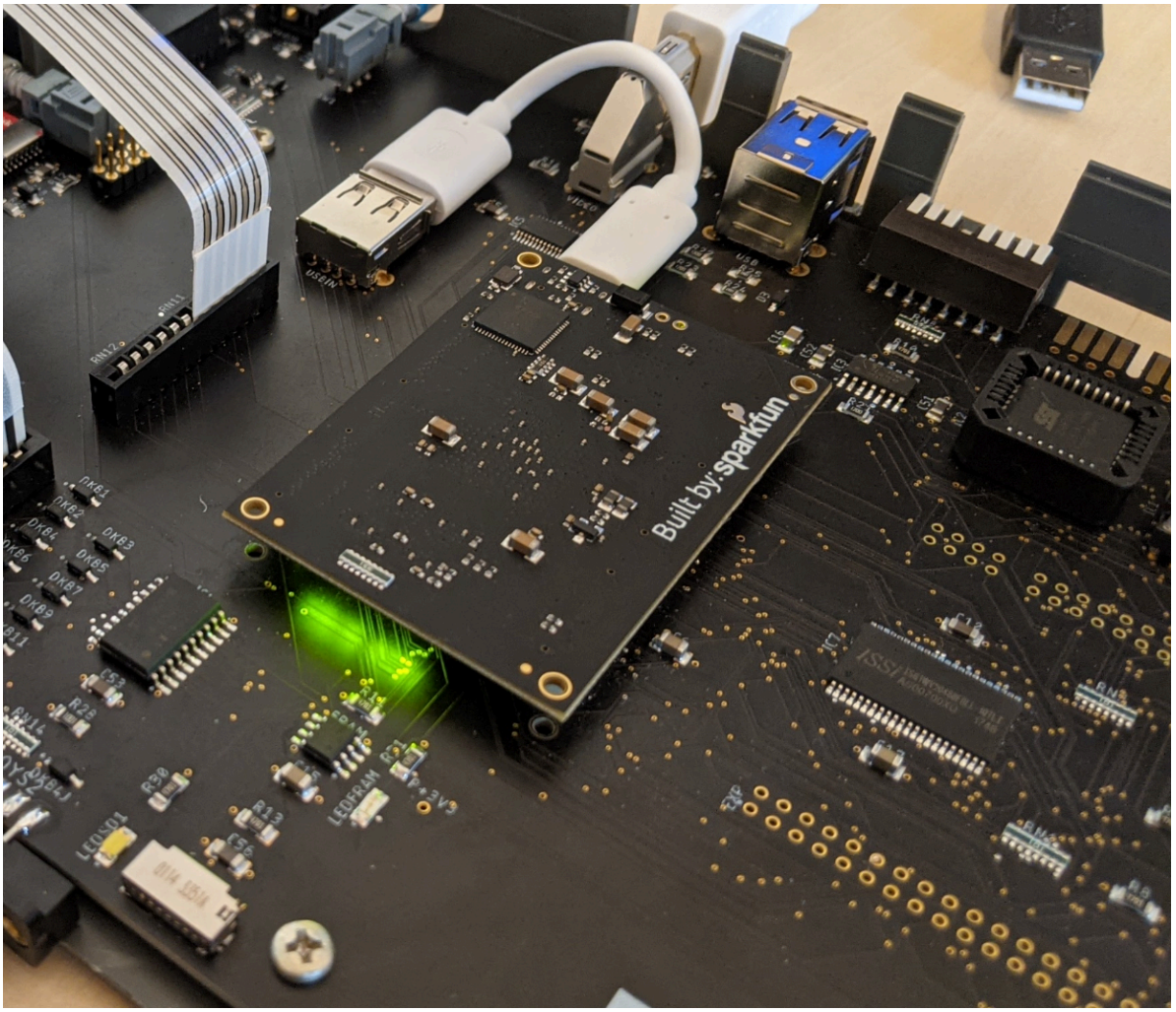
Programmer's Reference

LMN128 / Jan Kucera



version 1.28 / updated 2024-05-18

Content

Notes:

I/O ports in the eLeMeNt ZX hardware

The port layout was done in accordance with the **specification ZXi**: https://zxdesign.itch.io/zxi

All ZXi and other new eLeMeNt ZX ports are writable and readable.

Note: A bit with a prefix means the signal is active in 0. The signal without a prefix is active in 1.

ZXi 16-bits ports - general range

common with other devices like the ***MB03+ Ultimate*** interface

```
31803  h7C3B   port0 - select port column hx00 - hxFF
32059  h7D3B   port1 = column1/row1 = g_zxi_000 to g_zxi_0FF
```

eLeMeNt range

```
30779  h783B   port0 - select port column hx00 - hxFF
31035  h793B   port1 = column1/row1 = e_zxi_000 to e_zxi_0FF
```

ZX Uno range

```
64571  hFC3B   port0 - select port column hx00 - hxFF
64827  hFD3B   port1 = column1/row1 = u_zxi_000 to u_zxi_0FF
```

Programming example

```
ld bc,h783B          //ZXi portselect
ld a,2               //Audio
out (c),a
ld bc,h793B
ld a,%11111110       //disable ULA sound
out (c),a
```

Other ports

Fast 8bit extra memory ports
Old write-only ports made readable (where possible)
Reserved ports - in the Annex of this manual

General ZXi ports

g_zxi_000 - Device select

bit0 = 0-external/1-internal select device (for features like Ultimate memory mode)

g_zxi_001 - Extra graphics modes

00 - none
01 - HiResColour(HRC) - 512x192 pixels / 64x48 attributes / Spectrum attributes
02 - HiResindeXedColour(HRXC) - 512x192 pixels / 64x48 attributes / indeXedColours
03 - HGFX 1 - 256x192 pixels - two buffers - ZX/linear memory layout
04 - HGFX 2 - 512x192 pixels - one buffer - linear memory layout
05 - HGFX 3 - 512x384 pixels x 16 colour - one buffer - linear memory layout
06 - HGFX 4 - 720x546 pixels x 4 colour - one buffer - linear memory layout (Chunky mode only)

g_zxi_002 - Advanced video parameters

bit0 = 1 - activate first video buffer to page 6 instead of page 5
bit1 = 1 - enable BigBorder in HGFX modes (obsolete -don't use it)

g_zxi_003 - KeyLayer mode

00 - none
01 - KeyLayer - a key colour in the first screen is transparent for the second one (and vice versa), using ZX128 VRAM0 and VRAM1)

g_zxi_004 - KeyLayer mode additional parameters

Key Layer only - value of Key colour 0-15 for ZX Screen and HiResColour
value of Key colour 0-31 for HiResindeXedColour

g_zxi_008 - GigaScreen - old TV/Monitor sum colours effect

00 - none (disabled)
01 - GigaScreen mode0: 2 VRAMs always mixed
02 - GigaScreen mode1: two video frames mix
03 - GigaScreen mode2: auto detect mode1

g_zxi_009 - ScanLine (old CRT/TV) effect

00 - none (disabled)
01 - 1/2
02 - 1/4
03 - 1/8
04 - 1/16
05 - 1/32
06 - 1/64
07 - 1/128

g_zxi_010 - Ultimate layout type activate/Extra memory mode - reflects g_zxi_000

00 - none
01 - Ultimate Full mode - all RAM/FLASH/FRAM access (dev. purposes only)
02 - Ultimate Protect mode - access to user RAM only. System regions as DivSD, ZX, MB02+ is protected

g_zxi_011 - RAM in Ultimate mode - low byte - reflects g_zxi_000

read only - number of free 8KB banks - low byte

g_zxi_012 - RAM in Ultimate mode - high byte - reflects g_zxi_000

read only - number of free 8KB banks - high byte

g_zxi_01C - HGFX line interrupt Y position - low byte - default 0

g_zxi_01D - HGFX line interrupt Y position - high byte - default 0

g_zxi_01E - HGFX beam Y line position - low byte - reflects g_zxi_000

read only - low byte

g_zxi_01F - HGFX beam Y line position - high byte - reflects g_zxi_000

read only - high byte

g_zxi_020 - HGFX - general settings

bit0 = memory layout: 0 - standard ZX; 1 - linear
bit1 = border colour: 0 - standard ZX; 1 - indexed (index 0-7)
bit2 = ZX transparency: 0 - enable; 1 - disable
bit7 = write to HGFX plannar videoram, registers & colour table:
0 - enable; 1 - disable

