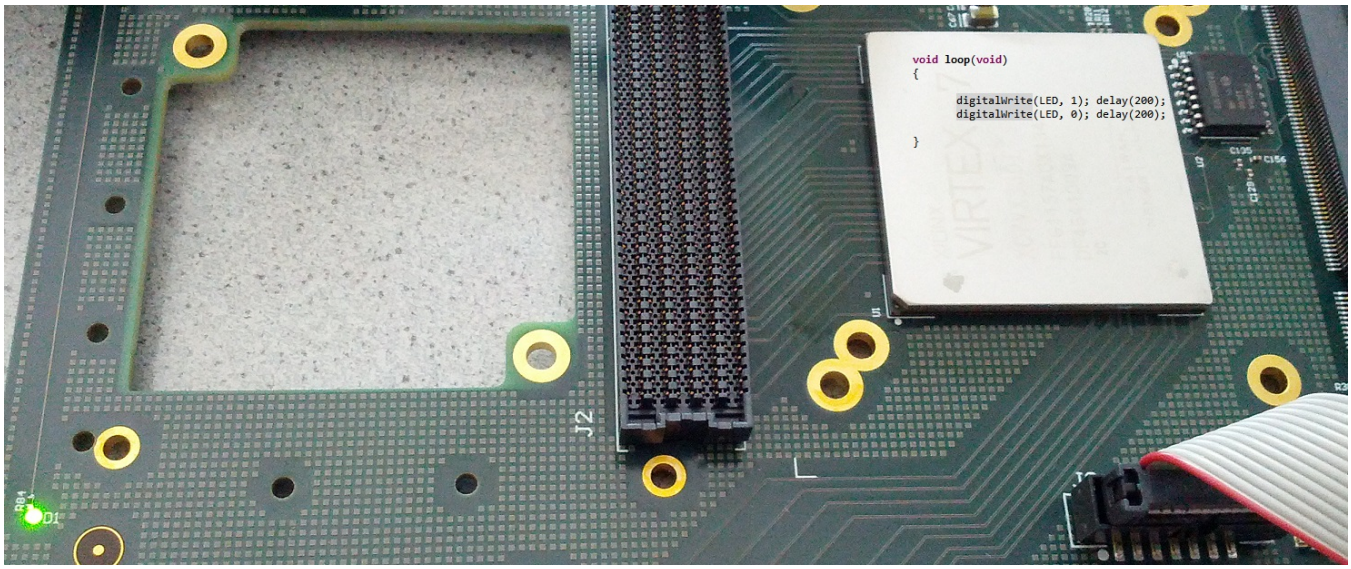
Arduino with VIRTEX-7

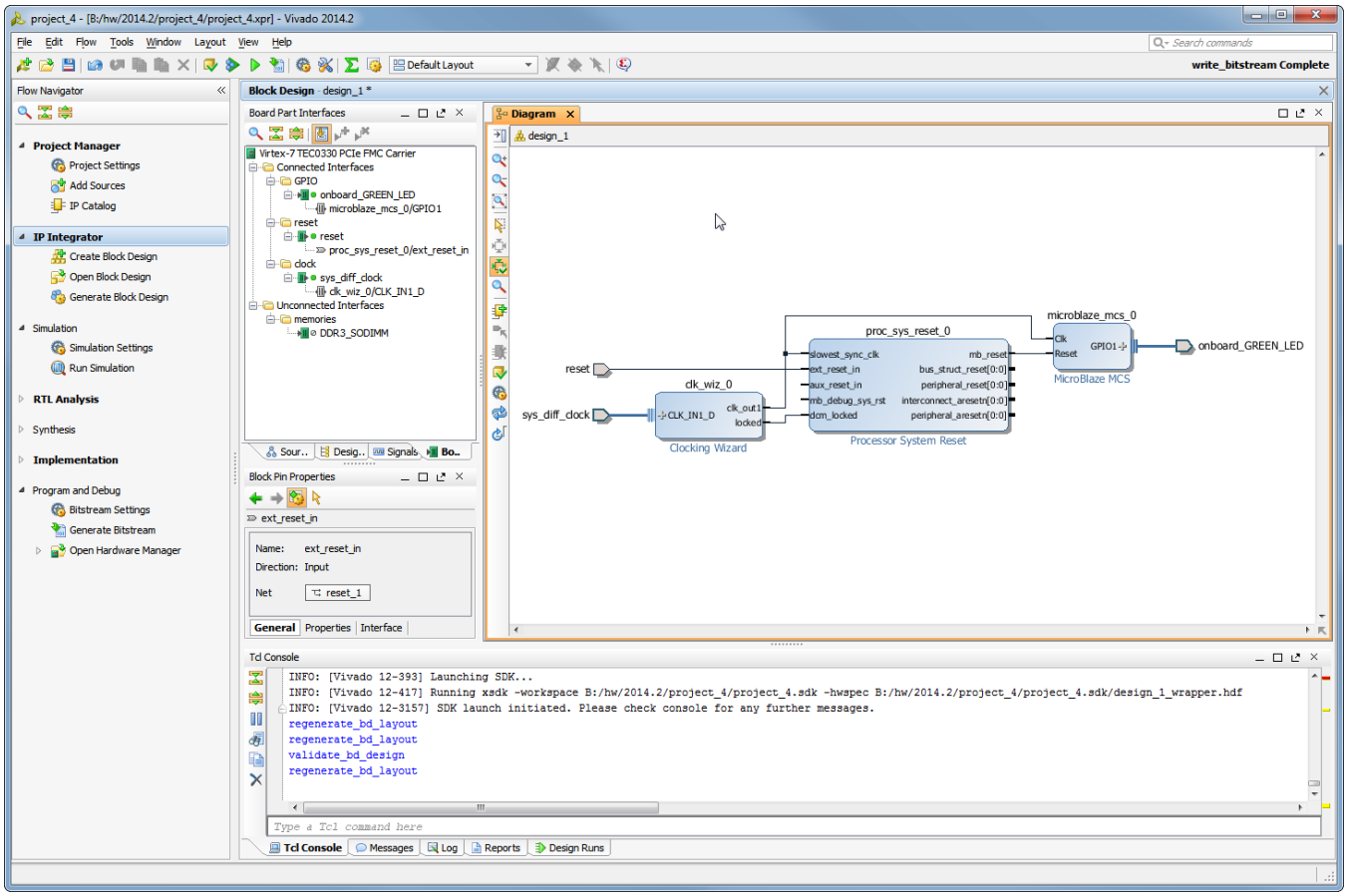


Why should anyone want to run Arduino code on Virtex-7 ever you may ask? Let's look the prices first: as of current listing XC7VX330T-2FFG1157C costs 3204 USD or 2366 EUR.

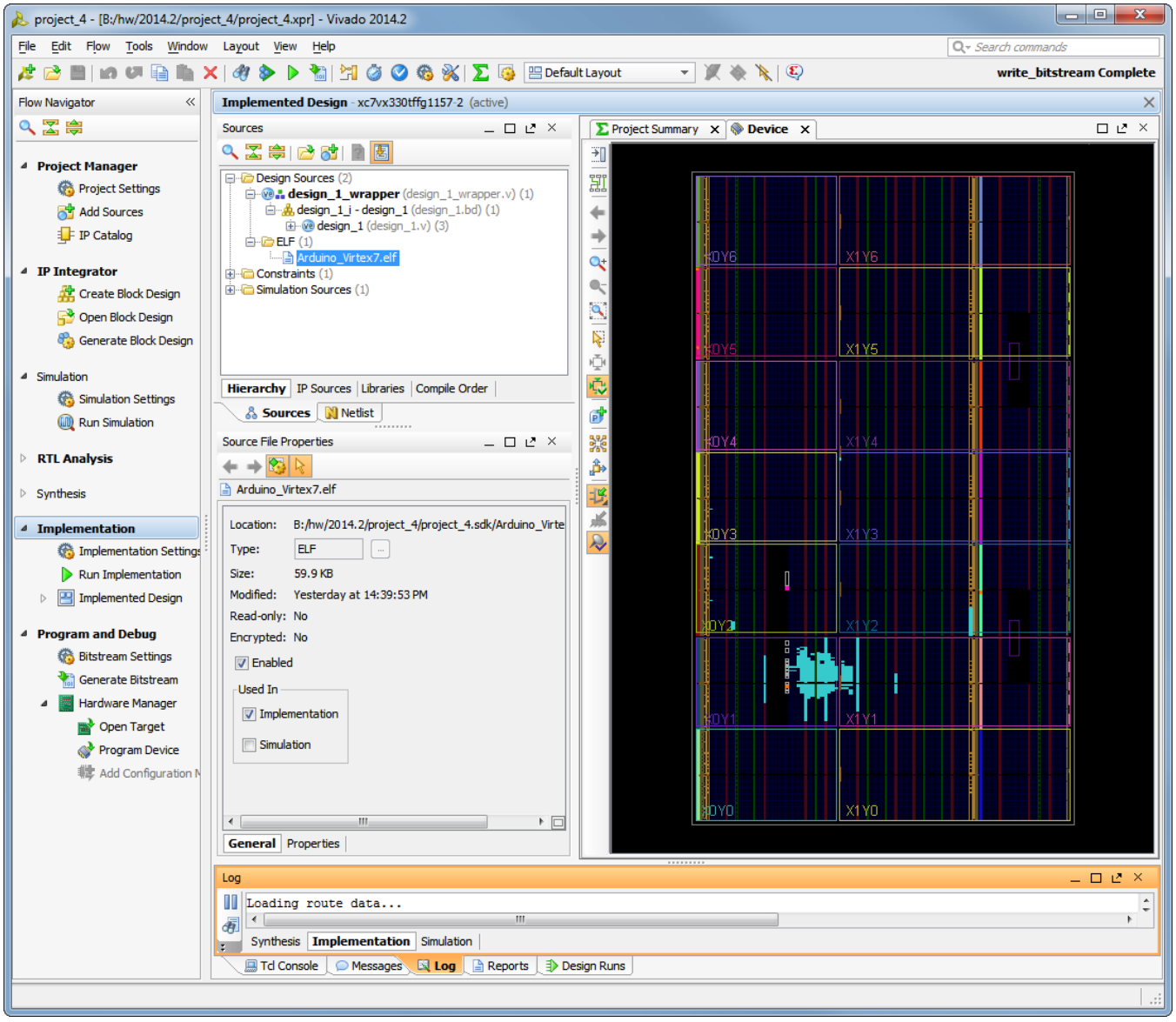
On my desk there is a PCB board that did come from our Vapor Phase oven yesterday. On the board there is a LED. This LED blinks, it says hello, I am alive. There is a chip on the board, and on that it reads: XC7VX330T-2FFG1157C

Lets look closer how the LED was made to Blink. Why it makes sense I tell another day.

