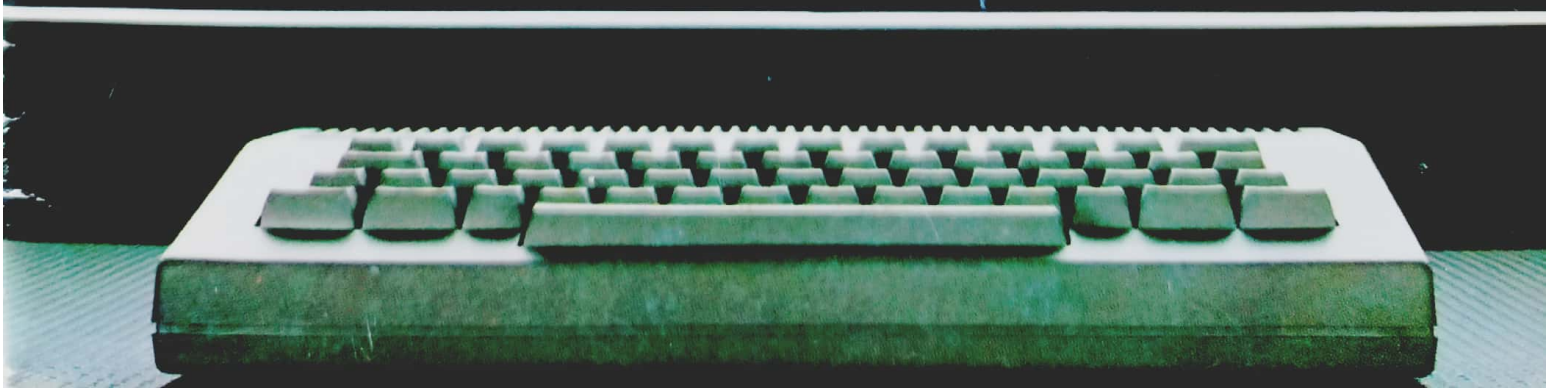
CAMPUTERS

LYNX

96K USER MANUAL



CONTENTS

Chapter 1: INTRODUCTION

The first in a planned series of expansions for the Lynx microcomputer is the 96K version. The addition of extra RAM means that usable workspace to Basic programs in high resolution colour is increased to 37.5K, plus a 23K data store which gives a total of 60.5K RAM available to the user.

Software to drive both serial and parallel printers is built in to the extension ROM supplied (although a serial printer will require a lead of the correct type and a parallel printer will require an interface pack).

All new and upgraded 96K Lynxes are now supplied with version 2.0 of the Basic ROM. Improvements to the interpreter are detailed in Chapter 3.

Chapter 2: THE 96K LYNX EXTENSION EPROM

This EPROM (which is the type 2732 device) is located in the third internal ROM socket on the board. It occupies memory from location **4000H** and is configured for the 96K Lynx with Version 2.0 Basic EPROMS (at positions **0000H** to **3FFFH** in memory). The presence of this EPROM is automatically tested for on power-up by checking location **4000H**. This extension EPROM cantains built-in serial and parallel printer routines, additional Basic commands and control of the Data Store.

PRINTER COMMANDS

The printer commands **LPRINT** and **LINK ON/LINK OFF** for which jumps already exist in Basic are implemented for both serial and parallel printers. On power-up the parallel printer mode is initialised. Serial or parallel may be explicitly selected at any time by choosing one of the following commands :

EXT PPRINT for **parallel** printers
EXT SPRINT for **serial** printers

The routines are already configured for the popular printers of each type.

Parallel	Epson series
Serial	Seikosha GP250X (at 2400 Baud)

A wide variety of other parallel printer makes and models can be accommodated by changing the default values of flag bytes **61BC** to **61CF**. See the Lynx Printer Technical manual for details of this.

LIGHT PEN

LIGHT PEN (Ø), LIGHT PEN (1).
Returns **X** and **Y** values respectively read from the CRTC light pen registers offset by values stored at **61D4H** in units of 2 pixels. Initialised value is **88H X,Y.** (**88H** where first digit is the variable **X** offset (**0** to **F**) and the second, the variable **Y** offset, both in units of 2 pixels.)

