



## User's Manual

Simulators EPROM

# **SIMEPROM-01B**

Simulation EPROM 2716...27512, 120ns

# **SIMEPROM-02aLV**

Simulation EPROM 2716...27040, 85ns

ELNEC s.r.o.  
Presov, Slovakia  
July 2003



**COPYRIGHT © 1994 - 2002  
ELNEC s.r.o.**

This document is copyrighted by ELNEC s.r.o, Presov, Slovakia. All rights reserved. This document or any part of it may not be copied, reproduced or translated in any form or in any way without the prior written permission of ELNEC s.r.o.

The control program is copyright ELNEC s.r.o., Presov, Slovakia. The control program or any part of it may not be analyzed, disassembled or modified in any form, on any medium, for any purpose.

Information provided in this manual is intended to be accurate at the moment of release, but we continuously improve all our products. Please consult manual on **[www.elnec.com](http://www.elnec.com)**.

ELNEC s.r.o. assumes no responsibility for misuse of this manual.

ELNEC s.r.o. reserves the right to make changes or improvements to the product described in this manual at any time without notice. This manual contains names of companies, software products, etc., which may be trademarks of their respective owners. ELNEC s.r.o. respects those trademarks

ZLI-0013A

---

## **Table of contents**

<b>Introduction</b> .....	<b>5</b>
<b>Hardware</b> .....	<b>9</b>
SIMEPROM-01B hardware installation .....	10
SIMEPROM-02aLV hardware installation .....	12
<b>Software</b> .....	<b>17</b>
The simulator software .....	18
Using the SIM012 control program .....	20
Using the SIM012W control program .....	48
Using the SIM01BR control program .....	61
Using the SIM02R control program .....	64
<b>Specification</b> .....	<b>69</b>
SIMEPROM-01B specifications .....	70
SIMEPROM-02aLV specifications .....	72
<b>Warranty</b> .....	<b>75</b>
<b>Appendix</b> .....	<b>77</b>
Appendix A - reset circuit wiring .....	78
Appendix B - caution for ESD (electrostatic sensitive device) .....	79
Appendix C - registration card .....	81



## ***Conventions used in the manual***

The manual applies to SIMEPROM-01B and SIMEPROM-02aLV simulators. The term **SIMEPROM** is used for the description of components common to both of the types. If there are any differences, the description specifies the respective types.

References to the control program functions are printed in bold, e.g. **Load**, **File**, **Device**, etc. References to control keys are given in brackets <>, e.g. <**F1**>.



---

# *Introduction*

---



The SIMEPROM is an EPROM simulator designed for the research workplaces using EPROM and PROM memories. The SIMEPROM-01B can replace EPROMs from 2716 through 27512, with the access time of  $\geq 120$  ns. The SIMEPROM-02aLV can replace EPROMs and PROMs 2716 through 27040 (LV types too), with the access time of 85 and 120 ns, respectively.

Because the SIMEPROM simulator is equipped with the CMOS RAM whose contents can be easily changed anytime, it can be used in place of the obsolete debugging cycle (about 20 minutes):

- switching off the target
- removing the EPROM from the socket
- EPROM erasing
- EPROM programming
- inserting the EPROM into the socket
- switching on the target

A much simpler procedure (depending on EPROM type and size – 1s...15s) is as follows:

- SIMEPROM "programming".

In addition, the SIMEPROM enables the user to force the RESET signal on the device under development. Thus, software can be debugged without handling the target.

By interconnecting two SIMEPROM simulators you obtain a word configuration required for EPROM simulation in 16-bit systems. Or, they can be used as two independent simulators. By interconnecting four simulators you obtain a double word configuration, or they can be used as four independent simulators.

The SIMEPROM is a peripheral device for IBM compatible PC, AT or higher, portable or desktop PC.

The SIMEPROM-01B can be connected to a computer via a parallel (printer) port designed for connecting a printer. The SIMEPROM-01B can be supplied from the system that is just being debugged, or from an external power supply (option).

The SIMEPROM-02aLV can be connected to a computer via a parallel (printer) port, or an RS-232 serial port using serial communication module. A serial link is optically isolated connection. The SIMEPROM-02aLV can be supplied from the

**Dear customer,**

*thank you for the purchasing  
of our EPROM simulator  
SIMEPROM*

system that is just being debugged, or from the external power supply, included in the delivery.

**Terminology** used in the text and the control program:

<b>Buffer</b>	an area in the main store or on the disk used for storing data required for simulator loading.
<b>Printer port</b>	a parallel port on the PC used for connecting a printer (equivalent term - LPT port)
<b>HEX format</b>	a data entry format readable by a standard text editor, i.e., the 5AH byte, for example, is entered as the character 5 and the character A, i.e., 35H and 41H, respectively. One line of the entry of such a file also contains a data address. The whole entry is protected by a check sum, which makes it possible to detect possible data errors.
<b>HW simulator</b>	a physical simulator connected to an LPT port, either directly or by means of a reduction, or to a serial port.
<b>Logical simulator</b>	individual HW simulators can be integrated into set-ups called logical simulators (LS). A logical simulator can include one, two, or four HW simulators, and thus simulate systems of EPROM BYTE, WORD, or DOUBLE WORD configurations.
<b>Target</b>	device under development (the debugged system) in which is the simulation socket of simulator (named <b>POD</b> ) inserted.

### **Minimum PC system requirements:**

- IBM PC AT or above, free 512 KB RAM
- FDD 3.5", hard disk with 2 MB free space
- operating system: DOS 3.2 or higher, Windows 3.x or Windows 95/98/Me/NT/2000/XP
- free parallel (printer) port or (for SIMEPROM-02aLV only) free serial (RS-232) port





---

# *Hardware*

---



---

## **SIMEPROM-01B hardware installation**

Plug the connecting cable from the simulator into the computer connector, which serves for connecting a printer (printer port). Provided that your computer has one printer port only, remove the printer cable and plug in the simulator cable instead. If several printers can be connected to your computer it is suitable to connect the simulator to a free printer port. If you have one printer port only and you find the previous procedure uncomfortable, we recommend to get additional I/O card with separate parallel port. Connect the other cable connector to the simulator.

The SIMEPROM-01B simulator features with a 28-pin simulation connector (POD). If you need to simulate the EPROM 2716 or 2732, attach a precise socket included in the delivery on the simulation connector. Pin 12 of the socket must be connected to pin 14 of the simulation connector. Unless necessary, do not remove the 28-pin precise socket protecting the simulation connector outlets.

Insert the simulation socket into the debugged system as an EPROM. The simulation socket orientation is illustrated on the topside of it, pin 1 on the flat cable is differentiated by another color.

### **Note:**

*Insert the SIMEPROM-01B connecting cable into the socket in the target so that pin 14 (GND) of the simulation socket will be the first to come into contact with the socket in the device. It should be noted that the device under development must be OFF. If you find this procedure difficult, make a permanent connection between the negative pole of the device under development and the PC frame. The target and the simulator ground potentials must be identical. The simulation connector orientation is illustrated on the simulator cover.*

If is necessary together with simulation of EPROM to control the RESET signal of the target, connect the resetting cable to the simulator connector; connect the red measuring hook to the "+" pole of the resetting capacitor, and the other hook to the "-" pole of it. Moreover, the hooks may be connected to the resetting button, the red one to the button outlet of higher potential. Polarity reversing is non-dangerous but non-functional. Since the resetting circuit control is optically isolated from the SIMEPROM-01B, the measuring hooks can

be connected in parallel to the resetting capacitor, irrespective of its "resetting" to H, or L.

Switch on the power supply of the device under development.

## **Connecting several SIMEPROM-01B simulators**

When working with several simulators, there are several possibilities on hand, depending on the required number of simulators and the computer hardware. Simulators can be connected either directly to printer ports or by using of special adapters 2x8bit (double adapter, optional accessories). The following are numbers of parallel ports (LPTs) and reduction types for all possible combinations that may occur with several simulators used:

### **2 simulators:**

- 1 pc LPT, 1pc double adapter
- 2 pc LPT

### **3 simulators:**

- 2 pc LPT, 1 pc double adapter
- 3 pc LPT

### **4 simulators:**

- 2 pc LPT, 2 pc double adapter
- 4 pc LPT

### **Note:**

*If four parallel ports are needed we cannot manage it due only three standard addresses (H3BC, H378, H278) of LPT ports. H368 and H268 are next addresses which control program can manage.*



---

## **SIMEPROM-02aLV hardware installation**

Select a type of communication between your computer and the simulator (serial or parallel). This selection will depend on the specific conditions (number and type of free ports, need of galvanic separation,...).

### **Parallel communication**

Connect the one connector of supplied interconnecting cable to the computer connector, which serves for connecting a printer (printer port). If your computer is equipped with one printer port only, remove the printer cable from your computer and insert the simulator cable in its place. If several printers can be connected to your computer it will be advantageous to connect the simulator to a free printer port. If only one printer port is available, and you find the above procedure uncomfortable, we recommend to get additional I/O card with separate parallel port. Connect the other cable connector to the simulator.

### **Serial communication**

Insert serial communication module into serial port (COM). Connect connection cable to the serial communication module and the other cable connector to the simulator. Take care for direction of serial communication module. In case of 9-pin connector of PC serial port, use please suitable 9-25 adapter.

The SIMEPROM-02aLV simulator features a 32-pin simulation connector. If you wish to simulate the EPROM 2716 or 2732, put a precise 24-pin socket on the simulator POD. Pin 12 of the socket must be connected to pin 16 of the POD. If you wish to simulate the EPROM 2764 through 27512 put a precise 28-pin socket on the POD. Pin 14 of the socket must be connected to pin 16 of the POD.

Insert the SIMEPROM-02aLV POD into the socket in the target system as an EPROM. The simulation connector direction follows from the designation of connector pins. In addition, you can follow the red-colored strip on the flat cable - it is on the side of pin 1 of the simulation connector.

### **Note:**

*If you connect the SIMEPROM-02aLV to the computer via a printer port, insert the simulation cable into the socket of the device under development so that pin 16 will be the first to come into contact with the socket. Note that the device under development must be OFF. If you find this procedure*

*uncomfortable, make a permanent connection between the ground of the device under development and the PC frame. The target and simulator ground potentials should be identical.*

The SIMEPROM-02aLV can be powered from an external power supply (standard accessory), which should be used if the VCC power source in the device under development does not have any reserve of power for supplying the simulator. Second reason is emulation of LV EPROM – in this case is using of external power supply necessary. If no external power supply is connected, the simulator will be supplied from your application.

If, simultaneously with EPROM simulation, the RESET signal in the target must be controlled, connect the red measuring hook to the "+" pole of the resetting capacitor, and the other hook to the "-" pole. Moreover, the hooks may be connected to the resetting button, the red one to the button outlet of higher potential. Polarity reversing is non-dangerous but non-functional. Since the resetting circuit control is optically isolated from the SIMEPROM-02aLV, the measuring hooks can be connected in parallel to the resetting capacitor, irrespective of its "resetting" to H, or L.

Switch on the power supply of the target.

