypress ISR Software to Program FLASH370i in a Serial Chain with Multiple Manufacturers

This section assumes that the other manufacturer's parts have single function pins and are JTAG compliant. If this is not the case then refer to the previous section to determine the steps required to program the devices. *Figure 3* is used in this section to illustrate a serial chain with multiple manufacturers. The first thing to consider is how to put select devices into bypass mode so that one or more other devices can be exercised. The first case we will consider is when using the Cypress ISR Software to program any Cypress devices in the chain. This method requires that all non-Cypress devices in the chain contain the standard JTAG TAP controller and support a bypass instruction which uses the specified code of all 1s.

Bypass Mode with Cypress ISR Software

Using Cypress PC cable Software, it is easy to put other manufacturer's devices into bypass mode. The software reads a configuration file and performs the instructions provided within. An example of a configuration file is provided below for the chain shown in *Figure 3*.

```
CY7C371i p c:\control.jed;

NONCYPRESS 5;

NONCYPRESS 5;

CY7C374i r c:\chirumbo.jed

NONCYPRESS 4;

CY7C372i p c:\jwiz.jed;
```

This configuration file tells the Cypress ISR software where the non-Cypress devices are and to put them into bypass mode. The '5' after the word 'NONCYPRESS' is a number indicating the length of the instruction register for the respective devices in the chain. A detailed description of the configuration file can be found in the application note “Designing with FLASH370i for PC Cable Programming”. The Cypress devices in the chain will be treated according to the commands given. Information on the commands available and what they do can be found in the ISR Software User's Guide which comes with the InSRkit package.

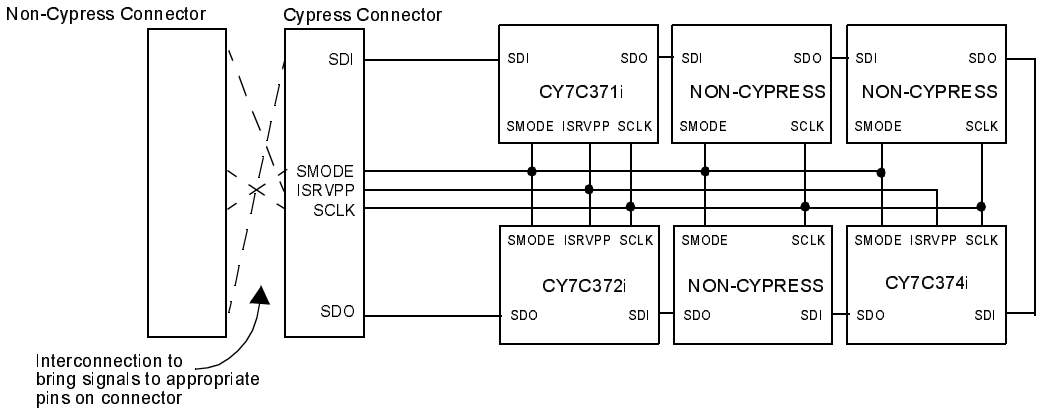

Figure 3. Block Diagram: Serial Connection with Multiple Manufacturers in Chain

The first device in the chain is a Cypress CY7C371i and will get programmed with the file control.jed located in the c:\ directory. The next two devices in the chain are non-Cypress devices and will be placed in bypass mode. The fourth device in the chain is a Cypress CY7C374i. The ISR software will read the JEDEC map out of the device and create a file named c:\chirumbo.jed. The fifth device in the chain is again a non-Cypress device and gets placed in bypass mode. The last device in the chain is a Cypress CY7C372i and will be programmed with the file c:\jwiz.jed.

Programming Other Vendor's Devices

Note that the Cypress ISR software can only be used to program Cypress devices. Devices from other vendors must be programmed with software from that vendor. When programming their devices with Cypress devices that are in JTAG mode, their software must be told that the Cypress FLASH370i instruction register is 4 bits long. Also note that Cypress parts do comply to the all 1's specification for the bypass instruction as set forth in the IEEE1149.1 specification.

Conclusion

The Cypress FLASH370i family of complex programmable devices can be chained together in a number of ways and with other manufacturer's devices to simplify on-board programming and diagnostic functions. The Cypress ISR software has the ability to support other manufacturer's in a chain by accommodating various instruction register lengths. Despite this, the recommended utilization is to use a serial chain with each manufacturer in its own chain. This recommendation is based on the complications incurred when dealing with variations between various manufacturer's programming and diagnostic software and instruction register lengths. The existence of dual function pin devices which may only support bypass mode in one configuration additionally complicate matters.

There are a number of issues to consider if a implementing a single chain with multiple manufacturer's devices is desired:

1. Are all the devices JTAG compliant? Do they support the IEEE 1149.1 specification? If not, what portion is supported?
2. Does the programming / diagnostic software for each manufacturer allow specifying the instruction register length? This is necessary so that any device in the chain can be bypassed. IEEE 1149.1 does not specify a fixed length for this register.
3. What are the different connector pinouts for each manufacturer? This needs to be identified so that all the ramifications of implementation can be resolved.

Once these issues have been understood and addressed an accurate evaluation can be made.