School of Computing and Informatics

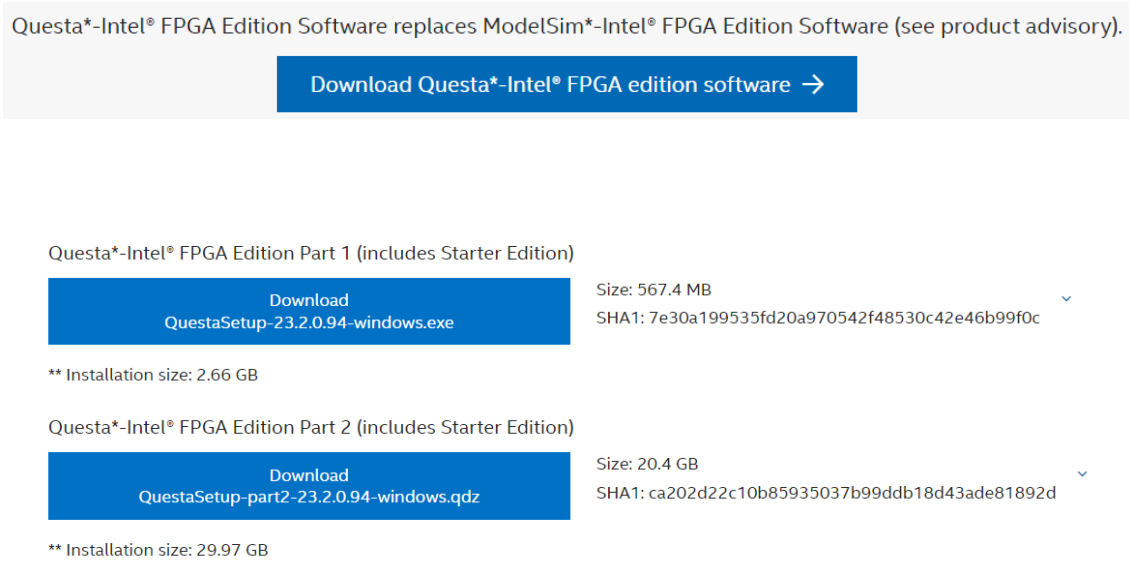
CSCE 585 – VLSI Design – Fall 2023

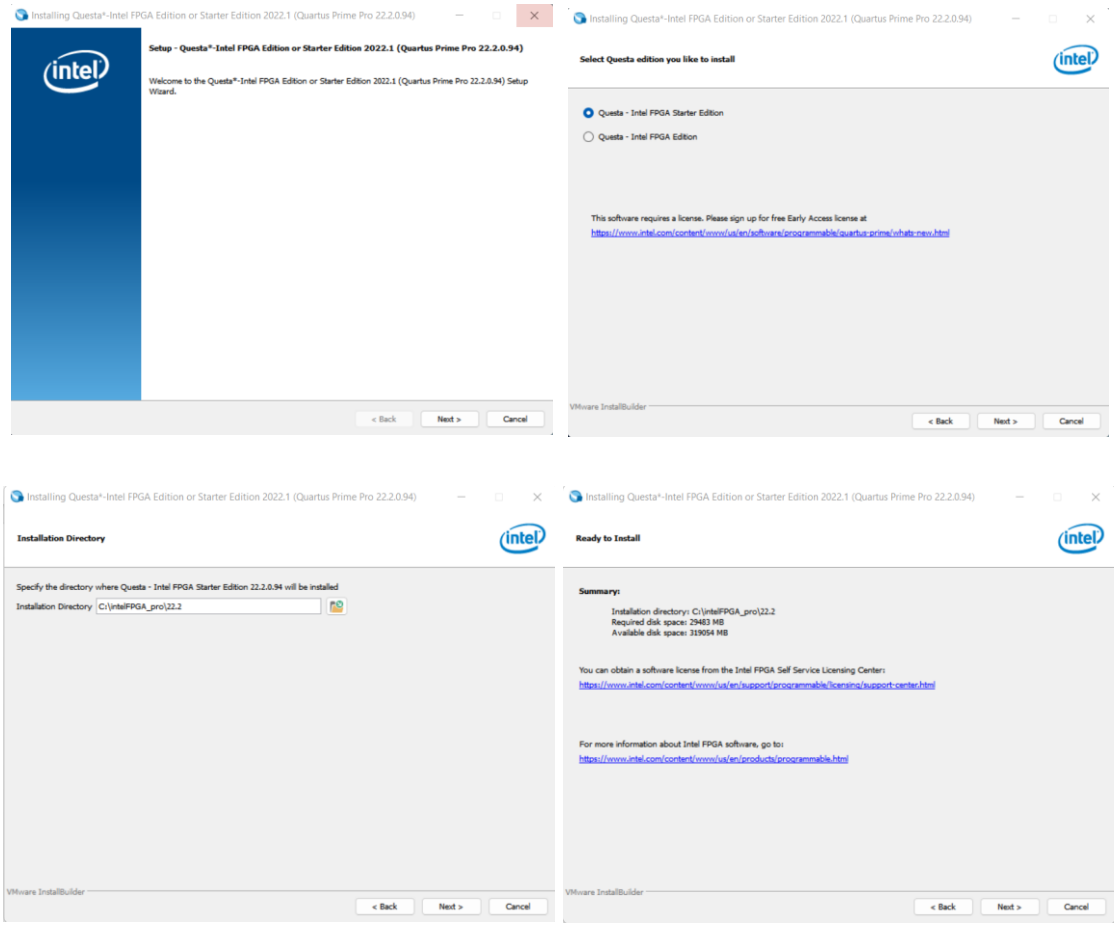
Computer Assignment 1

In this assignment, you will get familiar with Verilog coding and HDL simulation using Siemens EDA Questa (Modelsim). You start by installing/running Questa, writing