## **Connecting several SIMEPROM-02aLV simulators to PC**

When using several simulators, there are different possibilities depending on the required number of simulators and the computer hardware. Simulators can be connected either directly to printer ports and serial ports, or by means of a special adapter (option). This adapter allows the connecting two simulators to one parallel port. The following are numbers of parallel ports (LPTs), serial ports (COM), and adapters for all possible combinations that may occur with several simulators used:

### **2 simulators:**

- 1 pc LPT, 1 pc double adapter
- 2 pc LPT
- 1 pc LPT, 1 pc COM
- 2 pc COM

### **3 simulators:**

- 1 pc LPT, 1 pc double adapter, 1 pc COM
- 2 pc LPT, 1 pc double adapter
- 2 pc LPT, 1 pc COM
- 3 pc LPT



- 3 pc COM

#### **4 simulators:**

- 1 pc LPT, 3 pc COM
- 2 pc LPT, 2 pc double adapter
- 1 pc LPT, 2 pc double adapter, 2 pc COM
- 4 pc LPT
- 4 pc COM

#### **Note:**

*If four parallel ports are needed we cannot manage it due only three standard addresses (H3BC, H378, H278) of LPT ports. H368 and H268 are next addresses which control program can manage.*

### **Important notes:**

- When inserting the SIMEPROM into the target system socket the power supply both SIMEPROM and target must be OFF.
- The RESET circuit is adapted to the discharging of the resetting capacitor.
- The simulator ground is connected with the PC ground (both - pole). Thus, when using the SIMEPROM-01B, and also the SIMEPROM-02aLV with parallel connection, the PC ground will be hard connected to the target system ground. It is recommended to supply both the PC and the device under development from the same wall socket. Or, interconnect the PC and the target system grounds by a sufficiently dimensioned conductor.
- When using several SIMEPROM-01B simulators, the Download function requires all simulators to be powered.
- Keep simulator socket contacts clean; replace the protective socket if seems to need be.
- Though SIMEPROM circuits are protected against electrostatic charge, it may happen that they do not absorb all your power.

Since it is difficult to estimate the number of all parallel port kinds, and since the older style parallel interface (Centronics) used for connecting a printer is not standardized either in terms of a protocol or in terms of technical specifications, it may happen that our unit and the PC cannot "understand" each other. Should such a problem occur, which is manifested by communication failure between the PC and the simulator, or by unreliable communication between them, check whether the same communication problem occurs with some other PCs, or other parallel ports in your environment. If not, we would appreciate your documenting the circumstances, i.e. providing us with a precise description of your PC configuration (type, manufacturer, rate, operating system, resident programs; manufacturer, type, or IO at your parallel port), including all other the relevant facts.

**Note:**

*See Appendix A for the RESET circuit wiring.*





---

# *Software*

---



---

## ***The simulator software***

The simulator package contains a CD with the control programs and additional information. The permission to freely copy the content of the CD is granted in order to demonstrate how ELNEC's simulator works. Differences and modifications to this manual (if they exist) may be found in [www.elnec.com](http://www.elnec.com) web site.

### ***Installing of simulator software***

Installing the simulator software is very easy. Insert delivered CD to your CD drive and instal program starts automatically. Install program (setup.exe), which will guide you through the installation process and which will do all the necessary steps before you can first run the control program.

If you use the operating system **DOS**, run **Install** program from CD.

Programs SIM012.EXE (for DOS) and SIM012W.EXE (for Windows) are common control programs for all ELNEC's simulators. We guarantee running of these programs under all of above mentioned operating systems without any problems. Also background operation under Windows is error-free.

For DOS versions of the control program running under WINDOWS 3.11/95/98, we suggest the following: (adjust "Properties" on the icon on your screen):

- Run the control program in **full-screen DOS** (in text mode). In graphic mode the control program might be somewhat slower.
- If you want the control program to run in the **background**, it is necessary to set: (WIN95/98) RIGHT mouse button on the DOS icon/ Properties/ Others/ Background/ Enable; (WIN3xx) in the appropriate PIF file, set the option to Background

### ***New versions of simulator software***

In order to exploit all the capabilities of simulators we recommend using the latest version of SIM012/SIM012W. You may download the latest version of simulator software (file SIM12ARC.EXE /SIMWARC.EXE) from our Internet site [www.elnec.com](http://www.elnec.com), part download.

## **Upgrading the simulator software**

Copy SIM12ARC.EXE /SIMWARC.EXE to a temporary directory then launch it. After extraction you will see all available files needed for the installation process. Then redo a standard installation (run the Setup / Install program). You may delete all files from the temporary folder after the installation process is complete.

## **Using the simulator software**

*The control program delivered by ELNEC, included on the CD in your package, is granted to be free from any viruses at the moment of delivery. To increase their safety our programs include a special algorithm for detecting possible virus infections.*

### **Run the control program**

From command line of MS/PC-DOS write:

SIM012

in Windows environment: double click to icon Sim012w.

After start, control program SIM012 / SIM012W automatically scan all existing ports and search for the connected any ELNEC's simulator. Program SIM012 / SIM012W is common for all the ELNEC's simulators, hence program try to find all supported (SIMEPROM-01B, SIMEPROM-02, SIMEPROM-02aLV) simulators.

**Notes:** *When the SIM012 / SIM012W program is started, program is checked for its integrity. Than the program display a standard user menu and waits for your instructions.*

If the control program cannot communicate with the simulator, an error message appears on the screen, including error code and description of possible reasons (disconnected simulator, bad connection, power supply failure, incompatible printer port, ...). Eliminate the error source and press any key. If error conditions still exist, the program resumes its operation in the demo mode and access to the simulator is not possible. By implication, the simulator download is not possible. If you cannot identify the error cause contact the simulator manufacturer.



---

## Using the SIM012 control program

### Starting the program

After starting the SIM012.EXE control program is checked for its integrity and presence of SIMEPROM.

When the SIM012 program is started it automatically finds the connected simulator(s) at all existing LPT and COM ports (the COM address is entered only in the case of SIMEPROM-02aLV). This mode can be suppressed by specifying a parameter, including the LPT or COM hex address, to which the simulator is connected. In this case, the control program searches for the simulator only at the desired port. The parameter form is /Ayyy, where yyy is the address of the particular LPT or COM specified in a hexadecimal form. If this parameter is used more than once, each time with a different LPT or COM address, the program will be provided with a list of ports at which the simulator(s) can be found. This list is also accepted by the **Device / Presentation** service during the program run.

An example of starting the program with parameters:

```
SIM012 /HE /A378 /A2F8
```

If everything is O.K., the program provides a standard user screen and waits for your commands.

If the control program cannot communicate with SIMEPROM it gives an error message, including a description of possible reasons (simulator disconnected, or connector not inserted properly, incompatible printer port or serial port, or printer port set up as a bidirectional port...). When the error is eliminated press any key. If you fail to eliminate the cause of error, the program resumes running in the **Demo** mode after depressing any key. By implication, the simulator download is not possible. If you cannot identify the error cause contact the simulator manufacturer.

The SIM012 control program (from the 4.13 variant) includes the logging of communication errors of the SIMEPROM-02aLV simulator into the SIM012.ERR file. The file is created in the working (current) catalogue; if the file already exists, new error protocols are connected to its end.

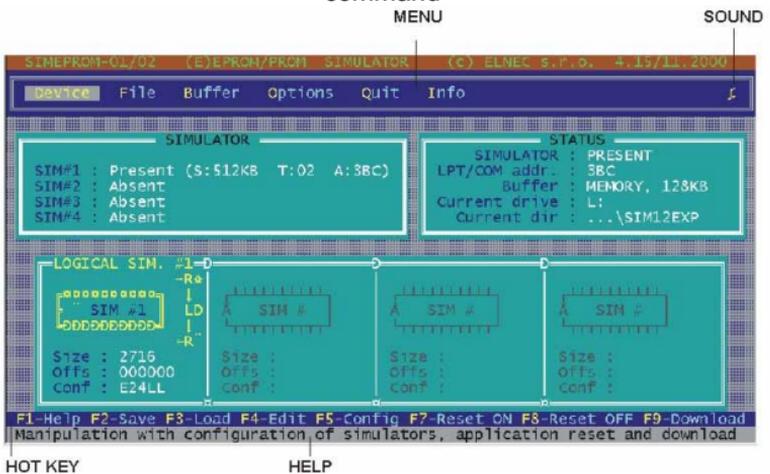
#### Note:

*SIMEPROM-02aLV must be powered up before starting the control program.*

## User screen

The user screen includes the following parts:

- **IMATRICATION** name, copyright, and control program version
- **MENU** offer of control program options
- **SIMULATOR** window information about the presence of HW simulators at the ports
- **STATUS** window information about the simulator and control program status, about the buffer size and its place (memory, disk), and about the directory name from which the program was started
- a window for displaying the current configuration and highlighting the currently active logical simulator.
- **HOT KEYS** a list of hot keys to which the program responds.
- **HELP** a brief description of a selected command



Menu selection is carried out in the normal GUI fashion - either by cursor moving plus pressing **<Enter>**, or by typing the highlighted letter in the wanted menu or - of course - by mouse. Hot-keys are available for even quicker selection of intensely used commands.

### Note:

*The data entered through a keyboard are in the HEXA format, excepting ASCII blocks in Buffer/View/Edit command.*



## List of hot keys

<F1>	Help	Calls the Help
<F2>	Save	Save a file
<F3>	Load	Load a file into the buffer
<F4>	Edit	Viewing/editing of buffer
<F5>	Config	Adjusting the configuration
<F7>	RESET ON	RESET signal activation in active LS
<F8>	RESET OFF	RESET signal deactivation in active LS
<F9>	Download	Downloading to active LS

## Help

The context-sensitive Help can be selected anytime by pressing the <F1> key, with the exception of cases when the simulator executes an operation (e.g. downloading). When the Help window is open, depress <F1> once more to get a list of all Help messages used in the system. You can retrieve and display any of the messages. A Help text may include the so-called **key-words** (highlighted items) some of which may provide you additional information (the so-called cross references).

The following items are identified by colour:

- words identifying the keys referred to by the current Help
- any other important words
- current cross reference; additional information may be obtained by depressing the <Enter> key.
- non-active cross references; use cursor keys to select them and depress <Enter> to get additional information.

## Device

This menu provides commands for setting up the HW simulator presence, their assignment to logical configurations (logical simulators - LSs), and for handling with these LSs. The selected configuration is always graphically displayed in a window at the display background. Since it can contain

several LSs, one of them is active at any moment. Activity means that the use of the **Reset ON**, **Reset OFF**, and **Download** commands will be aimed at this active LS. In the window, this LS is distinguished from all other LSs graphically and by colour. LS activity can be switched over by using the corresponding command in this menu or by keys <Alt> + the LS ordinal number. The <Tab> and <Shift+Tab> keys are



used to increase and decrease, respectively, the LS activity. The active LS is also referred to by the **Additional info** command which provides additional information which are not displayed in the background window.

### **Device / Presentation**

This command provides information about the presence of HW simulator at the LPT or COM port. If the program runs in the **Demo** mode, the information follows from the SIM012.SET file (if available) or from the default set up. Since the program automatically identifies the connection of HW simulators at the program start, it cannot identify any changes at the ports during program run (if any of the simulators is either connected or removed). Therefore, use the **<F10>** service for this command and for making the information about the HW simulator presence at a port more accurate. This service executes automatic simulator identification and updates the information in the working window, including the complete display background. This information is closely related to the command for setting up a configuration, which provides all the possible logical simulators (LS) that can be composed from the current number of present HW simulators.

If the program fails to find at least one simulator at any of the ports (irrespective of the type), it will automatically run in the **Demo** mode. Control of **reset** signal and simulator **download** is not possible in this mode. The current configuration will be set up according to the SIM012.SET file. If it does not exist, the **default** configuration is selected – presence of one HW simulator of the 01 type assigned to the **BYTE** LS type. This is the minimum requirement for starting the program, because it is conditioned by creating a buffer with the minimum size of 64kB just for one 01 type HW simulator. If the program finds a simulator(s) at the port, it compares the acquired information with the SIM012.SET file content, and if no differences in presentation are found, it starts running, with the configuration corresponding to that file. If any differences are found (the status at the port does not correspond to the file content), it attempts to maintain the LS at least partially; the other LSs will be cancelled. If the file is absent it will generate a **BYTE** type LS from each present HW simulator.

