FPGA Programming

Class 5: Programming the Chip

September 15, 2017

Louis W. Giokas

DesignNews
CEC CONTINUING EDUCATION CENTER
Presented by:
Digi-Key ELECTRONICS
ROHDE&SCHWARZ

This Week's Agenda

Monday	FPGA Device Description
Tuesday	Design Flow
Wednesday	HDL
Thursday	Synthesis and Layout
Friday	Programming the Chip

Course Description

We start with an introduction to the class of devices called Field Programmable Gate Arrays (FPGAs). The layout and design of several types and critical parameters will be described and discussed. It is important to understand the way the device is constructed to develop effective algorithms.

The device we will be using this week will be the Microsemi IGLOO2. We will also discuss other devices and their structure.

We will introduce two common Hardware Description Languages (HDL), but give examples in one (Verilog).

Today's Agenda

- Overview
- Configuration
- Generation
- Programming the Device
- Debugging
- Production
- Conclusion
- Contact/Resources

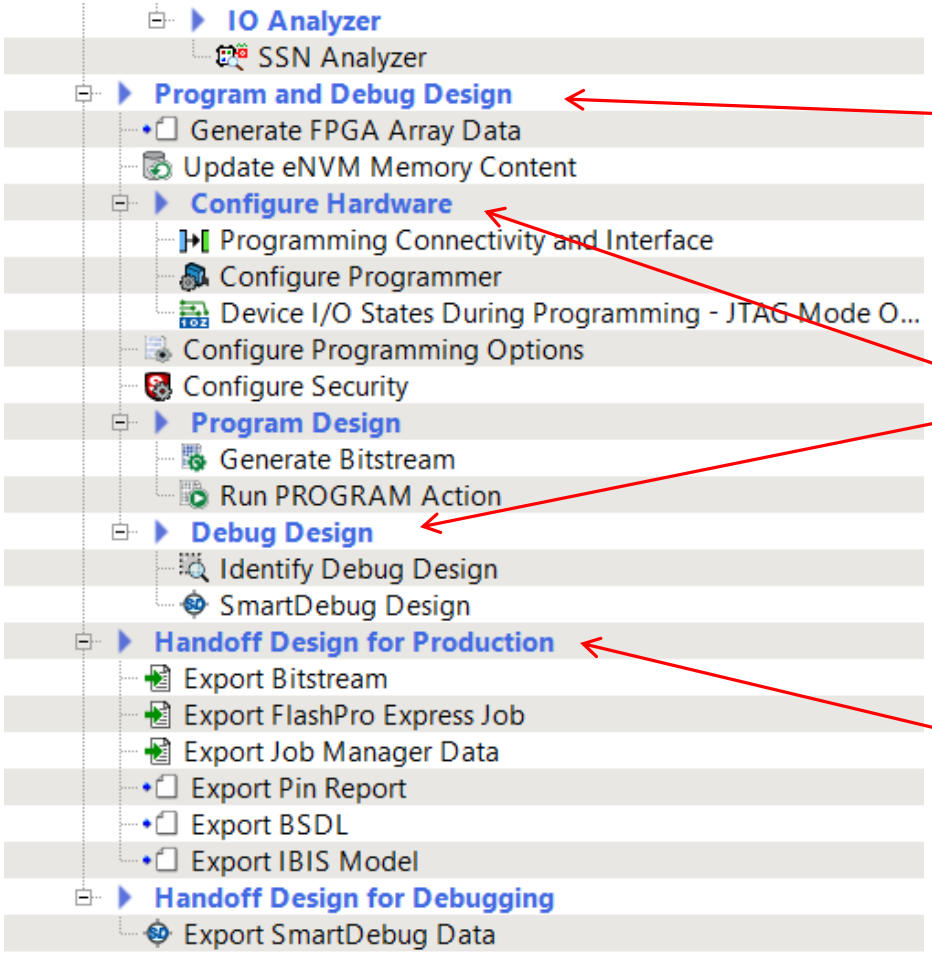
Overview

- We now come the topic of programming in the way that FPGA vendors consider it
  - Basically this consists of creating the bitmap and loading it onto the chip
- Seems simple, but there is a lot we need to do, and can do, at this stage
  - Debugging on the actual device
  - Monitor operation in real time

Overview

- Development tools also allow us to prepare everything necessary to program chips in a production environment
  - This includes more than the bitstream
    - Also includes the jobs and reports used to apply that bitstream and load data onto devices in bulk
  - We also generate data used for independent debugging

Overview



We are at the last part of the process.

