



VC709 MultiBoot Design

April 2013

XTP236

Revision History

Date	Version	Description
04/03/13	1.0	Initial version.

© Copyright 2013 Xilinx, Inc. All Rights Reserved.

XILINX, the Xilinx logo, the Brand Window and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.

NOTICE OF DISCLAIMER: The information disclosed to you hereunder (the “Information”) is provided “AS-IS” with no warranty of any kind, express or implied. Xilinx does not assume any liability arising from your use of the Information. You are responsible for obtaining any rights you may require for your use of this Information. Xilinx reserves the right to make changes, at any time, to the Information without notice and at its sole discretion. Xilinx assumes no obligation to correct any errors contained in the Information or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE INFORMATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

Overview

- **Virtex-7 MultiBoot Capability**
- **Overview of 7 Series Fallback MultiBoot**
- **Xilinx VC709 Board**
- **Software Requirements**
- **VC709 Setup**
- **Compiling the MultiBoot Design**
- **Run MultiBoot Design**
- **Virtex-7 MultiBoot Details**
- **References**

Virtex-7 MultiBoot Capability

► What is MultiBoot?

- The MultiBoot feature allows the FPGA application to load two or more FPGA bitstreams under the control of the FPGA application. The FPGA application triggers a MultiBoot operation, causing the FPGA to reconfigure from a different configuration bitstream. After a MultiBoot operation is triggered, the FPGA restarts its configuration process as usual.
- More details can be found in [UG470](#)

► MultiBoot Capability

- FPGA application controlled configuration
- Bitstream selection of multiple applications

► Safe Update

- Golden bitstream
- Upgradeable bitstream
- Failure recovery
 - Possible Triggers (CRC error, IDCODE error, WDT timeout)

Virtex-7 MultiBoot Capability

► Overview of 7 Series Fallback MultiBoot

- The 7 series FPGAs MultiBoot and fallback features support updating systems in the field. Bitstream images can be upgraded dynamically in the field. The FPGA MultiBoot feature enables switching between images on the fly. When an error is detected during the MultiBoot configuration process, the FPGA can trigger a fallback feature that ensures a known good design can be loaded into the device.
- The MultiBoot feature allows the FPGA application to load two or more FPGA bitstreams under the control of the FPGA application. The FPGA application triggers a MultiBoot operation, causing the FPGA to reconfigure from a different configuration bitstream. After a MultiBoot operation is triggered, the FPGA restarts its configuration process as usual.

► Reference Design Source and Application

- rdf0234.zip
- Available through <http://www.xilinx.com/vc709>

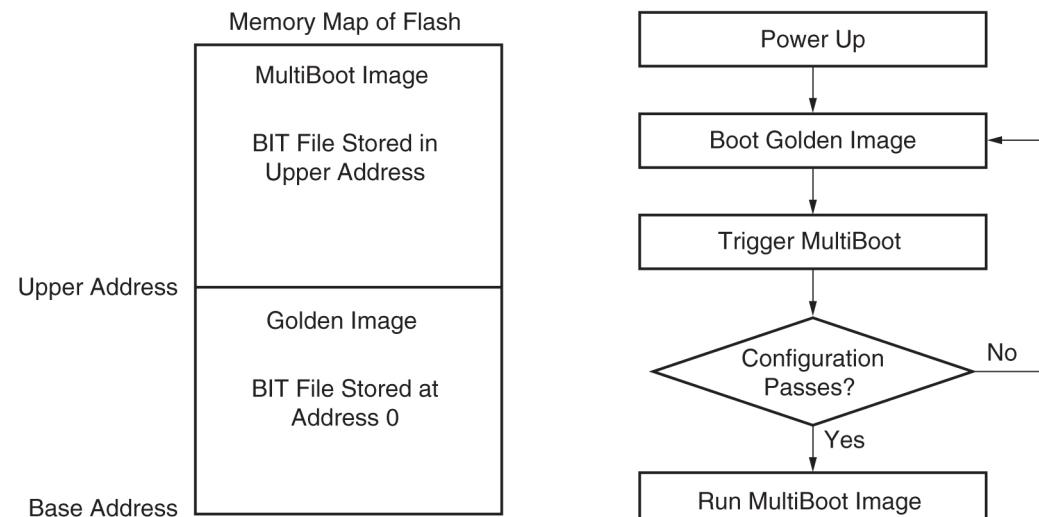
VC709 MultiBoot Design Description

► Description

- Design consists of three bitstreams, golden_iprog.bit, multiboot.bit, and corrupted.bit loaded in the PROM at 0x0, 0x400000, and 0x800000
- The golden_iprog.bit has software to allow a reboot to the second or third bitstreams with certain switch settings
- The multiboot.bit is compiled with this property set:
set_property BITSTREAM.CONFIG.CONFIGFALLBACK Enable [current_design]
- The corrupted.bit is a multiboot.bit with one byte changed to force a CRC error when loaded

► This diagram shows the general process

- Since the corrupted.bit doesn't pass configuration, the Golden Image is booted



Upgrade Example

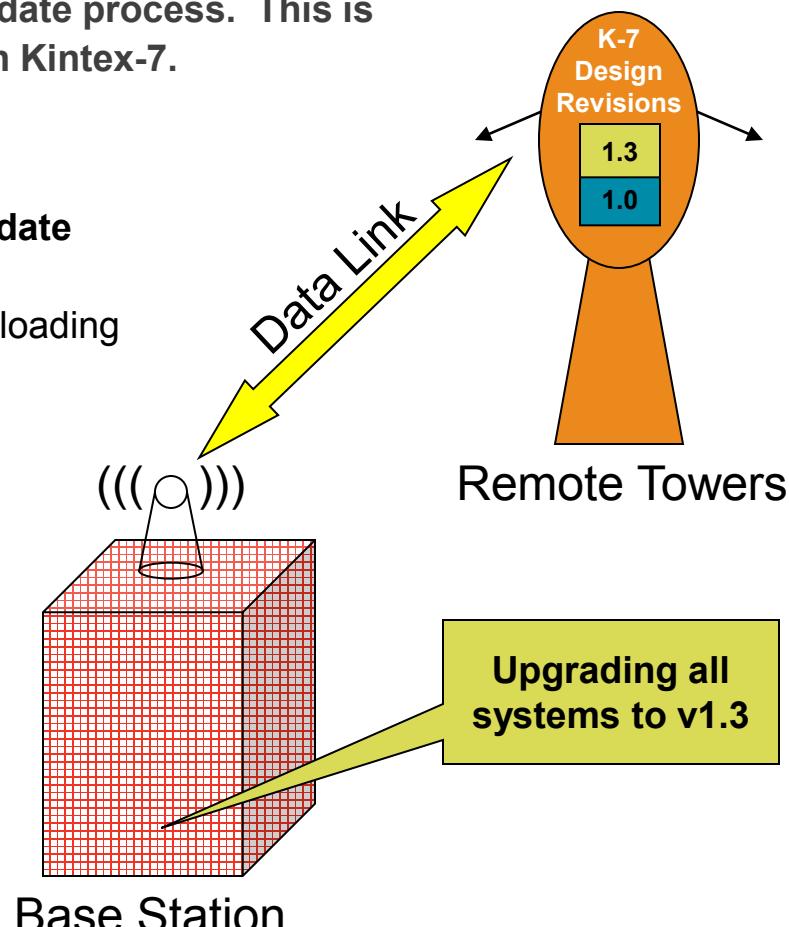
- **MultiBoot:** Process by which the FPGA selectively reprograms and reloads its bitstream from an attached external memory.
- **Safe update:** Field updating bitstream storage with a new bitstream in such a manner to prevent any failure due to a failure in the update process. This is accomplished with the enhanced MultiBoot available in Kintex-7.

Can a multi-boot system be implemented without safe update design considerations?

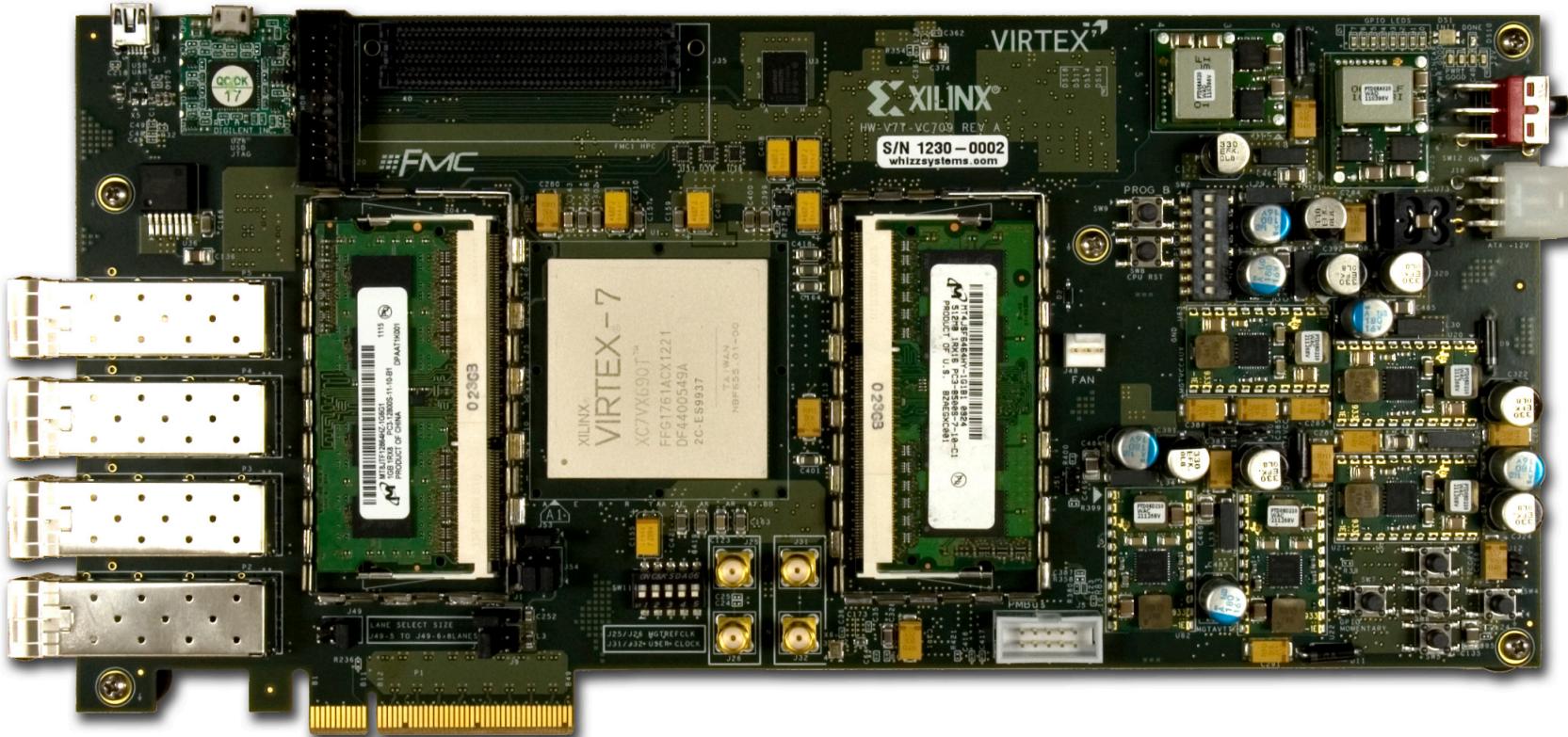
Yes – If there are no potential for disruptions during flash loading

How is a system upgraded?

1. New multi-boot image is created
2. System setup to receive the new image
3. User application erases section of Flash
4. The new image is delivered into the system's Flash
5. User application resets system

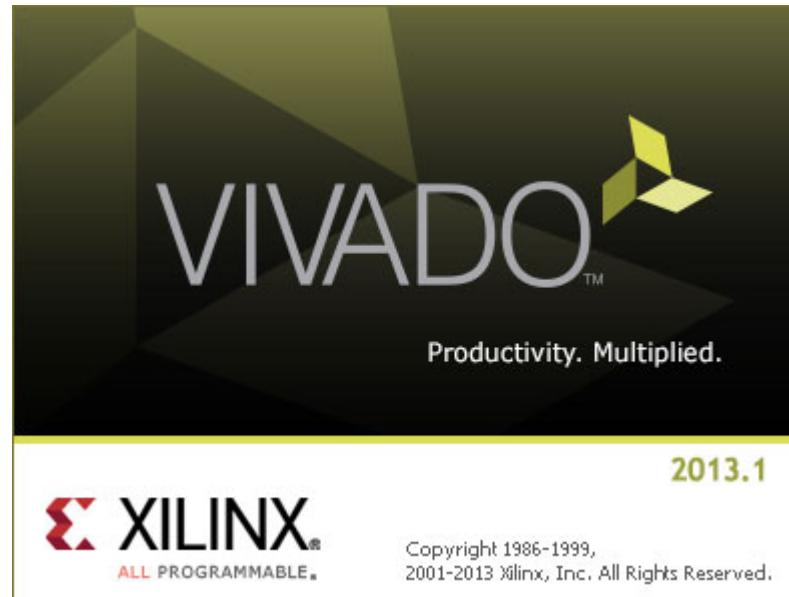


Xilinx VC709 Board



Vivado Software Requirements

- Xilinx Vivado Design Suite 2013.1, Design Edition + SDK
 - Combined installer



ISE Software Requirement

► Xilinx ISE 14.5 software

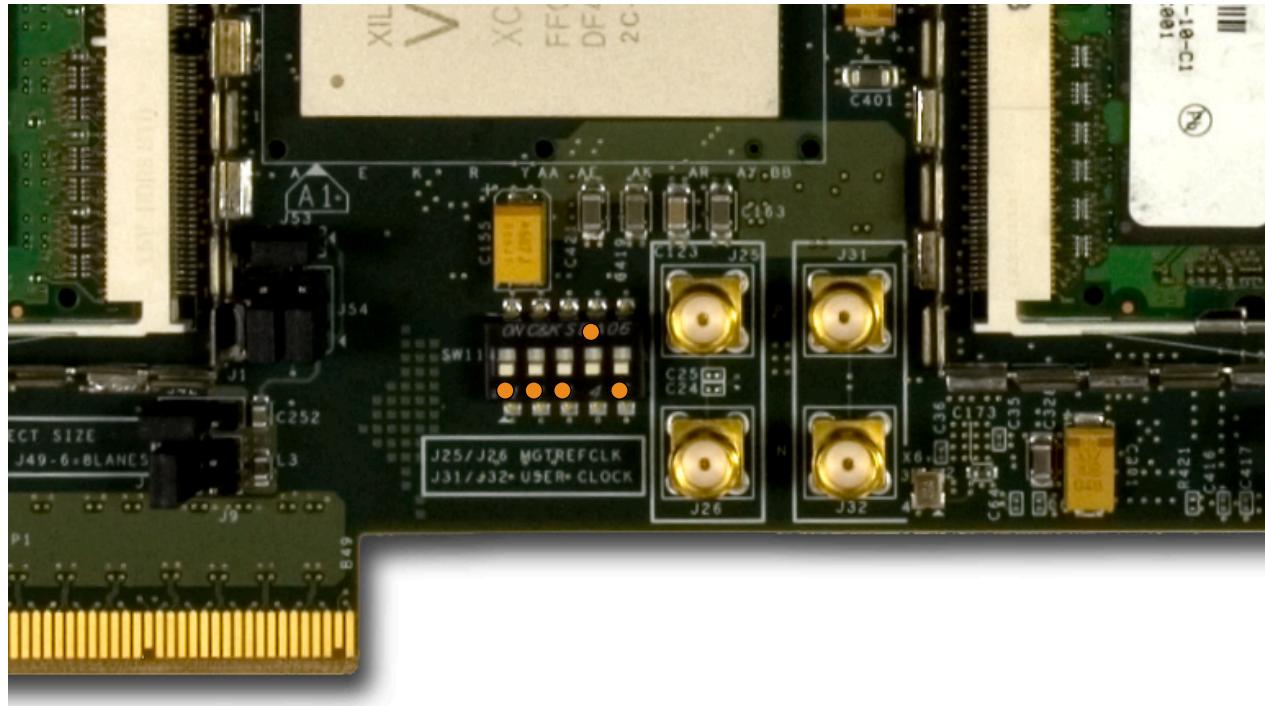
- Needed for iMPACT



Mode Switch Setup

► Set SW11 to 00010 (1 = on, Position 1 → Position 5, left to right)

- This enables Master BPI configuration from the Linear Flash
 - Flash A25, A24 = 00
 - FPGA mode pins M[2:0] = 010



Hardware Setup

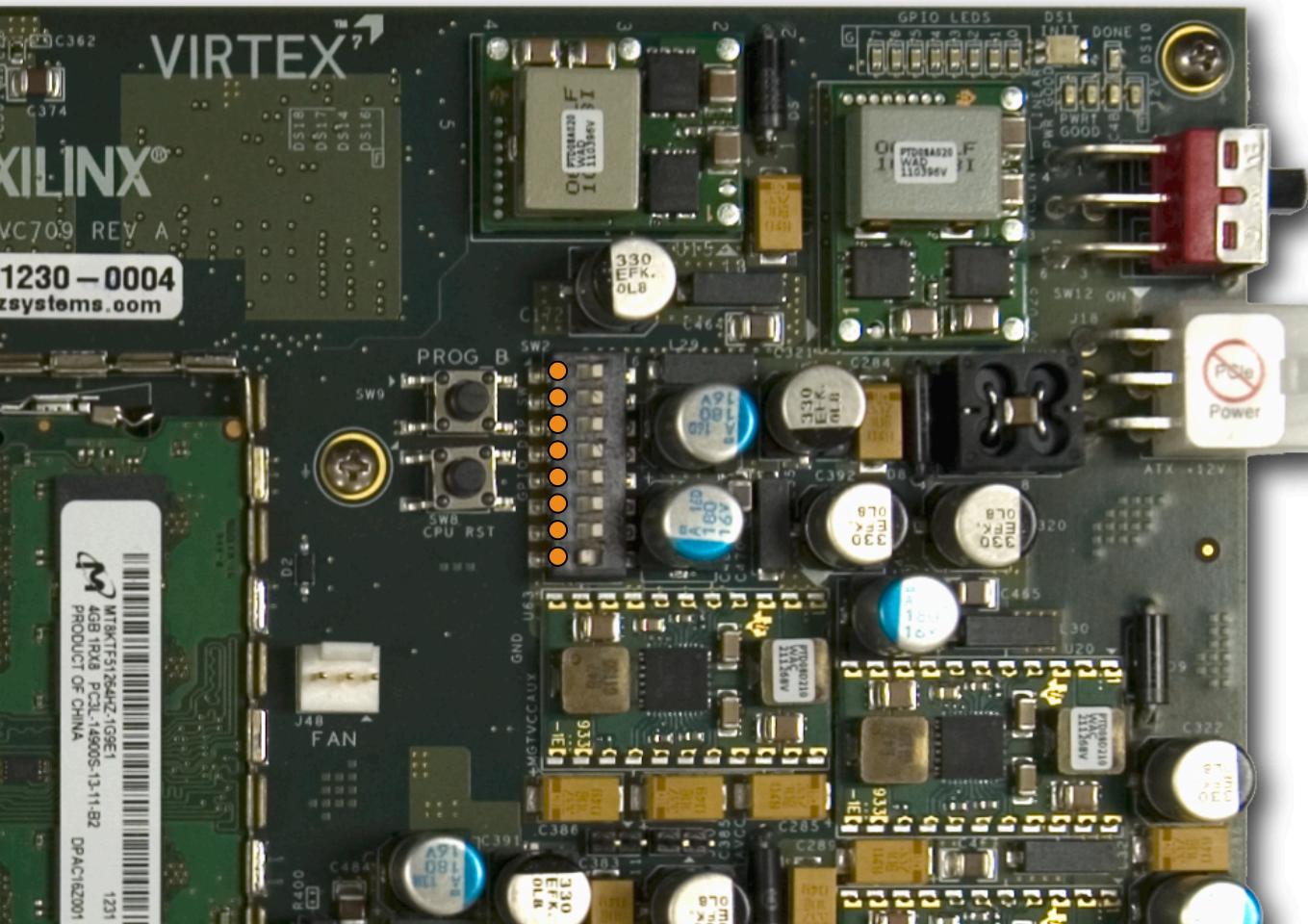
- ▶ Connect a USB Type-A to Micro-B cable to the USB JTAG (Digilent) connector on the VC709 board
 - Connect this cable to your PC



Hardware Setup

► Set SW2 to 00000000 (1 = on, Position 1 → Position 8)

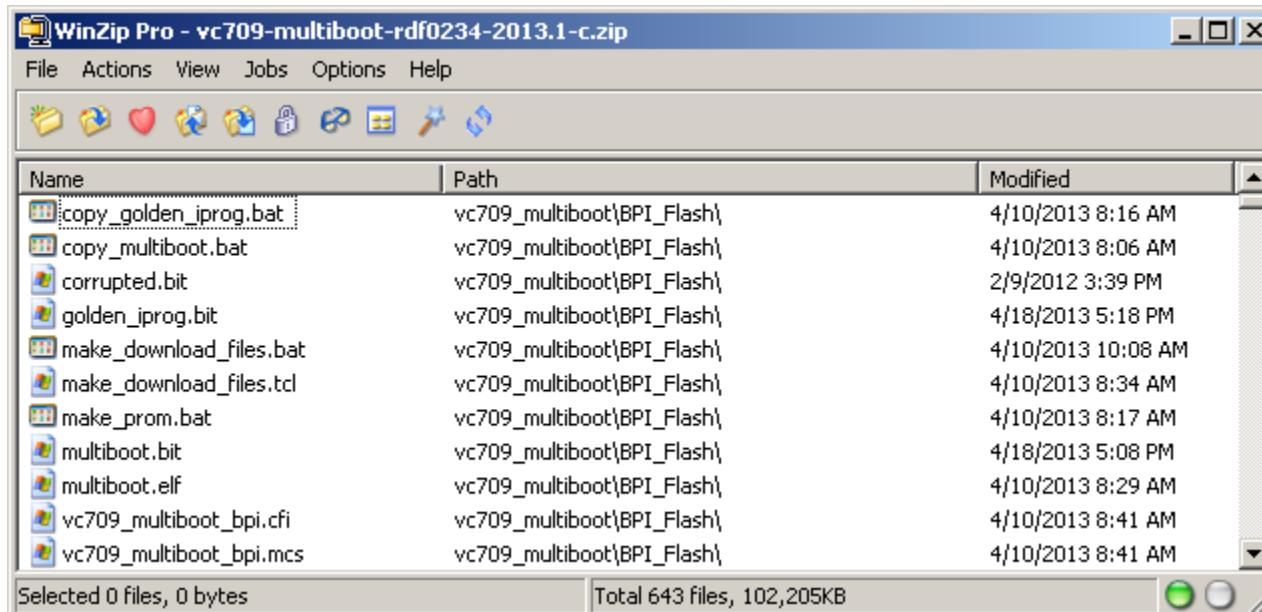
- This sets the MultiBoot functions to off