A special case obtains, if the program start is conditioned by creating a buffer larger than admitted by the main store and if the disk size is not sufficient either. The program selects the default option based on which it assigns the first HW simulator present at the port to the **BYTE** configuration. In the case of **Demo** mode, it will be the type 01 simulator irrespective of the SIM012.SET content. Thus, the minimum requirement for program start is fulfilled.

**Note:**

If a buffer of at least min. size (64kB) cannot be created, the program offers manual entering the path to disk where a buffer may be created.

**Device / Presentation / Status**

This working window provides you a current status of HW simulator presence at the port, and enables you to change this status. Use selectional keys (<↓>, <↑>) to select a HW simulator. Acknowledge by <Enter> to change its status (present / absent). Based on this information, the **Configuration** command provides all possible logical simulator variants following from this status. This method of updating is only recommended for demonstration and testing,

<b>SIM#1</b>	<b>: Present</b>
type	: 01
size	: 64KB
=====	
<b>SIM#2</b>	<b>: Absent</b>
type	: 01
size	: 64KB
=====	
<b>SIM#3</b>	<b>: Absent</b>
type	: 01
size	: 64KB
=====	
<b>SIM#4</b>	<b>: Absent</b>
type	: 01
size	: 64KB

because it is only applicable if the program runs in the **Demo** mode. In other cases, it is blocked, and the only way of updating this information is to use the <F10> service. The latter automatically identifies the HW simulators at the port. In addition, it is the only way of switching over the program run from **Demo** to default and vice versa. All LPT and COM ports are scanned, and the set-up of the information in this window and in the **STATUS** window at the display background as well as all relevant program control variables depends on the scanning operation.

The identification system works according to the following rules:

- at first, SIMEPROM-01B is searched at all LPT addresses in the order of 3BCH, 378H, 278H, and additional 268H, 368H.
- SIMEPROM-02aLV is searched at the addresses, which are not occupied by the 01 type simulators. The search starts at point 1.
- at last, ports 3F8H, 2F8, 3E8, 2E8 are scanned in the given order, and SIMEPROM-02aLV is searched.

Since simulators can be connected to LPT directly or via reduction, the following rule applies – the positions are only maintained at the first reduction encountered by the program during scanning (irrespective of type). Any other simulators (including those in reductions) gradually occupy the remaining free positions (i.e. HW sim #1 to #4). Therefore, the method of connecting the simulators to a PC, especially if their number is high, should be carefully considered.

**Caution:**

*Any change in this window results in cancelling the current LS configuration without any prior warning.*

**Note:**

*The **DEVICE/PRESENTATION/TYPE** and **.../SIZE** items serve illustration purposes only, and are updated according to the particular simulator type.*

## **Device / Configuration**

HW simulators can be connected to a PC via parallel ports (SIMEPROM-01B and SIMEPROM-02aLV) either independently or through a special-purpose reduction, or through serial ports (only SIMEPROM-02aLV - independently, i.e. one port – one simulator). The connection methods can be arbitrarily combined, for example, two HW simulators can be connected to one LPT port through a "2x" reduction, or to two independent LPT ports, etc. Depending on the number of connected HW simulators (irrespective of the type), this command provides all possible combinations (i.e. logical simulators-LSs) that can be built from these simulators. When one of the combinations is selected, a working window is generated which makes it possible to combine simulators to logical configurations (i.e. the above-mentioned LSs).

*HW simulators can be freely combined irrespective of the type (SIMEPROM-01B, 02aLV) and PC connection (parallel, serial port), i.e., the Word configuration can be built from, for example, SIMEPROM-01B connected to LPT through a reduction and SIMEPROM-02aLV connected to a serial port.*

Moreover, the command enables you to assign LS the basic features, starting from the EPROM type to be simulated, through the position in buffer, up to the description of loading sequence. All set-up elements in the working window are described in detail in the context-sensitive Help accessible via the <F1> key. The configuration must be set up each time the number and/or sequence of the connected HW simulators changes by the **Presentation** command. This command turns the LS information window at the display background inactive, and the access to the LS is blocked. The working window for configuration is also used to change any of the features of any LS (EPROM type, position in buffer, and loading sequence).

If you wish to change the current configuration, the program gives a warning indicating that the former configuration will be cancelled. If you don't assign LS all HW simulators, they will be considered to be inactive, and the access to them will be blocked.

**Caution:**

*If you change the HW simulator assignment to LS in the current configuration, all RESET signals will be set up to an*



*inactive state in order to avoid a conflict due to various levels of these signals at the individual HW simulators.*

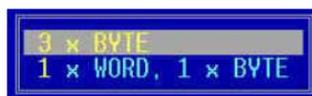
The Figure on the right illustrates the information window in the case of a single simulator connected:



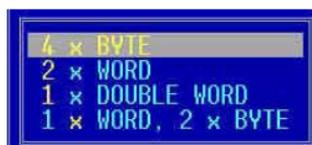
When two simulators are connected, there are two options:



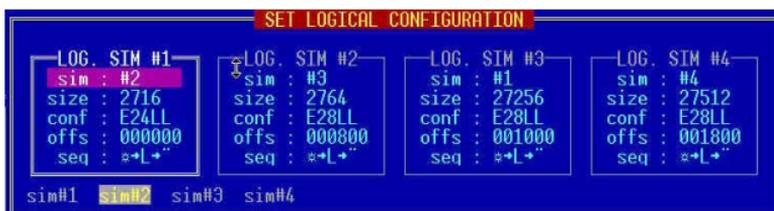
When three simulators are connected, there are also two options:



When four simulators are connected, there are four options:



### **Device / Configuration / 4 x BYTE**



This command generates a working window enabling you to assign 4 present HW simulators to logical simulators (LSs) of the BYTE type, and to assign these LSs their respective sizes (EPROM type to be simulated), place in the buffer (offset), and downloading sequence.

### Device / Configuration / 2 x Word



This command generates a working window enabling you to assign 4 present HW simulators to logical simulators (LS) of the WORD type, and to assign these LSs their respective sizes (EPROM type to be simulated), place in the buffer (offset), and downloading sequence.

### Device / Configuration / 1 x DOUBLE WORD



This command generates a working window enabling you to assign 4 present HW simulators to a logical simulator (LS) of the DOUBLE WORD type, and to assign this LS its size (EPROM type to be represented), place in the buffer (offset), and downloading sequence. Be sure to assign all HW simulators; otherwise the LS will be inactive.

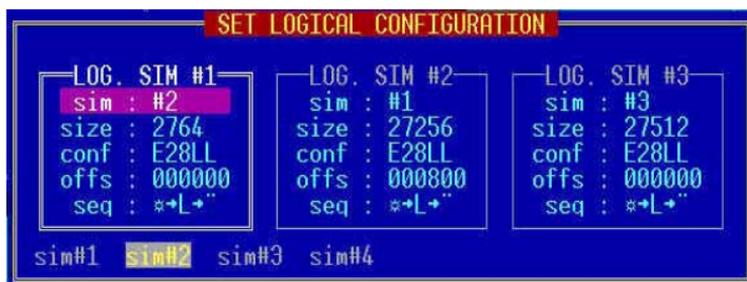
### Device / Configuration / 1 x WORD, 2 x BYTE





This command generates a working window enabling you to assign 4 present HW simulators to logical simulators (LS) of the WORD and BYTE types, and to assign these LSs their respective sizes (EPROM type to be represented), place in the buffer (offset), and downloading sequence.

### **Device / Configuration / 3 x BYTE**



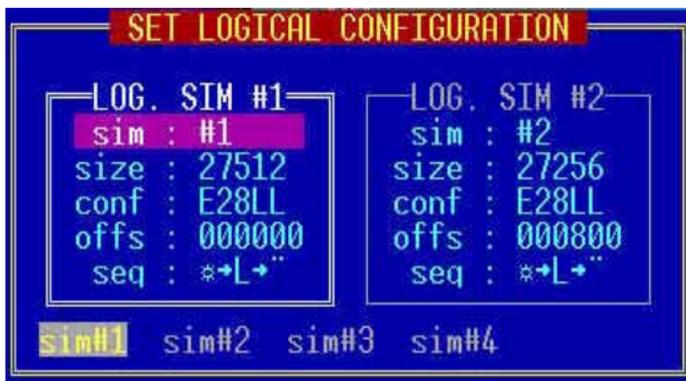
This command generates a working window enabling you to assign 3 present HW simulators to logical simulators (LS) of the BYTE type, and to assign these LSs their respective sizes (EPROM type to be simulated), place in the buffer (offset), and downloading sequence.

### **Device / Configuration / 1 x WORD, 1 x BYTE**



This command generates a working window enabling you to assign 3 present HW simulators to logical simulators (LS) of the WORD and BYTE types, and to assign these LSs their respective sizes (EPROM type to be represented), place in the buffer (offset), and downloading sequence.

### Device / Configuration / 2 x BYTE



This command generates a working window enabling you to assign 2 present HW simulators to logical simulators (LS) of the BYTE type, and to assign these LSs their respective sizes (EPROM type to be simulated), place in the buffer (offset), and downloading sequence.

### Device / Configuration / 1 x WORD



This command generates a working window enabling you to assign 2 present HW simulators to a logical simulator (LS) of the WORD type, and to assign this LS its size (EPROM type to be represented), place in the buffer (offset), and downloading sequence. Be sure to assign both HW simulators; otherwise, the LS will be inactive.

#### ***sim, size, conf, ofs, seq***

When the desired combination is selected, a working window is generated which makes it possible to combine the simulators to logical configurations (the above mentioned LSs) in a simple way. HW simulators can be arbitrarily combined irrespective of the type (SIMEPROM-01B, 02aLV) and PC connection



(parallel port without or with reduction, serial port) i.e. the **WORD** configuration can be built from, for example, SIMEPROM-01B connected to LPT through reduction (or without it) and SIMEPROM-02aLV connected to a serial port. In addition, the working window allows for assigning LS its basic features, ranging from the EPROM type to be simulated and the place in buffer, up to the description of the downloading sequence. All set up elements in the working window are described in detail in the context-sensitive Help accessible via the **<F1>** key. The configuration must be set up each time the number and/or sequence of the connected HW simulators changes by the **Presentation** command. This command turns the LS information window at the display background inactive, and the access to the LS is blocked. The working window of configuration is also used to change any of the features of any LS (EPROM type, position in buffer, or loading sequence). If you wish to change the current configuration, the program gives a warning indicating that the former configuration will be cancelled. If you don't assign LS all HW simulators, they will be considered to be inactive, and the access to them will be blocked.

#### **sim HW simulator assignment**

This position enables you to assign an active logical simulator one of the free HW simulators. The number and the order of HW simulators are based on the **Presentation** item set-up. An active logical simulator is one highlighted by means of a double box, symbolizing a HW simulator configuration. In addition, the **WORD** configuration distinguishes between the upper and the lower bytes, similarly to the **DOUBLE WORD** configuration (upper word – upper and lower bytes, lower word – upper and lower bytes). Use the selectional keys (**<←>**, **<→>**) to select a free HW simulator from the options in the lowest line of the configuration window. Press **<Enter>** to assign it to the current logical simulator (LS). You can perform the same operation by means of keys of characteristic colour. The assigned HW simulator cannot be subsequently assigned to any other LS. If you wish to reorder the assignments, use the **<Ctrl + Enter>** key combination to release the HW simulator from the current LS and assign it to some other LS. If HW simulators have not been assigned to all LS components (for example, the upper and the lower bytes in the **WORD** configuration), the size, the place in buffer (offset), and the downloading sequence cannot be set up for such an LS. After completing the configuration process (press **<Esc>**), this LS is not accessible too. Be sure to create complete LS by assigning it all physically present simulators.