Step 1: Hardware Design



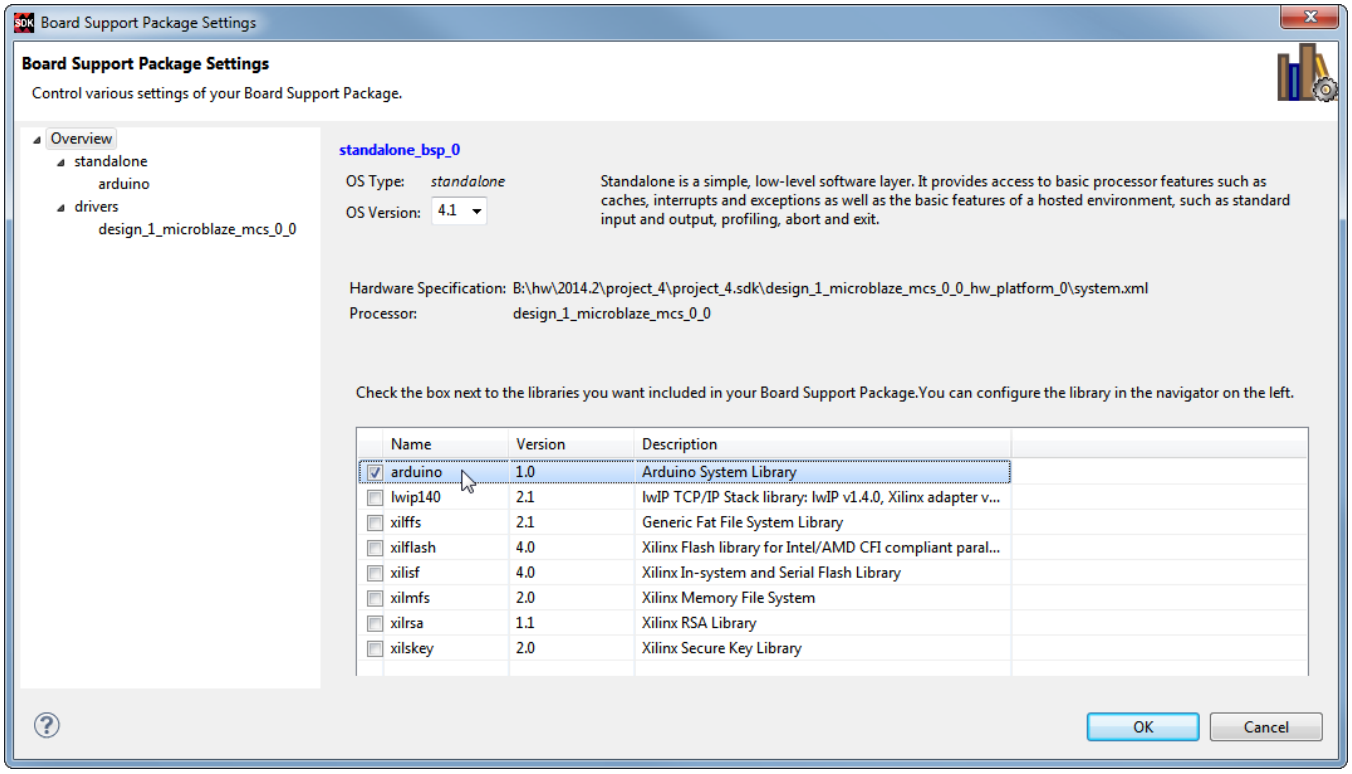
I have dropped Xilinx Microblaze MCS onto the Block design. all the Interface ports to the real hardware (clock, reset and LED) are connected automatically using Vivado Board Part Interface flow. So for making this hardware design, there is nothing to look in any documents, just select the Board to work with, and go. All constraints are managed by Vivado. We can now just run generate bitstream.



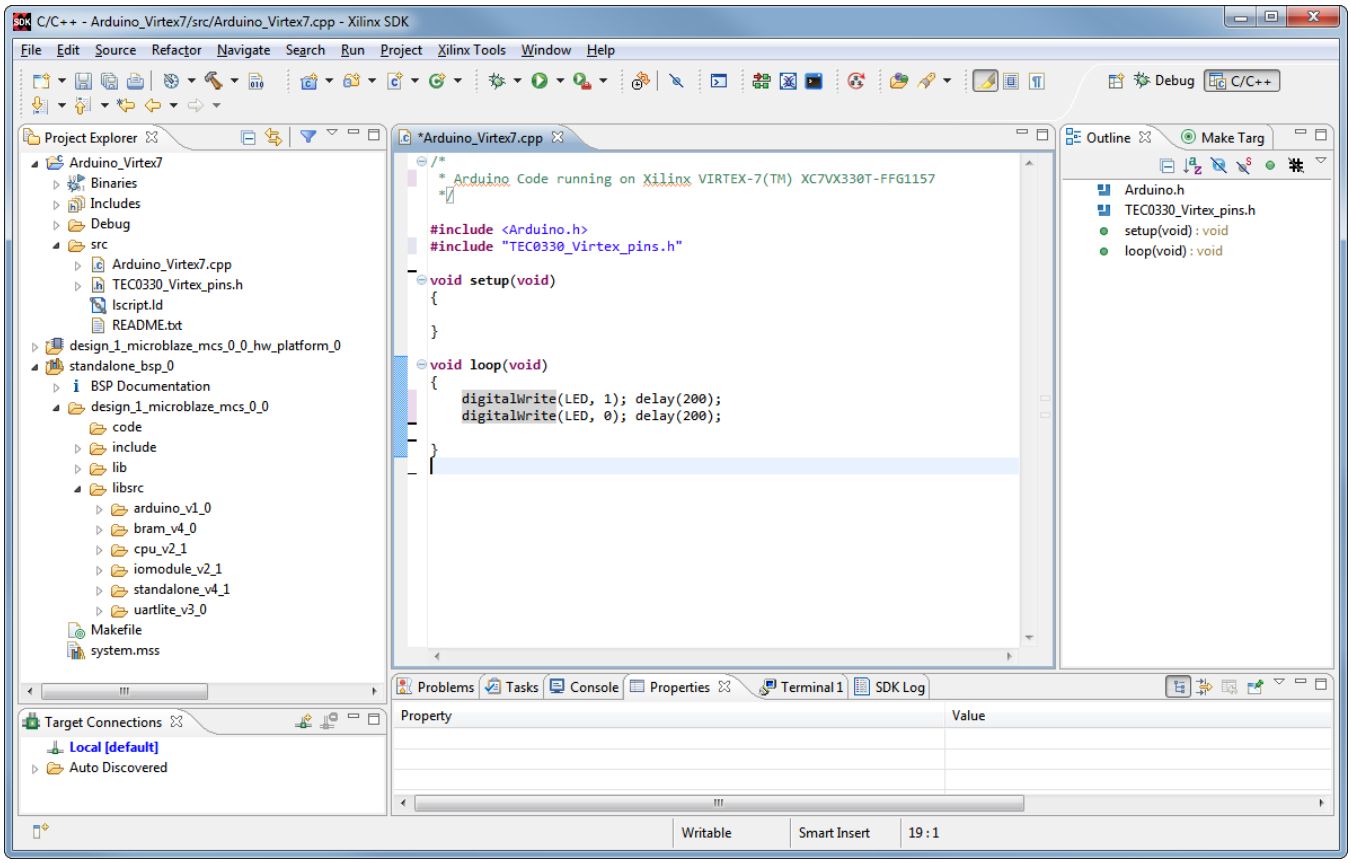
Design is implemented, we go to software design flow now.