If index colour no. 0 is used, then an attributed pixel from the ZX-screen will be displayed at the place of the HGFX pixel.

g_zxi_021 - HGFX - high byte of Videoram area

h40 - default

g_zxi_022 - HGFX - high byte of Registers area

h5B - default

g_zxi_023 - HGFX - high byte of Indexed colour table area

start address for a table of 768 bytes, 256 positions
h5C - default

g_zxi_02F - HGFX status

bit0 = 0 - ready; 1 - busy (e.g. erase/copy full buffer)

g_zxi_030 - UART speed - read, UART buffer FIFO erase - write

read gets last UART speed written to port #143B
write #01 cases fast FIFO buffer erase

g_zxi_040 - FastDMA - mode

bit0 = 0-disable/1-enable FastDMA
bit1-7 = reserved, must be 0

g_zxi_041 - FastDMA - high byte of Registers area

h5D - default

g_zxi_048 - SoundDMA - mode

bit0 = 0-disable/1-enable SoundDMA
bit1-6 = reserved, must be 0
bit7 = 0-enable; 1-disable write to registers

g_zxi_049 - SoundDMA - high byte of Registers area

h5E - default

g_zxi_04A - SoundDMA - channel #A high byte of current sample position

read only

g_zxi_080 - SafeNMI

bit0 = 0-disable/1-enable RAM Write protection
bit2-6 = reserved, must be 0
bit7 = 0-enable/1-disable push NMI button

g_zxi_081 - SafeNMI - low byte

read only

g_zxi_082 - SafeNMI - high byte

read only

g_zxi_0FF - FPGA Core FLASH control

internal use only

eLeMeNt ZXi ports

e_zxi_000 - Machine - memory/timing mode

```
00 - ZX Spectrum 48
01 - ZX Spectrum 128+2
02 - reserved
03 - ZX Spectrum 128+2A
04 - reserved
05 - Pentagon 128
06 - Pentagon 512
07 - Pentagon 1024 v2.2
08 - Pentagon 4096 (KAY/Profi/
Scorpion/Phoenix/ATM1)
09 - Didaktik Gama 80
```

Note: Switching a machine does not switch ROMs. See **e_zxi_00F**.

e_zxi_001 - Storage/Memory Interface

```
00 - none
01 - DivSD (512KB divIDE)
02 - DivSD+MB (512KB divIDE & MB02+)
03 - SD only
FE - Rescue mode
FF - obsolete - 
```
don't use it! (former Ultimate mode, use **g-zxi_010** instead)

e_zxi_002 - Audio output

```
bit0 = enable ULA audio
bit1 = enable TurboSound FM
bit2 = enable MonsterBlaster (SounDrive, etc.)
bit3 = enable SID (C64 sound chip)
bit4 = enable SAA1099 (SamCoupe sound chip)
bit5 = enable TAPE input
```

e_zxi_003 - Joystick

```
00 - none
01 - Kempston
02 - Sinclair Left
03 - Sinclair Right
04 - Keyboard (OPQA+Space+mn+bhr)
```

e_zxi_004 - Mouse/Keyboard

```
00 - none
01 - K-Mouse Master (external keyboard disabled)
02 - K-Mouse Slave (external keyboard disabled)
03 - External Keyboard
04 - External Keyboard + K-Mouse Master
05 - K-Mouse Master + Slave (external keyboard disabled)
```

e_zxi_00F - Flash memory (ROM Areas) - *eLeMeNt ZX v2*

```
00 - Area 0
01 - Area 1
02 - Area 2
03 - Area 3
```

Note: For the *eLeMent ZX v1*, see mechanical DIP-switching, in the *User Manual*.

e_zxi_010 - Timex graphics modes

bit0 = enable Timex gfx mode

e_zxi_011 - ULAplus

bit0 = enable ULAplus
bit1 = enable Timex mode select by ULAplus regs

e_zxi_018 - TurboSound FM

bit0 = select TSFM chip model: 0-AY, 1-YM
bit1 = select TSFM Stereo mode: 0-ACB, 1-ABC

e_zxi_019 - SID

bit0 = select SID chip model: 0-6581, 1-8580
bit1 = select SID frequency: 0-PAL, 1-NTSC

Note: This port is for SetUp only! In applications, use ZXi SID Control registers **h6C3B** and **h6D3B.**

e_zxi_01A Joystick/GamePad type

00 - standard 8-bits era
01 - Sega controller (3-8 buttons)

e_zxi_01D - GigaScreen - old TV/Monitor colours sum effect

00 - none (disabled)
01 - GigaScreen mode0: 2 VRAMs always mixed
02 - GigaScreen mode1: two video frames mix
03 - GigaScreen mode2: auto detect mode1

e_zxi_01E - ScanLine (old CRT/TV) effect

00 - none (disabled)
01 - 1/2
02 - 1/4
03 - 1/8
04 - 1/16
05 - 1/32
06 - 1/64
07 - 1/128

e_zxi_01F - Digital Video Interface - sync and other parameters

bit0 = enable legacy DVI - use on older monitors
bit1 = enable INT frame synchronization for Ultimate timing - works only on some monitors

e_zxi_020 - EXTRA button - short press

00 - none (only hold CPU)
01 - CPU speed
02 - Machine
03 - GigaScreen
04 - Warm reset
05 - Joystick/Gamepad interface mode

e_zxi_021 - EXTRA button - long press

00 - none (disabled)
01 - Spectrum menu

In the “Spectrum Menu” you see the speccy-rainbow in the border and you can press a key:

“1” - CPU speed 3.5 MHz
“2” - CPU speed 7 MHz
“3” - CPU speed 14 MHz
“4” - CPU speed 20 MHz
“5” - CPU OverClock 22MHz
“6” - CPU OverClock 28-MHz - 1T delay in ROM
“7” - CPU OverClock 28MHz - RAM only software
“8” - CPU OverClock 30MHz - RAM only software

e_zxi_02B - RTC

```
00 - none (disabled)
01 - FPGA internal - batteryless
```

e_zxi_02C - Z-Controller SD

```
00 - none (disabled)
01 - on first SD slot
02 - on second SD slot
```

e_zxi_02D - divSD

```
bit0 = select DOS rom: 0-FPGA(rescue esxDOS), 1-FLASH
bit1 = swap order first/second SD
bit7 = keep active over other memory mode
```

e_zxi_02E - DMA

```
bit0 = enable
```

e_zxi_02F - IO LIFO stack

```
bit0 = enable LIFO
```

e_zxi_030 - CPU speed

```
00 - 3.5 MHz
01 - 7 MHz
02 - 14 MHz
03 - 20 MHz
04 - OverClock mode 22MHz
05 - OverClock mode 28MHz - 1T delay in ROM area
06 - OverClock mode 28MHz - no delay - for RAM only software
07 - OverClock mode 30MHz - 1T delay - for RAM only software
```

e_zxi_031 - CPU OverClock mode

```
bit0 = enable OverClock
```

OverClock mode is intended for an experimental use only, because the system with an overclocked CPU may not work properly. You can try this mode to play games and view demos. It is NOT recommended for software that uses system-critical functions such as writing to an SD card. In this case, data stored on the SD card may be destroyed.