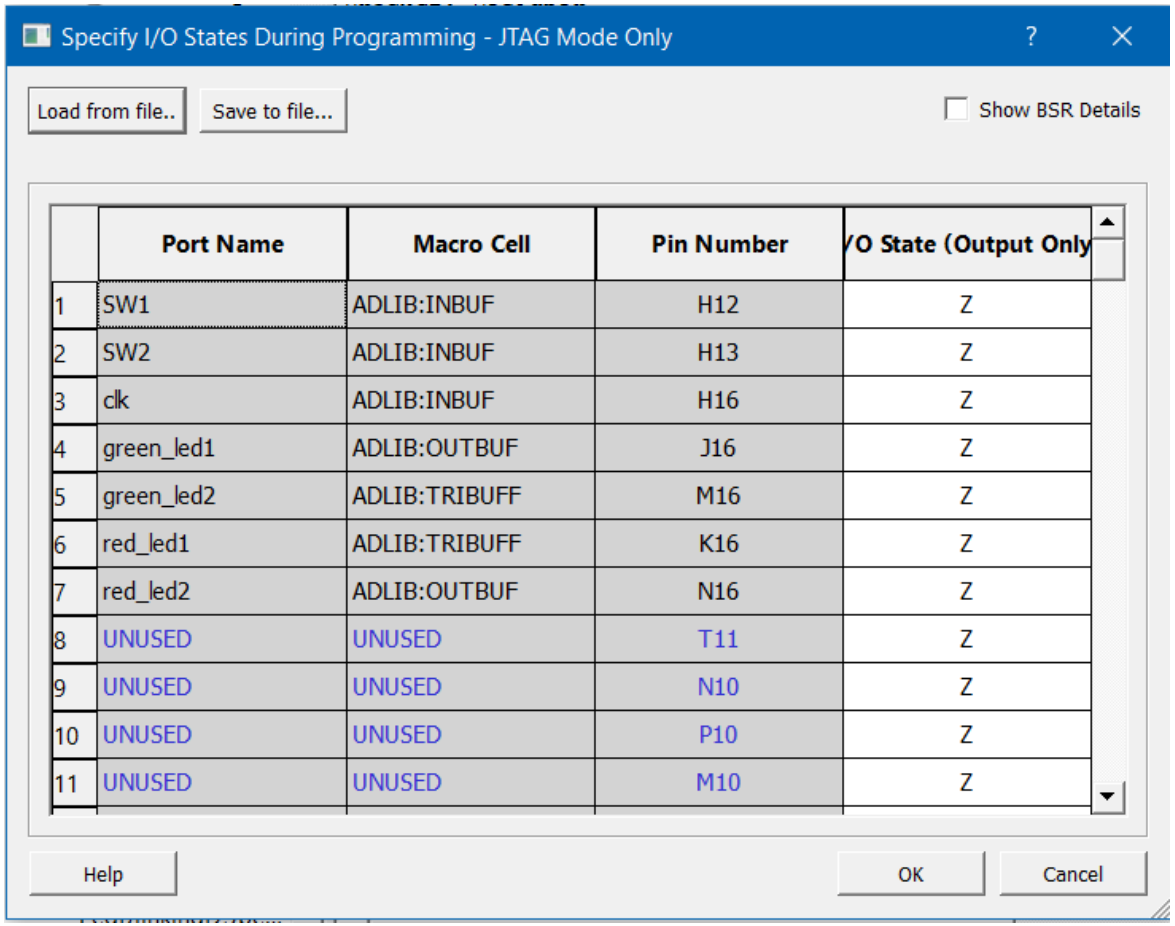
There are a number of activities that are performed at this stage for/on actual hardware.