Additionally, the **X** register value is offset 24 pixels early to allow for CRTC offset and light pen response delay.

JOYSTICK

JOYSTK (0), JOYSTK (1).
Returns left **(0)** or right **(1)** joystick value corresponding to position, plus 256 when **FIRE** is depressed.

There are **48 DATA STORE** identities. Any combination of these may extend to fill the available capacity. An overflow will give an `'out of memory'` error message.

Data can be added at any time to any store. The directory system maintains pointers to the next data entry to be used. The pointer can be reset independently for each store.

DATA STORE COMMANDS

EXT STORE N, data,data, ...,etc.

This command stores the data to the specific store ("N") indicated. Numeric variables and expressions will be evaluated prior to storing. String constants must be enclosed by quotation marks " ". String variables will be stored as the contents of the string in HEX.

EXT 1 STORE N, data ,data, ..., etc.

This command allows integer values between 0 and 65535 to be stored in a more economical way, saving memory space. Numbers outside this range will be stored as Modulo 2^{16}. (See **STORAGE CAPACITY**, page 7).

EXT FETCH N, variable, variable, ..., etc.

This command loads the variable(s) with the store data from the specific store (`"N"`) indicated, moving the pointer along as appropriate. When the data is exhausted the error message `'Out of memory'` is displayed.

EXT BACK N

In a similar way to `RESTORE`, this command resets the store data pointer back to the start of where the data store indicated.

EXT WIPE N

This command erases all data from the store specified and frees the area for further use by any store.

EXT S SAVE N, "name"

This command saves the contents of the store specified by `"N"` directly to tape storage.

EXT S LOAD N, "name"

This command loads from tape to store specified by `"N"`, any data previously saved independent of the originating store identity. That is, data stored on tape loses its store identity

EXT CHAIN N(1), N(2)

This command links two designated stores together specifically for saving to tape in the sequence given as one continuous data block by `EXT C SAVE N(1)`.

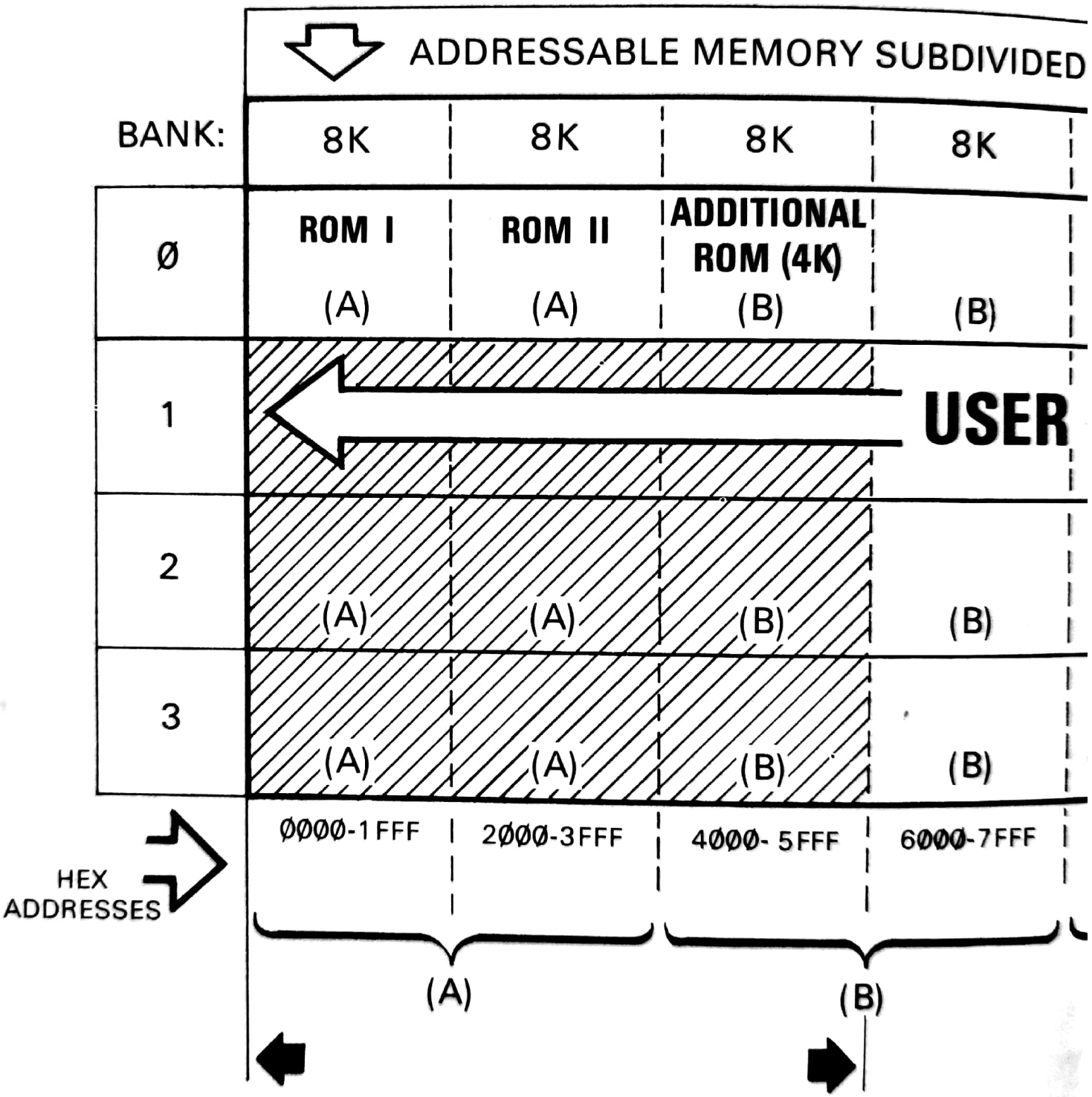
EXT C SAVE N, "name"

