EECS 151/251A Homework 1

Due Monday, Feb 4th, 2019

Problem 1: Moore's Law [20 pts]

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 [5 pts].
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 [5 pts].
3. List a major product innovation enabled by each of these processors [5 pts].
4. On another set of axis, plot the clock frequency of each over time (on a log plot) [5 pts].

Solution:

Chip	Year	Transistors	Clock Freq. Low (MHz)	Clock Freq. High (MHz)	Key Advance	Src
Intel 8080	1974	6K	2	3.125	One of the first widespread microprocessors owing in part to its 40-pin package, faster transistors (NMOS), compatibility with TTL, and and increased memory bus size (with ability to access 64 KB of memory)	https://en.wikipedia.org/wiki/Intel_8080
Intel 80386	1985	275K	12	40	First Implementation of 32-bit extension to 80286. A common ancestor of modern x86 processors and is also known by the name i386.	https://en.wikipedia.org/wiki/Intel_80386
Intel Pentium P55C	1997	4.5M	120	233	First Implementation of MMX (SIMD) extension in Intel Processors. Predecessor to SSE and AVX vector extensions	https://en.wikipedia.org/wiki/P5_(microarchitecture)#MMX
Intel Prescott	2004	125M	2400	3067	First of the Pentium line to introduce SSE3. Major overhaul that hit the heat barrier	https://en.wikipedia.org/wiki/Pentium_4, https://en.wikipedia.org/wiki/List_of_Intel_Pentium_4_microprocessors#Prescott_(90_nm)
Intel Ivy Bridge-E	2013	1.86B	3400	3600	Die shrinking of Sandy Bridge-E to 22 nm (finfet)	https://en.wikipedia.org/wiki/Ivy_Bridge_(microarchitecture), https://en.wikipedia.org/wiki/List_of_Intel_Core_i7_microprocessors#%22Ivy_Bridge-E%22_(22_nm)

Intel Ivy Bridge-E Details:	
Num Transistors	1860000000
Num Cores	6
L1 Cache Per Core (Bytes)	32768
L2 Cache Per Core	262144
L3 Cache Shared (Bytes)	15728640
Cahce Total (Bytes)	17498112
Cache Total (Bits)	139984896
Transistors Per Bit (Est)	6
Cache Transistors (Est)	839909376
Logic Transistors (Est)	1020090624
Ratio Memory/Logic	0.823367411

