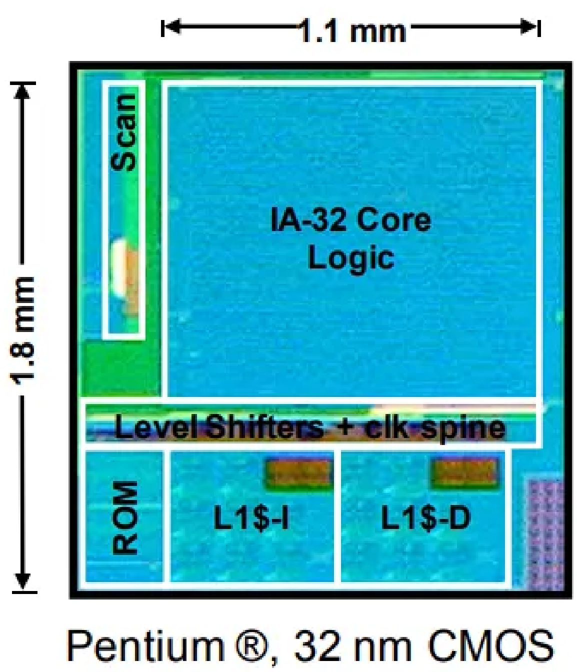
Remaking the Nokia 6110 and Psion Series 3 on 22nm with Solar Power

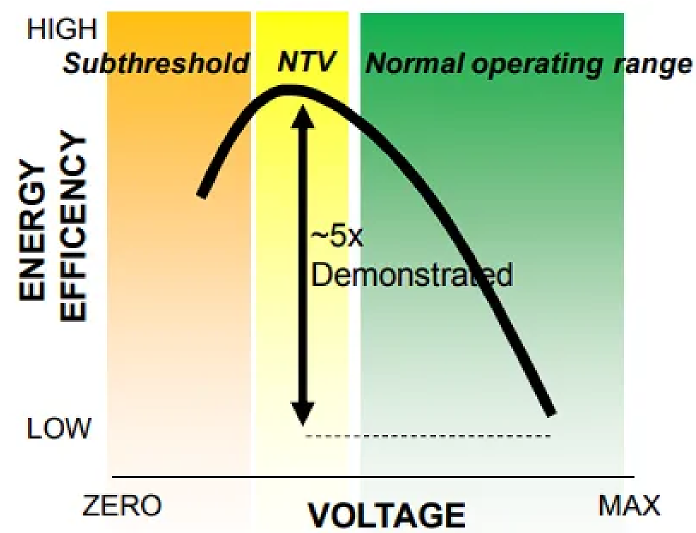
The ghost of Hunter S. Thompson shows up at an empty Las Vegas CES conference center in January of 2021, wondering where the world went.

