PCI Project Description : 10 Marks
(Bonus Section At The End)

In this project, you are required to model the whole PCI communication protocol in verilog. You have two main modules :
1- Device
2- Arbiter

You need to create multiple instances of “Device” and let them communicate with the (only one) Arbiter.

The Arbiter should operate in a “Priority” mode.

For every “Device”, I want you to have a special input called “force_request” for testing purposes. In the real world, every device will request the bus based on its type and what it needs from the bus. However, in your simulation, I want this input to work as a trigger for the req output signal. This way, we can trigger any request from any device as we want. If the “force_req” signal is asserted for one cycle, then the device will want the bus for one transaction. If it was asserted for two cycles, the device should want the bus for two different transactions and act accourdingly, and so on.
Also, I want another signal (again for testing purposes) called “AddressToContact”. Whenever you assert the ”force_request” signal, you should also set the address of the target you want to communicate with using this signal. Each device must have a unique address, of course, and must know about its own address.

You are free to add any ports, parameters, or internal registers you want inside your modules that will help facilitate the operation of the bus components.

Let every device have an internal memory of 10 rows. For simplicity, assume that when some device tries to write data into another target, it always starts putting data from the very top of this memory element. If the initiator device wants to have multiple data words per transaction, then the data it sends will be stored sequentially in the memory of the target device. [First word in place 0 , second word in place 1, and so on]

You must have a complete test bench for the whole communication process. You must have at least 3 devices (A,B,C). A has highest priority, then B, then C is the least important.

Model the following scenario ::

1- Device A requests the bus in order to communicate with device B and send 3 data words in the transaction. Assume that Device A always sends the word “AAAAAAAA”. So after this, we should have 3 rows of device B having the values “AAAAAAAA”.

2- Device B requests the bus in order to communicate with device A and send two data words. Assume that device B always sends the word “BBBBBBBB”

3- Device C requests the bus for two transactions, and at the same time device A requests the bus again to communicate with Device C. What should happen here is as follows :

- The arbiter will grant device A since it has a higher priority, then device A will send two data words to device C.
- The arbiter must then grant the bus to device C, C now wants to send data to device A, send one word.
- Device C still wants the bus to communicate with device B, and send another data word to B. So it will have the bus for a second transaction since both A & B don’t request the bus any more

If this scenario works as expected, then you are good to go. Feel free to add any more complicated scenarios you want, you might get extra marks for more sophisticated scenarios. Also feel free to add any additional features you want. The more you add, the more you will be appreciated.

YOU MUST provide a document (report) for your project explaining in detail your design choices. Also this document should have explanation of your test cases in a detailed step-by-step fashion, showing at every step that your expected output is really equal to the output that your code produced. A good document on its own is worth 3 marks of the project marks, assuming of course that the project works properly.

BONUS SECTION :: (Marks & Money):

To get bonus, you must do the project on your own, with the minimum possible questions asked.

Dr Ashraf promised he would give 1000 L.E for the best team. Well, the definition of “best” is relative. If you want those 1000 L.E, do your best to add as many features as possible, everything you add will be considered. For example you can have two modes for the arbiter, one “FCFS”, and the other “Priority”. Also, You can for example create another software version of this whole process (Software simulation of the PCI BUS, having GUI elements using python/java/C#/QT or whatever, where you specify a sort of scenario to test and the program will draw the timing diagram cycle by cycle, showing (by colors) which objects are currently communicating with each other, all the signal values and so on. Feel free to add any more ideas, as long as you document them properly, and test them properly.

You will be given upto 5 extra bonus marks if you create an excellent software (GUI-Y) version of the PCI.

Number of students :(5-6 per team)
Deadline : 24/12/2018 - Delivery Details Later

Groups Document : Please fill this document with your Names. This is very important if you want to compete for the 1000 L.E Prize.
ALSO : The teams here will be the same teams for the MIPS Project. So choose your groups wisely, they will not be allowed to change.

https://docs.google.com/spreadsheets/d/1uGhZiQLv3Z5klvd0luw-UXqMj4YQbRGdlIGNLqNFrzY/edit?usp=sharing

I might update this document later if the need arises, This is an initial draft for you to start working.