This command saves the contents of the chain sequence previously chained by `EXT CHAIN` starting from the indicated store onwards. To load contents use the command `EXT S LOAD N` as for any other saved data.

EXTENSION COMMANDS

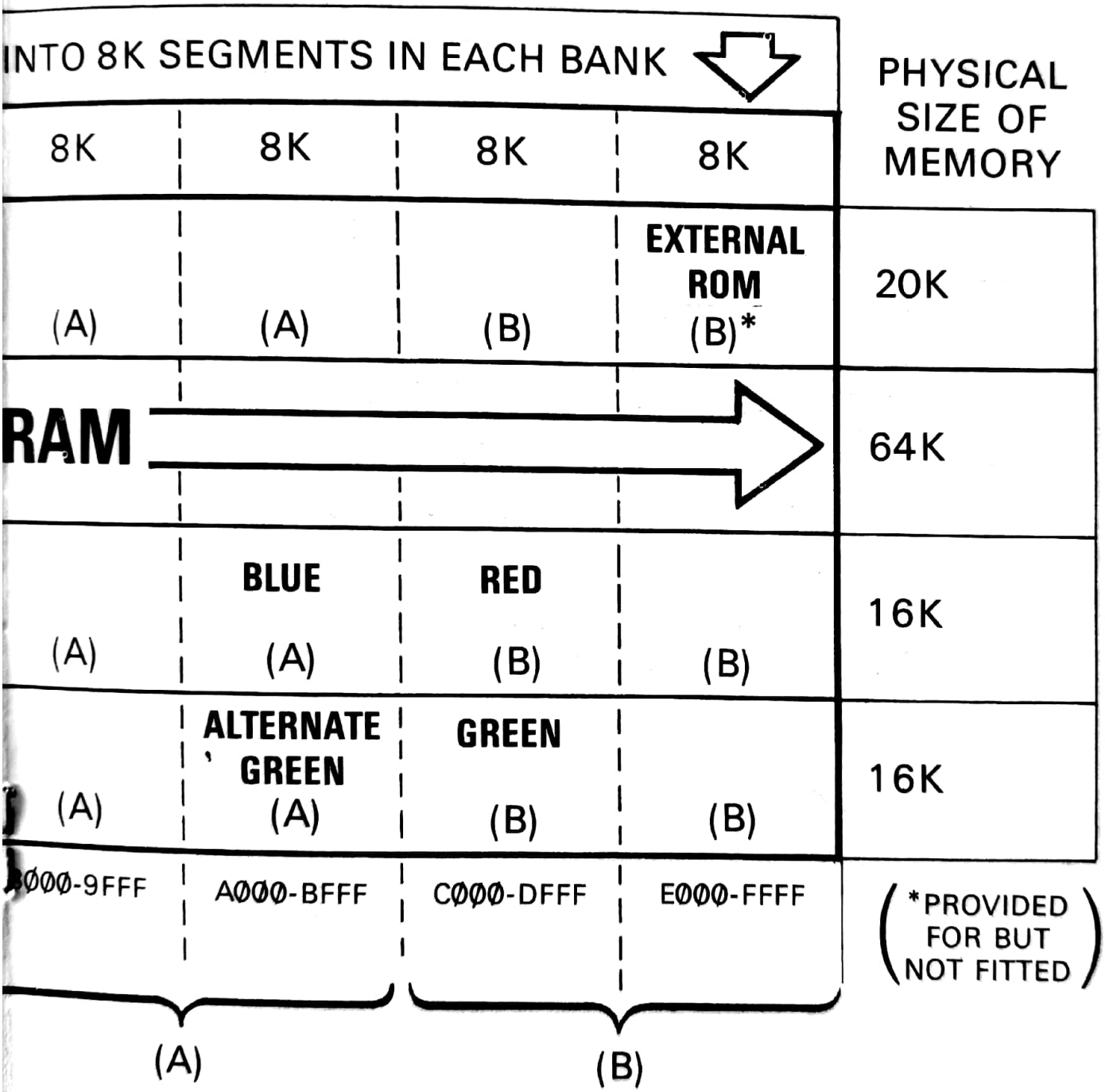
The following new commands are all executed from within the EPROM using a secondary table for commands prefixed by `"EXT"` (*[ESC]* + `"E"`). They can be abbreviated on entry to `EXT` followed by several letters and a full stop.