Program BPI MultiBoot Design

► Unzip the VC709 MultiBoot Design Files (2013.1 C)

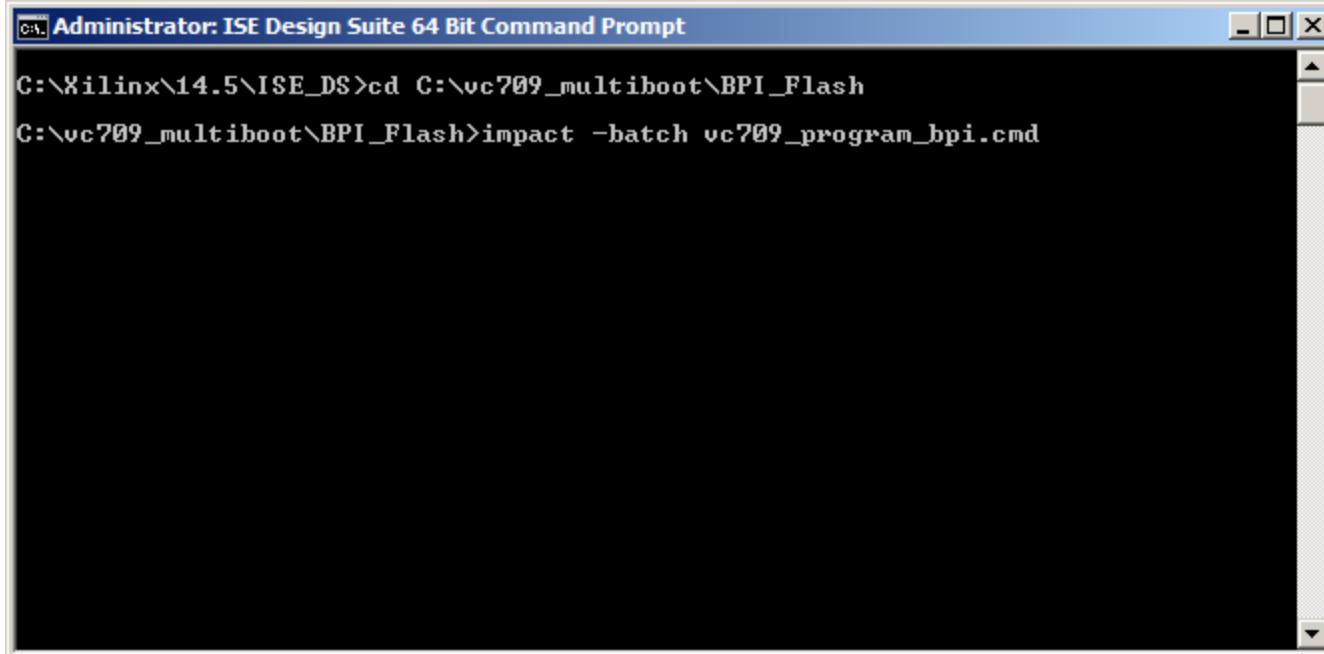
- Available through <http://www.xilinx.com/vc709>



Program BPI MultiBoot Design

- Open an ISE Design Suite Command Prompt and type:

```
cd C:\vc709_multiboot\BPI_Flash  
impact -batch vc709_program_bpi.cmd
```

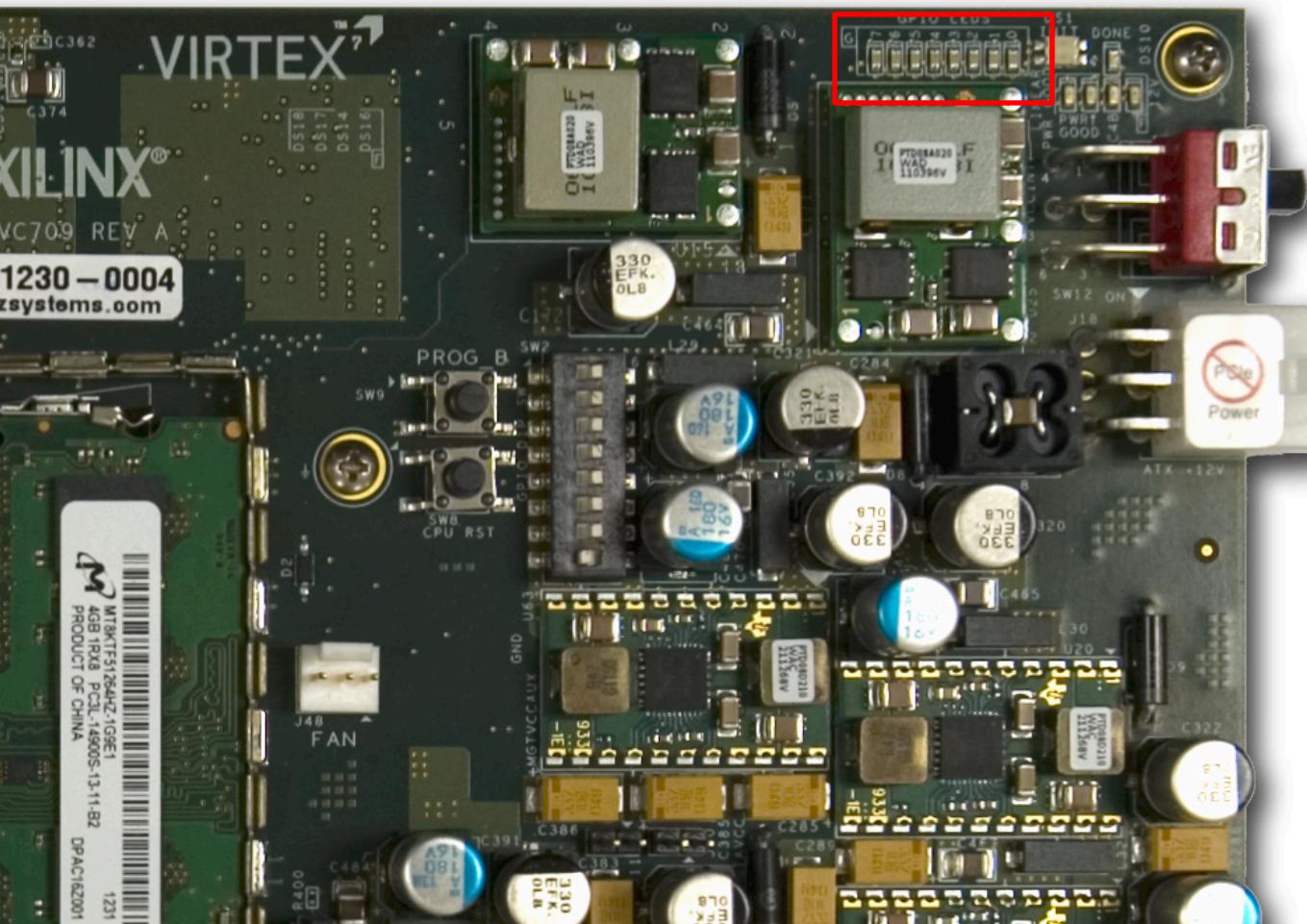


The image shows a Windows Command Prompt window titled "Administrator: ISE Design Suite 64 Bit Command Prompt". The window contains the following text:

```
C:\Xilinx\14.5\ISE_DS>cd C:\vc709_multiboot\BPI_Flash  
C:\vc709_multiboot\BPI_Flash>impact -batch vc709_program_bpi.cmd
```

Run MultiBoot Design

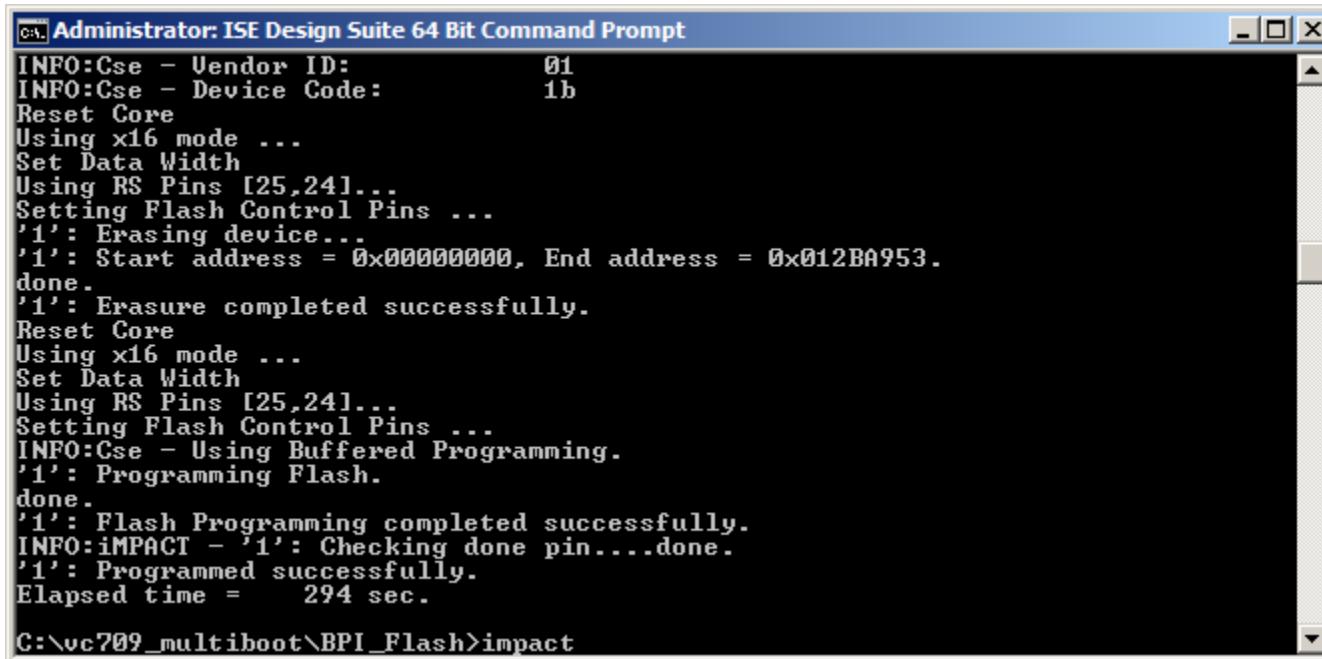
- The initial design, “golden_iprog.bit”, shows a cycling LED pattern on the GPIO LEDs



Run MultiBoot Design

► Run iMPACT:

impact



The screenshot shows a Windows command prompt window titled "Administrator: ISE Design Suite 64 Bit Command Prompt". The window contains the following text output from the iMPACT command:

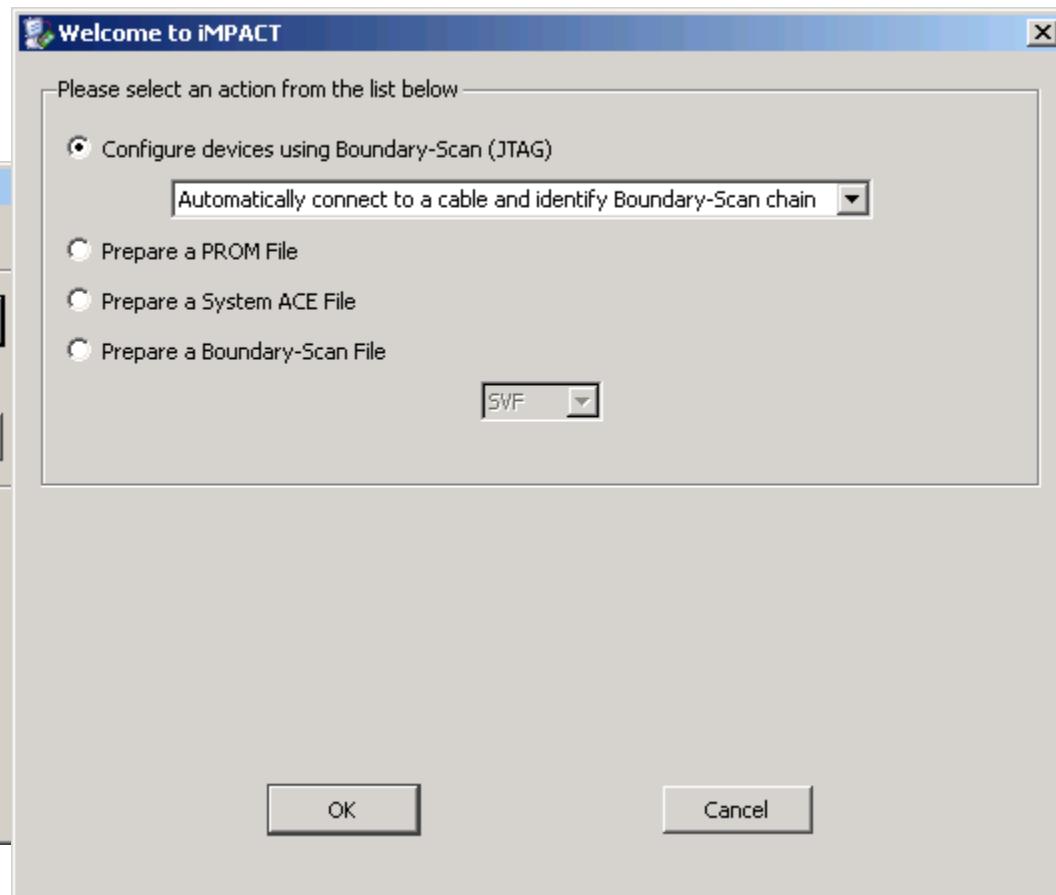
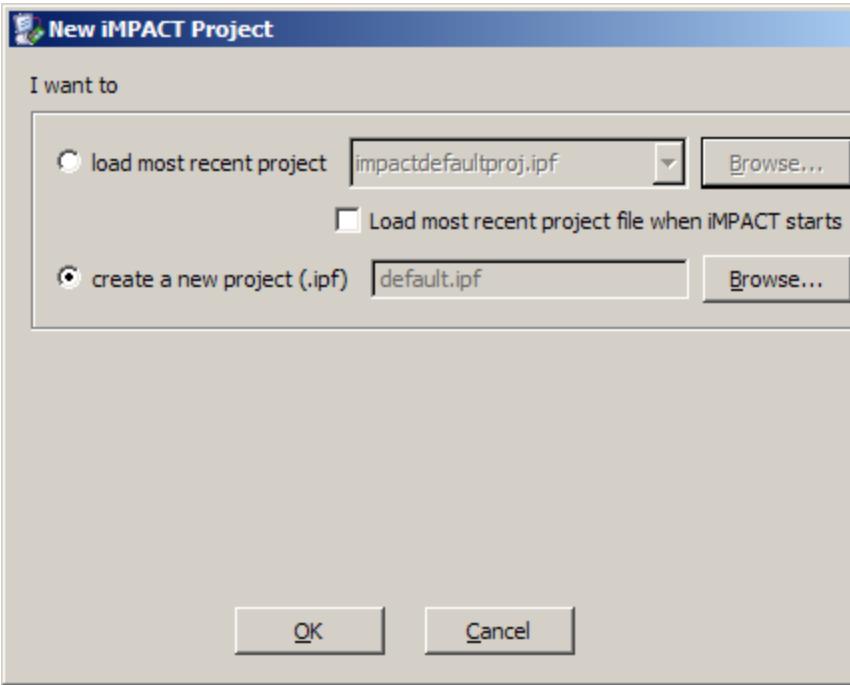
```
INFO:Cse - Vendor ID:      01
INFO:Cse - Device Code:    1b
Reset Core
Using x16 mode ...
Set Data Width
Using RS Pins [25,24]...
Setting Flash Control Pins ...
'1': Erasing device...
'1': Start address = 0x00000000, End address = 0x012BA953.
done.
'1': Erasure completed successfully.
Reset Core
Using x16 mode ...
Set Data Width
Using RS Pins [25,24]...
Setting Flash Control Pins ...
INFO:Cse - Using Buffered Programming.
'1': Programming Flash.
done.
'1': Flash Programming completed successfully.
INFO:iMPACT - '1': Checking done pin....done.
'1': Programmed successfully.
Elapsed time =    294 sec.

C:\vc709_multiboot\BPI_Flash>impact
```

Run MultiBoot Design

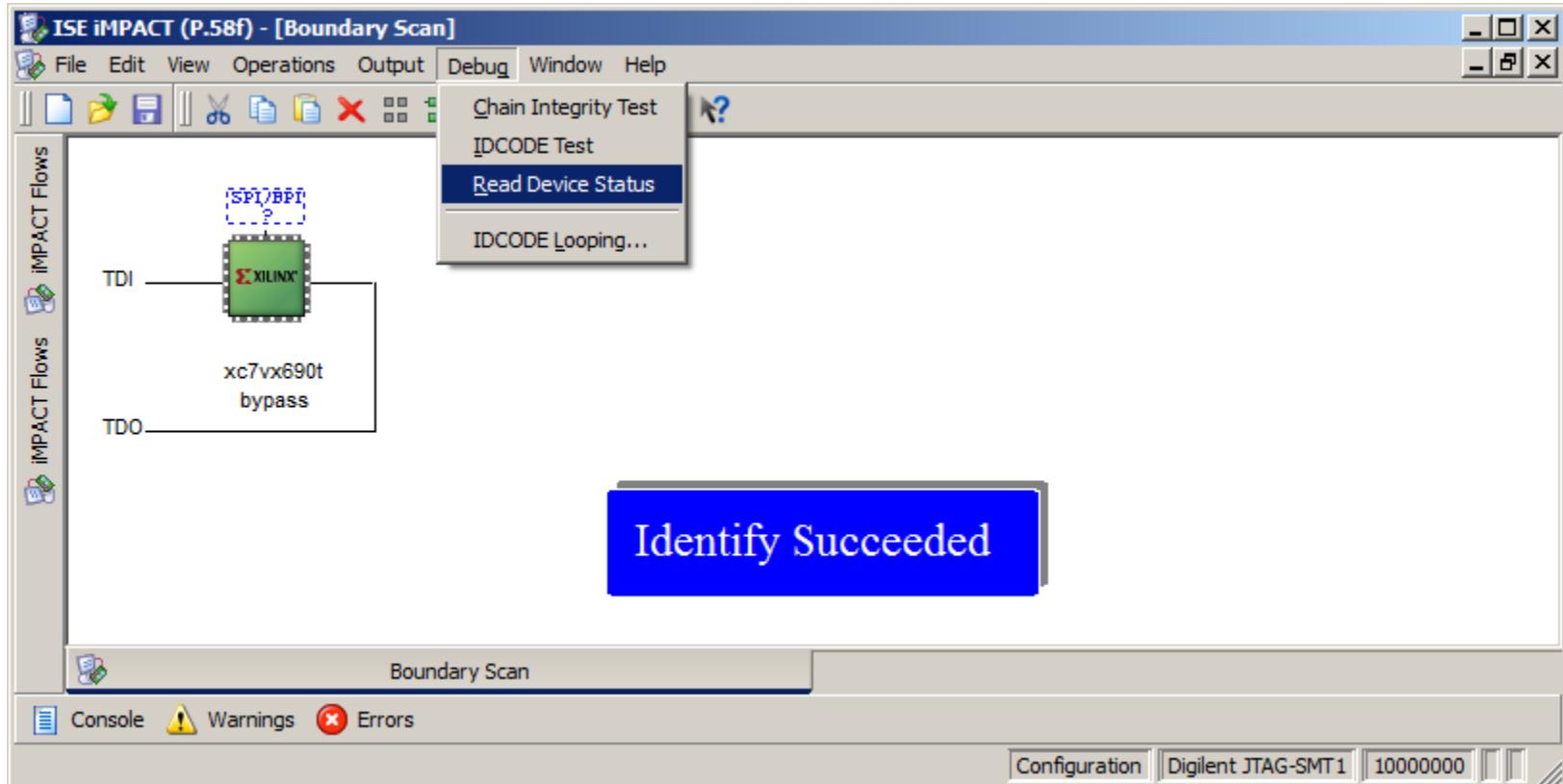
► Select

- Create a new project
- Configure devices using Boundary-Scan (JTAG)



