Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.313 Contemporary Computer Design

Problem Set #8 Solutions Fall, 1995

1. Power consumption of these processors is likely to be a significant problem. Current Pentium processors dissipate on the order of tens of watts, usually in the 10-30W range. If trends continue, the 2010 processor is likely to dissipate on the order of 100s of watts. The power dissipation is limited by cost in that most systems can only support a forced air cooling system since liquid cooling is generally prohibitively expensive, so a good focus for research would be an inexpensive liquid cooling technique.

Let's assume that gate capacitance accounts for 1/3 of the total chip capacitance. We can see from the 1995 Pentium numbers that this is reasonable. For a $1\mu m \times 0.6\mu m$ device with a 12nm gate oxide thickness, the capacitance ($\epsilon A/t$) is 1.7fF. The total gate capacitance for 10 million transistors is therefore 17nF. Now, we know that $CV^2 f = 20W$, with V = 2.9V and f = 133MHz, making C = 17.9nF. So, since about a third (or less) of the gates switch every cycle, the gates account for about a third of the total switched node capacitance. The other two thirds come from wire capacitance and source/drain capacitance.

So if we extrapolate this reasoning to the 2010 processor, the individual gate capacitance is $\epsilon(0.016\mu m^2)/4nm = 0.138aF$ so the total chip gate capacitance is 138nF. With a third of that accouting for a third of the total switched capactiance, the power will be close to $CV^2 f = (138nF \times 1.69 \times 4000MHz = 932W$. The problem here is that while V^2 and the individual gate capacitance have decreased slightly (giving us a factor of 60), the number of devices times frequency increased 3000x, so the total power increased by a factor of 50 from 20W to 1000W.

Obviously the Intel management must invest research money in low power circuit techniques. And ten years after 2010, with shrinking geometries and increasing die sizes, power dissipation will continue to rise. A number of limits are approached at that point. The geometries are close to the atomic scale, so gate oxides which are only a few atoms high become very difficult to manufacture, and quantum mechanical tunneling effects become significant. Low dopant densities become impossible when the channel area is only a few atoms by a few tens of atoms. A 4GHz clock speed is a 250ps period. Light travels 75mm in vacuum in 250ps, and signals are at least a factor of 2 slower in a wire on chip. Faster clock speeds don't leave much time for gates to switch.

And, one interesting limit being approached is energy dissipation. Erasing a bit of information, and at least one bit is lost during each standard CMOS operation, requires that at least $kT \ln 2 J$ of energy be dissipated. At 325K, this is $3 \times 10^{-21} J$. With the process of 2010, $CV^2 = 233 \times 10^{-21}$. Only a factor of 100 is available before energy recovery techniques which do not erase information are required to reduce the energy dissipation further. These energy recovery techniques often require that the chip run more slowly than its maximum possible frequency, but by 2020, the energy/power budget for the device is likely to be tighter than speed requirements, leaving speed to burn. For more information see Saed Younis' PhD thesis or Carlin Vieri's SM thesis.

2. The 200W of power dissipated by the chip are raising the temperature of the ambient air from 45°C (worst case) to 55°C. We are interested in the volume velocity (m^3 or l per second) of air required to maintain equilibrium. So, every second, 200J raises some volume of air 10°C. Let Q be the energy (amount of heat, 200J), m be the mass of the air (which we can find once we know the volume since we know the density), c be the specific heat, and ΔT the temperature change.

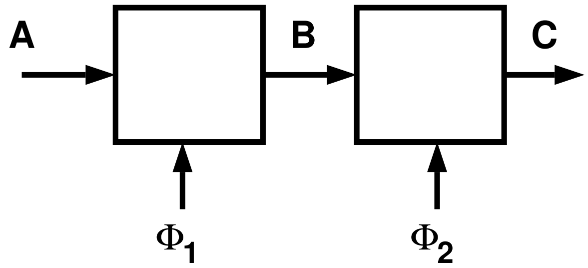
$$
\begin{aligned}
Q &= mc\Delta T \\
200 &= (1.08g/l \times V)(0.249cal/(gC^{\circ}))(10^{\circ}C) \\
V &= \frac{200}{(1.08g/l)(1.041J/gC^{\circ})(10^{\circ}C)} \\
V &= 17.8l
\end{aligned}
$$

So the volume velocity of air required is 17.8l/s (or $17.8 \times 10^{-3} m^3/s$)

3. This problem is fairly annoying. Let us first assume that the clock is being driven from only one side, a bit of a boneheaded assumption considering the length of the shfit register. This assumption allows us to assume that the skew is the same between any two elements (ie all skews are either +150ps or -150ps).

We will also make a few assumptions about how the latches work. Latches start to propagate their values as soon as the clock starts to go high, and the propagation delay through the element is the time between when the clock starts to change and when the output starts to change. Latches pass the correct value if the input is stable when the clock reaches a low value.

a) The problem in making the shift register fast is going to be that the clock skew will allow signals to pass through multiple blocks since we are using a transparent latch. We must make sure that the minimum delay through a latch is greater than the skew (plus some rise and fall corrections.)



Let us consider the case in which the data is being shifted from left to right and the skew is such that a block to the right is clocked 150ps after the block to the left (positive skew). In this case we must make sure that the correct value is passed from B to C and that A does not travel through both blocks. For this to happen, the clock on the rightmost latch (Φ_2) must be low before node B starts to change. In the worst case, this is 350ps after the clock on the left latch (Φ_1) starts to go high: the sum of the skew delay, one rise time, and one fall time. So,

$$t_{gate} > t_{skew} + t_{rise} + t_{fall} = 350ps \tag{1}$$

So the latch we use must have a delay of at least 350ps to avoid contaminating the next latch.

Now, let us look at the condition when the Φ_2 leads Φ_1. We know that we must make the gate delay at least 350ps, and since we want the clock to resemble a pulse it rises and falls (with no high time) in 200ps, we can now determine the period. The time that the signal must be low is determined by when the second pulse on Φ_2 can start to go high. Node B must be stable (a rise or fall time after a gate delay) with its new value when Φ_2 *finishes* falling. This happens in at most a rise and fall time plus a clock period minus a skew delay. Or,

$$\begin{aligned}
t_{gate} + t_{rise} &< t_{rise} + t_{fall} + t_{low} + t_{rise} + t_{fall} - t_{skew} && (2)\\
450ps &< 400ps - 150ps + t_{low} && (3)\\
200ps &< t_{low} && (4)
\end{aligned}$$

So the clock is high for 200ps (a rise and fall) and low for 200ps, for a frequency of 2500MHz. There is a cost in logic to create a delay element to stretch out the latch delay to 350ps. Minimum delays through logic are difficult to ensure.

b)

c)

4. The circuit is not guaranteed to work with an active time of 1.5ns because the contamination delay is 1ns. An Earle latch is transparent, so the 1ns delay of the An alternate answer is that it will not work with a 3ns period because the Earle latch structure will add some small delay to each stage and require a slightly greater than 3ns period.

To design the clock waveforms for this structure, you need to know which piece comes first, what the skew is between the two clock phases, and the (probably small) delay of the Earle latch itself.