Streaming 2D Convolution Accelerator: FPGA A* vs Python A*

Changwe Musonda
Electrical and Computer Engineering at California State Polytechnic University, Pomona
cbmusonda@cpp.edu

Kevin Wang
Electrical and Computer Engineering at California State Polytechnic University, Pomona
cbmusonda@cpp.edu

Emily Morales
Electrical and Computer Engineering at California State Polytechnic University, Pomona
emorales11303@gmail.com

Abstract - This project seeks to compare the performance of an FPGA A* versus a reference Python A*. We ran the code under an identical map, start/goal pairs, and heuristic settings. The path is run on a 16x16 grid with 8 different directional movements and a Manhattan heuristic. In order to visualize the path and performance, a python 2D visualizer is implemented, its display allows for visualization of errors and path routing. In order to help with cost effectivity, we have decided that the required hardware would be a Nexys A7 board

****Add a brief explanation of results and the conclusion***

Index terms - A*, FPGA, Manhattan heuristic, pathfinding acceleration

- ***4-8 pages***

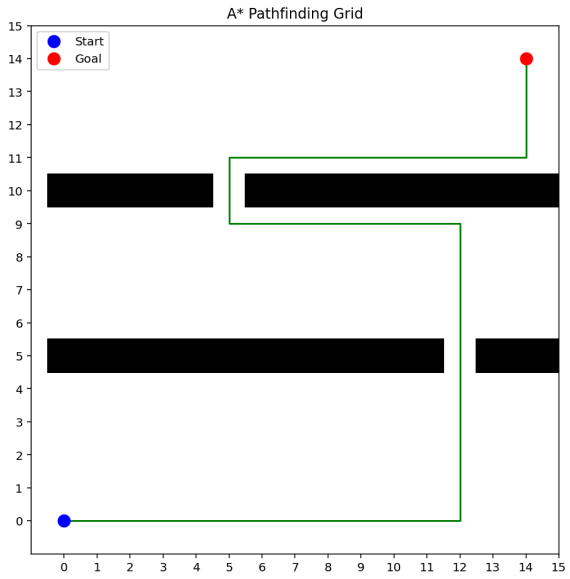
I. INTRODUCTION

A. Hardware Acceleration and Applications

There is a need for pathfinding acceleration in a number of different applications. It is used in video games, robotics, VR/AR, and a lot of other examples. One of the reasons its application is so important, is that we are able to use specialized hardware such as an FPGA in our case, to significantly improve the speed and performance of the product.

B. FPGAs in Hardware Acceleration

In our case, we decided to go with an FPGA and compare the speed of the path routing. A few benefits to using FPGAs for acceleration are that it consumes a relatively small amount of power, is easy to reconfigure, and allows for customizable logic [1]. After deciding on our FPGA, we continued by working on our A* algorithm and implementation of the Manhattan heuristic. The Manhattan Heuristic allows the algorithm to find the shortest path by moving diagonally versus only being able to move horizontally and vertically.



CONCLUSION AND RESULTS

Insert a table to compare python and verilog*

To be completed, list of references

1. https://www.geeksforgeeks.org/computer-vision/hardware-acceleration-for-computer-vision-algorithms-1/
2. https://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html