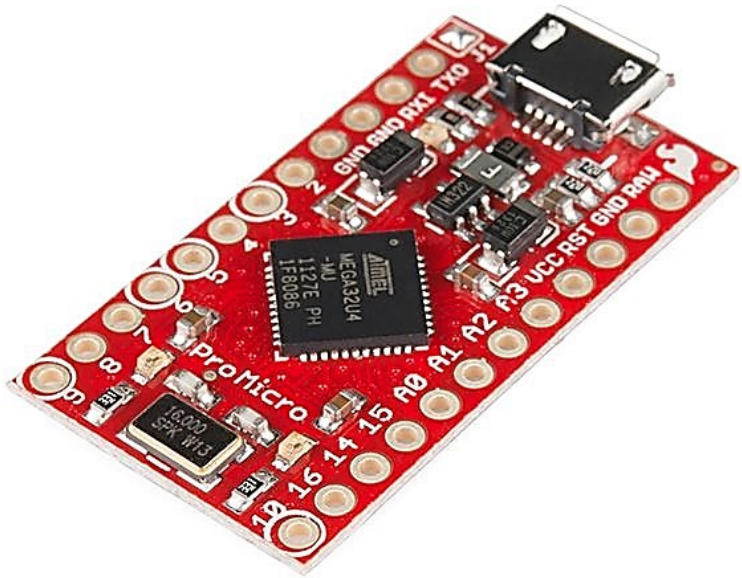
Electrical and Computer Engineering
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Dr. Tooraj Nikoubin



Project #5:
(1) Interconnecting submodules
(2) FPGA implementation

EEDG/CE 6302
Microprocessors and Embedded Systems

Objectives:

- Understanding hierarchical designs in HDLs
- Creating testbench to validate the whole design
- Creating a bit file and upload it to an FPGA
- Interfacing FPGA evaluation board with inputs and outputs

1. Introduction:

This is the last part of the MCU project. Until now, you create all required submodules and now you need to interconnect them to create your own MCU.

2. Hierarchical Design

What you did is called bottom-up hierarchical design. You wrote codes for all submodules (bottom) and now you want to create your main module (up). Sometimes you need to write in up-bottom hierarchical coding style and sometimes none of them; you start from the middle and go up or down and continue to complete your whole design.

2.1. Interconnections

You have to use wire (Verilog) or signal (VHDL) to connect all modules together. To understand howtheys should be connected, see the block diagram of the MCU illustrated in Figure (1). The clock (clk), Reset (rst) are single-bit inputs. The Input is an 8-bit input and Output_MSB and Output_LSB are 8-bit outputs of your MCU.

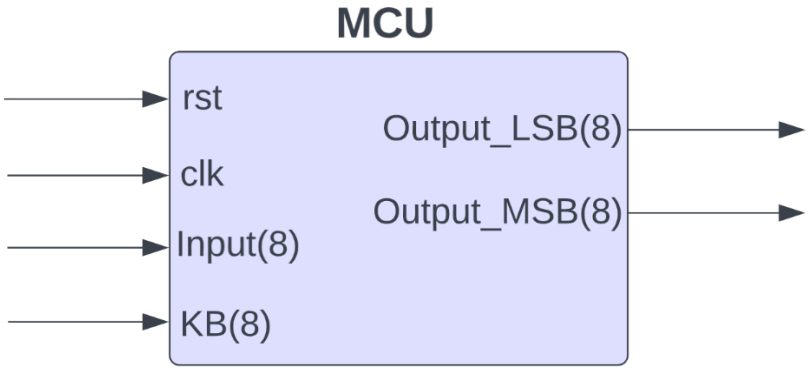


Figure (1): Block diagram of the MCU

Inside the MCU, there are three main submodules: CPU, Data Memory and Instrcution Memory. The interconncetion between these modules are shown in Figure (2). The remaining modules, all of which you have already desgined are placed in the CPU. Those interconnections that are used inside of the CPU are depicted in the Figure (12) at the end of this manual.