DON DOS PASSOS

NOV 6, 2022

Share

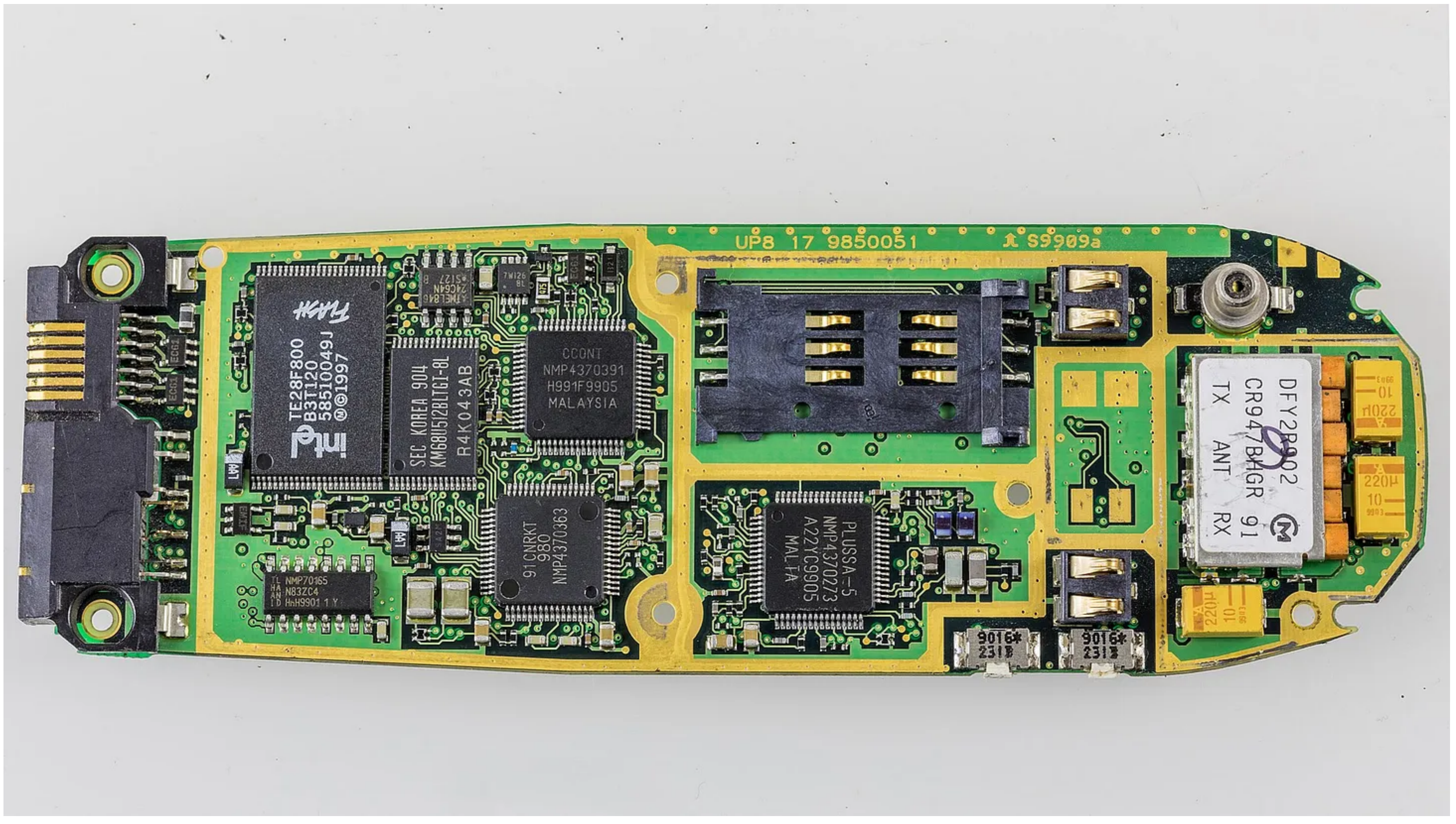




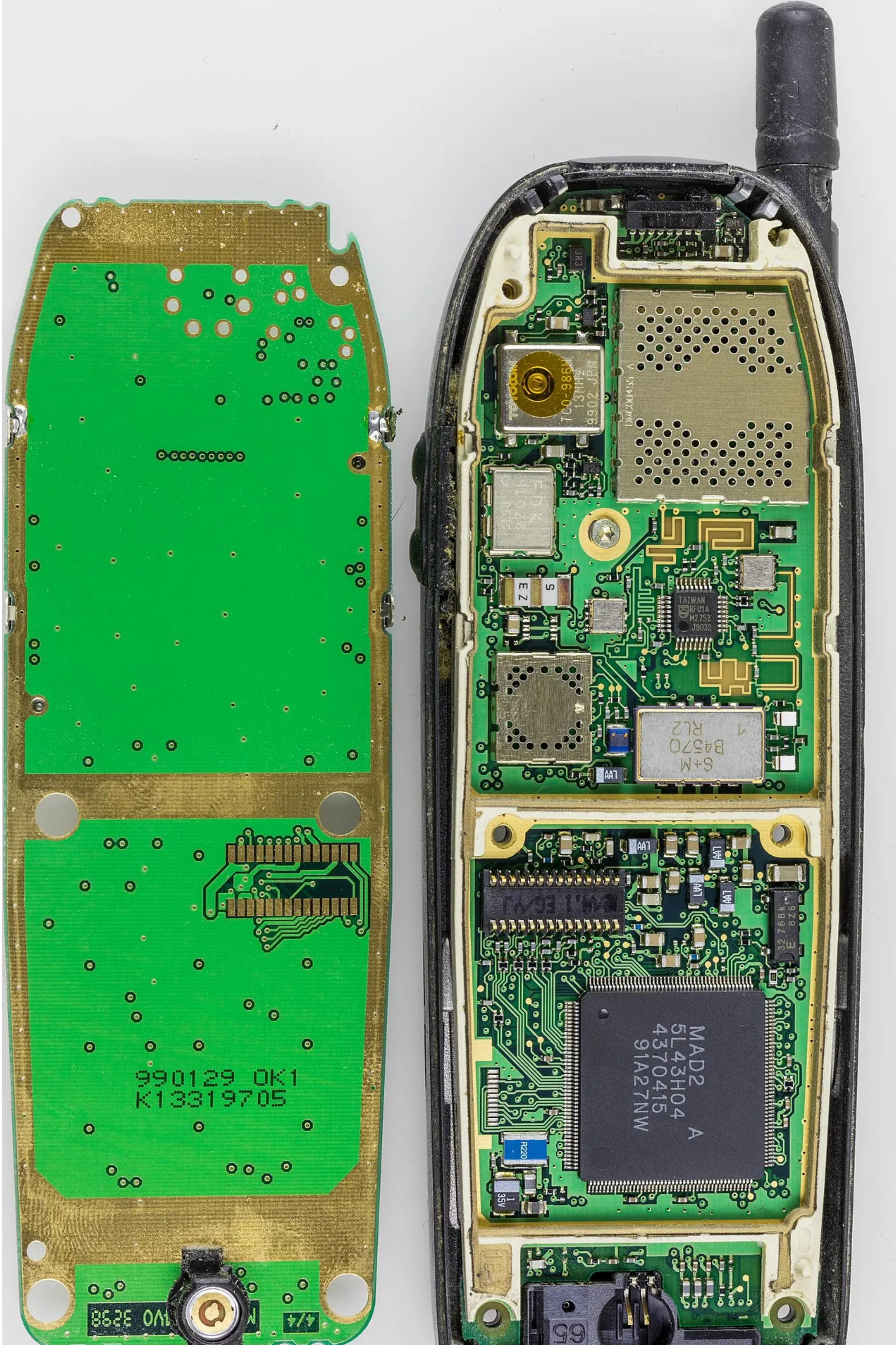
Ultra-low Power	Energy Efficient	High Performance
280 mV	0.45 V	1.2 V
3 MHz	60 MHz	915 MHz
2 mW	10 mW	737 mW
1500 Mips/W	5830 Mips/W	1240 Mips/W

