AC701 MultiBoot Design

April 2015

XTP226

Revision History

Date	Version	Description
04/30/14	11.0	Recompiled for 2015.1.
11/24/14	10.0	Recompiled for 2014.4.
10/08/14	9.0	Recompiled for 2014.3.
06/09/14	8.0	Recompiled for 2014.2.
04/16/14	4.0	Recompiled for 2014.1.
12/18/13	3.0	Recompiled for 2013.4.
10/23/13	2.0	Recompiled for 2013.3. Added AR58291.
06/19/13	1.0	Initial version.

Overview

- Artix-7 MultiBoot Capability
- Overview of 7 Series Fallback MultiBoot
- Xilinx AC701 Board
- Software Requirements
- AC701 Setup
- Compiling the MultiBoot Design
- Run MultiBoot Design
- Artix-7 MultiBoot Details
- References

Note: This presentation applies to the AC701

Artix-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)

Note: Presentation applies to the AC701

Artix-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**
  - AC701 MultiBoot Design Files (2015.1 C) ZIP file
  - Available through http://www.xilinx.com/ac701

Note: Presentation applies to the AC701

AC701 MultiBoot Design Description

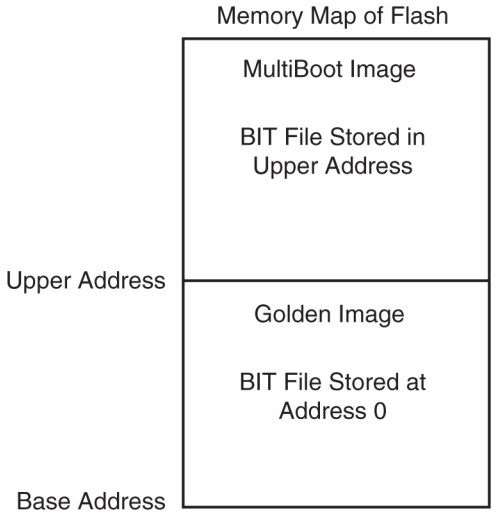
Description

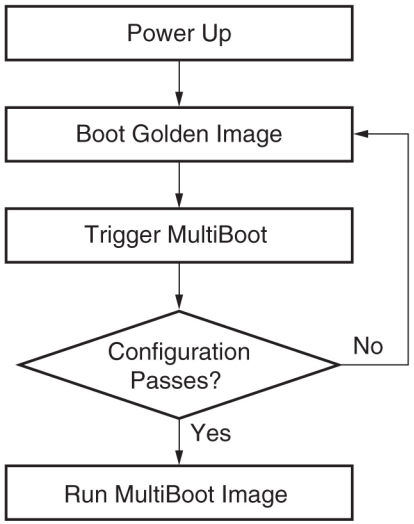
- Design consists of three bitstreams, golden_iprog_spi.bit, multiboot.bit, and corrupted.bit loaded in the PROM at 0x0, 0x400000, and 0x800000
- The golden_iprog_spi.bit has software to allow a reboot to the second or third bitstreams with certain switch settings
- The multiboot.bit is compiled with 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





Note: Presentation applies to the AC701

Upgrade Example

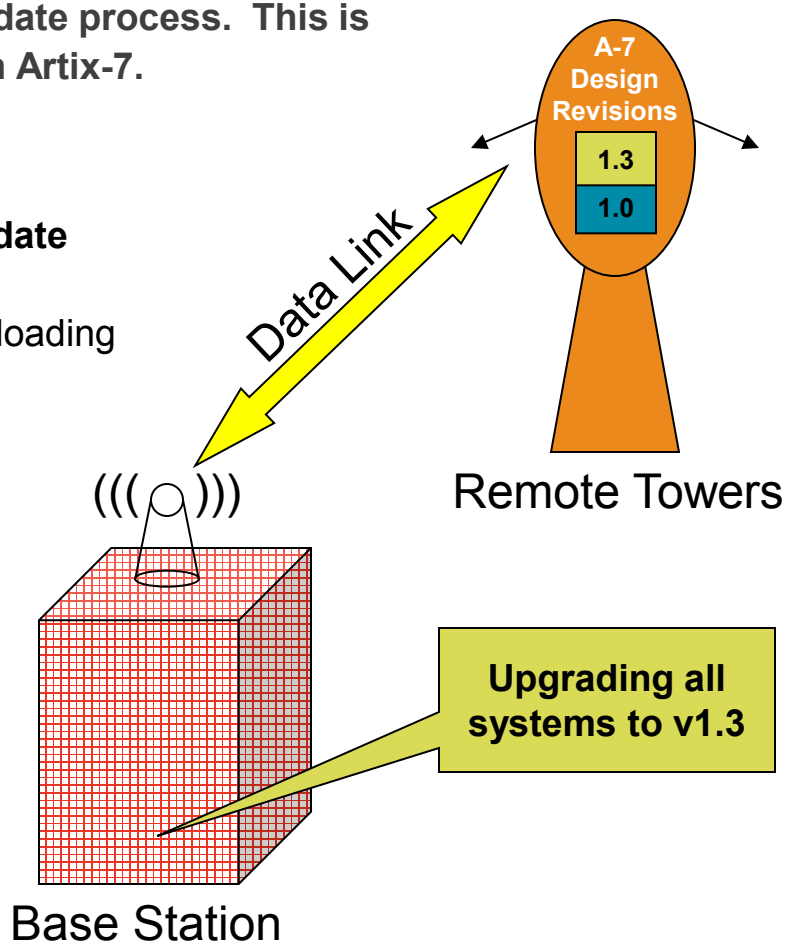
- **MultiBoot:** **Process by which the FPGA selectively reprograms and reloads its bitstream from an attached external memory.**
- **Safe update:** **Field updating bitstream storage with a new bitstream in such a manner to prevent any failure due to a failure in the update process. This is accomplished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

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

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

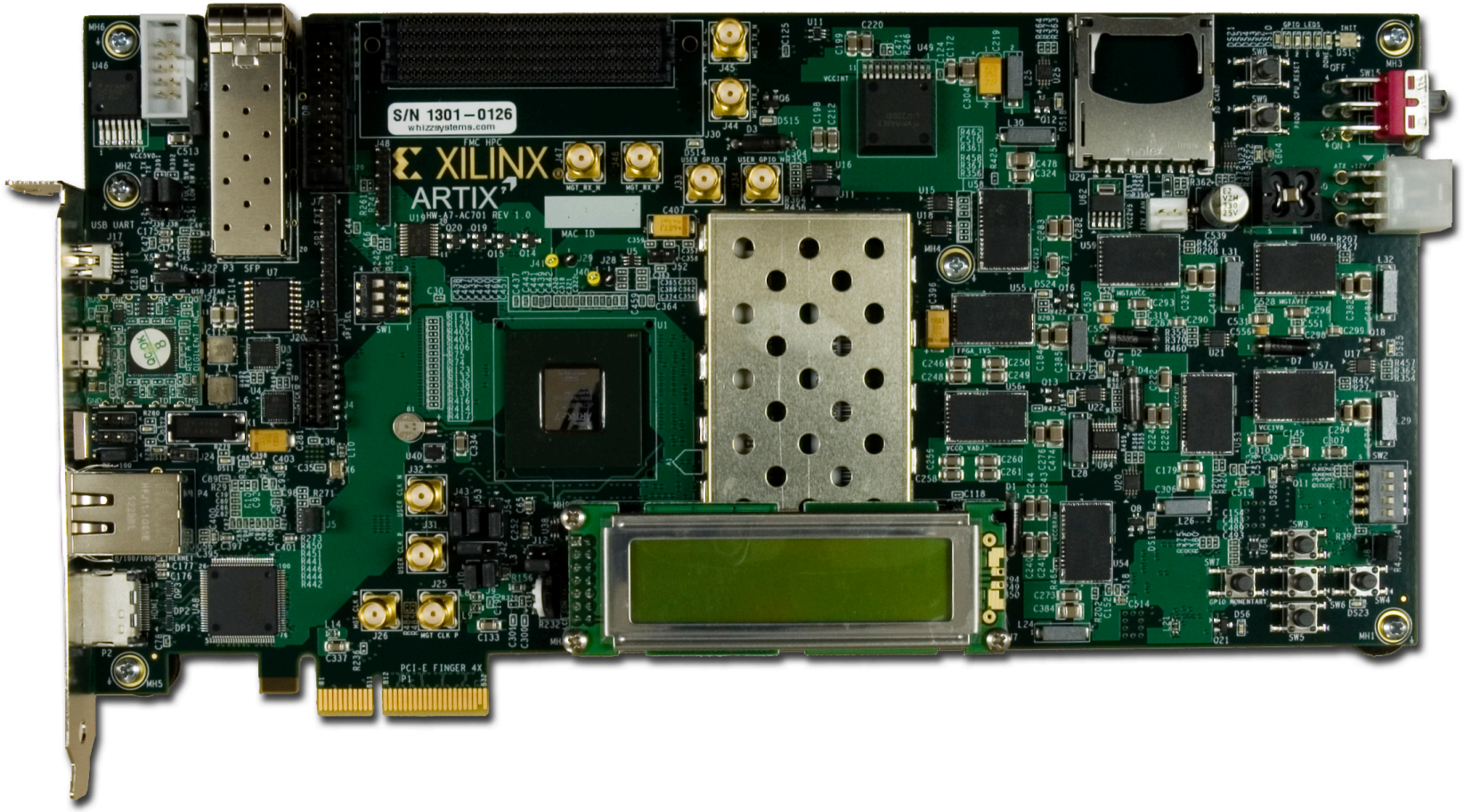
How is a system upgraded?

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



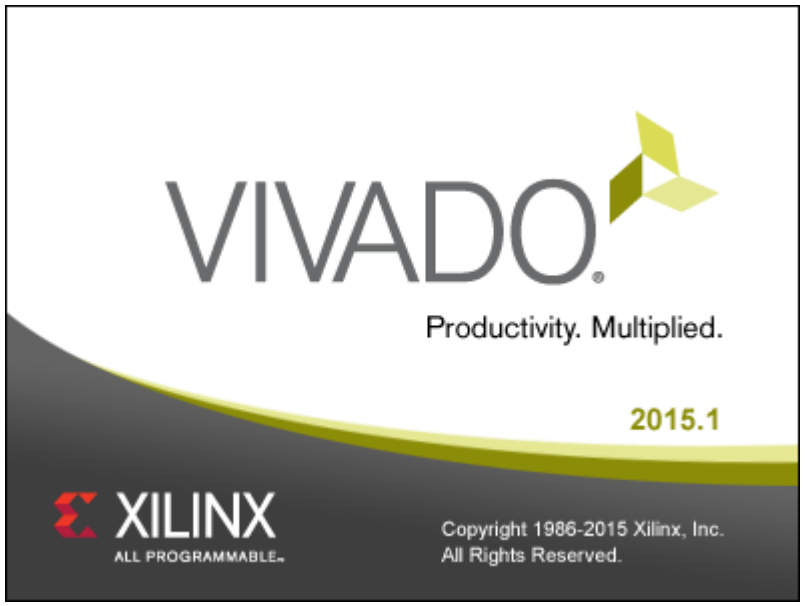
Note: Presentation applies to the AC701

Xilinx AC701 Board



Vivado Software Requirements

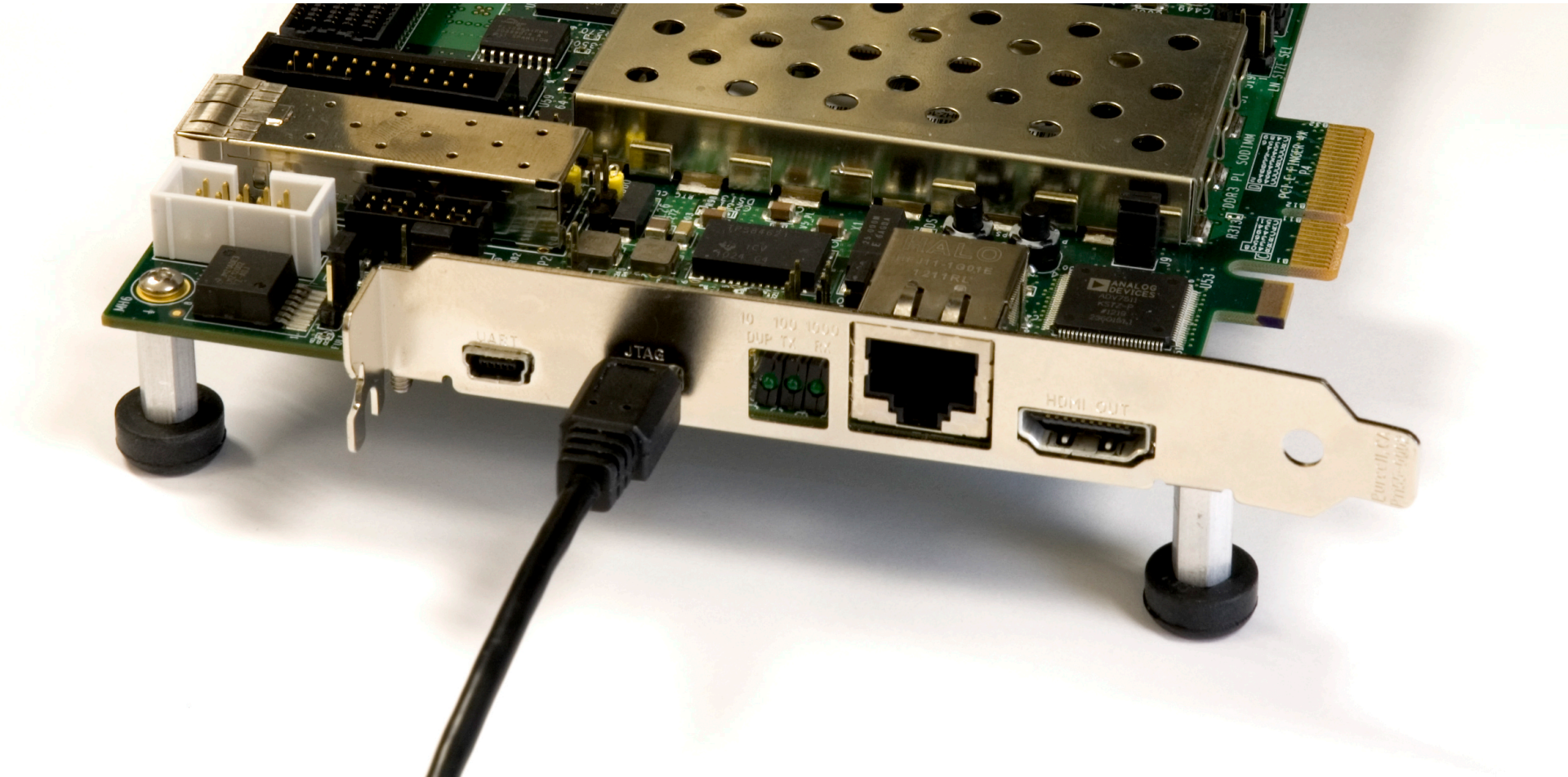
- Xilinx Vivado Design Suite 2015.1, Design Edition + SDK



Note: Presentation applies to the AC701

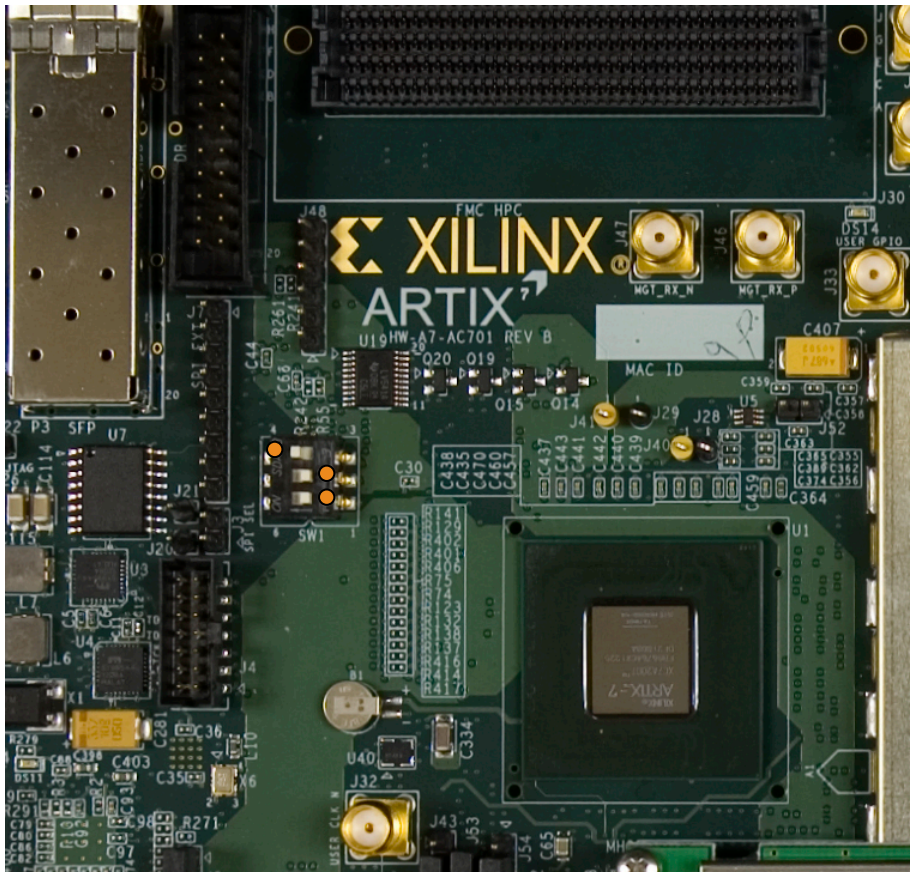
Hardware Setup

- **Connect a USB Type-A to Micro-B cable to the USB JTAG (Digilent) connector on the AC701 board**
  - Connect this cable to your PC
  - Power on the AC701 board



Hardware Setup

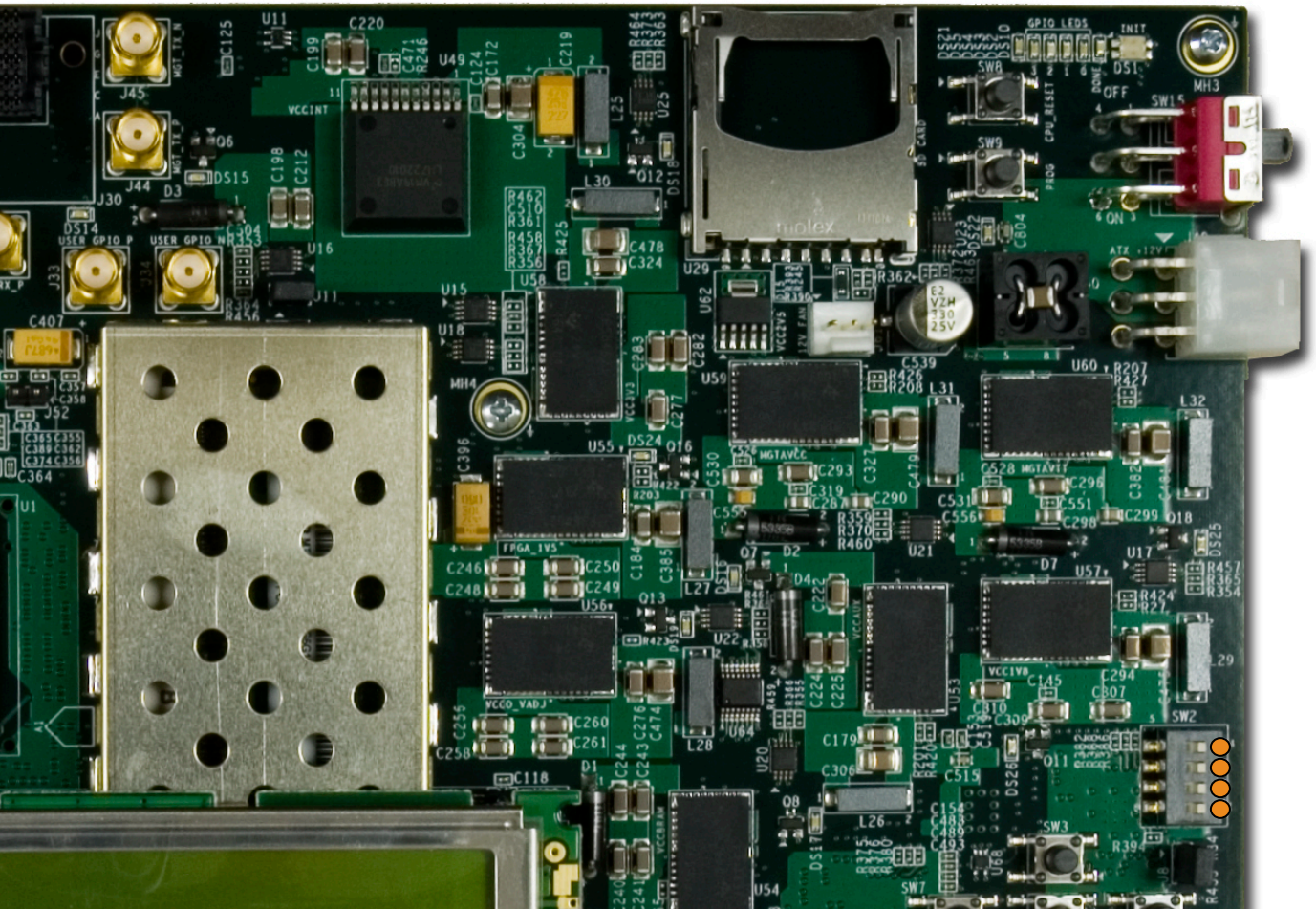
- **Set S1 to 001 (1 = on, Position 1 → Position 3)**
  - This enables Master SPI configuration from the N25Q SPI Flash
  - FPGA mode pins M[2:0] = 001