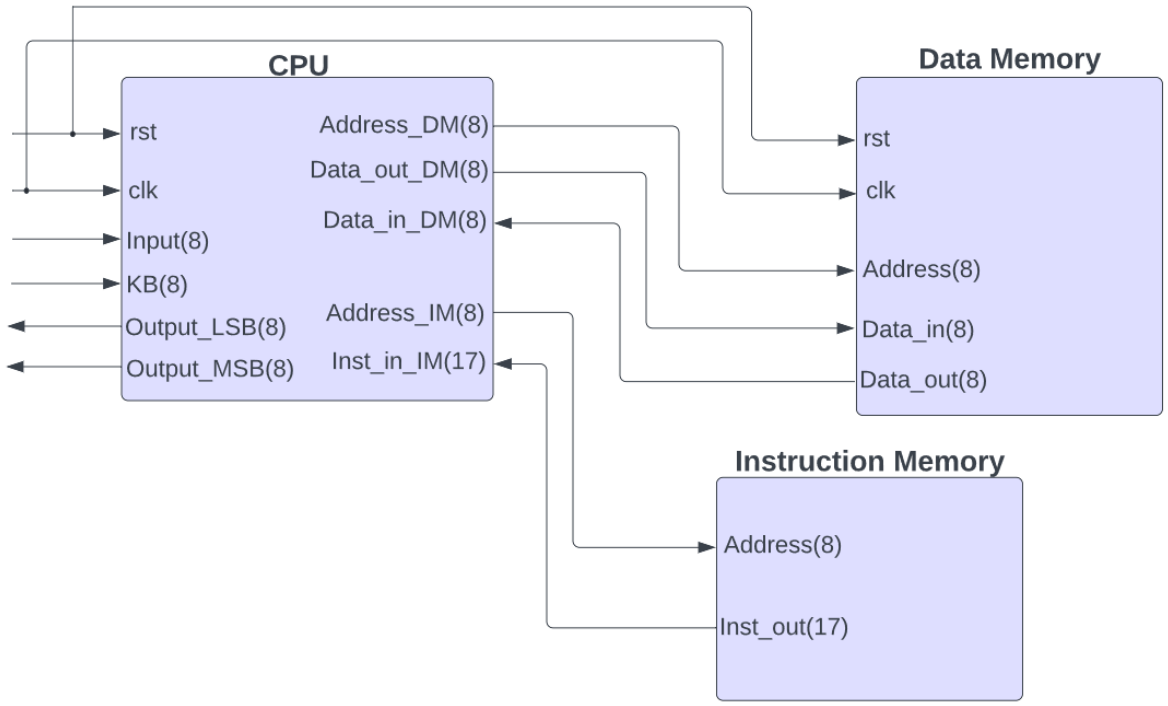


Figure (2): Block diagram of internal components of the MCU

3. FPGA evaluation board

A Digilent's Nexys 3 evaluation board is given to you to upload your codes. This board contains a Xilinx Spartan-6 FPGA. You can find the FPGA in the middle of the board with the following part number: "XC6SLX16-2CSG324C". its block diagram is shown in Figure (3).