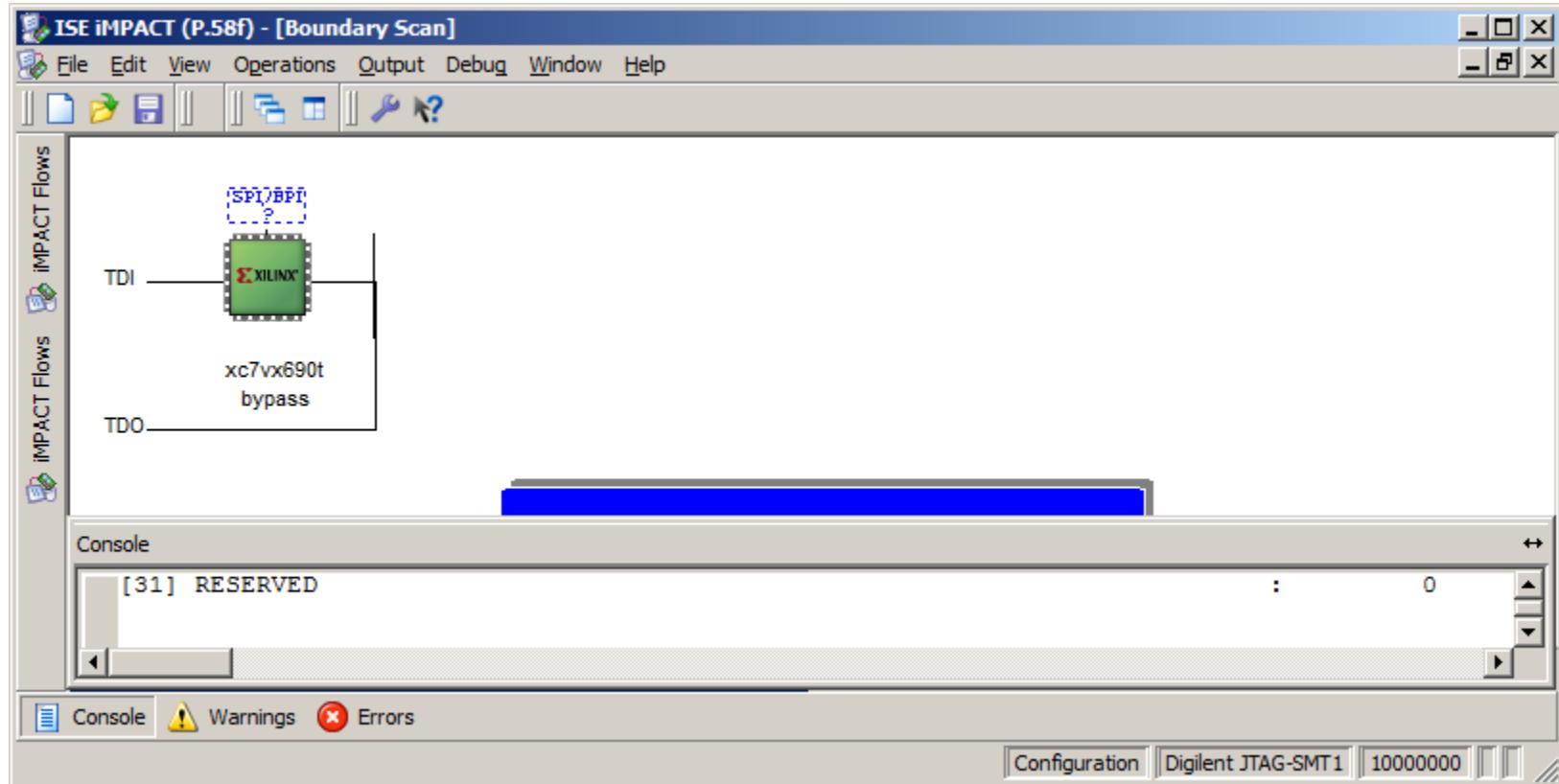
Run MultiBoot Design

- Select the FPGA
- Select the menu item Debug → Read Device Status



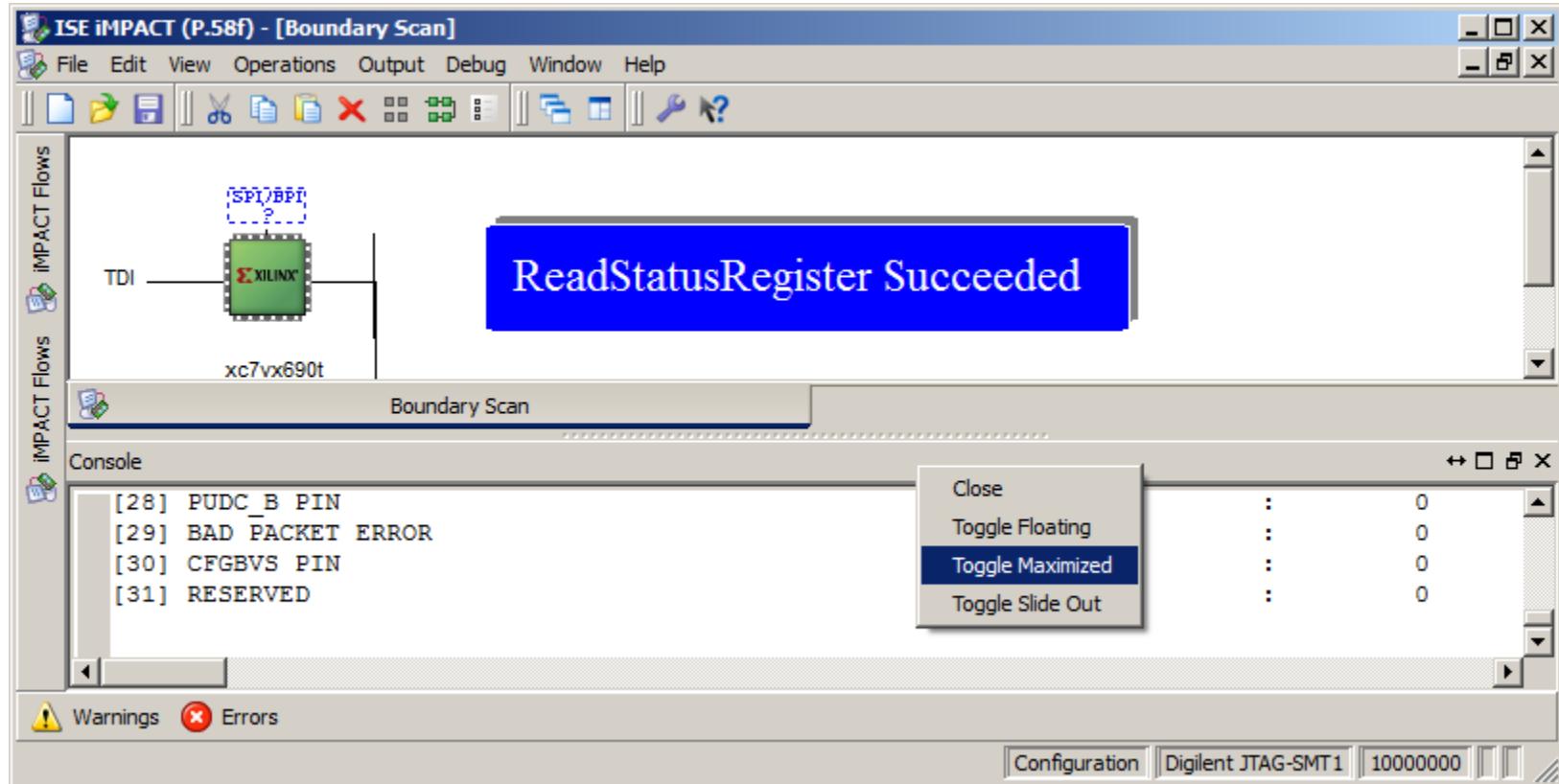
Run MultiBoot Design

- ▶ Click console tab and then right click on the console panel and select “Toggle Slide Out”



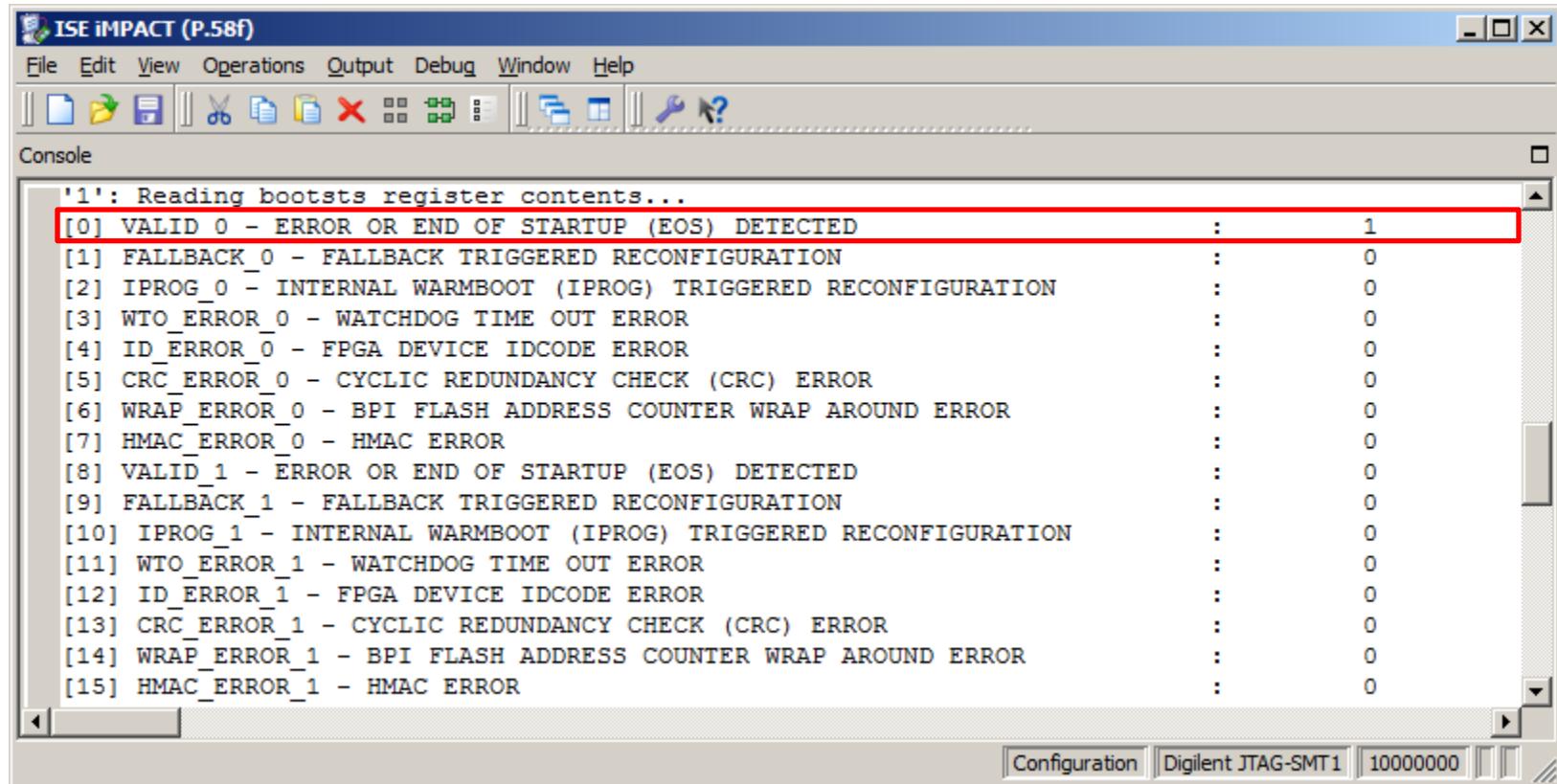
Run MultiBoot Design

- Right click on the Console panel and select Toggle Maximized



Run MultiBoot Design

- Scroll down to the “Reading bootsts register contents...” message
- Since this was a regular boot, only “VALID_0” is set



The screenshot shows the ISE iMPACT software interface with the title bar "ISE iMPACT (P.58f)". The menu bar includes File, Edit, View, Operations, Output, Debug, Window, and Help. The toolbar below the menu has icons for file operations like Open, Save, and Close, as well as configuration and debug tools. The main window is titled "Console". The output text in the console is as follows:

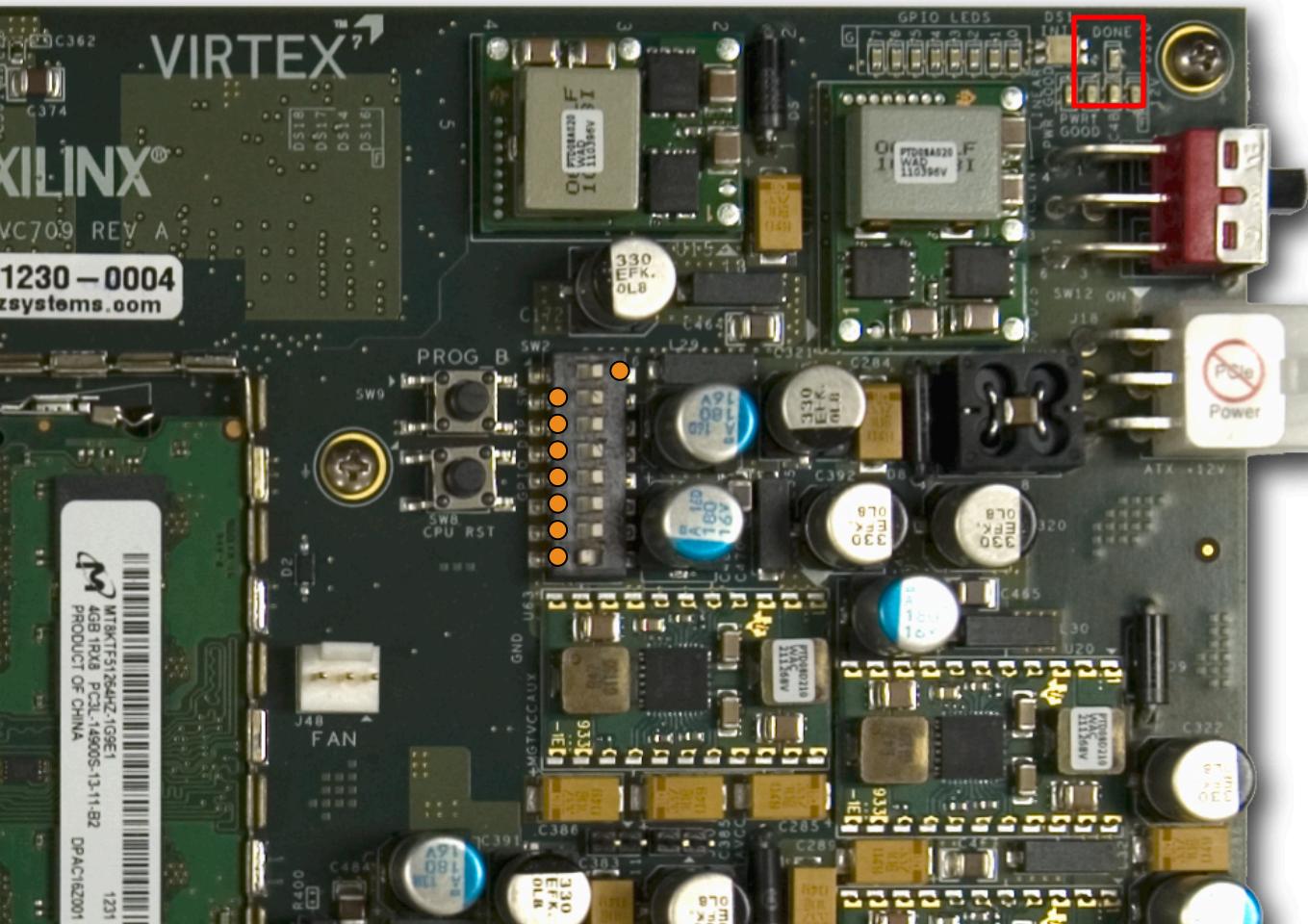
```
'1': Reading bootsts register contents...
[0] VALID_0 - ERROR OR END OF STARTUP (EOS) DETECTED : 1
[1] FALBACK_0 - FALBACK TRIGGERED RECONFIGURATION : 0
[2] IPROG_0 - INTERNAL WARMBOOT (IPROG) TRIGGERED RECONFIGURATION : 0
[3] WTO_ERROR_0 - WATCHDOG TIME OUT ERROR : 0
[4] ID_ERROR_0 - FPGA DEVICE IDCODE ERROR : 0
[5] CRC_ERROR_0 - CYCLIC REDUNDANCY CHECK (CRC) ERROR : 0
[6] WRAP_ERROR_0 - BPI FLASH ADDRESS COUNTER WRAP AROUND ERROR : 0
[7] HMAC_ERROR_0 - HMAC ERROR : 0
[8] VALID_1 - ERROR OR END OF STARTUP (EOS) DETECTED : 0
[9] FALBACK_1 - FALBACK TRIGGERED RECONFIGURATION : 0
[10] IPROG_1 - INTERNAL WARMBOOT (IPROG) TRIGGERED RECONFIGURATION : 0
[11] WTO_ERROR_1 - WATCHDOG TIME OUT ERROR : 0
[12] ID_ERROR_1 - FPGA DEVICE IDCODE ERROR : 0
[13] CRC_ERROR_1 - CYCLIC REDUNDANCY CHECK (CRC) ERROR : 0
[14] WRAP_ERROR_1 - BPI FLASH ADDRESS COUNTER WRAP AROUND ERROR : 0
[15] HMAC_ERROR_1 - HMAC ERROR : 0
```

At the bottom of the console window, there are buttons for Configuration, JTAG, and a serial port indicator. The status bar at the bottom right shows "Diligent JTAG-SMT1 | 10000000".

Run MultiBoot Design

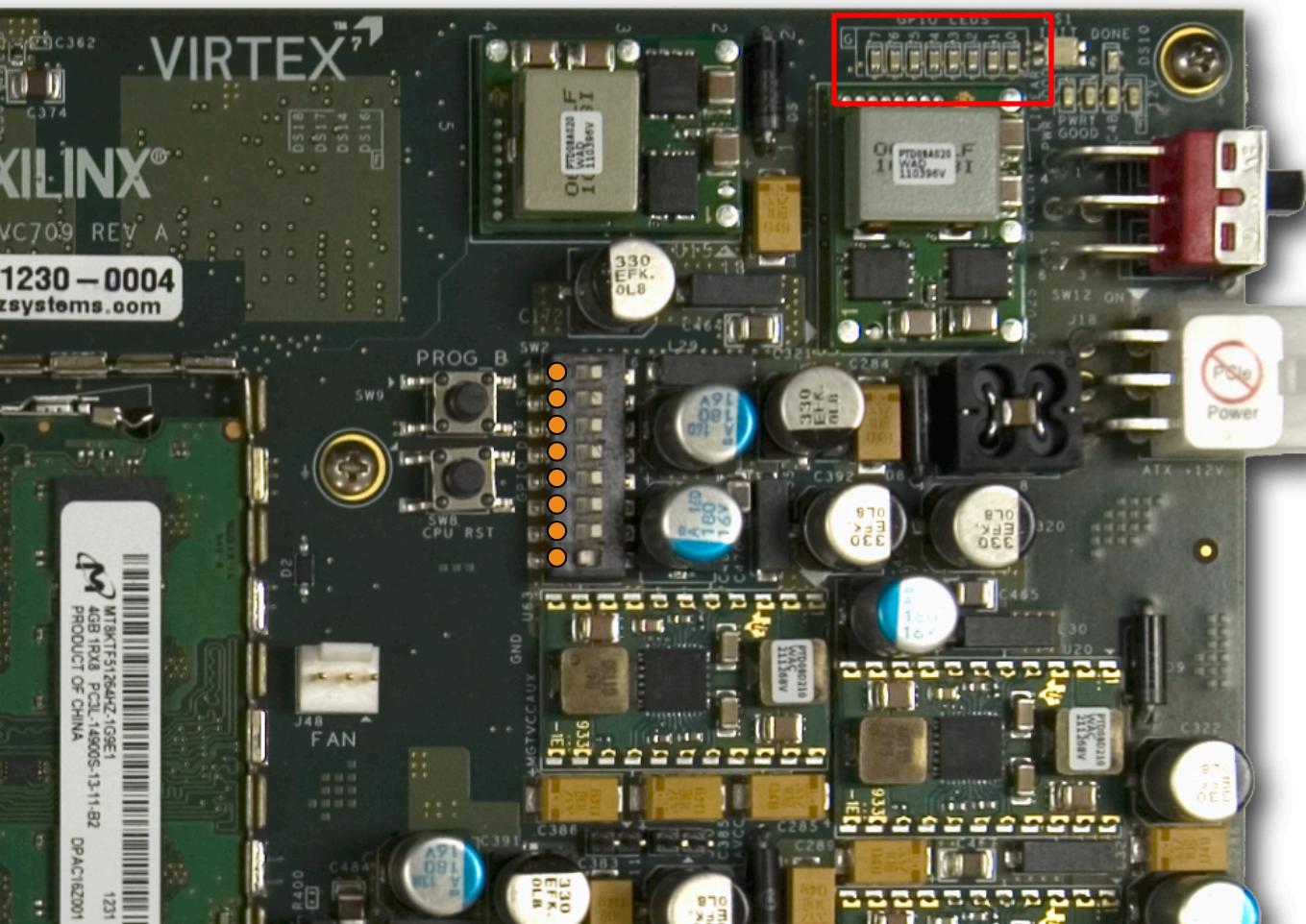
► Set SW2 to 10000000 (1 = on, Position 1 → Position 8)

- Issues an IPROG command to re-configure from a good MultiBoot bitstream
- Set it back to 00000000 when the DONE LED goes out



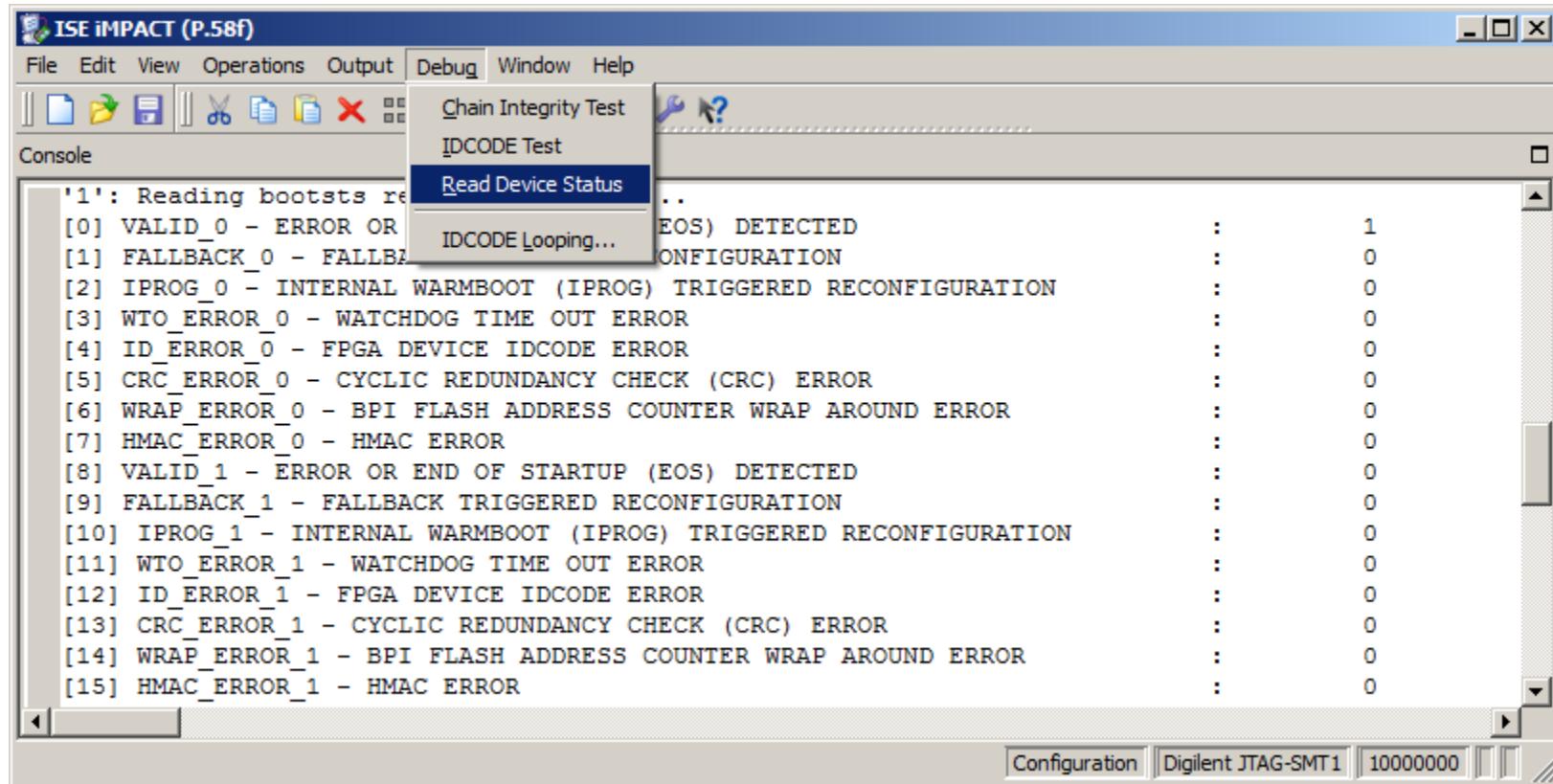
Run MultiBoot Design

- The MultiBoot design, “multiboot.bit”, shows a rapid blinking LED pattern on the GPIO LEDs