Note: Presentation applies to the AC701

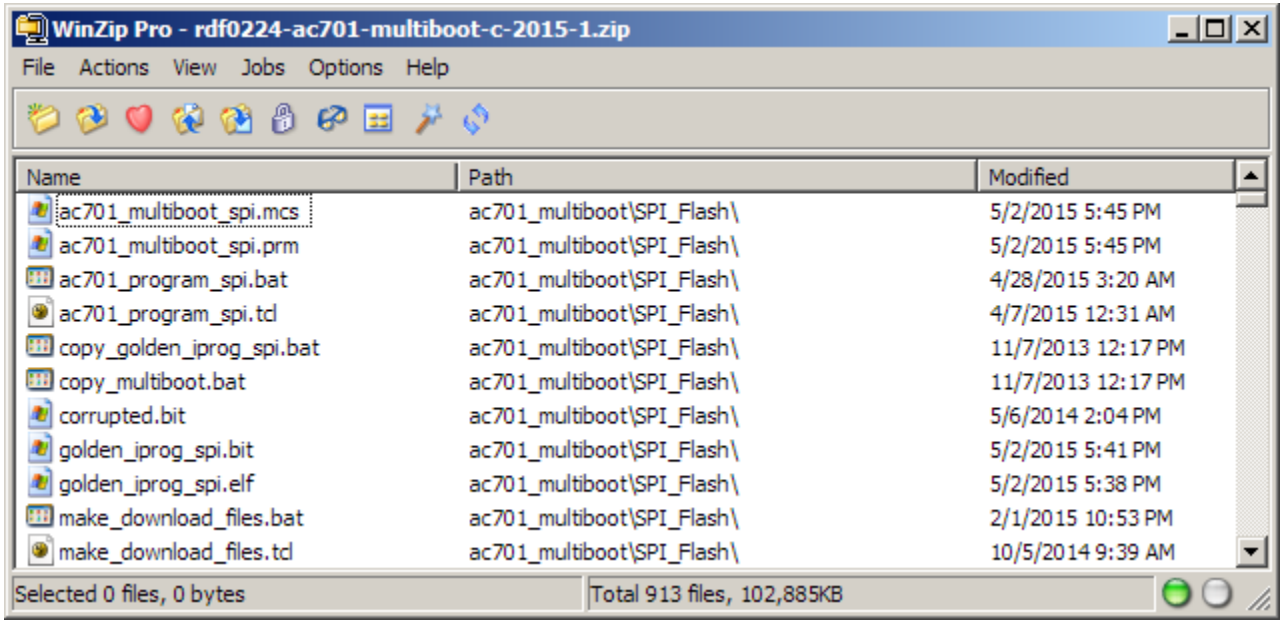
Hardware Setup

- **Set S11 to 0000 (1 = on, Position 1 → Position 4)**
  - This sets the MultiBoot functions to off



Program SPI MultiBoot Design

- **Unzip the AC701 MultiBoot Design Files (2015.1 C) ZIP file**
  - Available through http://www.xilinx.com/ac701

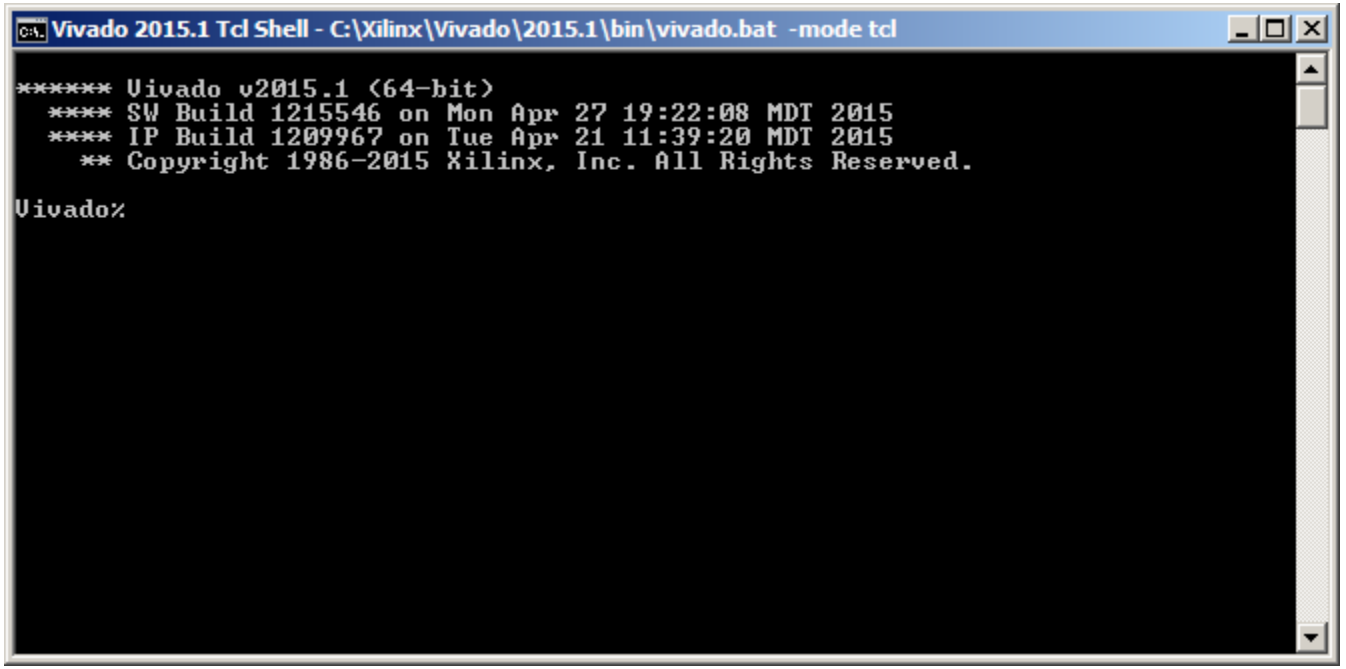


Note: Presentation applies to the AC701

Program SPI MultiBoot Design

- Open a Vivado Tcl Shell:

  Start → All Programs → Xilinx Design Tools → Vivado 2015.1 → Vivado 2015.1 Tcl Shell



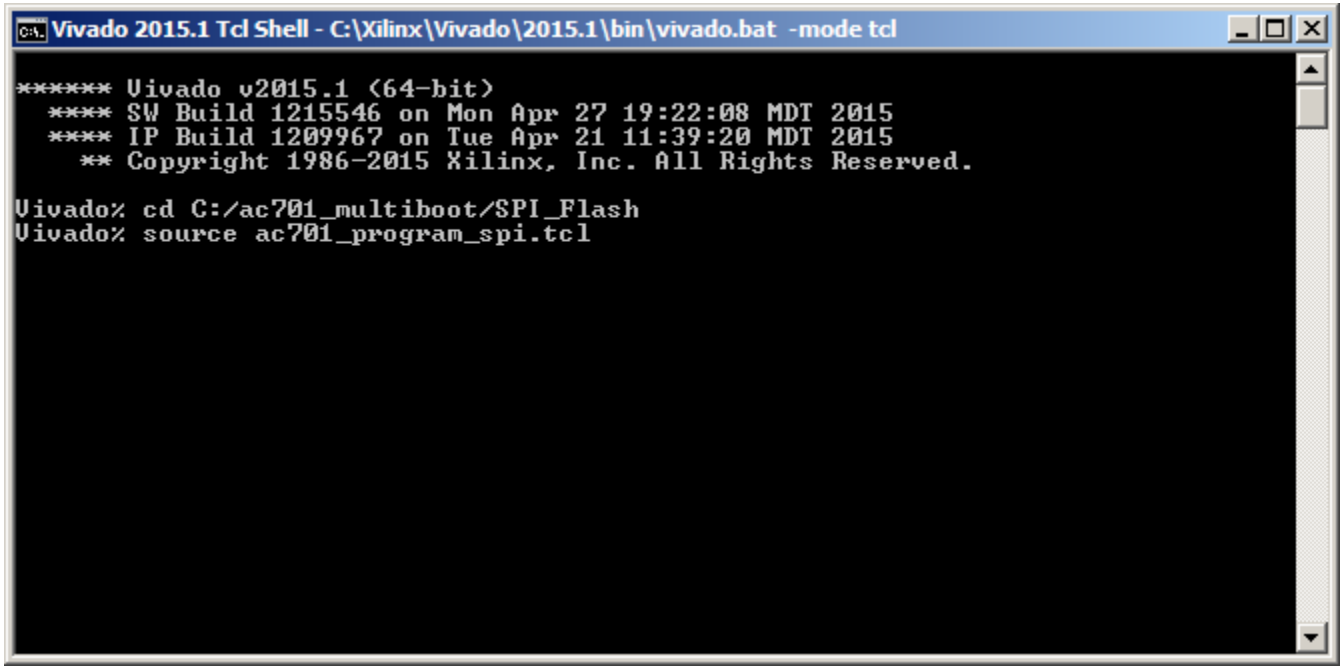
Note: Presentation applies to the AC701

Program SPI MultiBoot Design

- In the Vivado Tcl Shell type:

  cd C:/ac701_multiboot/SPI_Flash

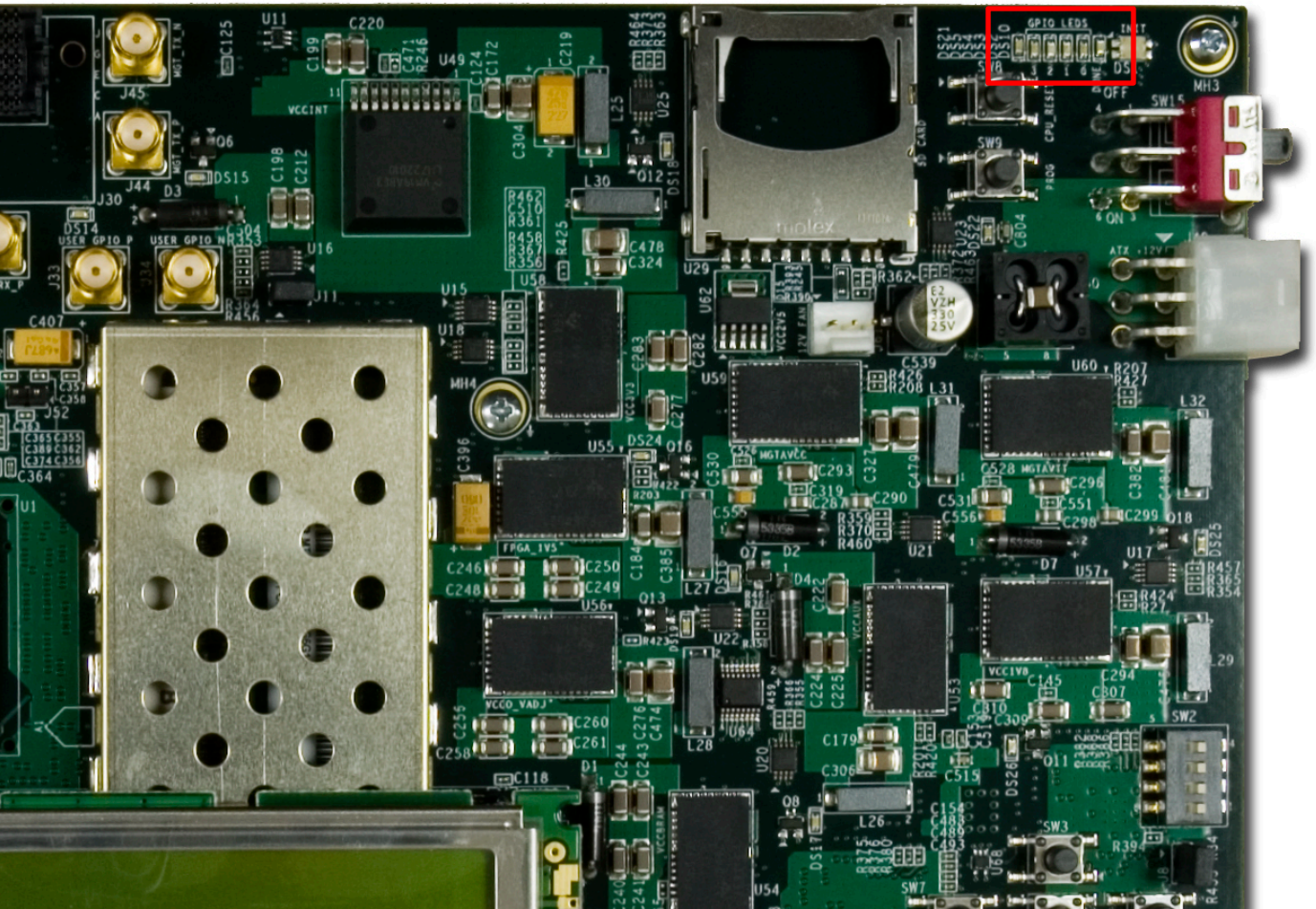
  source ac701_program_spi.tcl



Note: Takes about 10 minutes

Run MultiBoot Design

- Cycle board power
- The initial design, “golden_iprog_spi.bit”, shows a cycling LED pattern on the GPIO LEDs

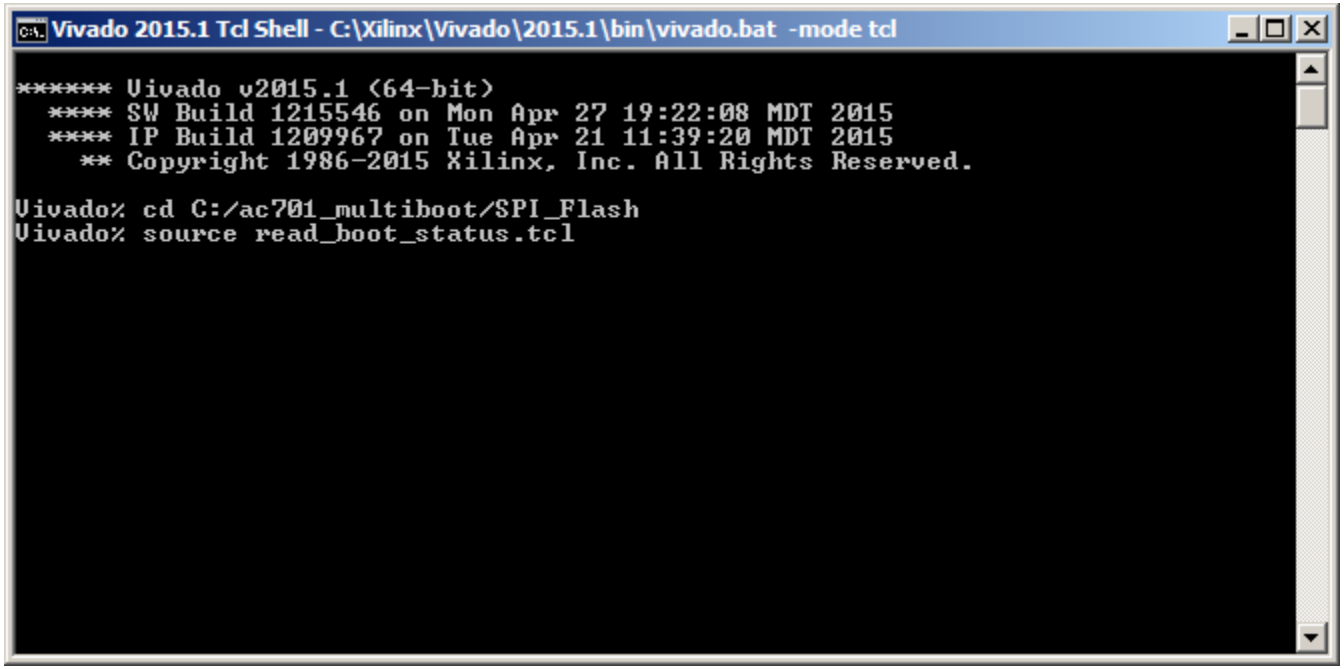


Run MultiBoot Design

- In a Vivado Tcl Shell type:

  cd C:/ac701_multiboot/SPI_Flash

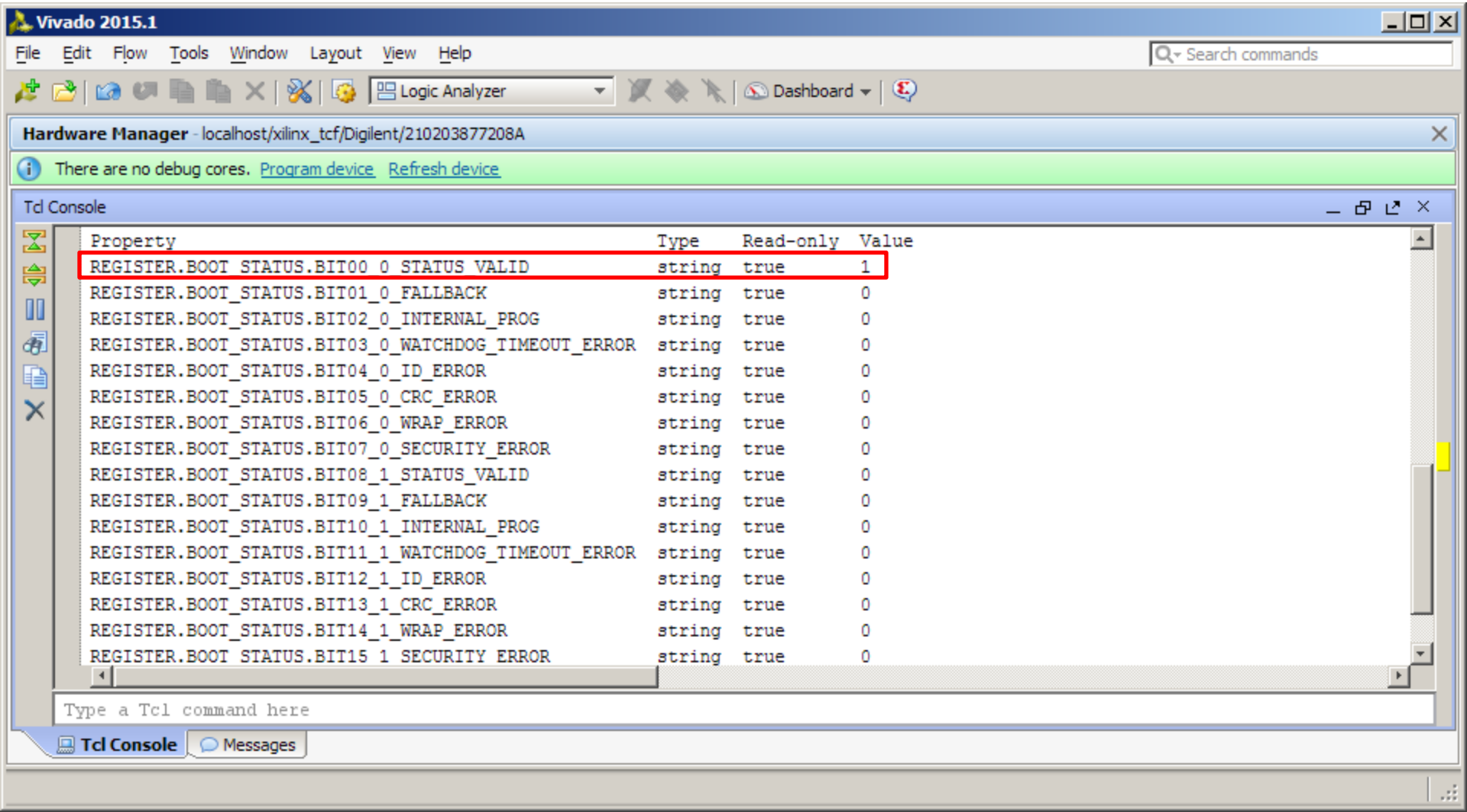
  source read_boot_status.tcl



Note: Presentation applies to the AC701

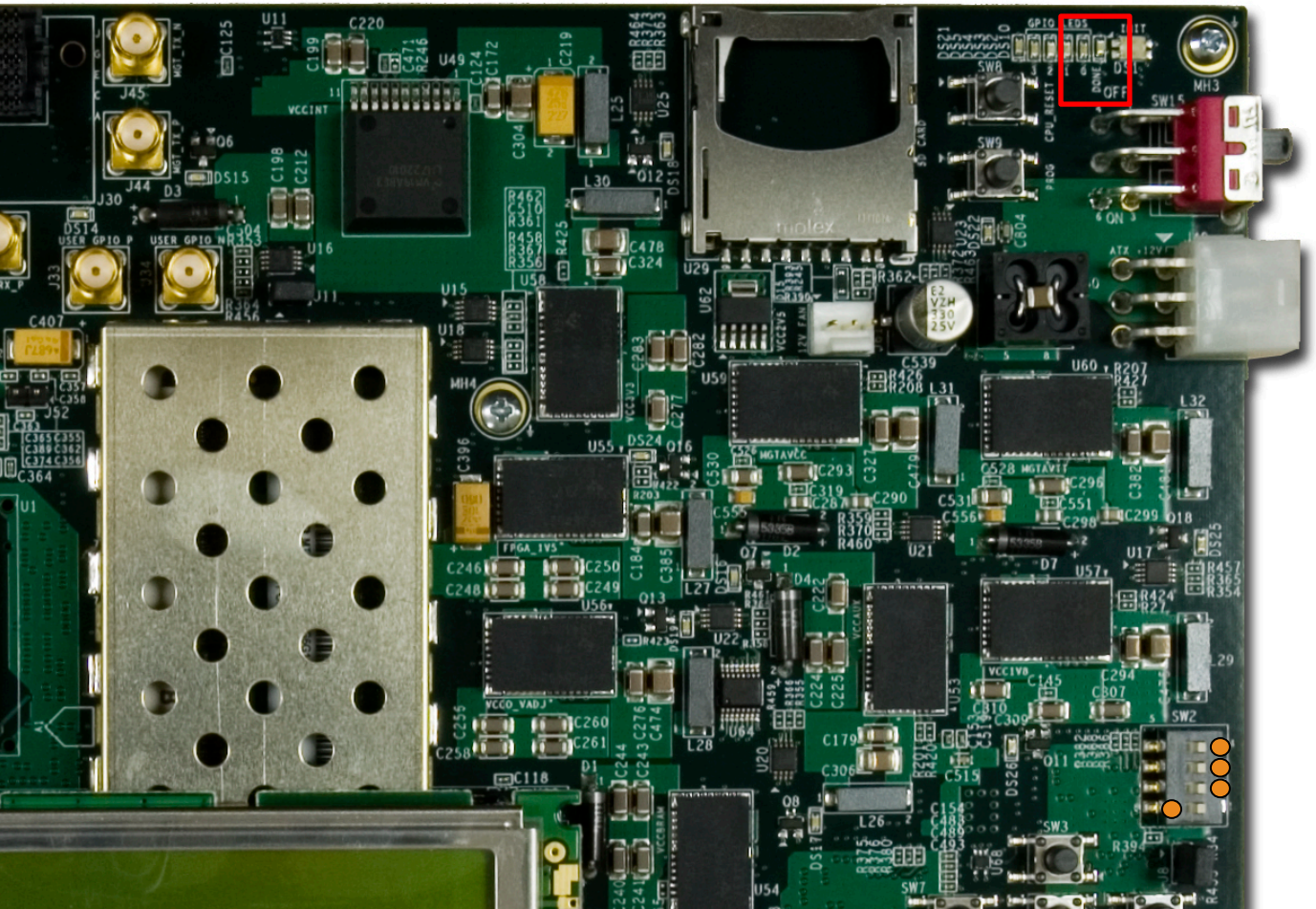
Run MultiBoot Design

- View the “REGISTER.BOOT_STATUS.BIT00_0_STATUS_VALID”
- Since this was a regular boot, only “0_STATUS_VALID” is set



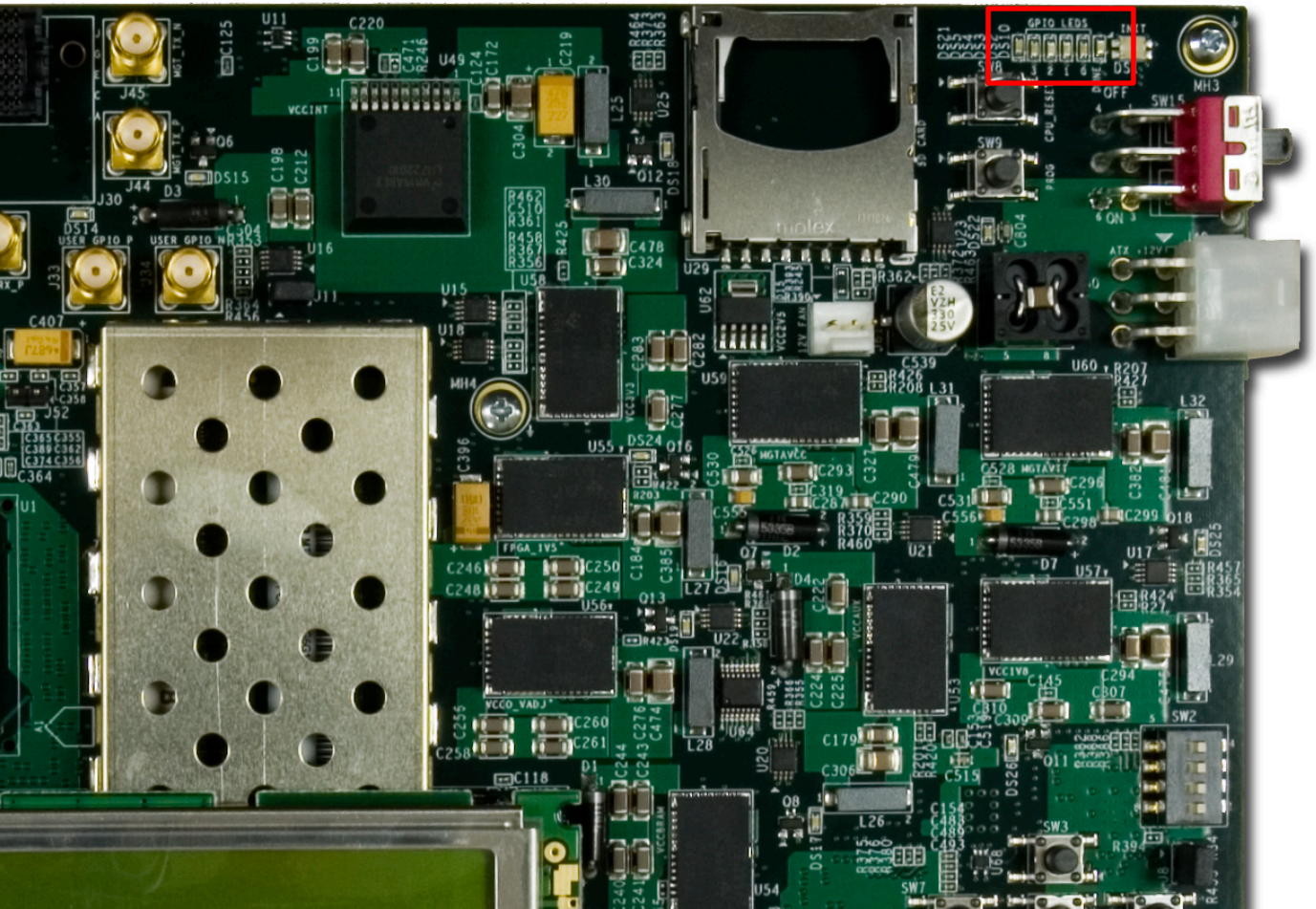
Run MultiBoot Design

- **Set S2 to 1000 (1 = on, Position 1 → Position 4)**
  - Issues an IPROG command to re-configure from a good MultiBoot bitstream
  - Set it back to 0000 when the DONE LED goes out



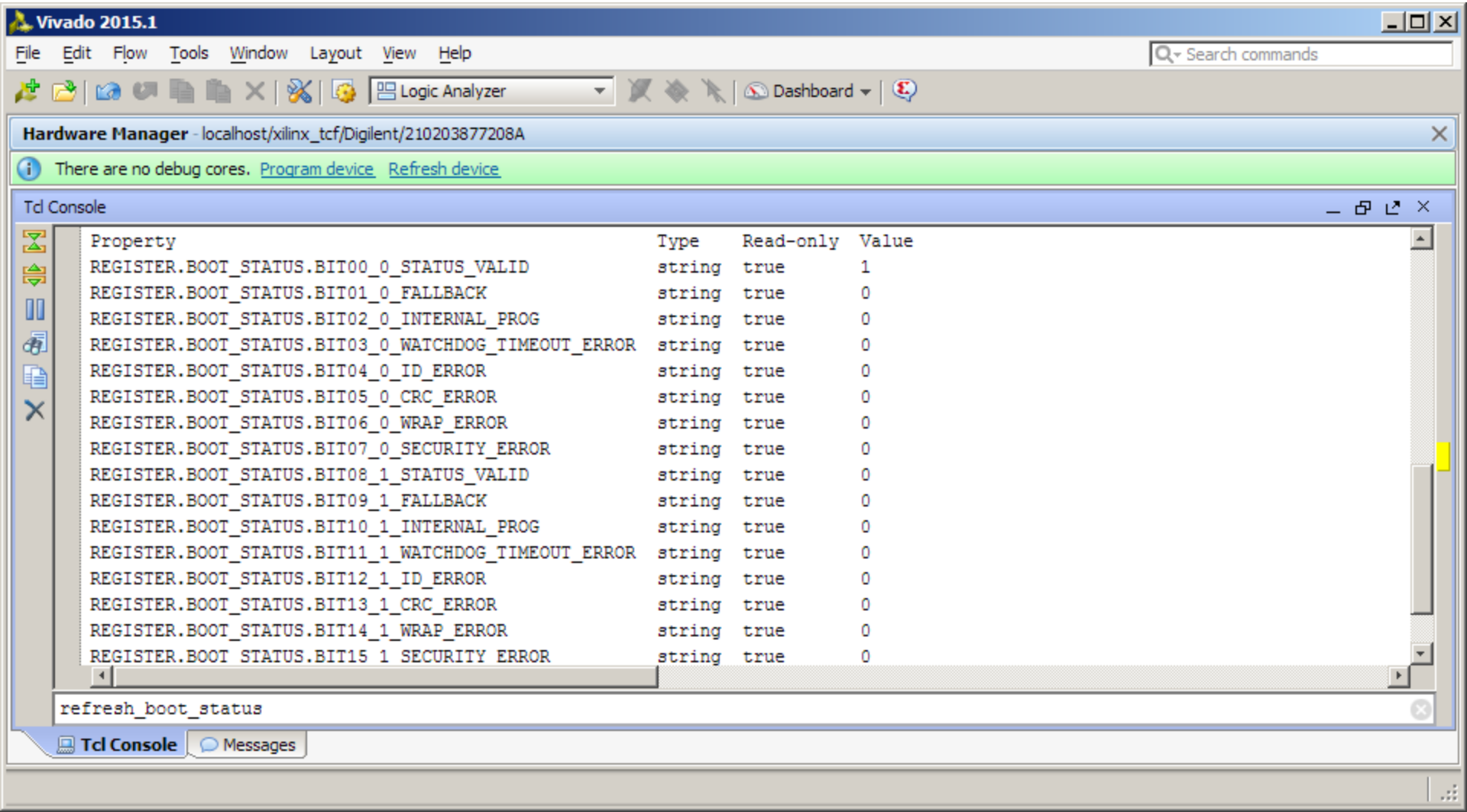
Run MultiBoot Design

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



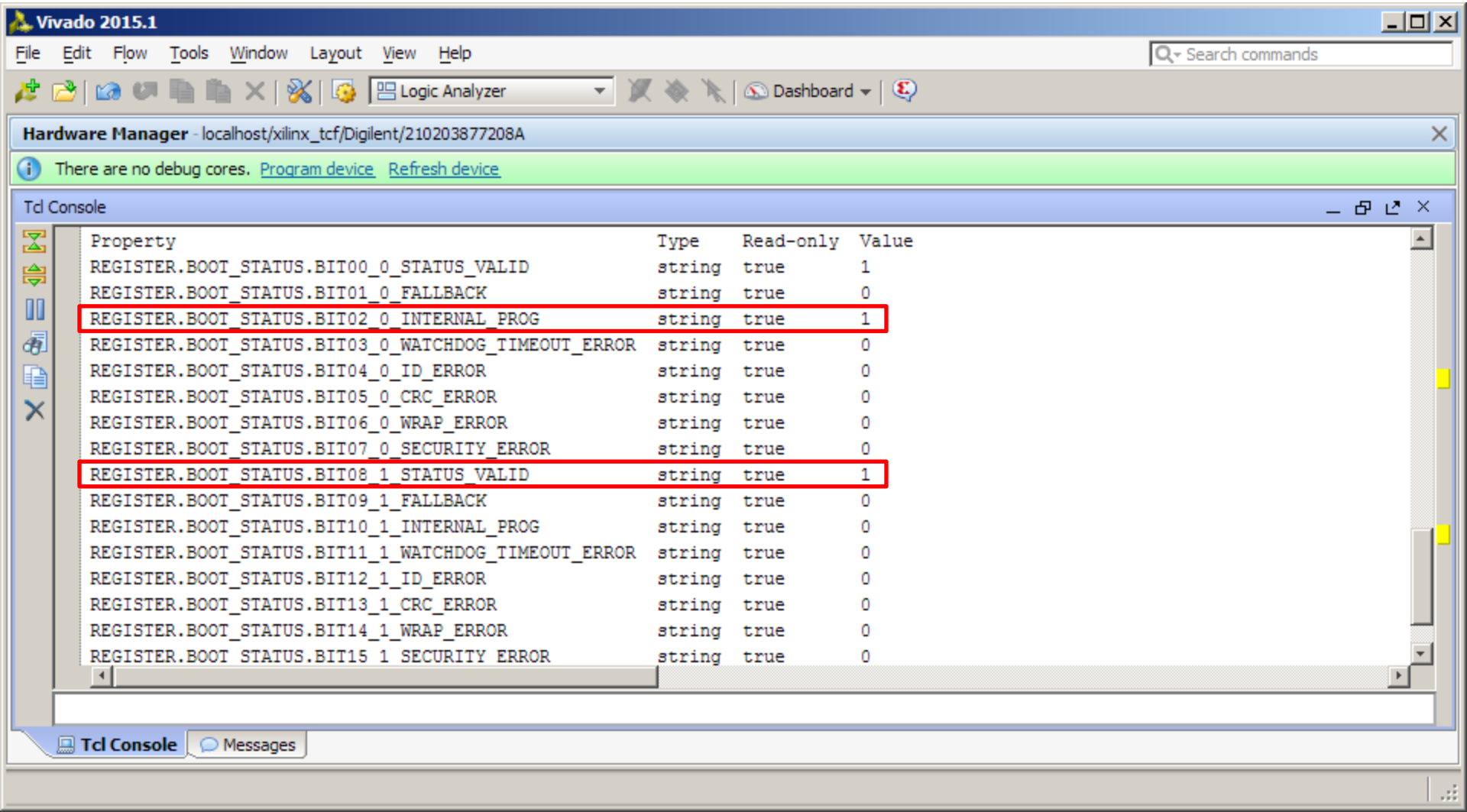
Run MultiBoot Design

- The script adds a procedure “refresh_boot_status”
- Type refresh_boot_status and press Enter



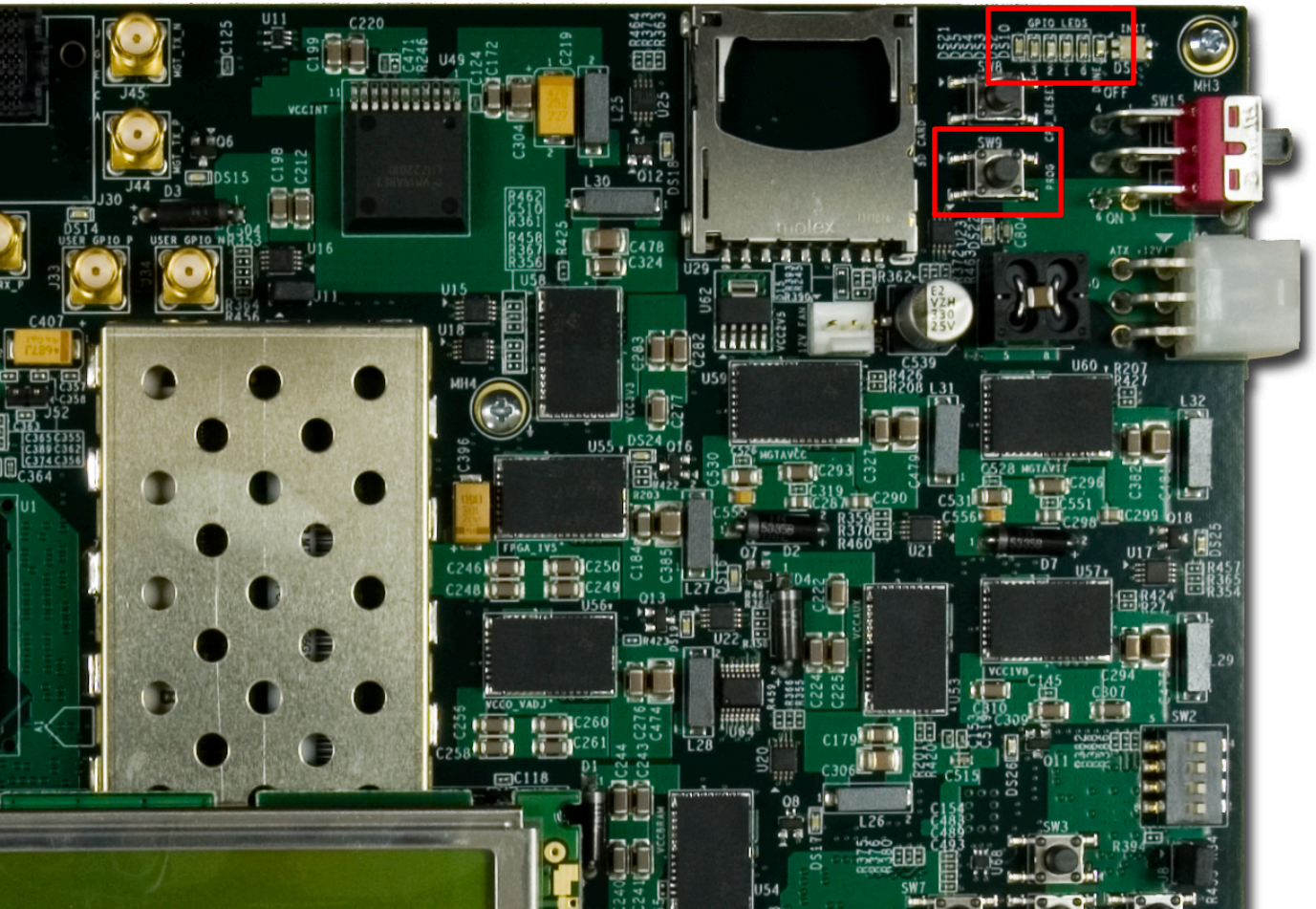
Run MultiBoot Design

- The following status register messages are now set:
  - INTERNAL_PROG and 1_STATUS_VALID