LYNX 96K



Shaded area (ØØØØ-5FFF) is masked out by ROM. The 24K of USER RAM thus "hidden" can be used via EXT commands provided.

memory map



Note: A and B (in brackets) show where addresses are reflected, that is:
(A) appears twice, at addresses 0000-3FFF and 8000-BFFF;
(B) appears twice, at addresses 4000-7FFF and C000-FFFF.

EXT CIRCLE M,X,Y,R

This command allows the user to draw a circle in current INK with the centre defined by coordinates **X,Y** and the radius by **R**. For a line circle set parameter M=0. For a filled circle set M=1. The maximum radius is 511.

NOTE: It is possible to execute a circle whose centre is offscreen as the screen wrap round function is automatically inhibited. Very small circles will appear square as the pixel map is on a square grid.

EXT CLW

This command will clear the window with the current paper colour and home the cursor to the top of the window.

NOTE: If you first redefine the window this command can be used as a rectangle fill.

EXT TRIANGLE X1, Y1, X2, Y2, X3, Y3

A filled triangle in the current ink colour results. The three **X, Y** pairs correspond to the triangle vertices (corners) which must be "on screen": **X**: 0 – 255 ; **Y**: 0 – 251

EXT TRAP, EXT NO TRAP

This command will set or unset the error trap facility. Use EXT TRAP Q to trap the error code to variable, say **Q**. The program will run on ignoring the error. It is therefore necessary to test **Q** at program lines which could cause an error. Error trap should be disabled by EXT NO TRAP except when the trap is required, since with the normal error default mode inhibited, program corruption is very likely.

EXT VRESET

This command resets the video to normal default power up values for CRTC, INK, PAPER, PROTECT and standard height characters. It also clears the screen.

This command is particularly useful to recover normal screen operation following *[ESC]* from the programs where the CRTC is being modified. Depending on the state of the screen, this command may have to be typed in "blind", i.e., without it being seen on the screen.

```
EXT LASER
EXT EXPLODE
EXT ZAP
EXT KLAXON
```

Are a range of preformated sound effects suitable for incorporating into games programs.

