3 Weird Machines from Serena Butler's TV Typewriter

by Travis Goodspeed

In the good old days, one could make the argument–however fraudulent!–that memory corruption exploits were only used by the bad guys, to gain remote code execution against the poor good guys. The clever folks who wrote such exploits were looked upon as if they were kicking puppies, and though we all knew there was a good use for that technology, we had little more than RMS's paranoid ramblings about fascism to present as a legitimate use-case. Those innocent days in which exploit authors were derided as misfits and sinners are beginning to end, as children must now use kernel exploits to program their own damned cell phones. If we as authors of weird machines are to prepare for the future, it might be a good idea to work out a plan of last resort. What could be build if computers themselves were outlawed?

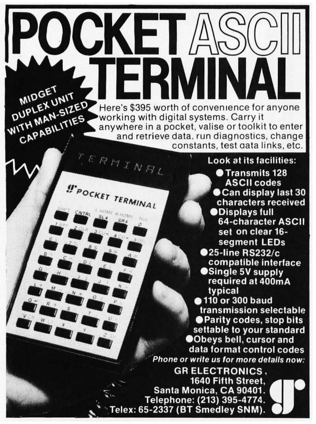
I'm writing to share with you the concept of a Butlerian Typewriter, loosely inspired by Cory Doctorow's 28C3 lecture and strongly inspired by many good nights of fine scotch with Sergey Bratus, Meredith Patterson, Len Sassaman, Bx Shapiro, and Julian Bangert. It's a little thought experiment about what weird machines could be constructed in a world that has outlawed Turing-completeness.

In the universe of Frank Herbert's Dune, the war on general-purpose computing is over, and the computers lost—but not before they struck first, enslaved humanity, and would have eliminated it if it were not for one Serena Butler. St. Serena showed the way by defenestrating a robotic jailer, leading the rest of humanity in the Butlerian Jihad against computers and thinking machines. Having learned the hard way that building huge centralized systems to run their lives was not a bright idea, humans banned anything that could grow into one.

So general-purpose computers still exist on the black market, and you can buy one if you have the right connections and freedom from prosecution, but they are strictly and religiously illegal to possess or manufacture. The Orange Catholic Bible commands, "Thou shalt not make a machine in the likeness of a man's mind."

Instead of general purpose computers, Herbert's society has application-specific machines for various tasks. Few would argue that a typewriter or a cat picture are dangerous, but your iPhone is a heresy. Siri would be mistaken for the Devil himself.

Let's simplify this rule to Turing-completeness. Let's imagine that it is illegal to possess or to manufacture a Universal Turing Machine. This means no ELF or DWARF interpreters, no HTML5 browsers. No present-day CPU instruction set is legal either; not ARM, not MIPS, not PowerPC, not X86, and not AMD64. Not even a PDP11 or MSP430. Pong would be legal, but Ms. Pac-Man would not. In terms of Charles Babbage's work, the Difference Engine would be fine but the Analytical Engine would be forbidden.



POCKET ASCII TERMINAL

MIDGET DUPLEX UNIT WITH MAN-SIZED CAPABILITIES

Here's $395 worth of convenience for anyone working with digital systems. Carry it anywhere in a pocket, valise or toolkit to enter and retrieve data, run diagnostics, change constants, test data links, etc.

Look at its facilities:

- Transmits 128 ASCII codes
- Can display last 30 characters received
- Displays full 64-character ASCII set on clear 16-segment LEDs
- 25-line RS232/c compatible interface
- Single 5V supply required at 400mA typical
- 110 or 300 baud transmission selectable
- Parity codes, stop bits settable to your standard
- Obeys bell, cursor and data format control codes

Phone or write us for more details now:

GR ELECTRONICS,
1640 Fifth Street,
Santa Monica, CA 90401.
Telephone: (213) 395-4774.
Telex: 65-2337 (BT Smedley SNM).

Now comes the fun part. Let's have a competition between Ada Lovelace and Serena Butler. Serena's goal is to produce what we will call a Bulterian Typewriter, an application-specific word processor of sorts. She can use any modern technology in designing the typewriter, as such things are available to her from the black market. She even has access modern manufacturing technology, so producing microchips is allowed if they are not Turing-complete. She may not, however, produce anything contrary to the O.C.B.'s prohibition against thinking machines. Nothing Turing-complete is legal, and even her social standing isn't sufficient to get away with mass production of computers.

So Serena designs a Butlerian Typewriter using black market tools like Verilog or VHDL, then mass produces it for release on the white market as a consumer appliance with no Turing machine included. One might imagine that she would begin with a text buffer, wiring its output to a 1970's cathode-ray television and its input to a keyboard. Special keys could navigate through the buffer. Not very flashy by comparison to today's tweety-boxes, but it can be done.