Tasks for production

Generation

- There are a number of steps in the generation process that allow us to control the environment and the image on the device
  - Configuration of the design (version, etc.)
  - Security
  - I/O states
- We can set parameters for these separate from the algorithms we are implementing

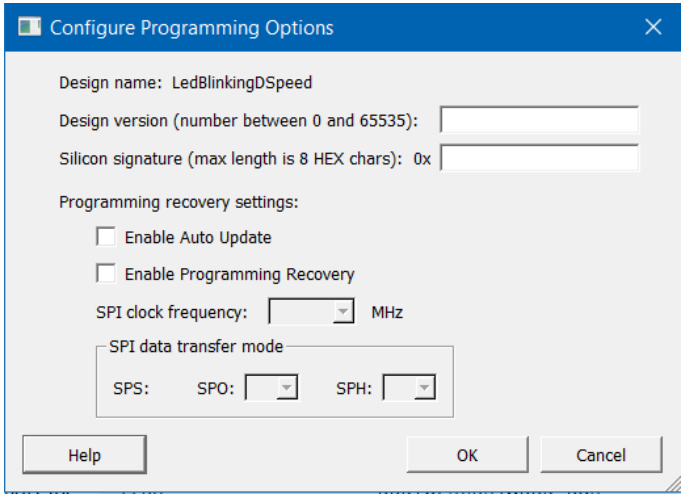
Generation



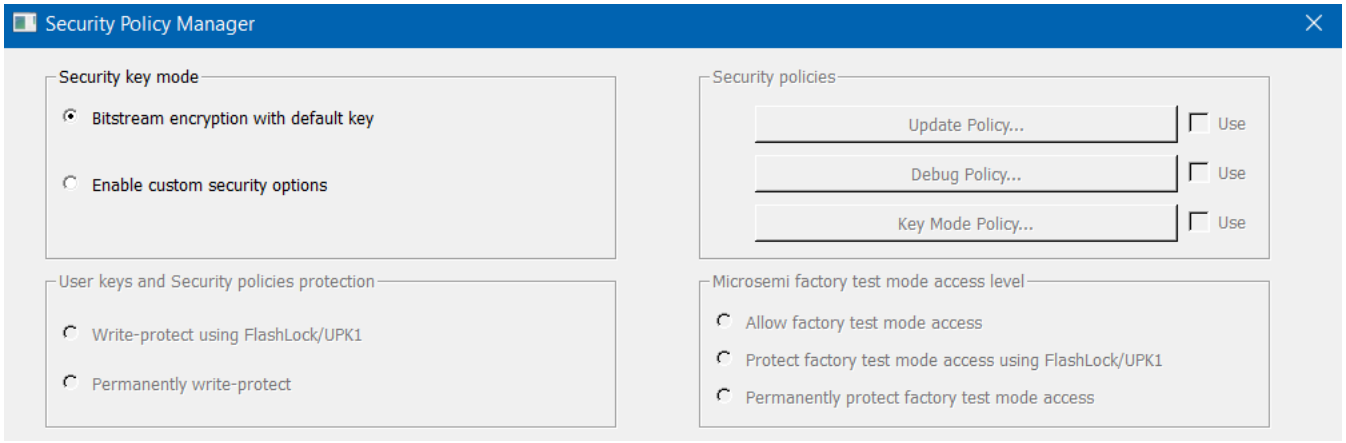
	Port Name	Macro Cell	Pin Number	/O State (Output Only
1	SW1	ADLIB:INBUF	H12	Z
2	SW2	ADLIB:INBUF	H13	Z
3	clk	ADLIB:INBUF	H16	Z
4	green_led1	ADLIB:OUTBUF	J16	Z
5	green_led2	ADLIB:TRIBUFF	M16	Z
6	red_led1	ADLIB:TRIBUFF	K16	Z
7	red_led2	ADLIB:OUTBUF	N16	Z
8	UNUSED	UNUSED	T11	Z
9	UNUSED	UNUSED	N10	Z
10	UNUSED	UNUSED	P10	Z
11	UNUSED	UNUSED	M10	Z