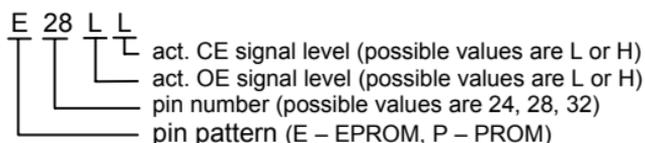
#### **size setting up size for log. simulator**

This position is used to set up the EPROM type to be replaced by the current logical simulator (LS). In addition, this data indicates the size occupied in the buffer. The selected EPROM

size can be multiplied according to the specific configuration (for **WORD 2x**, for **DOUBLE WORD 4x**). The value is related to the set-up of **offset**, which assigns the current LS an start address in the buffer. By selecting a circuit you automatically update the **conf** item used for circuit description (i.e. pin number and pattern - as EPROM or PROM, active OE and CE signal level). Use the selection keys (<=>,<=>) to directly select the EPROM type to be replaced by the simulator. The max. EPROM size for the **BYTE** configuration is limited by the HW simulator size. Since the program makes it possible to combine both types of HW simulators to **WORD** and **DOUBLE WORD**, the max. EPROM size is determined by the smallest simulator in the configuration.

### conf setting up the (E)PROM description

This item symbolically describes the currently simulated circuit type in terms of pin pattern and number as well as active level of OE and CE signals. The meaning of the individual symbols is as follows:



If you require a different default pin pattern, e.g., for the simulation of PROM memories, press **<Enter>** to generate a secondary window, which enables you to describe the simulated circuit by pin number and pattern (as EPROM or PROM) and by defining the active level of OE and CE signals.

### Caution:

*This item is only available for SIMEPROM-02aLV simulators. For SIMEPROM-01B and combined configurations (e.g. WORD with 01 and 02aLV), it is blocked. The pin number is, however, automatically modified according to the selected EPROM size.*

### seq setting up the loading sequence

Use the selectional keys to select one of the sequence options, given in the lowest line of the working window to be used in simulator 'programming'. Acknowledge the selected option by **<Enter>**. The meaning of the individual symbols is as follows:

- R\* - RESET active
- L - download
- R• - RESET deactive

Thus, the R\* -> L -> R• expression means that the simulator filling is preceded by reset. When it is full, reset is finished. If



any of R symbols is missing at the beginning or at the end, the current reset state before or after simulator filling will be preserved. For LS of the **WORD** and **DOUBLE WORD** types, handling with the RESET signal applies to all HW simulators, which are integrated in the logical simulator.

### **offs setting up the offset for log. simulator**

This command places the current logical simulator (LS) in the buffer within the range of its size. Since the program does not block the already occupied buffer areas they may overlap. Therefore, try to avoid such a solution in order to prevent any undesired collisions. The data loading from a disk file into the buffer area occupied by the current LS is undesired. If this area overlaps with another one, it will be partly or completely overwritten with new data. The buffer size is sufficient to accommodate all LSs without any overlap. Use the selectional keys (<<=>, <=>>) to immediately change the start address of the current LS in the buffer by a step of the EPROM 2716 size in the complete buffer range. If you wish to increase the step combine the selectional keys with the left-hand <Shift> key. The step increases to the EPROM 27128 size. In addition, if the <Enter> key is depressed the offset can be entered manually. The program automatically transfers the selected value to the nearest address divisible by 800H and checks the 'overflow' of the current area size through the buffer end. Precise information about the place of the current area in the buffer is graphically and numerically provided by the **Additional Information** service (key combination <Ctrl+F1>).

### **Device / Reset ON**

This command executes RESET in the development system, if a reset cable is connected to a reset capacitor or to a RESET button in the development system. This operation is equivalent to keeping the RESET button depressed. The command is applied to the currently active logical simulator (LS). The current status of the RESET signal is displayed in the information window reserved for the particular LS.

### **Device / Reset OFF**

This command releases the RESET in the development system, if it is in the RESET state. This operation is equivalent to releasing the resetting button. The command is applied to the currently active logical simulator (LS). The current status of the RESET signal is displayed in the information window reserved for the particular LS. The effect upon the SIMEPROM-01B simulator with regard to RESET signals is equivalent to the **Reset ON** function.

### **Device / Download**

This command transfers the data from the selected buffer area to the currently active logical simulator (LS). The buffer area, which fills the LS depends on the **offset** variable and the

EPROM size (**size** variable) represented by LS. In addition to data transfer to the simulator, the individual operations take place in accordance with the adjusted downloading sequence, for example, **Reset ON** before downloading and **Reset OFF** after downloading.

The SIMEPROM-01B content fills independently of the simulated EPROM size: the data for EPROM 27256 is written twice, in particular from address 0000h and address 8000h (thus, the pin 1 level is insignificant). The data for EPROM 27128 is written four times, etc.

The SIMEPROM-02aLV download is different and follows from the number of pins of the currently simulated EPROM. The entry is as follows:

- 24 pin circuits – the filled area size is 4KB  
2 x 2KB for EPROM 2716  
1 x 4KB for EPROM 2732
- 28 pin circuits - the filled area size is 64KB  
8 x 8KB for EPROM 2764  
4 x 16KB for EPROM 27128  
2 x 32KB for EPROM 27256  
1 x 64KB for EPROM 27512
- 32 pin circuits - the filled area size is 512 (256) KB  
4 (2) x 128KB for EPROM 27010  
2 (1) x 256KB for EPROM 27020  
1 x 512KB for EPROM 27040

**Note:**

*The numbers in brackets refer to the 256KB version of SIMEPROM-02aLV.*

**Caution:**

*If a communication error occurs during communication with SIMEPROM-02aLV, an error message is displayed and the relevant LS is deactivated. The HW simulator at which the error occurred is considered to be absent. In that case, assess all possible error causes by following the error message recommendations, and select the <F10> service in the **Device / Presentation** menu for automatic reinitialisation of the connected simulators.*

## Device / Additional information

This command provides additional information about the current logical simulator (hereinafter

LS), which is the LS highlighted by a specific colour attribute and a double box at the symbolically represented EPROMs in

```
Additional informations about logical sim. #1
Drive : C:
Directory : \TEMP
Filename : TEMP.BIN
Buffer : 000000H + 03FFFFH (*)
EPROM : 000000H + 00FFFFH (*)
Position :
```



the information window at the display background. The information includes the complete file name (corresponding to the LS) divided to a disk, a path, and a file name with an affix. In the second part, the position of LS (or the size of equivalent EPROM) in the buffer is described both numerically and graphically. Each graphical representation item has its specific colour, which makes it possible to provide complete information about the configuration structure. The information is completed with the current buffer size and the start and end addresses of the area occupied by the simulated EPROM. The window can be cancelled by operating any key. However, if you depress any of the keys for switching-over the LS operation (<Tab>, <Shift+Tab>, Alt+ LS ordinal number) the window does not disappear; instead, it provides new data about the just activated LS.

### **Device / (LOG. SIM #1 - #4)**

This item activates the corresponding logical simulator. Hence, the **Reset ON**, **Reset OFF** a **Download** command are aimed at this active LS. The activity change is captured in the permanently lit up information window at the display background by highlighting the particular LS using colour and graphics. The buffer area occupied with the active LS is then offered in the form of an start and end addresses of this area in the working windows, for saving or reading the files as default values. These addresses are also used for jump operations in buffer scanning or editing (jump to the start or end address of the active area). The switching of LS can be accelerated by using key combination <Alt> + LS ordinal number, or <Tab> and <Shift+Tab> for gradual switching up or down, respectively.

### **File**

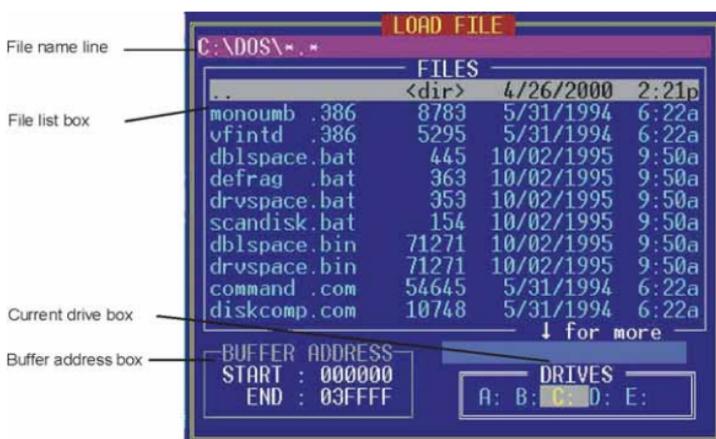
This menu enables a user to load a file into the buffer, save the buffer content in a file, execute basic operations with the files, set up and view the directories, change the current disk and modify the start and end buffer



addresses for file reading and saving in the following format:

- **binary**
- **IntelHEX**
- **Tektronix**
- **Motorola**
- **Exormax**
- **MOS technology**

In executing the individual commands from the **File** menu, an information window for reading (saving) is displayed. This working dialogue window includes the following parts:



**File name line** for the manual entering of the file name to be loaded (to keep the buffer content) and the file name with pseudo-characters in order to create a mask for filtering out the displayed type of files. Key **<F3>** is used for viewing the content of the text file referred to by the cursor. The **<Ctrl+Enter>** key combination speeds up the change from the current mask to the **\*.\*** mask. With regard to the current input line content it can acquire the following modifications:

COMMUNICATION LINE	NEW MASK
... \tmp	... \tmp*.*
... \tmp.	... \tmp.*
... \tmp.e	... \tmp.e*
... \tmp.ex	... \tmp.ex*
... \tmp.ext	... \*.*
... \tmptmptm	... \tmptmptm.*

If the input line contains an invalid file name or an old mask the new mask will be **\*.\***.

#### Notes:

*The maximum length of text in the communication line is 256 characters. The current position is identified by the cursor. The program makes it possible to modify the file name only. A directory can be changed either through the file entry window or by operating the **<F10>** key which displays a tree structure. If a character string is very long the text is shifted to the left. The program filters all illegal characters. An error is displayed if the established syntax is not observed.*



**File list box** contains a list of files (max. 500) in the current directory, which correspond to the input line mask and a superordinate directory, including all sub-directories; each of them is identified by its respective colour. If the file number in the current directory exceeds 500 the program displays a warning. If the file to be read does not belong to them, write it manually or reduce the displayed file number by selecting an appropriate mask.

The window enables you to move and select alphabetically arranged files in the current directory. The designated file is automatically rewritten to the communication line, which thus carries the information about the complete path. When you move through the directories, the current mask is written instead of the file name. All editing keys are active (<←>, <→>, <↑>, <↓>, <Home>, <End>, <PgUp>, <PgDn>). Moving through the directories and changing the current disk is simple (key combination **Alt** + initial letter).

#### **Directory tree structure**

Depress <F1> to display the directory tree structure, which enables you to find and change the desired directory very quickly. This function can display max. **500** directories of the current disk. If this number is higher or if the free memory size is insufficient for displaying all directories, only those directories are displayed, which do not exceed this size. A warning displayed on the right-hand side of the lower working window indicates this state. Moreover, the program checks the number of subdirectory nesting levels and limits them to 13. For larger number of nesting levels, it provides a warning by displaying a message on the left-hand side of the lower working window box. The cursor keys are used to move along the directory tree (<Home>, <End>, <PgUp>, <PgDn>); use <Enter> to change the current directory to that highlighted by the cursor. <Esc> is used for cancelling the tree and returning to the original directory.