Step 2: From Hardware to Software

As soon as Vivado has generate the output products and bitstream we can move from Hardware design to Software. In Vivado Export and Launch SDK. We create new BSP and select Arduino System Library to be included.



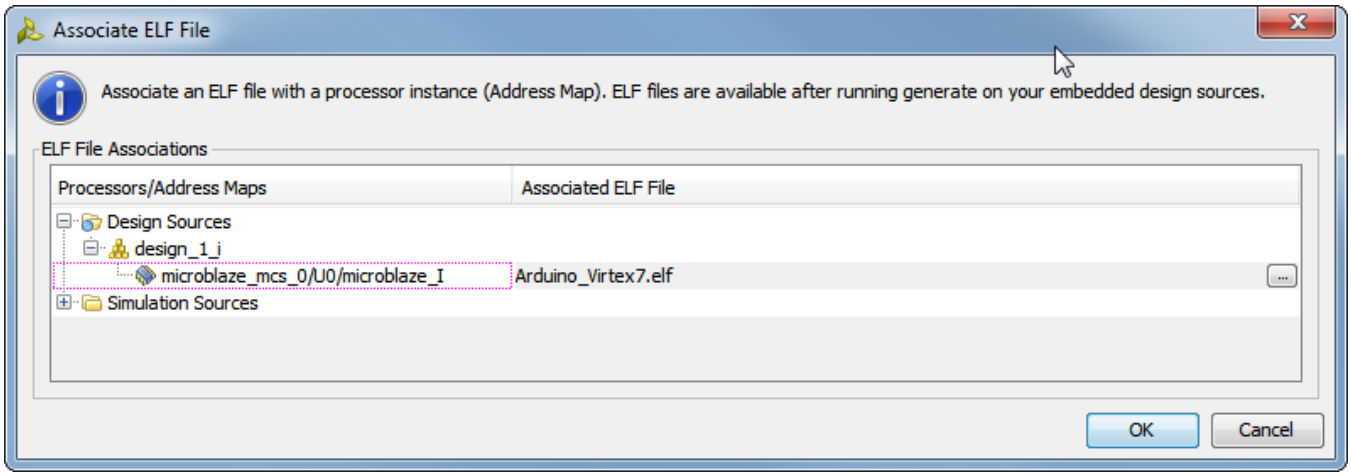
Now we are ready to write some Arduino Code! A LED Blinky would do for now.



