CSC110 Final Project Proposal Write up

Ricky O’Connor

Alex Pateroulakis

problem description:

The combinational circuits contain boolean operations for input and output. Classic examples of combinational circuits are half adders and full adders. Half adders operate by taking 2 inputs and producing a sum and carry output. By combining half adders we can get a full adder that not only adds 2 inputs, but also adds a carry. Just as we combined half adders to make a full adder, full adders can be connected in series. In this series the carry bit “ripple” from one adder to the next, which is why this process is called the ripple-carry adder. This is the most basic method of a circuit for addition. Our method proposes a faster way. Instead of adding the numbers and carries in series, our method suggests we add all the numbers in parallel and store the sum as a separate number. Then we would add the sum of the two numbers with the separate carry number, and repeat this process until the separate carry number becomes all 0.

Our goal for this method is to reduce the amount of cycles required by the CPU to perform binary addition. We would want to do this because by reducing the amount of cycles, we can reduce the amount of time taken to perform an operation and overall reduce the amount of delay in between multiple operations. We want to improve upon the efficiency of performing these operations, compared to the ripple-carry method and other methods that have been tested and researched that aimed to improve upon the ripple-carry method. One of the main points of focus is to test the validity of our method on all sets of numbers. Our method may not significantly reduce the amount of cycles for smaller sets of numbers, however it may make a huge difference with larger sets of numbers.

Distribution of work among team members:

Ricky O’Connor - Programing research

Alex Pateroulakis - Topic research

Timeline:

Every Mondays - 5:00pm - 10:00pm

Every Wednesday- 5:00pm - 10:00pm

Every Friday- 5:00pm - 10:00pm

	Monday	Wednesday	Friday
Week one	Research topic	Research topic	Programming
Week two	Programming	Research topic	Research topic
Week three	Programming	Finalize	Finalize