**Buffer address box** is used for entering the current buffer area (start and end addresses) for saving in a disk file (all formats) and for loading a file from the disk (binary format only; the HEXa format controls the loading itself). If the set-up area is smaller than the file size, only that file part is loaded which corresponds to the selected area size. Addresses are entered in the hexadecimal form. The acknowledgement is followed by extent check. This buffer address change is local only for one **Load (Save)** procedure, which means that it has no influence on the addresses set up in the **Device / Addresses** menu. An appropriate set-up in the **Options / HEX load** menu can correct the control of loading in the HEXa format. This

window can be activated by the <Tab> key. If you wish to return press <Tab> once more.

**Current drive box.** Use the key combination <Alt> + the letter designating the particular disk to change the current disk.

The information window

**Loading data from disk to buffer (Saving data from buffer to disk)** can be displayed using the **Load (Save)** command. It provides information about the buffer and file

addresses. The first window line (**DEVICE POINTER**) is not used (cf. the Device menu). When working with small files you can hardly notice the window content – it is displayed only during the **Load (Save)** operation. With large files, you can interrupt the process by depressing the <Esc> key, which is followed by the **Access cancelled by user** message.



INFO	
Loading data from disk to buffer	
DEVICE POINTER	000000
BUFFER POINTER	0276BE
FILE POINTER	0276BD

## Load

This command is used for loading the data from a disk file to the buffer. When you select file from the file list, the control program automatically detect the input file format. After you confirm detected file format, then the content of selected file will be loaded to the buffer.

### Notes:

- **Binary** – *binary* loading of data from a file to the buffer.
- **Intel HEX** – it loads the **Intel** (extended) **HEX** data from a file to the buffer. All entries are checked against a security byte at the end of each entry. In the case of conflict, the program displays an error and interrupts the loading process. If you acknowledge the Interactive HEX loading option (YES) in the **Options/ HEX Load** menu, the program asks for a segment for replacing the lowest segment (02 type entry) in the load HEX-a file.
- By file loading from disk to buffer by Intel HEX format with using a negative offset, addresses out of range of buffer are ignored. In this case a control program reports warning - "The file wasn't loaded all ". From resident versions is the negative offset system supported by SIM02R control program only.
- **MOTOROLA** – It loads from a disk file to the buffer, using the Motorola's **S-format** The S0 data entry type is ignored. Simultaneous occurrence of several data entry types, such as S1 and S2 (such a case should be, however, avoided) is not considered to be an error. All entries are checked against a security byte at the end of each entry. In the case of



*conflict, the program displays an error and interrupts the loading process.*

If your disk contains a large number of files used for EPROM programming, you are sure to have made notes, saved in the text files. Depress **<F3>** to display the retrieved text file.

## Save **<F2>**

This command is used for saving the data created in the buffer, modified or loaded from a circuit to a disk file by a selected format (binary, Motorola, or Intel (extended) HEX format). The **<F2>** hot key is equivalent to the sequence File **<Enter>** Save **<Enter>** **<Enter>**, i.e., the saving format is the same as that used for the last time.



- **Binary** – It saves the buffer to a disk file in the **binary** format.
- **Intel Hex** – It saves the buffer to a disk file using the **Intel** (extended) **HEX** format.
- **Motorola** – It saves the buffer to a disk file using the Motorola's **S-format**. It automatically specifies the data entry type to be used (depending on the selected buffer block size).
- **Exormax** – It saves the buffer to a disk file using a special **MOTOROLA** format designed for the **WORD** systems. It automatically specifies the data entry type (**S-format**) depending on the selected buffer block size.
- **MOS Technology** – It saves the buffer to a disk file using the **MOS Technology** format. The maximum size of the selected data block may not exceed **64KB**.

## Buffer

It provides various options of using the buffer, such as block operations, erasing the content, string filling, 4 types of checksum, and, certainly, editing. The latter includes additional services (finding a string and its substitution, output to the printer, etc.).



## Buffer / View/Edit <F4>

BUFFER : 256 Kbytes		VIEW / EDIT BUFFER																CURRENT ADDRESS : 000000		MODE : View	
Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF				
000000	AD	05	5B	77	BB	6A	AB	7C	CF	82	63	14	87	8F	C0	1E	s. [w]jz]0éc. cC <sup>L</sup> .				
000010	A8	A3	04	9C	AF	58	26	11	99	CF	BE	1B	EF	EC	2C	52	Eú. t>X8. 00z. v. R				
000020	16	D1	09	62	D7	99	D1	65	2A	57	D0	83	49	4F	4A	CD	.B. b]0Be-WdãIQ]-				
000030	F2	75	EF	57	FD	24	7F	96	20	21	BA	FA	FF	D0	8C	CD	.u Wf\$ I !!] di-				
000040	78	B6	78	4B	2D	30	42	03	54	73	E2	6F	BC	C6	A6	D0	x&K-0B. T50o <sup>A</sup> AZd				
000050	21	FC	A6	4D	B3	35	6F	4D	E1	D6	E8	11	6C	C7	89	97	!RZ] 5oMIR. lae\$				
000060	AA	ED	BE	AC	18	E8	96	50	24	0F	B2	50	39	AA	6A	1F	YzC. RIP\$. P9 j.				
000070	0C	70	42	F6	27	42	8D	2D	B5	27	60	DA	8E	88	BB	A9	.pB+ BZ-A' rAze				
000080	84	AD	F4	FB	ED	79	65	43	72	63	57	9E	17	B6	2F	B2	ás_üVyeCrcWx. A/				
000090	8C	0B	D8	CA	B3	04	72	30	73	4B	3A	CC	BE	CC	BA	FB	i. e <sup>A</sup> . r0Sk: lz}ü				
0000A0	7E	2F	32	B2	06	9A	47	D4	16	A6	EB	D1	AE	A0	8D	82	0/2. UGd. Z00«aZe				
0000B0	DD	03	83	41	AF	32	B6	4D	F4	7B	8D	5E	53	4A	1D	85	w. ä»2AM' [Z'SJ. ü				
0000C0	91	AB	8F	48	BA	04	D3	FB	E9	11	84	61	57	21	1C	85	LzCH] Eüü. aaM! .ü				
0000D0	A7	90	5A	D4	72	86	F1	7C	0F	EF	73	0A	A5	BB	7D	40	zEZrc' ] . s. qj]@				
0000E0	74	58	25	34	61	6E	A2	B0	C2	DB	3B	C7	69	F0	73	B4	t%4ano. :. s. qj]@				
0000F0	F4	EB	74	F9	54	B5	BA	B6	9C	DE	02	47	0D	D7	72	22	Ut TA]A7U. G. Ir-				

This command is used for viewing or editing the buffer data. Use cursor keys to move in the whole buffer and find the required block for editing. The modified data is identified by a specific colour (changes at the previous 64 address labels are stored).

### Buffer hot keys:

<Tab>	HEX / ASCII editing type change
<F10>	viewing mode change / editing
<F9>	current address change
<F8>	finds and substitutes a string (max. 16 ASCII characters)
<Shift+F8>	repeats the last string substitution command
<F7>	finds a string (max. length 16 ASCII characters)
<Shift+F7>	repeats the last finding command
<F6>	prints out the selected buffer block (see <b>Print block</b> )
<Ctrl+F6>	block printing with the "Insert new paper ...", poll used for printing on separate paper
<↑>, <↓>, <→>, <←>	cursor up, down, right, left
<Home/End>	jump to the begin / end of line
<PgUp/PgDn>	jump to the previous. / next page
<Ctrl+Home/End>	jump to the begin / end of page
<Shift+Home/End>	jump to the begin / end of device
<Ctrl+PgUp/PgDn>	jump to the begin / end of buffer
<Left Shift+PgUp/PgDn>	paging per each 1 000H bytes
<Right Shift+PgUp/PgDn>	paging per each 10 000H bytes

- **Note 1:**

Characters 20H - FFH (in the **Edit ASCII** mode) and numerals 0..9, A..F (in the **Edit HEX** mode) immediately changes the edited buffer content.

- **Note 2:**

Double-underlined keys are only effective in the **Edit** mode.



**Print block** This command is used to print out the content of the selected buffer area. In addition to the start and end block addresses, a comment with max. 67 characters may be entered. The comment is then printed out at each page heading. The default comment is the file name previously loaded from the disk. The heading with the current page number, the start and end addresses of the selected block, time, date, and the above-mentioned comment are printed as first. Max. 300H bytes in the HEX and ASCII format are printed out at each page. For the sake of lucidity, bytes are divided into blocks of 128 bytes both horizontally and vertically. Print can be finished before its completion by means of **<Esc>**. The program displays a warning indicating that the printer will be reset. The command will be executed after acknowledgement.

**Jump to addr.** Use **<F9>** to change the current address and place the cursor to a new address in both modes, i.e., View / Edit. All editing keys, including the switching of the **Insert / Overwrite** mode, are active. The cursor shape indicates the current mode. The default mode is **Overwrite**. When **<Enter>** is operated an address is tested for syntax (only hexadecimal characters) and extent (within the buffer size). An error is indicated by a audio alarm, and the editing process goes on.

**Find** This command displays a dialogue window in which a string in the HEXa or ASCII form, which you wish to find, may be entered. In addition, you can set up several



options affecting the retrieval. The **<Tab>** key can set up the individual options. The **<Shift+F7>** key combination is used for repeating the previous retrieval command. All settings adjusted in the last Find window, remain in force.

- Write a string you wish to find in the **Find string** box, and acknowledge by **<Enter>**. This starts the retrieving operation and the set-up configuration is saved as current. Operate **<Ecs>** to leave the **Find string** operation.
- **Direction** specifies the direction of retrieving. **Forward** (from the current cursor position or from the buffer begin to the end) is a default value. **Backward** retrieves towards the beginning.
- **Origin** specifies the retrieval beginning. If you select **From cursor**, the retrieval starts from the current cursor position (in the **Edit** mode) or from the current address

(View mode). The **Entire scope** option starts retrieving from the buffer begin or end.

- **Options** enables you to decide whether you wish to find only the first expression or all expressions. If you select the **First expression** option, the retrieving operation is completed after finding the first string. In the case of **All exp. with ask** the program stops at the found string and goes on retrieving another string occurring in the buffer as soon as the <Y> or <Enter> key is depressed.

**Find & Replace.** This command displays a working window for replacing a string in which you may enter both string to be found and string to replace the former one. In addition, several options affecting the process may be selected. Some of them are identical to those used in the **Find** service (see **Direction** and **Origin**). Use the <SHIFT+F8> key combination to repeat the last replacement command. All settings adjusted in the last replacement window, remain in force.



### String replacement -

Write a string to be found in the **Find string** box, and a string to replace the former one in the **Replace with** box. Acknowledge and start retrieving by operating the <Enter> key. The set-up configuration is saved as current. Depress <Esc> to stop this operation.

**Options** provides you with the following options: if the program finds a string it asks you whether the string should be replaced. If you select **First expression**, the operation will be completed after the first expression found. If **All exp. with ask** is selected, all strings complying with the specifications in **Direction** and **Origin** will be replaced. If **All expression** is selected all expressions will be replaced without any poll for acknowledgment. The task completion is indicated with an audio signal and the information about the number of replaced strings is displayed.

### Buffer / Fill block

The command makes it possible to fill the buffer part, defined by the start and end addresses (**Block start - End**), with a specified string. The string may be entered in





the ASCII or HEXA form. The ASC and HEX options are selected by cursor keys <↑>, <↓>. The maximum string length is 16 ASCII characters.

