Homework #2

CSC456-005

Cameron Hetzler

Due: 4/4/2018

**Time comparison table of different sorting methods**

|  |  |  |
| --- | --- | --- |
| Mode | time(ns) | delta (compared to mode 0) |
| 0 | 16,308,343 | 0 |
| 1 | 16,001,251 | -307,092 |
| 2 | N/A | N/A |

Time Complexity

I tried writing a “bottom-up” variant of the merge sort due to complications with recursion and stack-based operations on the FPGA board. My algorithm would have a worst-case scenario, Big-O, of O(n\*log(n)).

Performance Improvement

In my implementation, I set it up so that the board would be used as a single core and I wouldn’t have to work with different work groups and local memory changing, because of this I normally wouldn’t think the FPGA would work any faster than a single threaded workload on the ARM processor. Since the FPGA is being programmed for this specific task though and wouldn’t be limited to the ARM instruction set, I also wouldn’t be surprised if it ran better due to the lack of overhead once compiled.

Comparing Performance

I did not in fact use the same algorithm as I did with pthread, where I used recursion. Due to the lack of stack on the FPGA I had to rethink my process, which is why I decided to use a single threaded “bottom-up” algorithm instead as it was easier to implement with workgroups and didn’t require the use of stack. There was less overhead in my new algorithm so I would like to think it was able to run faster because of this.

NOTE:

The kernel itself doesn’t work. I couldn’t get it to work after spending some time on it just compiling. I was receiving a “Error: Internal Compiler Error” after 45 minutes of compiling. I assumed this was because of a lack of memory due to using two arrays that are 64MB each when the board also only has 64MB of memory. I didn’t know how to deal with this and tried getting rid of one of them and hard programming in a local array of size 16777216 into the kernel. After I did this the compiler worked for nearly two hours before my VCL connection timed out and I lost my work.