Crossing *the* threshold

Why engineers are looking to run transistors from lower supply voltages. By **Chris Edwards**.

For good reason, CMOS chip designers pay careful attention to the equation that governs dynamic power consumption: ½CV²f.

We have seen the benefit over the past two or three decades, as supply voltages for microprocessors and logic have dropped from 12V to 5V to 3V and now less than 1V. Much of that reduction has been of necessity; the core logic transistors of deep sub micron processes cannot take much more than 1V without sustaining permanent damage.

If 1V is good, why not push to 0.5V or less? Some designers have done that. Hearing aids designed in the early 2000s could run at 0.47V, while Imperial College spinout Toumaz Technology was launched on the back of 'smart plasters' that used very low supply voltages to let them operate from a tiny battery.

There is, naturally, a catch: circuits that operate from extremely low supply voltages are extremely slow. Logic transistors are designed to pass reasonable quantities of current when saturated. Generally, the higher the gate voltage, the more current will pass. Until saturation, the relationship between on-current and voltage is pretty steep, so reducing the voltage a little causes a dramatic reduction in on-current. This means it takes longer to charge the capacitances in the next logic stage ready for the next switching cycle. To compensate, the clock rate has to drop.

Process engineers play with the relationship between supply voltage and speed by tuning the threshold voltage – the point at which the channel should start conducting. However, lowering this threshold tends to result in the channel never being fully switched off – it's always leaking some current.

Subthreshold logic uses this threshold-related leakage to operate – the transistor simply becomes more leaky as the gate is switched to its higher voltage state, which still might be less than the nominal threshold voltage: as the name implies.

So a circuit that can switch at gigahertz frequencies at 1V is unlikely to have a clock rate of more than 1MHz when that voltage falls towards the threshold – around 0.25V for a standard 28nm process core logic transistor. This clock rate can work for smart plasters or hearing aids. The former only needs enough power to deliver a reading once every few seconds or minutes to a data collector, while the DSPs used in hearing aids are generally highly parallelised so they can run at very low clock rates, but still maintain audio-rate filtering.

Loss of performance

Loss of performance might be tolerable if you can deploy enough parallelism. With nanometre processes, chip designers are more worried about what happens if you try to fire up more than a fraction of the die at any one time than of running out of transistors. If you could run many operations in parallel at low clock speed and low voltage, you can potentially eke out battery power more efficiently than today.

Unfortunately, there is a second catch, which lies in the leakage current used to switch the device (see fig 1). The beauty of CMOS logic is that its active switching energy is very low because it only consumes energy when the capacitance of a downstream logic path is being charged or discharged. Normally, this does not take long. But switchover time increases dramatically as voltage tumbles.

There is, naturally, a catch: circuits that operate from extremely low supply voltages are extremely slow.

As the voltage sinks to less than 0.3V in a modern process, the effect of the leakage, which is not suppressed by the lower supply voltage, shoots up dramatically (see fig 2). Worse, any attempt to lower the threshold voltage to improve speed tends to increase leakage, negating the benefit. So, beyond a certain minimum voltage, active power gains from the lower supply voltage are quickly lost. For this reason, attention has shifted amongst processor designers from the subthreshold region to the near-threshold area that lies just above the threshold voltage – roughly between 0.3V and 0.8V.

Here, designers believe they can make big gains in power efficiency. At February's International Solid State Circuits Conference, Intel outlined a graphics processor that operates close to the threshold region, delivering an efficiency, measured in GFLOP/W, 2.7 times higher than that achieved at a 'normal' voltage of about 0.4V higher.

There is one last catch: variability. Variability between transistors is already a major problem in nanometre processes. As you cut the supply voltage, these effects from the

Fig 1: Leakage sources in a transistor

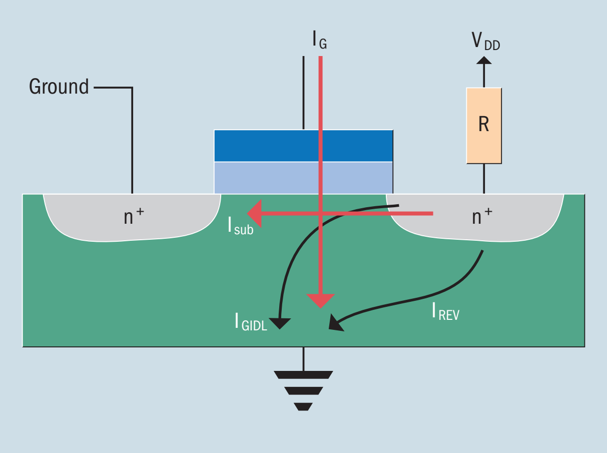
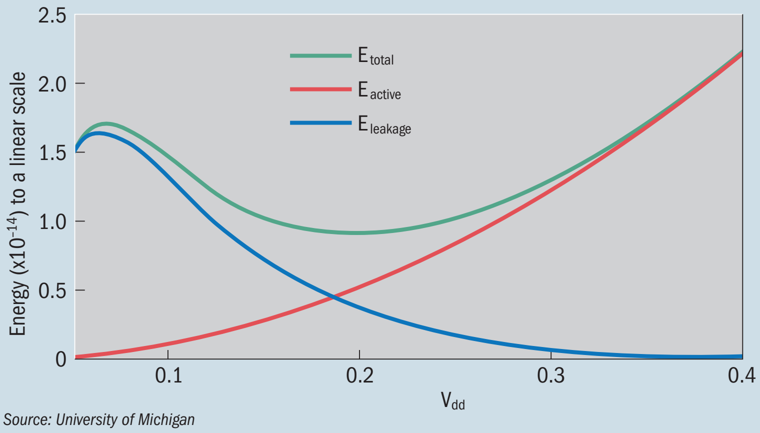