Run MultiBoot Design

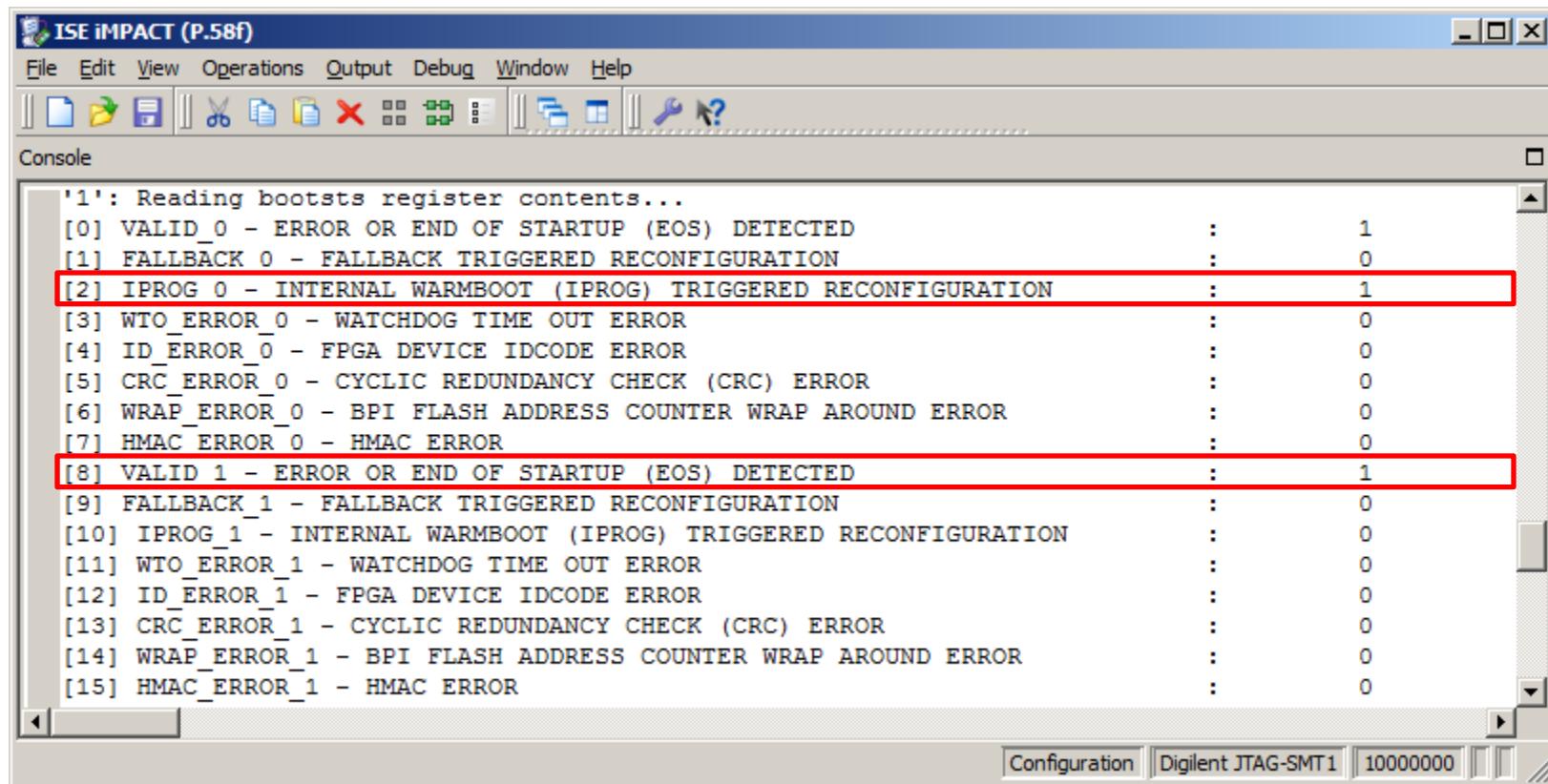
- Select the menu item Debug → Read Device Status
- Scroll down to the “Reading bootsts register contents...” message



Run MultiBoot Design

► The following status register messages are now set:

- IPROG_0
- VALID_1



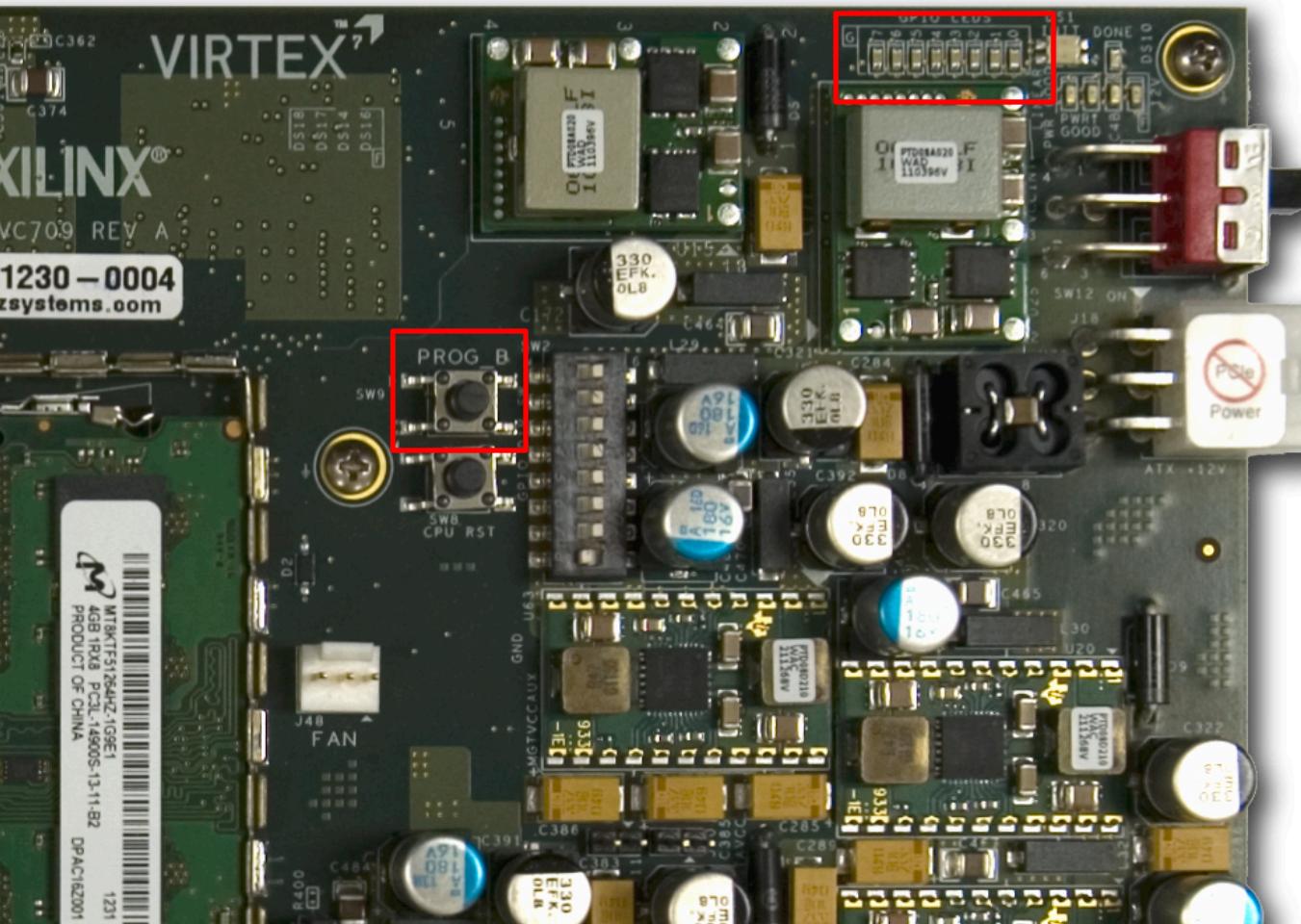
The screenshot shows the ISE iMPACT software interface with a "Console" tab selected. The window title is "ISE iMPACT (P.58f)". The console output displays a list of status register messages. The messages are numbered from [0] to [15] and include the following details:

Message ID	Description	Value
[0]	VALID_0 - ERROR OR END OF STARTUP (EOS) DETECTED	1
[1]	FALLBACK_0 - FALLBACK TRIGGERED RECONFIGURATION	0
[2]	IPROG_0 - INTERNAL WARMBOOT (IPROG) TRIGGERED RECONFIGURATION	1
[3]	WTO_ERROR_0 - WATCHDOG TIME OUT ERROR	0
[4]	ID_ERROR_0 - FPGA DEVICE IDCODE ERROR	0
[5]	CRC_ERROR_0 - CYCLIC REDUNDANCY CHECK (CRC) ERROR	0
[6]	WRAP_ERROR_0 - BPI FLASH ADDRESS COUNTER WRAP AROUND ERROR	0
[7]	HMAC_ERROR_0 - HMAC ERROR	0
[8]	VALID_1 - ERROR OR END OF STARTUP (EOS) DETECTED	1
[9]	FALLBACK_1 - FALLBACK TRIGGERED RECONFIGURATION	0
[10]	IPROG_1 - INTERNAL WARMBOOT (IPROG) TRIGGERED RECONFIGURATION	0
[11]	WTO_ERROR_1 - WATCHDOG TIME OUT ERROR	0
[12]	ID_ERROR_1 - FPGA DEVICE IDCODE ERROR	0
[13]	CRC_ERROR_1 - CYCLIC REDUNDANCY CHECK (CRC) ERROR	0
[14]	WRAP_ERROR_1 - BPI FLASH ADDRESS COUNTER WRAP AROUND ERROR	0
[15]	HMAC_ERROR_1 - HMAC ERROR	0

The messages for IPROG_0 and VALID_1 are highlighted with red boxes.

Run MultiBoot Design

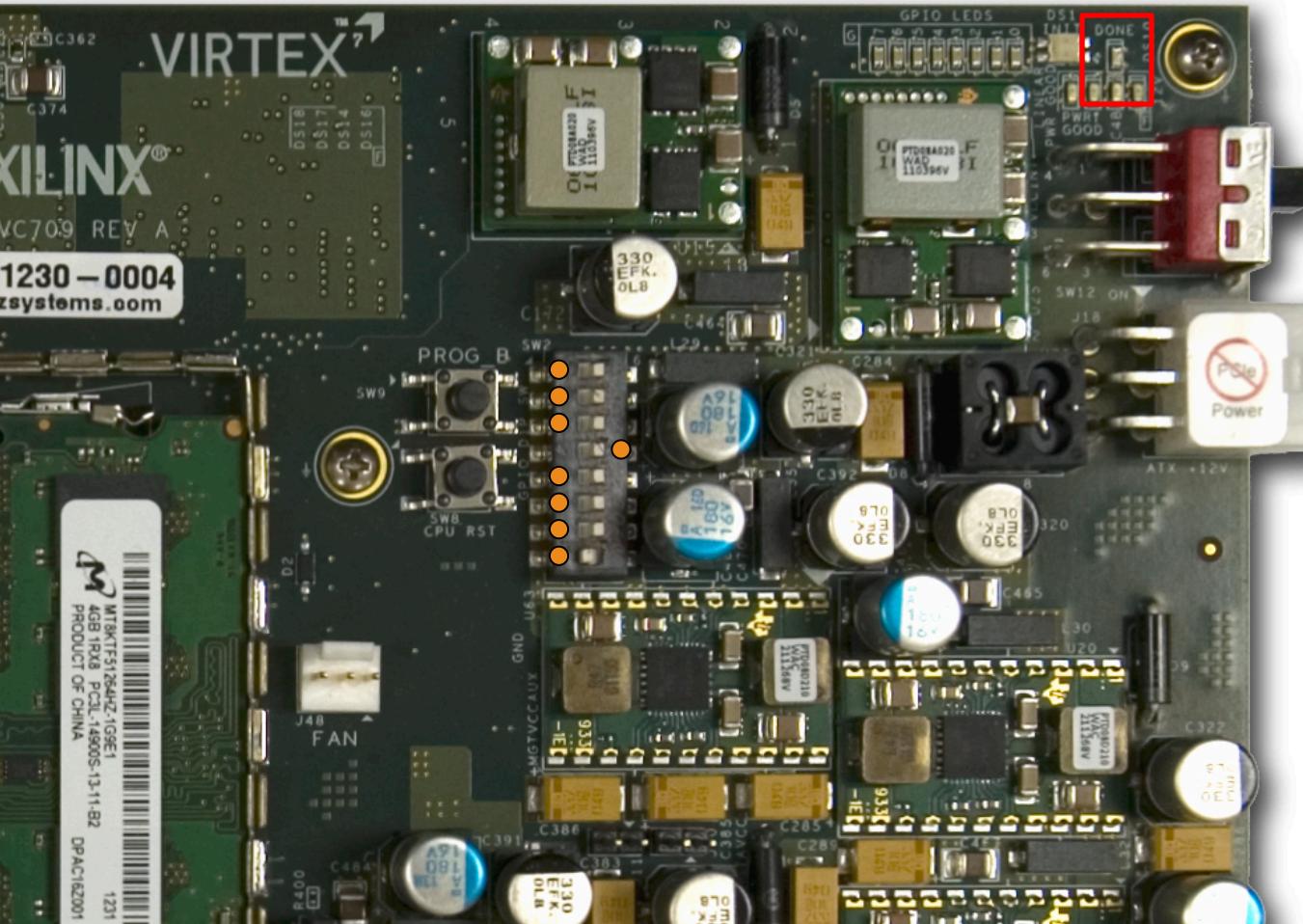
- ▶ Press PROG to return to the golden bitstream
- ▶ Wait until you see the cycling pattern on the LEDs



Run MultiBoot Design

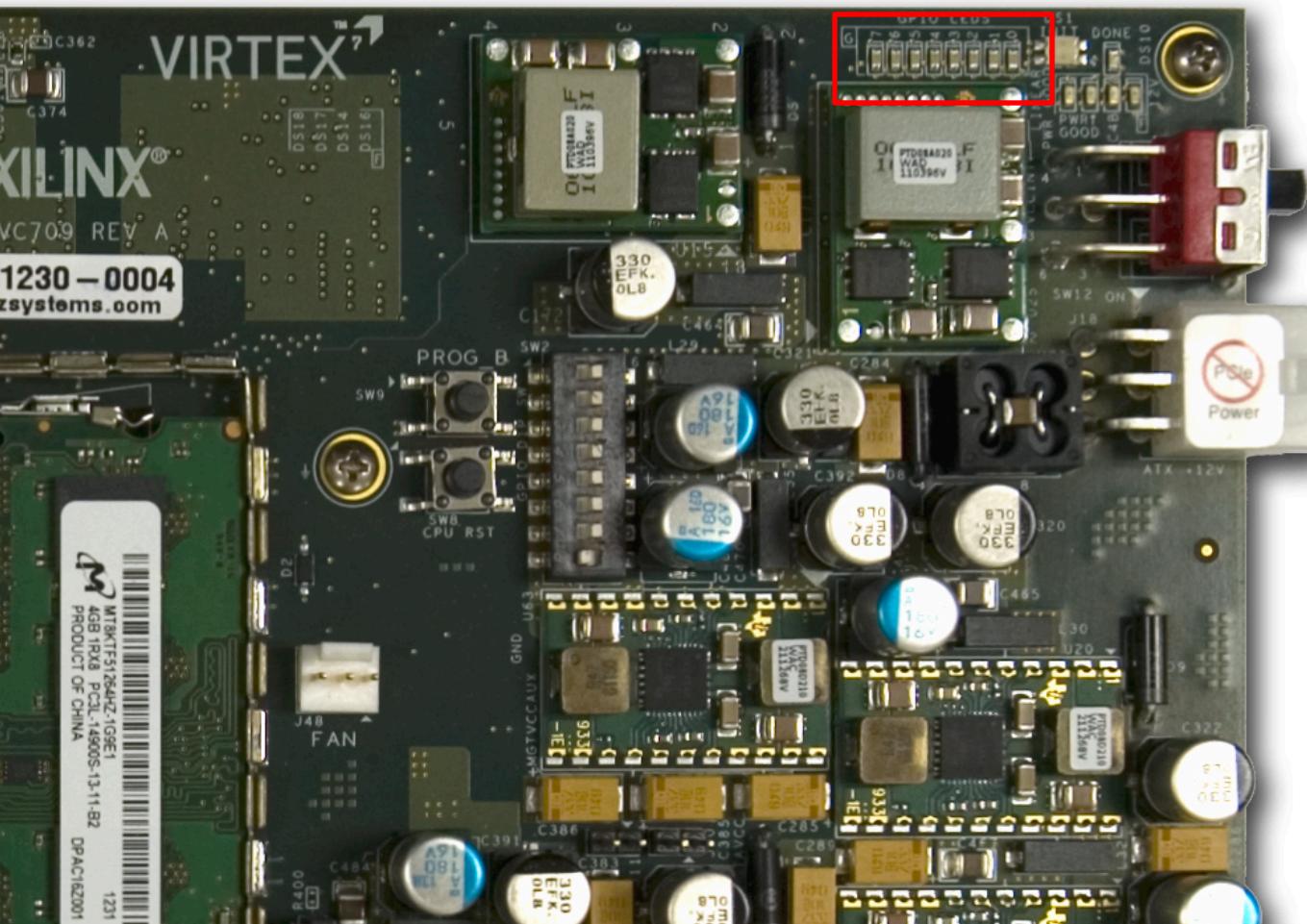
► Set SW2 to 00010000 (1 = on, Position 1 → Position 8)

- Issues an IPROG command to re-configure with a corrupted MultiBoot bitstream
- Set it back to 00000000 when the DONE LED goes out



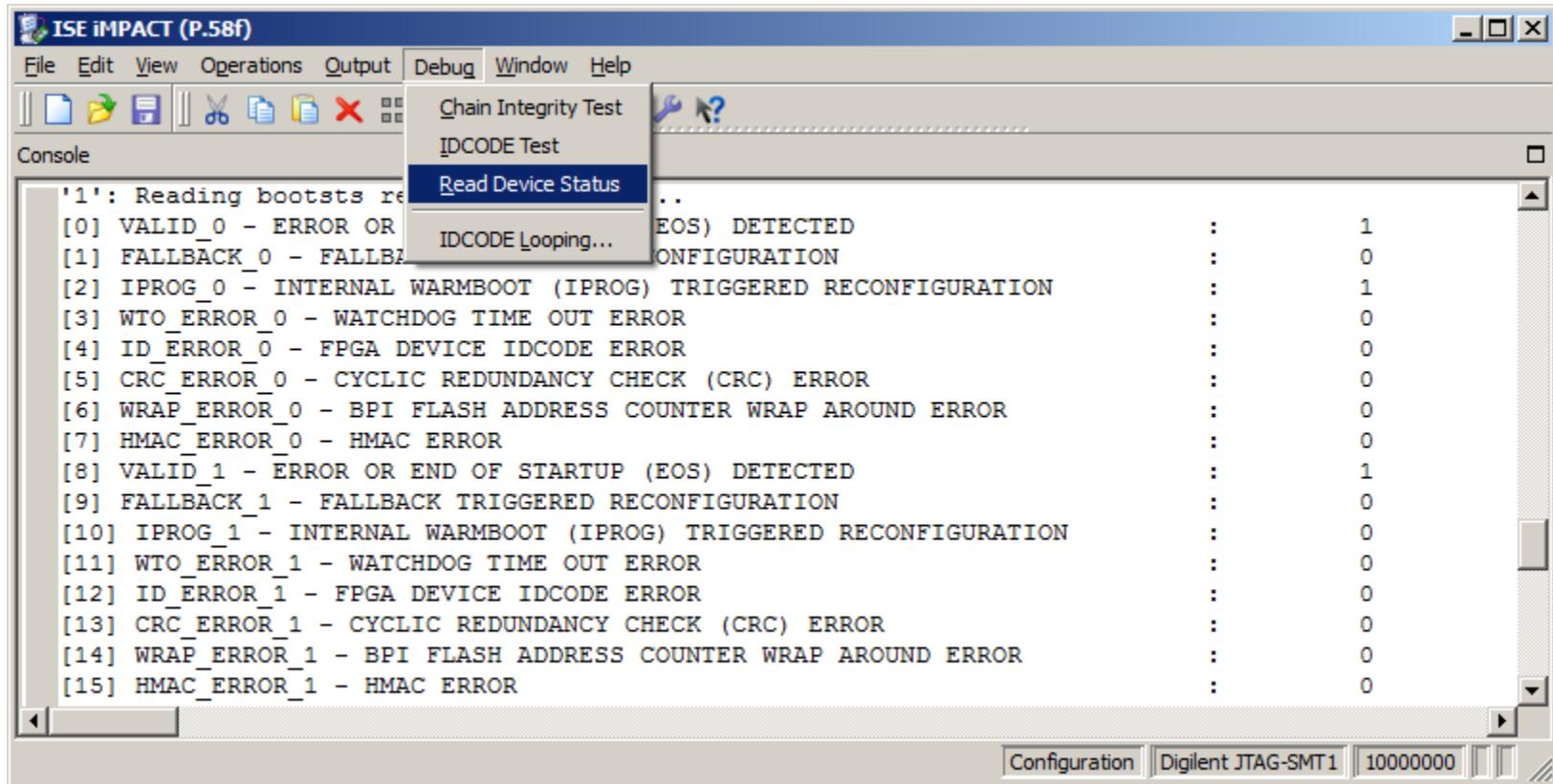
Run MultiBoot Design

- The corrupted.bit generates a CRC error and cannot load the FPGA
- The FPGA falls back to the golden bitstream with a cycling LED pattern