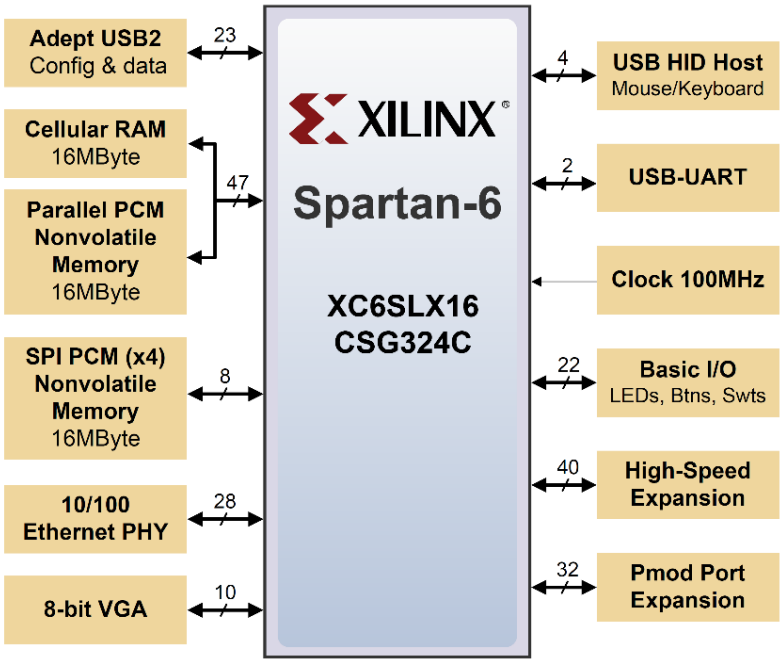


Figure (3): The block diagram of the evaluation board

To find out more information about the board, read the reference manual available on eLearning.

Note that since Xilinx Vivado does not support Spartan 6 series FPGAs, you need Xilinx ISE to create .bit file for the FPGA.

4. Interfacing keypad and monitor

4.1 Keypad

You are given a 4×4 matrix keypad, and you need to connect the keypad to the port JC1 of the evaluation board. They are shown in Figure (4).

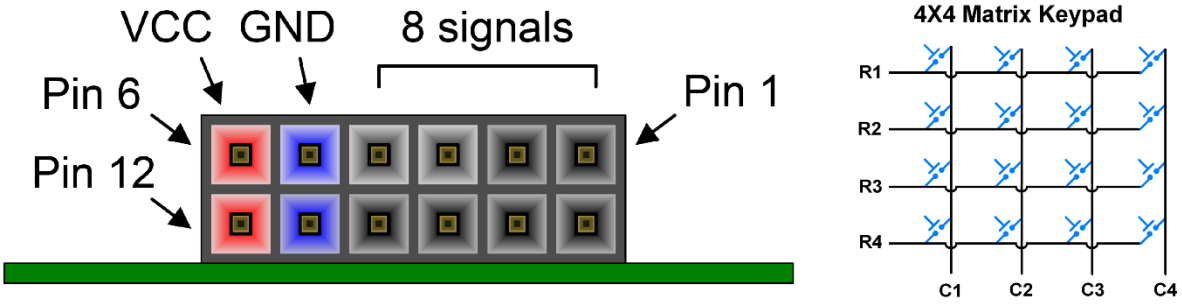


Figure (4): Keypad 4×4 and JC1 port

Rows of the keypad are connected to JC[1] to JC[4] and its columns are connected to JC[7] to JC[10]. Pins for connections to VCC and GND are not used. Each push button on the keypad is used to select one of the 16 blocks which divide the VGA display.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Figure (5): VGA display grids

4.2. Monitor

You have to connect your board to a monitor by using standard VGA cable.

