Class: AMATH 583 Name: Hongda Li HW3 Questions.rst

Quetion 1

What level of SIMD/vector support does the CPU your computer provide?
List of SIMD provided by 4900HS:

1. AVX2, AVX
2. SSE, SSE3, SSSE3 SSE4.1, SSE4.2, SSE2

The SSE is just old type of AVX.

Questions 2, 3

What is the maximum operand size that your computer will support?
A in this case, AVX 2 provides the longest register length for data, in this case it's 256 bit.

Question 4

What is the clock speed of your CPU? (You may need to look this up via "About this Mac" or "lscpu").
Base: 3.0 GHz, Boost: 4.3 GHz, usually I get 3.9GHz.

Question 5

Based on the output from bandwidth.exe on your computer, what do you expect L1 cache and L2 cache sizes to be? What are the corresponding bandwidths? How do the cache sizes compare to what "about this mac" (or equivalent) tells you about your CPU? (There is no "right" answer for this question – but I do want you to do the experiment.)

This is the verbatim of the out of from "./bandwidth.exe" command:

```
    read
bytes/elt       #elts   res_bytes   ntrials          usecs      ttl_bytes      bytes/sec
        8          16         128  67108868          15000     8589935104    5.72662e+11
        8          32         256  33554436           8000     8589935616    1.07374e+12
        8          64         512  16777220           4000     8589936640    2.14748e+12
        8         128        1024   8388612           1000     8589938688    8.58994e+12
        8         256        2048   4194308              0     8589942784              0
        8         512        4096   2097156           1000     8589950976    8.58995e+12
        8        1024        8192   1048580              0     8589967360              0
        8        2048       16384    524292              0     8590000128              0
        8        4096       32768    131074              0     4295032832              0
        8        8192       65536     65538              0     4295098368              0
        8       16384      131072     32770              0     4295229440              0
        8       32768      262144     16386              0     4295491584              0
        8       65536      524288      8194              0     4296015872              0
        8      131072     1048576      2049              0     2148532224              0
        8      262144     2097152      1025              0     2149580800              0
        8      524288     4194304       513              0     2151677952              0
        8     1048576     8388608       257              0     2155872256              0
        8     2097152    16777216       129              0     2164260864              0
        8     4194304    33554432        65              0     2181038080              0
        8     8388608    67108864        33              0     2214592512              0
        8    16777216   134217728        17              0     2281701376              0
write
bytes/elt       #elts   res_bytes   ntrials          usecs      ttl_bytes      bytes/sec
        8          16         128  67108868          62000     8589935104    1.38547e+11
        8          32         256  33554436          62000     8589935616    1.38547e+11
        8          64         512  16777220          61000     8589936640    1.40819e+11
        8         128        1024   8388612          61000     8589938688    1.40819e+11
        8         256        2048   4194308          61000     8589942784    1.40819e+11
        8         512        4096   2097156          61000     8589950976    1.40819e+11
        8        1024        8192   1048580          61000     8589967360    1.40819e+11
        8        2048       16384    524292          61000     8590000128     1.4082e+11
        8        4096       32768    131074          32000     4295032832     1.3422e+11
        8        8192       65536     65538          33000     4295098368    1.30154e+11
        8       16384      131072     32770          31000     4295229440    1.38556e+11
        8       32768      262144     16386          31000     4295491584    1.38564e+11
        8       65536      524288      8194          32000     4296015872     1.3425e+11
        8      131072     1048576      2049          22000     2148532224    9.76606e+10
        8      262144     2097152      1025          23000     2149580800      9.346e+10
        8      524288     4194304       513          51000     2151677952    4.21898e+10
        8     1048576     8388608       257         364000     2155872256    5.92273e+09
        8     2097152    16777216       129         351000     2164260864    6.16599e+09
        8     4194304    33554432        65         366000     2181038080    5.95912e+09
        8     8388608    67108864        33         360000     2214592512    6.15165e+09
        8    16777216   134217728        17         375000     2281701376    6.08454e+09
read/write
bytes/elt       #elts   res_bytes   ntrials          usecs      ttl_bytes      bytes/sec
        8           8         128  67108868         108000     8589935104    7.95364e+10
        8          16         256  33554436         130000     8589935616    6.60764e+10
        8          32         512  16777220          65000     8589936640    1.32153e+11
        8          64        1024   8388612          44000     8589938688    1.95226e+11
        8         128        2048   4194308          37000     8589942784    2.32161e+11
        8         256        4096   2097156          34000     8589950976    2.52646e+11
        8         512        8192   1048580          32000     8589967360    2.68436e+11
        8        1024       16384    524292          31000     8590000128    2.77097e+11
        8        2048       32768    131074          17000     4295032832    2.52649e+11
        8        4096       65536     65538          32000     4295098368    1.34222e+11
```

```
8        8192      131072     32770          32000     4295229440    1.34226e+11
8       16384      262144     16386          33000     4295491584    1.30166e+11
8       32768      524288      8194          33000     4296015872    1.30182e+11
8       65536     1048576      2049          21000     2148532224    1.02311e+11
8      131072     2097152      1025          20000     2149580800    1.07479e+11
8      262144     4194304       513          44000     2151677952    4.89018e+10
8      524288     8388608       257         265000     2155872256    8.13537e+09
8     1048576    16777216       129         264000     2164260864    8.19796e+09
8     2097152    33554432        65         262000     2181038080    8.32457e+09
8     4194304    67108864        33         265000     2214592512    8.35695e+09
8     8388608   134217728        17         203000     2281701376    1.12399e+10
```