an HDL code, and simulating the hardware description of a simple function. You then receive the RTL design of a simple MIPS processor and modify its arithmetic and logic unit (ALU) to add a new instruction.

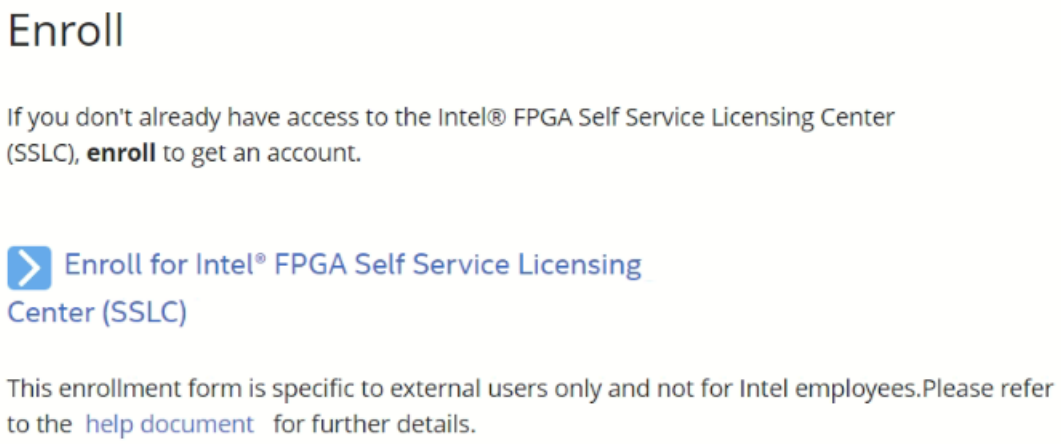
a) Download and install the **free starter edition of Questa** from Intel website:

https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/questa-edition.html

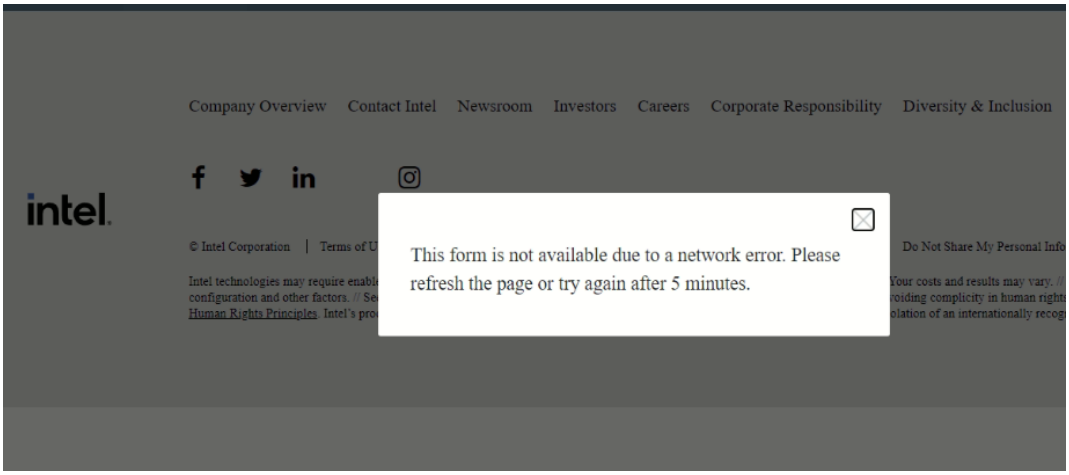


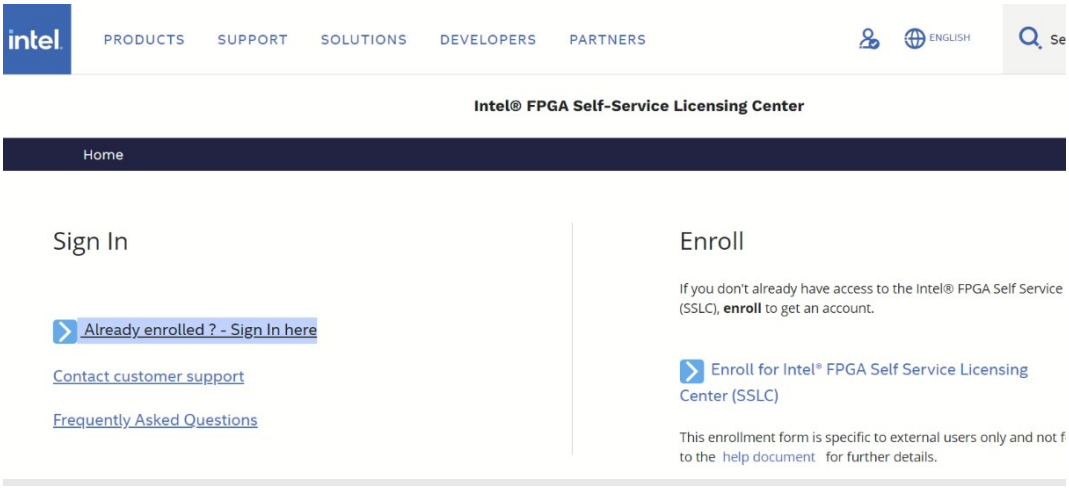


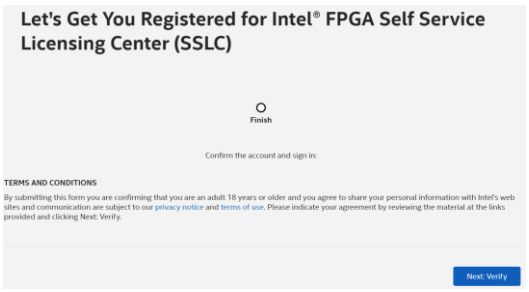
Obtain a free starter license from:
https://fpgasupport.intel.com/Licensing/license/index.html

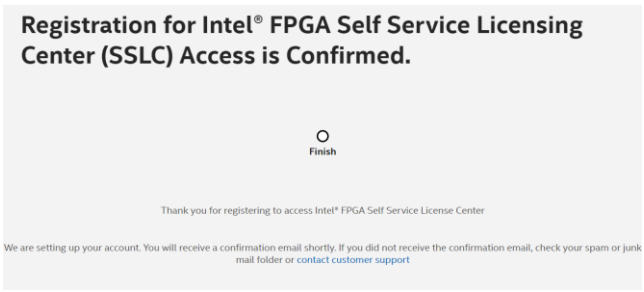


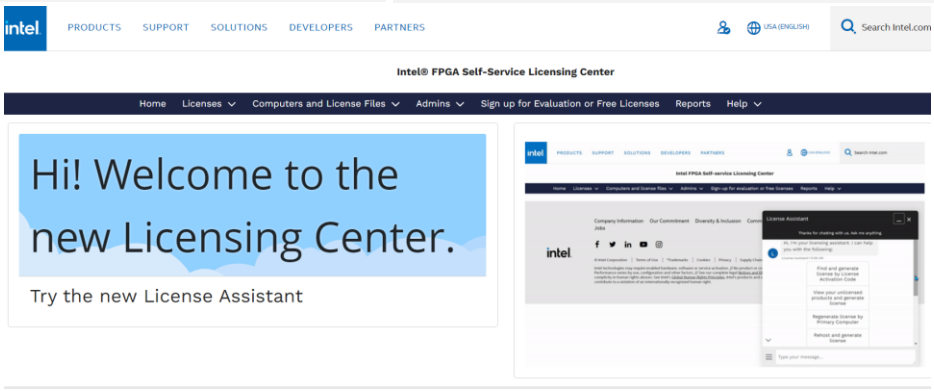
After signing up if you see the following error go back to that webpage again and click on "*Already enrolled ? - Sign In here*"