One step is to configure the I/O. Here we are configuring pins in JTAG mode (for debugging)

Generation



Allows us to have these different versions of the design and to determine a number of operation functions

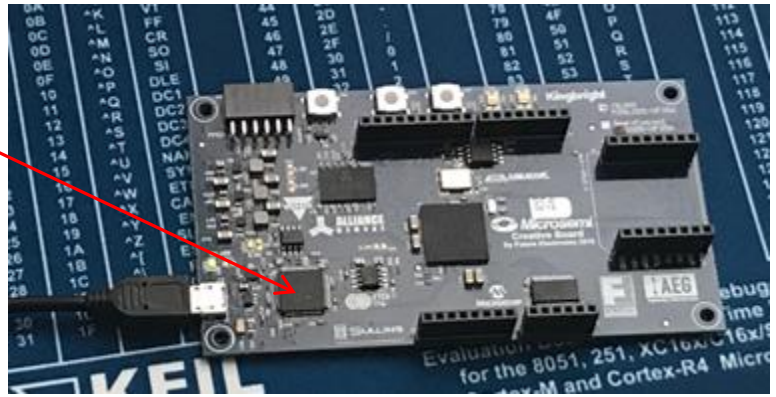


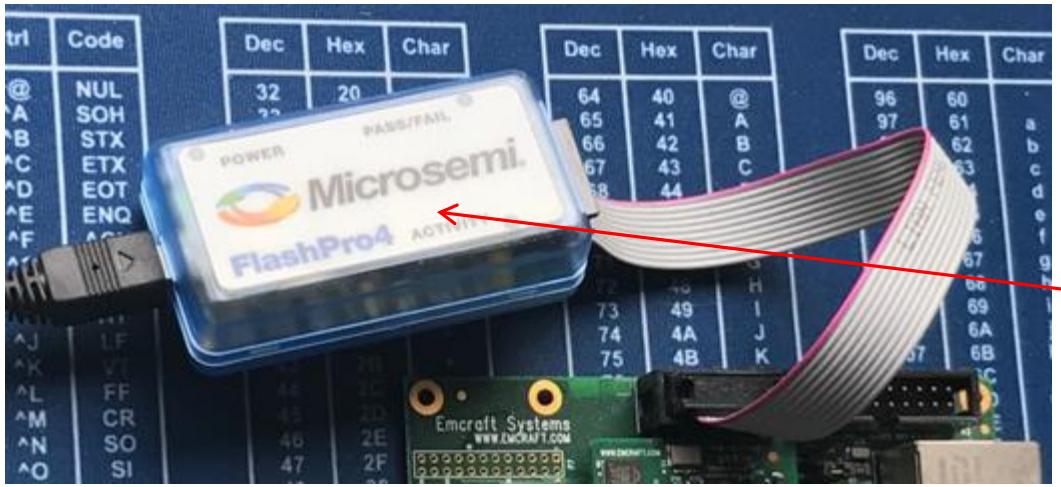
Programming the Device

- At this point tools are provided that will finally put the bitstream onto the device
- For development we need to use hardware that will load the bitstream onto the chip
  - This is not included in the chip itself
- We typically use a JTAG device which interfaces with the In-application programming functional block

