Spartan-6 MicroBoard Demo

- This document will guide you through compiling and testing the provided mcx16 demonstration code on the Avnet Spartan-6 LX9 MicroBoard. The code was tested using Xilinx ISE 13.2.
- The MicroBoard is inexpensive ($89 when I bought one), but even if you don't have one you can still use the provided sample code to experiment with synthesizing the mcx16 microcontroller.

 © 2011 Gregory Bredthauer

mcx16 Introduction

mcx16 is a 16-bit microcontroller written in VHDL. It was inspired by PicoBlaze, an 8-bit microcontroller designed by Ken Chapman at Xilinx. mcx16 uses an instruction set and architecture similar to PicoBlaze, so if you've already learned one, using the other should be straight-forward. The mcx16 code is tested on Xilinx FPGAs.

Some highlights of mcx16:

- 16-bit addresses, data, port addresses and ports
- Can use 1 (non-pipelined) or 2 (pipelined) clocks per instruction
- 74MHz (74 MIPS) on a Spartan 3 XC3S200AN-4 non-pipelined, and 119 MHz when pipelined (59 MIPS), using 110/114 slices
  (area optimized, 16 registers, 32 deep stack, 512 word program, distributed stack)
- 107 MHz (107 MIPS) on a Spartan 6 XC6SLX9-2 non-pipelined, and 166 MHz when pipelined (83 MIPS), using 46/41 slices
- Pipelining, register count (1 – 256), stack depth (1 – 65536), and program/stack type (BRAM/distributed) are all user-set generics; program size up to 65536 words
- Cross-platform assembler
- Optional JTAG loader using Kris Chaplin's technique and cross-platform Xilinx tools

 © 2011 Gregory Bredthauer

Source Files

- mcx16s6mb.xise: Project file
- mcx16s6mb.vhd: Top level VHDL
- mcx16s6mb.ucf: Pin and clock constraints
- mcx16.vhd: Microcontroller VHDL
- mcx16s6mb_rom.txt: Program ROM source
- mcx16s6mb_rom.vhd: Program ROM VHDL
- mcx16loader.vhd: JTAG loader VHDL
- mcx16uart.vhd: UART VHDL
- mcx16util.vhd: Shared utility VHDL