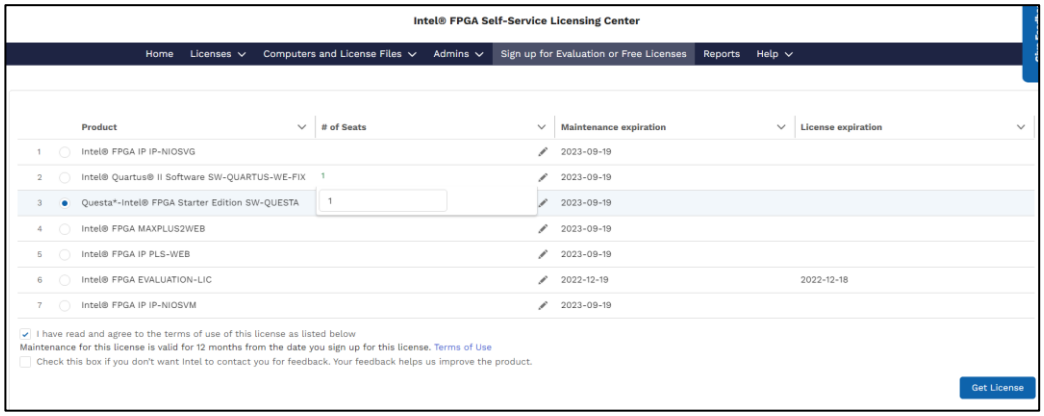

Company Overview
Contact Intel
Newsroom
Investors
Careers
Corporate Responsibility
Diversity & Inclusion
intel
This form is not available due to a network error. Please refresh the page or try again after 5 minutes.

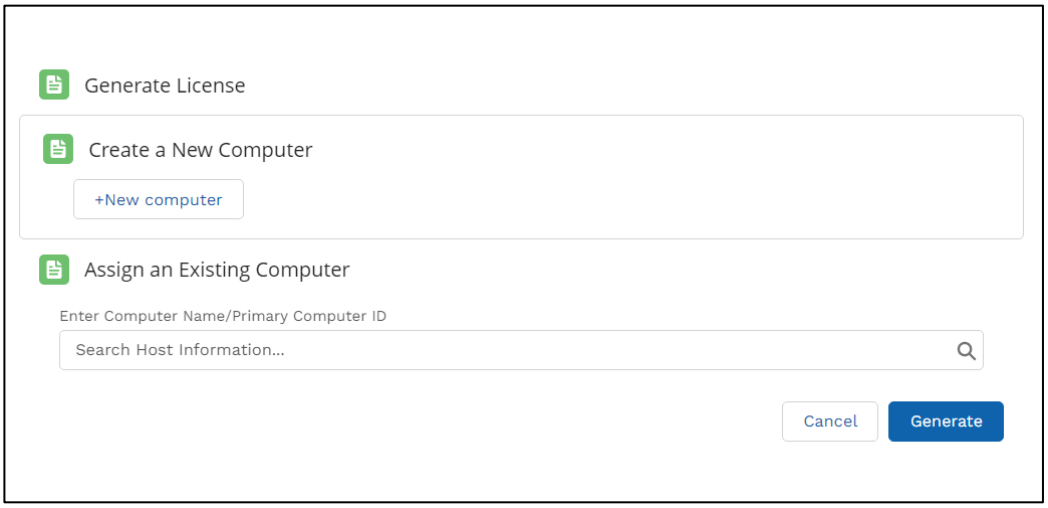

intel
PRODUCTS
SUPPORT
SOLUTIONS
DEVELOPERS
PARTNERS
ENGLISH
Intel® FPGA Self-Service Licensing Center
Home
Sign In
Already enrolled ? - Sign In here
Contact customer support
Frequently Asked Questions
Enroll
If you don't already have access to the Intel® FPGA Self Service (SSLC), enroll to get an account.
Enroll for Intel® FPGA Self Service Licensing Center (SSLC)
This enrollment form is specific to external users only and not f
to the help document for further details.