Fig 2: The influence of voltage on energy consumption



Source: University of Michigan

Lower voltage circuits look to be inevitable as the focus continues to be on energy consumption in computing.

perspective of the circuit designer become magnified (see fig 3). Work by Professor David Blaauw and colleagues at the University of Michigan, who have been working with ARM on near-threshold logic development, showed that random dopant fluctuations cause the biggest problems. These small changes tend to change the effective channel length, which in turn alters the effective threshold voltage. You can reduce this effect by making the transistor wider, but as designers expect to use parallelism to make up for speed, this is not good news for density and cost.

To be on the safe side, designers increase the supply voltage to ensure that all gates will switch within a given clock cycle, even in voltage droop, the result of many transistors switching in close proximity. This rise in operating voltage increases the power consumption. For its graphics unit, Intel used a calibration scheme during test to better characterise the threshold voltages for transistors and work out a safe minimum voltage and achieve an average supply voltage reduction of around 200mV.

Investigating other approaches

In its work with Michigan, ARM has investigated other approaches. The Razor technique uses error detection to work out if a sudden voltage droop has caused logic to switch too slowly and, after pushing the voltage up, allows the operation to start again.

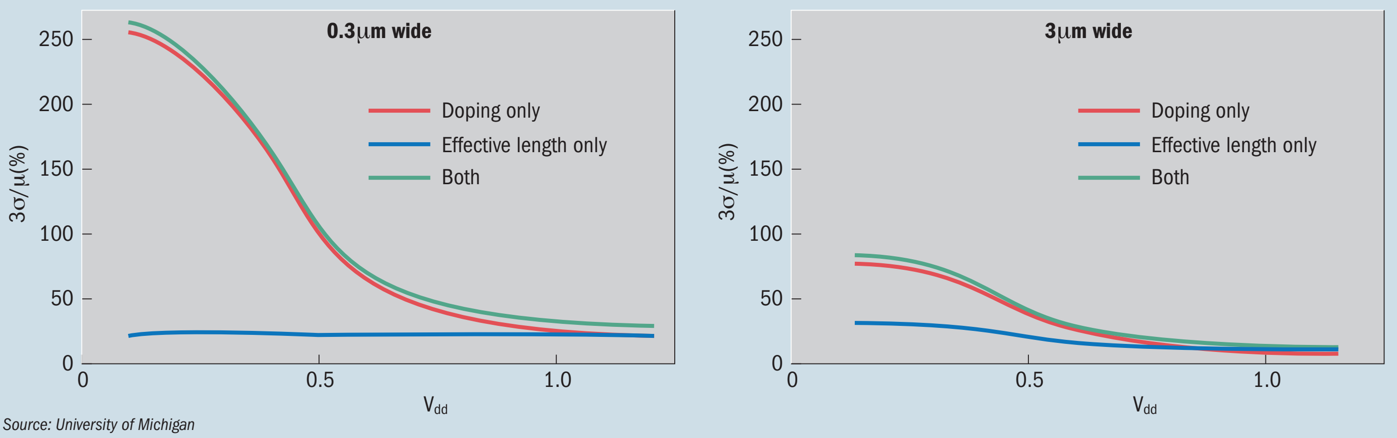
Leakage remains an issue when circuitry is not switching. As with regular logic, power gating is an option, but might be used in very different ways to normal logic. ARM is also working on near-threshold logic circuits that can shut off not only between operations but also midway through each one to reduce the impact of leakage when operating at very slow clock speeds. This demands the use of logic gates that can retain their state without power and reactivate quickly. One option is to design the gates so they leak contents very slowly and are refreshed periodically like a DRAM before the charge dissipates entirely.

Low voltage may not always be the answer, however. Some Michigan-based research has indicated that SRAMs can be more energy efficient at higher speeds. This makes it feasible to have clusters of slow processors using a higher speed shared cache without blocking each other.

CISC architectures may also have an advantage over RISC approaches. More efficient encoding may not just reduce the average bit-length of instructions, cutting the energy needed to transfer, but the use of more complex addressing modes can also reduce the number of instructions needed for a given operation.

It's still early days for near-threshold logic in mainstream applications, but lower-voltage circuits look to be inevitable as the focus continues to be on energy consumption in computing.

Fig 3: The variability of narrow and thick transistors around the threshold point



Source: University of Michigan