THE DATA STORE

The 96K Lynx is fitted with a full 64K of DRAM in the workspace area (Bank 1). Memory addressed from 6000H to FFFFH are available for Basic programs and variables, less the areas normally occupied by the operating system. This leaves approximately 37.5K.

The remaining area may be used as a **DATA STORE** for numeric and string data, less the space reserved for directory and system requirements. This leaves approximately 23K of **DATA STORE**.

Use of the **DATA STORE** is similar in many respects to DATA/READ/RESTORE available from Basic, except that the data is independent of the basic program and may be transported between programs by using **NEW** to remove the previous program. **EXTSSAVE** and **EXTSLOAD** can also be used to save and load data direct to/from cassette.

DATA STORE EXAMPLE

The following is an example of how to use the data stores using data generated by a procedure `'GEN'`. This data is then displayed by a second procedure `'DIS'` which may be in the same or another program.

```
100 CLS [RETURN]
110 PROC GEN [RETURN]
120 PROC DIS [RETURN]
```

Store contents are generated by procedure GEN.

```
1000 DEFPROC GEN [RETURN]
1010 EXT WIPE 1 [RETURN]
1020 PRINT "CALCULATING VALUES OF SINE WAVE" [RETURN]
1030 FOR X=0 TO 255 [RETURN]
1040 EXT ISTORE 1, -100*SIN(X/40)+120 [RETURN]
1050 NEXT X [RETURN]
1060 ENDPROC [RETURN]
```

Store contents are displayed by procedure `DIS`.

```
2000 DEFPROC DIS [RETURN]
2010 CLS [RETURN]
2020 EXT BACK 1 [RETURN]
2030 MOVE 0,120 [RETURN]
2040 FOR X=0 TO 255 [RETURN]
2050 EXT FETCH 1,Y [RETURN]
2060 DRAW X,Y [RETURN]
2070 NEXT X [RETURN]
2080 ENDPROC [RETURN]
```

STORAGE CAPACITY

	VALUE	*BYTES USED*
INTEGERS	0 – 15 16 – 4095 4096 – 65535	1 byte 2 bytes 3 bytes
FLOATING POINT		6 bytes
STRINGS		LEN+1 bytes

Chapter 3: CHANGES IN VERSION 2.0 BASIC ROM

Several improvements have been made to the Basic ROM supplied with all new LYNX 48K and 96K units:

NEW POWER-UP PROCEDURE

By testing whether `9FF8H` and `BFF8H` are distinct physical addresses, the Lynx automatically determines if it is the 48K or 96K version.

If it is the 48K version, the stack pointer is set to the value `9FF8H`. If the 96K version was found, a test is made to determine whether external ROM is present and the stack pointer is set to the highest available RAM location of the form `NFF8H`.

Tests are then carried out to determine whether internal ROM is present by reading from location `4000H`. If ROM is present then location `4000H` will give `0` (low); if no internal ROM is present then non-`0` (high) will be given.

If location `4000H` contains `0`, then `CALL &4000` is executed to initialise the internal ROM routines.

A power-up message is printed. No change has been made to the 48K version but an extra beep has been added to denote the 96K version.

DETECTION OF *[ESC]* DURING INPUT

The *[ESC]* key is now detected whenever pressed during the **INPUT** statement rather than at the start of the next line. To escape from an **INPUT** statement press *[ESC]* then press *[RETURN]* while still depressing the *[ESC]* key. Any characters typed before *[ESC]* will be ignored.

UNDERLINE CHARACTER

The underline character can now be implemented from the keyboard by pressing *[SHIFT]*`0` and will appear below the print line.

IMPROVEMENTS TO ROUTINES

The **Arccsin** and **Arccos** routines now use improved algorithms. A bug in the **Arctan** routine has now been corrected.

Subtraction of `0` now works correctly in all cases.

`LEFT$` and `MID$` now correctly return the empty string when the number of characters is `0`.

The monitor program now displays the **N** and **C** flags correctly.

The `DISK` command is now replaced by `XROM` which jumps to `E000H` to call an external ROM.
Do not use if no ROM fitted and selected on.