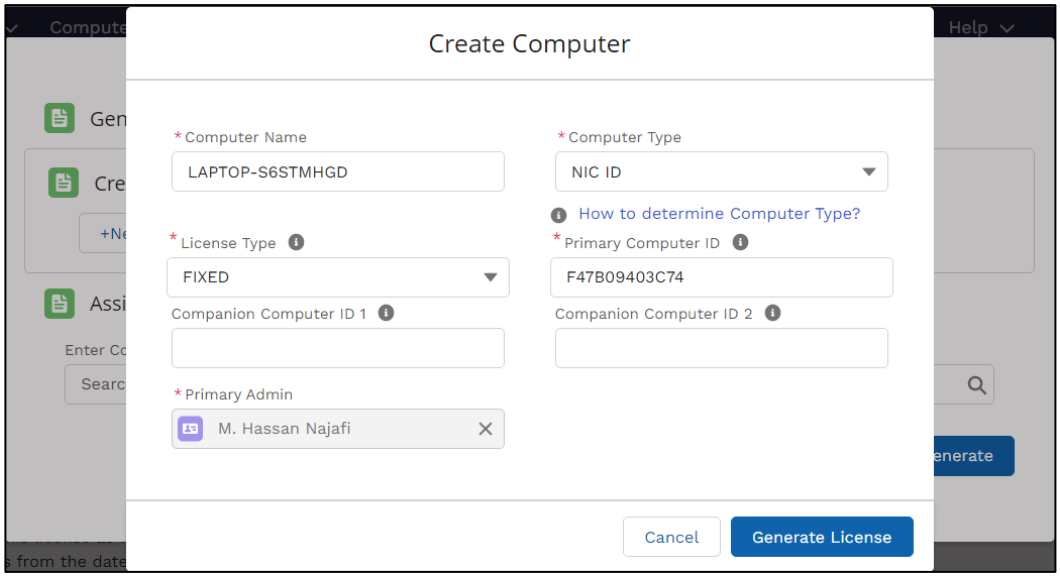

Let's Get You Registered for Intel® FPGA Self Service Licensing Center (SSLC)
Finish
Confirm the account and sign in:
TERMS AND CONDITIONS
By submitting this form you are confirming that you are an adult 18 years or older and you agree to share your personal information with Intel's web sites and communication are subject to our privacy notice and terms of use. Please indicate your agreement by reviewing the material at the links provided and clicking Next: Verify.
Next: Verify


Registration for Intel® FPGA Self Service Licensing Center (SSLC) Access is Confirmed.
Finish
Thank you for registering to access Intel® FPGA Self Service License Center
We are setting up your account. You will receive a confirmation email shortly. If you did not receive the confirmation email, check your spam or junk mail folder or contact customer support