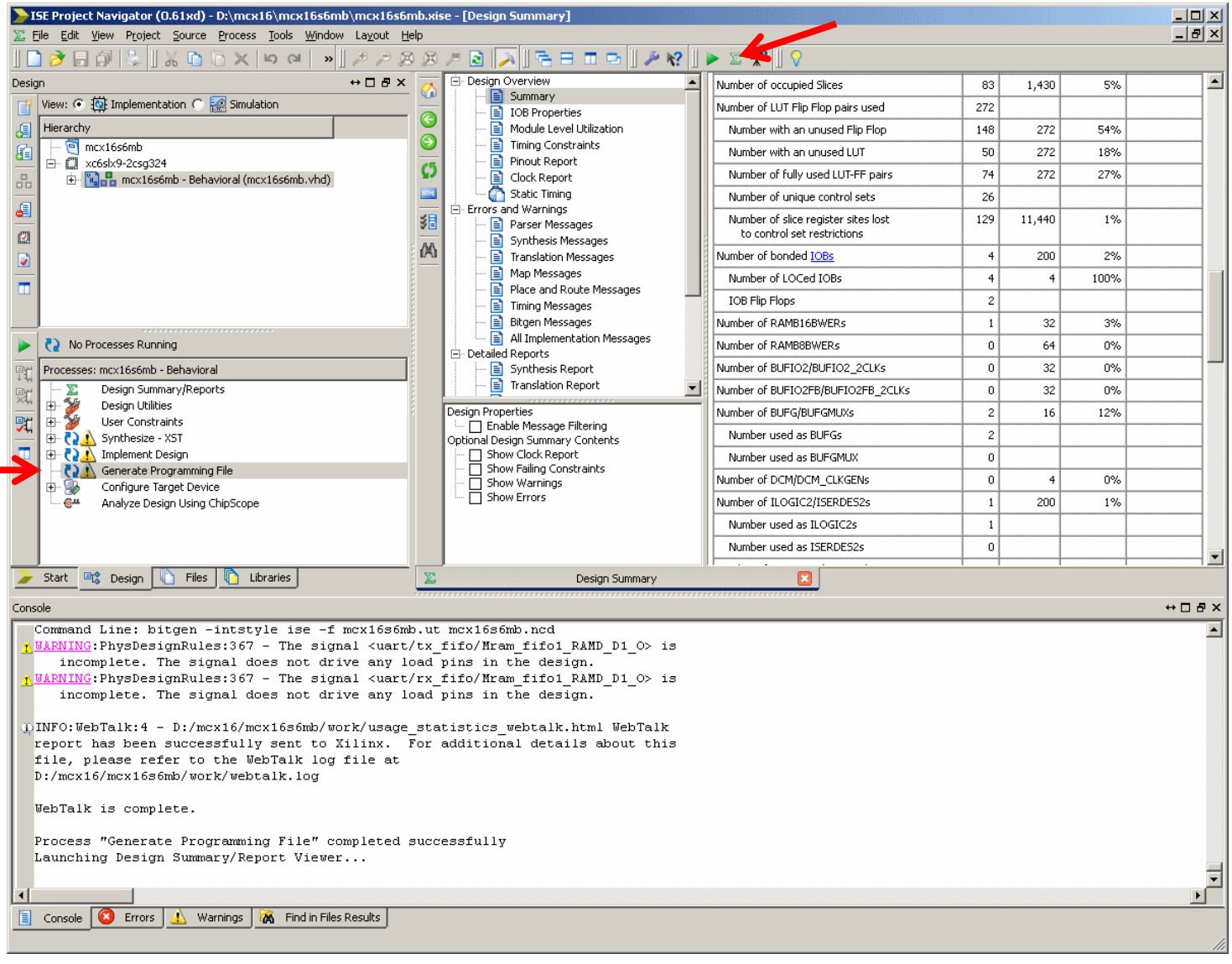
 © 2011 Gregory Bredthauer

Setup

- Extract the mcx16s6mb.zip file into a directory
- Open the mcx16s6mb.xise project file with ISE
- Let it create a “work” directory if prompted
- Double-click “Generate Programming File”
- This should complete without errors
- Click the “Design Summary” button to inspect the design’s resource usage

