

Chapter II

Physics of Computation

These lecture notes are exclusively for the use of students in Prof. MacLennan's *Unconventional Computation* course. ©2019, B. J. MacLennan, EECS, University of Tennessee, Knoxville. Version of August 24, 2019.

A Energy dissipation

As an introduction to the physics of computation, and further motivation for unconventional computation, we will discuss Michael P. Frank's analysis of energy dissipation in conventional computing technologies (Frank, 2005b). The performance R of a computer system can be measured by the number of computational operations executed per unit time. This ratio is the product of the number operations per unit of dissipated energy times the energy dissipation per unit time:

$$R = \frac{N_{\text{ops}}}{t} = \frac{N_{\text{ops}}}{E_{\text{diss}}} \times \frac{E_{\text{diss}}}{t} = F_{\text{E}} \times P_{\text{diss}}. \quad (\text{II.1})$$

Here we have defined P_{diss} to be the power dissipated by the computation and the *energy efficiency* F_{E} to be the number of low-level bit operations performed per unit of energy. The key parameter is F_{E} , which is the reciprocal of the energy dissipated per bit operation.

This energy can be estimated as follows. Contemporary digital electronics uses CMOS technology, which represents a bit as the charge on a capacitor. The energy to set or reset the bit is (approximately) the energy to charge the capacitor or the energy dissipated when it discharges. Voltage is energy

per unit charge, so the work to move an infinitesimal charge dq from one plate to the other is Vdq , where V is the voltage between the plates. But V is proportional to the charge already on the capacitor, $V = q/C$. So the change in energy is $dE = Vdq = \frac{q}{C}dq$. Hence the energy to reach a charge Q is

$$E = \int_0^Q \frac{q}{C}dq = \frac{1}{2} \frac{Q^2}{C}.$$

Therefore, $E = \frac{1}{2}(CV)^2/C = \frac{1}{2}CV^2$ and $F_E \approx (1 \text{ op})/(\frac{1}{2}CV^2)$.

Frank observes that Moore's law in the 1985–2005 period was a result of an exponential decrease in C resulting from decreasing feature sizes (since capacitance is proportional to area) and a decrease in logic voltage V from 5V to about 1V (further improving E by a factor of 25). The clock rate also went up with smaller feature sizes. (See Fig. II.1.)

Dennard scaling has been another important factor supporting Moore's law (Dennard et al., 1974). This refers to the fact that voltage and current both scale downward with feature size, and therefore power scales with transistor area. As a consequence, *power density* (the power dissipated per unit area) remains constant as the size of transistors decreases. Furthermore, with smaller transistors, computers could operate faster. We have seen that the energy to change state is $CV^2/2$. Therefore, if the clock frequency is f and transistors switch a fraction α of the time, the energy dissipated will be $\alpha f CV^2/2$. But the capacitance is proportional to the square of the linear dimensions, and decreasing them allows a corresponding increase in switching frequency, and so circuits have been able to operate faster with the same power density.

Unfortunately, Dennard scaling began to break down around 2005. The reason is that there are two sources of power dissipation in integrated circuits. A chip's *dynamic power density* results from its transistors changing state and it benefits from Dennard scaling. However, *static power density* is a result of leakage currents through the transistors when they are not switching. This is a result of quantum mechanical tunneling through the transistor's gate and increases with decreasing gate widths. It does not scale down with feature size; on the contrary it scales up rapidly: a 100-fold decrease in size has resulted in a 10^8 increase in static power density. Previously, with larger transistors, static power density was negligible, but it is now comparable to dynamic power density. By 2006–7 these conditions had created a “power wall” and ended Dennard scaling.

