

THE KILLER EMULATOR FROM HELL

-----  
John Pazmino  
NYSkies Astronomy Inc  
nyskies@nyskies.org  
www.nyskies.org  
1993 September 1

The Killer Emulator From Hell  
Sinc-Link  
Author(s): John Pazmino  
Date: September/October 1993  
Volume: 11 Issue: 5  
Page(s): 6-9

Introduction/overview of Z80, Gerton Lunter's shareware emulator  
for PCs. Copied from the Sept 1993 issue of the LISTing  
Newsletter from the Long Island Sinclair/TimeX Users Group.

THE KILLER EMULATOR FROM HELL by John Pazmino

By now the supplies and sources for Sinclair hardware are quite dwindling. It is tough to introduce newcomers to Sinclair for the general lack of apparatus to outfit them with. On top of this is the pervasion of the IBM type of computer among the vulgate which works fiercely against adopting Sinclair as a new platform.

What to do? In the UK, where Sinclair still rules in the 8-bit computer world, there were efforts to work a software solution: Turn the IBM into a Sinclair. In principle this is easy because the Z80 CPU architecture has been emulated on the 8088 (and higher) chips thru software. Many readers will remember, and perhaps still have, the CP/M emulators on the early IBM rigs.

However, emulators for the Sinclair have been, well, eh!. Why? Mainly they are written by Sinclair folk who on the whole are unversed in IBM. In deed, some Sinclair emulators are nothing but the Z80 code of the Sinclair ROM shoved into a Z80 CPU emulator. This was for many years not cricket because the code was the property of Sinclair and then Amstrad.

There's a physical barrier, too. Except for the very first IBM PC, issuing simulataneoulsy with the Sinclair ZX-81, the IBM has no innate means of receiving input from a cassette. Some emulators simply gave up at this obstacle and work only with type-in programs.

Well, now in this merry year of 1993 comes the Killer Emulator from Hell, a Sinclair emulator for the IBM that does everything a Sinclair emulator should do and does it right. This new emulator, Z80, is a shareware creation from Europe.

Shareware, not lucreware.

This point is crucial. For in early 1993 Amstrad, who holds the rights for the Spectrum and QL, formally threw the code for the ROMs into public use. That is, anyone may now copy and distribute the original ROM code in their own products PROVIDED that these products are noncommercial. Commercial use of the ROM code is still prohibited. Ergo, altho Z80 does have woven into it native Sinclair code it is copasetic and quite kosher.

Z80 dissolves the above -- and many many other -- problems in bringing the Sinclair to the IBM. It comes on an IBM stiffy with 720K of files. They rehydrate to about 2M on your harddisc. Two megabytes! That's, um, more than thirty Spectrumsful! What ARE all these crazy files!? Relax, already. Most of the files are sourcecode and literature. **You can shiv them, after printing or copying them off,** leaving 'just' 330K of working files. That's STILL about five Spectrumsful of stuff. For emulating a Spectrum?

They toto in uno are a symphony of several Sinclair systems: the Spectrum 48K model, 128K model (with pixel graphics and multichannel sound), Interface 1 (with serial ports), Sinclair and Kempston Joysticks, Multiface 1 (with memory capture), tape loading indicator, Z80 dissembler and monitor, header-reader, screen editor, RAMdisc, swappable ROMs, Microdrives, Disciple discs, and (of course!) tapedrive. All of these are provided via software in quite perfect replication of the original hardware gadgets. Thus, the complete inability to attach native Sinclair accessories to the IBM is much overcome by building the most crucial ones right into the emulator.

The emulator receives its original input from cassette only. This requires a cable connecting the IBM parallel port to the cassette deck, with some circuit bits along the way. The emulator has clear instructions for making this cable and it took me an afternoon to build it, including a stopoff on Canal Street to get the parts.

If you in giddy delirium shoved Z80's disc into your IBM without

between Sinclair's CR-only and IBM's CR/LF line terminations. All these conversions use the IBM file as the working medium.

The Microdrives are mimicked on IBM file. There are eight 'microdrives' in the emulator, the maximum capacity of the original Interface, and each 'cartridge' is an IBM file. You 'slot' a microdrive by allocating a file to a drive. Ah!, to use a new cartridge you must 'format' it ("FORMAT "m";3;<name>"; hey!, those extra IBM keys ARE cool!). This creates a new IBM file 137K long with 126K of 'tape'. Two of these fit on a 360K floppy or five on a 720K stiffy. Once you format an emulated cartridge you can work with it exactly as you would a physical cartridge. You even pull a 'catalog' of the file and 'erase' stuff from it!

The Disciple disc is, too, cloned in Z80, altho the United States never enjoyed this system. Again, the IBM file is the working medium. Being that on stateside we deal with many minor disc systems, can Z80 handle, say the Zebra system? Now comes the freako part. The code for the Disciple system is excisable from the primum corpus of the emulator. YOU CAN REPLACE IT WITH THE OPERATIONS OF YOUR PECULIAR DISC SYSTEM. Yes!, you may ultimately junk the hardware of the Zebra system and run everything from the Zebra code you wrote into Z80.

The total supplantation of Sinclair's physical media with IBM files lets you jettison just about every disc and cartridge utility in sight. With your stuff in IBM files you can apply any and all of the IBM file utilities on it. Farewell, Cartridge Doctor! Vale, KopyKat!

Wait a minute!!! What happens to all those luscious Sinclair cartridges and discs in those milkcrates? Since you simply can not feed them to the IBM you must revert to the original tapes. Load the files into Z80 from the tapes and save them onto the emulated disc or cartridge. Without such prime tapes you may have one revolting job before you! You must transfer the disc or cartridge files back to cassettes and then procede as just described.

