SystemC 2.3.3 development environment for Windows-10 with cygwin

COE838: Systems-on-Chip Design

This document is adapted mainly from a post on stackoverflow and the blog by Cristian Slav as given below.

1. http://cfs-vision.com/2017/06/17/learning-systemc-000-learning-materials-and-initial-setup/
2. https://stackoverflow.com/questions/69845570/how-to-integrate-eclipse-systemc-2-3-3-and-cygwin-on-windows

Main steps to integrate systemc-2.3.3, and cygwin on a windows-10 operating system.

1. cygwin: you can download and install it from https://cygwin.com/install.html. Make sure to include the following packages as given in Figure 1 at the end of the document. enter image description here. We installed cygwin in C:\cygwin64. directory.

autoconf	13-1	Keep	▼
automake	11-1	Keep	▼
cmake	3.20.0-1	Keep	▼
cygwin32-gettext	0.19.5.1-1	Keep	▼
gcc-core	11.2.0-1	Keep	▼
gcc-g++	11.2.0-1	Keep	▼
gcc-objc++	11.2.0-1	Keep	▼
make	4.3-1	Keep	▼
mingw64-x86_64-gcc-g++	11.2.0-1	Keep	▼
mingw64-x86_64-gcc-objc	11.2.0-1	Keep	▼
mingw64-x86_64-gettext	0.21-1	Keep	▼
mingw64-x86_64-libsigc++2.0	2.10.0-1	Keep	▼

Figure 1

2. systemc-2.3.3: you can download the zip file from https://www.accellera.org/downloads/standards/systemc and extract the folder in your local disk (e.g., c:\systemc2.3.3).
3. Please read the content of the INSTALL file in the systemc2.3.3 extracted folder.
4. Open CygWin64 terminal.
5. Navigate to the folder you have extracted the systemc zip file.
6. Create temporary directory "objdir" via "mkdir objdir" as explained in the INSTALL file.
7. Change to the temporary directory via "cd objdir" as explained in the INSTALL file.
8. Choose your compiler export CXX="g++ -std=c++14".
9. (Optional Step avoid it for your first-time setup) If you would like to include fixed-point library you need to add DSC_INCLUDE_FX flag to the compiler definition, i.e., export CXX="g++ -std=c++14 -DSC_INCLUDE_FX ".
10. run configure file as "../configure --prefix="location of the libs", e.g., "/sysclibs"
11. After MakeFile is successfully created run it via "make"

12. After the run is successful execute "make install". At this point you have created the system c libraries in "cygwin64/sysclibs".
13. Make sure to set PATH "C:\cygwin\bin" in the windows system environment variable. You can use command prompt admin and setx /m PATH "c:\cygwin64\bin"

Note: The Example is given for the LAB-1 from System-on-Chip Design (COE838/EE8221). However, the steps are similar for other SystemC codes.

i. Copy the files inside "home" repository with cygwin repository as shown below.
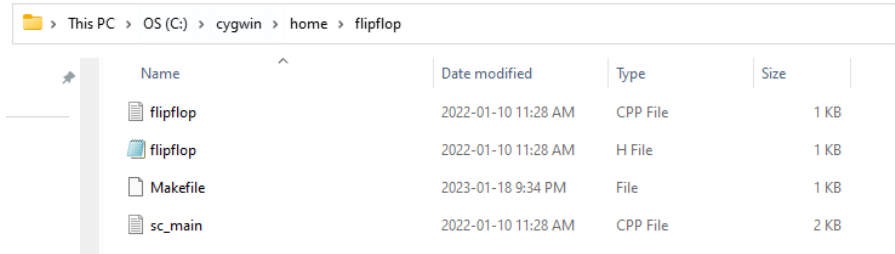
ii. Makefile provided in D2l or "home/courses/coe838/labs/lab1/tutorial/flipflop" is for linux distribution. Therefore, it won't work on windows. You might see an error as shown below