To set the Overclock mode, use the external PC keyboard or the Spectrum menu. Speeds of OverClock modes 06 and 07 (without 1T-state delay) are too high for the internal FLASH memory, therefore you must apply it on software which does not use a ROM.

e_zxi_032 - CPU speed alternative change*bit0* = enable ZX Uno speed change

```
bit1 = enable Pentagon 1024 v2.2 turbo
```

e_zxi_038 - Expansion pack A

```
00 - none (disabled)
01 - Memory 2048KB extension (AA-RAM)
```

e_zxi_039 - Expansion pack B - UART(WiFi) and second PS2

```
00 - UART - internal USB-UART (MB03+ mode) / PS2
01 - UART MB03+ / PS2
02 - UART ZXUNO / PS2
03 - UART ZX128-AY / PS2 disabled
```

e_zxi_03E - RTC Time Zone

```
00 - GMT+0          05 - GMT+5          hFB - GMT-5
01 - GMT+1(Zlin)      ..                hFC - GMT-4
02 - GMT+2          hF8 - GMT-8         hFD - GMT-3
03 - GMT+3          hF9 - GMT-7         hFE - GMT-2
04 - GMT+4          hFA - GMT-6         hFF - GMT-1
```

e_zxi_03F - Diagnostic - for internal development only - can be changed

```
00 - none (disabled)
01 - Video sync.
02 - Audio level
03 - CPU speed
FE - Video Test Text Pattern
FF - Video Test Colour Moving Pattern
```

e_zxi_0EF - Interface FPGA ID

```
- zxi_0EF - FPGA ID: 00-Au,01-AuPlus
```

e_zxi_0F3-0F8 - Real ZX Spectrum Clock Counter

Size of this Time Counter is 48 bits.

```
- zxi_0F3                    - zxi_0F6
- zxi_0F4                    - zxi_0F7
- zxi_0F5                    - zxi_0F8
```

Registers are locked when read occurred on reg zxi_0F3 to allow correct reading

e_zxi_0F9-0FF - Interface ID and Core Time Stamp

```
- zxi_0F9 - seconds              - zxi_0FC - days
- zxi_0FA - minutes              - zxi_0FD - months
- zxi_0FB - hours                - zxi_0FE - years

- zxi_0FF - Interface ID: hE1 for e-eLeMeNt ZX and 1-board version v1.x
```

ZX Uno ZXi ports

Note: ZX-Uno hardware ports, incl. four registers related to Radastan, are fully described on webpages https://www.zxuno.com/wiki/index.php/ZX_Spectrum#Nuevos_registros_E.2FS_para_control_de_ZX-Uno and http://uto.speccy.org/zxunofaq_en.html#programming

u_zxi_004 - SCANCODE

Gives the value of the last scancode generated by the keyboard. Use Scan code set 2.
More info: https://wiki.osdev.org/PS/2_Keyboard (use Scan code set 2.)

u_zxi_005 - KEYSTAT

Several bits that indicate whether or not there is a new key pressed, or released, and whether this is an extended or normal key.

bit 0 = PEN	value 1 indicates new data is ready to be read in the SCANCODE register after reading KEYSTAT, this bit becomes 0
bit 3 = ERR	value 1 indicates that the transmission from the PS/2 port had errors
bit 7 = BSY	value 1 indicates that the transmission from/to the PS/2 port is busy

u_zxi_009 - MOUSEDATA

Used to read/send direct commands to the PS/2 mouse. For example: to initialize the mouse, the value hF4 must be sent to this register.

u_zxi_00B - SCANDBLCTRL

bit 0-5 = not used
bit 6-7 = **TURBO** (ZX Uno Z80-CPU speed)

00	3.5 MHz
01	7 MHz
10	14 MHz
11	20 MHz (the highest speed of the real Z80-CPU in the eLeMeNt ZX)

u_zxi_040 - RADASCTRL

bit 2-7 = not used
bit 0-1 = value **b11** to turn radastan on

u_zxi_041 - RADASOFFSET

Contains the number of bytes that must be added to the base address of the screen (h4000h, h6000, hC000 or hE000 depending on the configuration) to obtain the address where the first two pixels are located in Radastan mode.

It is a 14 bit register. To write a value, the 8 least significant bits are written first, followed immediately by the 8 most significant bits (the 2 most significant bits of this value are ignored).

If the offset value is such that scanning the screen memory to create the image reaches the end of a 16KB page, the scan will continue from the beginning of that same page.

64 steps will produce **verticall scroll**.

u_zxi_042 - RADASPADDING

Contains the number of bytes (+64) that a scanline occupies in Radastan mode.

If this register is 0, the length in bytes of a scanline is 64 bytes (128 pixels).

If this record is 4, the length of a scanline is 68 bytes (136 pixels).

If it is 255, the length of a scanline is 64 + 255 = 319 bytes, or 638 pixels.

If the offset value is such that scanning the screen memory to create the image reaches the end of a 16KB page, the scan will continue from the beginning of that same page.

Example: write 1 to that register, it means 65 bytes, enabled to read only 64 bytes every 65 to paint the screen, leaving 1 byte hidden/unused per line. That allows **horizontal scroll**.

u_zxi_043 - RADASPALBANK

Set which section of the ULAplus palette will be used to define the colours in Radastanian mode, and how the border will behave.

bit 3-7 = not used

bit 2 = BOR3

Indicates whether the border colour will be 0 to 7 (0) or 8 to 15 (1).

It can be considered as the bit 3 of the border colour in Radastan mode.

bit 0-1 = RADPALQUARTER

Set a section of the ULAplus palette to use for the radastanian mode.

00 to use the input section 0 to 15

01 for inputs 16 to 31

10 for inputs 32 to 47

11 for inputs 48 to 63

u_zxi_0FF - COREID

Each read operation provides the next ASCII character in the string containing this text "**eLeMeNt zx**" identification. When the chain ends, subsequent reads emit bytes with the value 0 (at least one of them is emitted) until the chain starts again. This pointer returns to 0 automatically after a reset. The delivered characters that are part of the string are printable standard ASCII (codes 32-127).

Other important ports

Some Speccy, MB02+ and divIDE/MMC/SD ports **are readable** where possible, e.g.:

```
   23    h17   MB02+ memory
  227    hE3   divIDE/divMMC - control
  231    hE7   divMMC - select device
  254    hFE   ULA    - readable on 40701  h9EFD
  255    hFF   Timex  - readable on 40957  h9FFD

 8189  h1FFD   +2A/+3 and Pentagon memory
32765  h7FFD   128/+2/+2A/+3 and 256K+ memory
57341  hDFFD   256K+ memory (Profi)
65021  hFDFD   256K+ memory (ATM)
```

Bits 3 and 4 of the port h1FFD are not implemented, they return the last written value.
Ports h2FFD and h3FFD return hFF.