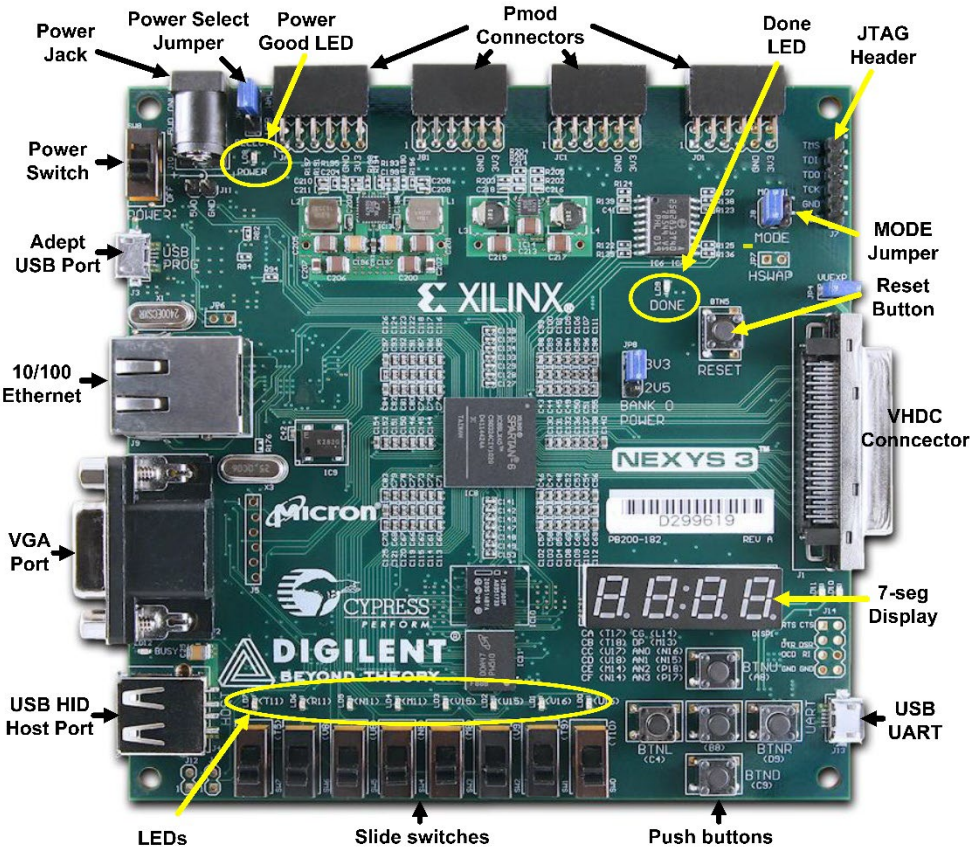


Figure (6): VGA port

4.3. Drivers

To connect keypad and monitor to your evaluation board, you need to use drivers. These drivers were written and gave to you in a zip file. To create the required logic, follow the instructions mentioned below step by step:

(1) Extract files to the same directory that your project was created.
(2) Add all Verilog files to your project.
(3) Then add an implementation constraint file named “MCU.ucf” to your project.
(4) Now you have to connect the drivers to your MCU. Create a new HDL file named “minsys.v” with the following ports. All names must be the **SAME** as below. You must **NOT** change any names here:

```
module minsys (
        input          clk,
        input          rst,
        input          EI,
        input   [7:0]  SW,
        output [9:0]   vga_cont,
        input   [3:0]  Keypad_rows,
        output [3:0]   Keypad_cols
    );
```

(5) In the “minsys” module, interconnect your “MCU” and “mcu_io” as depicted in Figure (7).

