

---

# ECE 436 Laboratory 6

---

## PIPELINE DESIGN AND HAZARD RESOLUTION

---

### Description:

Now that you have a cache design for your processor, memory latency is (hopefully) no longer such a bottleneck in the execution of programs. To further improve your processor's performance, instruction concurrency can be explored. The two different paths to concurrent execution are parallelism and pipelining. We will explore the latter in this lab, but you are encouraged to consider the former for your group design.

When exploring pipeline depth, you must consider a variety of factors. For example, the deeper the pipeline, the larger the area overhead, the more difficult it is to divide and balance the stages, and the increased frequency of hazards (resource, data, and control). On the other hand, deeper pipelines allow for more instruction concurrency and shorter stage delay (resulting in higher clock frequencies). These tradeoffs must be considered when searching for the best implementation of your system.

As you develop your pipeline design, it is important to recognize and resolve all of the hazards and stall conditions that the pipeline may introduce. Resource, data, and control hazards can cause stalls and/or wasted cycles, but these penalties can be reduced or eliminated by various techniques (e.g. data forwarding, branch prediction, etc.). However, these techniques often come at a cost (increased area, delay, complexity), which must be balanced against the benefits they provide.

For this lab, you are to design a pipeline that gives you the maximum performance (considering CPI and clock frequency) on at least two testbench programs from the Toolkit website. Clearly identify the functionality and required resources for each stage. In addition, you should identify all of the potential hazards that exist and how you are going to resolve them. Consider CPI, area, delay, and complexity in your analysis to justify your design decisions.

For your CPI analysis, you may assume certain cache sizes and branch prediction mechanisms, but you must clearly state those assumptions. You may handle the various hazards in any number of ways (e.g. data forwarding, stalling, etc.), but again clearly state your methods. For your clock frequency analysis, state your assumptions about the delays of individual components (use the Xilinx CAD tools for more accurate assumptions) while determining your worst-case delay.

### Deliverables:

We will be expecting the following items in the laboratory report:

- A schematic clearly showing your datapath, the distinct pipeline stages, and any hazard resolution logic (e.g. data forwarding routing and components).
- A list and explanation of the various hazards introduced by your pipeline design.
- A maximum clock frequency estimate based on your longest pipeline stage.
- A detailed CPI analysis for running the testbenches on your pipeline design (including a clear list of the assumptions you have made in the analysis).
- Explanations and support for the design decisions you made (e.g. CPI analysis, delay information, complexity analysis, etc.).