Run MultiBoot Design

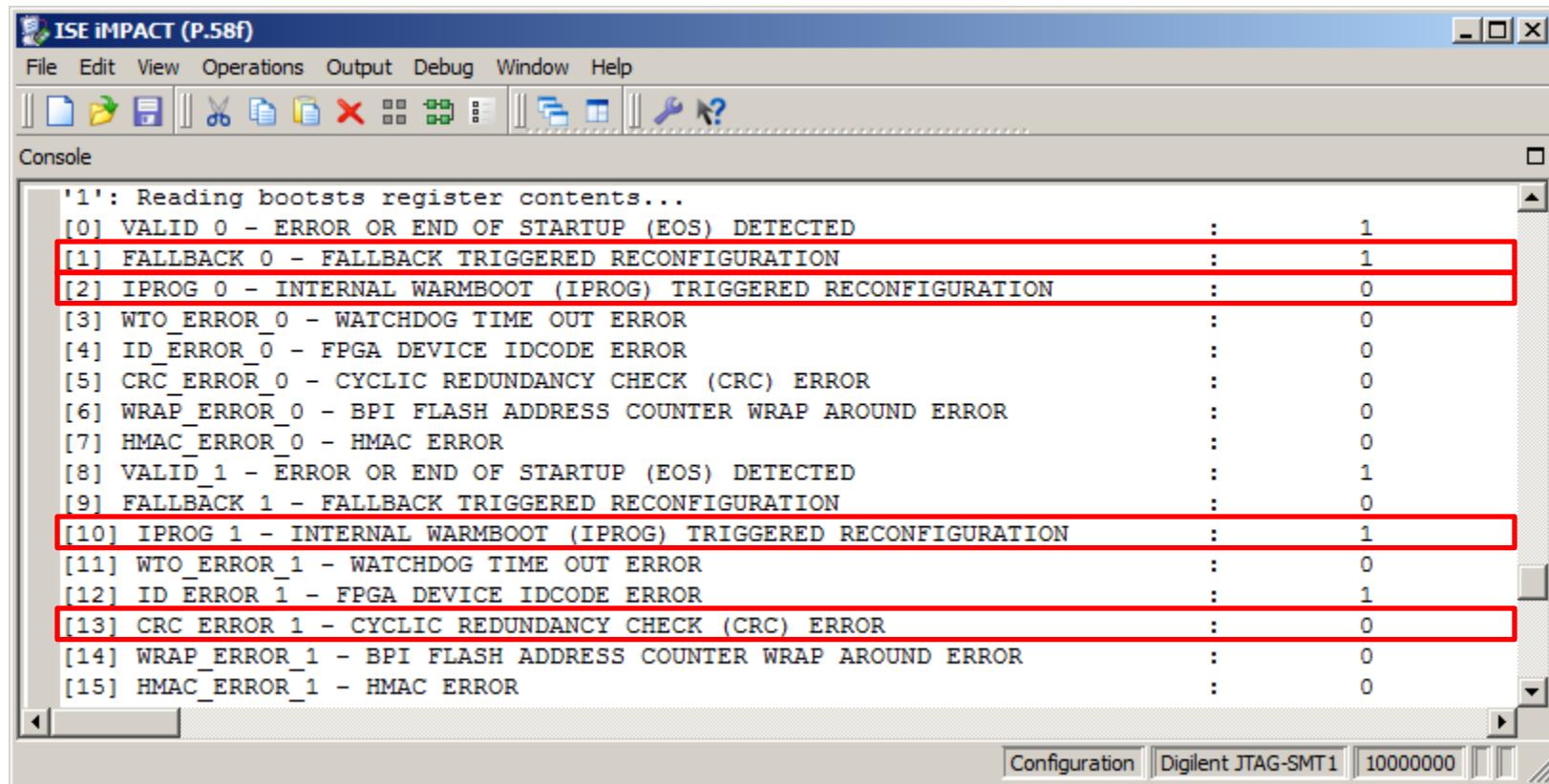
- Select the menu item Debug → Read Device Status
- Scroll down to the “Reading bootsts register contents...” message



Run MultiBoot Design

► The following status register messages are now seen:

- IPROG_0 is no longer active
- FALBACK_0, IPROG_1, and CRC_ERROR_1 are now active



The screenshot shows the ISE iMPACT software interface with a "Console" tab selected. The window title is "ISE iMPACT (P.58f)". The console output displays a list of status register messages. Several messages are highlighted with red boxes:

```
'1': Reading bootsts register contents...
[0] VALID_0 - ERROR OR END OF STARTUP (EOS) DETECTED : 1
[1] FALBACK_0 - FALBACK TRIGGERED RECONFIGURATION : 1
[2] IPROG_0 - INTERNAL WARMBOOT (IPROG) TRIGGERED RECONFIGURATION : 0
[3] WTO_ERROR_0 - WATCHDOG TIME OUT ERROR : 0
[4] ID_ERROR_0 - FPGA DEVICE IDCODE ERROR : 0
[5] CRC_ERROR_0 - CYCLIC REDUNDANCY CHECK (CRC) ERROR : 0
[6] WRAP_ERROR_0 - BPI FLASH ADDRESS COUNTER WRAP AROUND ERROR : 0
[7] HMAC_ERROR_0 - HMAC ERROR : 0
[8] VALID_1 - ERROR OR END OF STARTUP (EOS) DETECTED : 1
[9] FALBACK_1 - FALBACK TRIGGERED RECONFIGURATION : 0
[10] IPROG_1 - INTERNAL WARMBOOT (IPROG) TRIGGERED RECONFIGURATION : 1
[11] WTO_ERROR_1 - WATCHDOG TIME OUT ERROR : 0
[12] ID_ERROR_1 - FPGA DEVICE IDCODE ERROR : 1
[13] CRC_ERROR_1 - CYCLIC REDUNDANCY CHECK (CRC) ERROR : 0
[14] WRAP_ERROR_1 - BPI FLASH ADDRESS COUNTER WRAP AROUND ERROR : 0
[15] HMAC_ERROR_1 - HMAC ERROR : 0
```

At the bottom of the window, there are tabs for "Configuration", "Diligent JTAG-SMT1", and "10000000".

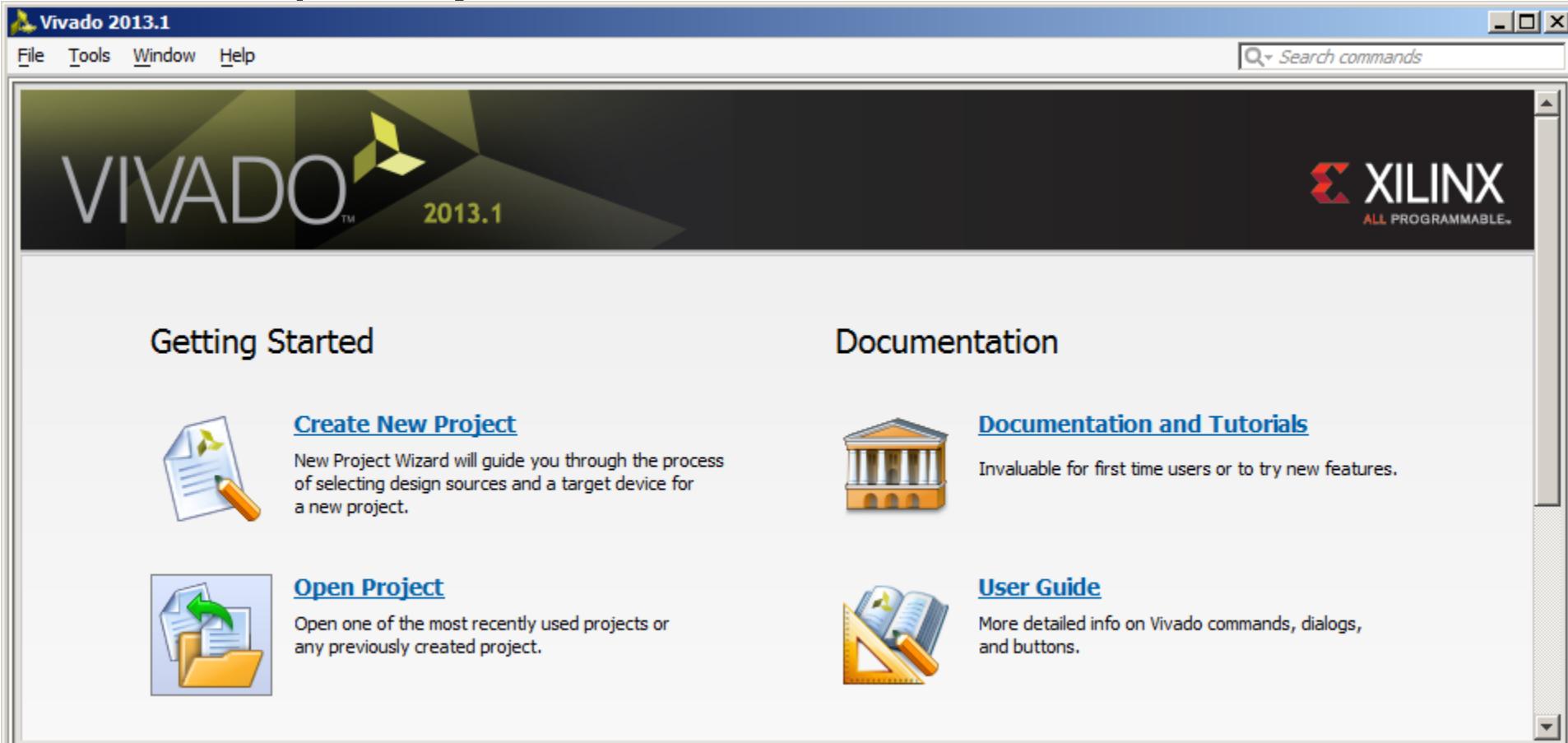
Compile VC709 MultiBoot Design

Compile VC709 MultiBoot Design

► Open Vivado

Start → All Programs → Xilinx Design Tools → Vivado 2013.1 → Vivado

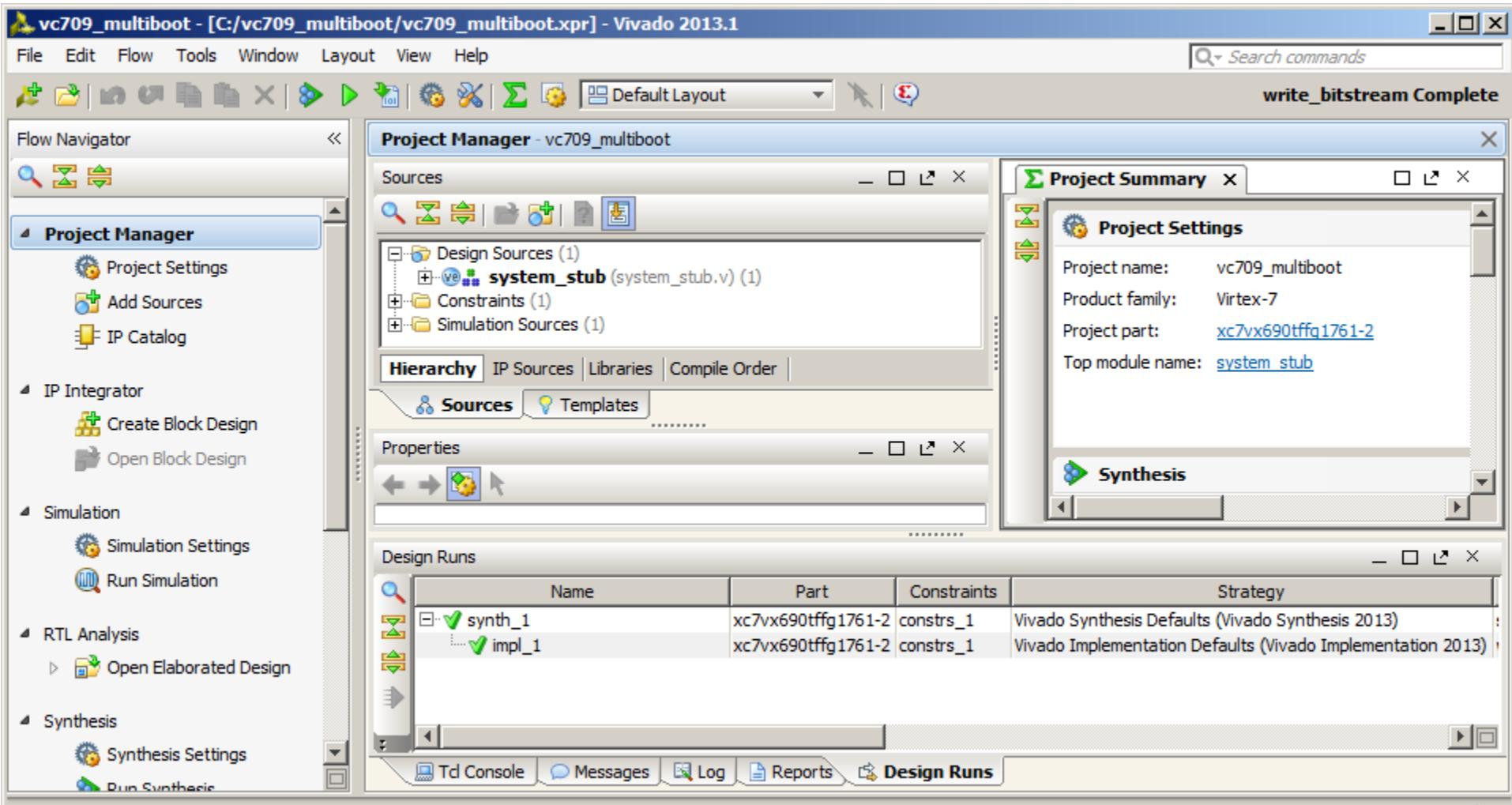
► Select Open Project



Compile VC709 MultiBoot Design

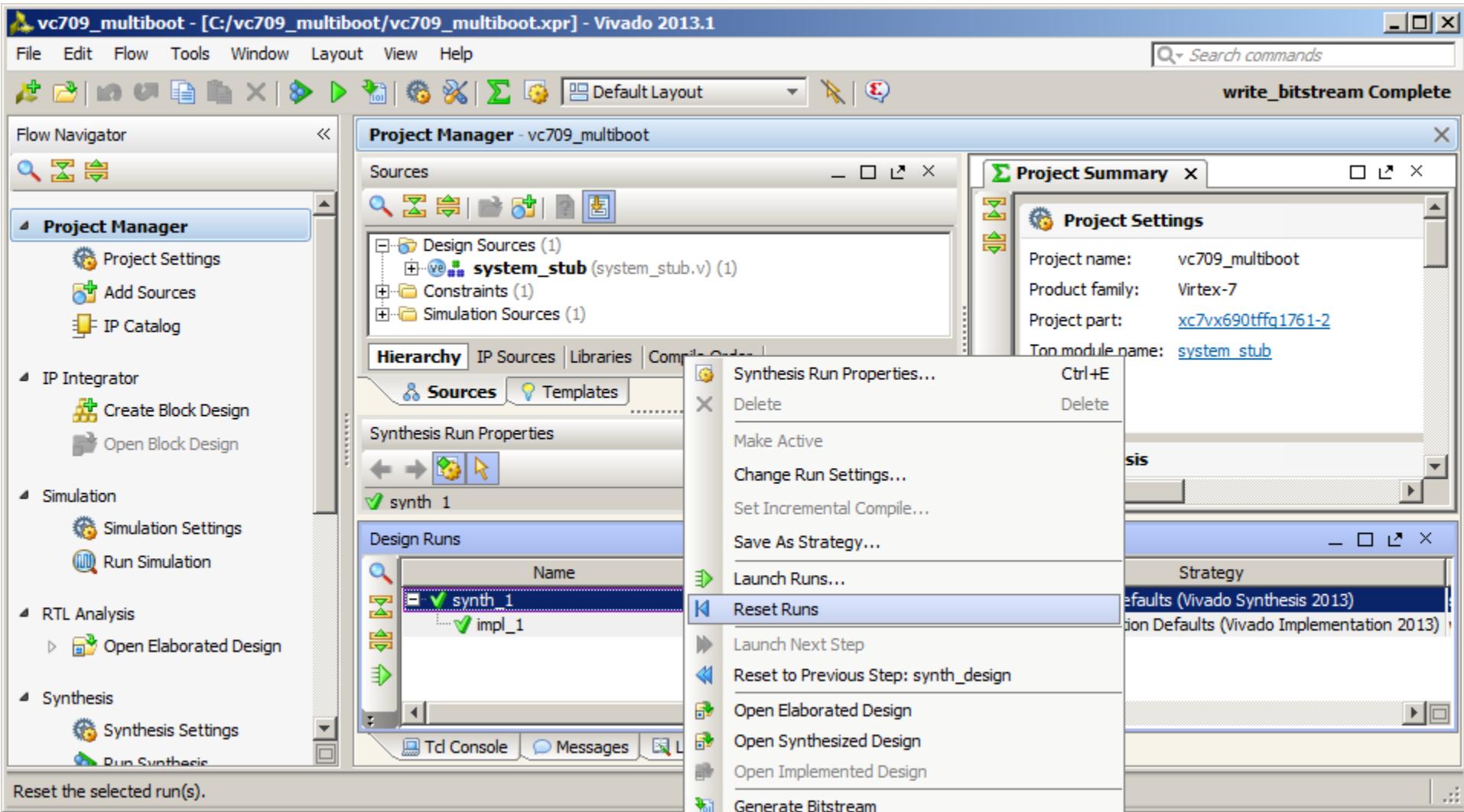
► Open the VC709 Design:

- <Design Name>\vc709_multiboot.xpr



Compile VC709 MultiBoot Design

- The design is fully implemented; you can recompile, or export to SDK
 - To recompile, right click on synth_1 and select Reset Runs

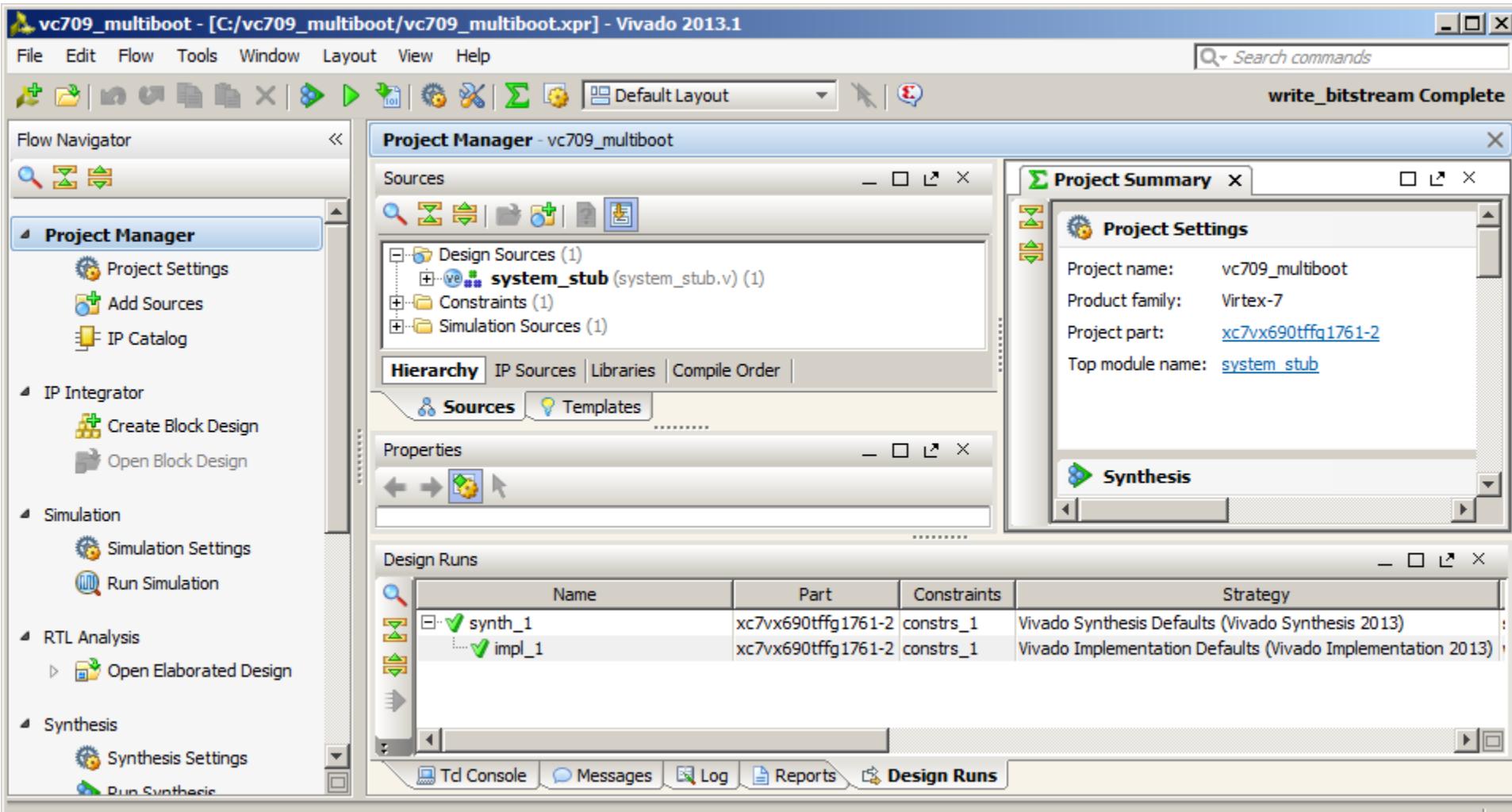


Note: Presentation applies to the VC709

XILINX ➤ ALL PROGRAMMABLE™

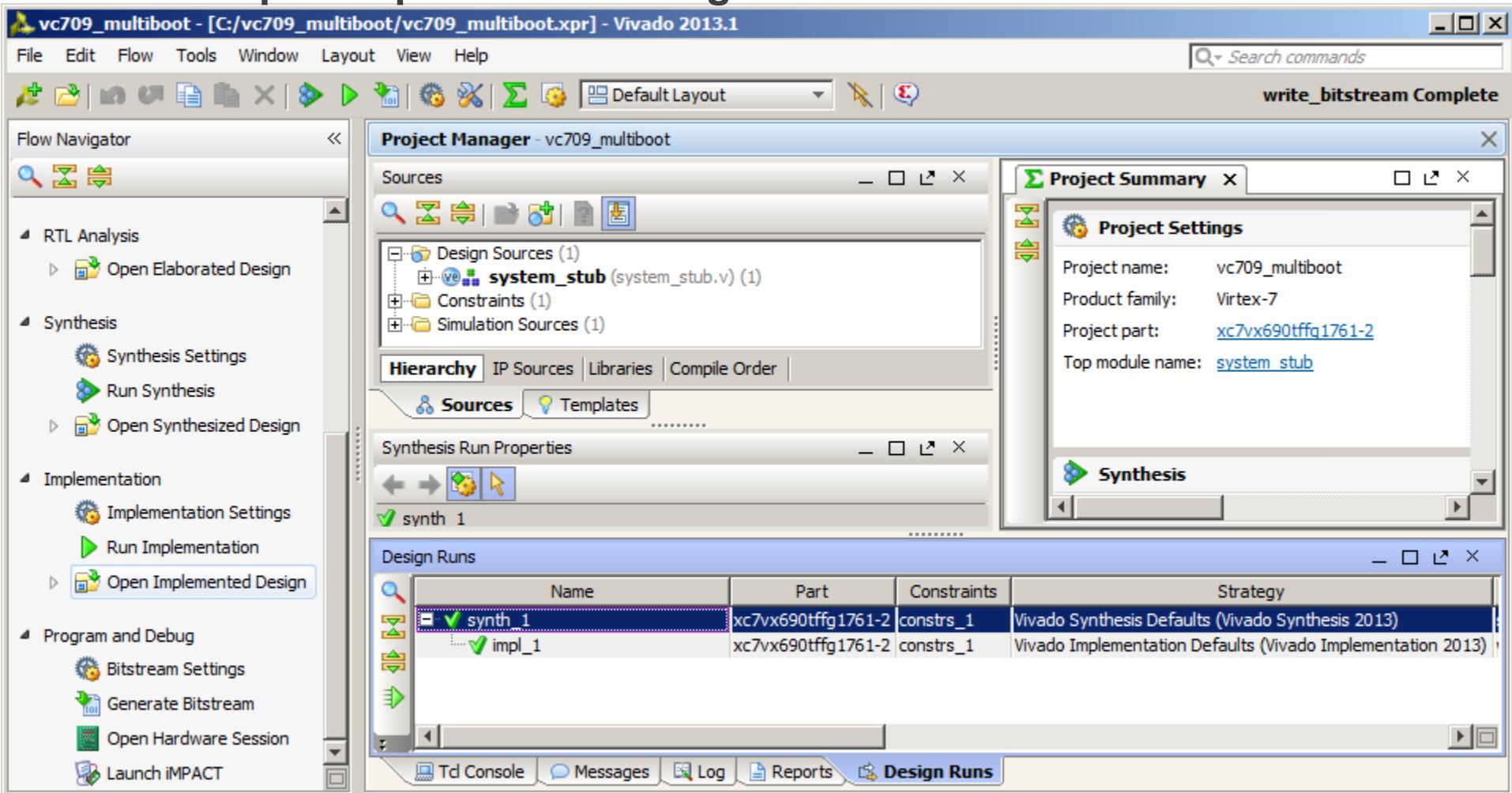
Compile VC709 MultiBoot Design

- Once done, both the Synthesis and Implementation will have green check marks



Compile VC709 MultiBoot Design

- Before exporting to SDK, the design must be opened
- Click Open Implemented Design



Analyze and constrain an Implemented Design.