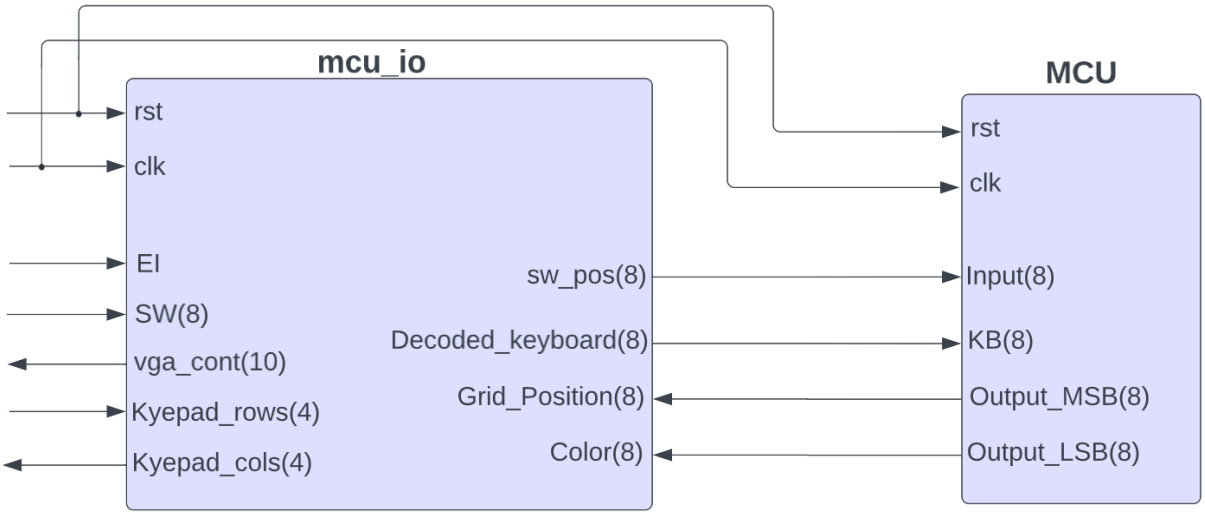


Figure (7): MCU and mcu_io connection

(6) You have to write a program to run on the MCU. You can write your own program. Here, a simple program is shown:

```
INST(0):    IN      R[1]          // IN    R[DA]
INST(1):    INK     R[2]          // INK   R[DA]
INST(2):    OUT     R[2] , R[1]   // OUT R[SA] , R[SB]
INST(3):    JMR     R[0]          // Jump to INST(0)
```

This program reads input port (connected to switches SW[0] to SW[7]) and save the value to R1. Next, the content of the keypad is read and saved to R2. Then, these two values are placed to the output register. Note that OUT instruction uses SA and SB to save the values to its LSB and MSB parts,

respectively. The LSB indicates the keypad values (16 blocks) and MSB determines the color (256 colors). You can see a sample result in Figure (8).

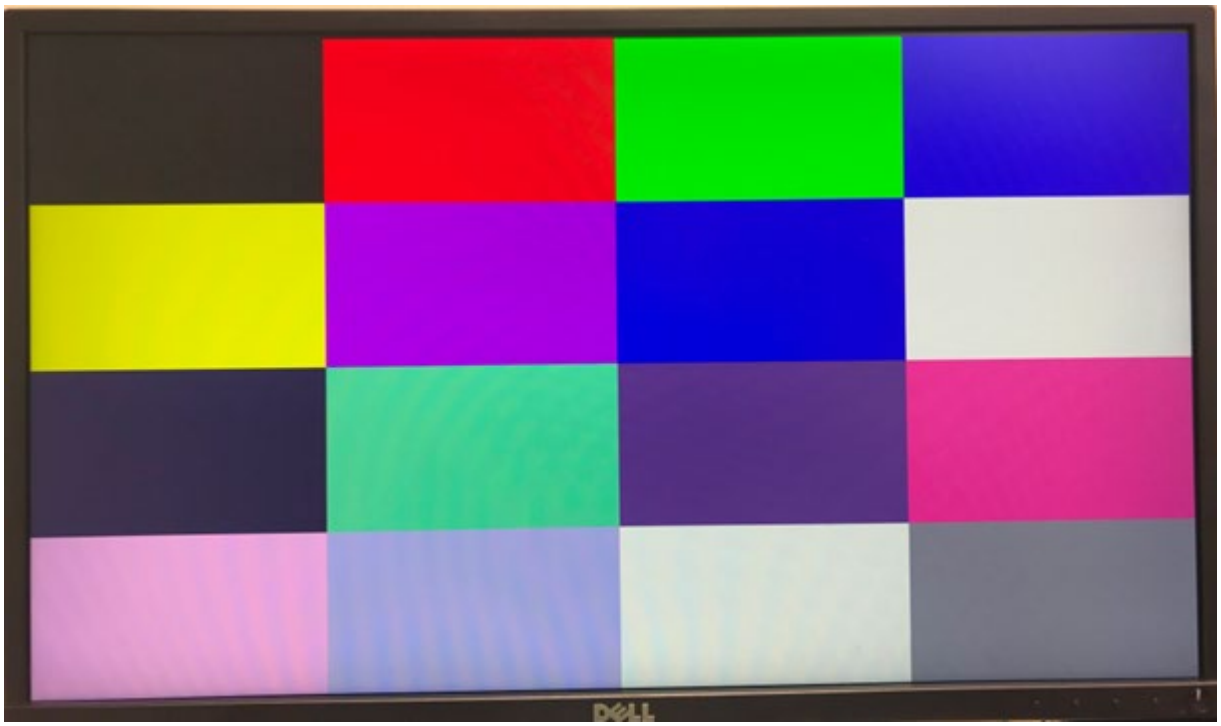


Figure (8): Sample result of running the program

(7) Now, you can synthesize your “minsys” module and create a related “.bit” file . This file will be uploaded to the FPGA.