With two exceptions stated above, the *DivIDE programming model* at
https://divide.cz/files/info/pgm_model.txt applies.

128K+ (16K banked) memory

128K memory

The shadow screen is in the bank 7. The normal bank 5 is always between h4000 and h7fff.

32765 **h7FFD**

bits 0-2 = RAM page (0-7) at hC000
bit 3 = normal (0) or shadow (1) screen to be displayed
bit 4 = 0 for the 128K ROM; 1 for the 48K BASIC
bit 5 = if set, memory paging is disabled (until computer reset)

+2A/B/+3 memory

8189 **h1FFD**

bit 0 = paging: 0 = normal, 1 = all-ram
bit 1 = in normal mode: ignored; all-ram: low bit
bit 2 = in normal mode: high bit of ROM selection; all-ram: high bit

ROM 0	128K editor, system, self-test
ROM 1	128K syntax checker
ROM 2	+3DOS
ROM 3	48K BASIC

bits 3-4 = not implemented (originally: FDD and printer)

All-ram: bits 2 and 1 of h1FFD set four RAM banks at 0, h4000, h8000 and hC000.

00	0 - 1 - 2 - 3	10	4 - 5 - 6 - 3
01	4 - 5 - 6 - 7	11	4 - 7 - 6 - 3

Note: If the system is in +2A/B/+3 all-ram mode, esxDOS ROM mapping is disabled.

Didaktik Gama 80KB

This model contains an extra memory, which is divided into two pages of 32 KB, from address 32768.
The usual way of paging is using the OUT 127,x command, where x is either 0 or 1.

Pentagon 1024 v2.2 memory

32765 **h7FFD**
bit 5 = 6. bit of RAM paging for 1024K (bit 2 of hEFF7 must be 0)
if bit 2 of port hEFF7 is set then this bit is used for a paging disable!
bit 6-7 = 4. and 5. bit for 512K

61431 **hEFF7**
bit 2 = 1 for 128K, 0 enables 1MB memory
bit 3 = 1 disable ROM and connect RAM page 0 at h0000-h3FFF

Pentagon 4096 memory

This memory expansion was mainly taken from https://github.com/rdacomp/Pentagon-4096. It offers several popular enhanced ZX128+ memory paging systems used in east-european clones, from 256K up to 2048K. And more, 4096K, with the 2 MB AA-SRAM memory expansion pack.

Paging system	**2 MB eLeMeNt ZX RAM**	**2 MB eL. + 2 MB AA-SRAM**
Scorpion	256K	512K
Profi, ATM1, Kay, Phoenix	512K	1024K

Combined paging systems		
Pentagon +Scorpion/Profi/ATM1/Kay	2048K	4096K

8189 **h1FFD**
bit 0 = 1 disable ROM and connect RAM page 0 at h0000-h3FFF
bit 4 = 4. bit for 256K **Scorpion** 256K
bit 7 = 5. bit for 512K **Kay** 512K

32765 **h7FFD**
bit 6-7 = 6. and 7. bit **Phoenix** 1024K, **Kay** 1024K

57341 **hDFFD** **Profi**
bit 0-1 = 4. and 5. bit for 512K
bit 2 = 6. bit for 1024K *AA-SRAM only*

65021 **hFDFD** **ATM**
bit 0-1 = 4. and 5. bit for 512K
bit 2 = 6. bit for 1024K *AA-SRAM only*

Combined memory variants

Pentagon + Kay /Scorpion	h7FFD bits 6-7	512K
	h1FFD bits 4 and 7	2048K
	h7FFD bit 5	4096K
Pentagon + Profi	h7FFD bits 6-7	512K
	hDFFD bits 0-2	4096K
Pentagon + ATM1	h7FFD bits 6-7	512K
	hFDFD bits 0-2	4096K

Extra memory

It is suggested to use the Ultimate Protect mode via **g_zxi_010**. Before using the Ultimate RAM on the eLeMeNt ZX computer, make sure you are using the internal, not the external MB03+ memory (see **g_zxi_000**). Check a number of free (available) Ultimate RAM pages with **g_zxi_011** and **g_zxi_012** registers.

The memory paging mechanism of the eLeMeNt ZX computer works with **four 16KB Zones** (4 parts of the Z80-CPU direct accessible memory from h0000 to hFFFF). Every Zone is split into **two** paging **8KB Areas**. This makes 8 selectable Areas (or 4 pairs of Area 0 + Area 1). Every Area you can assign to any 8 KB page from the Ultimate RAM space (up to 512 MB, currently 4 MB internal Ultimate RAM with the ExpBoard AA-RAM or 2MB on the basic eLeMeNt ZX).

Switching through the port **h73** redirects to the given Zone. All what is written to registers h07, h27, h47, h67 and h53 then, applies to this new Zone.

Zone0	h0000 - h3FFF	**Area0**	h0000 - h1FFF
		Area1	h2000 - h3FFF
Zone1	h4000 - h7FFF	**Area0**	h4000 - h5FFF
		Area1	h6000 - h7FFF
Zone2	h8000 - hBFFF	**Area0**	h8000 - h9FFF
		Area1	hA000 - hBFFF
Zone3	hC000 - hFFFF	**Area0**	hC000 - hDFFF
		Area1	hE000 - hFFFF

Fast 8-bits paging ports

```
007  h07  low  byte of Page in Area0 h0000-h1FFF
039  h27  high byte of Page in Area0 h0000-h1FFF
071  h47  low  byte of page in Area1 h2000-h3FFF
103  h67  high byte of page in Area1 h2000-h3FFF
083  h53  memory parameters
115  h73  memory Zone (00-03)
```

Memory parameters (port h53)

```
bit0  = enable write Area0
bit1  = enable write Area1
bit2  = enable one instruction delay ports 07,h27,h47,h67 - not ready
bit3  = enable MAPPER - not ready
bit4-7 = memory type layout
```

Memory type layout (bits 4-7)

0 - disable
1 - reserved
2 - **RAM** Area0 & **RAM** Area1
3 - reserved
4 - RAM Area0 & MAPPER config memory in Area1 - zone 0 only *- not ready*
5 - reserved
6 - **FLASH** Area0 & **FLASH** Area1 - zone 0 only
7 - reserved
8 - **HGFX** Area0 & **HGFX** Area1 (Chunky mode) - respect OffsetY and PlanarMask
9 - **RAM** Area0 & **HGFX** Area1 (Chunky mode) - respect OffsetY and PlanarMask
A - **LCU** Area0 & **RAM** in Area1 - zone 0 only
B - reserved
C - **RAM** Area0 & **LCU** in Area1 - zone 0 only
D - reserved
E - **RAM** Area0 & **FRAM** setup memory in Area1 - zone 0 only
F - reserved

Note: you can work with special RAM types (LCU, Flash) in the Zone 0 only.

Memory type layout in Zones 1, 2, 3

Ultimate RAM and HGFX chunky memory are pageable in all Zones and Areas.

Use memory type layouts **(**bits 0 - 4 of the port h53) of numbers **0, 2, 8** and **9** in the Zones 1 - 3.