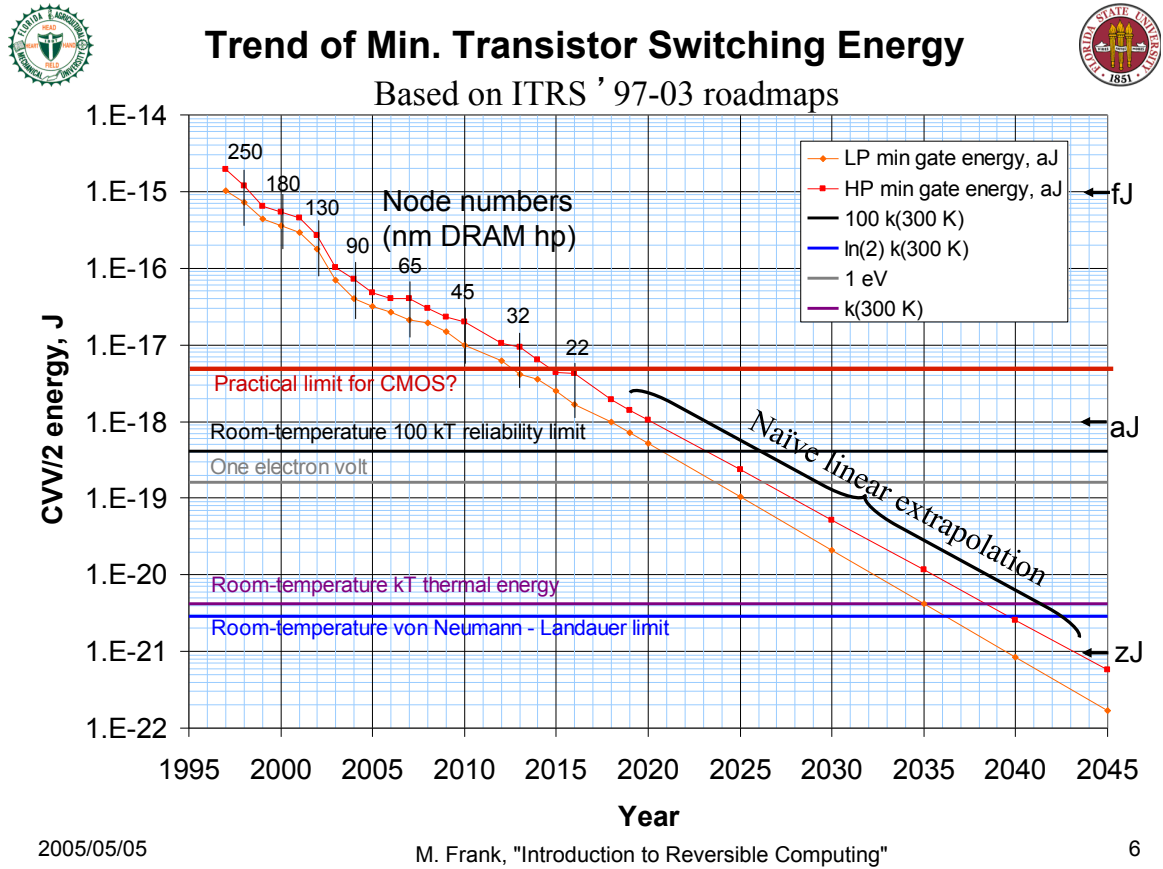


Figure II.1: Historical and extrapolated switching energy. Figure from Frank (2005b, slide 9).

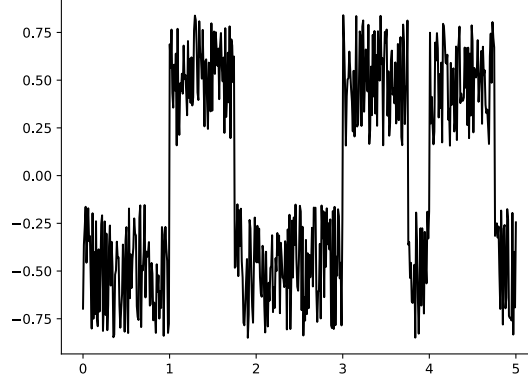


Figure II.2: Depiction of 0-1-0-1-1 pulses in the presence of high thermal noise.

In addition to the preceding issues, there are other limitations on the energy efficiency of computation, for if the signal is too small in comparison with thermal energy, then thermal noise will lead to unreliable operation, because the thermal fluctuations will be of the same order as the signals (Fig. II.2). The thermal energy is $E_T = k_B T$, where k_B is Boltzmann's constant and T is the absolute temperature. Since $k_B \approx 8.6 \times 10^{-5} \text{ eV/K} = 1.38 \times 10^{-23} \text{ J/K}$, and room temperature $T \approx 300\text{K}$, room-temperature thermal energy is

$$E_T = k_B T \approx 26 \text{ meV} \approx 4.14 \times 10^{-21} \text{ J} \approx 4 \text{ zJ}.$$

(Fig. II.1 shows E_T .)

We have seen that $E_{\text{sig}} = CV^2/2$, but for reliable operation, how big should it be in comparison to E_T ? Frank estimates $E_{\text{sig}} \geq k_B T \ln R$, where the reliability $R = 1/p_{\text{err}}$, for a desired probability of error p_{err} .¹ For example, for a reasonable reliability $R = 2 \times 10^{17}$, $E_{\text{sig}} \geq 40k_B T \approx 1 \text{ eV}$, which is the energy to move one electron with 1V logic levels. This implies a maximum energy efficiency of

$$F_E = 1 \text{ op/eV} \approx \frac{1 \text{ op}}{1.6 \times 10^{-19} \text{ J}} = 6.25 \times 10^{18} \text{ op/J}. \quad (\text{II.2})$$

¹Frank (2005b, slide 7).

A round $100k_{\text{B}}T$ corresponds to an error probability of $p_{\text{err}} = e^{-100} = 3.72 \times 10^{-44}$ (at room temperature). Therefore, a reasonable target for reliable operation is

$$E_{\text{sig}} \gtrsim 100k_{\text{B}}T \approx 2.6 \text{ eV} = 414 \text{ zJ}.$$

This, therefore, is an estimate of the minimum energy dissipation per operation for reliable operation using conventional technology. Nevertheless, these conclusions are independent of technology (electronic, optical, carbon nanotube, etc.), since they depend only on relative energy levels for reliable operation.²

One apparent solution is to operate at a lower temperature T , but it does not help much, since the effective T has to reflect the environment into which the energy is eventually dissipated (i.e., the energy dissipation has to include the refrigeration to operate below ambient temperature). Another possible solution, operating closer to $k_{\text{B}}T$ and compensating for low reliability with error-correcting codes, does not help, because we need to consider the *total energy* for encoding a bit. That is, we have to include the additional bits required for error detection and correction.

