LAB 1: How to Build a basic SoC

Blake Williams

09/06/2024

**Summary**

In this Lab the goal was to Implement a basic System-on-Chip design targeted toward the Arty-A7-100 development board. To do this the Vivado development platform was utilized. The SoC will utilize the Microblaze processor, USB UART serial controller, and a DDR Memory Controller.

**Design and Implementation**

To begin first (in Vivado) a new block design was created to house the SoC chip design. Then the system clock was added to run all processes. This clock has outputs being CLK\_OUT1 and CLK\_OUT2 which were set to 200 MHz and 100 MHz respectfully. The Clock was also set to reset on a active low signal. Next the MIG (Memory Interface Generator) which interfaces with the clk and takes a clk\_ref\_i and sys\_clk\_i. The clk\_ref\_i is bound to the CLK\_OUT1 and the sys\_clk\_i is bound to the CLK\_OUT2. Then we add the MicroBlaze processor with 32kb local memory, Real-Time control, and No Interrupt controller. The last IP that is added is the UART controller which allows interfacing with the serial connection. Next was adding design constraints. The reason for this is that Vivado would generate from the board’s clock pin instead of the system clock. That is why the following line was added to the constraints

set\_property CLOCK\_DEDICATED\_ROUTE true [get\_nets base\_soc\_i/clk\_wiz\_0/inst/clk\_out2]

After this the Block design was completed, and now a Bitstream is needed. This is done by running the “Generate Bitstream” which takes around 5 minutes to complete. Once this is completed the hardware was exported to a .xsa file and Vitis was opened. Here a new platform was created with the .xsa hardware description. Once the platform was created, an application was created to run on top of the platform. This C application simply printed the username and student ID 5 times. In order to do this the xil\_printf() function was utilized which cuts a lot of overhead of the normal printf() function. In order to see the users name and student ID print, the serial log was enabled and set to use baudrate 9600. After this the Users Name and Student ID defined in the C code would print to the screen 5 times.

A screenshot of a computer

Description automatically generated

*Example Output*

A diagram of a computer

Description automatically generated

*Final Block Design*

A computer screen shot of a code

Description automatically generated

*Example Code (Screen shot taken in VScode)*

**Conclusion**

Overall, this lab helped to introduce the Vivado suite and gain a better understanding of the inner workings of general SoC. Along with learning how to interface with the board over UART connection and going over how data is processed on this SoC.