5. Programming

For programming the device, you need the “.bit” file which is generated by ISE. Since you cannot program the FPGA from no-machine, you need another tool named “Digilent Adept 2”. This program can be downloaded for free from digilent.com. You have to install this program on your PC/Laptop. Note that the board must NOT be connected to your system when you want to install the Adept.
After installation, open the program and make sure that “Auto Initialize SC” has been already checked in the application settings as shown in Figure (9).

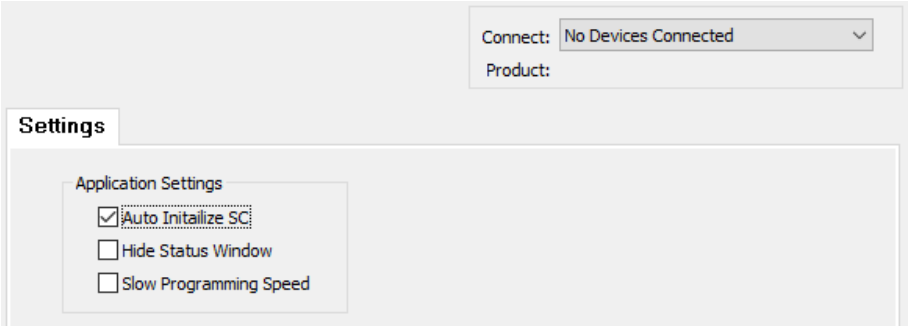


Figure (9): Adept startup after installation

Now connect Nexys-3 to your system. After a while, click on the “Connect” drop down list and select Nexys3. It now detects the board, and it should be ready to find the FPGA by clocking on “Initialize Chain” button, as illustrated in Figure (10). After initialization, device model and device ID are shown in the status window.