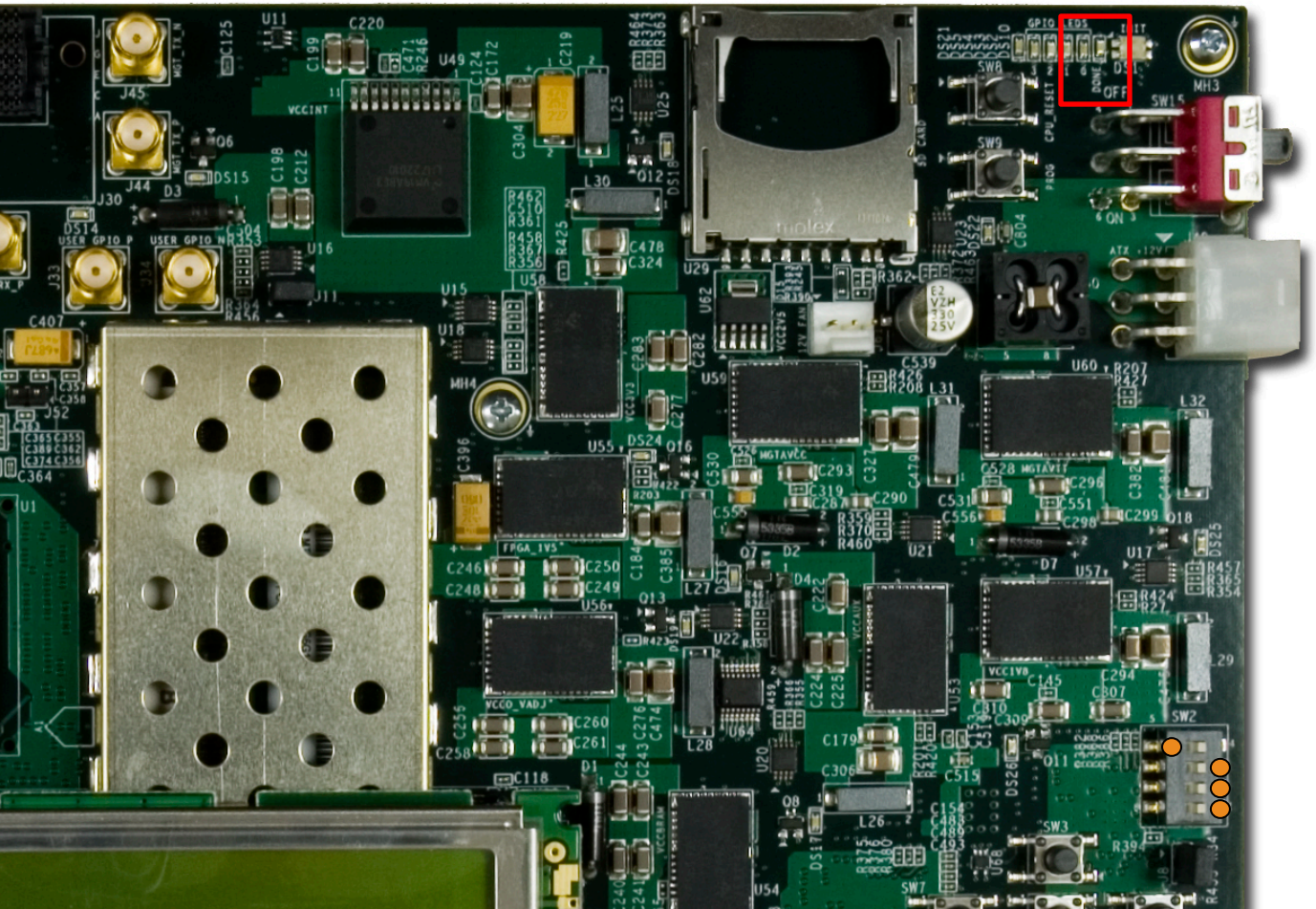
Run MultiBoot Design

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



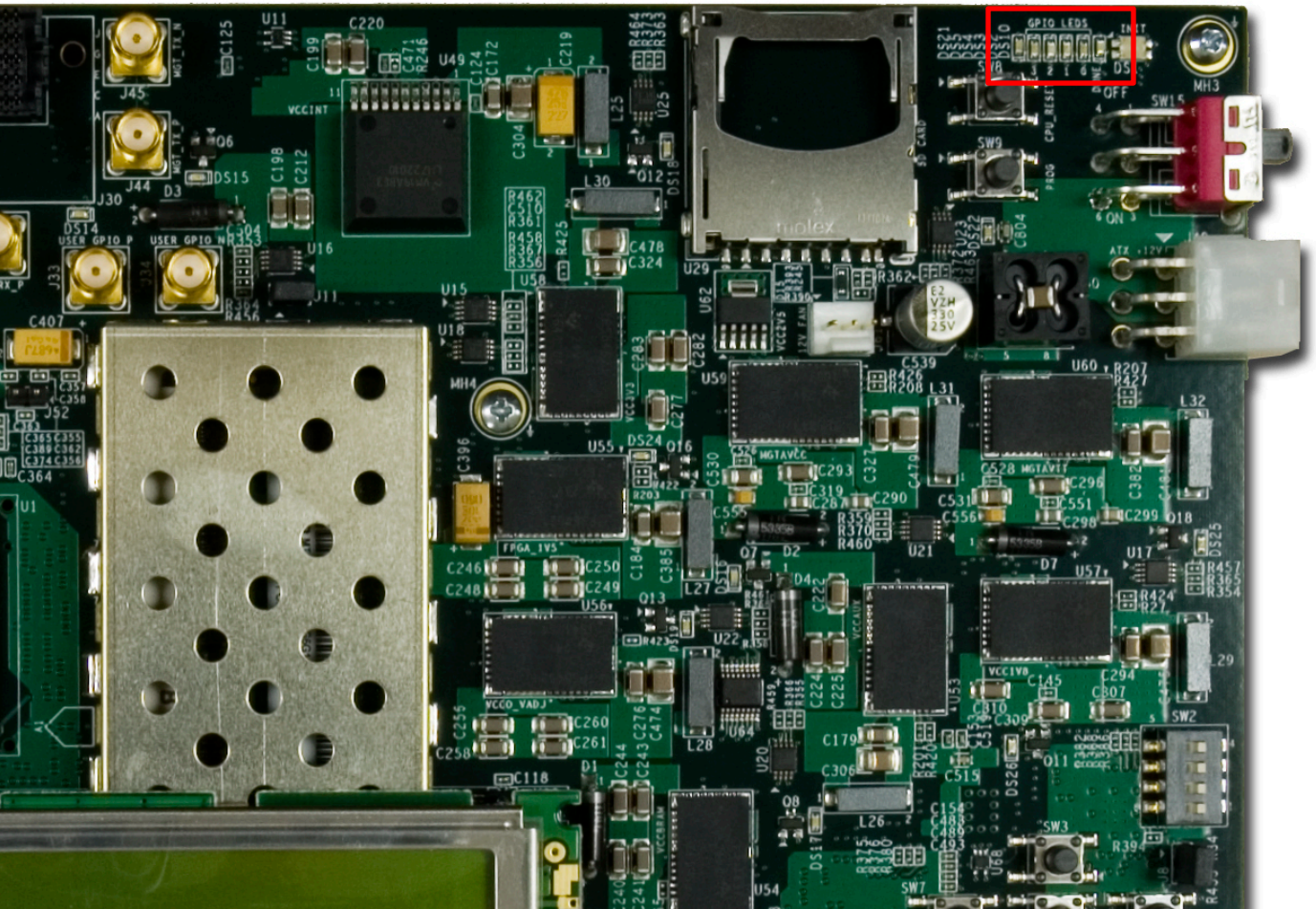
Run MultiBoot Design

- **Set S2 to 0001 (1 = on, Position 1 → Position 4)**
  - Issues an IPROG command to re-configure from a corrupted MultiBoot bitstream
  - Set it back to 0000 when the DONE LED goes out



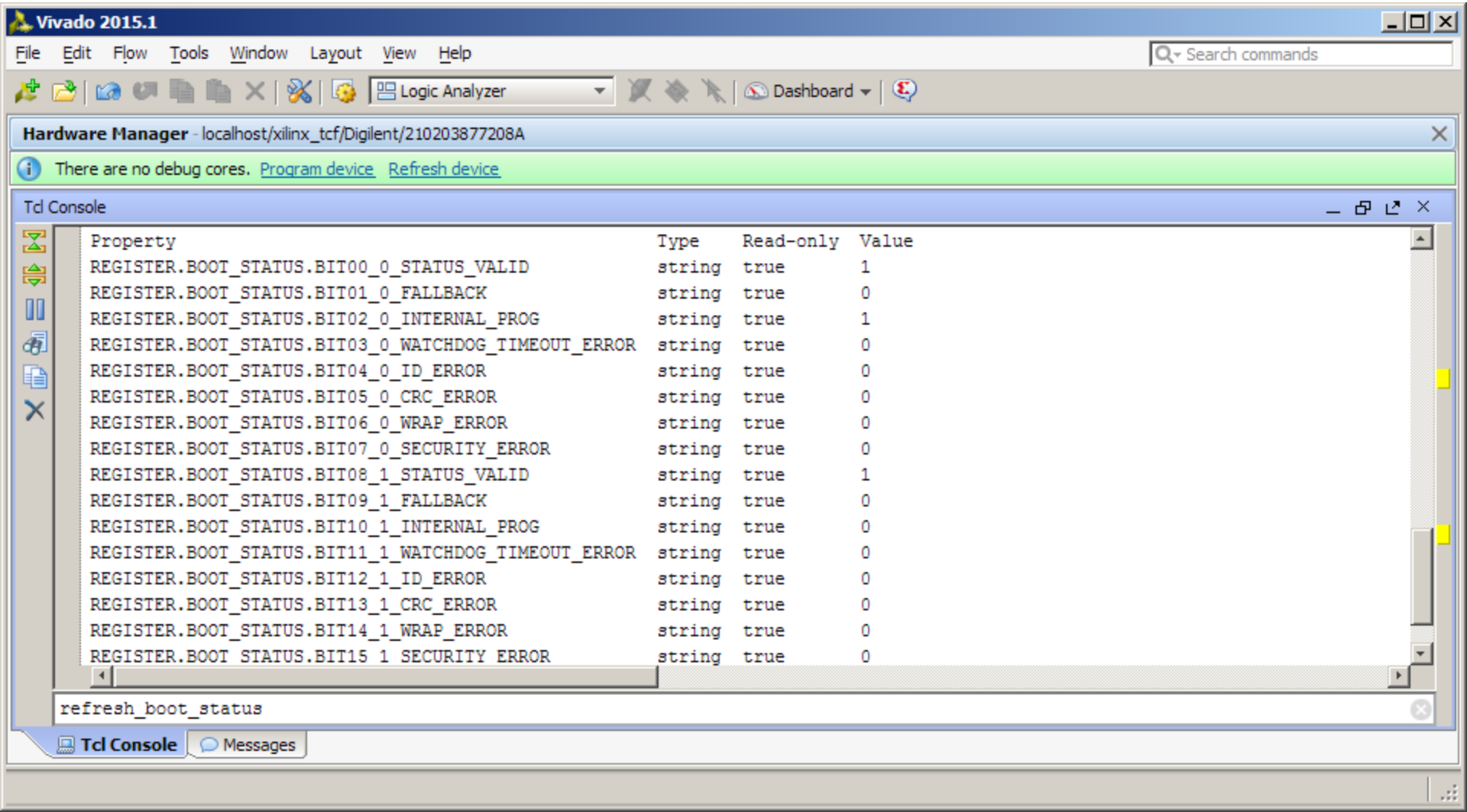
Run MultiBoot Design

- 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
- This happens very quickly; the DONE LED may appear to be blinking



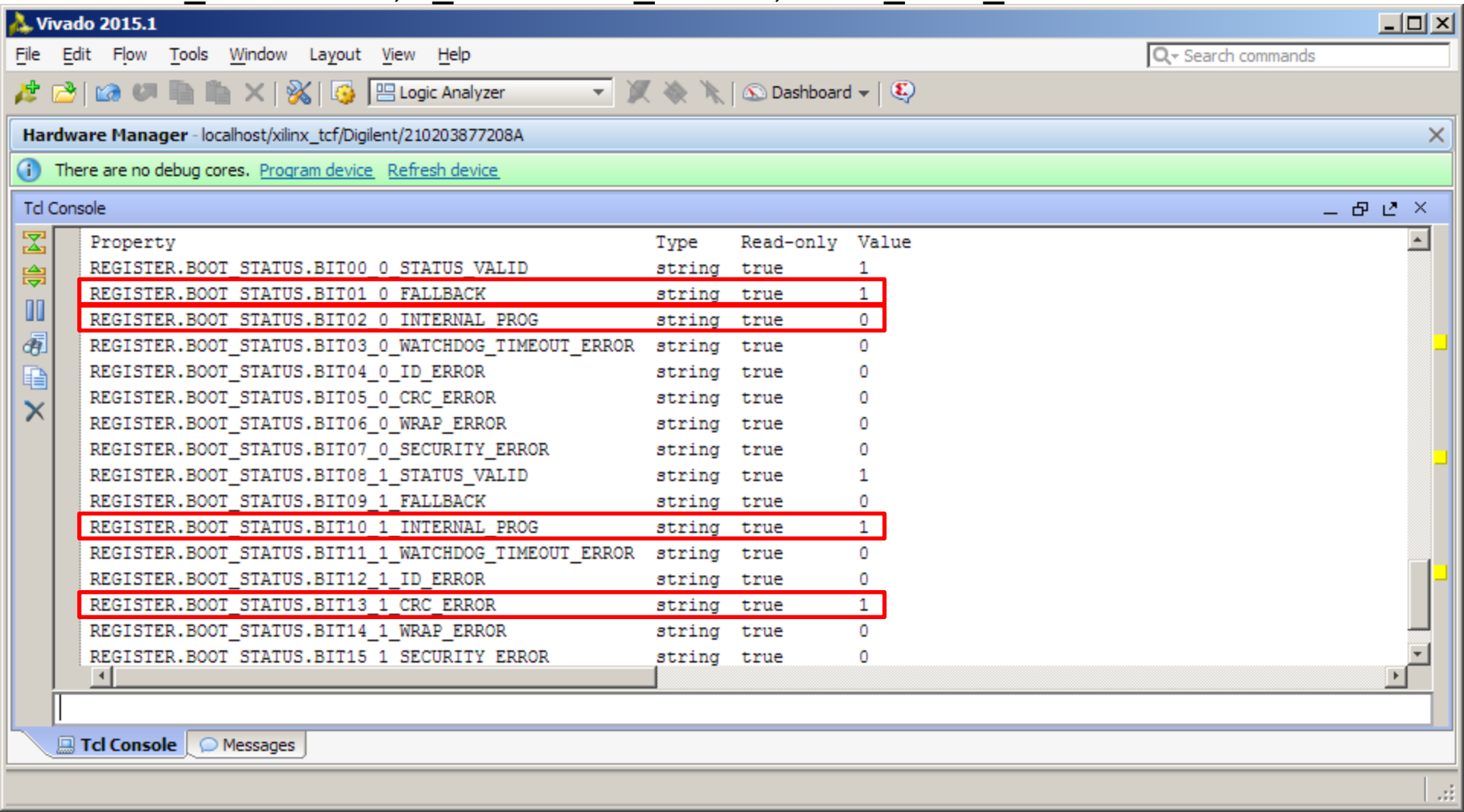
Run MultiBoot Design

- Type refresh_boot_status and press Enter



Run MultiBoot Design

- **The following status register messages are now seen:**
  - IPROG_0 is no longer active
  - 0_FALLBACK, 1_INTERNAL_IPROG, and 1_CRC_ERROR are now active