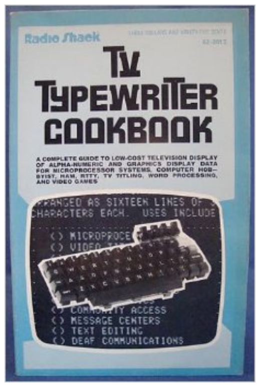
After this typewriter hits the market, Ada Lovelace comes into play. Ada's unpaid gambling debts prevent her from buying on the black market, so she has no way to purchase a computer. Instead, her goal is to build a computer from scratch out of the pieces of a Butlerian Typewriter. This won't be easy, but it's a hell of a lot simpler than building a computer out of mechanical disks or ticker-tape!

In playing this as a game of conversation with friends, we've come to a few conclusions. First, it is possible for Serena to win if (1) she's very careful to avoid feature creep, (2) the typewriter is built with parts that Ada cannot physically rewire, and (3) Ada only has a single machine to work with. Second, Ada seems to always win (1) if the complexity of the typewriter passes a certain threshold, (2) if she can acquire enough typewriters, or (3) if the parts are accessible enough to rewire.

As purpose of the game is to get an intuitive feeling for how to build computers out of twigs and mud, let's cover some of the basic scenarios. (The game is little fun when Serena wins, so her advocate almost always plays both sides.)

- If Serena builds her machine from 7400-series chips, Ada can rewire those chips into a general-purpose computer.
- If Ada can purchase thousands of typewriters, she can rewire each into some sort of 7400-equivalent, like a NAND gate. These wouldn't be very power-efficient, but Ada could arrange them to form a computer.
- If Serena adds any sort of feedback from the output of the machine to the input, Ada gets a lot more room to maneuver. Spellcheck can be added safely, but storage or text justification is dangerous.
- It's tempting to say that Serena could win by having a mask-programmed microcontroller that cannot execute RAM, but software bugs will likely give a victory to Ada in this case. This is only interesting because it's the singular case where academics' stubborn insistence that ROP is different from ret-to-libc might actually be relevant!

So how does a neighbor learn to build these less-than-computers, and how does another neighbor learn to craft computers out of them? If you are unfamiliar with hardware design languages, start off with a tutorial in VHDL or Verilog, then work your way up to crafting a simple CPU in the language. After that, sources get a bit harder to come by.


Radio Shack
TV TYPEWRITER COOKBOOK
A COMPLETE GUIDE TO LOW-COST TELEVISION DISPLAY OF ALPHA-NUMERIC AND GRAPHICS DISPLAY DATA FOR MICROPROCESSOR SYSTEMS, COMPUTER HOBBYIST, HAM, RTTY, TV TITLING, WORD PROCESSING, AND VIDEO GAMES
MESSAGE CENTERS
TEXT EDITING
DEAF COMMUNICATIONS

A primitive sort of Bulterian Typewriter is described by Don Lancaster in his classic article TV Typewriter from the September 1973 issue of Radio Electronics. His follow-up book, the TV Typewriter Cookbook, is as complete a guide you could hope for when designing these sorts of machines. Don Lancaster's book as well as his article are available for free on his website, but you'd do well to spend 15¢ on a paperback from Amazon.

Lancaster's TV Typewriter differs from Serena's in a number of ways, but chief among them is motivation. He avoided a CPU because he couldn't afford one, and he limited RAM because it was hellishly expensive in 1973. By contrast, Serena is interested in building what a brilliant engineer like Don might have made with today's endless quantities of memory and modern ASIC fabrication, while still avoiding the CPU and hoping to avoid Turing-completeness entirely.

In addition to Lancaster's book, those wishing to learn more about how to build fancy electronics without computers should buy a copy of How to Design & Build Your Own Custom TV Games by David L. Heiserman.

Published in 1978, the book is still the best guide to building interactive games around substantially analog components. For example, he shows how the paddles in a table-tennis game can be built from 555 timers, with the controllers being variable resistors that increase or decrease the time from the page blank to the drawing of the paddle.

To get some ideas for building computers out of twigs and mud, take a look at the brilliant papers by Dartmouth's Scooby Crew. They've built thinking machines from DWARF,[1] ELF,[2] and even the X86 MMU![3] I fully expect that by the end of the year, they'll have built a Turing-machine from Lancaster's original 1973 design.

Let's take a look at some examples of these fancy typewriters. I hope you will forgive me for asking annoying questions for each, but still more, I hope you will argue over each question with a clever neighbor who disagrees.

Simple BT: As a starting point, the simplest form of a Butlerian Typewriter might consist of a Keyboard that feeds into a Text Buffer that feeds into a Font ROM that feeds into an NTSC Generator that feeds into an analog TV. The Text Buffer would be RAM alternately addressed by the keyboard on the write phase and a line/row counter on the read phase. As the display's electron beam moves left to right, individual letters are fetched from the appropriate row of the Text Buffer and used as an address in the Font ROM to paint that letter on the screen.

