ECE 332 Lab 3: Booting Linux on DE1-SoC Board and Creating SD Card Image

Objectives:

- Understand the process of booting Linux on the DE1-SoC board.
- Learn to create an SD card image for the DE1-SoC board.
- Familiarize yourself with the MSEL button configurations for different boot modes.
- Gain hands-on experience with the Linux command line and basic file operations on the DE1-SoC board.

Prerequisites

- Completion of Lab 2 and familiarity with the DE1-SoC board.
- Basic knowledge of Linux commands.
- DE1-SoC board
- SD card (minimum 4 GB) provided with the kit
- USB to Micro USB cable
- D5M camera

Information to help get started with the Lab 2:

Preparation and Download Required Files:

1. Download the Linux image for the DE1-SoC board from the Terasic website. Download the Linux BSP (Board Support Package):
   MicroSD Card Image file from the following link - Linux Ubuntu Desktop

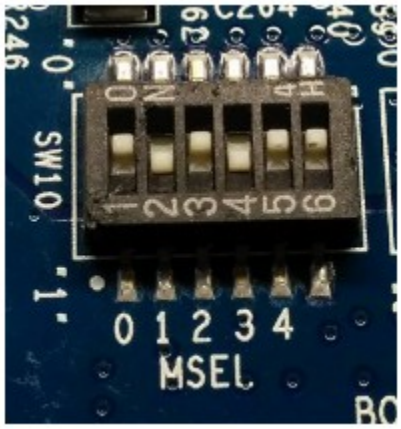
2. Extract the contents of the folder, and find the Linux image in the path:
   DE1-SoC -> DE1-SoC-IP-Linux -> DE1-SoC_UP_Linux_ROS.img
   This image file can be written on the SD card by following the 3 and 4 steps

3. Download and install an SD card flashing tool like Etcher or Win32 Disk Imager on your personal computer (since they are not available on lab machines). Please contact TAs if you need help, since this is a one-time step.

4. Connect the SD card reader along with the SD card to your PC and flash the downloaded image using the above downloaded software.

5. After successfully flashing the image,
   <optional, do this step to transfer files from host machine to Linux system>
   copy the file to the SD card which we will be using once DE1-SoC boots up with Linux.
   After flashing, safely eject the SD card reader.

6. **Keep the board powered OFF.** Switch the MSEL buttons under the board to the following configuration (you can toggle the switches with the help of a pen)

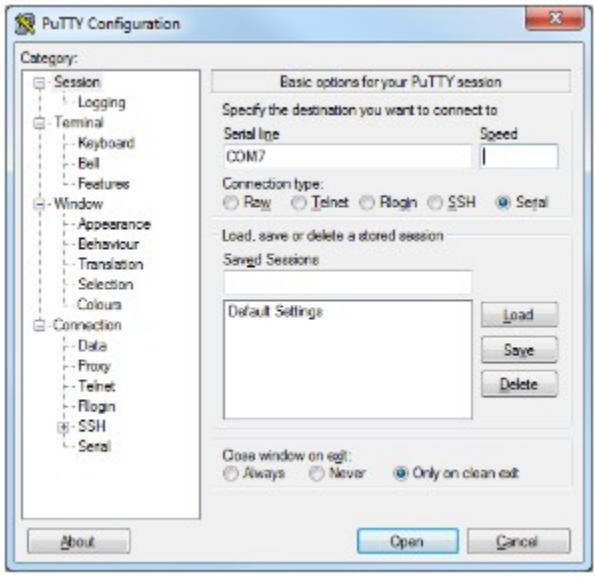


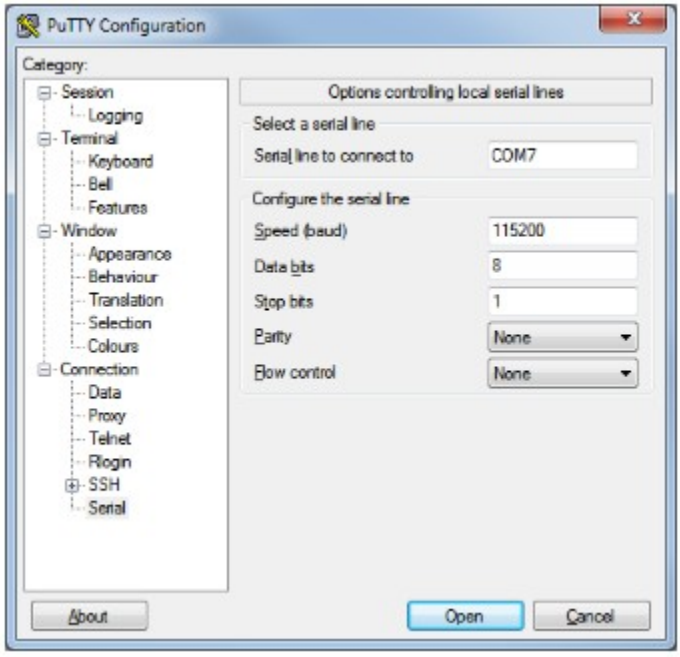
7. Insert the SD card in the SD card slot of the board and power ON the board.