Freigabe, the theme song to Everybody to the Limit

Everybody to the Limit




1/4
UE4_11
9850046


990129 OK1
K13319705
MAD2
5L43H04 A
4370415
91A27NW
S+M
B4570
RL2
TAIWAN
RFU1A
M2752
J9032

fr

Not a [nostalgic]() post. Ok maybe some.

This post is what I would consider a [forward looking]() statement:

“Our discussion may include predictions, estimates or other information that might be considered forward-looking. While these forward-looking statements represent our current judgment on what the future holds, they are subject to risks and uncertainties that could cause actual results to differ materially. You are cautioned not to place undue reliance on these forward-looking statements, which reflect our opinions only as of the date of this presentation. Please keep in mind that we are not obligating ourselves to revise or publicly release the results of any revision to these forward-looking statements in light of new information or future events. Throughout today’s discussion, we will attempt to present some important factors relating to our business that may affect our predictions.”

[Update 1/19/2023: See these notes: https://github.com/kragen/dernocua/blob/master/text/energy-autonomous-computing.md]

UI category	Series 30	Series 40	Series 60	Series 80
Key drivers	Cost-driven platform	Size-driven color platform	One-hand operated feature platform	Two-hand operated feature platform
Display image (images not to scale)				
Display resolution	96 x 65	128 x128	176 x 208	640 x 200
Supported application and content platforms	WAP/XHTML MIDP MMS	WAP/XHTML MIDP MMS	WAP/XHTML MIDP MMS Symbian OS	WEB browser MIDP, Personal Java MMS Symbian OS

Figure 34. Nokia’s user interface categories

“The **Nokia** [6110]() was a GSM mobile phone from Nokia announced on 18 December 1997 and released in 1998. It is not to be confused with the newer Nokia 6110 Navigator. It was a hugely popular follower of the Nokia 2110, and the first of the many Nokia 6xxx series business-targeted phones. Main improvements over the 2110 were reduced size and improved talk time. It was the first GSM phone to use an ARM processor,[1] as well as the first running on Nokia's Series 20 user interface.[2]"

http://www.nokia-tuning.net/index.php?s=series20

The Nokia 6110 may be remembered more for its Snake game than anything else, if only because its 450 hour standby battery life didn't require worrying about anything else. The phones worked, and even had tactile contours on the buttons so one didn't have to think so much about where to press on a touch screen (e.g. smartphones with capacitive touch)

The resolution was also not described in the same way. For example on the Nokia 3310, with the same chipset, it had "Resolution 4 x 13 characters". It also went by "5 Lines" on the 6110. The actual number of pixels was more than 4x13, obviously, but resolutions back then were probably advertised to indicate how how many rows of contacts one could view at a time.