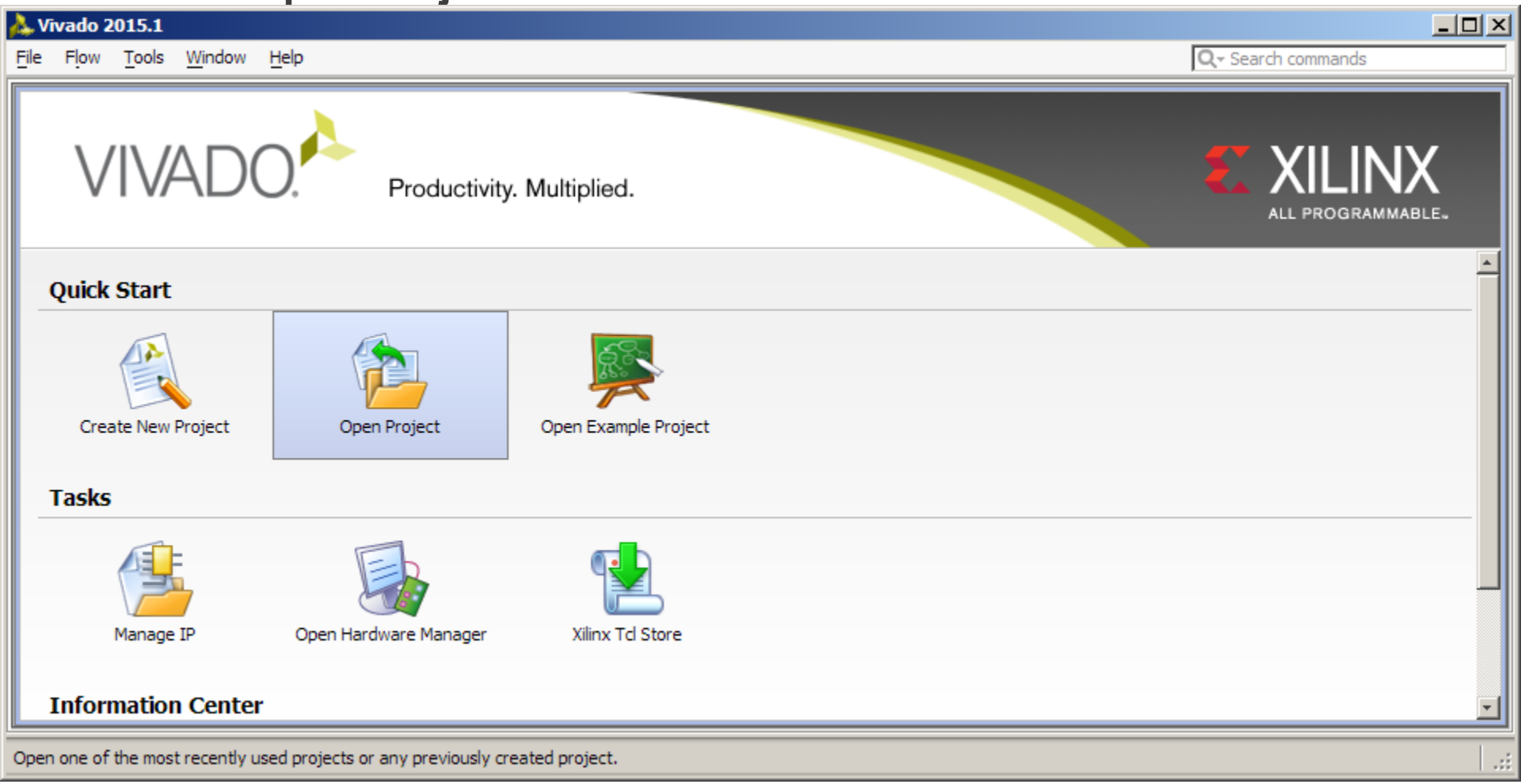
Compile AC701 MultiBoot Design

Compile AC701 MultiBoot Design

- Open Vivado

  Start → All Programs → Xilinx Design Tools → Vivado 2015.1 → Vivado

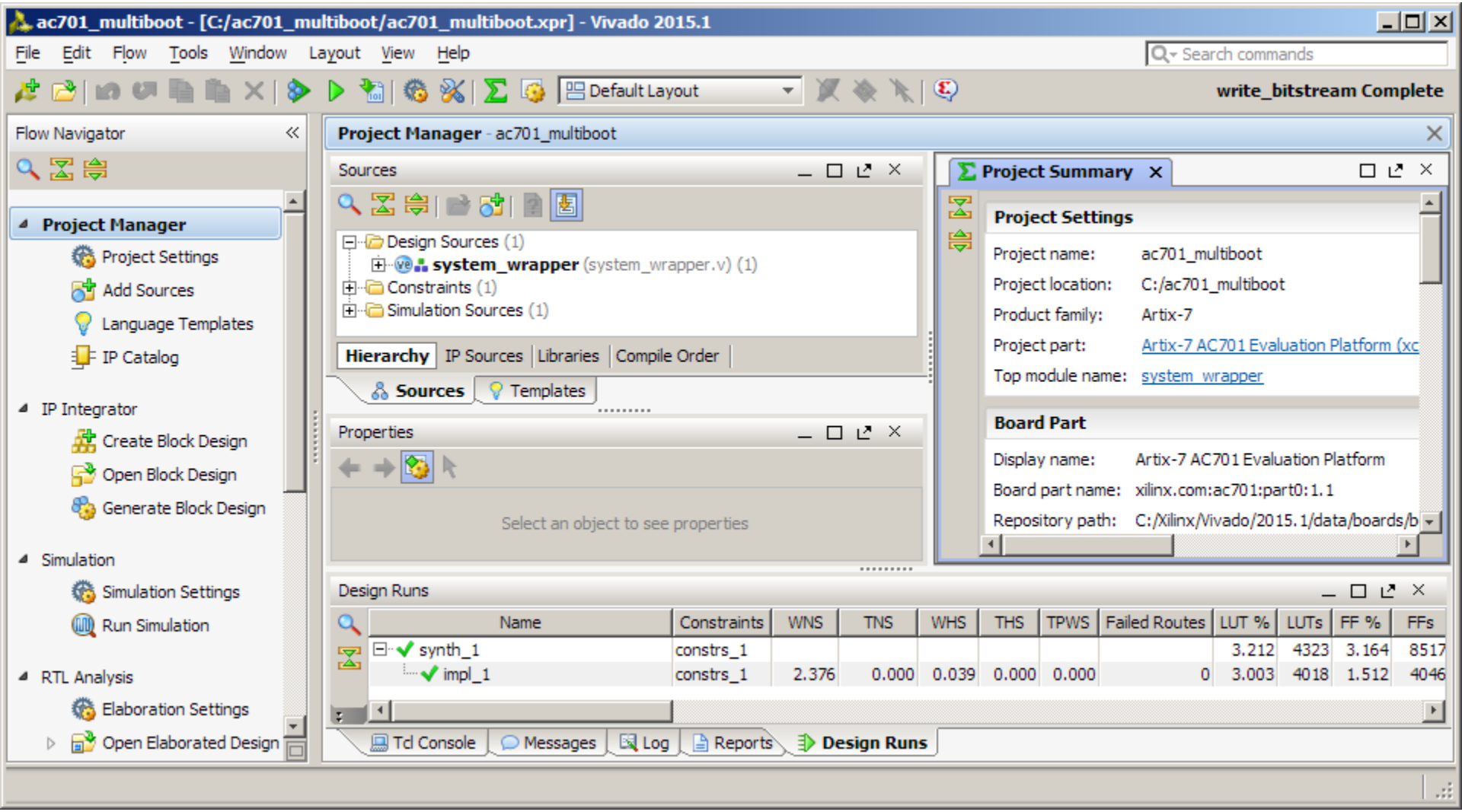
- Select Open Project



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

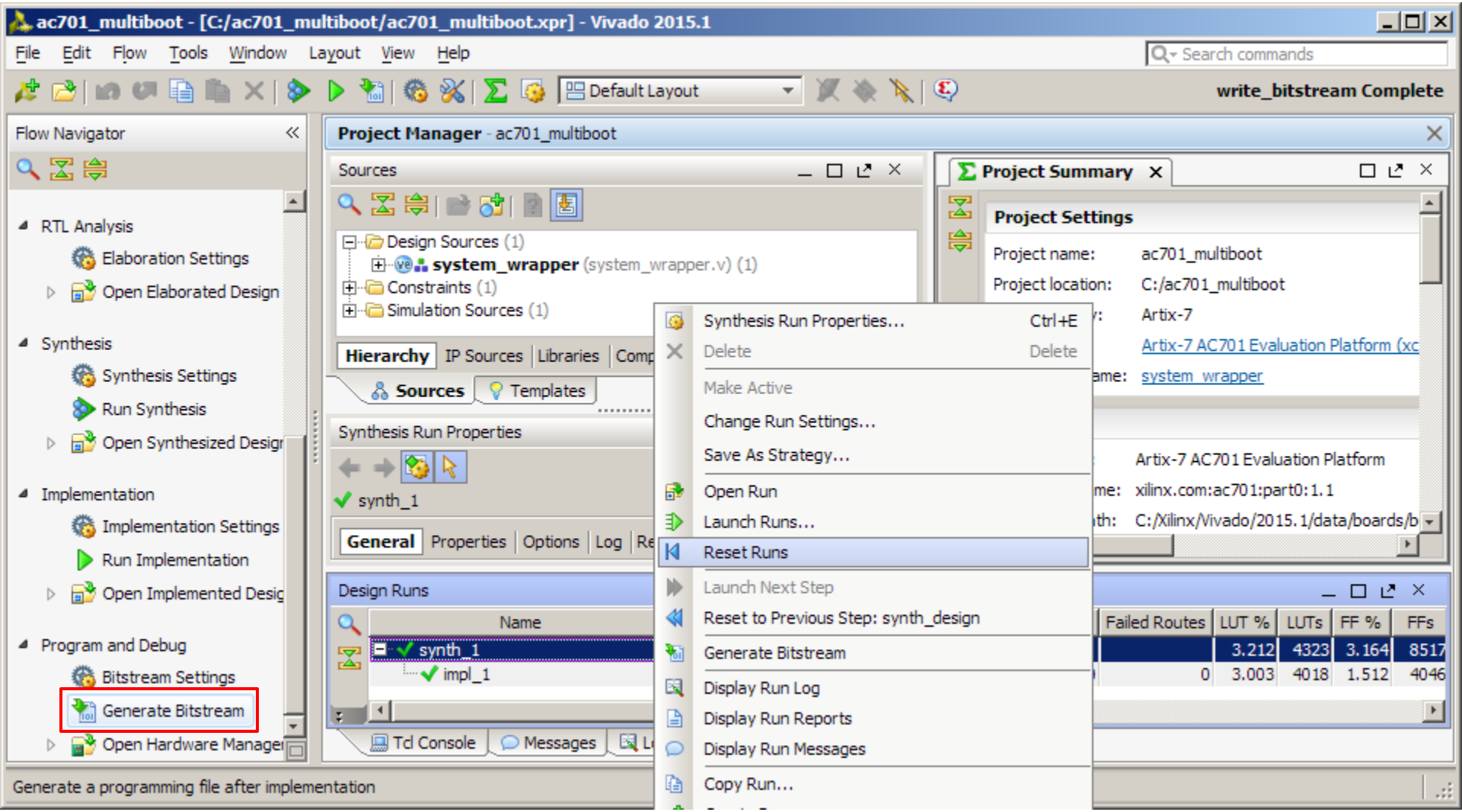
- Open the AC701 Design:
  - **<Design Name>\ac701_multiboot.xpr**



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

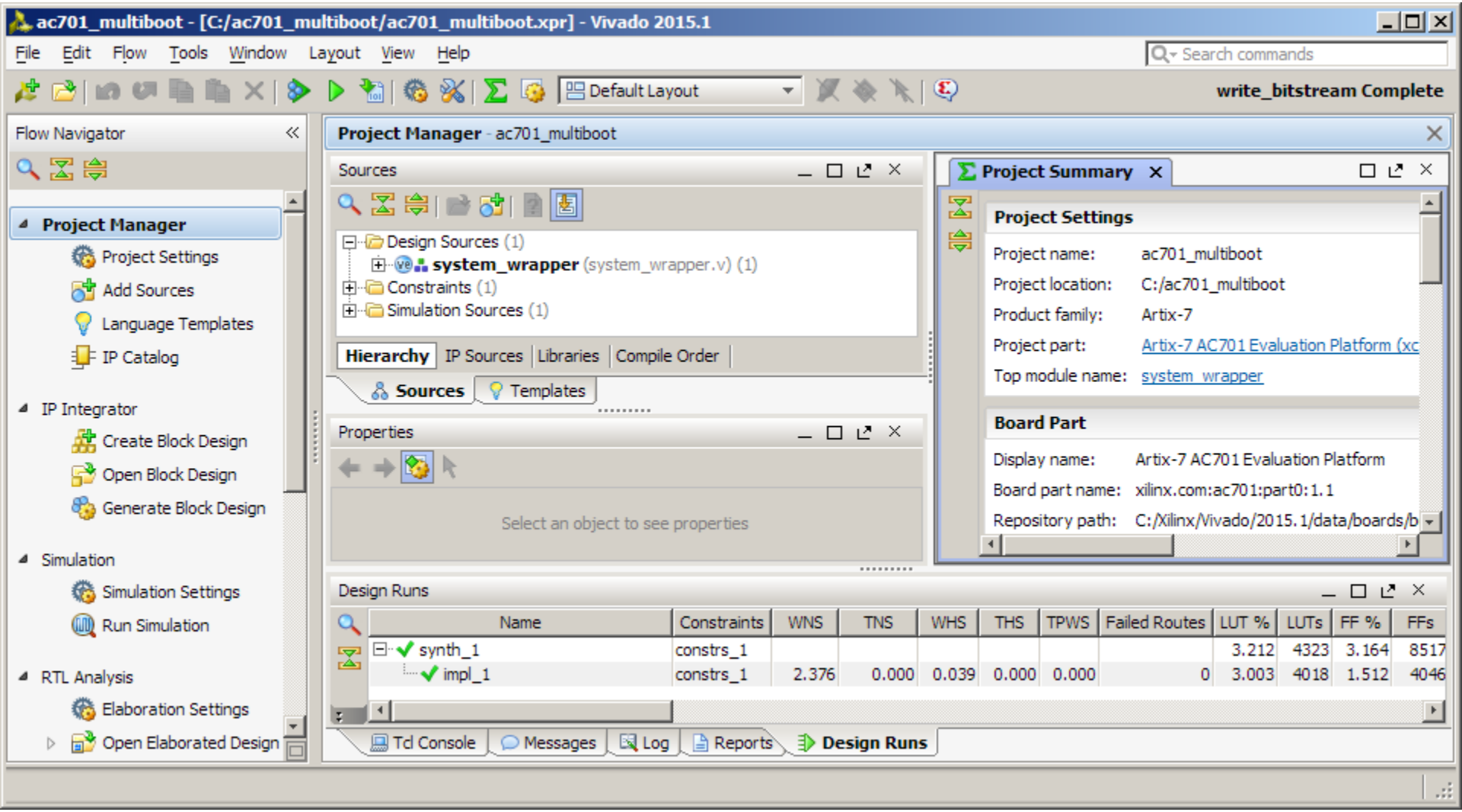
- **The design is fully implemented; you can recompile, or export to SDK**
  - To recompile, right-click **synth_1**, select **Reset Runs** then **Generate Bitstream**



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

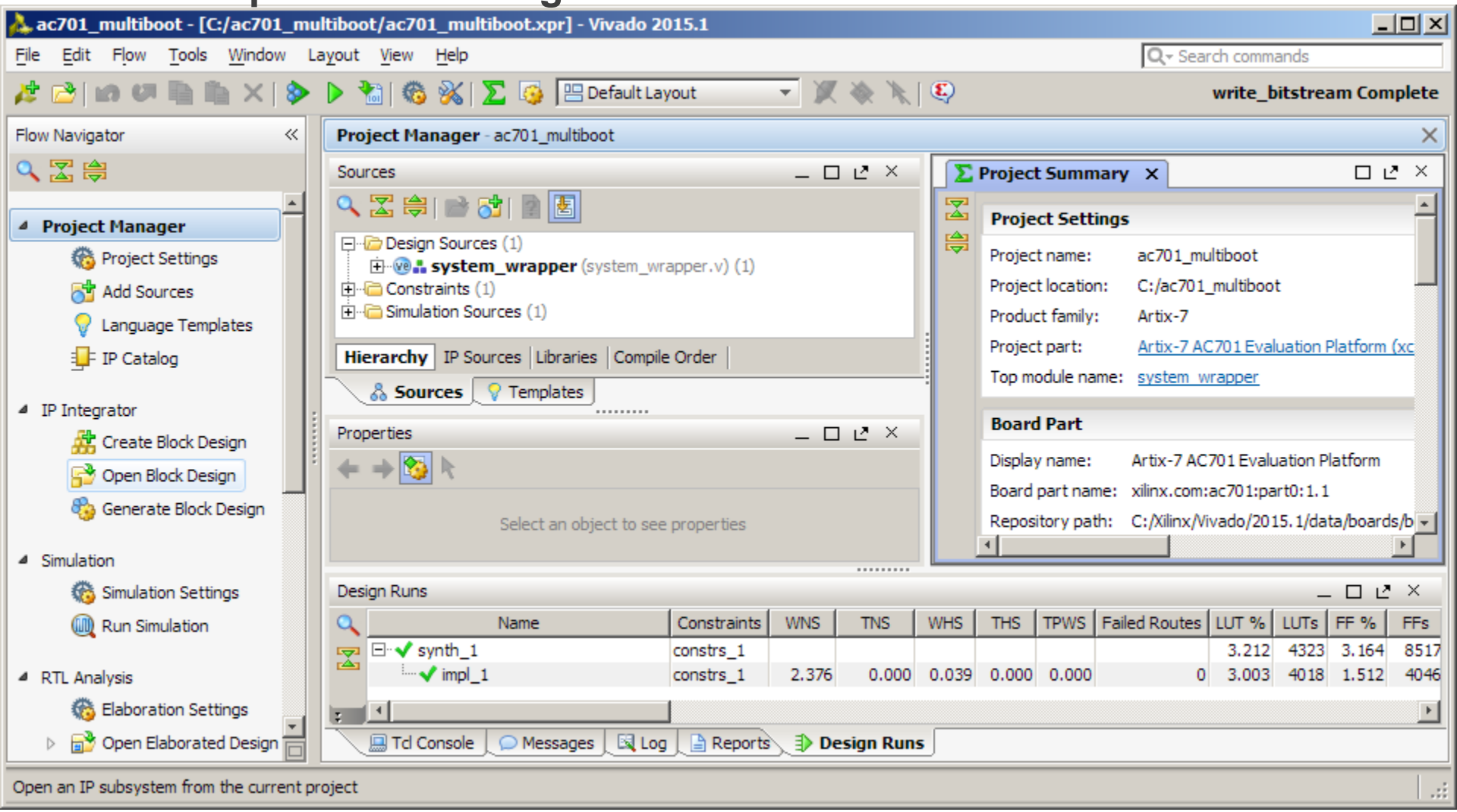
- Once done, both the Synthesis and Implementation will have green check marks



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

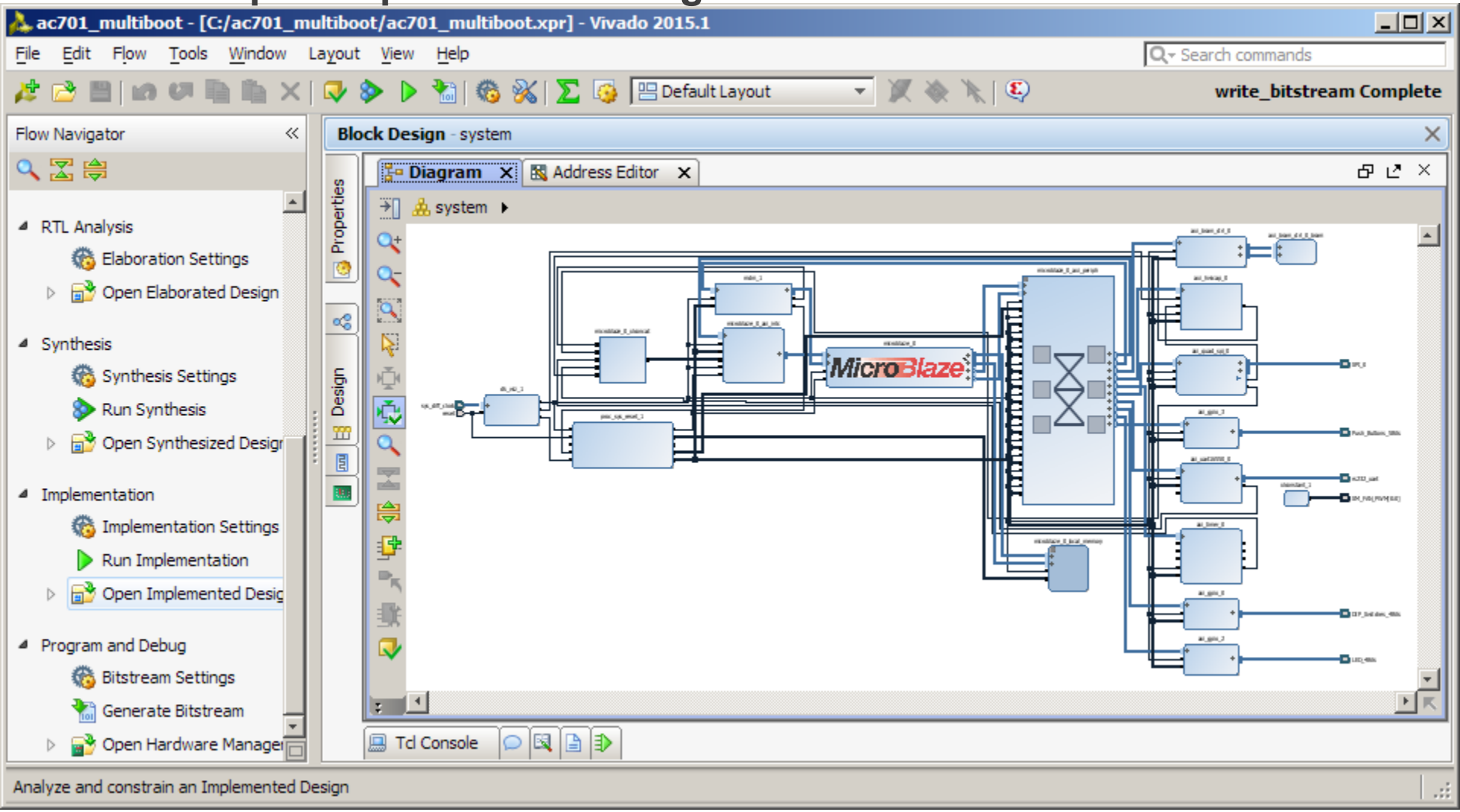
- The MultiBoot Design has been implemented with IP Integrator (IPI)
- Click Open Block Design



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

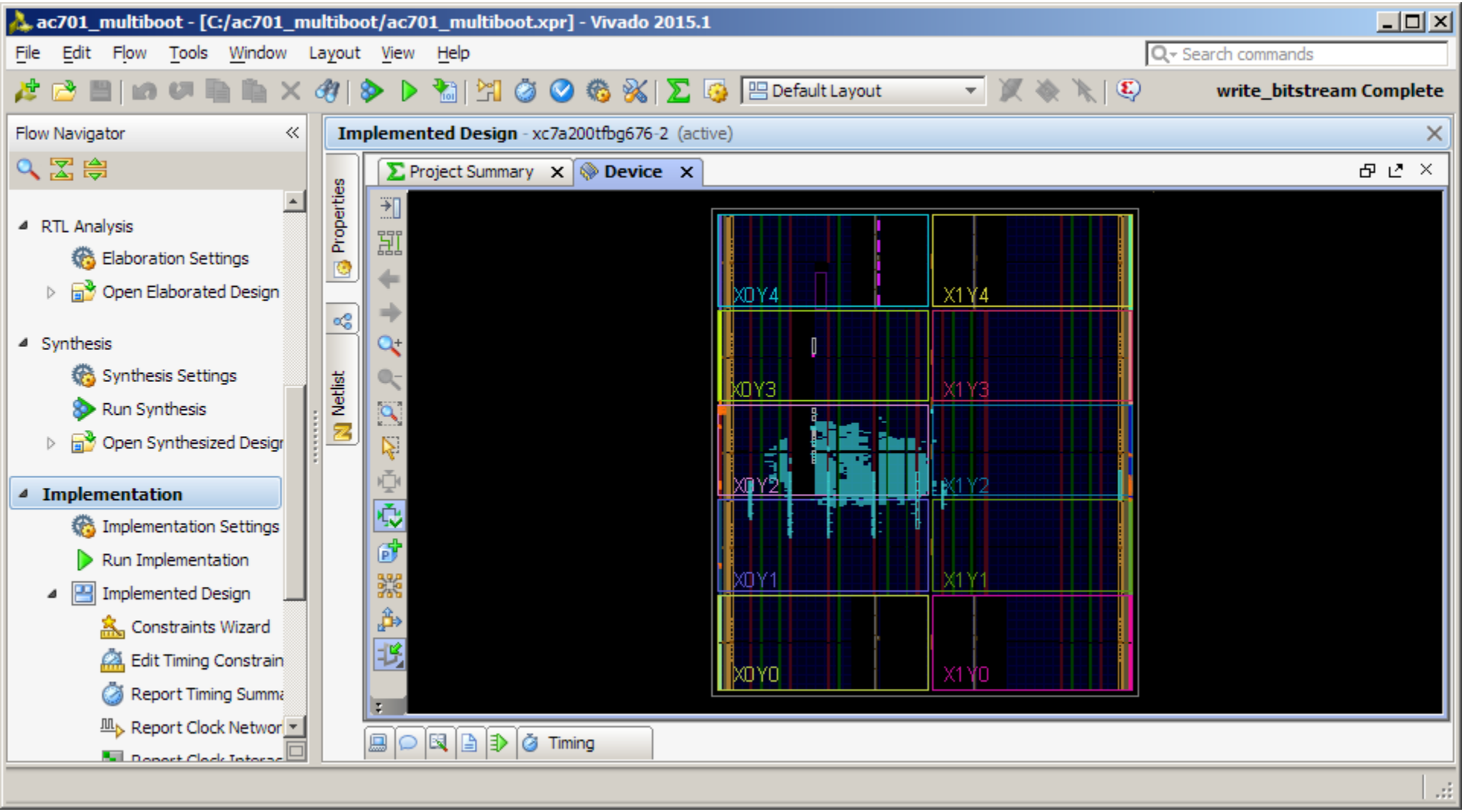
- All the IP Blocks used in the design can be seen in this view
- Click Open Implemented Design