Programming the Device

Current style JTAG
Is a chip on the
board



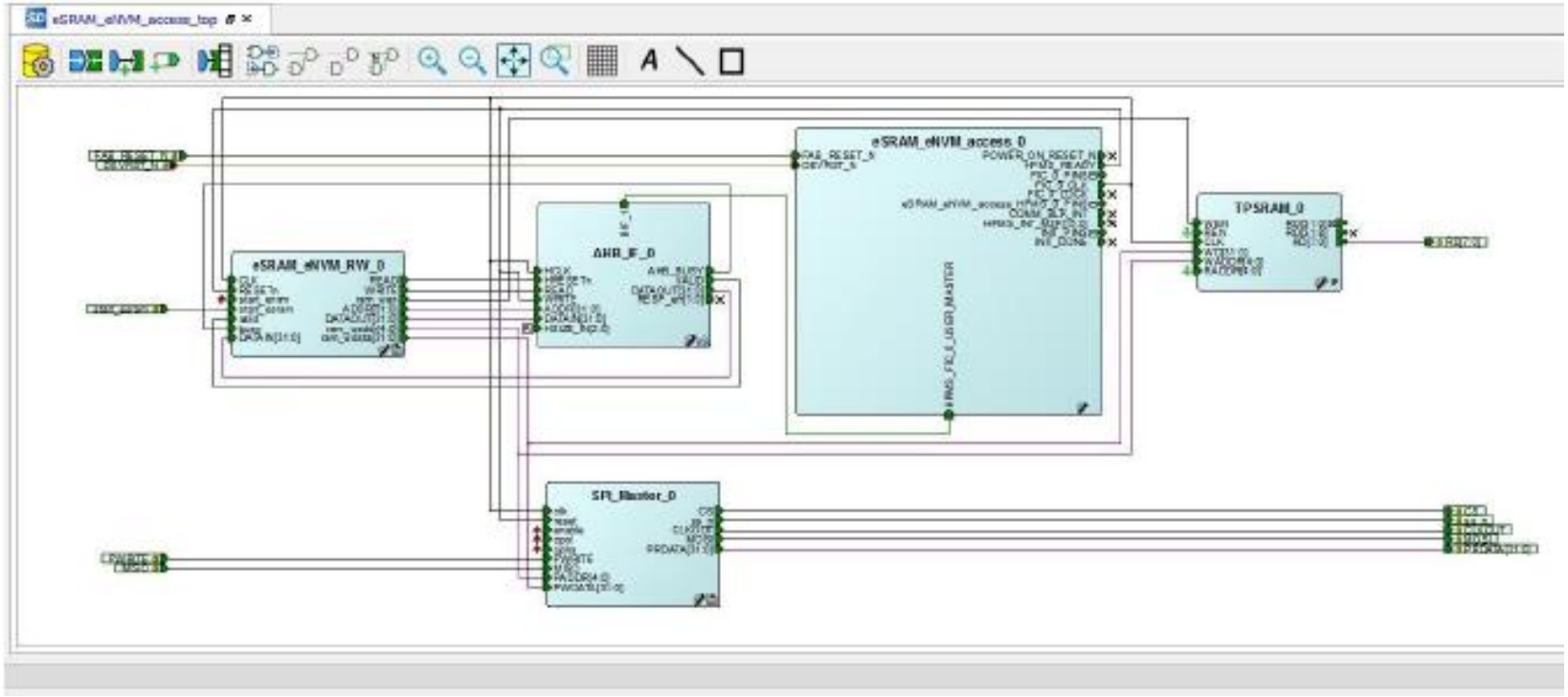


External JTAG

Debugging

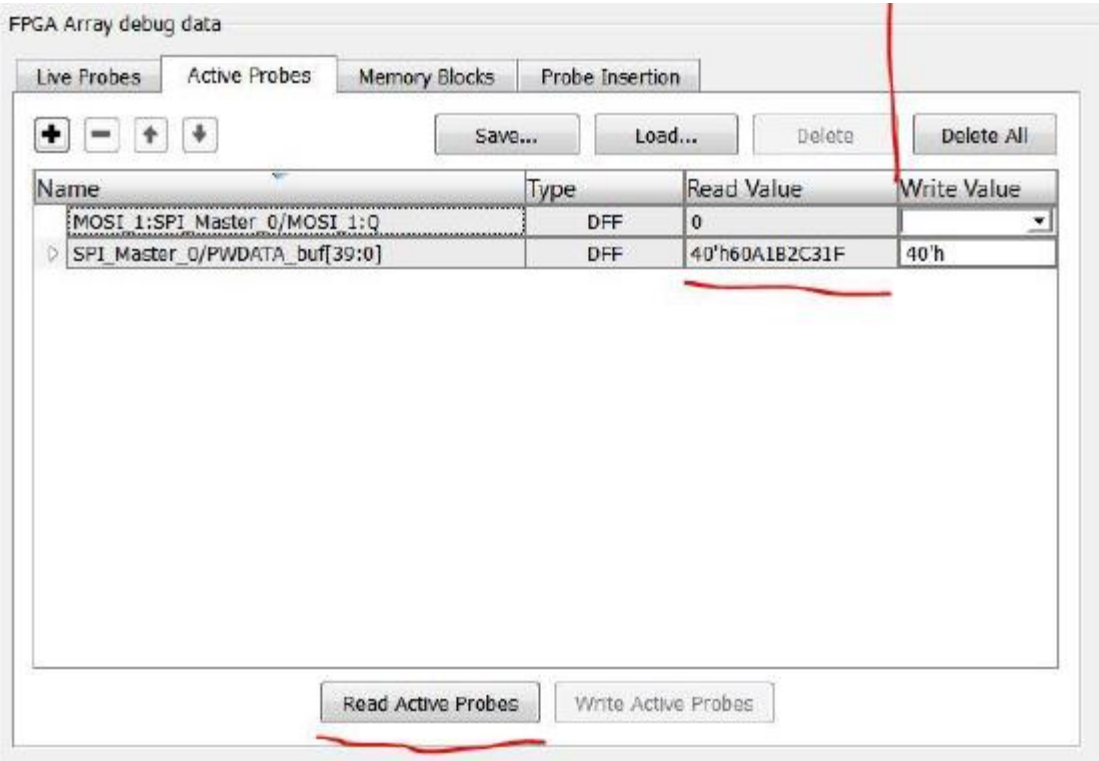
- We can debug the design right on the chip
- Much like regular software debugging we can set “breakpoints” called probes and inspect memory
  - This allows us to see the design in operation and to verify that we the design is operating correctly
  - If we find problems then we can make the necessary changes and regenerate

