LC3 Processor Implemented in FPGA
Nexys 2 System
Console emulation through a serial terminal

There are two folders along with this PDF:

- **vhdl**: VHDL code for the LC3 processor, the bus interconnection subsystem, the serial communication module, the memory interface, and a peripheral for console emulation.
- **asmtest**: a simple executable code, which is meant to be executed in the LC3 processor implemented in the FPGA. This code echoes all the characters received from the PC connected to the FPGA through an RS-232 serial cable. You can download this code to the FPGA with the instructions provided in this document.

**ASM Test code**:

- The executable code is available in the file: consoleecho.hex. It is a HEX-value file that contains the address where the program starts and its instructions/data.
- A partial asm code (in PDF) and a couple of cmd files are provided as well.
- The file consoleecho.asm.pdf contains only a few ASM instructions for the program. Note that the codes for the subroutines GETCHAR (GETC), PUTCHAR (OUT), and PUTSMSG (PUTS) were not provided. You are in charge of completing the code to be used in your practice.

**To download any executable code into the FPGA Memory, perform to following steps:**

1. Create an ISE Xilinx project and add all the VHDL files. Synthesize it and verify there are no errors and critical warnings. If you want to try the ASM test code, jump to step 4.
2. Assemble the ASM code to generate *.obj and *.sym files.
3. Convert the resulting code (.obj) to HEX values. In Linux, use the following command: hexdump -v -e '/1 "%02X\n"' yourfile.obj > yourfile.hex.
4. Remove the two first HEX values (00, 00 in consoleecho.hex) in the HEX file generated (Why?).
5. Count the number of 1-byte HEX values in the generated HEX file and divide it by 2. Keep this value in mind (CODESIZE = HEXValues / 2, Why?).
6. Use an RS-232 cable to connect the PC to the Nexys 2 system.
7. Launch a serial terminal such as CuteCom, Minicom, etc. in your PC and configure it for serial communication: 9600 bps, 8 bits for data, 1 bit for stop (What are these values?).
8. Configure the FPGA using the .bit file generated in step 1).
9. Send a 0x55 value from the PC to the FPGA through the serial terminal (Why?).
10. Use the protocol provided in this document to write the instructions and data

from the HEX code into the FPGA memory starting at address 0x0000 (Why this address?). Then, verify the writing using the protocol for reading. In the ASM Test code, the files consoleecho.wcmd and consolecho.rcmd might be useful. Any idea what are they for?

11. Start the LC3 processor using the protocol (How?). In the ASM Test code, you should see a message in the Cutecom inviting you to type characters.

**Protocol for Writing/Reading into/from the FPGA Memory and Starting the LC3 Processor**

Any time you configure or reset the FPGA, you have to send an initialization code, HEX value: x55, from the serial terminal to the Nexys 2. It allows the UART module to determine the communication speed.

For this protocol, the FPGA will answer any 1-byte HEX value you send from the serial terminal with a 1-byte HEX value, except when you send the data to be written into the FPGA memory.

1. Any communication sequence starts with the HEX value: xFF (Start). The FPGA must answer with the HEX value: x01.
2. To write data to the FPGA memory, send the HEX value: x01. Answer: x02.
   To read data from the FPGA memory, send the HEX value: x00. Answer: x02.
   To start the LC3 processor, send the HEX value: x53 ('S'). Answer: x00. Once the LC3 has started, the protocol is disabled and the serial communication system allows your LC3 program to interact with the serial terminal interface. The protocol is enabled again when the FPGA is reset.
3. For both writing/reading, you must indicate the number of words (2 bytes) to be written/read (size). A 16-bit value must be provided. For example, if you want to write 20 words, send two HEX values: x00, x14. Answers: x03, x04.
4. For both writing/reading, you must indicate the address to start writing/reading data. A 16-bit value must be provided. For example, if the address is 0x2050, send two HEX values: x20, x50. Answers: x05, x06.
5. For writing, provide n values of 16-bits each (two HEX values), where n is the size specified in step 3. Complete the protocol with the HEX value: xFA (Stop), Answer: x0A. For reading, complete the protocol with the HEX value: xFA (Stop), Answer: x0B and wait for n values of 16-bits each.

Any time you provide the Stop value or reset the FPGA, the protocol is restarted.