 © 2011 Gregory Bredthauer

Compile Project

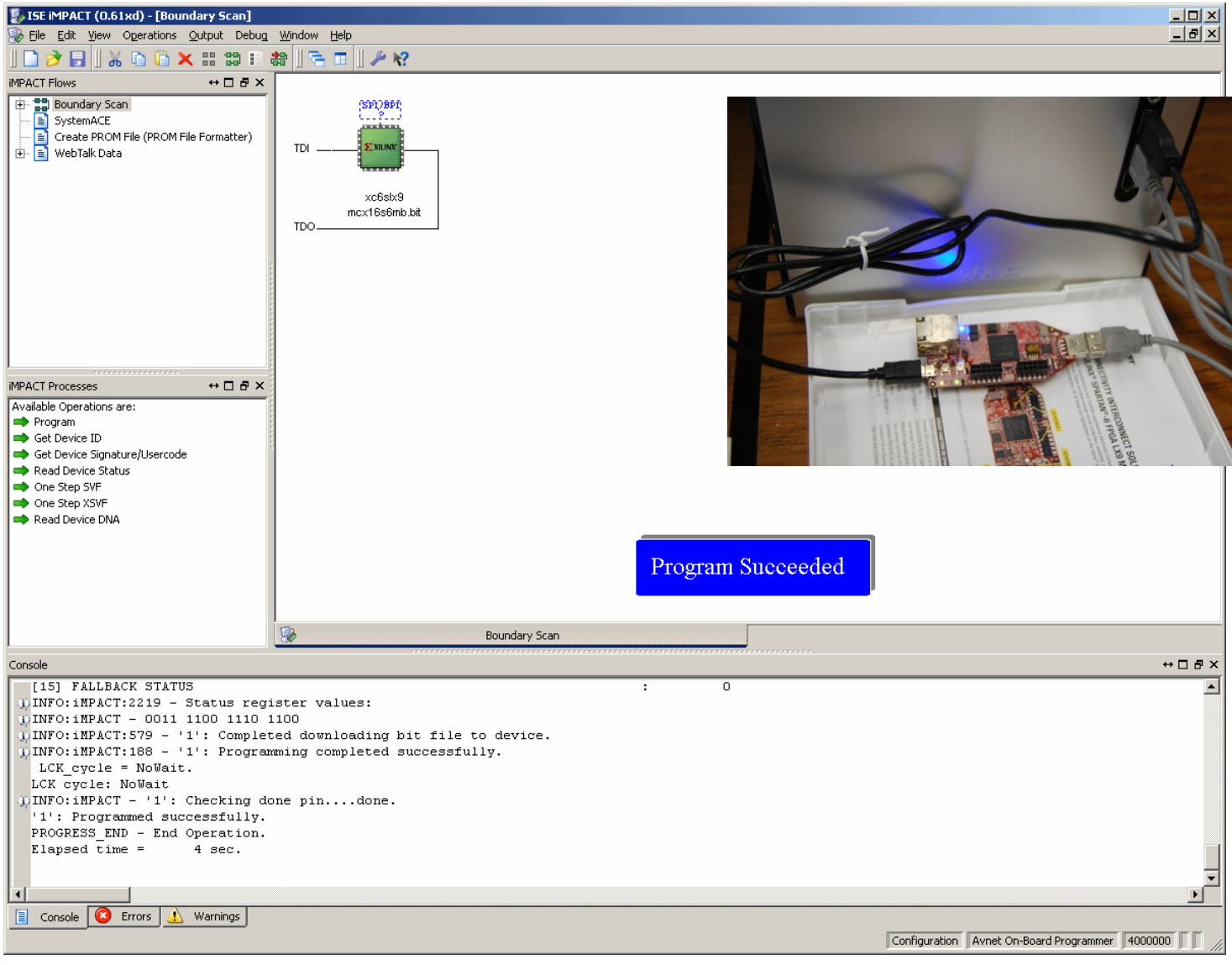


Launch iMPACT

- It is assumed that you've already installed the Digilent JTAG drivers for the MicroBoard, along with the Silicon Labs CP2102 serial port drivers (these steps are described in detail in the MicroBoard documentation)
- With the MicroBoard connected to your computer (both through the microUSB connector for serial and through the full USB connector for JTAG), launch iMPACT, create a new project, and assign the work/mcx16s6mb.bit file to the FPGA