The compiler optimized out the read forloop, making the results invalid. I currently don't have a good idea on how to get the read speed yet.

For the write, a Plateau appeared between the block size of "65536" to "131072", indicating a threshold of 2^{16} bytes for the L1 registers. Which is about "64kb" L1 Cache. The second Plateau appeared between size "524288" and "1048576 " bytes, meaning that L2 could have a size of "512"kb. The third plateau happens around 8mb, and that could potentially be entering the L3 cache.

For the read/write bandwidth to register, it's a different story. It seems like, for blocks that are too small, it's unable to achieve optimal speed, and the speed drops out, around "256kb".

L1 Cache has speed around "1.3425e+11"bytes/s, which is about "134.25 gb/s". And the decrease L2 speed is about "90 gb/s", and for L3, it's about "6gb/s".

The specs of the Ryzen 4900HS has 512kb for L1 register, and 4.0mb for L2 Register, and I think L3 is shared between 2 cluster of cores so we are not worrying about that.

Therefore, it seems like there are 2 types of speed for L1 cache for this CPU, I mistaken the first Plateau as falling L1 to L2, but actually it's not.

I googled a bit, and here is the source for 4900HS cache organization.

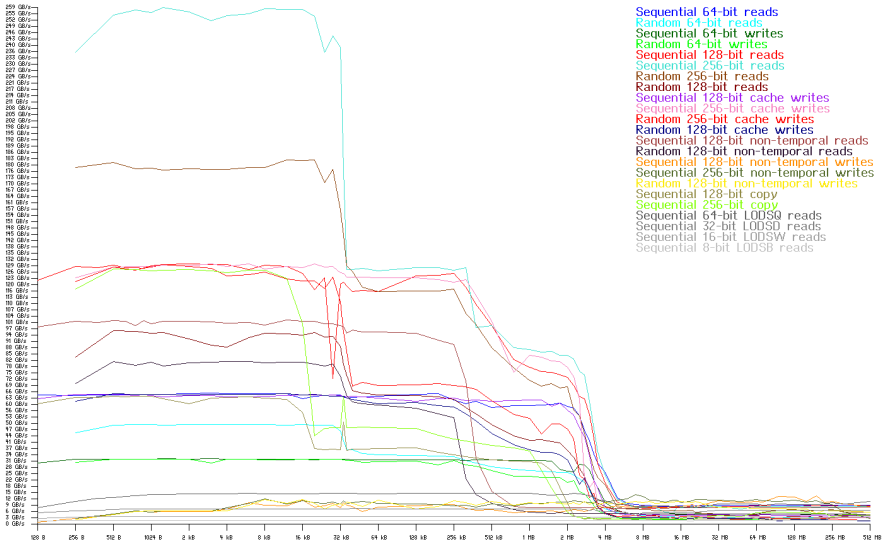
For L1, there are 2 types of L1, L1L, L1D, and for L2, it's 512kb shared for one core while 4 ways stitched together for 4 coers.

L3 is 2 times 4 mb (8 mb in total) and shared for all 8 cores.

Note: The Ryzen zen 2 CPU should also have a FMA4 (Fused, Multiply Add) SIMD instruction set for floating operations, however, it's not shown when "./cpuinfo583.exe" is executed.