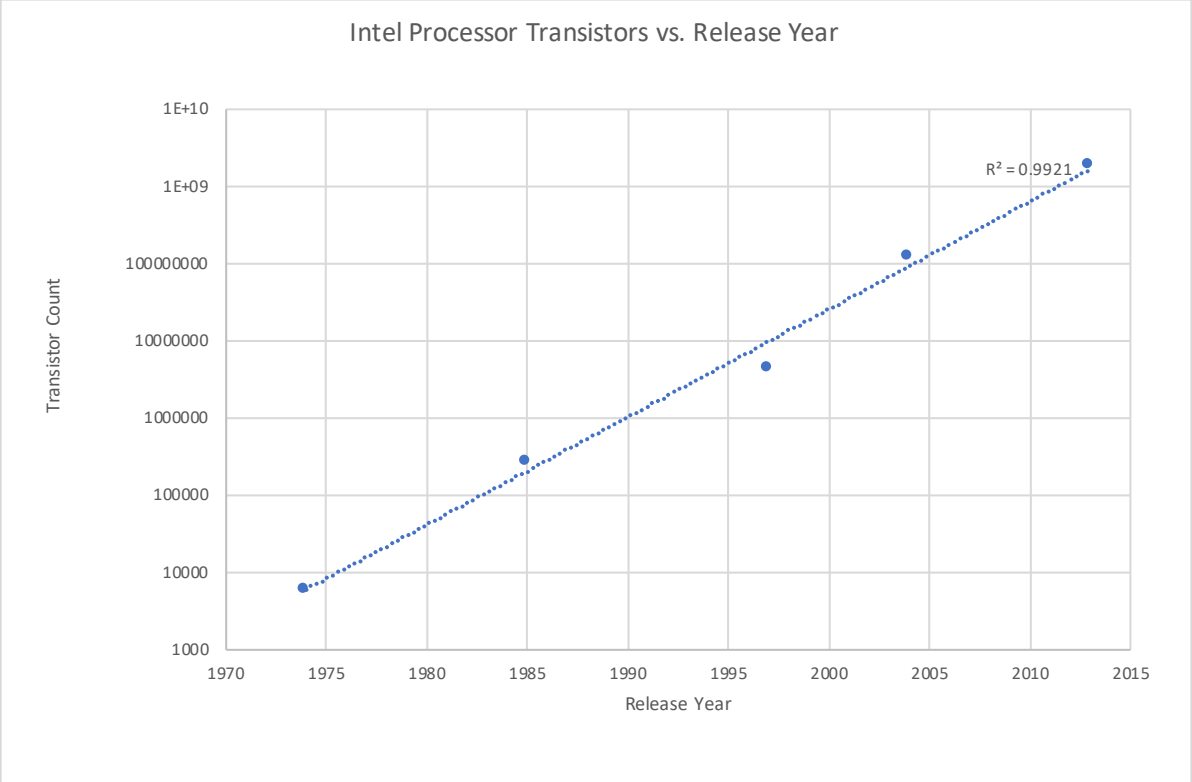

Figure 1: Transistors Trend

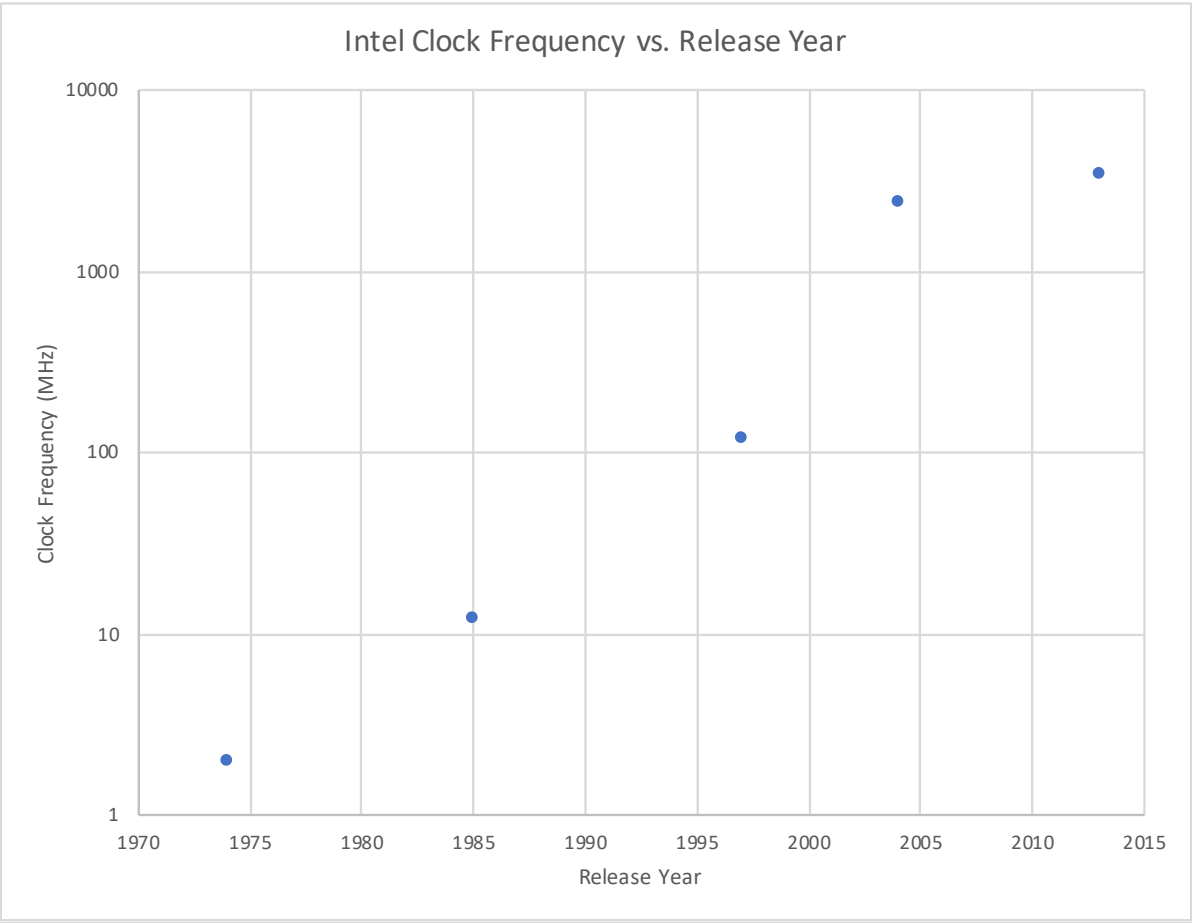

Figure 2: Frequency Trend

Problem 2: Design Tradeoffs [5 pts]

Sketch a 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.

Solution:

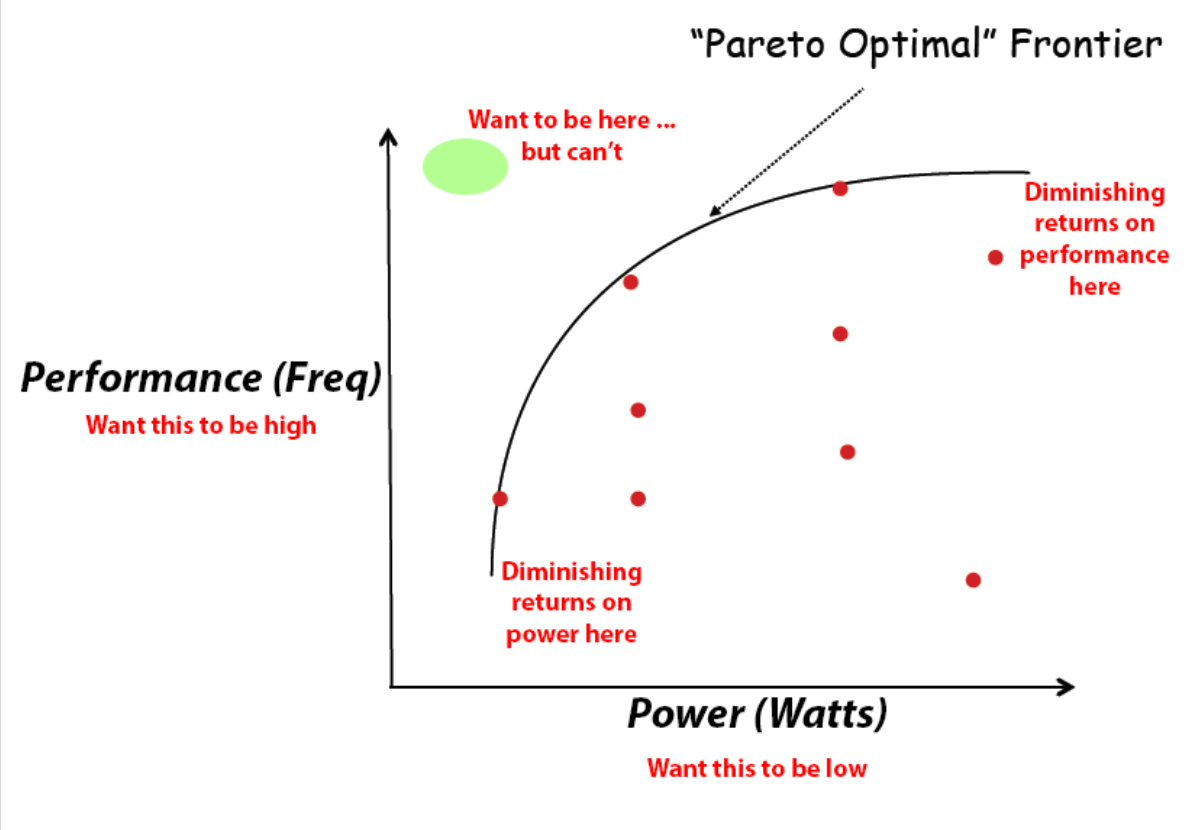

Figure 3: Performance, Power Design Space

Problem 3: 251A only — *Optional* Challenge Question for 151 [5 pts]

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?

Solution:

1. Voltages didn't scale down
2. From a power perspective, the die actually got bigger, so the total power consumption went up
3. Architectural changes, e.g. critical path reduction, deep pipelines, etc, let the frequency go up (which then explains why the power went up)

Problem 4: Full Custom versus Standard Cell [6 pts]

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.

Solution:

Standard Cell Approach:

Main advantage: Layout Fully automated and cells pre-verified and leading to higher designer productivity and reduced NRE costs Main disadvantage: Circuits Less optimized for performance, cost, power as may be needed by the application. No analog processing possible.

Full Custom Approach:

Main advantage: Circuits and layout can be optimized for performance, cost, power as needed by particular application.

Main disadvantage: Long design process, for sixing transistors, generating layout, verifying layout, etc.

Problem 5: Implementation Alternatives [12 pts]

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 [3 pts], b) per part costs [3 pts], c) performance [3 pts], and d) flexibility [3 pts].

Solution:

a) NRE costs Full Custom, Standard Cell, Gate Array, FPGA, micro-processor.

b) Per part costs Micro-processor, FPGA, Gate Array, Standard Cell, full-custom. *Also accept (in the worst case)*: Full-custom, micro-processor, FPGA, Gate Array, Standard Cell.

c) Performance Full Custom, Standard Cell, Gate Array, FPGA, micro-processor. *Also accept:* Full Custom, Standard Cell, Gate Array, micro-processor, FPGA.

d) Flexibilty for reuse in other applications Micro-processor, FPGA, Gate Array/Standard-Cell/Full-Custom. *Also accept*: FPGA, Micro-processor, Gate Array/Standard-Cell/Full-Custom.

Alternative orderings with sensible explanations may also be accepted.

Problem 6: Combinational Logic [4 pts]

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?

Solution:

A truth table has a 1 row for every combination of input values. Since there are two N-bit inputs, the total input width is $2N$ bits, and each bit has 2 possible values (0 or 1). This means that there will be 2^{2N} possible input states and 2^{2N} truth table rows. When there are two $N = 8$ bit inputs, the table has $2^{(2\times 8)} = 65536$ entries.

Problem 7: Combinational Logic Functions [4 pts]

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

Solution:

2^{2^N}. Why? There are 2^N input combinations to a CL block with N inputs an 1 output. For each of these input values, the CL produces a 1-bit output, which has 2 possible states. So there are 2^{2^N} different mappings from inputs to output state that the CL logic can encode. Blocks implementing different mappings are unique.

Problem 8: Combinational Logic Gates [6 pts]

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. There is a hypothesis that NAND-gates alone or NOR-gates alone are sufficient to implement any combinational logic function. Prove or disprove this hypothesis.

Solution:

You can turn a NAND gate into an inverter (connect inputs together, or fix one input high). With this inverter, you can turn a NAND gate into an AND gate. Then you have AND-gates and inverters, which we know to be sufficient to implement any combinational logic function. Similarly for NOR gates.