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

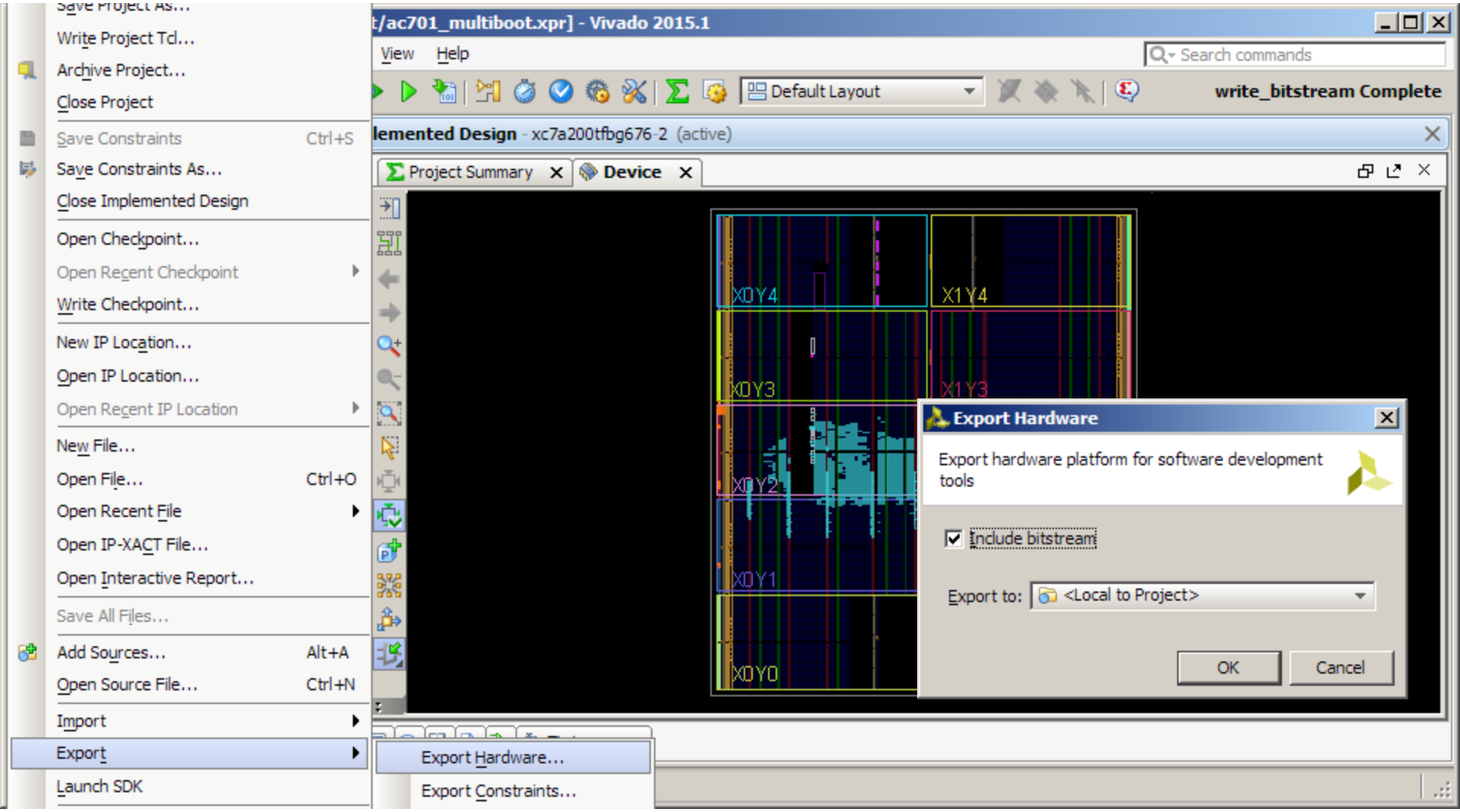
- View Implemented Design



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

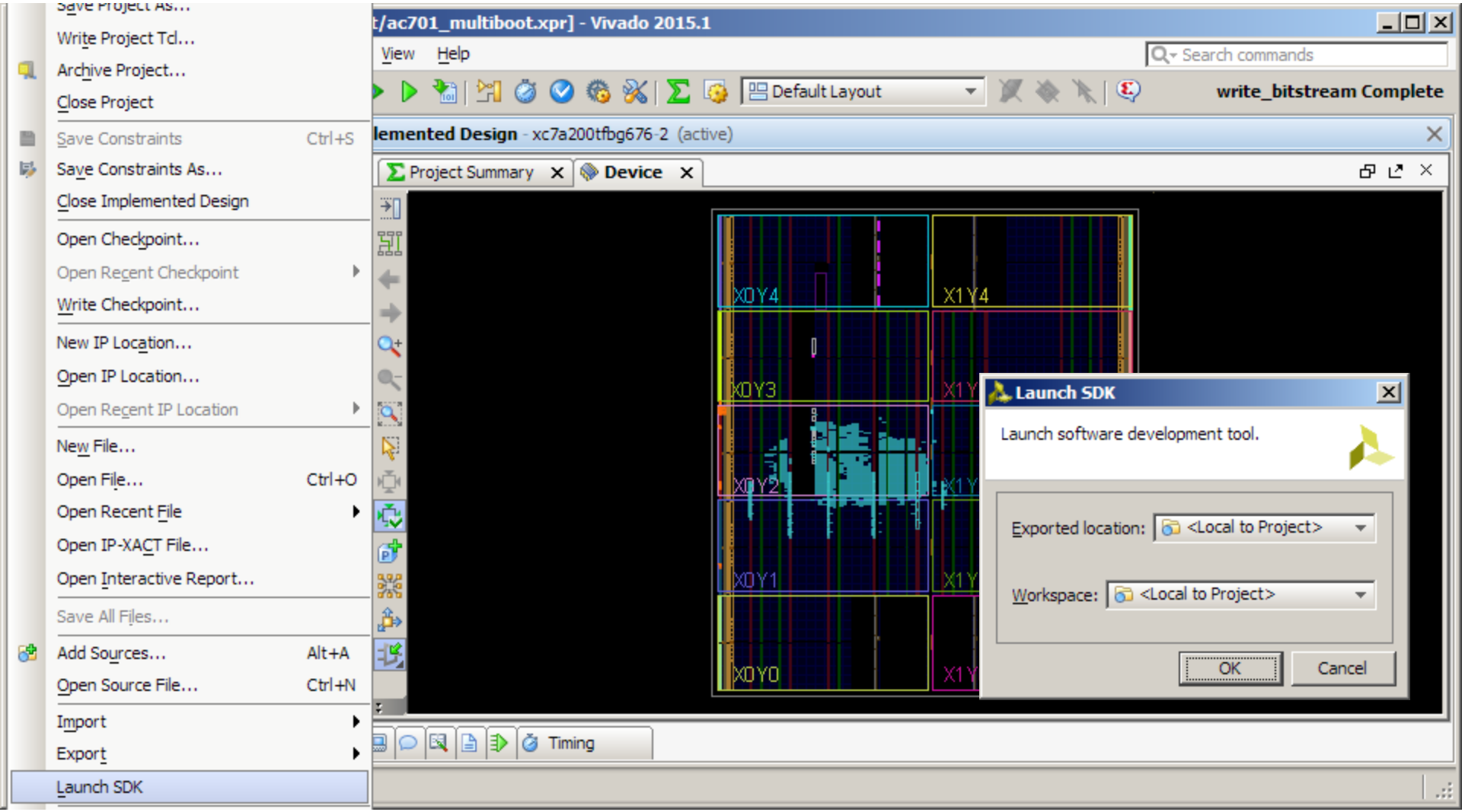
- Select File → Export → Export Hardware
- Click OK



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

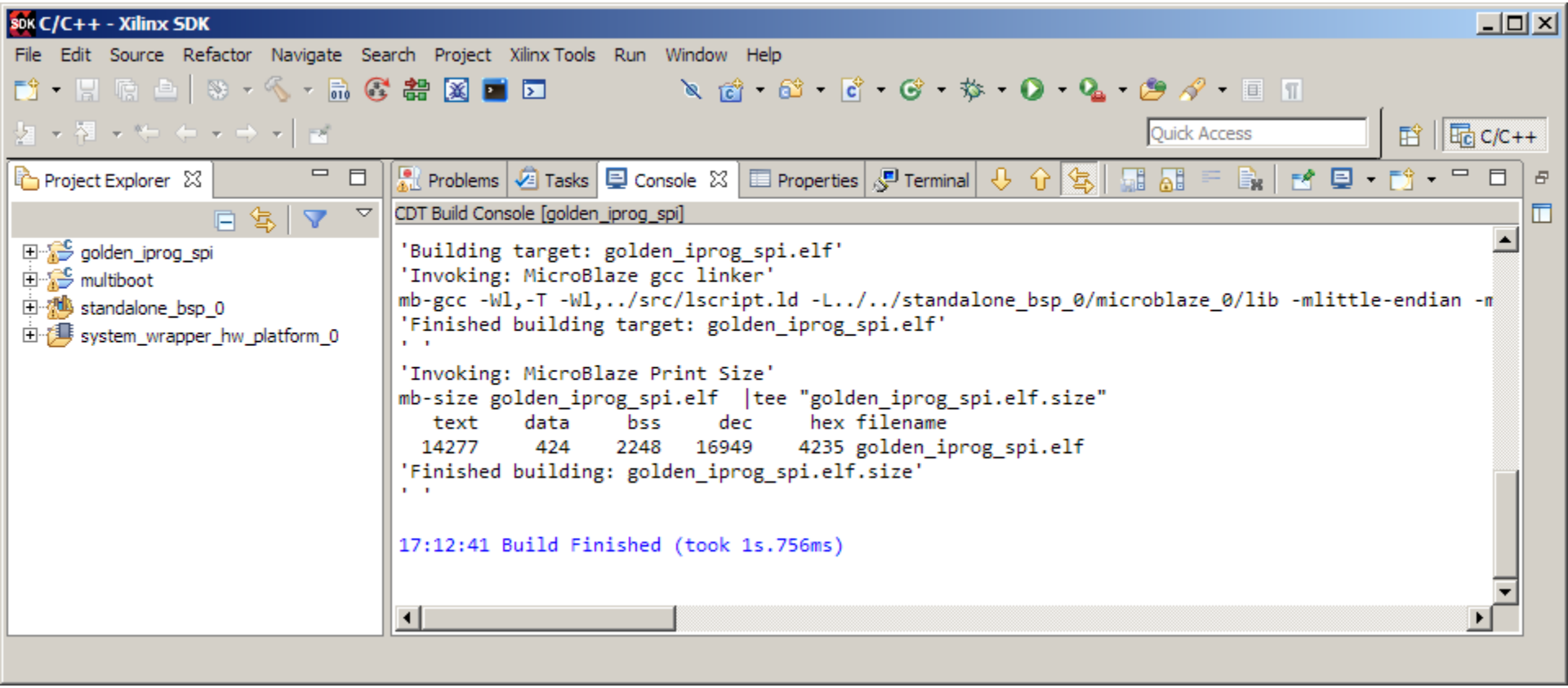
- Select File → Launch SDK
- Click OK



Note: Presentation applies to the AC701

Compile AC701 Software in SDK

- **SDK Software Compile - Build ELF files in SDK**
  - When done, close SDK and return to Vivado



Note: Presentation applies to the AC701

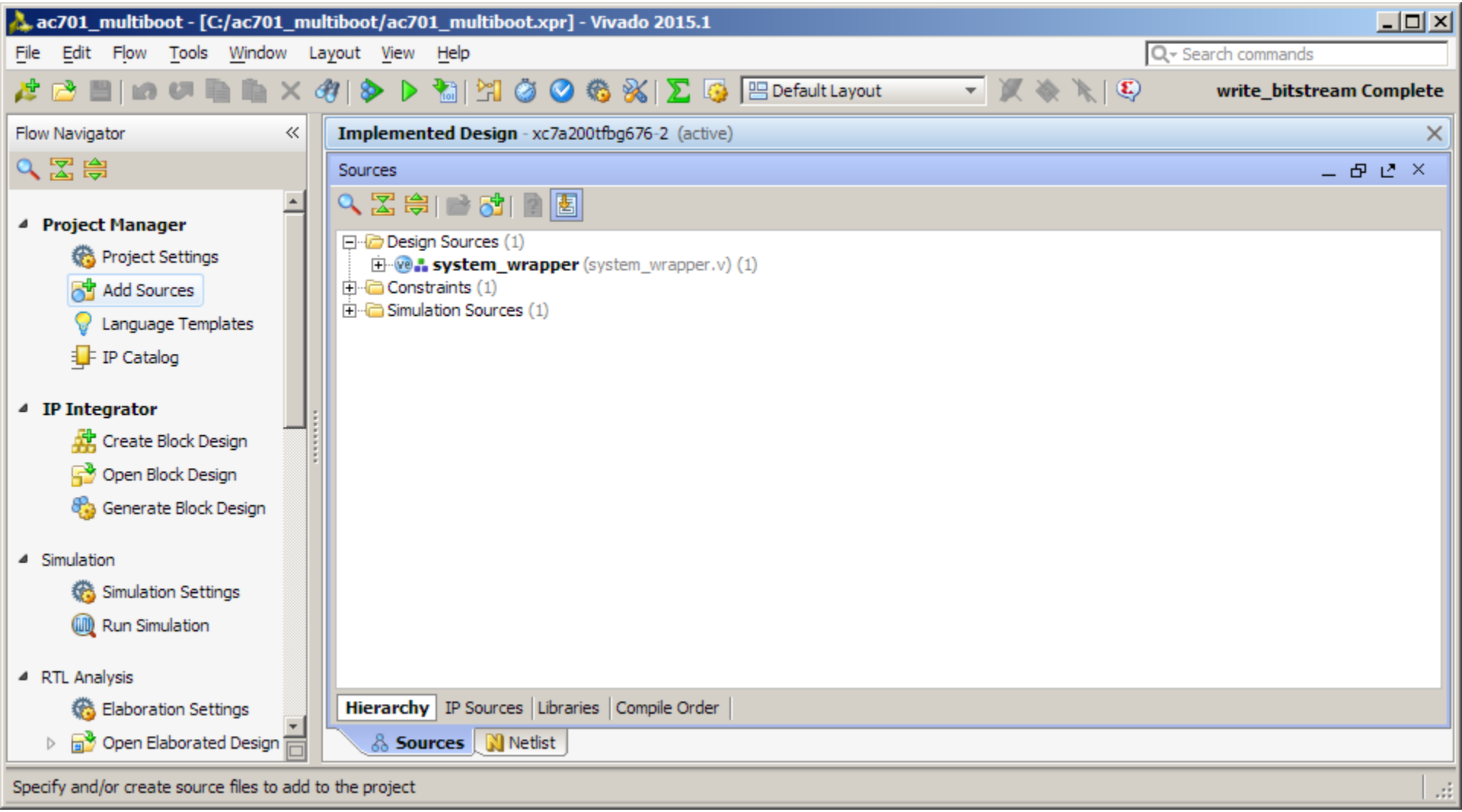
Compile AC701 MultiBoot Design

- **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 SPI 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

Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

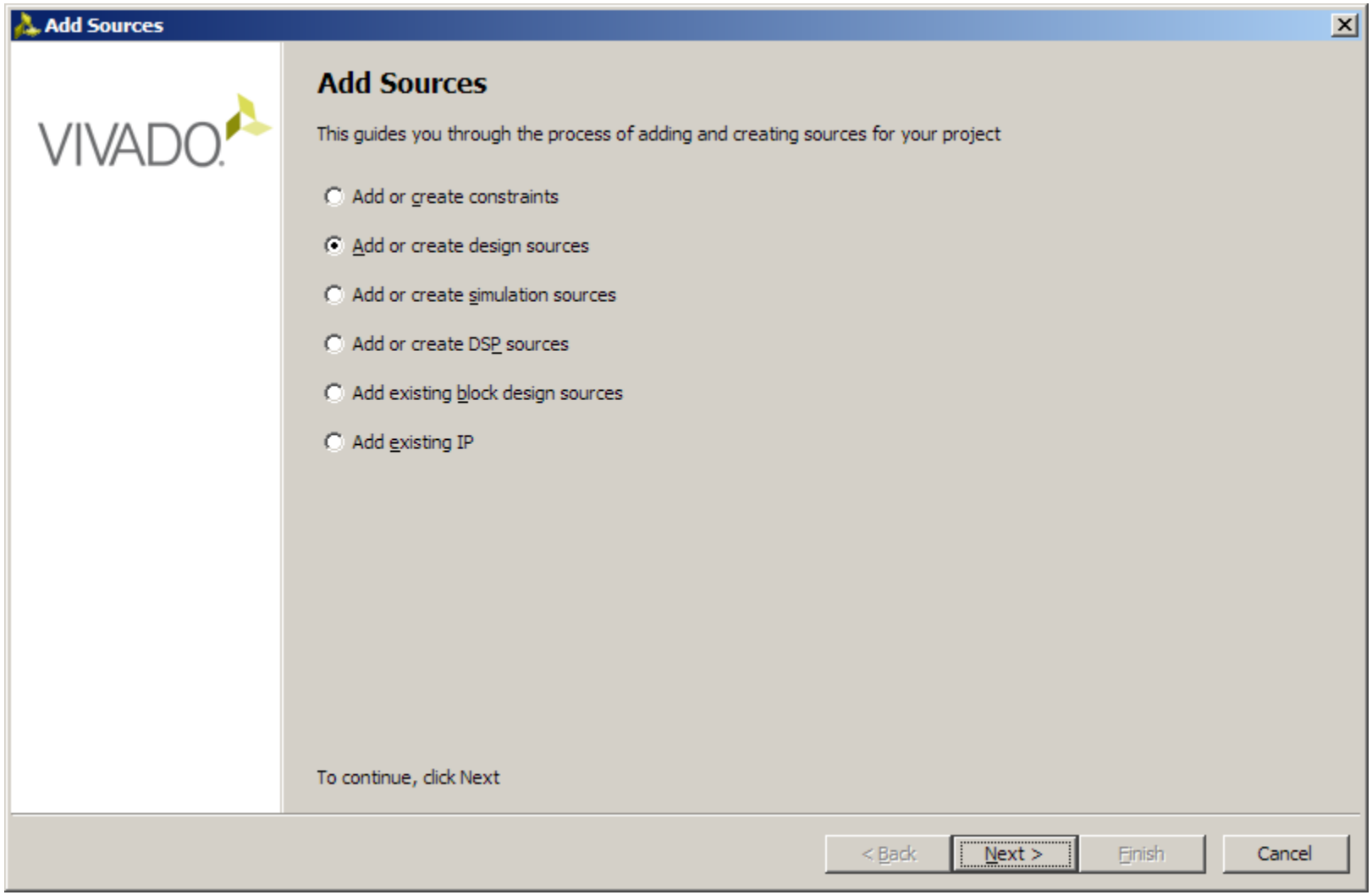
- Select Add Sources



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

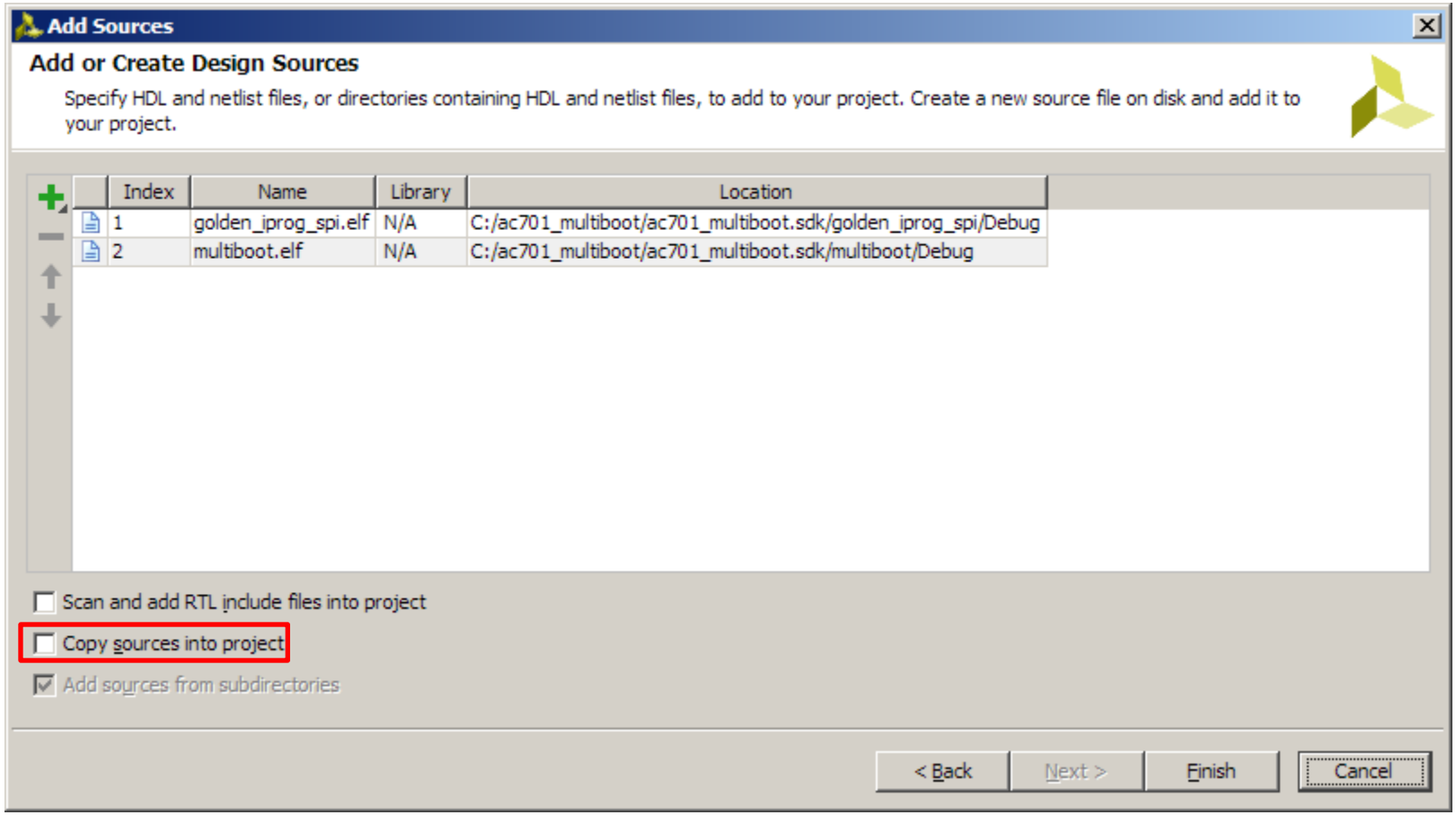
- Select Add or Create Design Sources



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

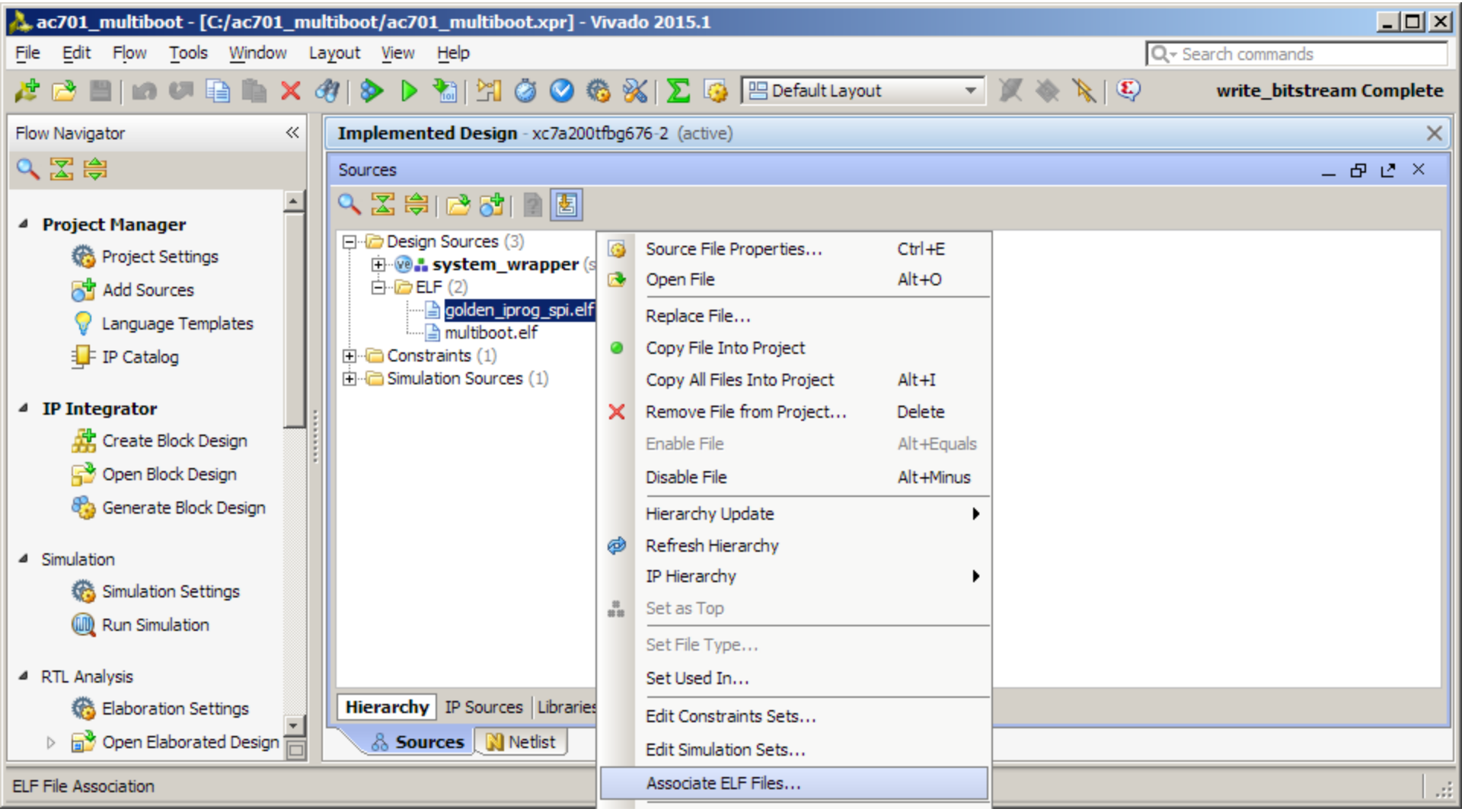
- Add golden_iprog_spi.elf and multiboot.elf from the SDK tree
- Make sure Copy sources into project is deselected
  - This eliminates the need to re-Associate the ELF each time it is recompiled
- Click Finish



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

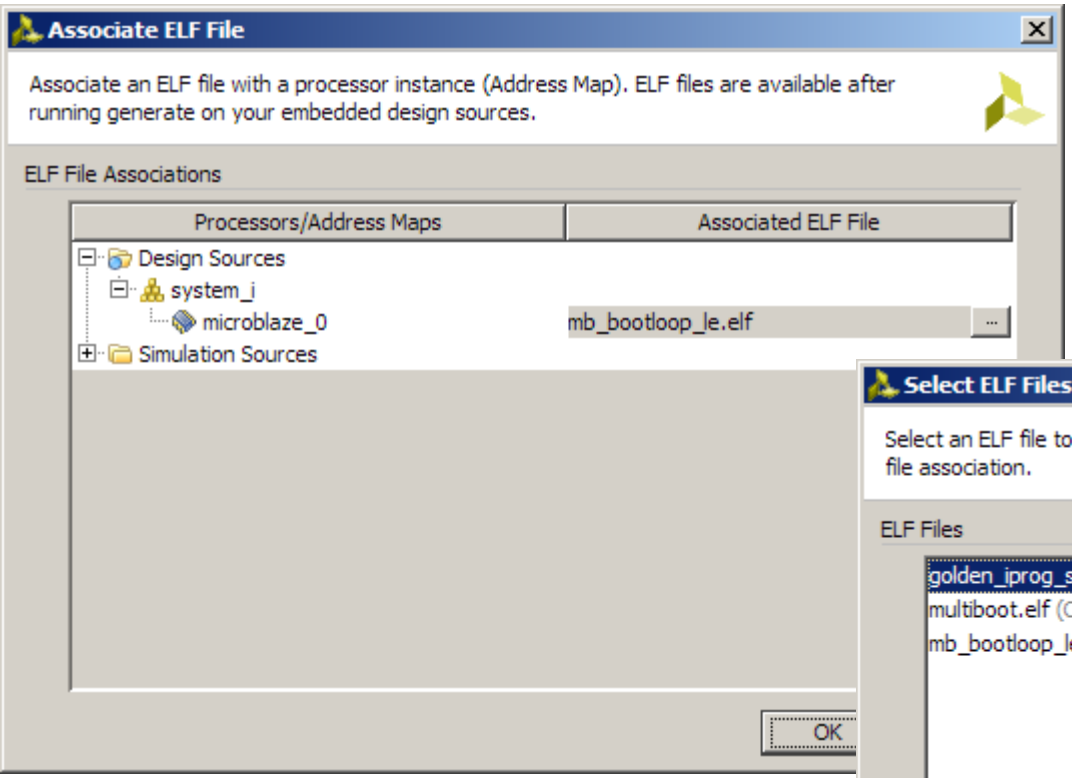
- Right-click on one of the ELF files and select Associate ELF files

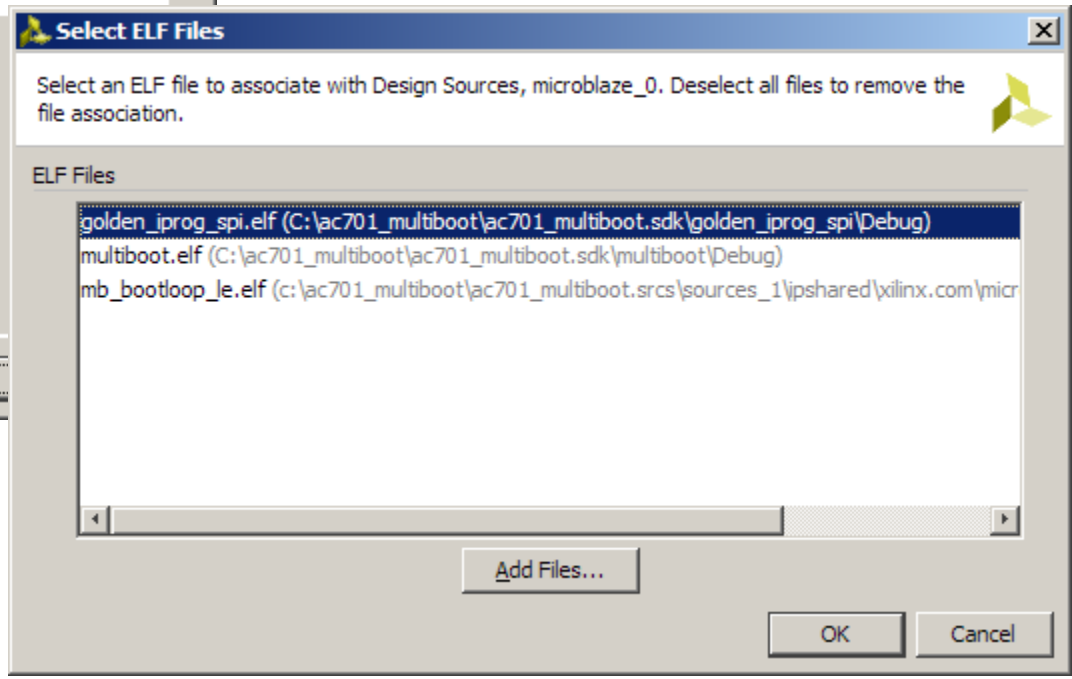


Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

- Click the button to the right; select the golden_iprog_spi.elf then click OK twice

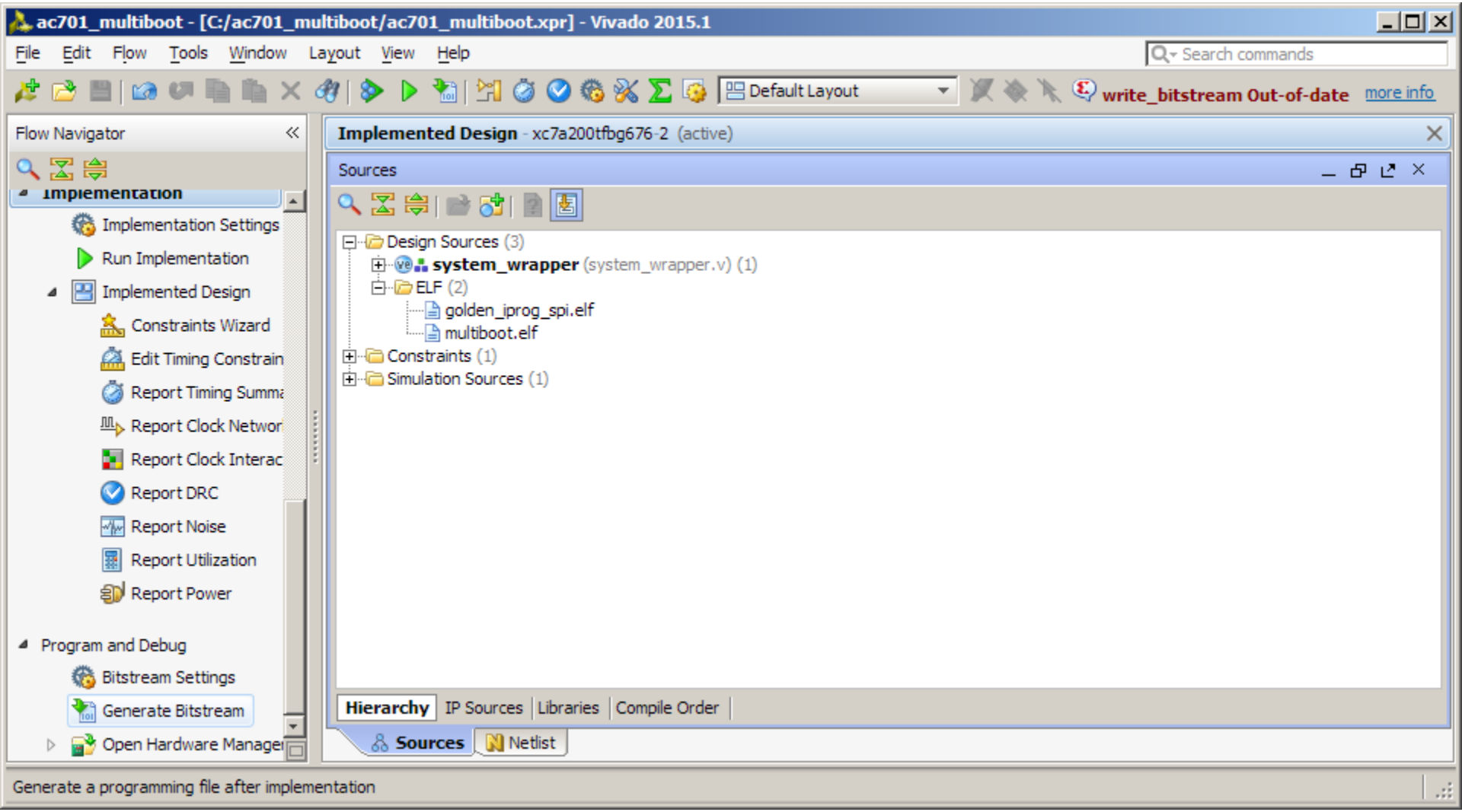




Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

- Select Generate Bitstream



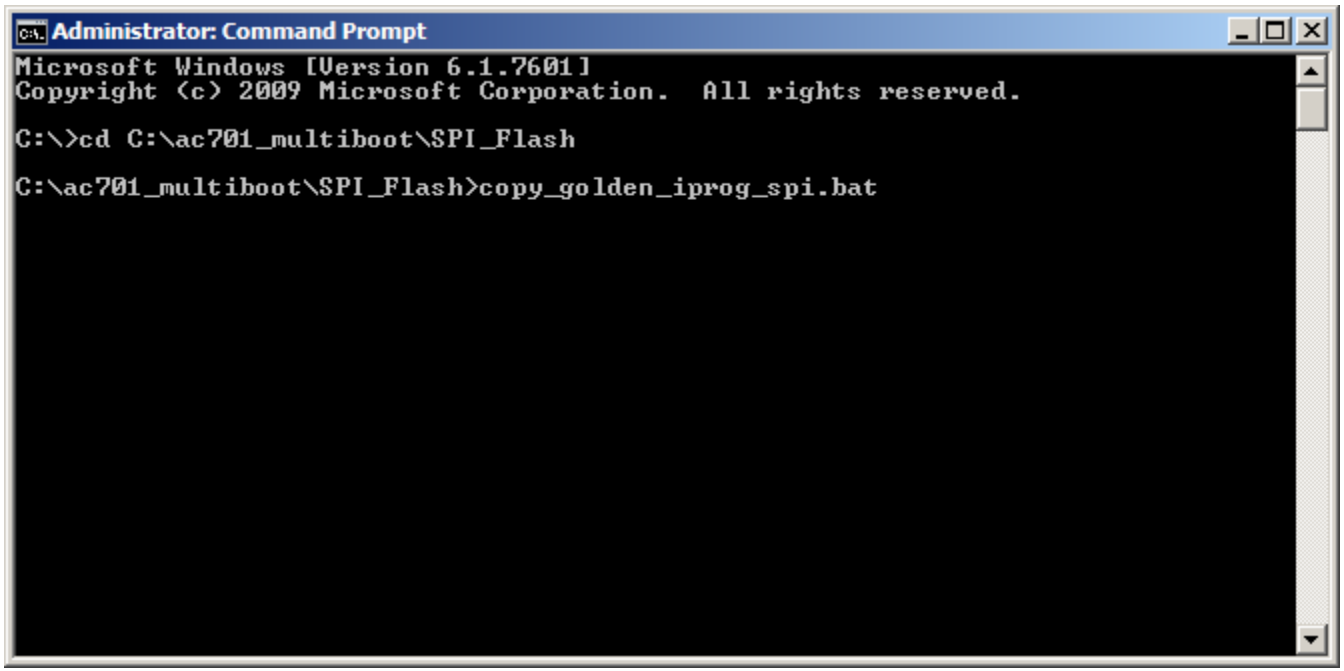
Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