Questions 6

Based on the output from bandwidth.exe on your computer, what do you expect L1 cache and L2 cache sizes to be? What are the corresponding bandwidths? How do the cache sizes compare to what "about this mac" (or equivalent) tells you about your CPU? (There is no "right" answer for this question – but I do want you to do the experiment.)



There are 3 drops in the data bandwidth from the experiment, at the size of 32kb, 512 kb, and 4mb. Each of them corresponds to part of the L1 cache, all of the L1 cache, L2. And by the time we entered L3 cache (around 4mb), all the speed is gone.

Question 7

What is the (potential) maximum compute performance of your computer? (The horizontal line.) What are the L1, L2, and RAM bandwidths? How do those bandwidths correspond to what was measured above with the bandwidth program?

1. Assume single core performance, The top speed is computed by:

   $$(\text{Core Speed}) \times (\text{Instruction per clcye}) \times (\text{Number of Floats it can process with SIMD})$$

   hence the number will be given by:

   $$4.3 \times 8[\text{GFlops}/s] \approx 34.4[\text{Gflops}/s]$$

   where the data "8" is for double precision floating points, using the "FMA4" instruction for ryzen processor. However, if the "256" bits register is used with the "AVX2" SIMD, then the data Gflops will be $4.3 \times 4 \approx 17.2$. With both contributing together, we can have: $4.3 \times 8 + 4.3 \times 4 \approx 51.6$ Gflops per second.

   I will say this is a very gross over estimate (It hardly even goes to the advertised 4.3 GHz to be honest.), and I will say it's about "15" Gflops under the best case for my CPU.

   This part of the HW Questions is done with the help from this stack overflow discussion.

2. L1: "255 GB/s", L2 is "88 GB/s", RAM is "9 GB/s", this is for single core I would assume.
3. My PC manufracture didn't specify any of this information, there is no comparison to make.

Question 8

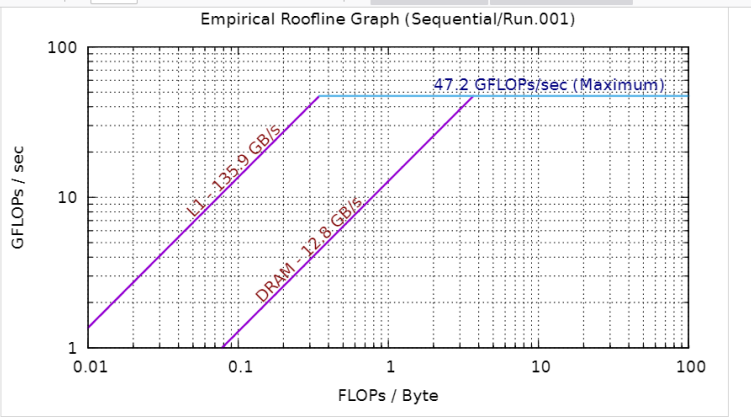
Based on the clock speed of your CPU and its maximum Glop rate, what is the (potential) maximum number of *double precision* floating point operations that can be done per clock cycle? (Hint: Glops / sec

:math:‘/div‘ GHz = flops / cycle.) There are several hardware capabilities that can contribute to supporting more than one operation per cycle: fused multiply add (FMA) and AVX registers. Assuming FMA contributes a factor of two, SSE contributes a factor of two, AVX/AVX2 contribute a factor of four, and AVX contributes a factor of eight of eight, what is the expected maximum number of floating point operations your CPU could perform per cycle, based on the capabilities your CPU advertises via cpuinfo (equiv. lscpu)? Would your answer change for single precision (would any of the previous assumptions change)?

1. Attempts on computing the maximal performance of the CPU is already been shown in the first part of question 7.
2. My answer will change for single precision. The AVX, or the FMA instruction set can be divided up for streaming single precision numbers, this allows for the performance to double when computing single precision floating points compare to double precision floating points.

Question 9

What is the maximum compute performance of your computer? (The horizontal line.) What are the L1, L2, and DRAM bandwidths? How do those bandwidths correspond to what was measured above?



The maximal compute performance is 47.2 GFlops per second. The L1 is 135.9 GB/s, the same as we observed from the homegrown benchmark. The L2 Cache bandwidth is gone under the eyes of this profiler or unknown reasons.
They are very similiar to the measurements above. The DRAM bandwidth is the same as results from the “amath583/bandwidth” docker run.