https://en.wikipedia.org/wiki/Nokia_3310

The Nokia 6110 and 3310 used an ARM7TDMI which TI made in their MAD2WD1:

Processor	Texas Instruments MAD2WD1
Market (main)	Smartphone
ISA	ARMv3 (32-bit)
Microarchitecture	ARM7
Family	Texas Instruments
Part number(s), S-Spec	MAD2WD1
Release date	Q4 1994
Lithography	132 nm
Transistors	74,209
Cores	1
Threads	1
Frequency	13 MHz
Details	1x ARM7TDMI @ 13 MHz
Cache memory	8 KB
Max memory capacity	1 KB
Memory types	SDRAM
TDP	5 W
GPU integrated graphics	None
Socket	SoC

This was Nokia's first use of the ARM processor, which had a display for *applications*, like calendar, contacts, and snake. All on 13 MHz.

Psion PDA

Technische specificaties [bewerken | brontekst bewerken]

Jaar	Model	Processor	Beeldscherm	ROM	RAM	Connectiviteit	Specificaties
1991	Series 3	V30 (3,84 MHz)	240×80 pixels	384 kB	256 kB, 512 kB		- 165×85×22 mm - 265 gram - 3 volt
1993	Series 3a	V30H (7,68 MHz)	480×160 pixels	1 MB	256 kB, 512 kB, 1 MB, 2 MB	19,2 kbps RS-232C	
1996	Series 3c					56,4 kbps RS-232C, infraroodpoort	
1998	Series 3mx	V30MX (27,68 MHz)		2 MB	1 MB, 2 MB	115 kbps RS-232C	

https://nl.wikipedia.org/wiki/Psion_Series_3 (Dutch Wiki entry, with Megahertz chart)

https://en.wikipedia.org/wiki/Psion_Series_3 (English, no Mhz specs)

Pictured: The Series 3 (1991) with 3.84 Mhz and a 240x80 pixel screen.

“The **Psion Series 3** range of personal digital assistants were made by Psion PLC. The four main variants are the *Psion Series 3* (1991), the *Psion Series 3a* (1993), the *Psion Series 3c* (1996), and the *Psion Series 3mx* (1998), all sized 165 by 85 by 22 millimetres (6.50 in × 3.35 in × 0.87 in).”

https://en.wikipedia.org/wiki/EPOC_(operating_system)

“EPOC was developed at Psion, a software and mobile-device company founded in London in 1980. The company released its first pocket computer in 1984: an 8-bit device named the Psion Organiser. In 1986 they released a series of improved models under the Organiser II brand, but the 8-bit era was ending. Psion saw a need to develop a 16-bit operating system to drive their next generation of devices.[5] First, however, they needed to engineer a 16-bit single-board computer, something that was extremely difficult at the time. They codenamed the project *SIBO*, for "single-board organiser" or "sixteen-bit organiser". To develop the SIBO hardware and software, they needed samples of the 16-bit microprocessors they would be programming; but it took more than a year to secure the chips, which caused a significant delay.[5]

By 1987, development of EPOC was underway: It was a single-user, preemptive multitasking operating system designed to run in read-only memory (ROM). The operating system and its programms were written in Intel 8086 assembly language and C. When the operating system started, it opened the pre-installed programmes in advance so that the system could switch between them quickly. To enable users to write and run their own programmes, EPOC featured an updated version of the Open Programming Language (OPL), which was first published with the Psion Organiser. OPL was a simple interpreted language somewhat like BASIC.

In 1989, Psion released the first 16-bit computers to be equipped with the new operating system: the MC200 and MC400 notebooks. Each of these had an Intel 80C86 processor, but differed in some other specifications, such as memory capacity. Among the later SIBO devices were the Psion Series 3 (1991), 3A (1993), 3C (1996), Workabout series, and the Siena 512K model (1996). The final EPOC device was the Psion Series 3mx (1998).[6]

The user interface differed by device. The notebook computers had a *windows, icons, menus, pointer* (WIMP) graphical user interface (GUI). The handheld computers, which had smaller screens and no pointing device, accept input from a keyboard or a stylus.[7]”