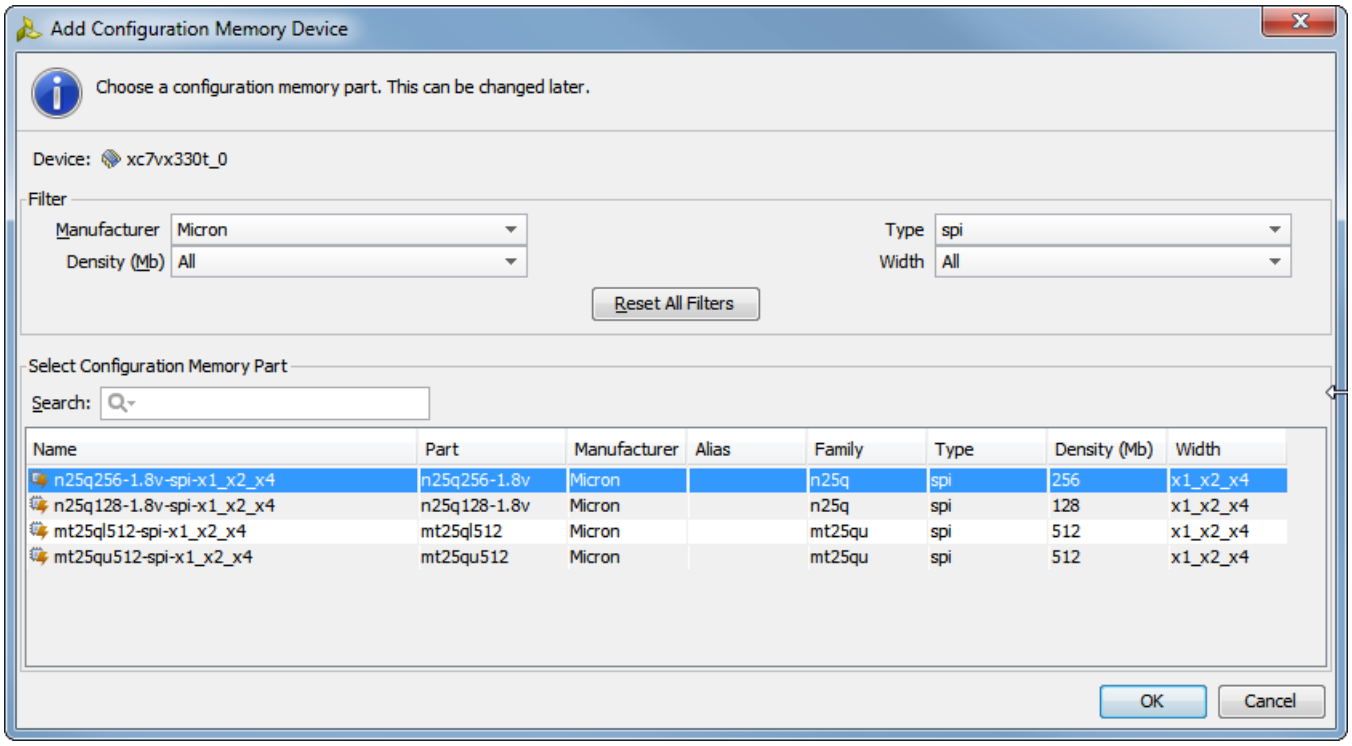
Done, we select debug-as and start in Debugger. LED Blinks!

Flash Programming

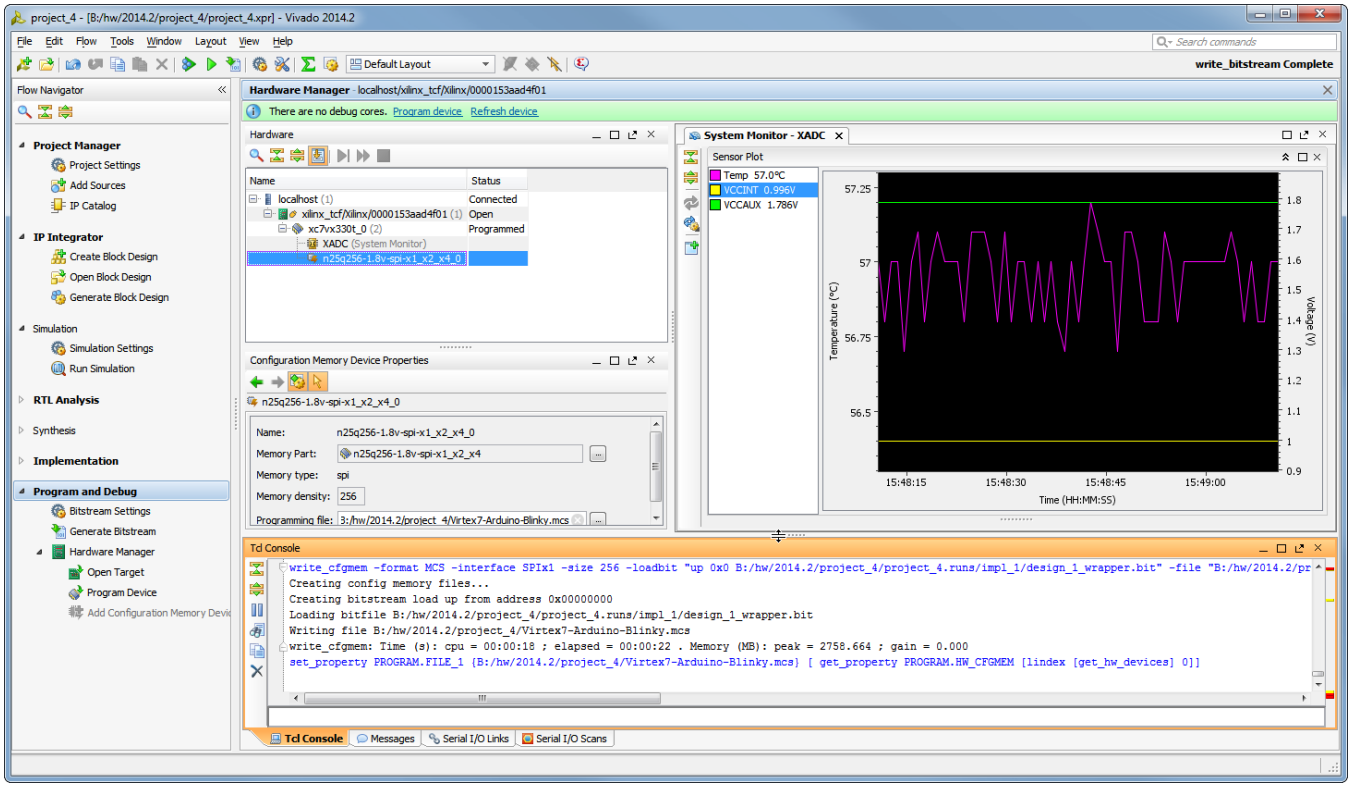
How about Programming the Arduino code into the Flash memory so that it starts when FPGA loads? First we have to tell Vivado what Software file we want to use, so we associate the proper ELF File.