 © 2011 Gregory Bredthauer

Assign bit file



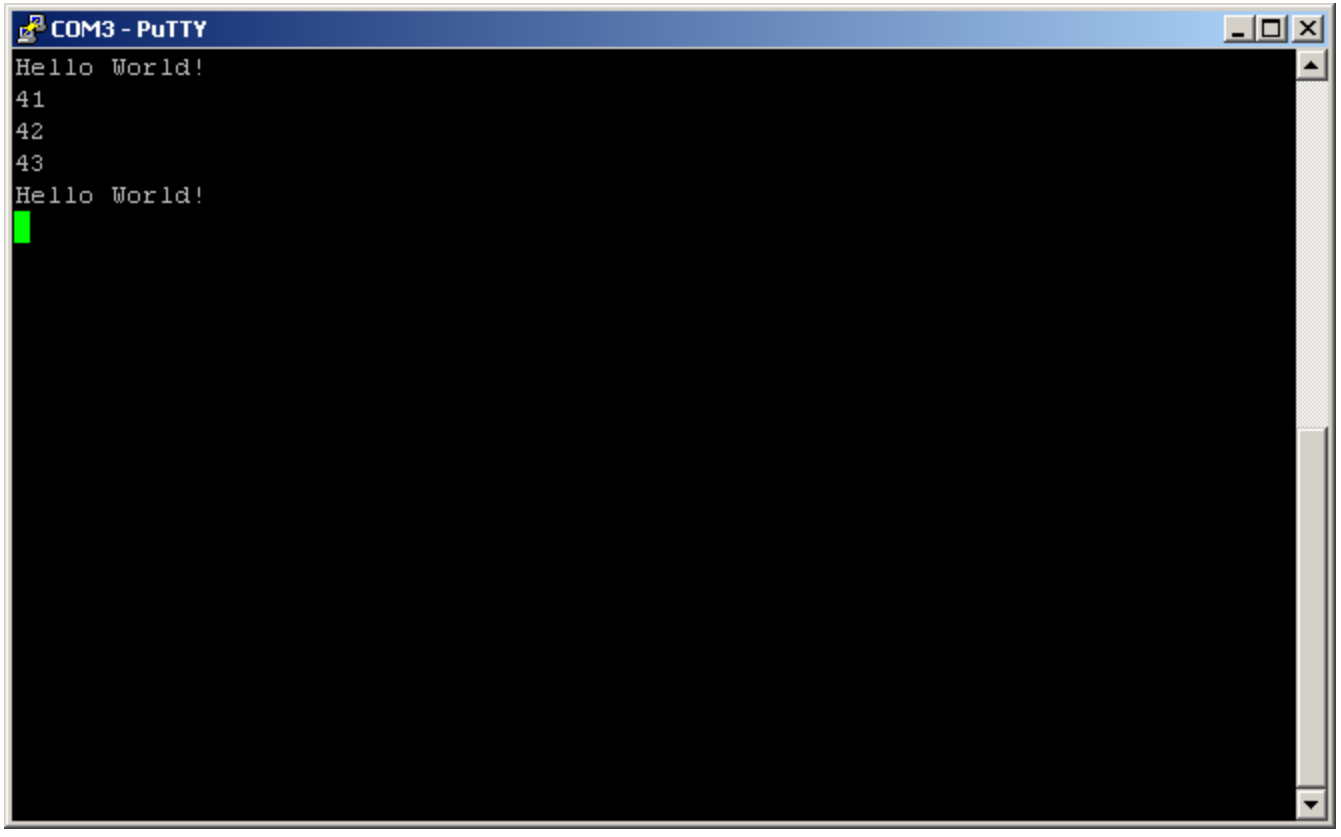
 © 2011 Gregory Bredthauer

Program and Test

- Open a serial terminal to the MicroBoard (using HyperTerm, PuTTY, etc, 115200 N81)
- Program the MicroBoard with the bit file
- You should see “Hello World!” after programming, and every time you press the reset switch on the MicroBoard (SW5)
- When you type into the terminal, it should respond with the two digit hex codes of the characters you typed (programmed, typed “ABC” and then pressed SW5 in the following screen)