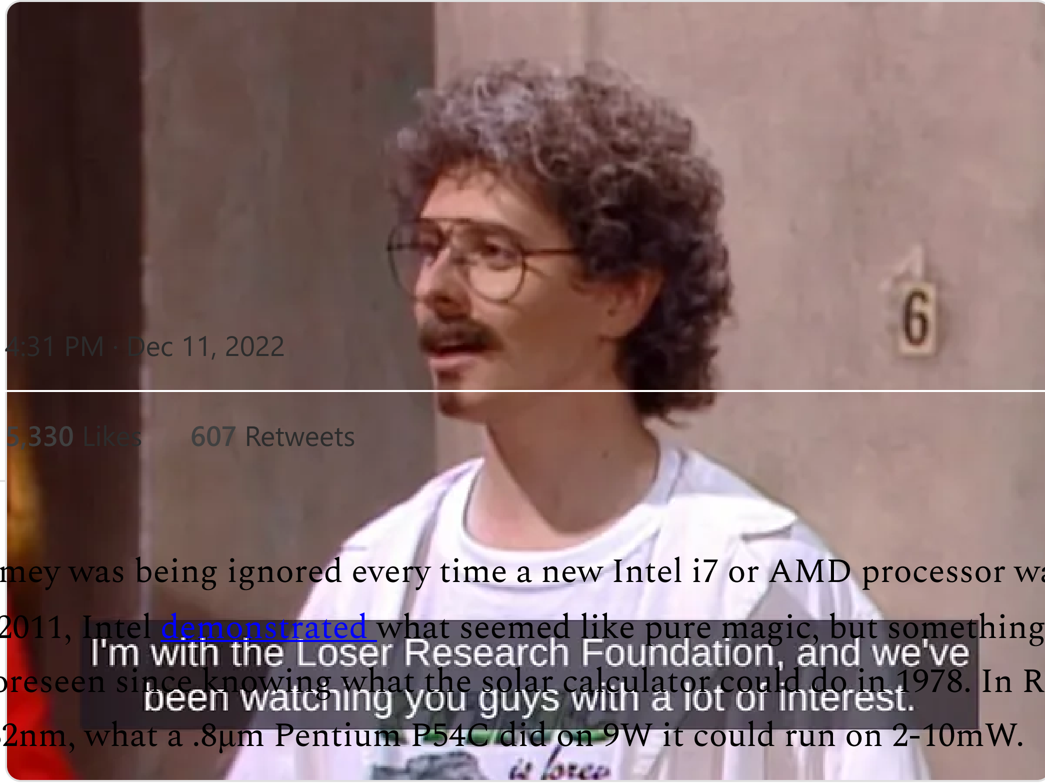
The Precedent

I eschew making predictions, because I do not want to be wrong when I don’t have to, and I especially eschew making bets, but I’d be willing to bet $2000 (or a used 90s Honda Civic, if I had one) that the above two operating systems could be solar powered without a battery using 22nm technology or less.

The reasoning starts with the first front-facing, and fully solar powered calculator, in 1978, the Teal Photon.

The Photon, had no coin battery like other calculators at the time did. It also didn't require constant computation, which definitely made the task easier for the designer. But in the 40+ years since the 7-function, 25-key Teal Photon, Koomey's Law happened (*continued* to happen). Now, you might be wondering, why I didn't say Moore's Law, or even Dennard's scaling. And the reason is, because no one in the for-profit world listens to an environmentalist. Moore is the normal paternalistic person that everyone needs first, like Maslow's hierarchy. I'm kidding of course, and no offense to any of the three. In fact, I was talking about myself, which no one really listens to.