Accessing the DE1-SoC terminal from the host system

8. The ARM processor on DE1-SoC would be booted with the Linux OS flashed on the SD card once it powers on. Connect the DE1-SoC board to the computer with the USB Type A to USB Mini cable provided in your kit.

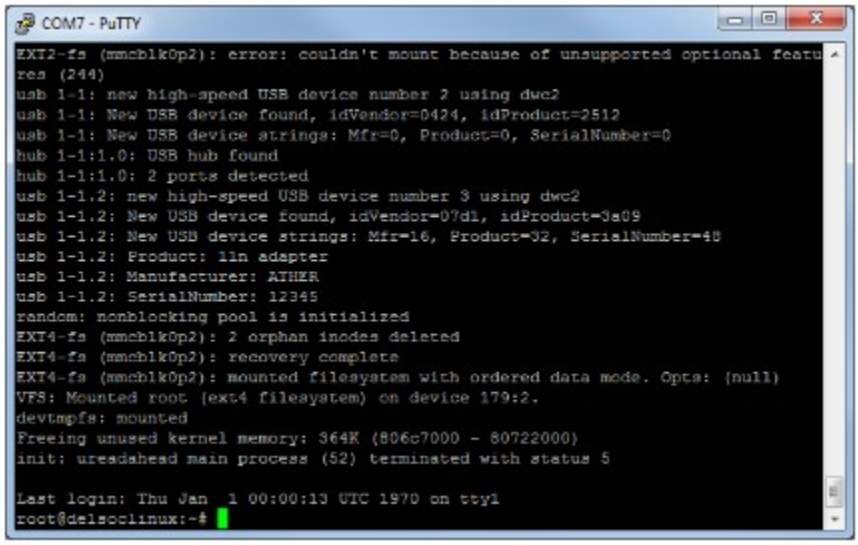
9. Launch putty on the computer and select serial connection with COM serial line and speed 115200





10. Click open will launch the putty session with the terminal and **power on the board now**

11. Linux command prompt can be seen, once the system boots up



Programming FPGA fabric from DE1-SoC terminal

12. First, the OpenCL environment has to be set up. Move to

    cd Opencl/
    source init_opencl.sh
13. Check the environment by using the

    aocl version

    This completes booting Linux OS from ARM processor on DE1-SoC

Compiling aocx file on the host machine (Windows machine)

1. Extract de1_soc folder from the resources file
2. Setup OpenCL environment:
   go to the directory C:\intelFPGA\hld\ and type init_opencl.bat in the command prompt
3. Copy the **hardware** folder to a10_ref folder in the path
   C:\intelFPGA\hld\board\a10_ref\

Replace the hardware folder, this is because we don't have permissions to change environment variables on the host machine. If you are working on your own system, instead of replacing the folder set the environment variable **AOCL_BOARD_PACKAGE_ROOT** to your own directory

4. Copy the **tests** folder to a10_ref folder in the path

5. In the command prompt type
   aoc -list-boards

This should list all the boards that are listed, delete any other directories that we are not working with on the host machine

6. Launch Quartus prime, open the project in hardware directory named top.qpf and perform “Compile and Synthesis” (only compile and synthesis not the entire flow)
7. Close Quartus project and go to the tests\vector_add directory and compile an example project using the following command

```
aoc device/vector_add.cl -o bin/vector_add.aocx -board=de1soc_sharedonly
```

This will compile .aocx file that uses the hardware project and the vector_add kernel.

Programming FPGA fabric with aocx file

1. Connect the SDcard that is flashed with the Linux image to the host machine using the card reader

2. Copy the .aocx file generated in the previous step to the SDcard (Don’t modify any files that are already present in the SDcard). This copied file will be available in the DE1-SoC linux in the path-

```
/media/fat_partititon
```

3. Boot DE1-SoC by inserting the SDcard and set up OpenCL environment as described earlier.

```
cd OpenCL/
source init_opencl.sh
```

4. After setting up OpenCL environment , copy the contents from SDcard which are in /media/fat_partititon/

```
mv /media/fat_partititon/<file_name>.aocx  Labs/Exercise_3/solutions/
```

5. Now, we will program the FPGA fabric with with our design compiled in the previous section

```
cd Labs/Exercise_3/solutions/
aocl program /dev/ac10 bin/<file_name>.aocx
```

6. Copy the capture_image.c and bmp_utilti.h files given in the resource files in to the /home/root/OpenCL/Labs/Exercise_3/solutions/

7. Move to /home/root/OpenCL/Labs/Exercise_3/solutions/ directory, compile the capture_image code and run

   g++ capture_image.c
   ./a.out

14. Above code starts streaming video from camera to monitor, pressing any button (only key1,2 and 3 not 0) will capture the image

Deliverables-

Part1 : Capture image using the capture_image.c file and save it as a .bmp file.

Part2 : Write the matrix multiplication kernel and test it using the Jupyter notebook file given in the resources