 © 2011 Gregory Bredthauer

Terminal Output



 © 2011 Gregory Bredthauer

Experiment

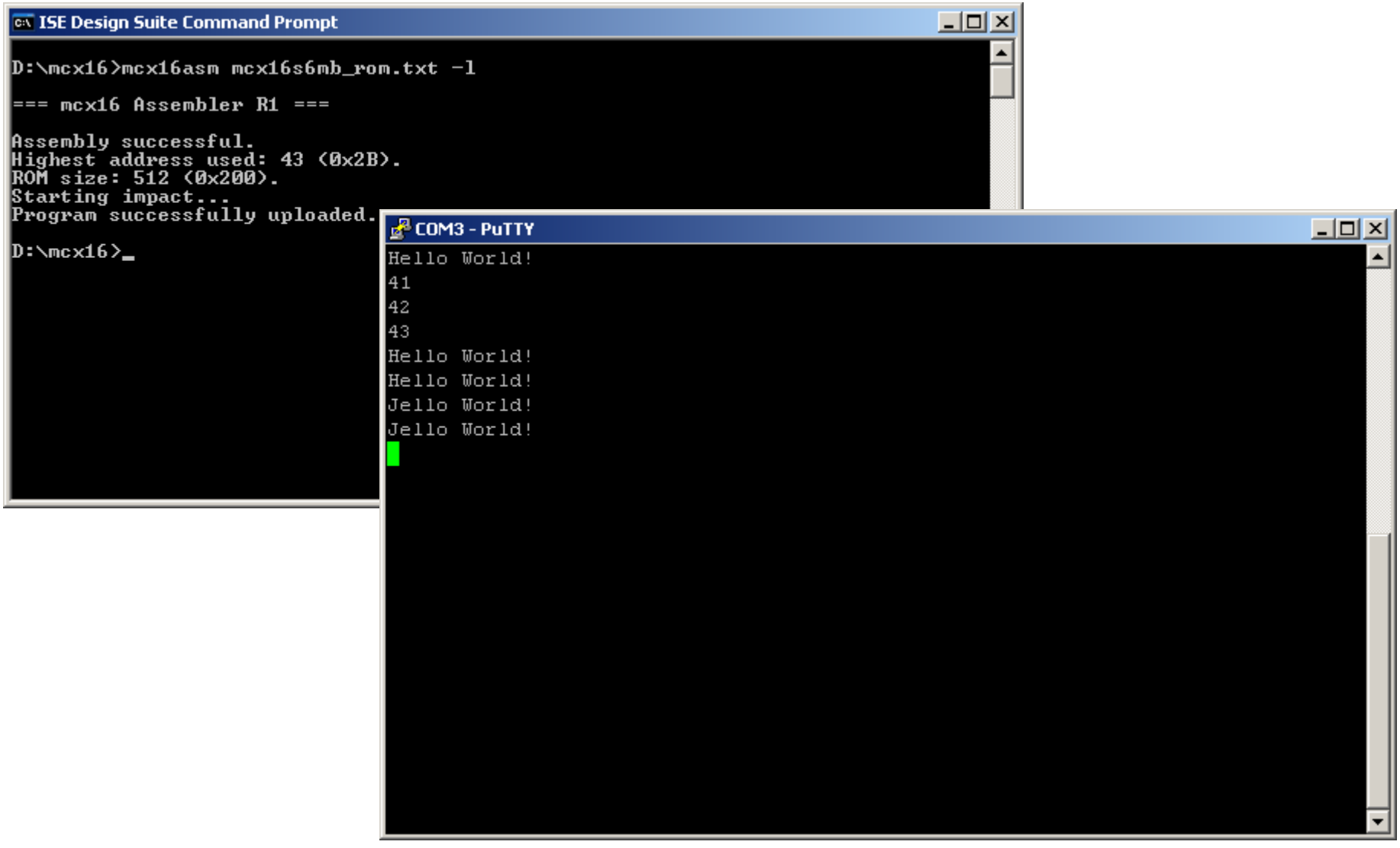
- Download the mcx16 files (which includes the assembler) into a directory, and copy the "mcx16s6mb_rom.txt" file into the same directory
- Launch the "ISE Design Suite Command Prompt" and navigate to the directory with the mcx16 files
- Edit the "mcx16s6mb_rom.txt" file (change the 'H' to 'J' in "Hello World", for example), and then run:

```
mcx16asm mcx16s6mb_rom.txt -l
```

- This will compile and load the new code via JTAG into the mcx16rom, and restart the processor

 © 2011 Gregory Bredthauer

Experiment



 © 2011 Gregory Bredthauer

Conclusion

- This concludes the quick mcx16 demo. Feel free to experiment - compare resource usage for different numbers of registers, stack depths, single/dual cycle, or check maximum clock frequencies (> 111 MHz for the example single cycle project).
- Thanks for giving mcx16 a try, I hope it proves useful! The latest mcx16 info will be posted at

  http://www.bitpond.com/mcx16

- Greg Bredthauer

 © 2011 Gregory Bredthauer