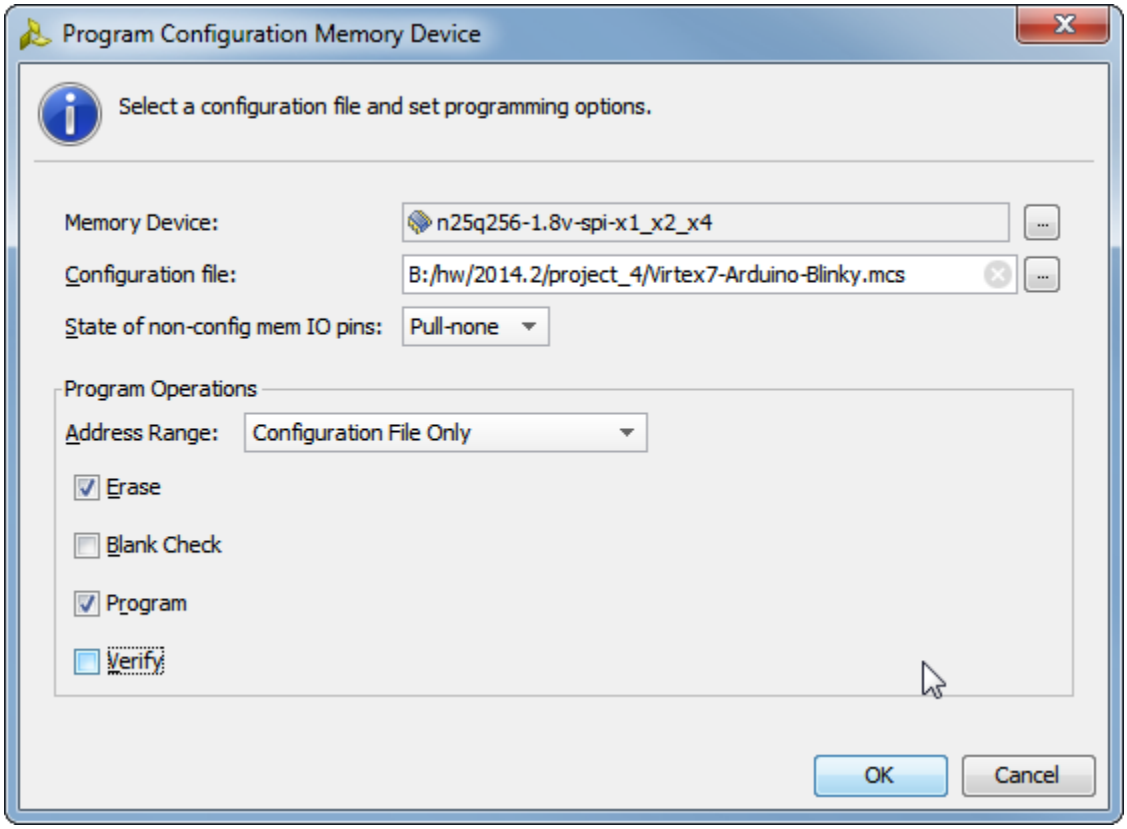
Now we tell Vivado what Flash Memory is used on our board:



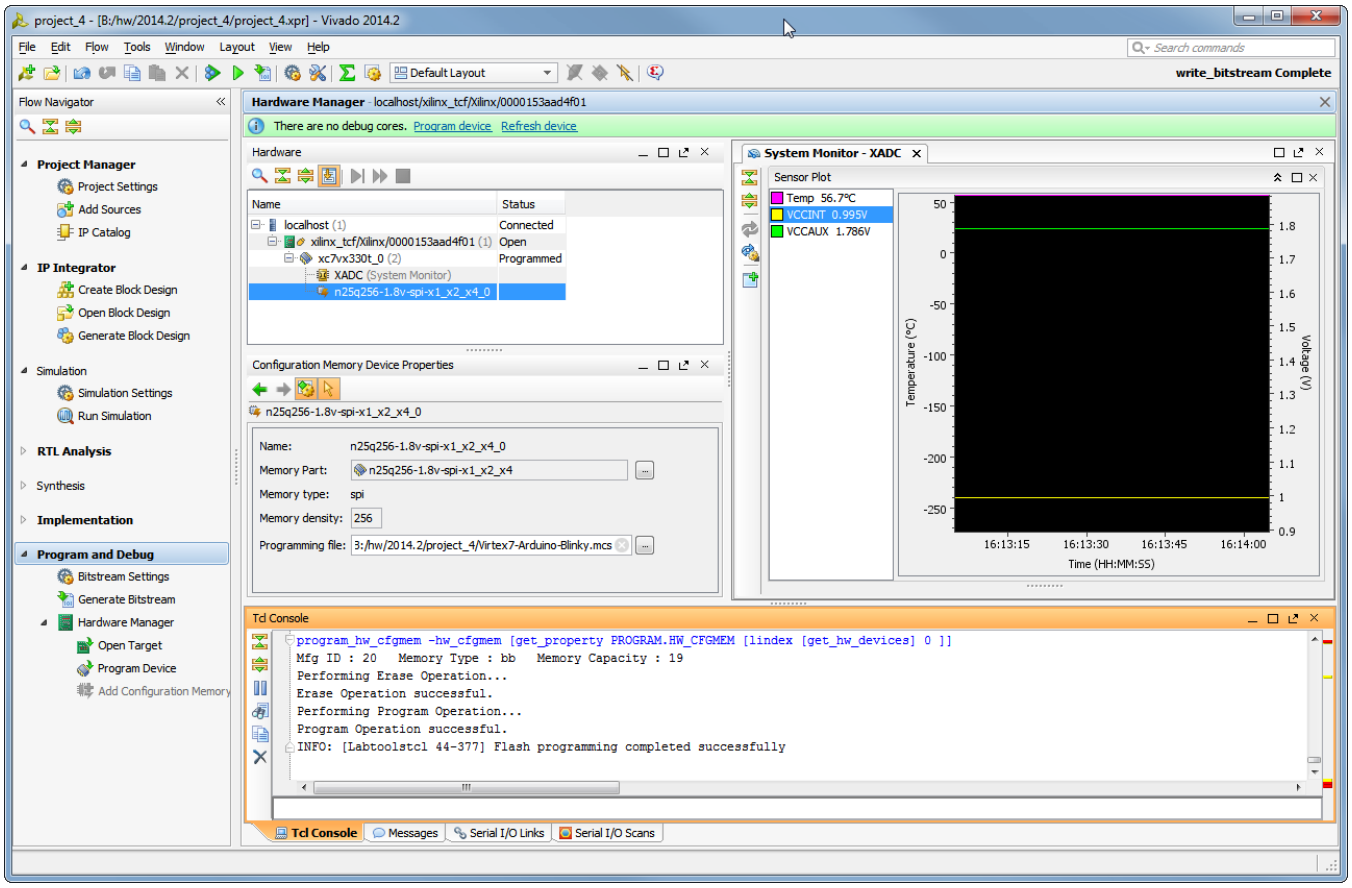
And ready we are to write the Flash memory. Almost.. there is one step still required that currently is not available from the menus. So single line of TCL code should be executed.