Debugging



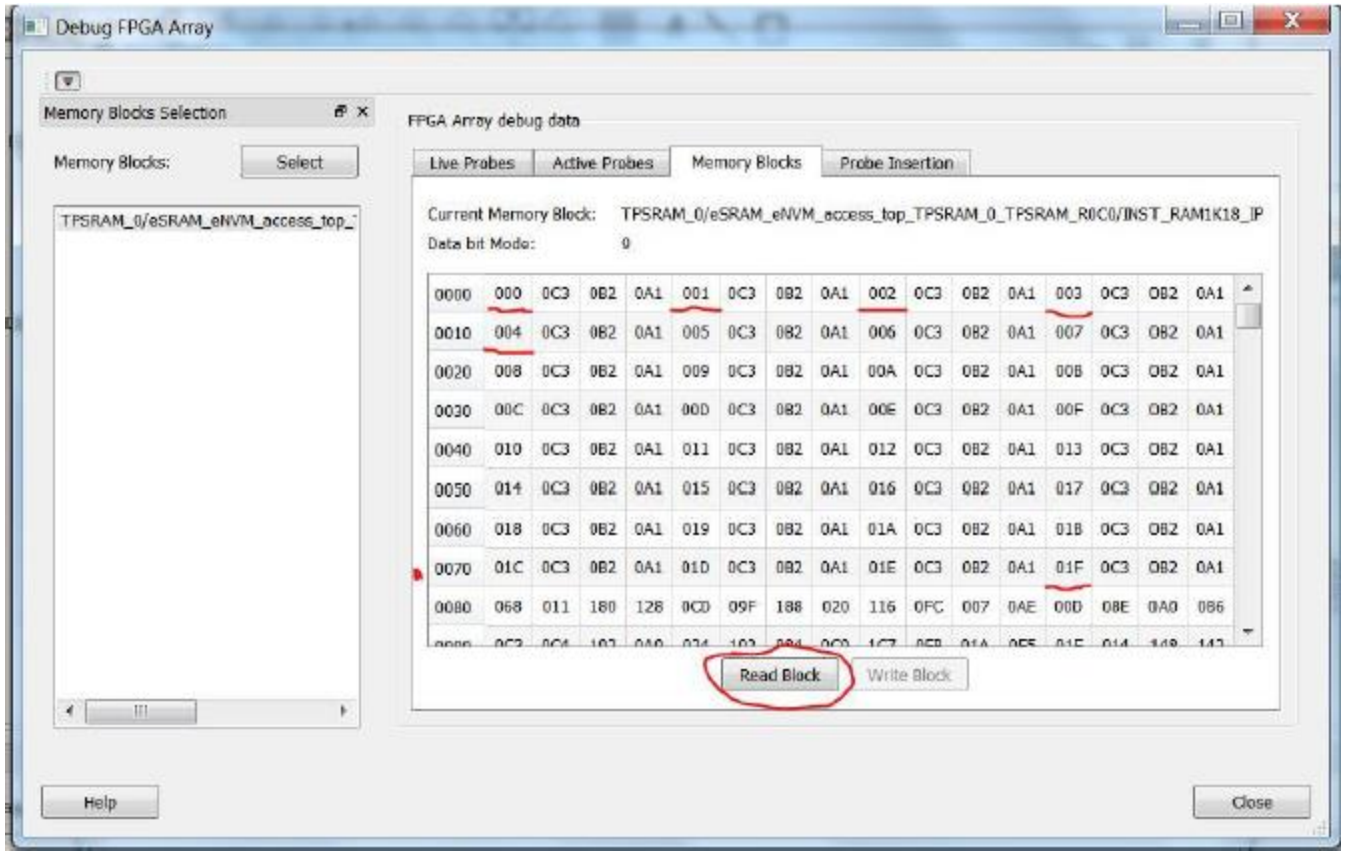
We will be looking at this design which manipulates memory blocks

Debugging



Here we have indicated specific elements we want to monitor.
As we step the chip through its cycle we will see values changing on those elements.