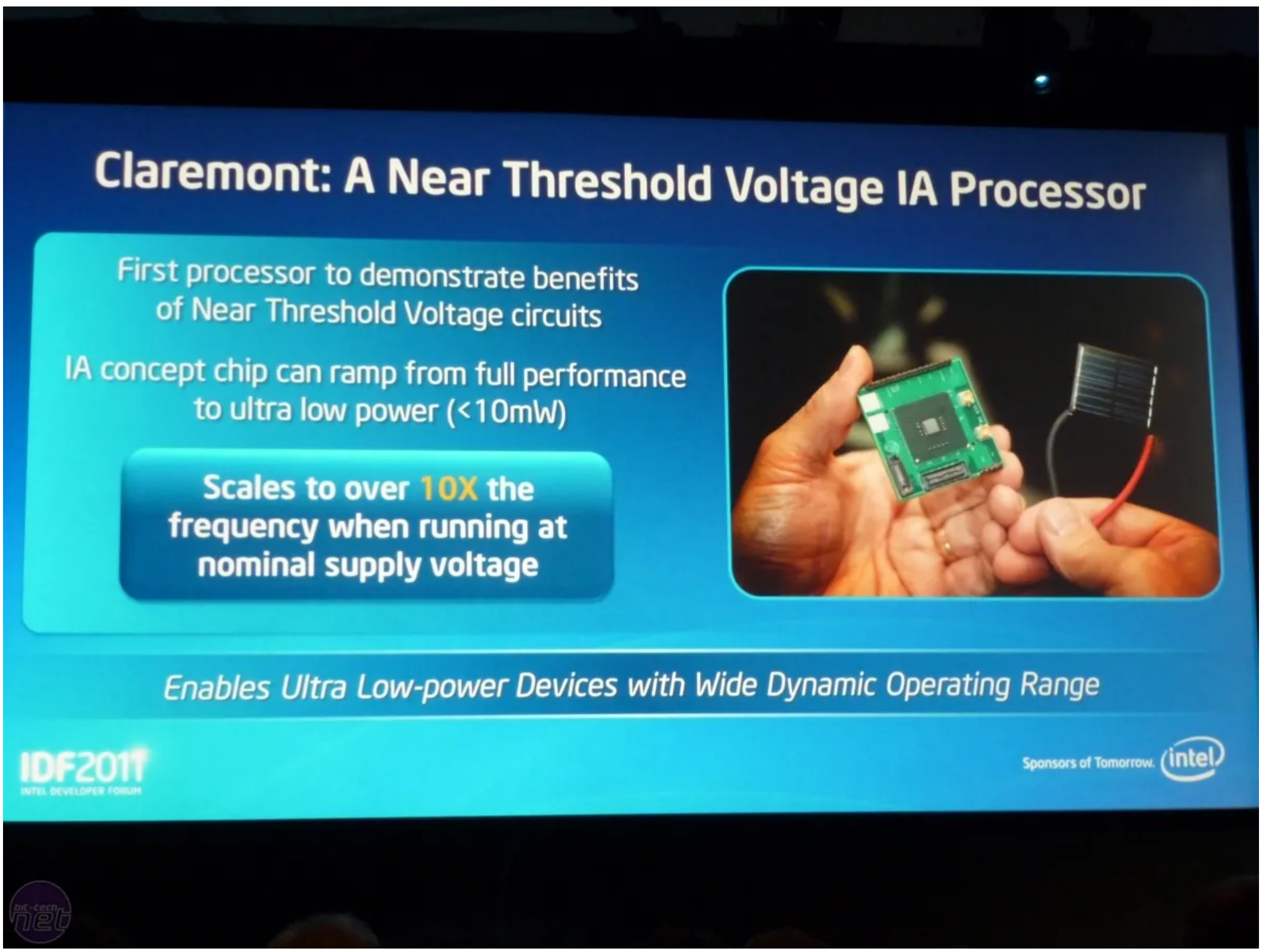
kids in the hall no context
@KITHnocontext



4:31 PM · Dec 11, 2022

5,330 Likes 607 Retweets

So while Koomey was being ignored every time a new Intel i7 or AMD processor was released, sometime in 2011, Intel demonstrated what seemed like pure magic, but something a Koomeyist would have foreseen since knowing what the solar calculator could do in 1978. In Remaking the Pentium on 32nm, what a .8µm Pentium P54C did on 9W it could run on 2-10mW.



After the demo, articles pointed out clearly that it was not for sale. Which raised questions, but only because I am naturally curious.

More notably, Intel resorted to dumpster diving (something I also did when I was at university in 2005, picking up an Optiplex PC with a 266mhz Pentium II, far faster than my 90s era hand-me-down Pentium 1 laptop, though the Optiplex was from a bio lab which could have been contaminated with biohazard waste or unstable radioisotopes):

"We were even told that Intel didn't have any motherboards lying around to use, so it had to search eBay and scrounge from dumpsters to find the Asus P/I-P55TZP4 pictured."

The Nokia 6110's 13mHz ARM7TDMI with a 5W TDP, likewise, could be emulated at 5mW on an 48mhz Ambiq Apollo3, a 1000x fold reduction in power. The original Nintendo, which ran at 1.79mhz, was emulated flawlessly on the microcontroller having 24x the speed. The Psion 3, ran at 3.84mhz, suggesting a hypervisor could emulate the software, if it could not be ported. Similar emulators exist: https://100r.co/site/uxn.html Thus porting user applications to a solar powerable hardware could be less a matter of rewriting all source code if an emulation layer can interpret the original code using a compatibility layer/ API.