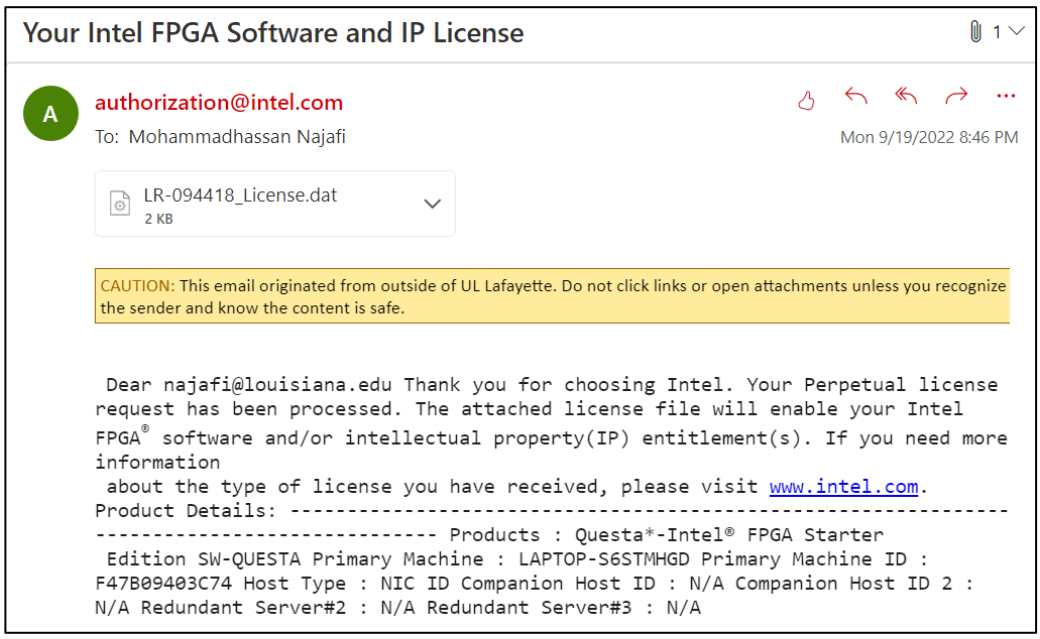

intel
PRODUCTS
SUPPORT
SOLUTIONS
DEVELOPERS
PARTNERS
USA (ENGLISH)
Search Intel.com
Intel® FPGA Self-Service Licensing Center
Home
Licenses
Computers and License Files
Admins
Sign up for Evaluation or Free Licenses
Reports
Help
Hi! Welcome to the new Licensing Center.
Try the new License Assistant







Your network interface card (NIC) ID can be found by “ipconfig /all” in command prompt.



Your Intel FPGA Software and IP License

1

A

authorization@intel.com

To: Mohammadhassan Najafi

Mon 9/19/2022 8:46 PM

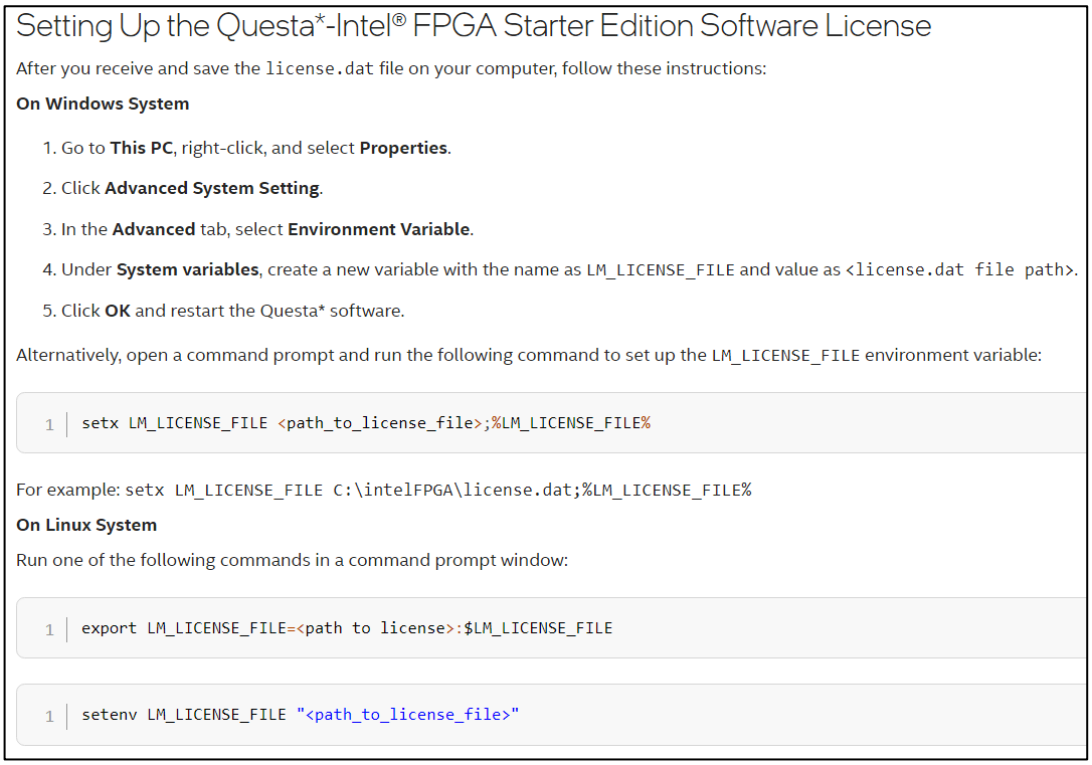
LR-094418_License.dat
2 KB

CAUTION: This email originated from outside of UL Lafayette. Do not click links or open attachments unless you recognize the sender and know the content is safe.