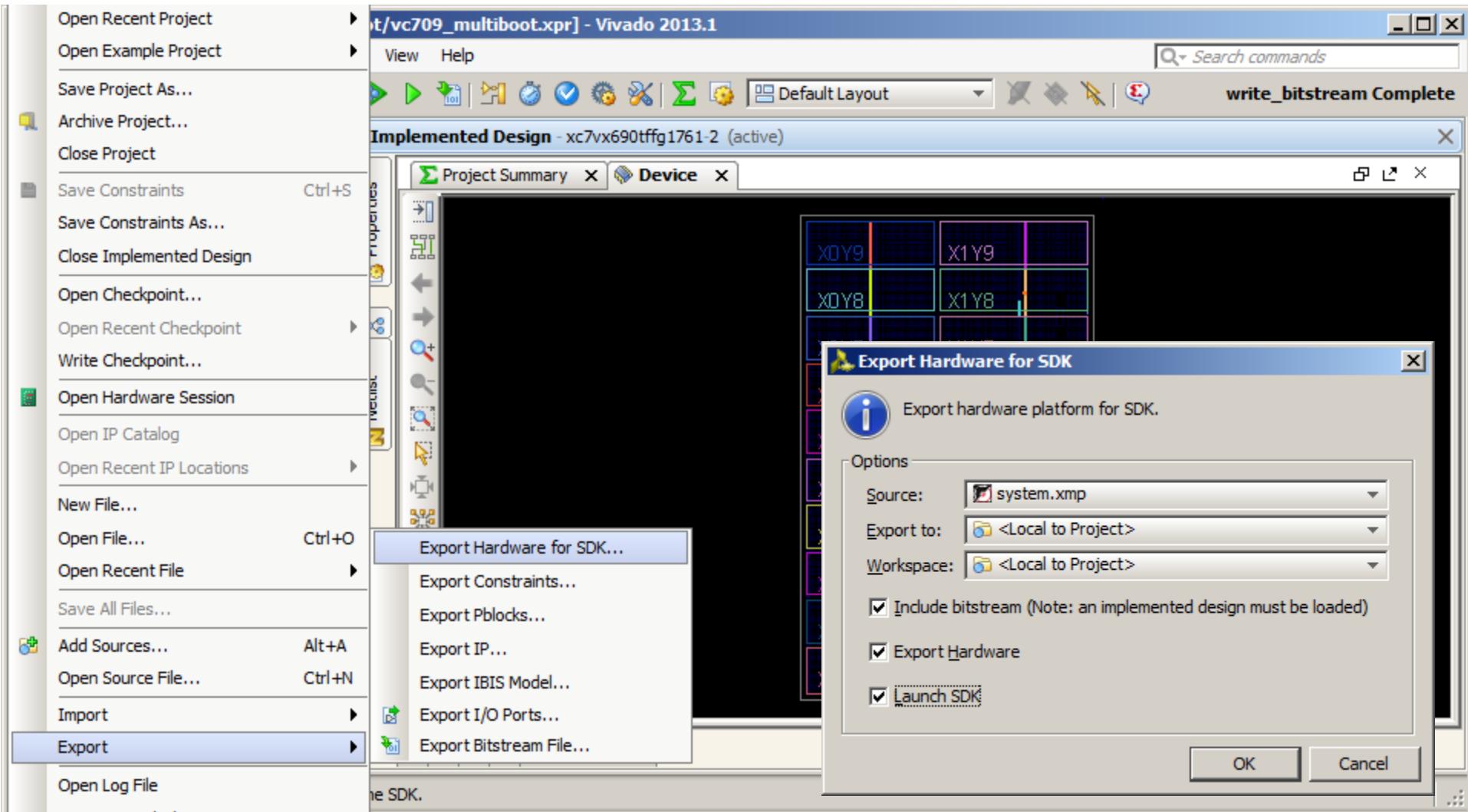
Note: Presentation applies to the VC709

XILINX ➤ ALL PROGRAMMABLE™

Compile VC709 MultiBoot Design

► Select File → Export → Export hardware for SDK...

► Select Launch SDK and click OK



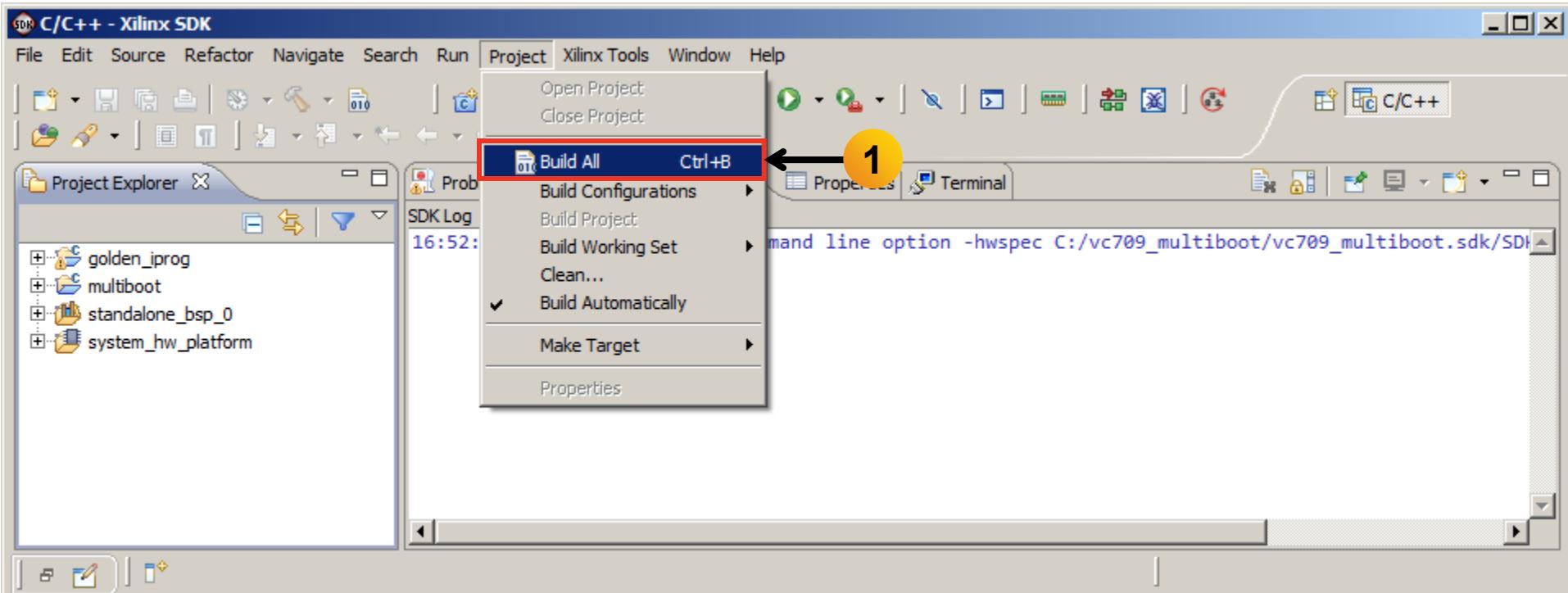
Note: Presentation applies to the VC709

 XILINX ➤ ALL PROGRAMMABLE™

Compile VC709 Software in SDK

► SDK Software Compile - Build ELF files in SDK

- Project should build automatically; if not, select Project → Build All (1)

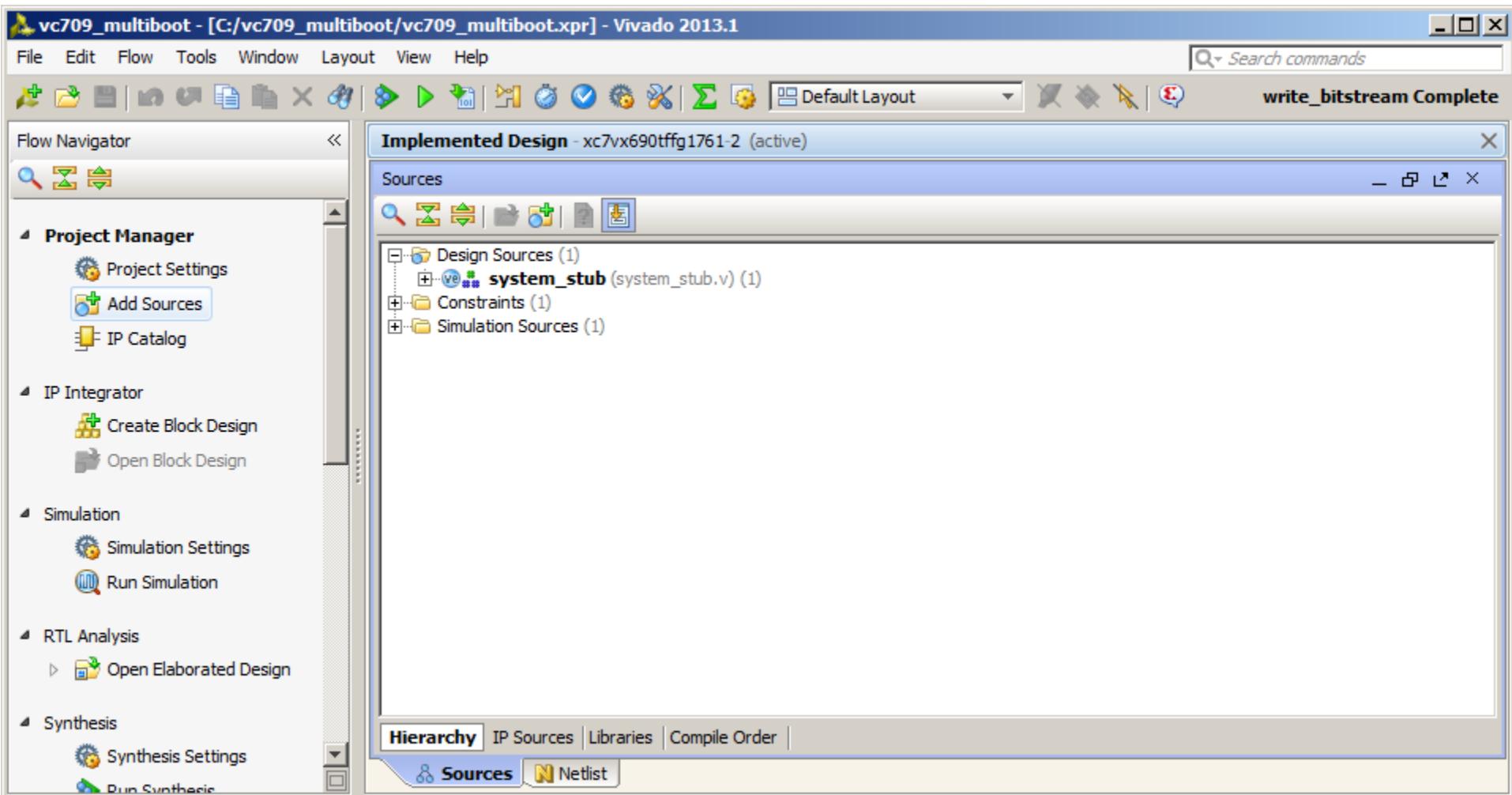


Compile VC709 MultiBoot Design

- With Vivado, a special flow is used for compressed embedded designs
 - SDK can only program an ELF file into an uncompressed bitstream
 - Compression is used here to fit the three MultiBoot bitstreams into a smaller area for programming the BPI flash
- After SDK has compiled the ELF files, they must be associated with the design
 - To do this, we will use the **Add Sources** command, followed by the **Associate ELF** command

Compile VC709 MultiBoot Design

► Select Add Sources



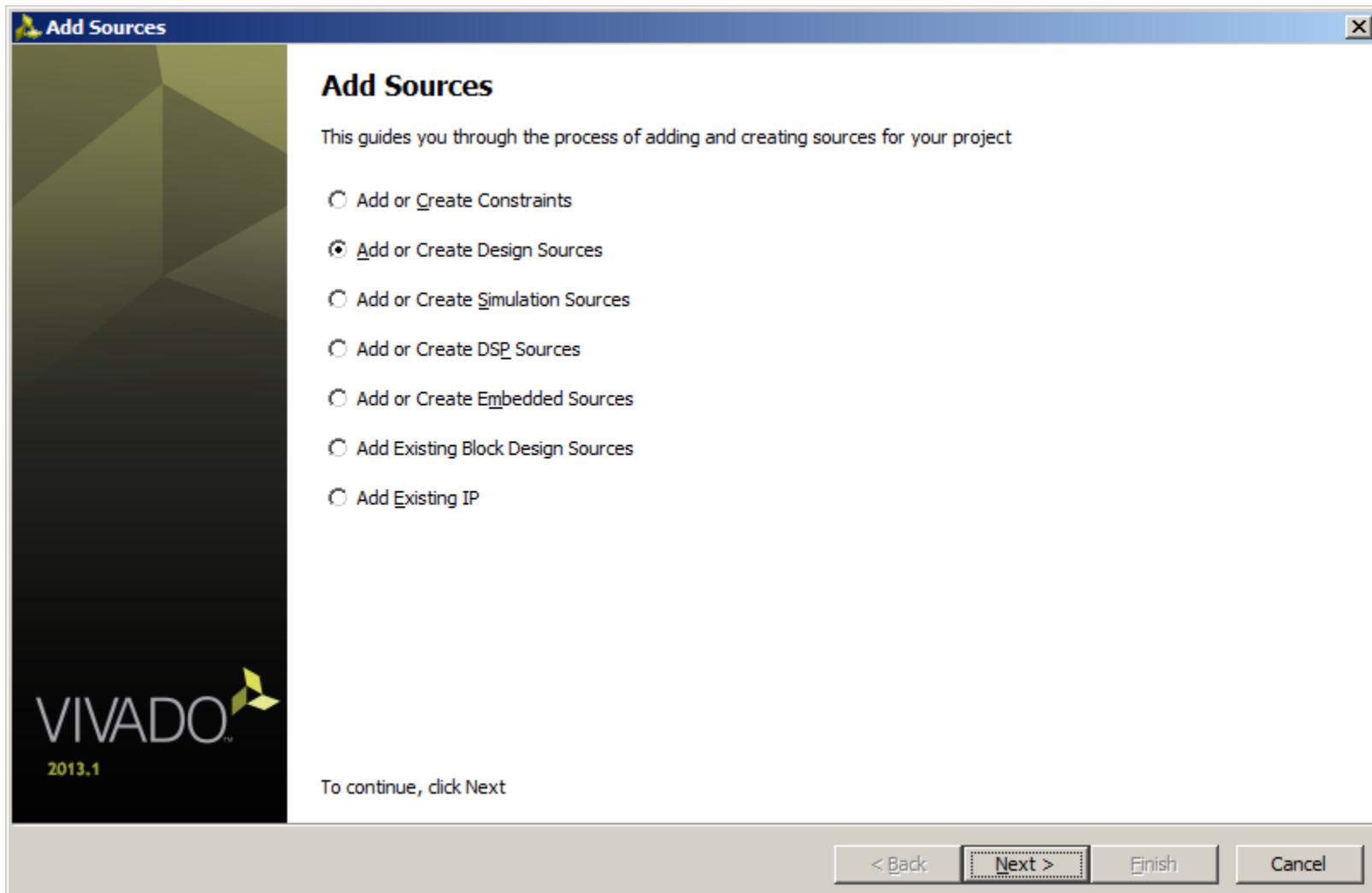
Specify and/or create source files to add to the project.