Anything under 4MHz, like how NES was ported from STMF7 to Commodore based GEOS, https://github.com/hatonthecat/ENGAGE-GEOS could be ported to Ambiq Apollo3 and 4- Z80, 8086, 68000k, whichever you can run efficiently enough. Most ports still need the HAL (hardware abstraction layer written, even if CortexM4 has something on a related STM32, as can be seen in the below links)

C64 Emulator Implementation To port GEOS, a C64 emulator for ARM would need to include it. A C64 emulator has been ported to STM32F4: https://github.com/Staringlizard/memwa3 https://hackaday.com/2014/10/23/a-complete-c64-system-emulated-on-an-stm32/

CP/M: https://github.com/MockbaTheBorg/RunCPM (STM32 available)

and even EPOC16: https://zedstarr.com/2021/12/08/emulating-the-psion-mc400/

Kragen, prolific polymath, calls this type of GUI, "Basic interactive computing:"

Estimating the necessary performance for basic interactive computation: 0.1 DMIPS

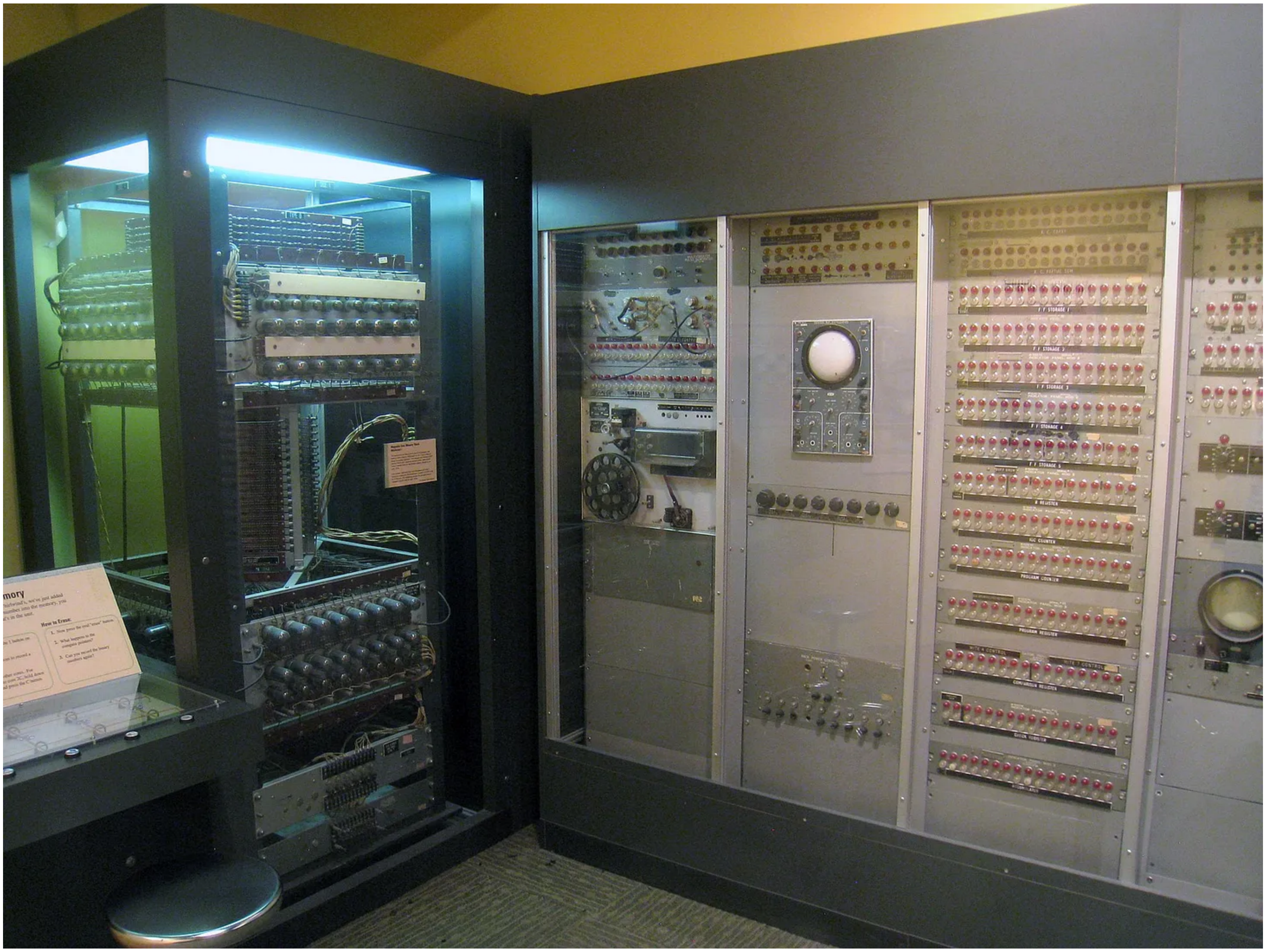
My previous estimate in Dercuano was that basic interactive computation like word processing takes about 7500 32-bit instructions per keystroke. At one point, I said, "WordStar on a 2MHz (≈0.5MIPS) 8-bit CPU would sometimes fall behind your typing a bit," but then later calculated that a Commodore 64 (now AR$12000 = US$80) or Apple][would only do about 200 000 8-bit instructions per second and were usable for word processing, and a 32-bit instruction is roughly equivalent to two 8-bit instructions, so you need about 0.1 32-bit MIPS, and you might be typing like 160 wpm (13.3 keystrokes per second), which works out to about 7500 instructions per keystroke.