Ultimate RAM and other memory types

HGFX buffers, available in the eLeMeNt paging mechanism as **chunky** areas, do not share any space with the abovementioned types of RAMs. They use their own FPGA memory, named **BRAM**.

Ultimate protect mode (**g_zxi_010**) ensures the divSD 512 KB RAM, MB02+ 512KB RAM and all ZX models (128KB, 256KB, 512KB, 1024KB, 2048KB) are not included into the Ultimate RAM Zones and Areas paging.

512 KB DivSD RAM is shared with 512KB MB02+ RAM.

Note on the full ZX 4096KB (Pentagon-4096) memory: it is not recommended to use the divSD RAM with the Pentagon 4096K because there is not any free Ultimate RAM left and therefore no Protect mode.

Here is an example of Ultimate Protected mode with a Pentagon with 128K RAM (8 x 16K pages) and divSD (512KB) attached:

- set g_zxi_010 to 02 (URAM)
- set e_zxi_000 to 05 (P128)
- set e_zxi_001 to 01 (divSD)
- read g_zxi_011 and g_zxi_012 to check a number of free Ultimate RAM pages

1408 KB	128 KB	512 KB
Ultimate RAM (Protect mode)	ZXRAM	divRAM 00 … 31

Flash memory (ROM Areas)

Default size is 512KB (alternatively it is possible use smaller 256KB or 128KB chip).
512KB is divided into 4 ROM Areas, selectable by **e_zxi_00F** (values 0 to 4)**.**
Note: Don't use ROM Area 0 for your personal roms to avoid reflash an area by wrong data!

Default ROM Area organisation per 16KB blocks

0 ROM0	ROM0
1 ROM1	ROM1
2 ROM2	ROM2
3 ROM3	ROM3
4 esxDOS	esxDOS
5	*reserved*
6 Rescue	Rescue system
7 SetUp	eLeMeNt SetUp

Switching ROM Areas on the eLeMeNt v1

On the eLeMeNt ZX board v1, ROM Areas can be switched by DIP 7 and DIP 8, on the back side of the board. 0 means UP and 1 DOWN.

```
DIP8  DIP7
 0     0    - backup of ZX128/+2 roms
 0     1    - backup of ZX128+2A roms
 1     0    - backup of ZX48 roms
 1     1    - free (SE Basic, DiagRom, etc.)
```

FRAM

8KB FRAM can be used for system internal purposes as BIOS/Setup (holds setting of hw features that are loaded on startup) or system counter (statistics of use).

h2000-h23FF	SetUp/Bios area
h2400-h27FF	User Area 01 - LNX
h2800-h3BFF	reserved
h3C00-h3FFF	System counter

I/O port-based LIFO stack

intended for new esxDOS syscalls and others, e.g. NMI menu

43	**h2B**	circular 256 bytes buffer

Audio expansions

MonsterBlaster

SounDrive, COVOX and CS/CZ DA Convertor compatible

```
15  h0F   Soundrive LeftA, Covox Left
31  h1F   Czech A, Soundrive LeftB
63  h3F   Czech B
95  h5F   Czech C, Soundrive RightD
79  h4F   Soundrive RightC, Covox Right
```

all ports are write only

AY Soundchips/TurboSound FM

65533 **hFFFD**
write: select an AY/YM register 0-14
read the value of the selected register

49149 **hBFFD**
write to the selected register

Second AY/YM

`OUT 65533,254`
to switch to the second AY chip (all the AY registers will be redirected on the second AY chip)

`OUT 65533,255`
to get back to the first AY chip

ZXi mirrored ports for TSFM

32315 **h7E3B**
32571 **h7F3B**

SAA1099

ZXi assigned ports

```
28219  h6E3B    data
28475  h6F3B    register
```

SAM Coupe compatible, write-only ports

```
255  h0FF     data
511  h1FF     register
```

Sound interface device (SID)

ZXi assigned ports

```
27707  h6C3B    select SID Control Register
27963  h6D3B    read/write selected SID Control Register
```

SID registers

h00-h1C = original SID registers
h1F = SID type and frequency
h20 = more SIDs

SID parameters

The eLeMeNt ZX and MB03+ use FPGA implementation of SID that can switch type with related filters.

```
h1F = select SID parameters
      bit 0 = SID type: 0-6581; 1-8580
      bit 1 = frequency:
            0-PAL  1-NTSC (985248Hz for PAL C64, 1022730Hz for NTSC C64)
      bits 2-7 = always 0
```

More SIDs

```
h20 = multiple SIDs (optional)
```

Note: It is not needed to use this option, currently only one SID is implemented.

General Sound (GS)

Note: The GS is available only in the core for the **AU+ board,** with a bigger Artix FPGA chip.

The eLeMeNt ZX version has got a 20 MHz "virtual Z80" CPU, 16KB ROM, 2MB RAM and four 8bit sound channels. For details, see the General Sound Programming Manual by STINGER & CYRAX, translated by iliks/Hugi.

HGFX

HGFX registers

All HGFX registers area is set via the g_zxi_022 (high byte only).

0 - Video parameters

```
bit0 = 0 - display video buffer 0; 1 - display video buffer 1
bit1 = destination video buffer for writing for all PlanarModes
bit2 = source video buffer for reading in COPY mode and CHUNKY access
```

1 - Planar Mode

```
0 - INK
1 - COPY from secondary to primary Offset
2 - ATTR (HGFX v2.1)
```

2 - Advanced commands

```
0 - NOP
1 - COPY_ALL - source to destination buffer - PlanarMask respected
2 - ERASE_ALL - destination buffer - PlanarMask respected
```

3 - IndexColour

Set index colour for graphics operation

4 - PlanarMask

Set bitplanes mask for graphics operation. 8 bits mask for index colour

Example:

```
PlanarMode:     INK
CurrentColour: 0 0 1 1 0 0 1 1  - Current Pixel Colour
PlanarMask:    0 0 0 0 1 1 1 1  - PlanarMask for IndexColour
IndexColour:   1 1 1 1 1 1 1 0  - IndexColour
-----------------------------
NEW Colour:    0 0 1 1 1 1 1 0    no change  changed
BYTE :         0 0 0 0 1 0 1 1  - byte written to videoRAM
PIXEL:         x x x x 1 x 1 1  - x: nochange; 1: set NEW Colour
```

