Muplitlier!

Digital Logic Design Final Project - Semester 4031

Overview

The goal of this project is to design approximate bit multiplier which we decided to name it Muplitlier!.

Step 1:

Design a 1 by 1 multiplier module.

Step 2:

Design a 4 by 4 multiplier module with four 1 by 1 modules.

Step 3:

Design a 2 by 2 multiplier.

Step 4:

Design a 4 by 4 multiplier with two 2 by 2 modules.

Step 5:

Find a way to optimize the modules provided in steps 3 and 4 with *approximation*. (Hint: include probability and statistics and use some creativity!)

Approximate Computing

From Wikipedia, the free encyclopedia:

> Approximate computing is an emerging paradigm for energy-efficient and/or high-performance design. It includes a plethora of computation techniques that return a possibly inaccurate result rather than a guaranteed accurate result, and that can be used for applications where an approximate result is sufficient for its purpose. One example of such situation is for a search engine where no exact answer may exist for a certain search query and hence, many answers may be acceptable. Similarly, occasional dropping of some frames in a video application can go undetected due to perceptual limitations of humans. Approximate computing is based on the observation that in many scenarios, although performing exact computation requires large amount of resources, allowing bounded approximation can provide disproportionate gains in performance and energy, while still achieving acceptable result accuracy. For example, in k-means clustering algorithm, allowing only 5% loss in classification accuracy can provide 50 times energy saving compared to the fully accurate classification.
>
> The key requirement in approximate computing is that approximation can be introduced only in non-critical data, since approximating critical data (e.g., control operations) can lead to disastrous consequences, such as program crash or erroneous output.

Further Reading: https://en.wikipedia.org/wiki/Approximate_computing

Deliverables

- VHDL code
- Test-bench(es)
- Simulation results: like waveform etc
- Documentation containing the following:

- Design Document: Detailed description of each module, please avoid unnecessary descriptions.
- This section may or may not exist ;D guess it yourselves.
- Testing Report: Summary of the tests conducted and their results.
- User Guide: Brief guide explaining how to use the project.

Evaluation Criteria

- Design completeness (30%): Adherence to all project requirements and functionalities.
- VHDL code quality (10%): Code organization, readability, modularity, and adherence to good coding practices.
- Testing and validation (10%): Quality of test-benches and evidence of successful testing.
- Documentation (10%): Clarity and completeness of design and testing documentation.
- Efficiency and innovation (40%): Creative and optimized solutions, and any additional/interesting features (e.g., error reporting or handling,).
- Statistic results and charts (10%).
- One hidden bonus! Do this project to find out ;D

Submission Guidelines

Until Sat 1st Feb 2025, 23:59

E-mail: mkoushan@macaw.me

SUBJECT: DLD4031_FP

BODY: *STUDENT_NO*