Also in Dercuano, I estimated that painting text in a framebuffer fast enough that it doesn't slow down reading at 350wpm might take about 50 bytes of I/O per glyph and 100 instructions / glyph × 350 wpm × 6 glyphs / word × 1 minute / 60 seconds = 3500 instructions/second. But that's orders of magnitude lower than the computations I discussed above. Indeed, old computers like the Sinclair ZX-81 with its 3.25-MHz Z-80, lacking an external framebuffer, would use the CPU to repaint the screen 50 times a second.

Modern microcontrollers run at around 1 Dhrystone MIPS per MHz

How fast are modern microcontrollers? dannyf compiled Dhrystone 2.1 with a modern compiler and got 921 repetitions per second per MHz on an STM32F1 with what I guess is the vendor compiler, or 736 with GCC; to convert that into Dhrystone MIPS I think we divide by 1757, so that's 0.52 and 0.42 DMIPS/MHz. However, other commenters say the ARM Cortex-M0+ used in the STM32F1 is 0.93 DMIPS per MHz, and the STM32F103x8/STM32F103xB datasheet says it's actually "1.25 DMIPS/MHz (Dhrystone 2.1)". And apparently a CoreMark is about half a DMIP.

To make the solar analogy more complete (though not Turing complete), one needs to go even further back in time, to when mechanical computers were feverishly getting built in the 1940s. If you were to walk into Bletchley Park or the MIT Servomechanisms Laboratory during the height of the of the War effort and claimed that the machines that they were working on would be powered by the sun in 32 years, you'd probably be thrown, head first, out. Not only would it be completely irrelevant to the immediate application, but solar power hadn't even been discovered yet (unless you count George Cove) and it would probably seem offensive and outlandish considering how heavy and large the machines were. The first electromechanical computers probably had a lot more functions than a 7 function calculator. But which machine could be most closely linked to the equivalent number of computations?



"Scientific calculators are used widely in situations that require quick access to certain mathematical functions, especially those that were once looked up in mathematical tables, such as trigonometric functions or logarithms. They are also used for calculations of very large or very small numbers, as in some aspects of astronomy, physics, and chemistry.

The first scientific calculator that included all of the basic ideas above was the programmable Hewlett-Packard HP-9100A,[2] released in 1968, though the Wang LOCI-2 and the Mathatronics Mathatron[3] had some features later identified with scientific calculator designs."

The HP-9100A, the first standard scientific calculator

Just 10 years before the Teal Photon was released, a machine was designed that virtually no one would imagine would be solar powered just a decade later (At least 4-7 of the functions). Granted, Moore's Law was progressing much faster than it was now. But if you told someone in 1968 if the HP-9100A would be one day be powered by solar, the idea would appear unintelligible, and quite far-fetched, though you probably wouldn't get tossed out of *every* bar. If you asked someone in Palo Alto or Murray Hill, they'd probably say "far out…" (Photovoltaics had been discovered at Bell Labs in 1954).

> At a critical moment when the very science of computing stood at a crossroads, its future uncharted, they transformed the machine from a glorified calculator into the marvel of graphical communication it is today. Its role in modern life was far from preordained when PARC's scientists convened. They charted the course.

From *The Dealers of Lightning* (1999), by Michael A. Hiltzik

By the 1990s, solar calculators were inexpensive enough to be in every classroom, although they did not immediately lead to a new generation of microelectronics until over 30 years later when Industrial IoT devices using energy harvesting became identified as a cost-saving strategy (in both labor and occupational safety). As is usually the case, state of the art technologies usually get adopted first not by the mass market but by the academics, industrialists, and governments as can be seen with the first large computers being used for scientific, defense, accounting and insurance companies.

It's very likely that with increased sales of energy harvesting sensors to industry, that the technology will find its way into consumers for other purposes. There isn't an immediate need for energy scavenging cell phones to send SOS texts, unless one is in a high risk area:

“"As Hester explains in the video below, the technology behind the ENGAGE could be of use to people in space or high-risk environments who can't have their devices running out of juice. "No