Note: Presentation applies to the VC709

 XILINX  ALL PROGRAMMABLE

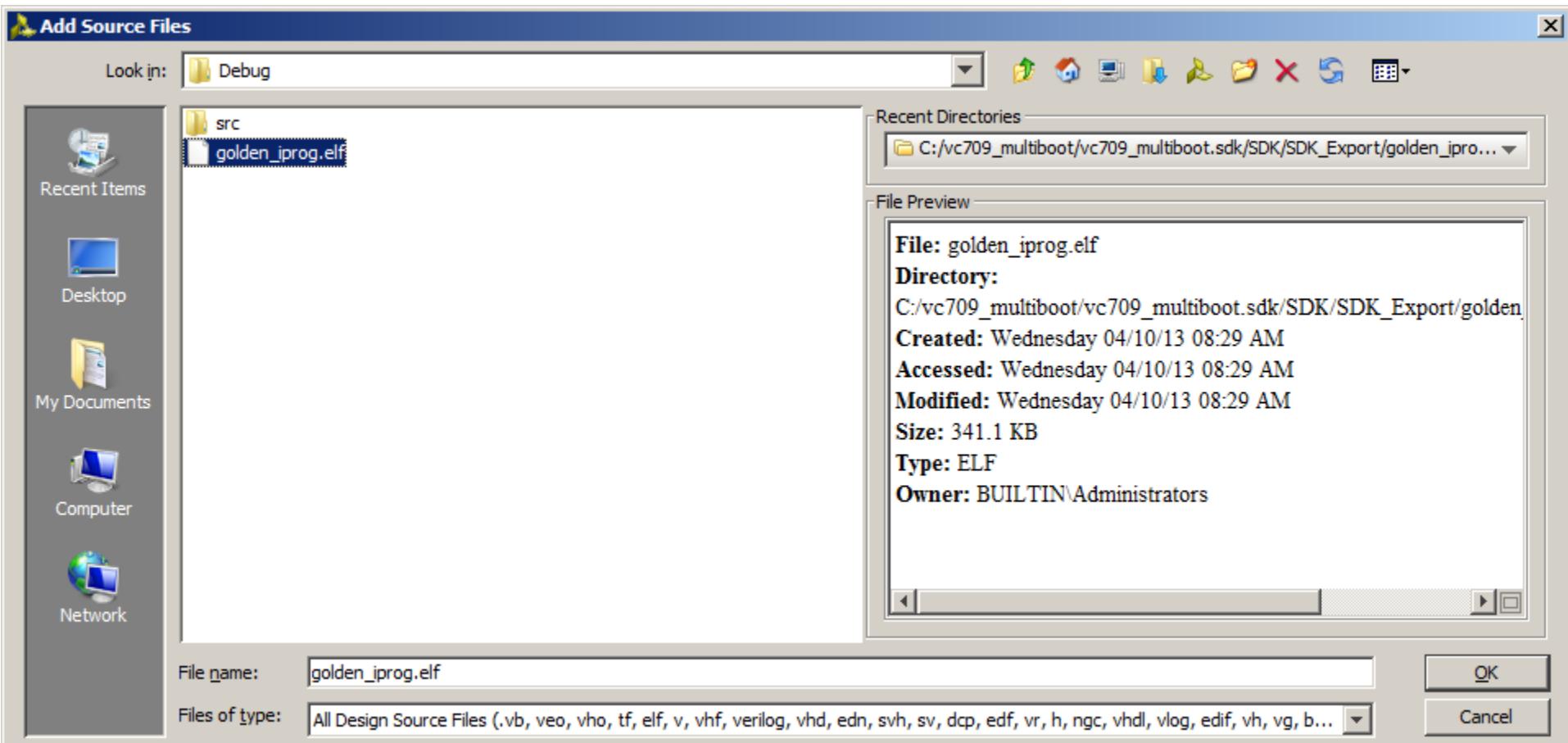
Compile VC709 MultiBoot Design

► Select Add or Create Design Sources



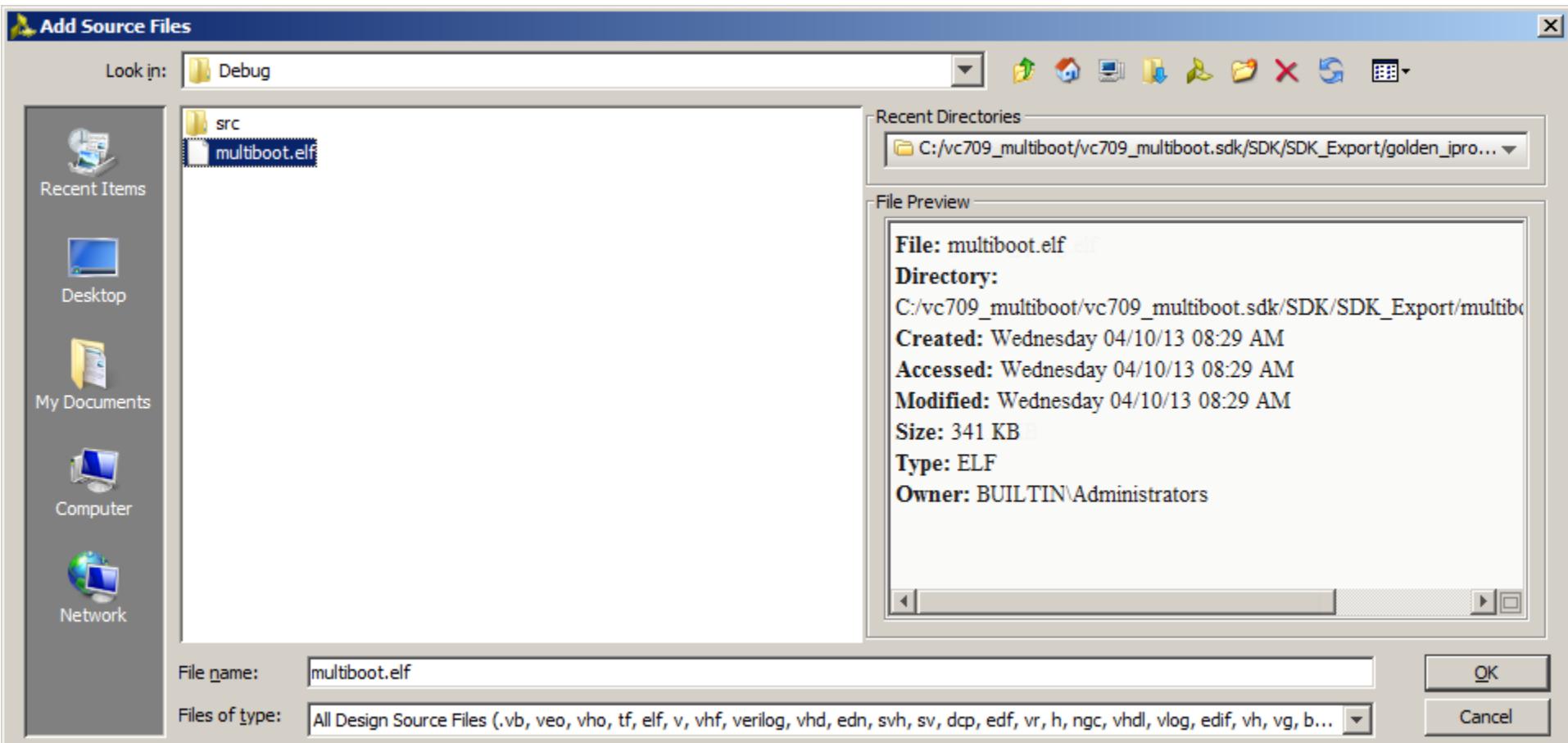
Compile VC709 MultiBoot Design

► Add golden_iprog.elf from the SDK tree



Compile VC709 MultiBoot Design

► Add multiboot.elf from the SDK tree

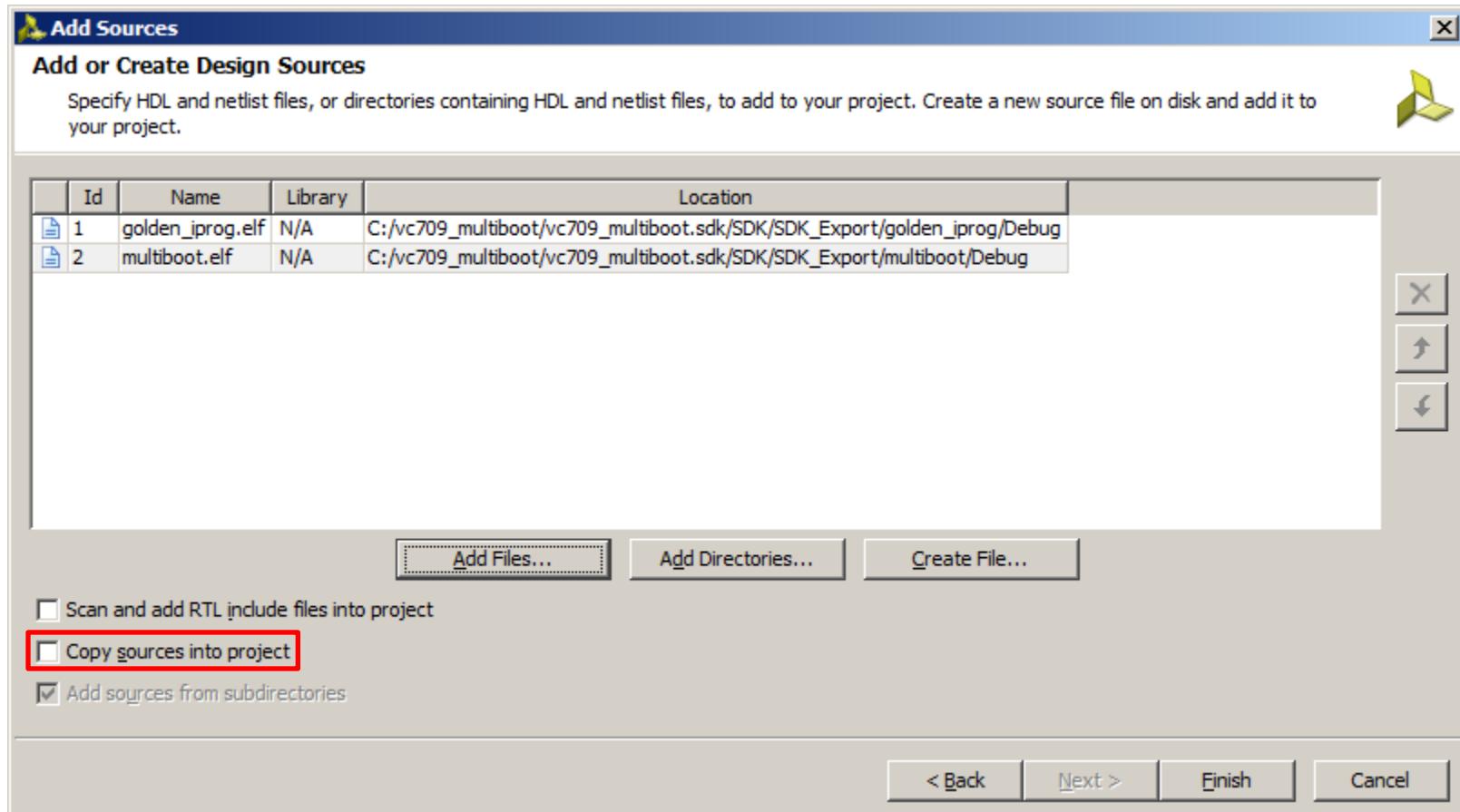


Compile VC709 MultiBoot Design

► Make sure Copy sources into project is deselected

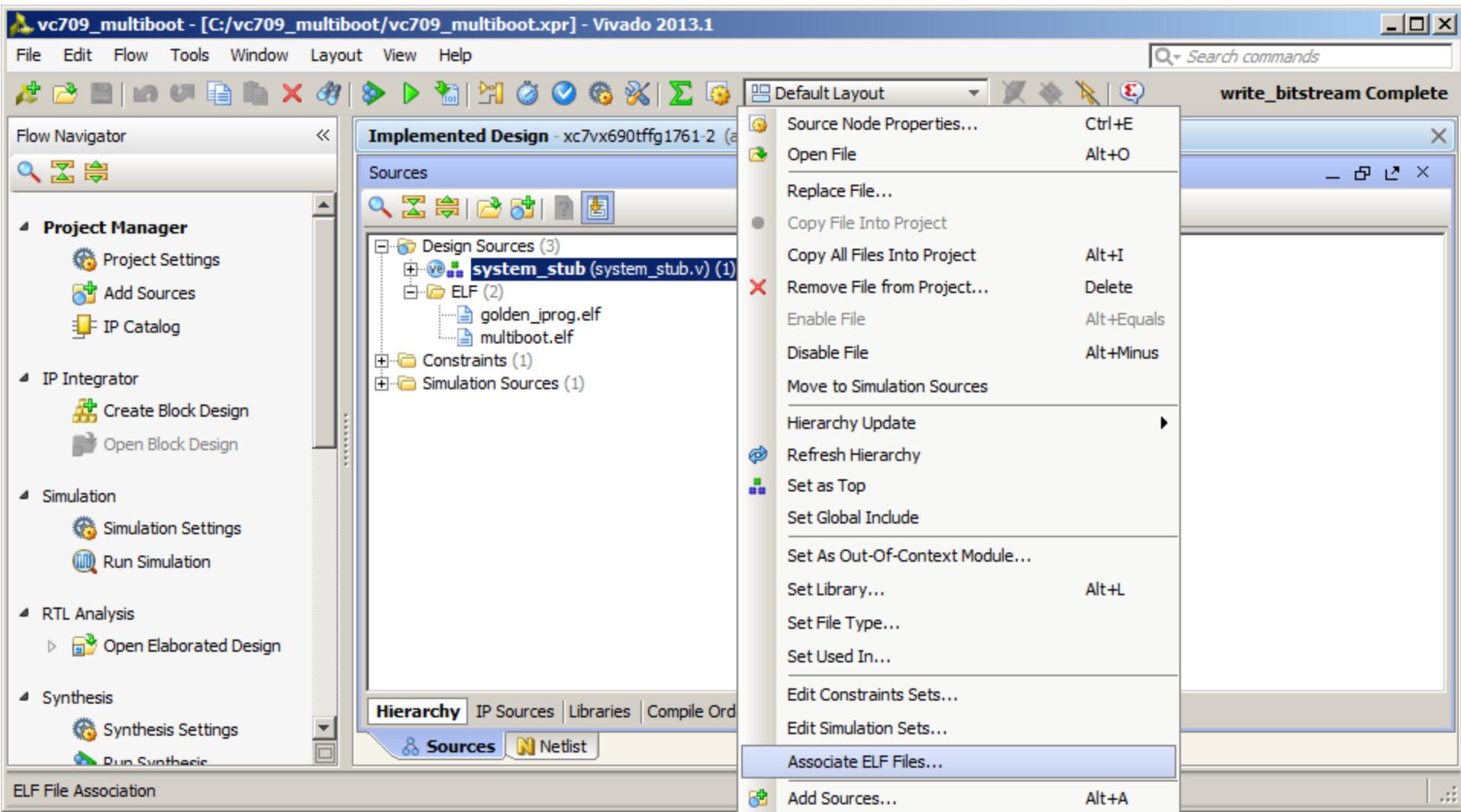
- This eliminates the need to re-Associate the ELF each time it is recompiled

► Click Finish



Compile VC709 MultiBoot Design

► Right-click on the Design and select Associate ELF files

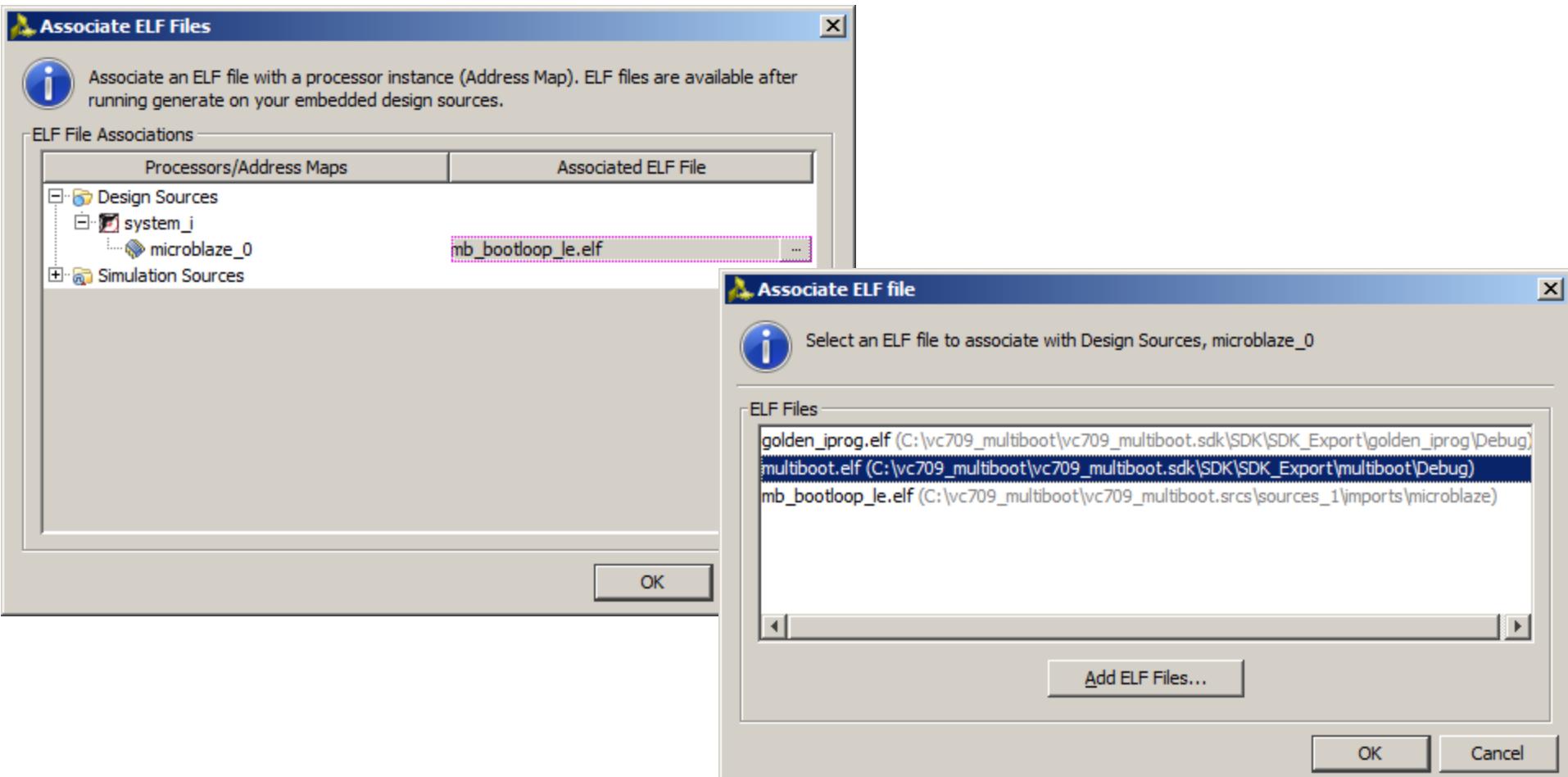


Note: Presentation applies to the VC709

XILINX ➤ ALL PROGRAMMABLE™

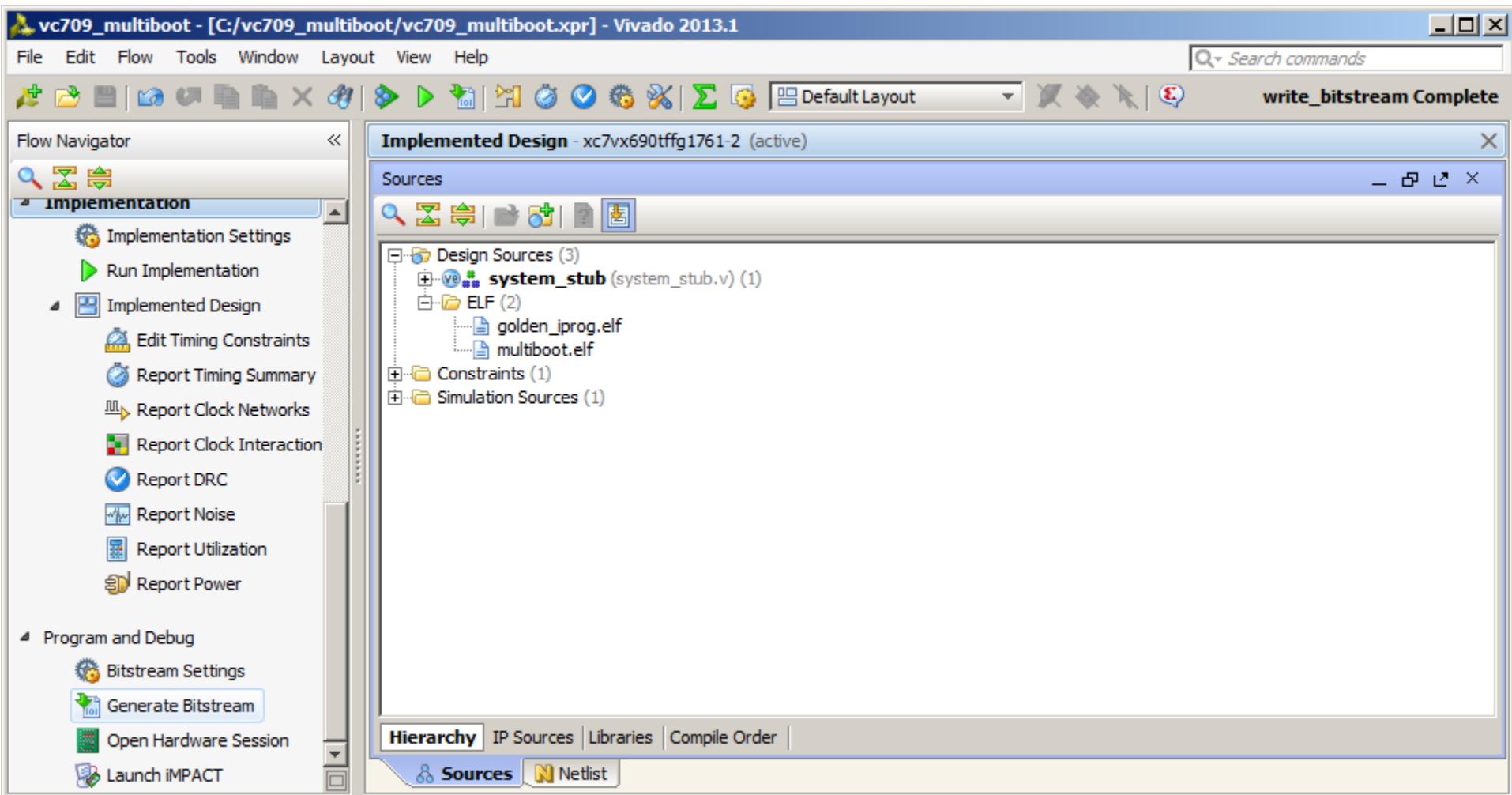
Compile VC709 MultiBoot Design

- By default, a bootloop ELF is associated with the Design
- Click the button to the right of the bootloop ELF; select the multiboot.elf then click OK twice



Compile VC709 MultiBoot Design

► Select Generate Bitstream



Generate a programming file after implementation.

Note: Presentation applies to the VC709

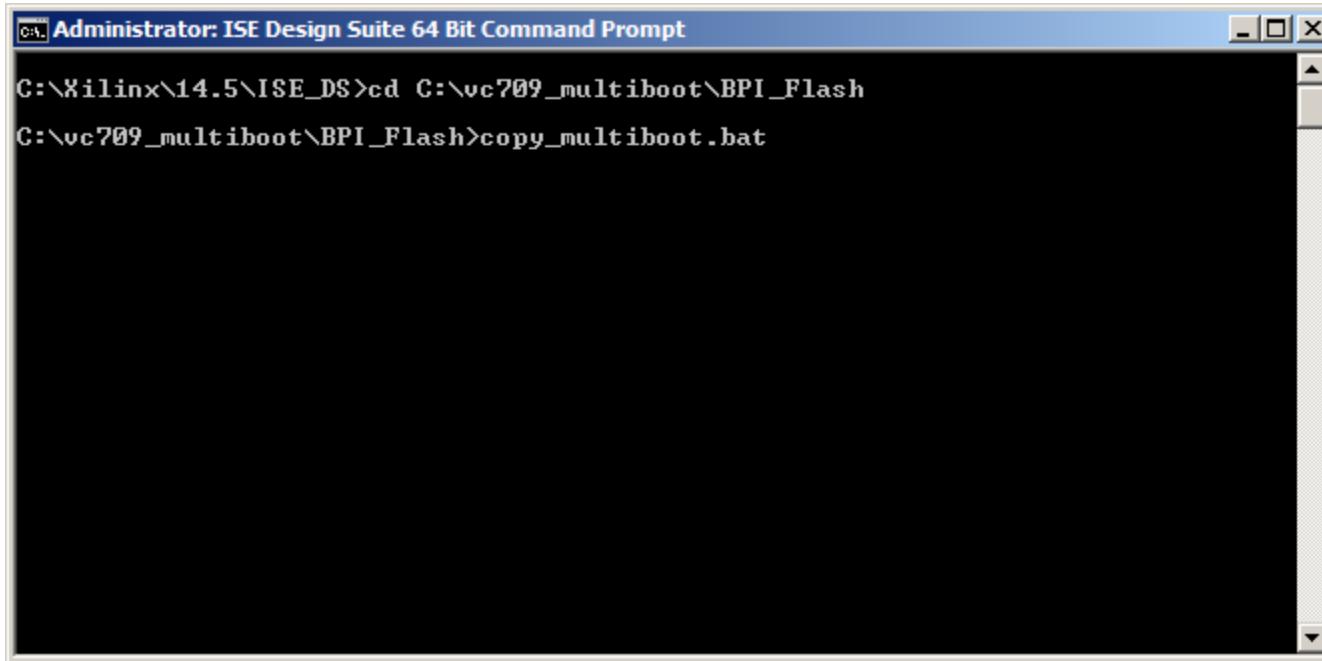
 XILINX  ALL PROGRAMMABLE

Generate VC709 PROM File

- Once the `write_bitstream` command finishes, start a ISE Design Suite Command Prompt and enter these commands:

```
cd C:\vc709_multiboot\BPI_Flash
```

```
copy_multiboot.bat
```

