Eric Paulz
CPSC 3300-002

Homework 1

1. Moore's Law

a.) Define exponential growth.

Growth whose rate becomes ever more rapid in proportion to the growing total number or size. (Dictionary.com)

b.) What did the original Moore's Law observe and project?

Moore's original projection was that the number of components per integrated circuit would double approximately every year.

c.) In your opinion, why has Moore's prediction been accurate over the years?

I believe that it has remained accurate because computers have and will continue to be incredibly useful tools for humanity. As we continue to progress, more and more computational capability will be required to meet our demands, therefore providing an incentive for chip companies to find ways to continue improving their products. Also, there is possibility for significant financial gain for those invested in these companies, since the use of computers appears to be going no where but up.

d.) What is Dennard Scaling and why is it important in processor technology evolvement.

It is a law which states that as transistors get smaller their power density stays constant, so that the power use stays in proportion with area: both voltage and current scale with length. It is important because if the power use scaled much slow than the size of transistors, modern computers with their millions and millions of transistors would consume an enormous amount of power and likely slow down the rate of progress significantly.

e.) According to Dr. Mack, scaling down or miniaturization marked the Moore's Law 2.0 era. Scaling down reduces the size of transistors. List the feature sizes over the years to today.

1961: 4
1965: 64
1971: 2,300
1975: 3,510
1985: 25,000
1991: 30,000
1999: 9,500,000
2007: 26,000,000
2011: 1,160,000,000
2013: 5,000,000,000

f.) In your opinion, why are people discussing whether Moore's Law is dead or not.

I think this is due to the fact that our chips have become so advanced that it is difficult to imagine progress continuing at the rate that it has been. Eventually, we will reach a point where transistors simply can't get any smaller, and that is probably why people believe that the rate of growth will slow down significantly in the near future.

Eric Paulz
CPSC 3300-002

2. Perf and Time Tools

a.) perf

user

```
[17:01:52] epaulz@troll14:~/spring18/3300/hw1 [24] perf stat -e instructions:u
-e cycles:u ./whetstone 200000

Loops: 200000, Iterations: 1, Duration: 5 sec.
C Converted Double Precision Whetstones: 4000.0 MIPS

 Performance counter stats for './whetstone 200000':

    22,853,012,319      instructions:u            #    1.54  insns per cycle

    14,863,016,151      cycles:u

       5.043677274 seconds time elapsed
```

kernel

```
[17:03:43] epaulz@troll14:~/spring18/3300/hw1 [25] perf stat -e instructions:k
-e cycles:k ./whetstone 200000

Loops: 200000, Iterations: 1, Duration: 5 sec.
C Converted Double Precision Whetstones: 4000.0 MIPS

 Performance counter stats for './whetstone 200000':

        28,716,032      instructions:k            #    0.93  insns per cycle

        30,913,085      cycles:k

       5.060027380 seconds time elapsed
```

b.) time

1.) Explain the timing output and the definitions.

real 0m10.893s - amount of time that passes on a wall clock between the start and finish of the program call. Includes waiting for I/O and other processes.

user 0m10.888s - amount of time the CPU spend in user-mode code (outside the kernel) within the process. Does not include other processes like ‘real’ does.

sys 0m0.000s - amount of time the CPU spent in the kernel within the proccess. Also does not include other processes.

2.) If the timings from perf and time are different, explain the cause.

‘time’ recorded a time of 5.09s and ‘perf’ recorded a time of 5.06s. I would venture to say that this difference could be considered negligible. However, perhaps ‘time’ was slightly longer because it is tracking every single event that occurs while the program is running while ‘perf’ is set specifically to track just instructions and cycles.