San Jose State University
Department of Electrical Engineering

Laboratory Assignment #2 (Due April 7, 1:00PM)

Rules on Cheating

Each student is expected to do his or her individual work for this course. Students who turn in identical lab solutions will be considered to have copied. **Two hundred percent** of the maximum possible grade will be deducted for each instance of cheating on lab assignments. All students must work alone. Sharing of code and data between students is considered cheating and will receive appropriate action in accordance with University policy.

Part 1 Instruction to set up Zybo environment.

1. Install Vivado 2016.2 HL Webpack.
2. Start Xilinx SDK 2016.2. A window will appear asking to select workspace directory. Choose your preferred directory.
3. Extract the provided zip file **SAD.zip** to a directory of your choice.
4. In Xilinx SDK click **File**, and select **Import**
5. In the window that opens, expand **General** and select **Existing Projects into Workspace**, click **Next**.
6. Click **Browse**, and select the **SAD.sdk** directory you got from extracting **SAD.zip** file.
7. Your window should look like the following:

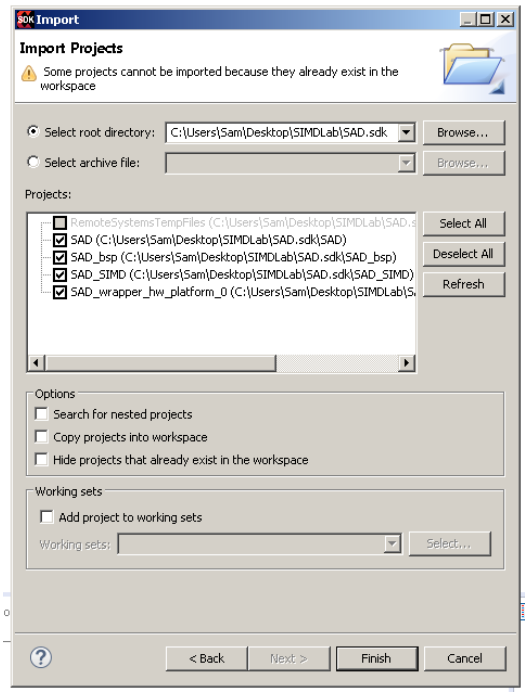

Fig. 1. Importing SAD project

8. Click **Finish**.
9. The **Project Explorer** will look like this after importing the project

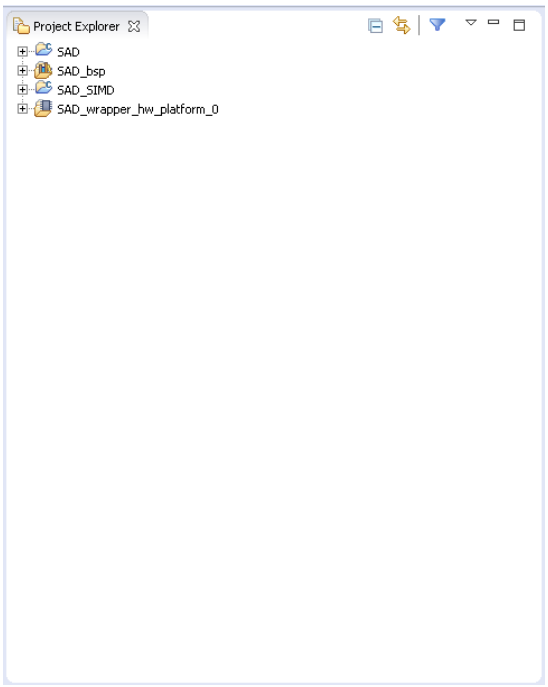

Fig. 2. Project Explorer

10. Expand **SAD** folder, and then expand **src** folder. Double-click on **withoutsimd.c** file. This file contains the source code without any **ARM NEON** instructions.
11. To run this code, click on **Xilinx Tools**, and click **Program FPGA**

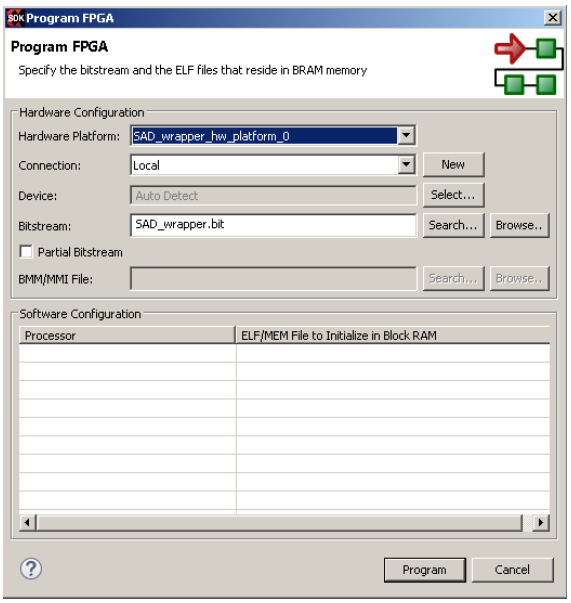

Fig. 3. Program FPGA

12. Click **Program**
13. Right-click on **SAD** folder in **Project Explorer**, select **Run As -> Launch on Hardware (GDB)**
14. The results will be displayed on the serial terminal.
15. To complete the lab, expand **SAD_SIMD** folder in **Project Explorer** and write your code using **NEON SIMD** instructions in the **withsimd.c** file.
16. You can run/test your code in the same way as mentioned previously.

GCC Compiler Flags
-c -fmessage-length=0 -MT"$@" -mcpu=cortex-a9 -mfpu=neon -mfloat-abi=hard
GCC Linker Flags
-mcpu=cortex-a9 -mfpu=vfpv3 -mfloat-abi=hard -Wl,-build-id=none -specs=Xilinx.spec

Part 2. (What to submit in Canvas) Present the speedup you achieve in Table. Your report should include **fully commented source codes** and execution log showing the Zynq global counter time. Lab 2 assignment is graded based on the performance-rank obtained by students. Better solutions are those which have a minimal cycles than other solutions.

References

[1] http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0491h/CIHJBEFE.html