EECS 151/251A Homework 1

Due Monday, Feb 4th, 2019

Problem 1: Moore's Law

Consider state-of-the-art processor chips from the 1970's, 1980's, 1990's, 2000's, and after 2010. Choose a processor from each period. (You may choose which every processor you like, but make sure they are spaced out by around 10 years. Use of Wikipedia or WikiChip is acceptable.)

1. For each look up the approximate number of transistor per chip. Plot the number of transistors per chip over time, with a log scale on the y-axis.
2. For your processor choice after 2010, approximate the ratio of transistors used for on-chip memory (caches) vs logic circuits. Assume 6 transistors per memory bit.
3. List a major product innovation enabled by each of these processors.
4. On another set of axis, plot the clock frequency of each over time (on a log plot).

Problem 2: Design Tradeoffs

Sketch plot showing what you expect the design space of a processor looks like in terms of power (watts) and performance (frequency). Show several possible design points as dots in the design space including some on the Pereto Optimal frontier. Also, draw the Pareto Optimal frontier.

Problem 3: 251A only — *Optional* Challenge Question for 151

Dennard's scaling says that if all transistor dimensions and the operating voltage is scaled linearly, the the resulting transitors will speed up linearly and that power density (power per unit area) will remain constant. However, as Moore's Law progressed and circuits were scaled down, processor chip performance and power both increased at a rate faster than linear. How can you explain why frequencies scaled at a rate higher than linear with process scaling. What was the effect of this frequency scaling on power consumption?

Problem 4: Full Custom versus Standard Cell

Consider custom chip design with the two methods described in class: Standard Cells and Full Custom.

Breifly describe for each the main advantage and main disadvantage relative to the other.

Version: 2 - 2019-01-25 15:50:46-08:00

Problem 5: Implementation Alternatives

Consider the following implementation approaches for building some digital system: Full Custom, Standard Cell, Gate Array, FPGA, micro-processor.

For the best implementation possible, rank order (a partial ordering might be okay) from highest to lowest a) for NRE costs, b) per part costs, c) performance, and d) flexibility.

Problem 6: Combinational Logic

Consider a combinational logic block that multiplies two N-bit numbers yeilding a 2N-bit result. How many rows are in the truth table that represents the function of this block? How many rows are there when N=8?

Problem 7: Combinational Logic Functions

Consider the set of CL blocks with N inputs and 1 output. How many unique such blocks exist?

Problem 8: Combinational Logic Gates

In class we said that any combinational logic function can be implemented with either AND-gates along with inverters or with OR-gates along with inverters. The hypothesis is that NAND-gates alone or NOR-gates alone are sufficient to implement any combinational logic function. Proof or disprove this hypothesis.