iii. We need to write a different Makefile to use the installed libraries on Windows. Please use the makefile given below
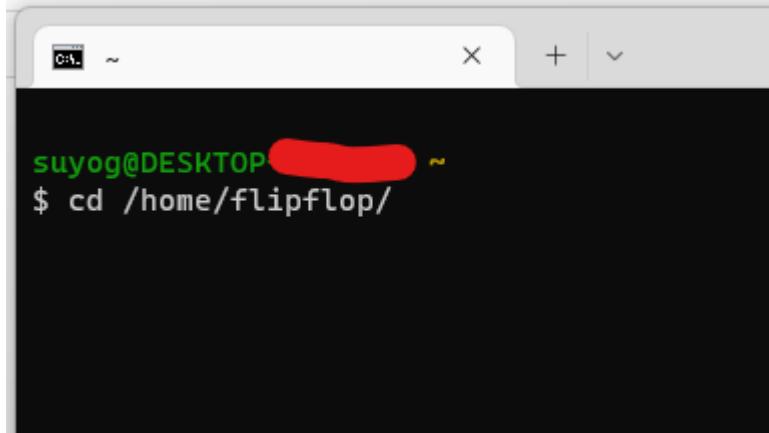
```
RM := rm -rf

all:
	g++ -I"C:\cygwin\sysclibs\include" -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"flipflop.d" -MT"flipflop.o" -o "flipflop.o" "flipflop.cpp"
	g++ -I"C:\cygwin\sysclibs\include" -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"sc_main.d" -MT"sc_main.o" -o "sc_main.o" "sc_main.cpp"
	g++ -L"C:\cygwin\sysclibs\lib-cygwin" -o "flipflop.exe" ./flipflop.o ./sc_main.o   -lsystemc

clean:
	-$(RM) flipflop.d flipflop.o sc_main.d sc_main.o flipflop.exe trace_file.vcd
	-@echo ' '
```
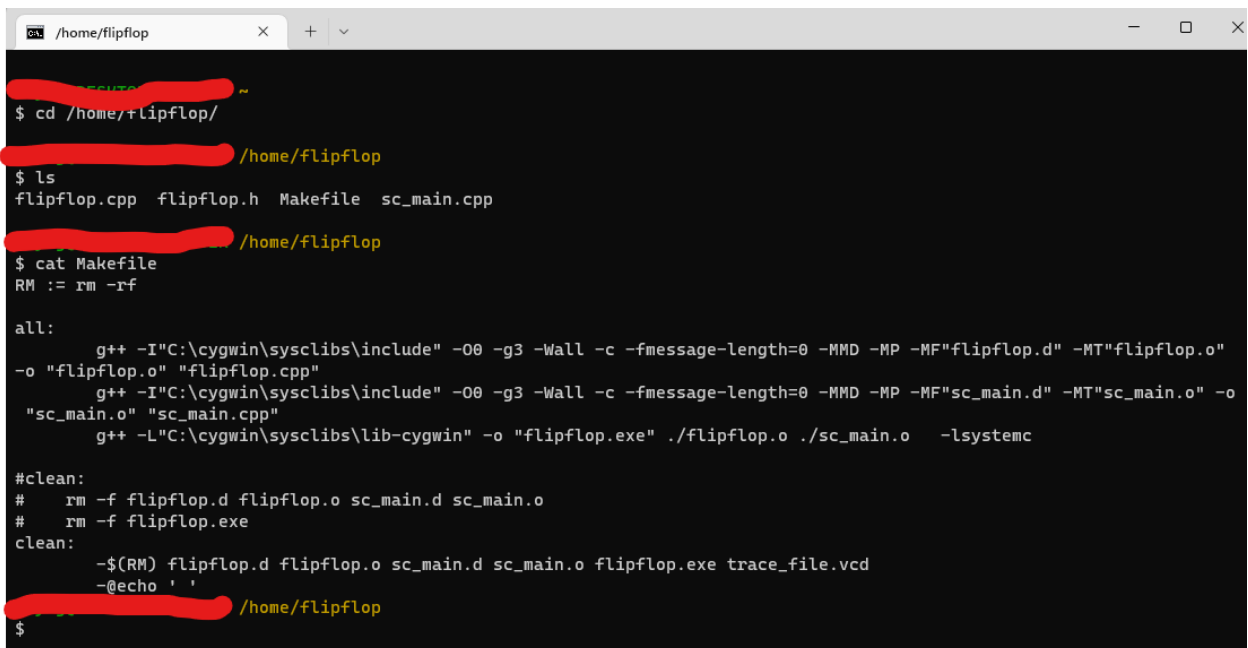
Note: Makefile for windows is also available at D2L.

iv. Now, we have to change the Makefile to compile flipflop on windows.
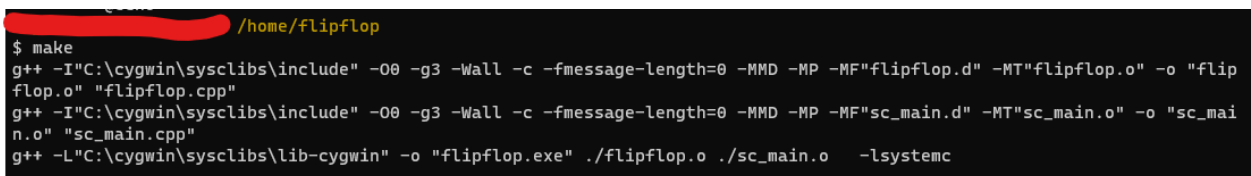v. Open Cygwin terminal and go to the copied folder.

vi. Check the Makefile

vii. Issue "make" command to compile the project.

viii. There should be no error if your installations are correct. Please feel free to edit the makefile to match the paths based on your installation.

ix.   Now, your report should have executable file ".exe" and other obj files.

Note: We still don't have trace file ".vcd" yet.

    x.   Execute the ".exe" file and now it should create the trace file.
   xi.   View and analyze the flipflop "trace_file.vcd" file using gtkwave package for windows.
  xii.   You can open that trace file on "gtkwave" and verify the test cases.