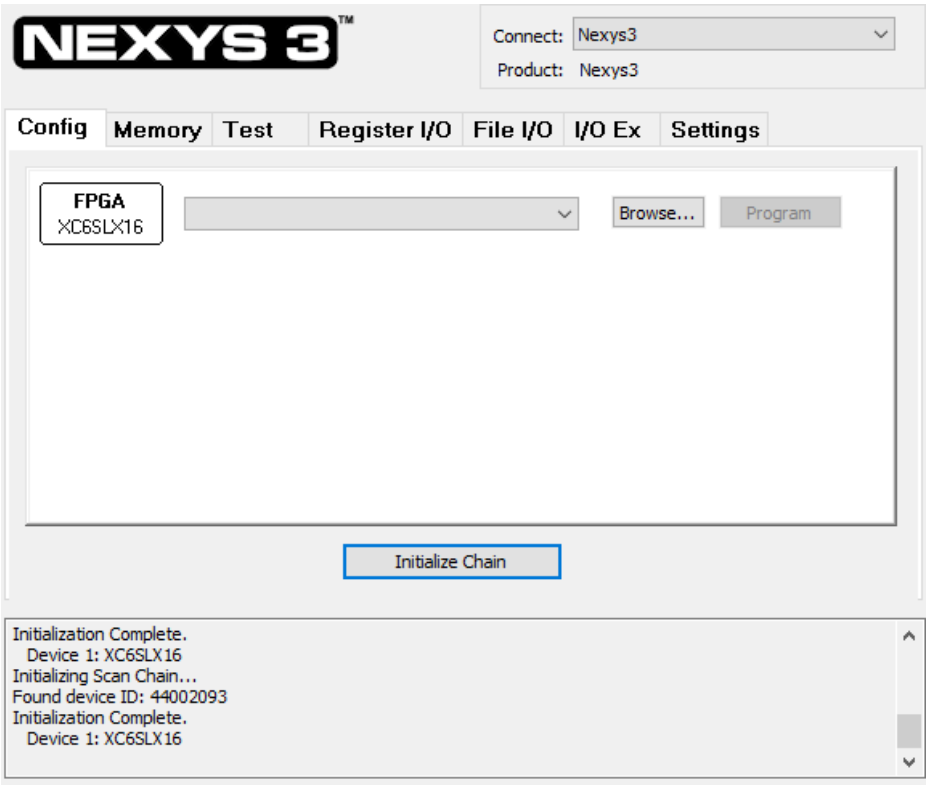


Figure (10): Initializing chain and detecting the FPGA

Now click on Browse and g to the location where your ".bit" file is, and select it. Then press Program button to program your FPGA. After a few seconds, it is uploaded to the FPGA and "Programming Successful" message will be reported as illustrated in Figure (11).

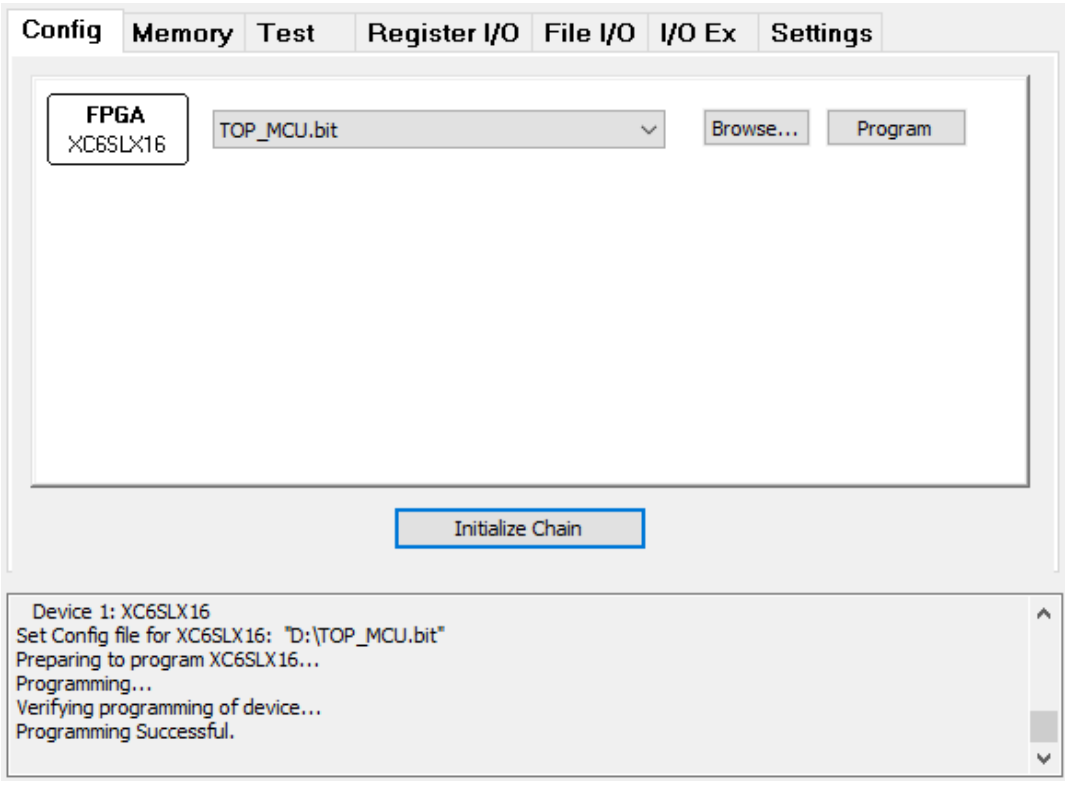


Figure (11): Programming the FPGA

Now, the board is ready for running your written HDL codes.