### Buffer / Copy block

This command copies the specified buffer block (**Block start - End**) to the selected address (**Target address**). The target area may overlap with the source one.

```
COPY BLOCK
BLOCK START : 000000
                END : 00FFFF
TARGET ADDRESS : 000000
```

### Buffer / Move block

This command moves the source buffer block (**Block start - End**) to the target address (**Target address**). The blocks may overlap. When moved the source block (or a part of it) is filled it with the current erasing constant (0FFH for EPROM).

```
MOVE BLOCK
BLOCK START : 000000
                END : 00FFFF
TARGET ADDRESS : 000000
```

### Buffer / Erase buffer <Ctrl+F2>

The buffer content is filled with the current erasing constant (implicit values are, for example, 0FFH for EPROM, 00H for some processor types, etc.). The constant value may be changed by keying-in the data in the hexa form.

### Buffer / Checksum

When the buffer area addresses are entered in the hexadecimal form, the checksum is calculated as follows:

```
CHECKSUM
From address : 000000
To address : 00FFFF
```

	SUM	NEG.	SUPPL.
BYTE	: B2B7	4D48	4D49
WORD	: 5E49	A1B6	A1B7
BYTE (CY)	: B336	4CC9	4CCA
WORD (CY)	: 9E4B	61B4	61B5

- **BYTE** means adding up to "word" by bytes. The CY attribute (transfer) is ignored.
- **WORD** means adding up to "word" by words. The CY attribute is ignored.
- **BYTE(CY)** means adding up to "word" by bytes. The CY attribute is added up to the result.
- **WORD(CY)** means adding up to "word" by words. The CY attribute is added up to the result.

The **Neg.** Column represents the result negation so that  $SUM + NEG. = FFFFH$ . The **Suppl.** Column gives the sum complement so that  $SUM + SUPPL. = 0$  (+ transfer).

## Options

The menu offers the commands enabling you to view and change various program defaults. The default values are saved in the SIM012.CFG file by the **Options / Save Options** command. Implicit default values for the individual items are marked as default.



### Options / Buffer name

This option is used for entering the full name of the buffer (i.e., including the path) if you wish to create it on the disk (the disk buffer is created automatically, when device size is larger than is available in the main memory). The entry is not checked for correctness. If the name entered is incorrect the error manifests itself as late as the attempt to create a buffer on the disk. The buffer default name is SIM012.\$\$.

When the buffer name is entered all editing keys, including the **Insert / Overwrite** mode selection key, are active. The current mode is indicated by the cursor shape (thick cursor - Insert, thin cursor - overwrite), with the default mode being **Insert**. The string size is 80 characters. When the window border is crossed, the string is automatically shifted.

### Options / Monitor

It adjusts colours for the EGA/VGA monitor type (i.e. monochrome, colour, or LCD). Any colour change is visible immediately after acknowledgement. The default is COLOR.

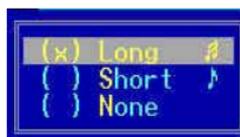


- **Color** The command "turns on" the colours for an **EGA/VGA** colour monitor. If this switch is used for a monochrome EGA/VGA monitor, some colour settings are not visible.
- **Mono** The command sets up the visible shades of grey for a **monochrome EGA/VGA** monitor.
- **LCD** This option is designed for the **NOTEBOOK** computer type with a **monochrome LCD** display.



## Options / Sound

It adjusts the volume of sound signals accompanying the display of some text information. The following options are available:



- **Long** An intensive sound signal accompanying the display of text information (default).
- **Short** A standard sound signal accompanying the display of text information.
- **None** Displaying text information without any accompanying sound signals.

## Options / Hex load

This command is used for setting up an alternative loading control in the Intel HEX format:



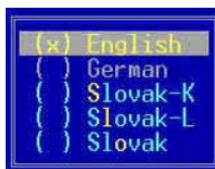
- Setting up the automatic buffer erasing prior to loading, by means of the specified erasing constant (0FFH for EPROM). Default - No.
- Setting up the so-called **interactive** method of loading, aimed at redirecting the file data to a physically existing buffer addresses. It means that, prior to loading, a user can enter a **segment** to be assigned to the lowest file segment (i.e. 02 type entry), and all other segments will be modified according to this difference. Default - No.

### Example:

*A file contains two entries of the type 02 with the F000H and F800H addresses. When a new segment, for example, 0H, is entered the data specified for the F000H segment will be directed to the 0H segment addresses, and, analogically, from the F800H segment to the 800H segment. This setting is meaningless for the simple Intel HEX format because there is no 02 type entry for this format. (A simple HEXa format range is 0...FFFFH).*

## Options / Help

This option activates the integrated Help system in the language you select. In addition to the **English** and **German** Help, there is also a Slovak version available in the **Kamenický format**,



**Latin 2** (page 852), or **without punctuation**. The Help

language can be changed anytime during the program run and the current setting can be saved in the CFG file using the **Options / Save Options** command. An introduction of a new **Help** system into the program is conditioned by the availability of the \*.HLP files on the disk. These files are available on the distribution CD along with the control program.

#### **English**

This option replaces the current integrated Help system with that in the **English** language under the condition that the required .HLP file is available on your disk.

#### **German**

This option replaces the current integrated Help system with that in the **German** language under the condition that the required .HLP file is available on your disk.

#### **Slovak - K**

This option replaces the current integrated Help system with that in the **Slovak** language, with punctuation in the **Kamenický** code under the condition that the required .HLP file is available on your disk.

#### **Slovak - L**

This option replaces the current integrated Help system with that in the **Slovak** language, with punctuation in the **Latin 2** (page 852) code under the condition that the required .HLP file is available on your disk.

#### **Slovak**

This option replaces the current integrated Help system with that in the **Slovak** language **without punctuation** under the condition that the required .HLP file is available on your disk.

### ***Options / Set masks***

This command is designed for setting up masks used for filtering the displayed file list in the save window (**Save**) for each format separately. A mask must contain at least one of the used pseudo-characters (\*, ?) and its syntax must be correct; otherwise, it will not be accepted. This setting is saved in the CFG file by the **Options / Save options** command.

Use selectional keys to select the mask to be modified. Acknowledgement by **<Enter>** provides you with the editing mode. At the same time, the **Edit** window is activated. The individual window operations are indicated by their respective box colours. All editing keys, including the **Insert / Overwrite** mode selection, are available. The cursor shape indicates the current mode. The default mode being **Insert**. Enter a new mask and be sure that its syntax is correct and that it contains at least one of the used pseudo-characters (\*, ?). Otherwise, when acknowledged by **<Enter>**, the mask will not be accepted. Pressing the



<Esc> key can stop the editing. The changes made will be ignored and you return to the mask selection mode. Use the <Esc> key to return to the main menu from the working window for mask set-up.

### Options / Save options

This command is used for saving the settings, selected in the menu, in the SIM012.CFG configuration file on the disk. In addition, the command enables you to select the location for creating the configuration file.



- **Current dir.** In the actual directory.
- **C:\ directory** In the root directory on the C: disk. This setting is advantageous for network configurations when the possibility of illegal access to shared server areas is eliminated.
- **SIM012.EXE dir.** In the directory from which the program was started.

When the SIM012.EXE is started the program starts searching for the configuration file in the current (i.e. working) directory (thus, it has the highest priority). If the file is not found, it continues with the root directory on disk C: and finally proceeds to the directory from which the program was started. If the configuration file is not found the program sets up the default values for all options.

### Options / Retrieve options

This command makes it possible to retrieve the settings executed in all **Options** menu items and save them on the disk by the **Save options** command. In addition, the command is used for selecting the location from which the configuration file is loaded (current directory, root directory on disk C: or the directory from which the program was started). If the program fails to find the file, it displays a warning, and continues with the original configuration.

### Quit <Alt+X>

This command is used to quit the program and return to the operation system. The following options are available:



- **No** Cancel the command for return to the operation system, and continuing with program run.

- **Yes** Quit the program, deallocate heap, cancel the buffer on the disk (if any), and return the control to the operation system.
- **Yes & Save** Quit the program, deallocate heap, cancel the buffer on the disk (if any), and return the control to the operation system. In addition, the command saves the complete information about the previously used configuration and about the number of currently available HW simulators, underlying the specific configuration, in the SIM012.SET file.

## Info

This command generates a window, including the basic information about the program version, "copyright", and the manufacturer's address.

```
          ABOUT
S I M E P R O M  -  01 / 02
(E)EPROM/PROM simulator
Version 4.15/11.2000
Copyright (c) by ELNEC

Customer support:
ELNEC s.r.o., Post 5, P.O.Box 22
SK-080 05 Presov
tel.: +421-91-7734328, 7731007
fax:  +421-91-7732797
e-mail: elnec@elnec.com
WWW:  http://www.elnec.com
```



---

## Using the SIM012W control program

### Starting the program

After starting the SIM012W.EXE control program is checked for its integrity and presence of SIMEPROM.

When the SIM012W program is started it automatically finds the connected simulator(s) at all existing LPT and COM ports (the COM ports are searched only in the case of SIMEPROM-02aLV). This mode can be suppressed by specifying a parameter, including the LPT or COM name, to which the simulator is connected. In this case, the control program is searches for the simulator only at the desired port. The parameter form is /LPTx or /COMx, where x is the ordinary number of the particular LPT or COM. If this parameter is used more than once, each time with a different LPT or COM address, the program will be provided with a list of ports at which the simulator(s) can be found. This list is also accepted by the **Device / Configuration** service during the program run.

An example of starting the program with parameters:

```
SIM012W /LPT1 /COM2
```

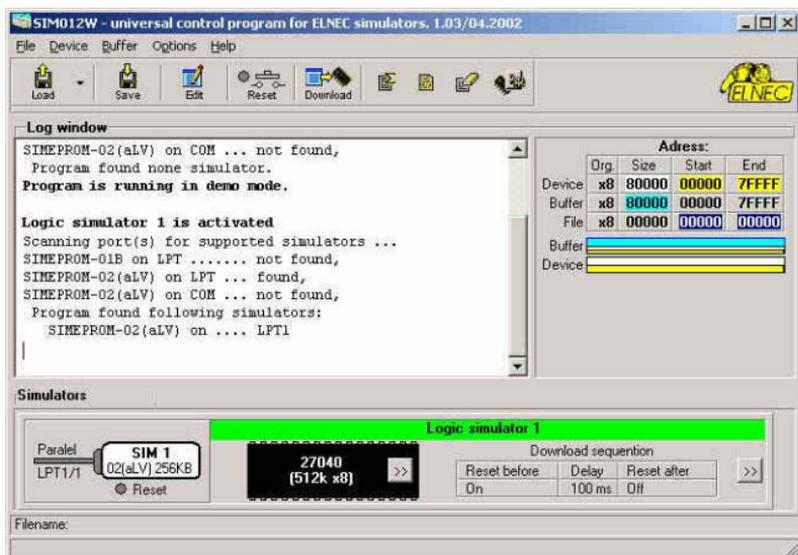
If everything is O.K., the program provides a standard user screen and waits for your commands.

If the control program cannot communicate with SIMEPROM it gives an error message, including a description of possible reasons (simulator disconnected, or connector not inserted properly, incompatible printer port or serial port, or printer port set up as a bidirectional port...). When the error is eliminated press any key. If you fail to eliminate the cause of error, the program resumes running in the **Demo** mode after depressing any key. By implication, the simulator download is not possible. If you cannot identify the error cause contact the simulator manufacturer.