- Once the write_bitstream command finishes, open a Windows Prompt and enter these commands:

  cd C:\ac701_multiboot\SPI_Flash

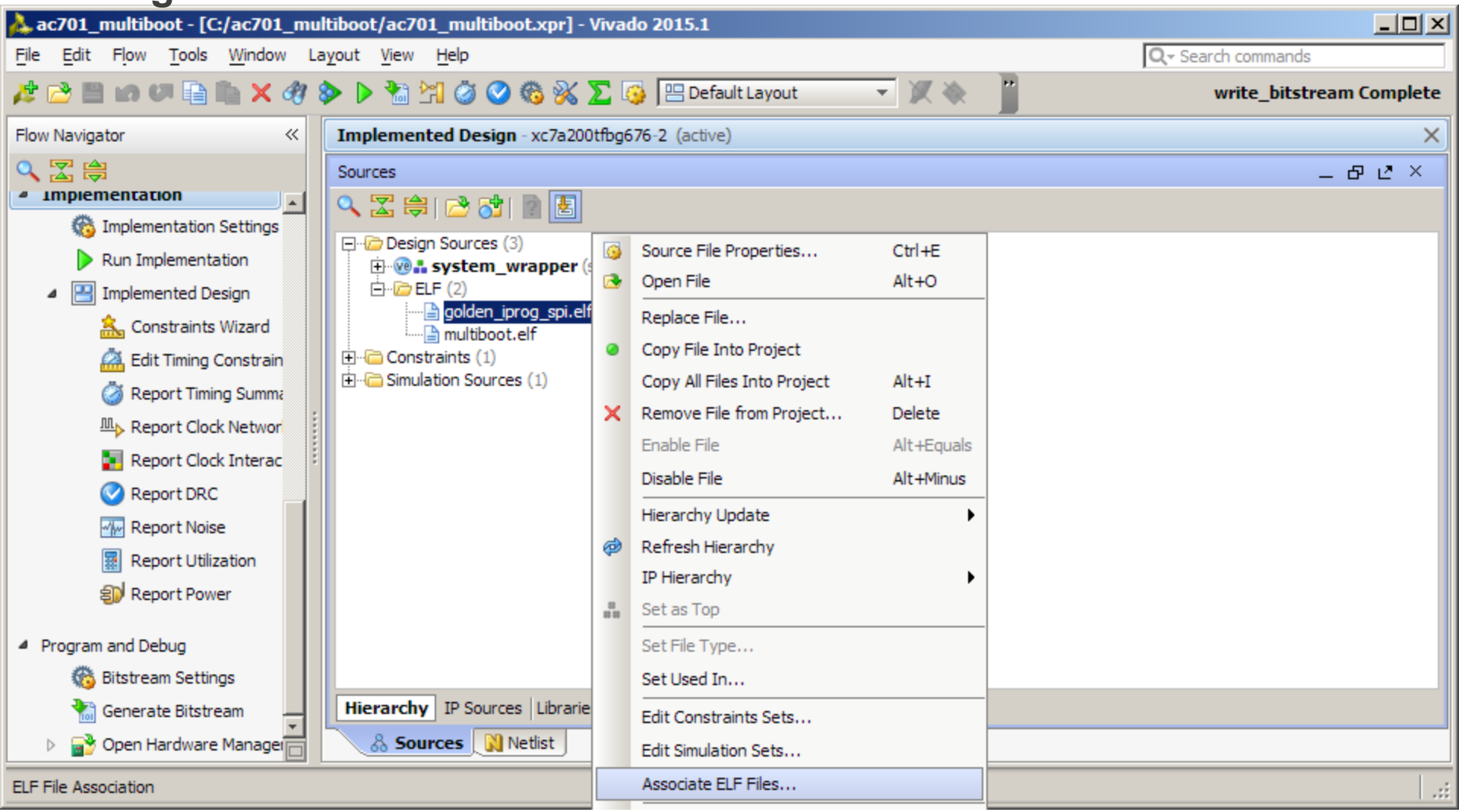
  copy_golden_iprog_spi.bat



Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

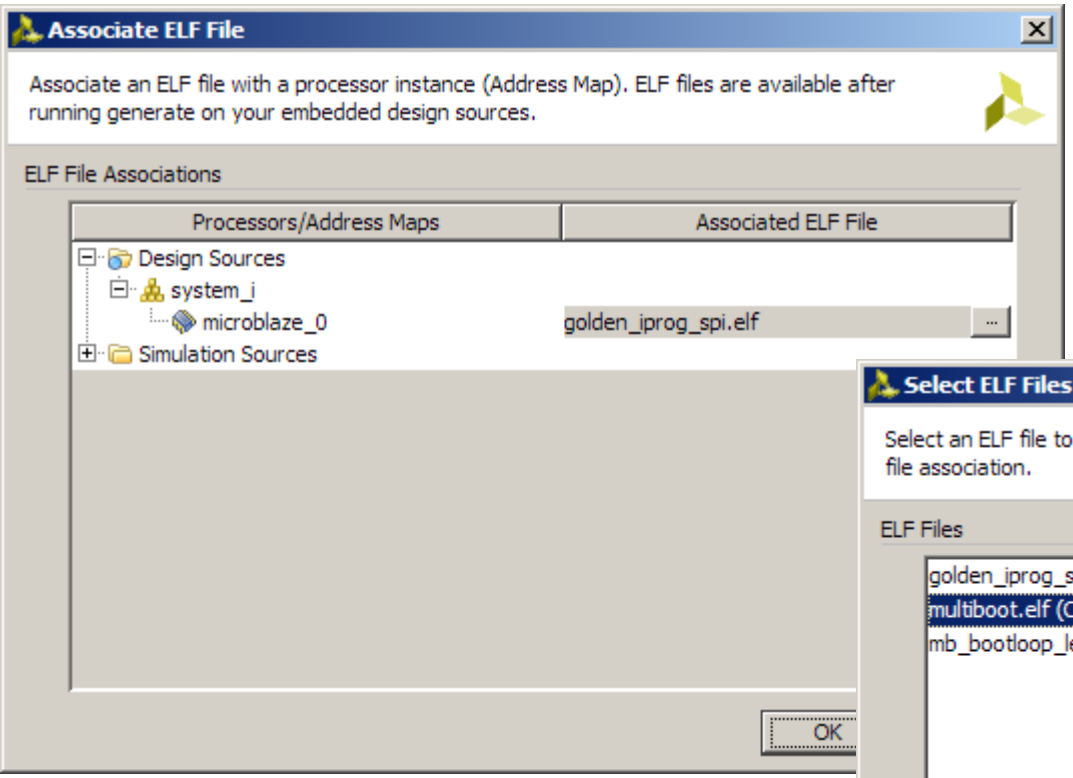
- Now associate the second ELF file
- Right-click on one of the ELF file and select Associate ELF files

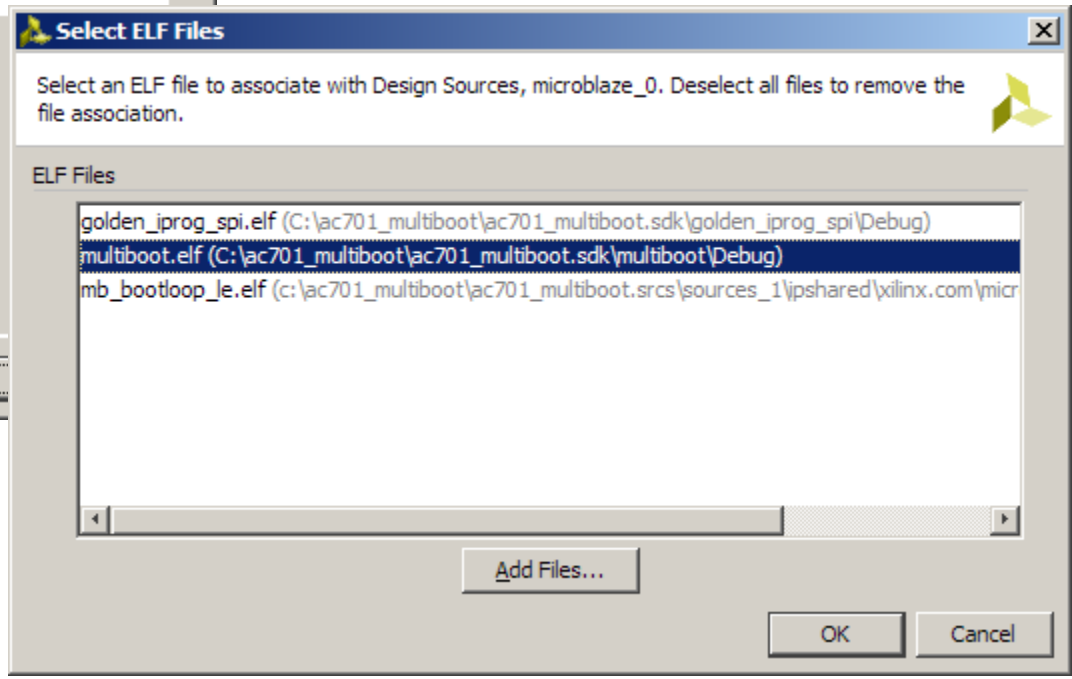


Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

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

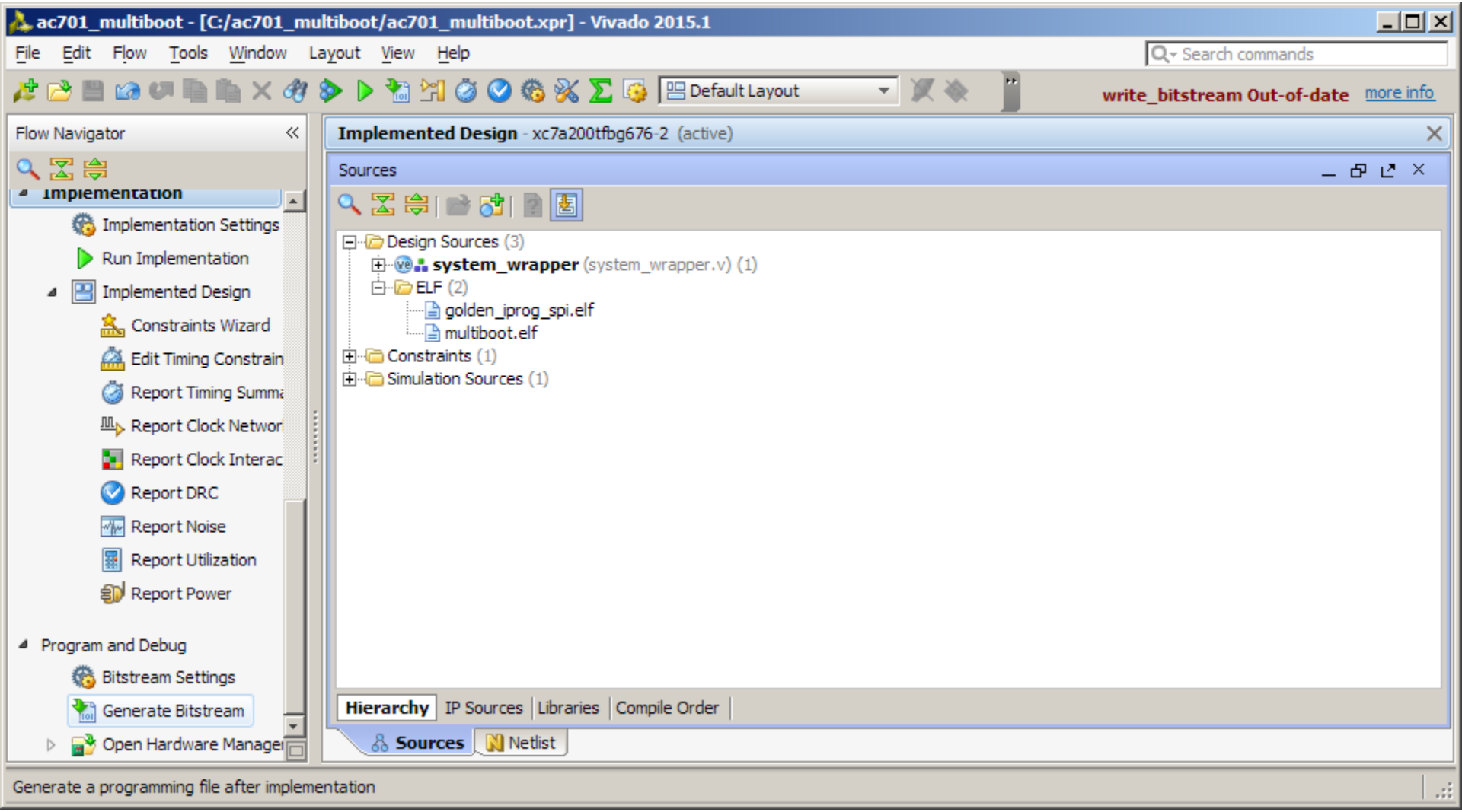




Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

- Select Generate Bitstream

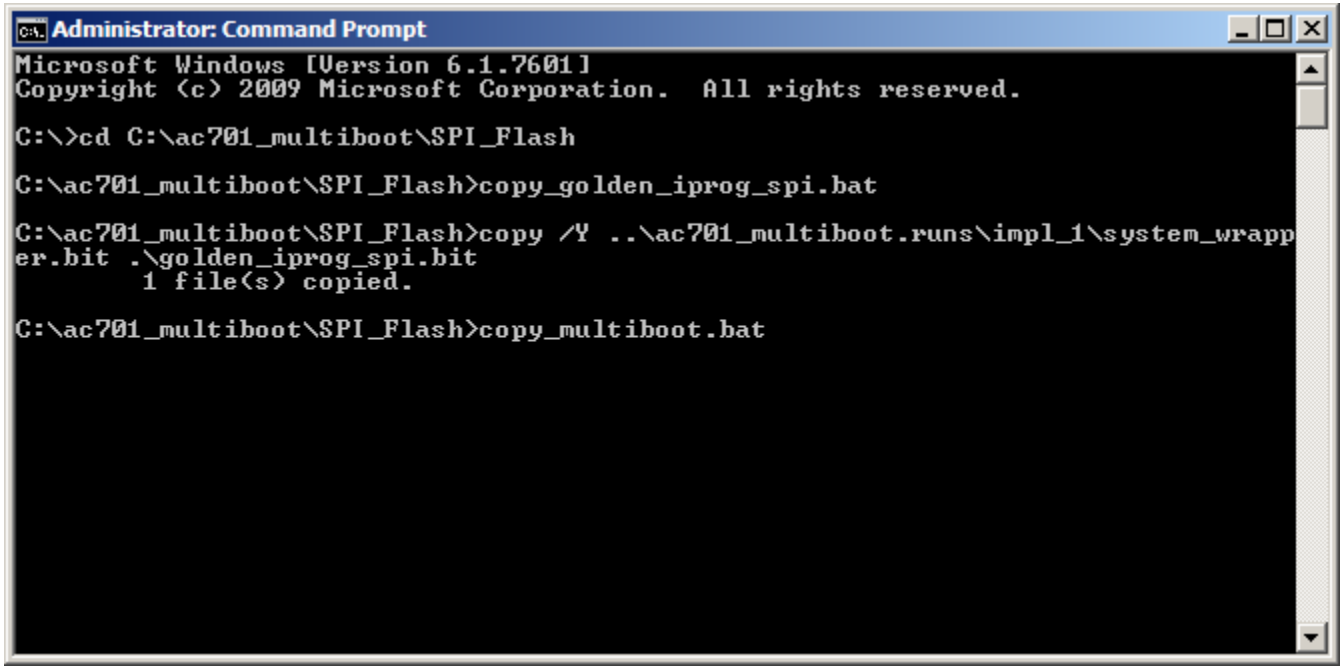


Note: Presentation applies to the AC701

Compile AC701 MultiBoot Design

- From the Command Prompt type:

  copy_multiboot.bat

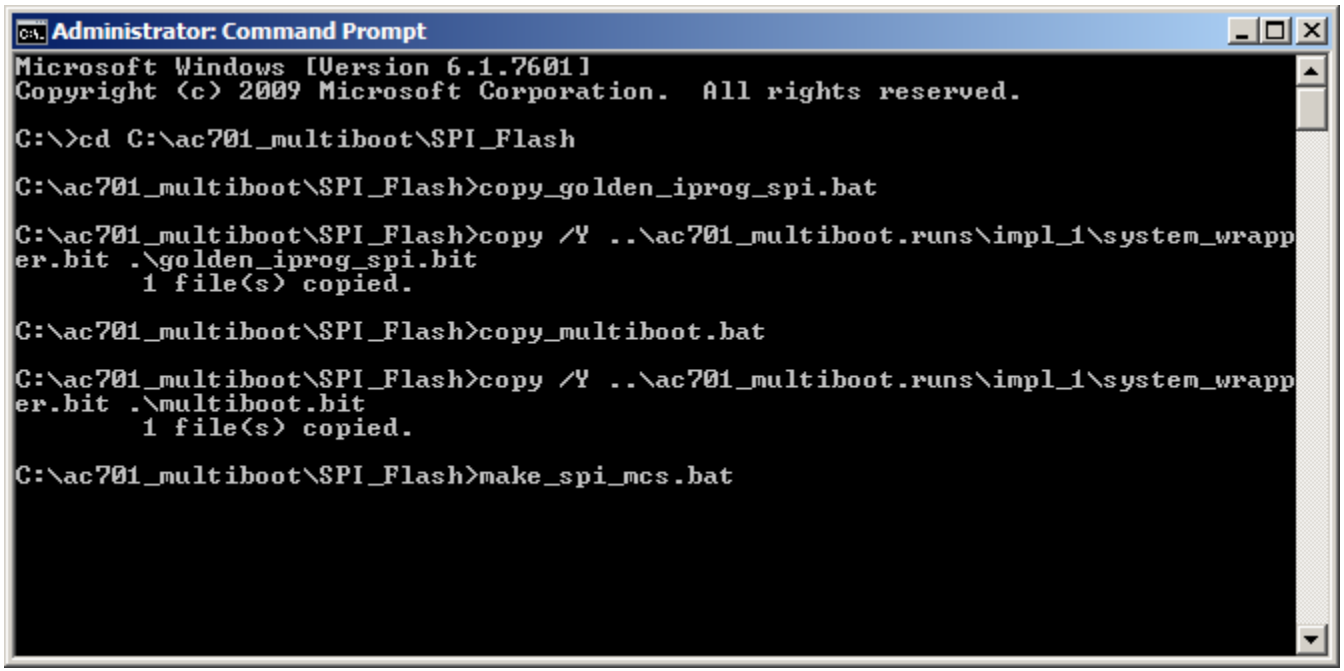


Note: Presentation applies to the AC701

Generate AC701 PROM File

- From the Command Prompt type:

  make_spi_mcs.bat



Note: Presentation applies to the AC701

Artix-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
FFFFFFFF	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

Note: Presentation applies to the AC701

MultiBoot Register Setup & Command

- IPROG command issued via the software in xhwicap_low_level_example.c

UltraEdit - [C:\ac701_multiboot\ac701_multiboot.sdk\SDK\SDK_Export\golden_iprog_spi\src\xhwicap_low_level_example.c]

File Edit Search Insert Project View Format Column Macro Scripting Advanced Window Help

```
//};

// Reusing IDCODE array and code to send IPROG
// From UG470 p. 126, Table 7-1
static u32 ReadId[HWICAP_EXAMPLE_BITSTREAM_LENGTH] =
{
  0xFFFFFFFF, /* Dummy Word */
  0xAA995566, /* Sync Word*/
  0x20000000, /* Type 1 NO OP */
  0x30020001, /* Write WBSTAR cmd */
  0x00400000, /* Addr in SPI Flash of Multiboot bitstream */
  0x30008001, /* Write CMD */
  0x0000000F, /* Write IPROG */
  0x20000000, /* Type 1 NO OP  */
};
```

Note: Presentation applies to the AC701

References

References

- **IP Integrator Documentation**
  - Vivado Design Suite Tcl Command Reference Guide – UG835
    - http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_1/ug835-vivado-tcl-commands.pdf
  - Designing IP Subsystems Using IP Integrator – UG994
    - http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_1/ug994-vivado-ip-subsystems.pdf
- **7 Series Configuration**
  - 7 Series FPGAs Configuration User Guide
    - http://www.xilinx.com/support/documentation/user_guides/ug470_7Series_Config.pdf

Documentation

Documentation

- **Artix-7**
  - Artix-7 FPGA Family
    - http://www.xilinx.com/products/silicon-devices/fpga/artix-7/index.htm
  - Design Advisory Master Answer Record for Artix-7 FPGAs
    - http://www.xilinx.com/support/answers/51456.htm
- **AC701 Documentation**
  - Artix-7 FPGA AC701 Evaluation Kit
    - http://www.xilinx.com/products/boards-and-kits/ek-a7-ac701-g.html
  - AC701 Getting Started Guide – UG967
    - http://www.xilinx.com/support/documentation/boards_and_kits/ac701/2014_3/ug967-ac701-eval-kit-getting-started.pdf
  - AC701 User Guide – UG952
    - http://www.xilinx.com/support/documentation/boards_and_kits/ac701/ug952-ac701-a7-eval-bd.pdf