So the BIT file is converted to MCS and ready for flashing!



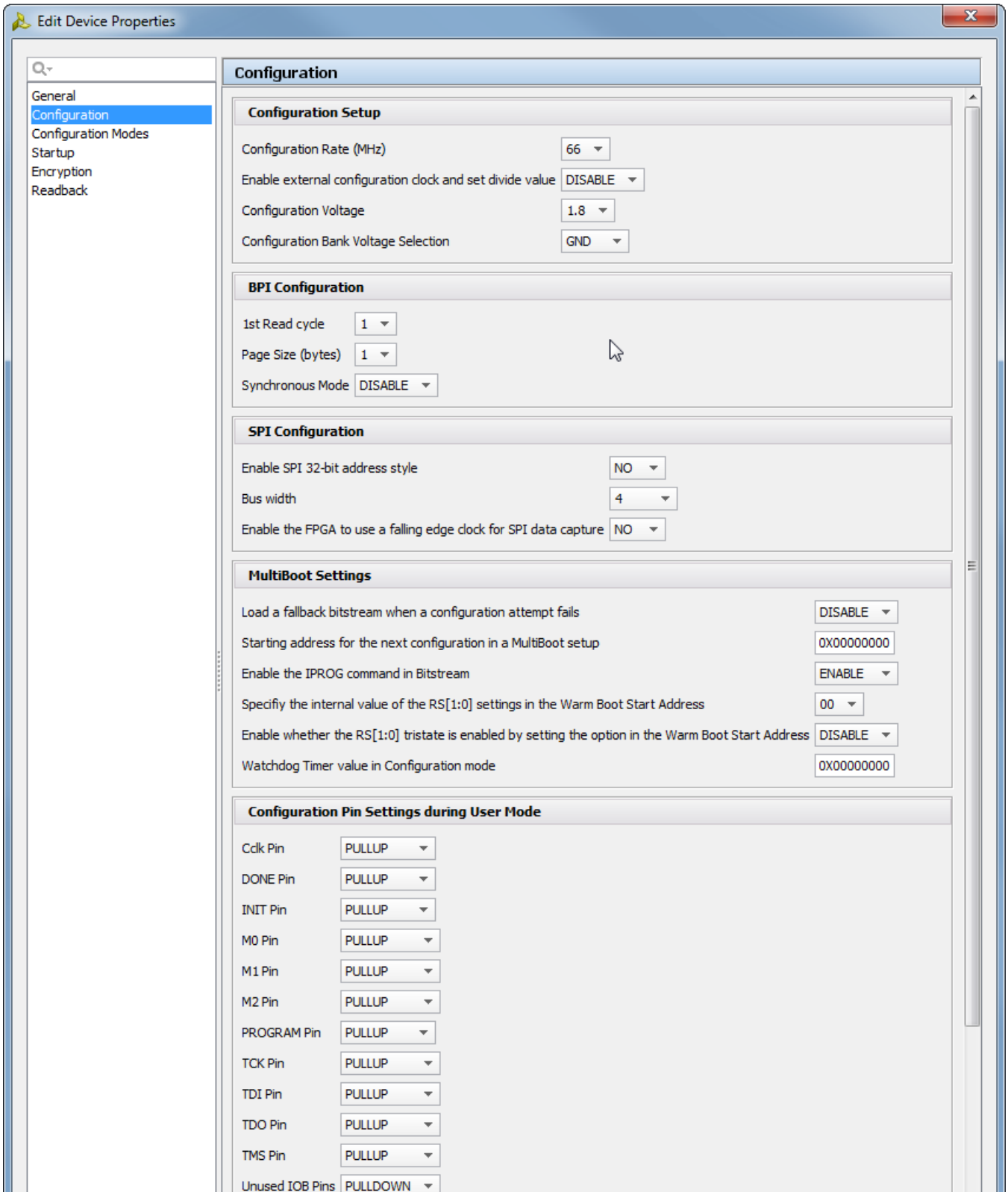
Click OK to Program..