Frank observed in 2005 that the smallest logic signals were about $10^4k_{\text{B}}T$, and therefore that there were only about two orders of magnitude improvement in reliable operation. “A factor of 100 means only around 10 years remain of further performance improvements, given the historical performance doubling period of about 1.5 years. Thus, by about 2015, the performance of conventional computing will stop improving, at least at the device level” (Frank, 2005b).

In fact, these limitations are becoming apparent. By 2011 computer engineers were worrying about “the 3 GHz wall,” since computer clock speeds had been stalled at about that rate for five years.³ Recent processors have gone a little beyond the barrier, but a “power wall” remains, for although individual transistors can be operated at higher speeds, the millions or billions of transistors on a chip dissipate excessive amounts of energy. This presents an obstacle for future supercomputers.

As of June 2018 the fastest supercomputer was Summit (OLCF-4).⁴ It is

²Frank presentation, “Reversible Computing: A Cross-Disciplinary Introduction” (Beyond Moore), Mar. 10, 2014. put in bib

³*Spectrum* (Feb. 2011) spectrum.ieee.org/computing/hardware/nextgeneration-supercomputers/0 (accessed Aug. 20, 2012).

⁴[https://en.wikipedia.org/wiki/Summit_\(supercomputer\)](https://en.wikipedia.org/wiki/Summit_(supercomputer)) (accessed Aug. 23, 2018);

rated at 200 petaflops and performed at 122.3 petaflops on LINPACK benchmark. It is also the first supercomputer to reach exascale speed, performing at 1.88 exaops during a genomic analysis. It is expected to reach 3.3 exaops using mixed-precision calculations. Summit has 2,282,544 CPU and GPU processing cores in 4608 nodes, each with two IBM Power9 CPUs and six Nvidia V100 GPUs. The 9216 POWER9 CPUs each have 22 cores (202,752 total), and the 27,648 V100 GPUs each have 80 streaming multiprocessors (SMs), each with 32 FP64 (double-precision) cores, 64 FP32 (single-precision) cores, 64 INT32 cores, and 8 tensor cores.⁵ Each node has over 500GB of coherent memory (high-bandwidth memory plus DDR4 SDRAM) addressable by all CPUs and GPUs, plus 800GB of non-volatile RAM. Summit has 250 PB total file storage, occupies 5,600 sq. ft. of floor space (approximately two tennis courts), consumes 13 MW total power and has an energy efficiency $F_E = 13.889$ GFlops/watt (the fifth most energy efficient supercomputer), which is about 72 pJ/flop. Its power consumption is comparable to a town of 3436 households and requires 4,000 gallons of water per minute for cooling.

To convert floating-point operations to basic logic operations, including all the overhead etc., one conversion estimate is 10^7 to 10^8 ops/flop.⁶ Therefore, we can compare the theoretical best energy efficiency (Eq. II.2), $F_E^{-1} = 1.6 \times 10^{-7}$ pJ/op ≈ 1.6 to 16 pJ/flop, with the 72 pJ/flop of Summit. The gap is only about one order of magnitude. Indeed, it has been estimated that scaling up current technology to 1 exaflops would consume 1.5 GW, more than 0.1% of US power grid.⁷ This is impractical.

It might be possible to get energy consumption down to 5 to 10 pJ/flop, but “the energy to perform an arithmetic operation is trivial in comparison with the energy needed to shuffle the data around, from one chip to another, from one board to another, and even from rack to rack.”⁸ Indeed, due to the difficulty of programming parallel computers, and due to delays in internal data transmission, it is difficult to use more than 5% to 10% of a supercom-

<https://www.olcf.ornl.gov/olcf-resources/compute-systems/summit/> (accessed Aug. 23, 2018)]

⁵<https://www.olcf.ornl.gov/for-users/system-user-guides/summit/nvidia-v100-gpus/>

⁶And so this is one estimate of the difference in time scale between computational abstractions and the logic that implements them, which was discussed in Ch. I (p. 5).

⁷*Spectrum* (Feb. 2011) spectrum.ieee.org/computing/hardware/nextgeneration-supercomputers/0 (accessed Aug. 20, 2012).

⁸*Spectrum* (Feb. 2011) spectrum.ieee.org/computing/hardware/nextgeneration-supercomputers/0 (accessed Aug. 20, 2012).

puter's capacity for any extended period; most of the processors are idling.⁹ So with those 2,282,544 cores, most of the time about *two million* of them are idle! There has to be a better way.

⁹*Spectrum* (Feb. 2011) spectrum.ieee.org/computing/hardware/nextgeneration-supercomputers/0 (accessed 2012-08-20).