### Note:

*SIMEPROM-02aLV must be powered up before starting the control program.*

## User screen



- **Header bar** name, copyright, and control program version
- **Menu** offer of control program options
- **Simulator window** information about the presence of HW simulators at the ports and current configuration and highlighting the currently active logical simulators. By clicking on icon >> you may change configuration of logical simulators.
- **Log window** information about the simulator and control program status
- **Address window** organization, size, start and end addresses of the device, buffer and file
- **Help** a brief description of a selected command

Menu selection is carried out in the normal GUI fashion - either by cursor moving plus pressing **<Enter>**, or by typing the highlighted letter in the wanted menu or - of course - by mouse. Hot-keys are available for even quicker selection of intensely used commands.

**Note:** Data entered through keyboard are in HEX format, excepting ASCII blocks in Buffer/View/Edit command.



## List of hot keys

<F1>	Help	Calls the Help
<F2>	Save	Save a file
<F3>	Load	Load a file into the buffer
<F4>	Edit	Viewing/editing buffer
<F5>	Config	Adjusting the configuration
<F7>	RESET ON	Activation of RESET signal in active LS
<F8>	RESET OFF	Deactivation of RESET signal in active LS
<F9>	Download	Downloading to active LS

## File

This submenu is used for source files manipulation, settings and viewing directory, changes drives, changes start and finish address of buffer for loading and saving files by **binary, MOTOROLA, MOS Technology, Intel (extended) HEX, Tektronix, ASCII space and Exormax**.

### File / Load

Analyse file format and loads the data from specified file to the buffer. You can choose the format desired (**binary, MOTOROLA, MOS Technology, Tektronix, Intel (Extended) HEX, ASCII space and Exormax**). The control program stores a last valid mask for file listing. You can save the mask into the configuration file by command **Options / Save options**.

Checking the check box **Automatic file format recognition** tells program to detect file format automatically. When program can't detect file format from one of supported formats, the binary file format is assumed.

When the check box **Automatic file format recognition** is unchecked program allows user to manually select wished file format from list of available file formats on panel **Selected file format**. When Binary file format is selected, there can be specified Buffer start value. **Buffer start** value is buffer address from which data read from file will be written to buffer.

### File / Save

Saves data in the buffer, which has been created, modified onto a specified disk. The file format of saved file can be chosen from supported formats list box. There can be also entered the Buffer start and Buffer end addresses which exactly specify part of buffer to save to file. Supported file formats now are **binary, MOTOROLA, MOS Technology, Tektronix, Intel (Extended) HEX, ASCII space and Exormax**.

### File / Reload

Choose this option to reload a recently used file.

When you use a file, it is added to the Reload file list. Files are listed in order depending on time of use of them. Lastly used files are listed before files used far off.

To Reload a file:

1. From the **File** menu, choose **Reload**.
2. List of lastly used files is displayed. Click the file you want to reload.

**Note:**

*When reloading a file the file format is used, by which the file was lastly loaded/saved.*

### **File / Exit without save**

The command deallocates heap, cancels buffer on disk (if exists) and returns back to the operation system.

## **Device**

### **Device / Configuration**

This command is used to assign hardware simulators to logical simulators. By selecting this command, window **Configuration of logical simulators**, which is divided into 3 columns, is displayed.

At left column are displayed unassigned hardware simulators. By check box **Present** you may enable or disable selected hardware simulator.

At middle column you must select type and number of logical simulators first. Then, by drag&drop, assign present hardware simulators to logical simulators.

At right column are two buttons. Button **Save configuration and reset simulators** use for confirm your desired configuration. When you don't change configuration click on button **Ignore changes**, while is accessible. When is not accessible, you already execute changes, which can't be ignored.

### **Device / Reset ON**

This command executes RESET in the development system, if a reset cable is connected to a reset capacitor or to a RESET button in the development system. This operation is equivalent to keeping the RESET button depressed. The command is applied to the currently active logical simulator (LS). The current status of the RESET signal is displayed in the information window reserved for the particular LS.

### **Device / Reset OFF**

This command releases the RESET in the development system, if it is in the RESET state. This operation is equivalent to releasing the resetting button. The command is applied to the currently active logical simulator (LS). The current status of the RESET signal is displayed in the information window reserved for the particular LS.



## **Device / Download**

This command transfers data from the buffer to the currently active logical simulator (LS). In addition to data transfer to the simulator, the individual operations take place in accordance with the adjusted downloading sequence, for example, **Reset ON** before downloading and **Reset OFF** after downloading.

## **Device / (LOG. SIM #1 - #4)**

This item activates the corresponding logical simulator. Hence, the **Reset ON**, **Reset OFF** a **Download** command are aimed at this active LS. Currently active logical simulator is highlighted. Nonactive logical simulators are greyed.

The switching of LS can be accelerated by using key combination **<Alt>** + LS ordinal number, or by mouse.

## **Buffer**

This submenu is used for buffer manipulation, block operation, filling a part of buffer with string, erasing, checksum and of course editing and viewing with other items (find and replace string, printing...).

## **Buffer / View/Edit**

This command is used to view (view mode) or edit (edit mode) data in buffer (for viewing in DUMP mode only). Use arrow keys for select the object for edit. Edited data are signified by colour.

You can use **<F4>** hot key also.

### **View/Edit Buffer**

- |                      |   |
|----------------------|---|
| <b>F1</b>            | display help of actual window   |
| <b>F2</b>            | fill block causes filling selected block of buffer by requested hex (or ASCII) string. Sets start and end block for filling and requested hex or ASCII string.  |
| <b>Ctrl+F2</b>       | erase buffer with specified blank value   |
| <b>Ctrl+Shift+F2</b> | fill buffer with random data  |
| <b>F3</b>            | copy block is used to copy specified block of data in current buffer on new address. Target address needn't be out from source block addresses.   |
| <b>F4</b>            | move block is used to move specified block of data in current buffer on new address. Target address needn't be out from source block addresses. Source address block (or part) will be filled by topical blank character. |
| <b>F5</b>            | swap bytes command swaps a high- and low- order of byte pairs in current buffer block. This block must started on even address and must have an even number   |

---

	of bytes. If this conditions do not fulfil, the program modifies addresses itself (start address is moved on lower even address and/or end address is moved on higher odd address).
<b>F6</b>	print buffer
<b>F7</b>	find string (max. length 16 ASCII characters)
<b>F8</b>	find and replace string (max. length 16 ASCII characters.)
<b>F9</b>	change current address
<b>F10</b>	change mode view / edit
<b>F11</b>	switch the mode of buffer data view between 8 bit and 16 bit view. It can be also do by mouse clicking on the button to the right of View/Edit mode buffer indicator. This button indicates actual data view mode (8 bit or 16 bit), too.
<b>F12</b>	checksum dialog allows to count checksum of selected block of buffer change mode view / edit
<b>Arrow keys</b>	move cursor up, down, right and left
<b>Home/End</b>	jump on start / end current line
<b>PgUp/PgDn</b>	jump on previous / next page
<b>Ctrl+PgUp/PgDn</b>	jump on start / end current page
<b>Ctrl+Home/End</b>	jump on start / end current device
<b>Backspace</b>	move cursor one position left (back)

**Note:** characters 20H - FFH (mode ASCII) and numbers 0..9, A..F (mode HEX) immediately changes content of edit area.

### **Print buffer**

This command allows write selected part of buffer to printer or to file. Program uses at it an external text editor in which selected block of buffer is displayed and can be printed or saved to file, too. By default is set simple text editor **Notepad.exe**, which is standard part of all versions of MS Windows.

In Print buffer dialog are following options:

#### **Block start**

Defines start address of selected block in buffer.

#### **Block end**

Defines end address of selected block in buffer.

#### **External editor**

Defines path and name of external program, which has to be used as text viewer for selected block of buffer. By default is set simple text editor **Notepad.exe**, which is standard part of all versions of MS Windows. User can



define any text editor for example **Wordpad.exe**, which is able to work with large text files. In user defined text editor user can print or save to file selected block of buffer.

The external editor path and name is saved automatically to disk.

### **Find dialog box**

Enter the search string to **Find** to text input box and choose **<Find>** to begin the search or choose **<Cancel>** to forget it.

**Direction** box specifies which way you want to search, starting from the current cursor position (In edit mode).

**Forward** (from the current position or start of buffer to the end of the buffer) is the default. **Backward** searches toward the beginning. In view mode searches all buffer.

**Origin** specifies where the search should start.

### **Find & Replace dialog box**

Enter the search string in the **Text to find** string input box and enter the replacement string in the **Replace with** input box.

In **Options** box you can select prompt on replace: if program finds instance you will be asked before program change it.

**Origin** specifies where the search should start.

**Direction** box specifies which way you want to search, starting from the current cursor position (In edit mode).

**Forward** (from the current position or start of buffer to the end of the buffer) is the default. **Backward** searches toward the beginning. In view mode searches all buffer.

Press **<Esc>** or click **Cancel** button to close dialog window.

By pressing **Replace** button the dialog box is closed and a Question window is displayed. This window contains following choices:

- |                     |   |
|---------------------|---|
| <b>Yes</b>          | replaces found item and finds next            |
| <b>No</b>           | finds next item without replacing current one |
| <b>Replace All</b>  | replaces all found items                      |
| <b>Abort search</b> | aborts this command                           |

### **Buffer / Fill block**

Selecting this command causes filling selected block of buffer by requested hex (or ASCII) string. Sets start and end block for filling and requested hex or ASCII string.

### **Buffer / Copy block**

This command is used to copy specified block of data in current buffer on new address. Target address needn't be out from source block addresses.

### **Buffer / Move block**

This command is used to move specified block of data in current buffer on new address. Target address needn't be out from source block addresses. Source address block (or part) will be filled by topical blank character.

### **Buffer / Swap block**

This command swaps a high- and low- order of byte pairs in current buffer block. This block must start on even address and must have an even number of bytes. If these conditions do not fulfil, the program modifies addresses itself (start address is moved on lower even address and/or end address is moved on higher odd address).

### **Buffer / Erase**

If this command is selected, the content of the buffer will be filled with topical blank character.

The reserved key **<Ctrl+F2>** will bring out this menu from any menu and any time.

### **Buffer / Fill random data**

If this command is selected, the content of the buffer will be filled with random data.

The reserved key **<Shift+Ctrl+F2>** will bring out this menu from any menu and any time.

### **Buffer / Checksum**

The checksum dialog is used for calculate checksums of selected block in buffer. The checksums are calculated by next way :

<b>Byte</b>	sum by bytes to "word". CY flag is ignored
<b>Word</b>	sum by words to "word". CY flag is ignored
<b>Byte (CY)</b>	sum by bytes to "word". CY flag is added to result.
<b>Word (CY)</b>	sum by words to "word". CY flag is added to result.
<b>CRC-CCITT</b>	sum by bytes to "word" using $RESULT=PREVIOUS + (x^{16} + x^{12} + x^5 + 1)$
<b>CRC-XModem</b>	sum by bytes to "word" using $RESULT=PREVIOUS + (x^{16} + x^{15} + x^2 + 1)$



Column marked as **Neg.** is a negation of checksum so, that  $\text{Sum} + \text{Neg.} = \text{FFFFH}$ .

Column marked as **Suppl.** is complement of checksum so, that  $\text{Sum} + \text{Suppl.} = 0$  (+ carry).

Dialog checksum contains following items:

**From address:** This is a start address of block selected for calculating checksums in buffer. Address is defined as Byte address.

**To address:** This is an end address of block selected for calculating checksums in buffer. Address is defined as Byte address.

**Insert checksum:** This is special item used for select which kind of checksum will be written into the buffer when, the **Calculate & insert** was executed.