This is roughly the sort described in Lancaster's original article. Note that it does not have storage, spell-check, justification, I/O, or any other fancy features, although he describes a few such extensions in his TV Typewriter Cookbook.

BT with Storage: There are a few different ways to implement storage. The simplest might be for Serena to battery-back the character buffer and have it as a removable cartridge, but that exposes the memory bus

[1]Exploiting the Hard Working Dwarf from WOOT 2011
[2]"Weird Machines" in ELF: A Spotlight on the Underappreciated Metadata from WOOT 2013
[3]Page Fault Liberation Army from 29C3

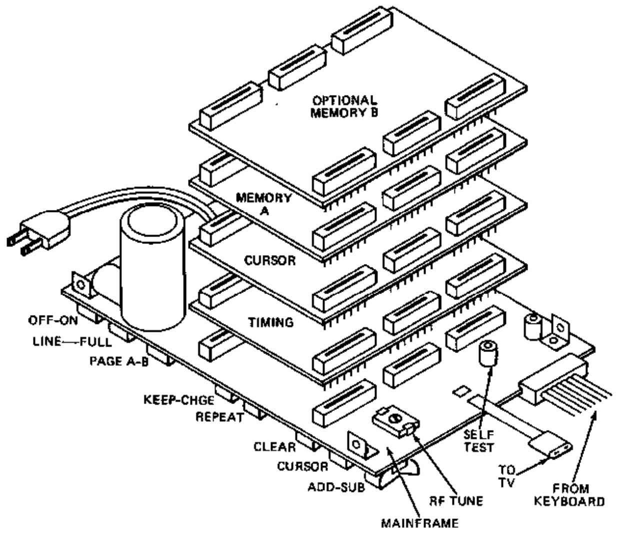


Figure 1: Don Lancaster's 1973 TV Typewriter

to Ada's manipulations. It's not hard to rewire a parallel RAM chip to be a logic gate by making its data a lookup table; this is how the first FPGA cells operated.

So if a removable memory isn't an option, what is? Perhaps Serena could make a removable typewriter module that holds everything but the keyboard, but that wouldn't allow for the copying of documents. Serial memory, such as an SPI Flash or EEPROM chip, is a possibility, but there's no good reason to think that it's any safer than parallel RAM.

A pessimist might say that external storage is impossible unless Ada is restricted to a small number of typewriters, but there's a loophole nearly as old as Mr. Edison himself. The trick is to have the typewriter flush its buffer to an audio cassette through a simple modem, and you'll find handy schematics for doing just that in Lancaster's book. Documents can be copied, or even edited, by splicing the tape in an old-fashioned recording studio.

Why is it that storage to an audio cassette is safer than storage to a battery-backed RAM module? At what point does a modem and tape become the sort of tape that Turing talked about?

BT with Spellcheck: Let's consider the specific case in which Serena has a safe design of a minimal typewriter and wishes to add spell check. The trick here is to build a hardware associative memory with a ROM that contains the dictionary. As the display's electron beam moves left to right, the current word is selected by division on spaces and newlines, and fed into the Spellcheck ROM, a hardware associative memory containing a list of valid words. The output of this memory is a single bit, which is routed to the color input of the NTSC Generator. With matching words in white and suspicious words in red, the typewriter could behave much like emacs' flyspell-mode.

So long as the associative memory is in ROM, this seems like a rather safe addition. What sort of dangers would be introduced if the associative spellcheck dictionary were in RAM? How difficult would it be to build a CPU from nothing but a few associative memory units, if you had direct access to their bus but could not change any internal wiring? How few memories would you need?

BT with Printing: Printing turns out to be much easier than electronic storage. The first method is to simply expose photographic film to the display, much as oscilloscopes were photographed in the good ol' days.

Another method would be to include a daisy wheel, dot matrix, or thermal print-head fed by a different Font ROM at a much slower scan rate. While much more practical than taking a dozen Polaroid photographs, it does give Ada a lot more room to work with, as the wiring would be exposed for her to tap and rewire.

I don't expect general purpose computing to be outlawed any time soon, but I do expect that the days of freely sharing software might soon be over. At the same time that app stores have ruthlessly killed the shareware culture that raised me as a child, it's possible that someday exploit mitigations might finally kill off remote code execution.

At the same time that we fight the good fight by developing new and clever mitigation bypasses, we ought to develop new and clever ways to build computers out of whatever scraps are left to us when straight-jacketed in future consumer hardware. Without Java, without Flash, without consistent library locations, without predictable heap allocations, our liquored and lovely gang continues to churn out exploits. Without general-purpose computing, could we do the same?

Please share this article with a neighbor,
and also share a bottle of scotch,
and argue in the kitchen for hours and hours,
—Travis