وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا

Ain Shams University – Faculty of Engineering – ECE Dept. – Integrated Circuits Lab.
Dr. Hesham Omran

Digital Design and Computer Architecture (DDCA)
Lab 04
Design of Pipelined RISC-V Processor

Objectives

1. Design a pipelined processor datapath and control logic.
2. Acquire a deep understanding of the operation of a pipelined processor.
3. Design and verify a complex digital system using HDL.
4. Run and debug logic simulations.

Submission and Peer-Grading Instructions

1. **DO NOT WRITE YOUR NAME ON YOUR SUBMITTED WORK. YOUR SUBMISSION WILL BE GRADED ANONYMOUSLY.**
2. Submit your solution on Canvas as a single zip file that contains all the deliverables.
3. You may discuss the questions together, but you are NOT allowed to share the answers.
4. If you cheat, attempt to cheat, or help someone else to cheat you will get ZERO marks.
5. Remember: قال رسول الله صلى الله عليه وسلم: من غش فليس منا
6. You will be invited to review three anonymous submissions after the due date.
7. You must complete the reviews within one week of the deadline. You must be fair and justify the grade you give in detail. Give the grade (for every deliverable) and justify it as a comment on Canvas. Do NOT attach extra files for your grading.
8. **If you do not complete the grading (with clear justifications) within one week of the deadline, 20% of the full mark will be deducted from your mark.**
9. **10% of the full mark will be deducted for every late day after the deadline (in addition to losing the peer-review marks).**
10. دعواتي لك بالتوفيق

Lab Instructions

1. Use the same instructions given in Lab 03.
2. Assume your processor is divided into five pipelined stages (IF, ID, EX, ME, WB).
3. You will implement a hazard unit as a standalone module. The hazard unit control signals should be connected directly to the datapath (you may use two-level muxes to combine the control logic and hazard unit signals).

Deliverables

Index	Deliverable	Points
1.	Implement your 5-stage pipelined processor without the hazard unit. Report all your HDL modules. **Hint: Read Section 7.5.3 in the textbook (DDCA 2nd edition by Harris and Harris) as it will be a good starting point.**	4
2.	Use Venus to generate the machine code for a Fibonacci sequence program. **Add nop's to your code to avoid any pipelining hazards.** Load the machine code in your	3

	instruction memory. Simulate the processor to calculate the 6th Fibonacci number (= 8). Report your assembly code. Report your testbench. Report your simulation results clearly.	
3.	Write a module for a hazard unit that uses stalling and forwarding to avoid data hazards. Modify the datapath accordingly. Use Venus to generate the machine code for an arbitrary sample program that includes several types of data hazards (test both register and load hazards). Report your HDL code. Report your assembly code. Report your testbench. Report your simulation results clearly.	3

Reference

Harris D and Harris S, Digital design and computer architecture, Morgan Kaufmann, 2nd ed., 2012.

Thanks to all who contributed to these labs. If you find any errors or have suggestions concerning these labs, please contact Hesham.omran@eng.asu.edu.eg.