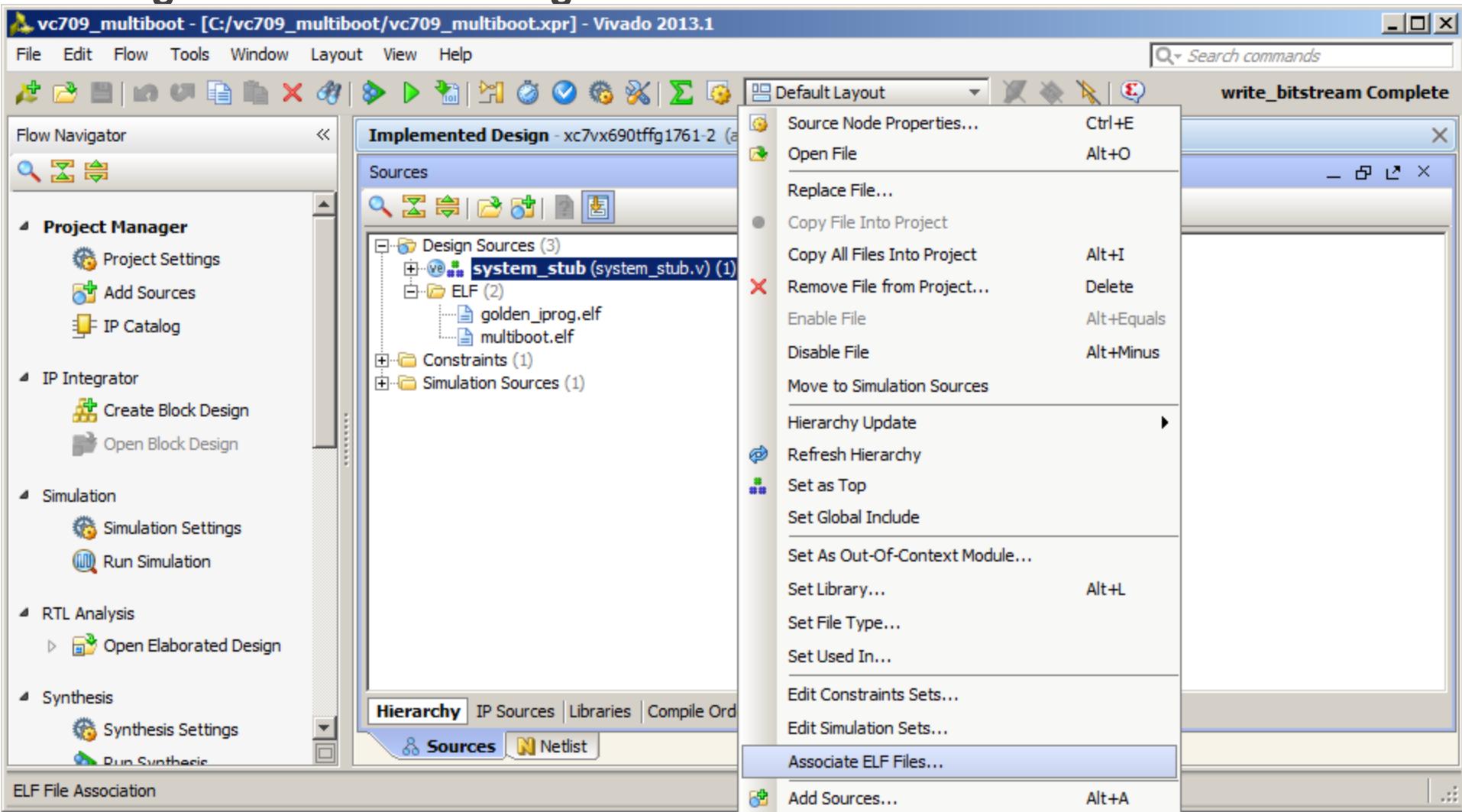


The image shows a Windows Command Prompt window titled "Administrator: ISE Design Suite 64 Bit Command Prompt". The window contains the following text:

```
C:\Xilinx\14.5\ISE_DS>cd C:\vc709_multiboot\BPI_Flash  
C:\vc709_multiboot\BPI_Flash>copy_multiboot.bat
```

Compile VC709 MultiBoot Design

- Now associate the second ELF file
- Right-click on the Design and select Associate ELF files

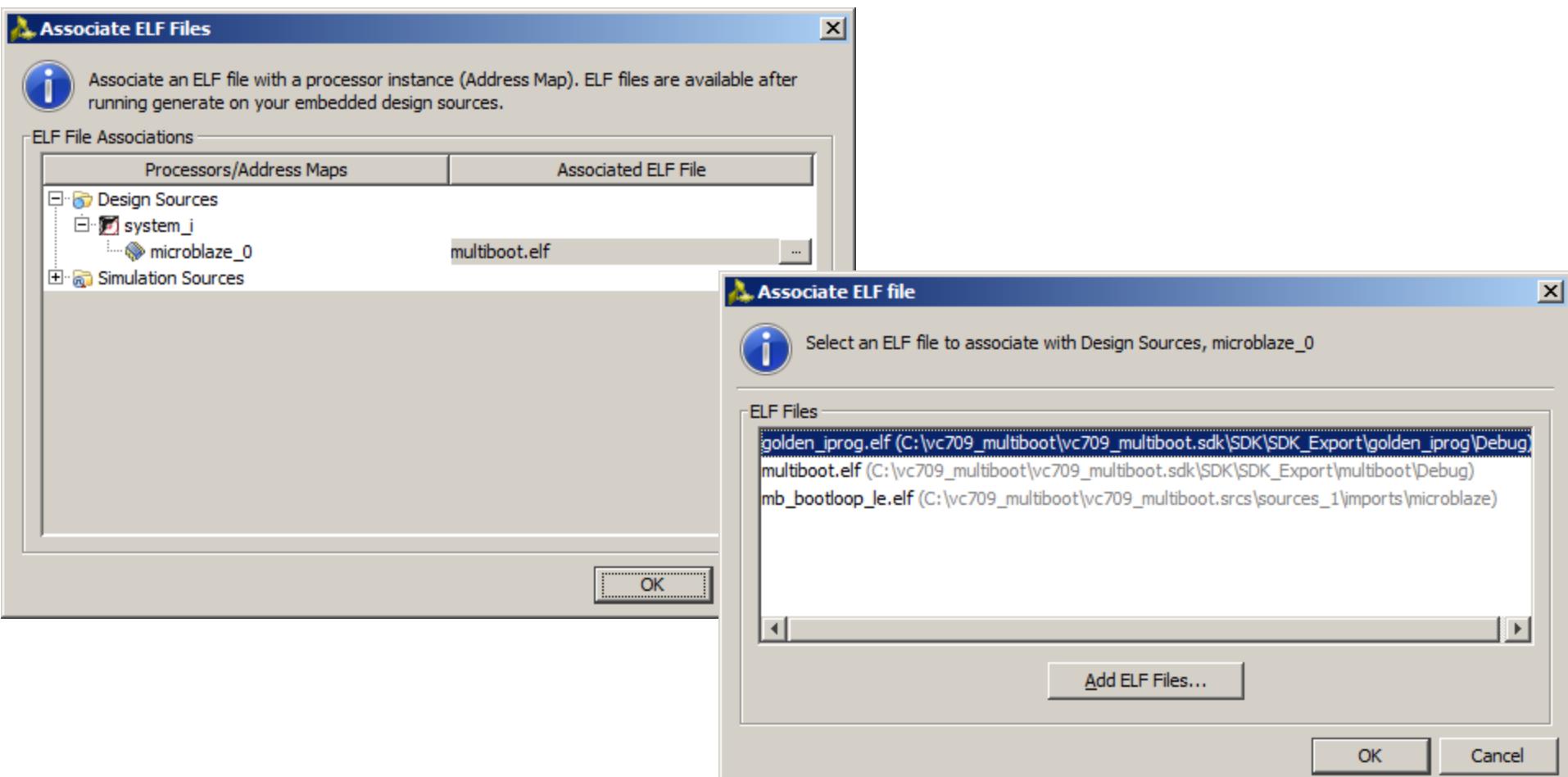


Note: Presentation applies to the VC709

XILINX ➤ ALL PROGRAMMABLE™

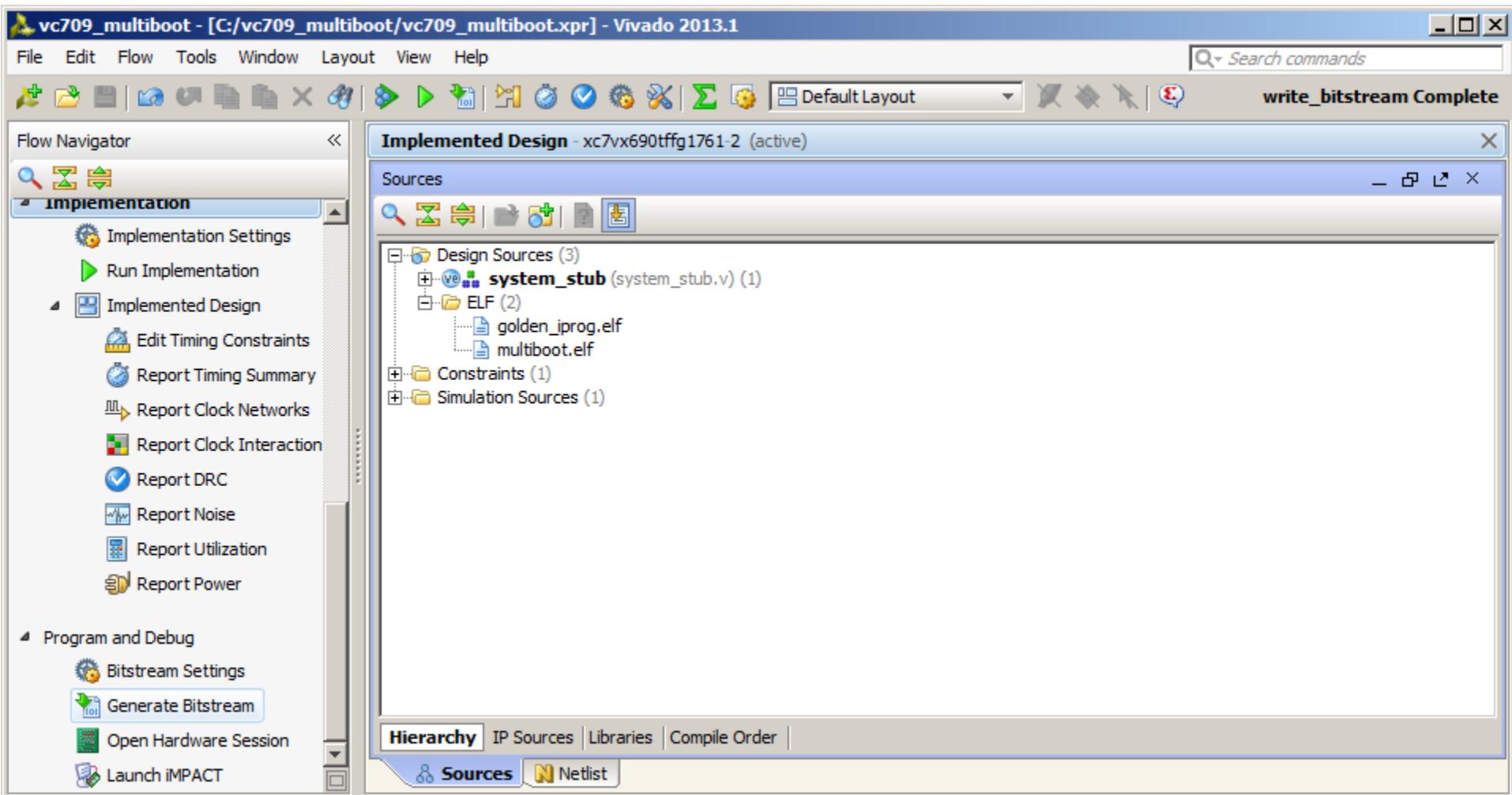
Compile VC709 MultiBoot Design

- Click the button to the right of the multiboot.elf; select the golden_iprog.elf then click OK twice



Compile VC709 MultiBoot Design

► Select Generate Bitstream



Generate a programming file after implementation.

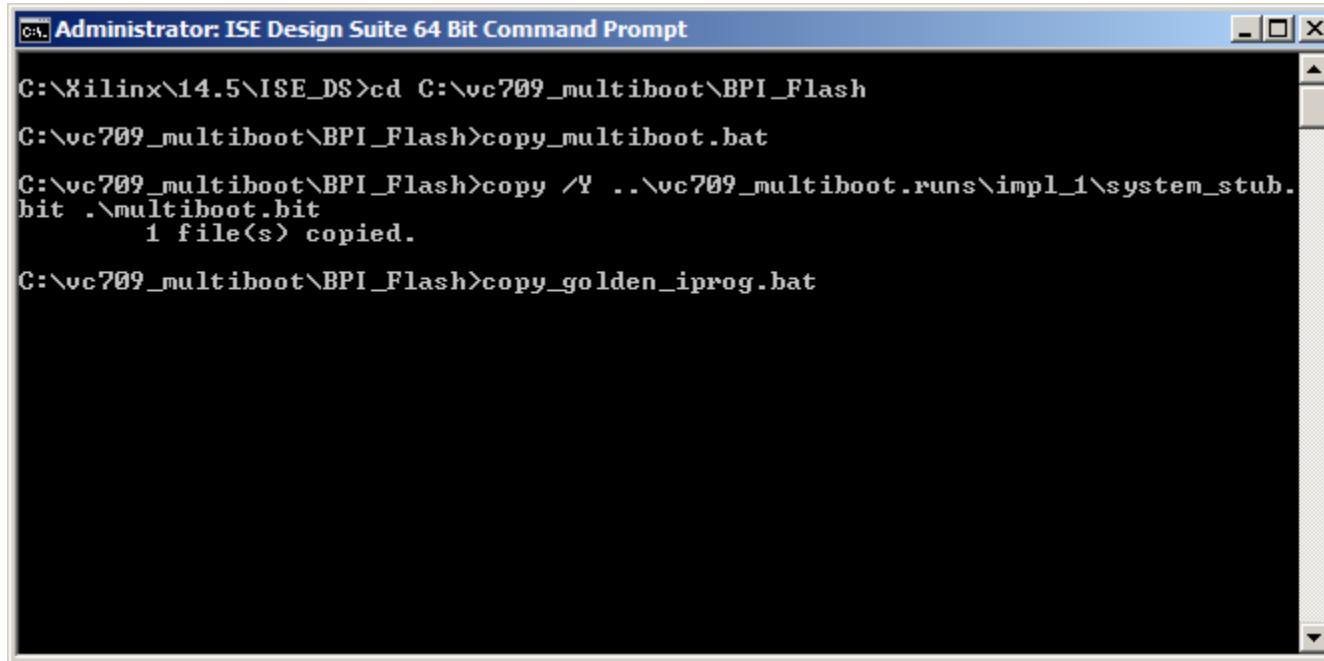
Note: Presentation applies to the VC709

 XILINX  ALL PROGRAMMABLE

Generate VC709 PROM File

► From the Command Prompt type:

copy_golden_iprog.bat

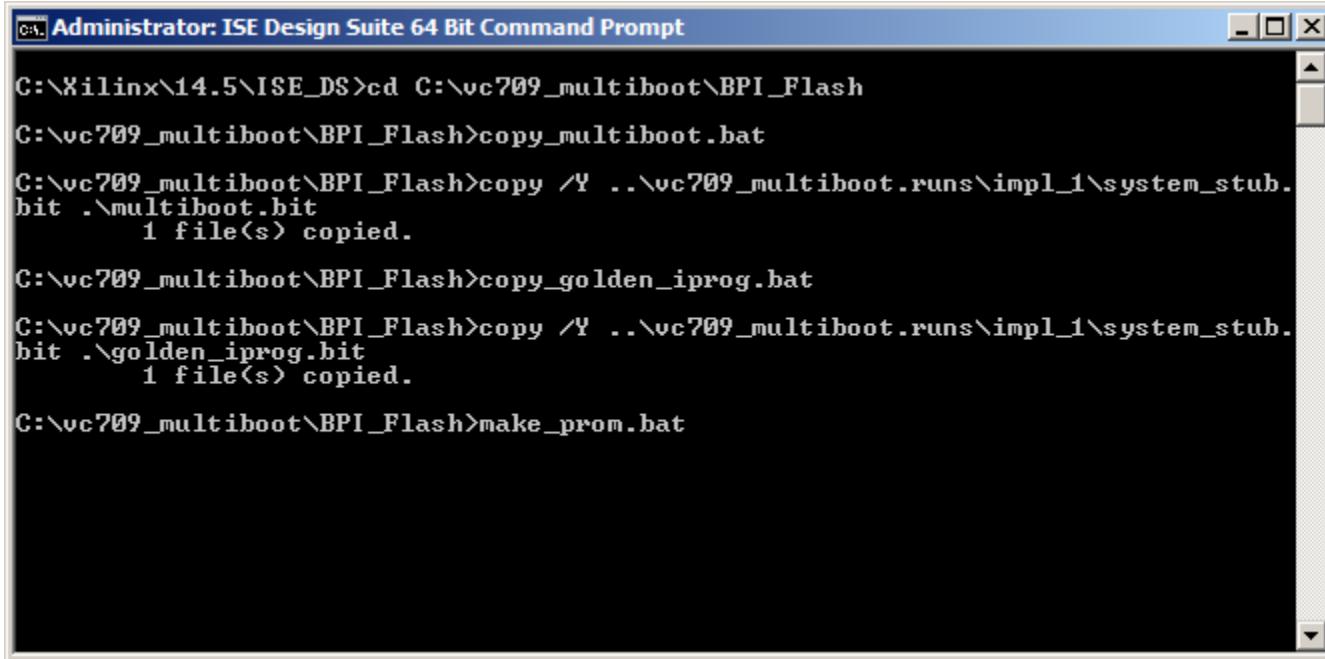


```
C:\Administrator: ISE Design Suite 64 Bit Command Prompt
C:\Xilinx\14.5\ISE_DS>cd C:\vc709_multiboot\BPI_Flash
C:\vc709_multiboot\BPI_Flash>copy_multiboot.bat
C:\vc709_multiboot\BPI_Flash>copy /Y ..\vc709_multiboot.runs\impl_1\system_stub.
bit .\multiboot.bit
      1 file(s) copied.
C:\vc709_multiboot\BPI_Flash>copy_golden_iprog.bat
```

Generate VC709 PROM File

► From the Command Prompt type:

make_prom.bat



```
C:\Administrator: ISE Design Suite 64 Bit Command Prompt
C:\Xilinx\14.5\ISE_DS>cd C:\vc709_multiboot\BPI_Flash
C:\vc709_multiboot\BPI_Flash>copy_multiboot.bat
C:\vc709_multiboot\BPI_Flash>copy /Y ..\vc709_multiboot.runs\impl_1\system_stub.
bit .\multiboot.bit
      1 file(s) copied.
C:\vc709_multiboot\BPI_Flash>copy_golden_iprog.bat
C:\vc709_multiboot\BPI_Flash>copy /Y ..\vc709_multiboot.runs\impl_1\system_stub.
bit .\golden_iprog.bit
      1 file(s) copied.
C:\vc709_multiboot\BPI_Flash>make_prom.bat
```

Virtex-7 MultiBoot Details

MultiBoot Register Setup & Command

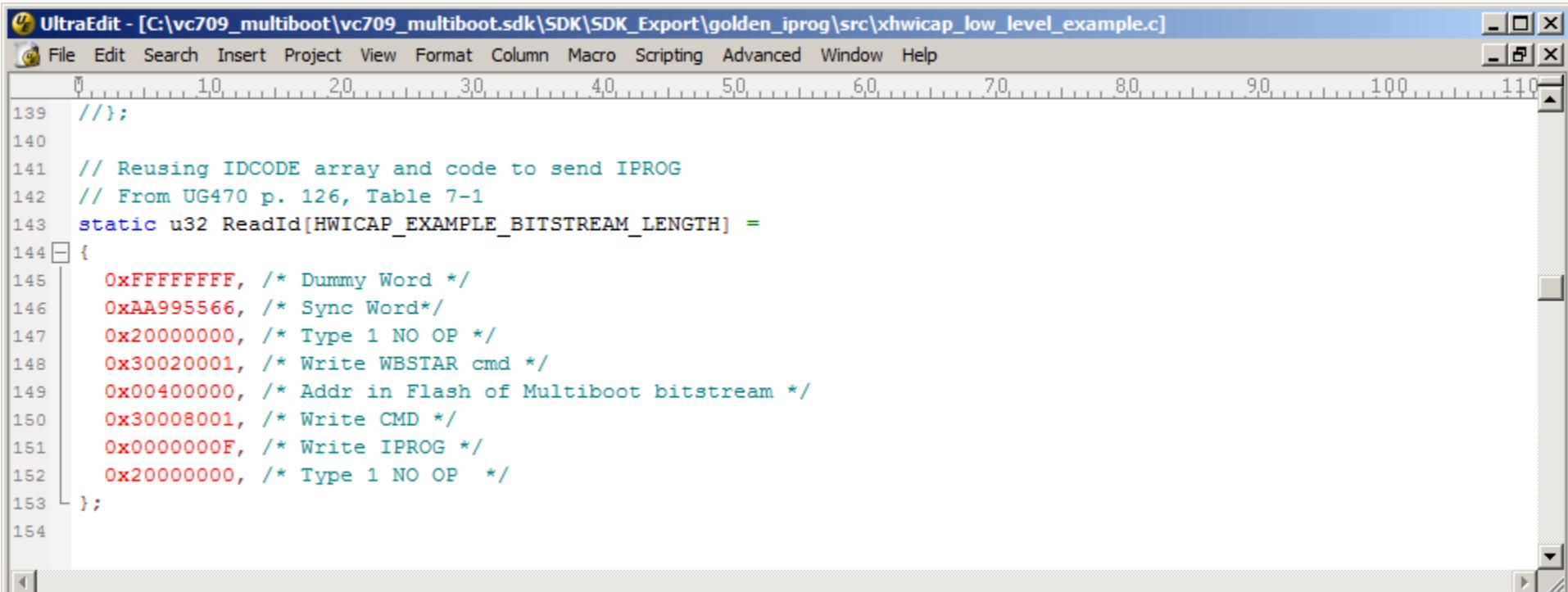
- Below are the values programmed into the ICAP via IPROG
 - This table is from UG470, 7 Series FPGAs Configuration User Guide

Table 7-1: Example Bitstream for IPROG through ICAP

Configuration Data (hex)	Explanation
FFFFFFFFFF	Dummy Word
AA995566	Sync Word
20000000	Type 1 NO OP
30020001	Type 1 Write 1 Words to WBSTAR
00000000	Warm Boot Start Address (Load the Desired Address)
30008001	Type 1 Write 1 Words to CMD
0000000F	IPROG Command
20000000	Type 1 NO OP

MultiBoot Register Setup & Command

- IPROG command issued via the software in xhwicap_low_level_example.c



The screenshot shows a code editor window titled "UltraEdit - [C:\vc709_multiboot\vc709_multiboot.sdk\SDK\SDK_Export\golden_iprog\src\xhwicap_low_level_example.c]". The menu bar includes File, Edit, Search, Insert, Project, View, Format, Column, Macro, Scripting, Advanced, Window, and Help. The status bar at the bottom shows line numbers from 139 to 154. The code itself is a C array definition:

```
139 //};  
140  
141 // Reusing IDCODE array and code to send IPROG  
142 // From UG470 p. 126, Table 7-1  
143 static u32 ReadId[HWICAP_EXAMPLE_BITSTREAM_LENGTH] =  
144 {  
145     0xFFFFFFFF, /* Dummy Word */  
146     0xAA995566, /* Sync Word*/  
147     0x20000000, /* Type 1 NO OP */  
148     0x30020001, /* Write WBSTAR cmd */  
149     0x00400000, /* Addr in Flash of Multiboot bitstream */  
150     0x30008001, /* Write CMD */  
151     0x0000000F, /* Write IPROG */  
152     0x20000000, /* Type 1 NO OP */  
153};  
154
```

References

References

► 7 Series Configuration

- 7 Series FPGAs Configuration User Guide
 - http://www.xilinx.com/support/documentation/user_guides/ug470_7Series_Config.pdf

Documentation

Documentation

➤ Virtex-7

- Virtex-7 FPGA Family
 - <http://www.xilinx.com/products/silicon-devices/fpga/virtex-7/index.htm>

➤ VC709 Documentation

- Virtex-7 FPGA VC709 Evaluation Kit
 - <http://www.xilinx.com/products/boards-and-kits/EK-V7-VC709-G.htm>
- VC709 User Guide
 - http://www.xilinx.com/support/documentation/boards_and_kits/ug887-vc709-eval-board-v7-fpga.pdf