Dear najafi@louisiana.edu Thank you for choosing Intel. Your Perpetual license request has been processed. The attached license file will enable your Intel FPGA® software and/or intellectual property(IP) entitlement(s). If you need more information about the type of license you have received, please visit www.intel.com.
Product Details: ---------- Products : Questa*-Intel® FPGA Starter Edition SW-QUESTA Primary Machine : LAPTOP-S6STMHGD Primary Machine ID : F47B09403C74 Host Type : NIC ID Companion Host ID : N/A Companion Host ID 2 : N/A Redundant Server#2 : N/A Redundant Server#3 : N/A

Download your license to your Questa installation folder and add its address to your system variables as a LM_LICENSE_FILE variable:

https://www.intel.com/content/www/us/en/docs/programmable/683472/22-1/and-software-license.html



Setting Up the Questa*-Intel® FPGA Starter Edition Software License

After you receive and save the `license.dat` file on your computer, follow these instructions:

On Windows System

1. Go to **This PC**, right-click, and select **Properties**.
2. Click **Advanced System Setting**.
3. In the **Advanced** tab, select **Environment Variable**.
4. Under **System variables**, create a new variable with the name as `LM_LICENSE_FILE` and value as `<license.dat file path>`.
5. Click **OK** and restart the Questa* software.

Alternatively, open a command prompt and run the following command to set up the `LM_LICENSE_FILE` environment variable:

```
1 | setx LM_LICENSE_FILE <path_to_license_file>;%LM_LICENSE_FILE%
```

For example: `setx LM_LICENSE_FILE C:\intelFPGA\license.dat;%LM_LICENSE_FILE%`

On Linux System

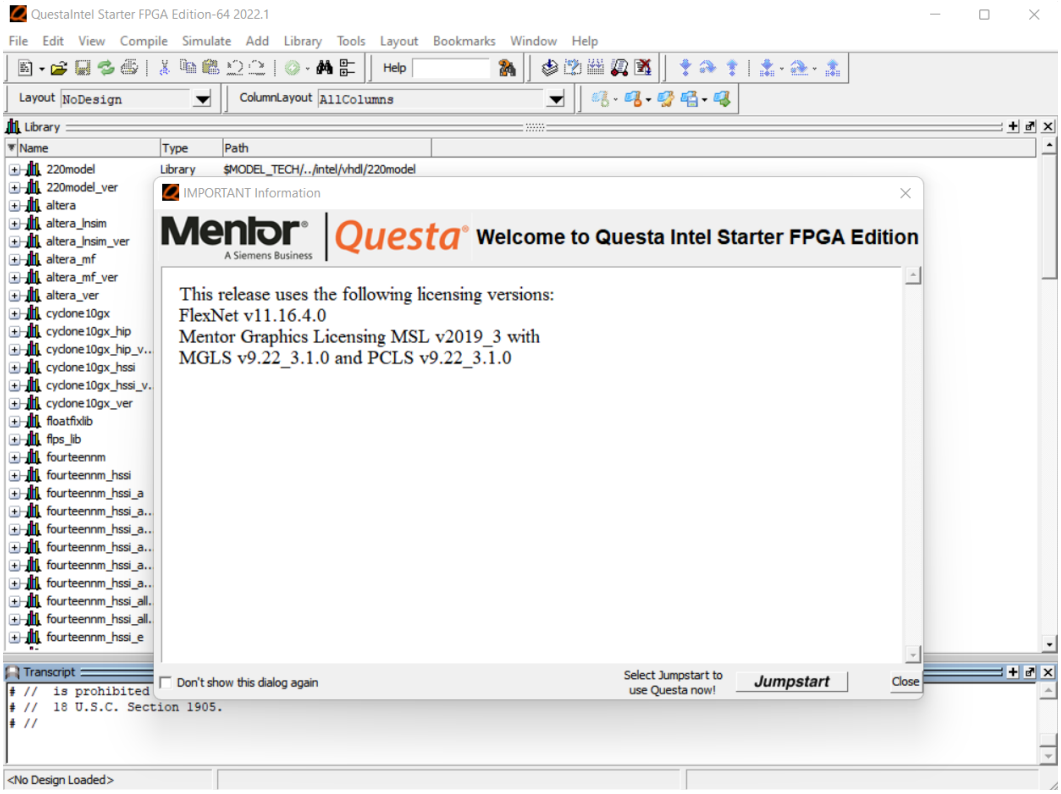
Run one of the following commands in a command prompt window:

```
1 | export LM_LICENSE_FILE=<path to license>:$LM_LICENSE_FILE
```

