

ZX Spectrum Next Hardware

(v1.1 - 12/10/2017)

This document will detail proposed additions to the ZX Spectrum Next machine in order to help operational needs, and graphical ones.

Copper

This module is dedicated to writing Next Registers at specific points on the display. Just like the Amiga copper list it consists of only 2 commands. WAIT and MOVE. This simple control allows updating of registers as the display renders.

ZX Spectrum Next Copper

The copper (as used in the Amiga) is a simple beast, with only 3 commands; **WAIT**, **MOVE** and **SKIP**. We won't be using the **SKIP** as we've compressed the copper into 16 bits rather than 32.

WAIT will pause processing until a certain point on the display (to a fixed resolution). This lets developers update things like scroll registers, colours, display locations and all from a simple command list, without having to do Raster IRQs and then trying to get timing right for nice clean splits. Only the lower 128 registers can be written to. (currently not an issue)

The copper is (currently) a maximum of 1K in size (512 instructions).

Writing to a **READ ONLY** register (like register 0) will act as a "NOP", and this means you can wait for a low res horizontal dot value, then output NOPs to get to a more precise location. This reduces the impact of the lowres horizontal dot clock.

WAIT	Wait for a specific point on screen	16 bit
	Specify WAIT instruction	15 = 1
	Horizontal position (0-55, every 8 pixels)	9-14 (6 bits)
	Raster line to wait for (0-311)	0-8 (9 bits)
MOVE	Move a value into a NextReg	16 bit
	Specify MOVE instruction	15 = 0
	Next register (register 0-127)	8-14
	Byte value to write	0-7

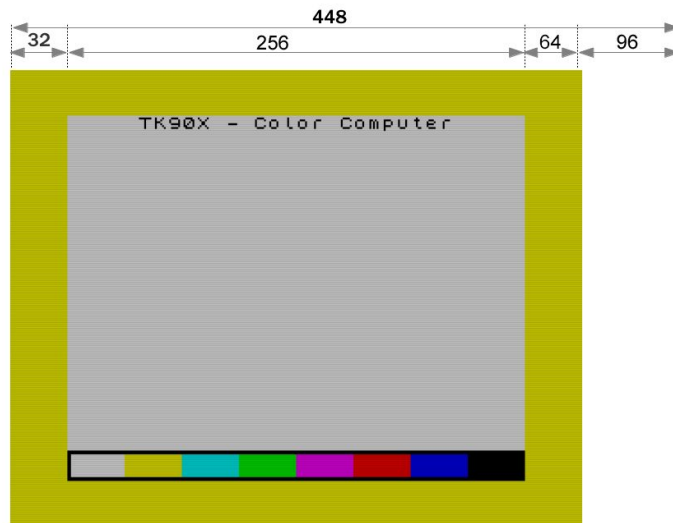
Details on the Amiga copper can be found here...

http://amigadev.elowar.com/read/ADCD_2.1/Hardware_Manual_guide/node0047.html

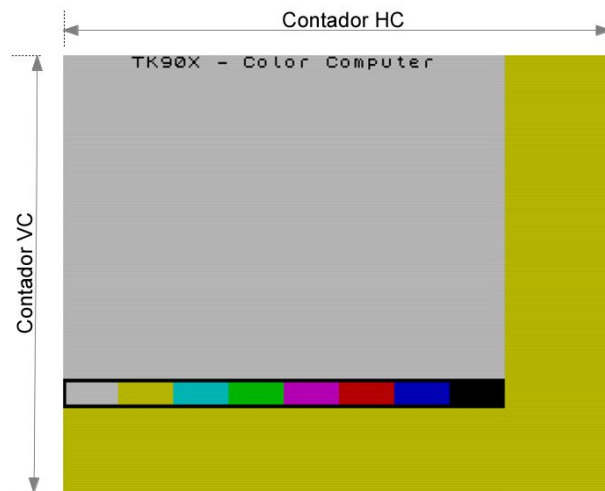
Register	NextReg
Copper Data (8 bits wide, must be 2 bytes at a time) BIG Endian format!	\$60
Copper Control (16 bits) Bits 15-14 (2 bits) %00 = STOP %01 = START then stop at end of stream and don't loop %10 = Free running will run and loop continuously %11 = START and automatically restart at next VBlank Bits 13-9 (5 bits) MUST be %00000 Bits 8-0 (9 bits) Instruction Index (0 to 511)	\$61 LO \$62 HI

Screen layout

The screen looks like this, with 312 scanlines. So the horizontal dot clock being from 0 to 55 (0 to 440). However users should be aware that 0,0 (raster,HPOS) is not the top left border!



Shown here is the actual raster and HPOS offsets. As you can see 0,0 (raster,hpos) is actually the first pixel of the screen, so care should be taken to wait until the last line so you can set up the next frame correctly if required.



Simple Examples

Auto swap turbo mode

Commands	Values	
WAIT 193,0	\$80c1	Wait for just into top/bottom border (untested line number)
MOVE 7,2	\$0702	Set CPU into 14Mhz mode
WAIT 311,0	\$8137	Wait for line before start of screen
MOVE 7,1	\$0701	Set CPU into 7Mhz mode
MOVE 0,0	\$0000	Fill rest of list NOP to auto restart.... (when using mode %10)

Split screen scrolling

Commands	Values	
WAIT 310,0	\$8136	Wait for before start of screen
MOVE 22,??	\$16??	Set Layer 2 X scroll
MOVE 23,??	\$17??	Set Layer 2 Y scroll
WAIT 96,0	\$8060	Wait for middle of the pixel screen
MOVE 22,0	\$1600	Reset Layer 2 X scroll
MOVE 23,0	\$1700	Reset Layer 2 Y scroll
MOVE 0,0	\$0000	Fill rest of list NOP to auto restart.... (when using mode %10)

This system can also obviously be used for colour changes to get sky effects, layer priorities, enabling/disabling screen modes etc.