Computer Engineering Dept.
Faculty of Engineering

Computer Architecture Lab #3

Objectives

After this lab, you should:

Understand VHDL basics:

1. Review on Conditional statements.
2. Combinational logic vs. registered logic processes.
3. Variables, Signals, and Constants.

Requirements

1. Build 4 32-bit registers connected to a single bi-directional bus.
2. Build 2 2x4 decoders one for the source and one for the destination to control reading & writing data to and from the bus.
3. Use the source decoder to control the Tri-State buffers and the destination decoder to control the enable of the registers.
4. Build a simulation wave form by performing the following steps:

Load r0 with data (00AA) from the bus as an input
Load r1 with data (00BB) from the bus as an input
Load r2 with data (00CC) from the bus as an input
Load r3 with data (00DD) from the bus as an input
Transfer data from register r0 to r1
Transfer data from register r2 to r0
Transfer data from register r3 to r0

Guidelines

- High impedance is assigned to a signal by using the following syntax: x <= 'Z'
- **Decoders must have an enable.** If the enable isn't asserted the o/p will be "0000", hence no register will be reading(or writing) from the bus at this time
- An inout bus can be used as **input to many** resources in same time, but only **one resource can out** on it at a time. To avoid such a problem, any resource sharing this bus must out 'Z' on it as soon as it finishes outing on it, this statement holds for all registers dealing with this inout bus.
- If 2 or more inputs/signals are writing on the same bus the data will be corrupted (**collision**) and will appear as 'X' in the simulation.
- The last statement is applied to the user of the simulator as well when using the inout bus to initialize the registers, when he/she finishes he/she **must release the bus** by -force Z or noforce the bus- so that other users, other registers in this case, can out on the bus without facing any problems.
- Be aware of the **starting conditions (Initializing inputs & using "rst")** of your circuit to avoid going in an undefined state in case of contention on the inout bus.

N.B. you will be graded for code neatness (indentations, variable names, …)
Good luck.