5 to 6 - OffsetX primary - write

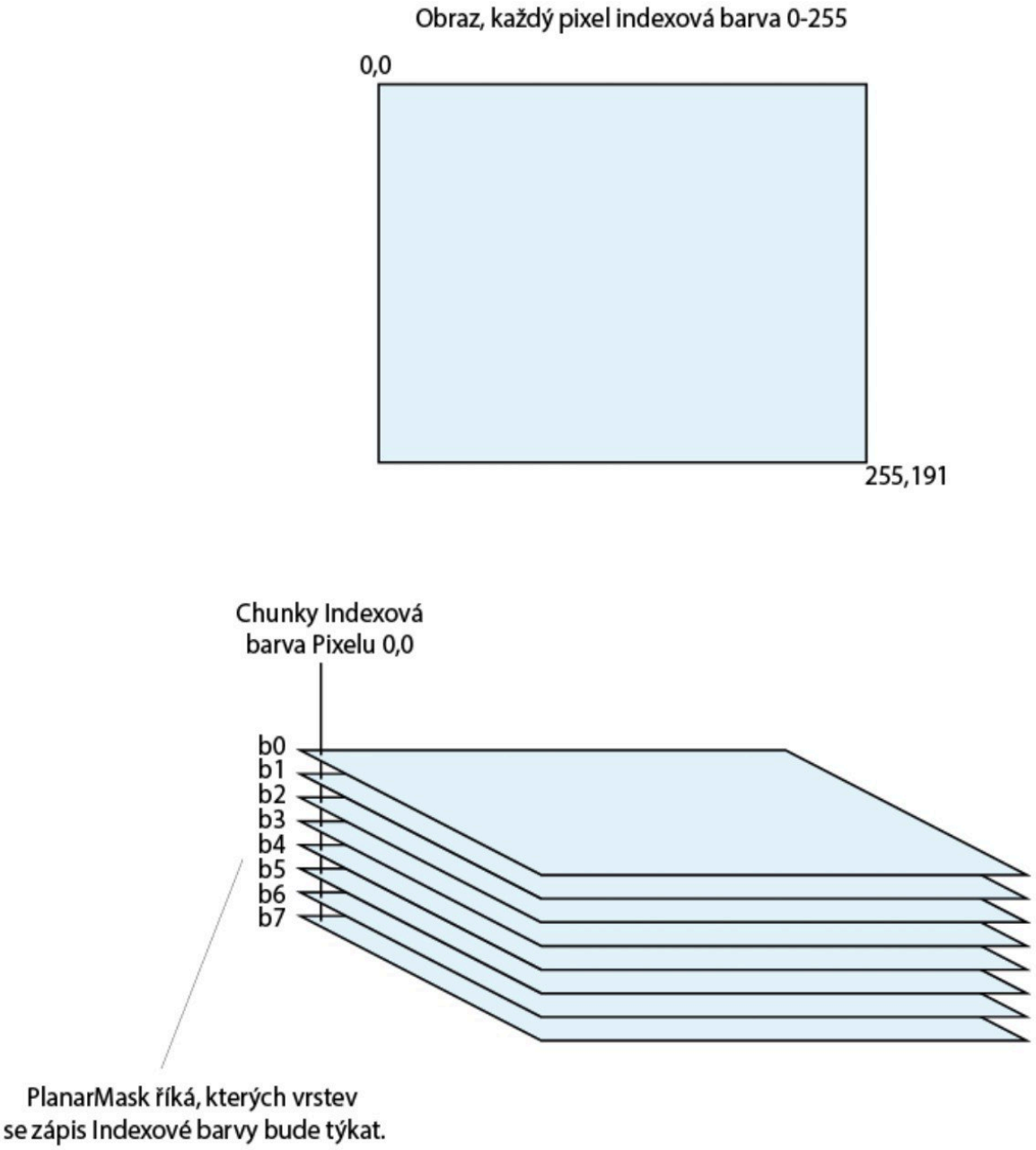
7 to 8 - OffsetY primary - write

9 to 0Ah - OffsetX secondary - read

0Bh to 0Ch - OffsetY secondary - read

All **Offsets** take **two bytes** and start on position 0,0 like ZX, but are signed. Due to this feature it is possible write outside in a hidden position and show only part of the data. e.g. position -1,-1 (OffsetX=65535,OffsetY=65535) show data shifted left and up by one pixel).

Set bitplanes mask for graphics operation. 8 bits mask for index colour:



0Dh - AdvancedGraphicsFeature

```
0 - NOP - disable
1 - FILL   (HGFX v2.0)
2 - HAM    (HGFX v2.0)
```

0Eh - SelPaperIndex *(HGFX v2.0)*

0Fh - FillPlannarMask *(HGFX v2.0)*

HGFX in chunky format

HGFX memory registers basically provide planar-oriented access and manipulations with graphics data. To handle a raw (graphics) memory of both HGFX videobuffers, use fast 8bit ultimate extra memory ports (see **memory parameters port 53h**).

Access to this “chunky” memory, when activated, reflects:

- OffsetY
- PlanarMask
- bit1 and bit2 of the HGFX register no. 0 (video parameters)

HGFX Colour table layout

low bytes :

h02 - red	h01 - green	h00 - blue	RGB values of indexed colour h00
h05 - red	h04 - green	h03 - blue	RGB values of indexed colour h01
..	..	..	..

HRC & HRXC graphic modes

HiResColour (HRC)

- 512x192 pixels and 64x48 attributes
- classic ZX Spectrum attributes format
- linear memory layout
- two videorams (ZX128 like buffer switching, per page 7)
- Keylayer and Gigascreen supported

Memory layout:

12 KB	pixels	h4000-h6FFF (hC000-hEFFF)
3 KB	attributes	h7000-h7BFF (hF000-hFBFF)

HiResindeXedColour (HRXC)

- 512x192 pixels and 64x48 attributes
- indexed colours attributes (16 unique INK & 16 unique PAPER colours in Index table)
- linear memory layout
- two videorams (ZX128 like buffer switching, per page 7)
- Keylayer supported

Memory layout:

12 KB	pixels	h4000-h6FFF (hC000-hEFFF)
3 KB	attributes	h7000-h7BFF (hF000-hFBFF)
96 B	Indexed Colour Table	h7C00-h7C5F (hFC00-hFC5F)

Because HRXC colours do not have bright and flash attributes, a PAPER colour may (or may not) be a different form an INK colour in the same attribute. Every colour from 32 colours (16 for INK and 16 for PAPER) in **Indexed Colour Table** is defined in TrueColour palette, by 3 bytes, in **B-G-R** format.

Notes:
A colour value holds a 24-bit (3 bytes) number, written in this way: The lower byte is stored first.
It is similar to how the Z80 CPU stores 16-bit numbers: `h8000 LD BC, h4000`
is stored as:

```
h8001 - h00
h8002 - h40
```

Similarly, a classic RGB value is in memory stored in a B-G-R format:

```
hC000 - hDC - Blue
hC001 - h00 - Green
hC002 - hFF - Red
```

KeyLayer Mode

The 3rd bit of the paging port `h7ffd`, which displays the main or shadow video-page on the ZX Spectrum 128k, determines which screen is a transparent one, "on top" and carries the KeyColour. The second screen, all pixels and colour information in attributes, are positioned below and being displayed through a KeyColour then.

KeyLayer in ZX and HRC modes

One colour from a group of 15 basic ZX colours can be marked as a transparent **Key colour** (KC, see the register **g_zxi_004** - Layer mode - additional parameters). Both screens use the ZX Spectrum colour palette with 8 brighted and non-brighted colors. Black colours in positions 0 and 8 are not distinguished by bright, so if KC = 0, the color no. 8 also becomes a transparent color (and vice versa).

The flash-attribute is not much usable and is not supported in the KeyLayer mode.

KeyLayer in HRXC mode

One colour from 32 unique colours can be marked as a transparent **Key colour** (KC, see the register **g_zxi_004** - Layer mode - additional parameters). The KeyColour is determined by a colour index, not by a colour content. Every screen uses its own colour palette. 16 colours for INK and next 16 colours for PAPER. Because HRXC colours do not have bright and flash attributes, a PAPER colour may (or may not) be a different from an INK colour in the same attribute.

Changing a color palette per line

The HRXC palette is always updated at the beginning of the frame, so changing it during the display process has no effect. The KeyColour can be changed only.