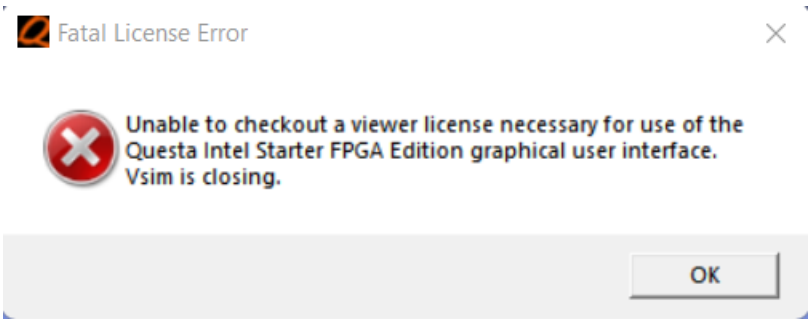
```
1 | setenv LM_LICENSE_FILE "<path_to_license_file>"
```

Confirm Questa and its license are installed correctly by running vsim.exe:

C:\intelFPGA_pro\23.2\questa_fse\win64\vsim.exe



- Note that by requesting "1 license seat" you can only run <u>one Questa window</u>. So, make sure you close the previous Questa window before running a new one to prevent the following error:



b) You can also use the **Windows machines in the SWAMP lab** (3rd floor – Oliver Hall) to run Questa if you don't have a Windows system or enough space on your computer.

Let the instructor know <u>by Sep 15</u> if you have any issue in installing or running Questa.

c) Install **Notepad++** or another editor for editing your Verilog codes.

==========

1. A) Sketch a **transistor-level schematic** for a CMOS logic gate implementing the following 5-input function: $\boldsymbol{Y = \overline{(A + B) \cdot (C + D + E)}}$. Sketch a stick diagram for the function. Estimate the area from the stick diagram (consider a length of $8\boldsymbol{\lambda}$ for each wire track).

   B) Write a **gate level** and a **transistor-level** Verilog module for this function. Verify correct functionality of the two implemented modules (Gate-level and Transistor-level) for **all 32 possible combinations** of the inputs with a **testbench** and **HDL simulation** using Questa. <u>Put your modules, testbench, Questa log output, and a screenshot of the Questa wave output in a separate folder.</u>

2. In class, we discussed the functionality of a simple **MIPS** processor and described the implemented instructions inside its ALU. Now we want to modify the arithmetic unit of the simple MIPS processor and add another instruction. We add the **XOR** instruction to the ALU of the processor (for this you need to modify ALU to add one new opcode [*just modify **alu** and **alucontrol** modules in the mips.v file*] to be able to execute programs including XOR instruction). Simulate the modified design with Questa and verify the functionally of processor. Automatic simulation and verification with the TCL script that we saw in the class can help you with fast simulation using Questa.

XOR -- *Bitwise exclusive or*

Description:	Exclusive ors two registers and stores the result in a register
Operation:	$d = $s ^ $t; advance_pc (4);
Syntax:	xor $d, $s, $t
Encoding:	`0000 00ss ssst tttt dddd d--- --10 0110`

As can be seen in *mipstest.v* and *mipstest_xor.v* testbench files, to verify correct functionality of the MIPS processor, the data at address **EE** of memory after running the original Fibonacci program (*fib.dat*) must be **8'h0D** and after running the modified Fibonacci program (*fib_xor.dat*) which includes an XOR instruction must be **8'hF2**.

Please put the following files in a separate folder:

1. Upload you MIPS project folder. This should include your modified *mips.v* file which supports XOR instruction (name the new modified file *mips_xor.v*)

2. Catch the moment that the output of the program is written at memory address 238 (or 8'hEE) in Questa wave window (hint: you can track /top/dut/memwrite signal). Verify that the "writedata" signal has the expected output (8'h0D or 8'hf2). Take two screenshots from the wave window of Questa, one for the original operation of MIPS with *fib.dat* and the other for the modified MIPS with *fib_xor.dat*.

4. Upload your results on Moodle as *LASTNAME_ULID.zip*