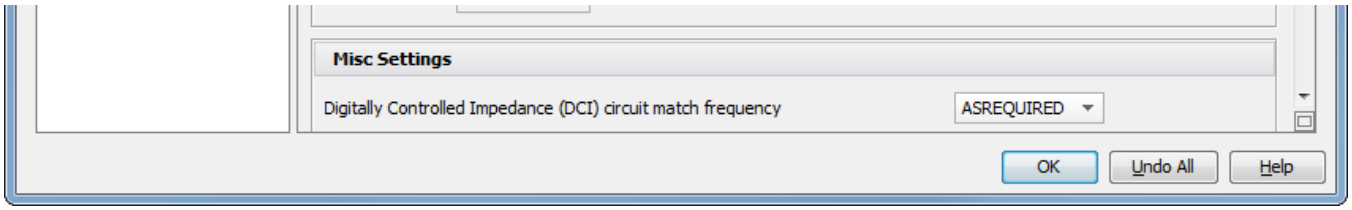


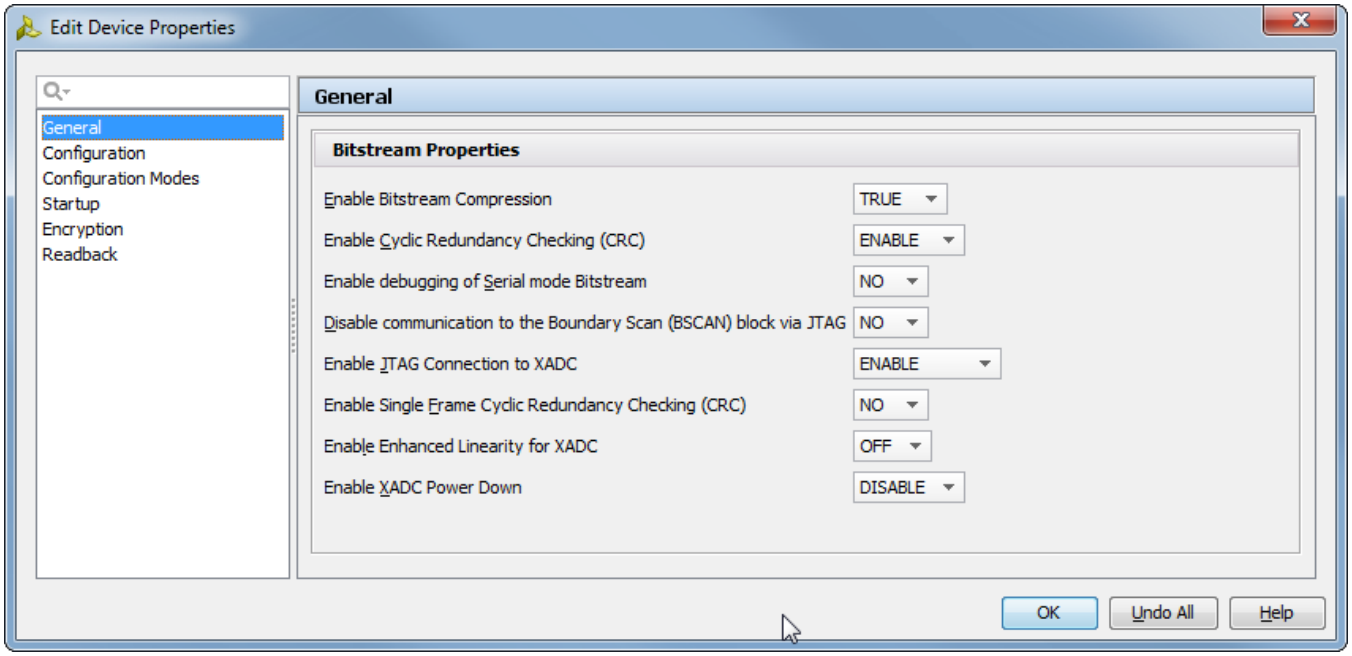
And ready we are. Now lets try power cycle - power off, power on.. and LED Blinks again.

Xilinx Virtex-7 has configured itself with Arduino Code and Says "Blink..Blink" to all of us.

And what did I miss in the process? I forgot to set the bitstream options, so it did take painfully long (About 1 minute to load the bitstream!) to start after cold restart. So here some screenshots of the property settings:


Edit Device Properties
General
Configuration
Configuration Modes
Startup
Encryption
Readback
Configuration
Configuration Setup
Configuration Rate (MHz) 66
Enable external configuration clock and set divide value DISABLE
Configuration Voltage 1.8
Configuration Bank Voltage Selection GND
BPI Configuration
1st Read cycle 1
Page Size (bytes) 1
Synchronous Mode DISABLE
SPI Configuration
Enable SPI 32-bit address style NO
Bus width 4
Enable the FPGA to use a falling edge clock for SPI data capture NO
MultiBoot Settings
Load a fallback bitstream when a configuration attempt fails DISABLE
Starting address for the next configuration in a MultiBoot setup 0X00000000
Enable the IPROG command in Bitstream ENABLE
Specifiy the internal value of the RS[1:0] settings in the Warm Boot Start Address 00
Enable whether the RS[1:0] tristate is enabled by setting the option in the Warm Boot Start Address DISABLE
Watchdog Timer value in Configuration mode 0X00000000
Configuration Pin Settings during User Mode
Cclk Pin PULLUP
DONE Pin PULLUP
INIT Pin PULLUP
M0 Pin PULLUP
M1 Pin PULLUP
M2 Pin PULLUP
PROGRAM Pin PULLUP
TCK Pin PULLUP
TDI Pin PULLUP
TDO Pin PULLUP
TMS Pin PULLUP
Unused IOB Pins PULLDOWN





A small hint: those properties are not always accessible, so if you can not find them, just run the flow and open implemented design.