On the contrary, in the HRC mode you can't change the palette but you can change a KeyColour per (micro)line.

Advanced 128K videoRAM

Through **`g_zxi_002`** you can activate the first video buffer in the RAM page no. 6 instead of the original RAM page no. 5. First, you need to set this ZXi port in order to make and display screen from the page 6. After that, you can start to write data to this page (not before!) and such a data will be displayed then.

ULAplus

ULAplus™ is controlled by two ports, the register port and the data port.

Register port

Port **hBF3B** is the register port (write only).
The byte output is interpreted as follows:

```
bits 0-5 = Select the register sub-group
bits 6-7 = Select the register group
```

Groups

Following groups are available:

```
00   Palette group
01   Mode
10   Extra (HRC and HRXC)
```

Palette Group

When this group is selected, the sub-group determines the entry in the palette table (0-63).

Mode Group

The sub-group is (optionally) used to mirror the video functionality of Timex port FF and the framebuffer select of port 7FFD as follows:

```
bits 0-2 = Screen Mode

     000  screen 0 (bank 5)          100  screen 0 (bank 7)
     001  screen 1 (bank 5)          101  screen 1 (bank 7)
     010  hi-colour (bank 5)         011  hi-colour (bank 7)
     110  hi-res (bank 5)            111  hi-res (bank 7)

bits 3-5 = Screen Colour in Hi-Res Mode (foreground/background)

     000  0 / 7                      100  4 / 3
     001  1 / 6                      101  5 / 2
     010  2 / 5                      110  6 / 1
     011  3 / 4                      111  7 / 0
```

Extra Mode Group

```
10   This is a proposal. Not approved yet.
bits 0-5 = Screen Mode (the same as the g_zxi_001 - Extra graphics modes)
     00   none - disable
     01   HiResColour (HRC)
     02   HiResindeXedColour
```

Data port

Port **hFF3B** is the data port (read/write).
When the palette group is selected, a written byte will describe the colour.
When the mode group is selected, a byte output will be interpreted as follows:

```
bit 0: ULAplus™ palette on (1) or off (0)
```

Reading from port hFF3B returns the last data byte written to the currently selected register. This can be used to read back the current palette or determine if palette mode is active. Implementations that support the Timex video modes use port **hFF** as the primary means to set the video mode, as per the Timex machines. It is left to the individual implementations to determine if reading the port returns the previous write or the floating bus.

Note: Modes 2 to 255 are reserved for future use. Groups 10 and 11 are reserved for future use.

Timex Graphics

```
255   hFF
   bits 0-2 = Mode
        000  ZX Spectrum classic screen in area 0
        001  ZX Spectrum classic screen in area 1 (2nd videoram)
        010  HiColour
        110  HiRes

   bits 3-5 = PAPER & INK for HiRes
        000  Black on white
        001  Blue on yellow
        010  Red on cyan
        011  Magenta on green
        100  Green on magenta
        101  Cyan on Red
        110  Yellow on blue
        111  White on black

   bit 6     unused (originally - disable timer interrupt)
   bit 7     unused (originally - Timex MMU bank select)
```

HiColour mode

256x192 pixels, with an attribute per each 8x1 pixels block
Pixels located at h4000 (area 0), attributes at h6000 (area 1). Both areas take h1800 bytes.

HiRes mode

512x192 pixels, monochrome (one INK and one PAPER colour with active BRIGHT for the whole screen)
HiRes also uses areas h4000 and h6000, alternating each one per each 8pixels1char-column.

In addition to the original Timex screen modes, two separate video areas of the ZX Spectrum 128K can be accessed. This is done by using bit 3 of the port 0x7ffd. This gives the ***ZX Spectrum SE*** videosystem with a total of 27K of RAM which can be used for up to four standard screen areas or two HiRes or HiColour screens, from h4000 and h6000 or from hC000 or hE000.

Kempston Mouse

Default (master) mouse

```
64479  hFBDF    X-axis (horizontal) position
65503  hFFDF    Y-axis (vertical) position
64223  hFADF    buttons
     bit 0 =    zero if right button pressed
     bit 1 =    zero if left button pressed
     bit 2 =    zero if middle button pressed
     bit 3 =    unused (originally: 4th button)
     bits 4-7 = mouse wheel position (b1111 defaults)
```

This port returns 255 if the mouse is stationary and no buttons are pressed.

```
65247  hFEDF    detection, h80 if active
```

Second (slave) mouse

```
15327  h3BDF    X-axis (horizontal) position
16351  h3FDF    Y-axis (vertical) position
15071  h3ADF    buttons
     bit 0 =    zero if right button pressed
     bit 1 =    zero if left button pressed
     bit 2 =    zero if middle button pressed
     bit 3 =    unused (originally: 4th button)
     bits 4-7 = mouse wheel position (b1111 defaults)
16095  h3EDF    detection, h80 if active
```

In order to operate a K-Mouse implemented in the eLeMeNt ZX, an USB mouse has to support a PS/2 protocol. The list of the compatible mice is in the file “eLeMeNt ZX peripherals - notes” in https://sites.google.com/view/elementzx/docs

Kempston joystick

```
31  h1F

     bit 0 = right              bit 4 = button1
     bit 1 = left               bit 5 = button2
     bit 2 = down               bit 6 = Sega Controller
     bit 3 = up                 bit 7 = is always 0
```

External PC Keyboard

A PC keyboard with USB socket and compatible with PS/2 standard or a keyboard with a PS/2 socket and a PS/2 to USB adapter can be attached to the eLeMeNt's upper USB port.

The **ZX Spectrum keyboard** is mapped as follows:

Caps Shift	..…..……..	Shift	**Up**	*CS+7* ……..	Up
Symbol Shift	……..	Ctrl	**Right**	*CS+8* ……..	Right
Extend mode	*CS+SS* ……	Tab	**Graph**	*CS+9* …….	Insert
Edit	*CS+1* ……..	accent (`)	**Delete**	*CS+0* ……..	BackSpace
Caps Lock	*CS+2* ……..	CapsLock		*CS+SS+0* …..	Delete
TrueVideo	*CS+3* ……..	PageUp	**Break**	*CS+Space* ….	Escape
InvVideo	*CS+4* ……..	PageDown	**<=**	*SS+q* ……..	Home
Left	*CS+5* ……..	Left	**>=**	*SS+e* ……..	End
Down	*CS+6* ……..	Down	**:** (colon)	*SS+z* ……..…	\ (backslash)

Additional control keys are:

Win + F1	CPU 3.5 MHz
Win + F2	CPU 7 MHz
Win + F3	CPU 14 MHz
Win + F4	CPU 20 MHz
Win + F5	CPU 22 MHz (overclock)
Win + F6	CPU 28 MHz (overclock, 1T-state delay in ROM area)
Win + F7	CPU 28 MHz (overclock, for RAM based SW only)
Win + F8	CPU 30 MHz (overclock, for RAM based SW nly)
Win + F11	EXTRA button
Win + F12	NMI button
Win + d	DVI mode
Win + g	Gigascreen mode
Win + p *or* ***Pause***	pause
Win + Alt + d	diagnostic and debug info
Win + Alt + b	border ZXS diagnostic under the screen
Win + Alt + F1	ZX Spectrum 48K
Win + Alt + F2	ZX Spectrum 128K / +2
Win + Alt + F3	ZX Spectrum +2A
Win + Alt + F4	Pentagon 128
Win + Alt + F5	Pentagon 512
Win + Alt + F6	Pentagon 1024 v2.2
Win + Alt + F7	Pentagon 4096
Ctrl + Alt + Delete	reset
Ctrl + Alt + Space	reset - useful for esxdos reboot, press ***Ctrl + Alt + Space*** keep ***Space*** pressed

For more info about OverClock see the chapter ***e_zxi_031*** - *CPU OverClock mode*. The list of the compatible keyboards is in the file "eLeMeNt ZX peripherals - notes" in https://sites.google.com/view/elementzx/docs

Scancodes generated by an external keyboard are also directly readable at **u_zxi_004 - SCANCODE** and **u_zxi_005 - KEYSTAT** ports.