5. Questions:

1. Similar to Figure (2), draw the interconnections of the submodules in the CPU.

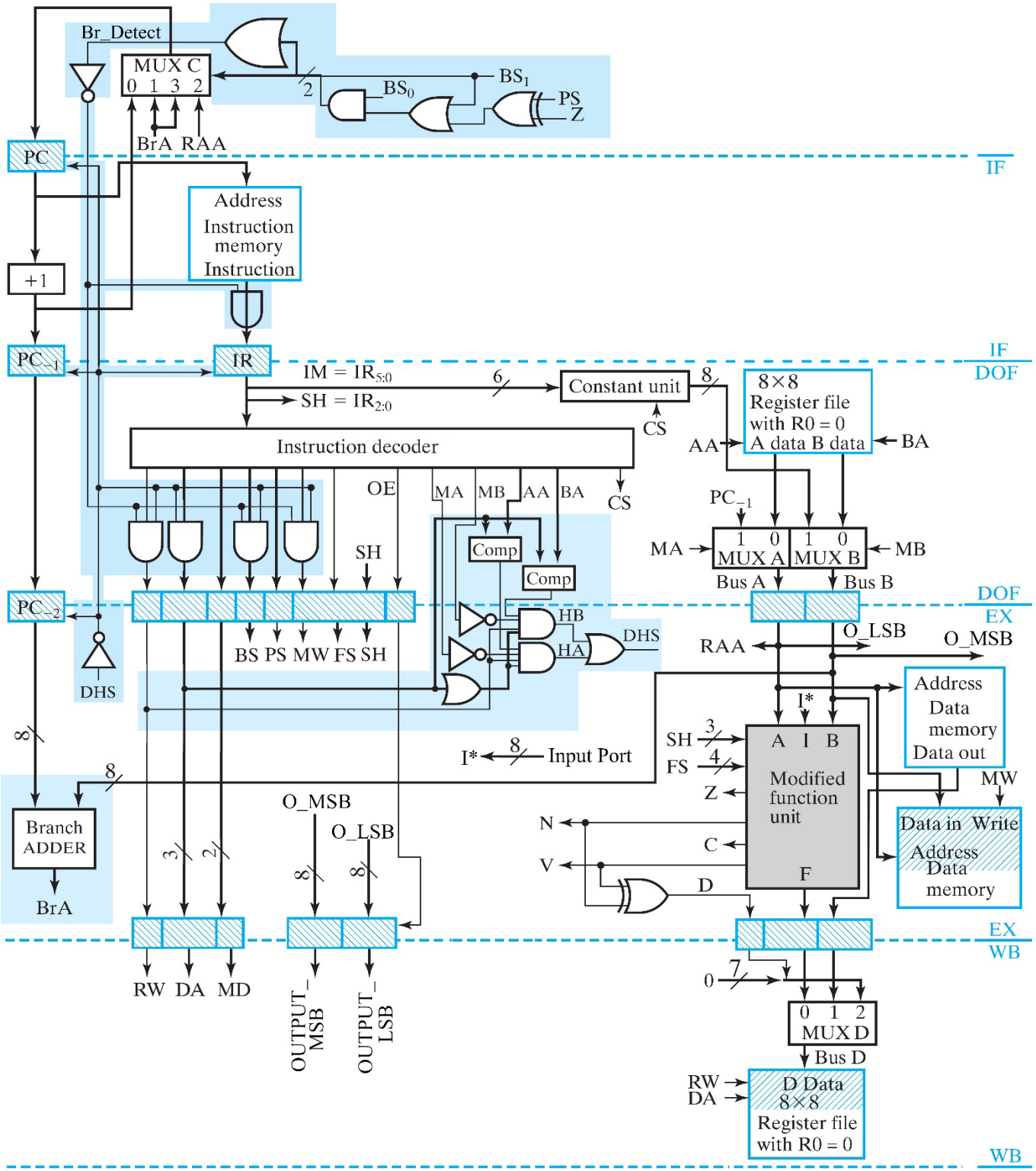


Figure (12): MUC Block Diagram

Reference:
Mano, M. M. AND Kime, C. R. AND Martin M. "*Logic and Computer Design Fundamentals*", 5th ed. Hoboken, NJ: Pearson Higher Education Inc., 2015, pp 585-615.