Debugging



We can inspect memory blocks in detail with the operation halted

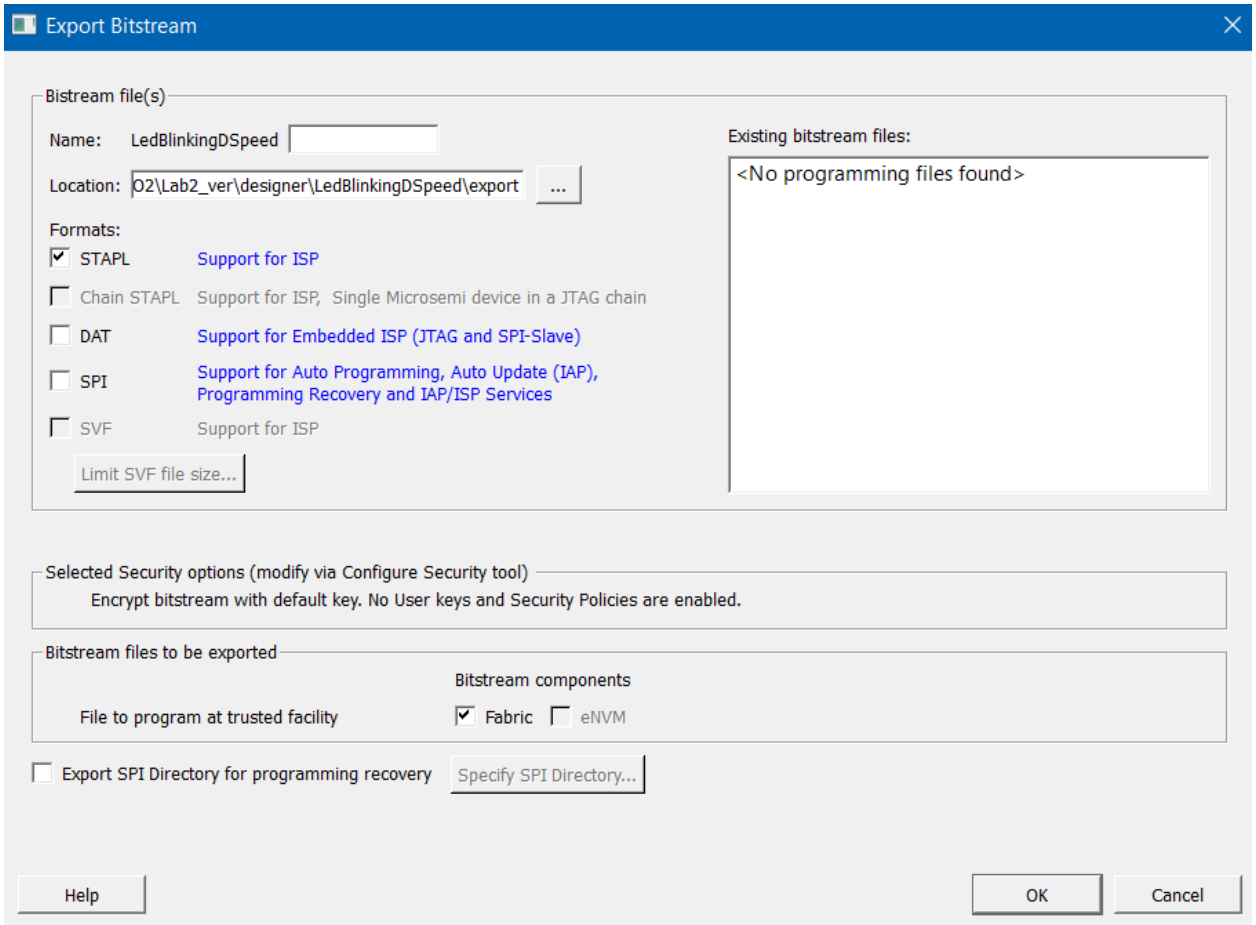
Production

- Once a design has been developed, simulated, debugged and verified it is ready for production
- Various production setups are available, and we will not go into the hardware here
- Suffice it say that we will not be programming large quantities of these units using a JTAG device attached to our laptops

Production

- The basic task is to package the bitstream for handoff to the production system
- This includes more that the bitstream
  - A job file for controlling the process of programming the device
    - FlashPro Express
    - Standard

Production



Options for bitstream export

Production

- Reports
  - Pin Report: assignment of all pins on the device
  - BSDL (Boundary Scan Description Language)
    - Used with VHDL
  - IBIS (Input Output Buffer Information Specification)
    - Allows one to hide the IP while providing information to users to understand the I/O required

Conclusion

- We have looked at the tools and processes available to create a bitstream and load it onto the chip
- This finishes the processes required to go from a design description to an operating FPGA
- We have touched on some of the debugging capabilities
- Finally we have discussed the process of getting the design out for production

Contacts/Resources

- I can be contacted several ways:
  - On Design News (naperlou, or search for Louis Giokas)
  - Email: l.giokas@ieee.org
  - Twitter: @naperlou for me or #DNCEC
  - Linkedin
- Resources
  - Vendor: websites have copious amounts of information available to prospective customers
  - Verilog Courses:
    - https://www.altera.com/support/training/course/ohdl1120.html
    - https://rtlacademy.com/verilog/overview
    - https://electronics.stackexchange.com/questions/26193/learning-verilog-online
    - http://www.multisoftvirtualacademy.com/embedded-systems/verilog-online-training