LCU - LMN Coprocessor Unit

This is a versatile calculator, a hardware multiplier and divider was designed by Busy soft.
The implemented mathematical expression is: `Z = ( A * B + C ) / D`

LCU memory map:

`h2000 to h201F (h0000 to h001F)`

Input members - set by the program

`h2000 [4]` ... member A (to be added)
`h2004 [4]` ... member B (to be added)
`h2008 [4]` ... member C (to be added to the multiplication result)
`h200C [4]` ... member D is a divisor

Intermediate result of multiplication - set by the program or by the FPGA

`h2010 [8]` ... a dividend = intermediate result A*B+C as a member for division

Division results - set by the FPGA

`h2018 [4]` ... division result - quotient
`h201C [4]` ... division result - remainder

Extended command

`h2020 [1]` … h00 - reset all input to defaults

In brackets [ ] is the width of the data (in bytes). In order to compute the whole mathematical expression it is necessary to fill all members A, B, C and D and the result will appear in the next 16 bytes.

Other calculation options are:

Addition

set A=1
set members B and C
output: result of addition of B and C numbers

Multiplication

set members A and B
set C=0
output: result of multiplication of A and B numbers

Division

set a dividend into the cells of the intermediate result
set D=divisor
output: division results in quotient and reminder

How to process signed numbers?

For multiplications and divisions, all negative values must first be converted to positive (this is the negation of all bits + the increment) and the sign of the result is then the XOR of the signs of the operands.

When calculating X+C (where X is the result of A*B), it is first necessary to determine whether the absolute value of the negative C is greater than X, and if not, C does not need to be converted. If it is greater, the result of A*B+ will be negative and must be converted to positive and after then the division must be

performed. It's a lot of testing, negations, increments, but they are all trivial operations at the level of addition/subtraction, which are quickly done.

FastDMA

See `g_zxi_040` and `g_zxi_041`.

FastDMA registers

All FastDMA registers area is set via the g_zxi_040 (high byte only).

0 - FastDMA - Parameters (byte)

```
bit0 = source address 0-keep/1-change value after transfer
bit1 = destination addr. 0-keep/1-change value after transfer
bit2 = transfer type: 0-LDIR/1-LDDR
bit3 = floating source skip: 0-disable/1-enable
bit4 = floating destination skip: 0-disable/1-enable
bit5-7 = reserved, must be 0 (eg. select memory/port)
```

Write to this register causes immediate transfer!

1 - FastDMA - Block length of transferred data (word)

```
h0000 - default (means 65536bytes)
```

3 - FastDMA - Source - address (word)

```
h00 - default
```

5 - FastDMA - Source - batch length (word)

```
h0000 - default (means 65536bytes)
```

7 - FastDMA - Source - skip length (word)

```
h0000 - default
```

9 - FastDMA - Destination - address (word)

```
h00 - default
```

B - FastDMA - Destination - batch length (word)

```
h0000 - default (means 65536bytes)
```

D - FastDMA - Destination - skip length (word)

```
h0000 - default
```

SaveINT

DMA transfers are complemented by the SaveINT feature. If an INT interrupt occurs during a DMA transfer, it will not be discarded. The INT is executed immediately after the end of the DMA transfer.

SoundDMA

Main characteristics:

- 6 audio channels #A - #F
- each channel can play 8bits mono or stereo sound
- stereo VOLUME and PAN have 256 levels
- channel #A can play 16bits mono sound
- normal 16bits or extended 24bits period for fine-tunning
- unsigned and signed sample support
- play and loop mode
- wavetable synth ready (variable loop lenght)
- streaming ready (from SD)
- highest data-transfers priority, above the FastDMA and the classic Z80-DMA
- amiga frequencies compatibility

SoundDMA registers:

All SoundDMA registers are set via the g_zxi_049 (high byte only).

00 - SoundDMA Channel #A - Parameters (byte)

bit0 = 0-stop(pause)/1-play
bit1 = 0-play from start/1-continue
bit2 = 0-play only one/1-loop
bit3 = 0-unsigned/1-signed sample format
bit4 = 0-normal /1-extended period
bit5 = not used, must be 0
bit6 = 0-8bits /1-16bits:channel #A mono only, max volume
bit7 = 0-mono/1-stereo
Writing this reg starts transfer

01 - SoundDMA Channel #A - Volume Left (byte)

bit7-0 = Volume Left *max recommended value is 64. it's related to 0dB*

02 - SoundDMA Channel #A - Volume Right (byte)

bit7-0 = Volume Right *max recommended value is 64. it's related to 0dB*

03 - SoundDMA Channel #A - Start address (3-bytes)

works with protected URAM only

06 - SoundDMA Channel #A - Loop address (3-bytes)

09 - SoundDMA Channel #A - Length (3-bytes)

0C - SoundDMA Channel #A - Period (2 or 3-bytes)

Channels B to F follow… e.g:

10 - SoundDMA Channel #B - Parameters (byte)
50 - SoundDMA Channel #F - Parameters (byte)

Sample rates

normal period:

$$\text{SampleRate} = \frac{\sim 7.1\text{MHz}}{\text{Period(16bits)}}$$

extended period:

$$\text{SampleRate} = \frac{\sim 113\text{MHz}}{\text{Period(24bits)}}$$

There are _NO_ standard sample rate for the samples used in modules. But most often the samples are done at the rate called C-3. Sometimes drums are sampled at A-3 (around 28 kHz), and some sounds are at ~8 kHz or anything else to save space.

The sample rate on each of the channels can be selected by a period value, which tells the hardware how many ~3.5 MHz clocks to count down before playing the next sample. If you have a 16 kHz sample you simply play it at a note that gives you a 16 kHz sample rate. If you play it one octave lower, you get a 8 kHz sample rate (and a double period value).

So, the most normal rate is (C-3, period 214):

$$\frac{\sim 7.1\text{MHz}}{214 * 2} = 16588.78\ \text{Hz}$$

Annex - Reserved ports

these ports comes from MB02+ where are not used, but locked by non full decoder

035 **h23** b00100011 - *reserved*
067 **h43** b01000011 - *reserved*
099 **h63** b01100011 - *reserved*
115 **h73** b11100011 - Ultimate memory *reserved*
075 **h4B** b01001011 - FLASH FPGA Core *reserved*
107 **h6B** b01101011 - *reserved (currently used by Datagear, Next)*
055 **h37** b00110111 - *reserved (Next-Kempston Joystick 2, Joystick I/O)*
087 **h57** b01010111 - *reserved (Next-Sprite Attribute Upload)*
119 **h77** b01110111 - *reserved (currently used by ZController)*