DMA Project [10 marks + upto 17 bonus]

In this project, you are required to implement a very simple DMA controller. If you want a real-world DMA controller, you can check the 8237A controller here :
https://en.wikipedia.org/wiki/Intel_8237
And here is the datasheet.
https://pdos.csail.mit.edu/6.828/2018/readings/hardware/8237A.pdf

However, this is very complex to implement and test, and in this project you only need to implement an extremely simplified version of this. You can implement the 8237A [or some of its modes] for the bonus part. However, you will implement it completely on your own without asking me about any details. Asking me about details for the bonus part will automatically remove you from it. It is a hard task, but it will certainly make you a better engineer since this is a real-world project with very few tutorials out there. This is similar to what you will find while working in a real company after graduation isa.

You can find a version of the simplified DMA controller I request from you in this link :
https://hps.ece.utexas.edu/people/suleman/class_projects/soc_lab3.pdf

Use this link to get a feel of what I want you to do. I will not give you detailed descriptions or diagrams, I will just give you some functional requirements. You do not need to implement everything in this link also, just get a feel of how the world operates

- You need to have a processor module, a DMA controller module, a RAM module, and two IO device modules.
- The DMA controller can transfer data from memory to memory , from memory to IO, and from IO to memory.
- Use the memory-mapped IO scheme. So the address is shared between memory and IO devices. Every device should know what addresses belong to itself. For every IO device, assume there is 32 available addresses.
- In this project, you will need inout ports in verilog. For example the data can be transfered from processor to DMA, or from DMA to processor over the same 32-bit bus. You can learn about inout ports from this link :
https://docs.google.com/document/d/1RuQjonuT5VQVAgQRSKMDe7pln3IrcMX5Lc9J4ixDLqw/edit
- Now, here is a functional scenario that you must show me when you deliver the project :

1) The processor must be able to send and receive data to/from RAM or IO devices directly if it wants to.
2) The processor can choose to give this task to DMA controller if it wants.
3) Of course when DMA has control over the bus, the processor cant send data over the bus and vice versa. So you need to show me a scenario where the DMA is using the bus, and the processor wants to send more data over the bus but it waits until the DMA controller finishes its operation.
4) Feel Free to add any control signals and internal registers you want. This is completely a design choice.

5) The processor doesnot need to be a complete processor. It is just there to test the DMA module. So make it a minimal processor. I mainly need three main functionalities in this processor :
    a) The processor can send data to RAM or IO
    b) The processor can ask DMA controller to do so
    c) The processor can add two numbers and store the result in an internal register file.

6) Now I want you to show me that while the DMA is transferring data, the processor is actually doing some useful work (i.e , adding numbers). This is the main purpose of DMA, right ?

7) During project delivery, I expect you to show me a complete scenario showing me all those possibilities in one test case. Meaning, you show me the processor sending data directly to RAM while the DMA has no access to the bus, then the processor asking the DMA to transfer some data while doing some arithmetic operations while the DMA is transferring data. You must also show me a scenario where the processor tries to send data while DMA is transferring data, but the processor waits because the processor does not own the bus at this moment.

8) Again, feel free to add any control signals you want without asking me. This is your design, do all you need to do.

9) If you ask me about any details for this project, most probably I will just tell you “Etsarraf” , so : Etsarraf :D . I will appreciate any meaningful design choice, so don’t worry about that at all as long as you did something meaningful.

Delivery :

- When you deliver this project, I will ask you to show me the code along with sufficient test scenarios like the one I described before
- I also need you to create a simple power-point presentation explaining to me what you did while you deliver to me. This presentation shouldn't take more than 10 minutes to cover. Not all members need to speak during this presentation, I just need you to summarize what you have done.
- If you decide to implement the real 8237A or parts of it, your presentation can be extended to as long as 20 mins. If you do so, you will be required to explain to me all the details you have implemented and match that with the specifications in the datasheet.

Delivery time and BONUS :

- **Number of team members : 5 to 8.**

- **There** are three types of bonus for this project :
    - The early delivery bonus. If you deliver this project on **Friday 20/12**, you will get extra **3 marks as bonus**. The final deadline for delivery will be on **27/12.** So you have three weeks at most, and this is really a simple project that doesn't need that much time at all. I am giving you that as I am sure you must be busy these days with other subjects.

    - If you connect your verilog design to a software that will allow me to visualize everything, and choose data transfers as I want, and visualize the data transfer happening on the wires in a straightforward way, you will get upto extra **4 marks.**

    - **The real 8237A implementation bonus**. Well, this is much much harder than the basic project. If you do this or at least some of the modes of operation of this chip, you can get **upto 10 extra marks.** And of course you will have a better chance of winning the competition. The more you implement, the more bonus you get, and the better chance at the competition you have.

    - To enter the competition, you must deliver on Friday 20/12 , not on 27/12.

    - If no one does the real 8237A, deciding the winner will be based on the other requirements
    - FPGA implementation will not be a priority for this project at all. A better code [either verilog or software] will win this time.

    - **Please write your team-member names in this document :**

      https://docs.google.com/spreadsheets/d/1HWK6hJC9tIFKbjVt2WVqIfSi9h5CTHAC8cM1Of7Um5s/edit?usp=sharing

Good luck ！