Communications thru the emulator use the cloned serial ports of the Interface 1. Remember my series a year back on PostScript on the Sinclair? (Yesyesyes, I know, LISTings missed out the fourth and final part.) Now you can actualize this by running Z80 on an IBM fitted with a PostScript printer. But there's a weirder prospect, attainable with Z80: Pass data from the Spectrum to an other IBM program. You in this case do not need a PostScript printer; use your existing printer! You run a PostScript software emulator like Emulaser or GhostScript on the IBM and send Spectrum generated PostScript files to it. Ugh! such disgustingly gorgeous output. From a Spectrum. FROM A SPECTRUM!

Because the emulator is European the presumption is that you use the serial port for printing and the instructions detail conversing with a printer thru it. In the US printers are routinely hung from the parallel port and the serial port is the avenue to a modem. Hence, in making the cassette cable, include a 'Y' connector or A-B switch so the printer and cassette can coexist. To accommodate the possibility of a parallel printer, Z80 allows a redirection of output to LPTx.

However, there is a clumsiness in using the printer, one of the [very] few downpoints of Z80. The LPRINT, LLIST, and COPY commands do not fire characters directly to the attached printer. You have to first open a channel to the printer ("OPEN #3,"t") and then send output to that channel. I already wrote, via Internet, to the author about this and suggested that he make LPRINT, LLIST, and COPY send output to a DOS printer driver of the sort included with word processors. If he can work this into a future edition of Z80 you'll be printing to whatever device you got attached to the IBM.

What the deal about other ROMs? You recall that the American flavor of Spectrum, the Timex 2000, has a dockport into which an external ROM plugged to override the onboard ROM. Also, when the Zebra

making the connector, chill out! Z80 comes with seven ready-to-run Spectrum programs. Nothing fancy, some games and utilities.

I can not here elaborate on the very many details of this emulator. That would amount to describing the entire Spectrum world! I here highlight a few major features. This emulator, for starts, in fact does what every Sinclair fan sweats in sleep for: IT BODILY TRANSFERS TAPES TO DISC. Yes, it takes the files from tape and mirrors them on a regular IBM file. And this file to the emulated Spectrum quacks and flies and waddles exactly like the original tape. The major positive(!) difference is that you never 'spot' or 'rewind'. This feature alone virtually eliminates the 'tape loading error' from a tapefile that failed to catch. It'll pass around again in, oh, a millisecond for another go. Each 360K disc holds several, depending on length, cassettes of programs.

With Z80 you may choose between a replica of a cassette OR AN ORDINARY IBM FILE. That is, you may load from EITHER the emulated tape OR from an IBM file that contains the program in DOS form! Hand up? Yes? Sure, Z80 converts the one kind into the other!

You over there? Voce alta, de favore. OK, you have several short tapes or programs and you want to combine them on one cassette. What a magilla on the real Sinclair! Load from one tape; swop tapes; spot it; save to it. Swop for the next tape ... . With Z80 you merely knit together the separate 'tapes' in any order you want and get one consolidated new 'tape'. Yes, that right. Uh, let's continue, please?

These grand goodies so far are alone enough to justify the nuisance of reaching overseas for this emulator. In one weekend you can put your entire Spectrum collection onto disc WITH ABSOLUTELY NO MODIFICATION OF THE ORIGINAL CODE. You 'bung the tape' by specifying the tape's IBM file, do a LOAD and the 'tape' goes ahead and loads.

Please do understand that this is utterly NOT a 'RAMdump', 'memory capture' or 'snapshot'. Z80 does this, too, as an altogether separate function. In the tape mirror each file of the tape is actually in the IBM file and you even use the (included!) header-reader to see them. What's more, the header-reader browses the tape and loads ANY tape file you want. You don't have to let the tape run thru to load the program way off at the tail end. Hmmm, a random-access cassette tape.

The Spectrum keyboard is exactly mapped to the IBM keyboard. You use all the keywords and tokens. Being that the IBM has no Sinclair keytops, you popup a Sinclair keyboard diagram. It's really a rather faithful depiction of the chicklet Spectrum with the corner colorband and all that. There is no such mapping for the Spectrum 128K because this model does not use keywords and tokens. You type in everything litteratim with all the regular IBM keys.

Besides the replicated Spectrum keys, the extra IBM keys are energized. You, for instance, get the <=> symbol by <sym<L>> or by just punching the <=> key. Either the IBM <alt> or <ctl> keys stands for the Sinclair <sym> key. I do see a danger in this convenience! Play with the Spectrum-in-IBM for a while. Then go back to the real Spectrum. Where the eff is that <[> symbol!?

The numberpad is the cursorpad, the Sinclair joystick, or the Kempston joystick -- as you wish by selection. The cursorkeys work, too, for editing the command line. <esc> is the EDIT key, as is <sft<1>>; <bsp> and <del> do DELETE along with <sft<0>>.

The IBM functionkeys are the adit to the emulator's forest of functions, with <F1> being the general 'help' feature and <alt<F1>> popping up the Spectrum keyboard layout.

When you do a screensave ("SAVE <name> SCREEN\$"), THE SCREEN\$ FILE CAN BE SHARED WITH OTHER IBM PROGRAMS. What?! Uh, you see, this emulator converts a Spectrum SCREEN\$ file into a GIF or PCX file! You share textfiles, too, with other IBM programs by a conversion