**Insert address:** This is special item that specifies an address from the buffer where a result of chosen checksum will be written, when the **Calculate & insert** was executed. Address cannot be specified inside the range **<From address>** to **<To address>**. Address is defined as Byte address.

**Size:** This item is used for setting a size of chosen checksum result, which will be written into the buffer. A size of checksum result may be 8 (byte) or 16 (word) bits long. If word size was selected, whole checksum value will be written into the buffer. In other case will be written only low byte of checksum value.

**Note:** *If word size was selected, a low byte of checksum value will be written on address specified in box Insert address and a high byte will be written on address incremented by one.*

**Calculate:** Click on the button Calculate starts calculating checksums for selected block in buffer. No writes into the buffer are executed.

**Calculate & insert:** Click on the button **Calculate & insert** starts calculating checksums for selected block in the buffer and writes the chosen checksum into the buffer on address specified by **Insert address**.

## Options

The Options menu contains commands that let you view and change various default settings.

### Options / General options

General options dialog allows user to control following options of program.

### **File options**

File options page allows you to set file masks, auto-reload of current file and choose file format recognizing for loaded files.

**File format masks** is used for setting file-name masks to use as a filter for file listing in **File / Save** and **File / Load** file window for all file formats. Mask must contain one of wildcards (\*, ?) at least and must be applied correctly by syntax.

In group **When current file is modified by another process** can be set mode of reloading of actually loaded (current) file. There are three choices:

1. Prompt before reloading file
2. Reload automatically
3. Ignore change scanning of current file

**Load file format** allows to set mode of file format recognition for loading files. When automatic file format is selected, program analyses format of loading file and test file for each of supported formats, which are available in program. If file format matches one of supported formats, the file is read to buffer in detected format.

Manual file format allows user to select explicitly wished file format from list of supported file formats. File may be loaded no completely or incorrectly, if file format does not match to user selected format.

### **Hex file options**

This page contains several options for loading control by any of HEX formats.

The first option sets **erasing** buffer (with desired value) automatically before the loading by any of HEX formats.

The second option sets a **negative offset**, which is used for data addresses modification by loading from any HEX file so, that data can be written to existing buffer addresses.

#### **Example:**

A file contents data by Motorola S - format. A data block started at address FFFF0H. It is a S2 format with length of address array of 3 bytes. For all data reading you can set a value of negative offset to FFFF0H. It means, that the offset will be subtracted from current real addresses and so data will be written from buffer address 0.

**Warning:** *The value of negative offset is subtracted from real address and therefore a result of subtraction can be negative number. Because take care of correct setting of this value.*



The third option sets data **redirecting** from file by extended Intel HEX format to physically existing buffer addresses. It means, that the user will be prompted to enter segment, which is associated with the lowest segment in file (record type 02) and the other segments will be modified with this difference. This option isn't valid for "simple" Intel HEX format. Default set means inactive status.

**Example:**

The file contains two records of type 02 with addresses F000H and F800H. When we enter new segment for example 0H, data from segment F000H will be directed to segment 0H and similar data from segment F800H to segment 800H.

Default setting is no active.

**Language**

This page allows you to select another language for user interface such as menu, buttons, dialogs, information and messages. It also allows to select wished help file in another language. For another language support of user interface the language definition file is required.

**Sound**

Sound page allows user to select the sound mode of program. Program generates sounds after some activities with simulator. Program generates sound also when warning or error message is displayed. User can now select sound from MS Windows system sound (required installed sound card), PC speaker or none sound.

**Other**

Page Other allows user to manage other program settings. In the panel Tool buttons, hint display options on toolbar buttons in main program window can be modified. In the panel Start-up directory can be selected mode of selecting directory when program starts. Default start-up directory means directory, from which program is called. Directory in which program was lastly ended means the last current directory when program was lastly ended. This directory assumes the first directory from directory history list.

**Save options**

Save options page allows you to select the program options saving when exiting program. Three options are available here:

**Don't save options** - don't save options during quitting program and don't ask for saving options.

**Auto save options** - save options during quitting program without asking for saving options.

**Prompt for save options** - program asks user for saving options before quitting program. User can select to save or not to save options.

## **Options / View**

Use the View menu commands to display or hide different elements of program environment such as toolbars.

### **Options / View / Main toolbar**

Choose this command to show or hide the Main toolbar.

## **Options / Log file**

This options associates with using of **Log window**. All reports for this window can be written into the **Log file** too. The Log file name as **REPORT.REP** and the control program creates this file in current directory.

Sets **New** caused deleting old Log file if exist and creating a new file for reports. Sets **Append** adds all reports into existing Log file. If file not exist, the new file will be created. Settings are applied only at program start.

This setting can be saved to disk by command **Options / Save options**. Default form is set to a using Log window without Log file i.e. all reports will be displayed to a Log window only.

## **Options / Save options**

This command saves all settings that are currently supported for saving, even if auto-save is turned off. Following options are saved: options under the Options menu, ten last selected devices, file history, main program window position and size.

## **Help**

Pressing the <F1> key accesses the Help. When you selecting menu item and press <F1>, you access context-sensitive help. If SIM012 / SIM012W is executing an operation with the simulator <F1> generates no response.

The following Help items are highlighted:

- words describing the keys referred to by the current Help
- all other significant words
- current cross-references; click on this cross-reference to obtain further information.

*Since the Help system is continuously updated together with the control program, it may contain information not included in this manual.*

Detailed information on individual menu commands can be found in the integrated on-line Help.

**Note:** *Information provided in this manual is intended to be accurate at the moment of release, but we continuously*



*improve all our products. Please consult manual on [www.elnec.com](http://www.elnec.com).*

## **About**

When you choose the Info command from the menu, a window appears, showing copyright and version information.

## Using the SIM01BR control program

The SIM01BR program is designed for the SIMEPROM-01B. Depending on the parameters of the start it may be used in two versions:

- TSR (resident) version
- Version for BATCH files.

### TSR version

Syntax for applying the SIM01BR as a resident program:

**SIM01BR /R [/D] [/M]**

**/R** - a request for installation (the request is accepted if the SIMEPROM is available)

**/D** - a start in the DEMO mode - no SIMEPROM is available (default port: LPT2)

**/M** - setting suitable colors for the monochrome VGA monitor

Resident program for SIMEPROM-01B supports simulator download from file by three file formats: binary, Intel HEX and Motorola S-format. The file format autodetection is only way for set of format type, if the control program runs as resident - during download is current format indicated by character B (binary), I (Intel HEX) or M (Motorola) on first line of program window by blinking attribute. The program erases SIMEPROM before all HEX download automatically. You can suppress this action in BATCH version by using of **/NOERASE** parameter.

After the start, activate the program by depressing the following combination of keys: **<Ctrl+Alt+BackSpace>**. The following window appears on the display:

```
SIMEPROM-01B 4.15R/11.2000 (c) ELNEC
  2 x (64Kx8 bit) SIMEPROM-#1
EPROM TYP : 27512 RESET (") PASSIVE
FILE :
F1-HELP F3-(HEX)DOWNLOAD F5-EPROM TYPE
F7-ACTIVE RESET F8-PASSIVE RESET
<ESC>-EXIT
```

Select the individual program functions by means of the following function keys:

**<F1>** - brief HELP

**<F5>** - EPROM type selection. This function enables you to select the required EPROM type by means of **<↑>** **<↓>** keys. Then, depress **<Enter>**. When using several



- simulators this function applies to the selected simulator only (<F9>).
- <F7> - RESET signal activation. The active state is indicated by ACTIVE in the RESET line. When using several simulators this function applies to the selected simulator only (<F9>).
  - <F8> - RESET signal deactivation. The inactive state is indicated by PASSIVE in the RESET line. When using several simulators this function applies to the selected simulator only (<F9>).
  - <F9> - Switching over to an active simulator; this function is used when working with several simulators. Select the required simulator (Byte configuration), or a couple of simulators (WORD configuration) by consecutively depressing the <F9>. If only one simulator is connected, the setting is on SIMEPROM#1.
  - <F10> Switching between the BYTE and WORD configurations. The selected configuration type is displayed in the top line. This function is designed for operating several simulators.
  - <F4> - Setting/change of the file default menu for download. When using several simulators this function applies to the selected simulator only (<F9>).
  - <F3> - Download. This function activates the cursor in the highlighted FILE line: Acknowledge the file name by <Enter>, or key in the file name you wish to load into the simulator, including possible paths, if the file does not reside in the present directory. Depress <Enter> to execute download; the latter is indicated by displaying the line being loaded. The termination is indicated by an acoustic signal (if switched ON). The file data loaded into the simulator are read from the first address. If the size of a data file exceeds that of the simulated EPROM, excessive data are ignored. If a file is smaller, missing data will be replaced by the 00H value.
  - <F6> - Acoustic indication ON/OFF. The ON state is indicated in the upper left corner by "♪".
  - <F2> - Storing the set program parameters. When the program is restarted, the required parameters needn't be readjusted.

## **BATCH Version**

Syntax for applying the SIM01BR in the batch file:

### **SIM01BR [options] [filename]**

**filename** - The file name whose contents is to be sent to the simulator. If the file has an .HEX extension, its contents is interpreted by the SIM01BR program as in the Intel HEX format; otherwise it is taken to be a binary file.



- /S<type>** - EPROM type, <type> =2716, 2732, 2764, 27128, 27256,27512
- /CBx** - BYTE configuration, x = 1,2,3,4 for the corresponding HW simulators
- /CWxy** - WORD configuration, xy = 21, 43 for the corresponding HW simulators, where x designates the high byte and y the low byte
- /B, /H** - Download format - binary/Intel HEX. This selection is necessary if, for a file in HEX, you use an extension different from the .HEX, or if you wish to download a file with the .HEX extension as a binary one.
- /NOERASE** To suppress erase SIMEPROM-01B before HEX loading.
- /NOMSG** - To suppress all messages writing without error reports in BATCH version for SIMEPROM-01B.
- /Rx, /Nx** - RESET signal activation/deactivation; x = HW simulator number(s) (e.g. /R1 for RESET HW #1, /N123 for NORESET HW #1, #2, and #3).
- /P<seq>** - download sequence, <seq> -  
\*L. = Reset ON, Download, Reset OFF  
\*L = Reset ON, Download
- default = /S27512 /CB1 /PL

The program erases SIMEPROM before all HEX download automatically. You can suppress this action in BATCH version by using of /NOERASE parameter. Use this parameter, if you wish to download simulator from two or more HEX files. We recommend the following sequence:

```
SIM01BR /R1
SIM01BR file1.hex
SIM01BR file2.hex /NOERASE
.
.
.
SIM01BR /N1
```

The SIM01BR control program version 2.00 can be only used with a logical simulator in the BYTE or the WORD configuration, with constraints imposed on the assignment of individual HW simulators to logical set-ups of the WORD type:

- LS#1 can be set up by HW#1 and HW#2 only
- LS#2 can be set up by HW#3 and HW#4 only.

The SIM01BR control program does not transform segments for an extended HEX format of a file. The only way to calculate the address for download is as follows: (segment shl 4)+offset.



## Using the SIM02R control program

The SIM02R program is designed for the SIMEPROM-02aLV. Depending on start parameters it can be used in two versions:

- TSR (resident) version
- Version for BATCH files.

### TSR version

Syntax for applying the SIM02R as a resident program:

#### SIM02R /R [/D] [/M]

- /R** - a request for installation (the request is accepted if the SIMEPROM is available)
- /D** - a start in the DEMO mode - no SIMEPROM is available (default port: LPT2)
- /M** - setting suitable colors for the monochrome VGA monitor

Resident program for SIMEPROM-02aLV supports simulator download from file by three file formats: binary, Intel HEX and Motorola S-format. The file format autodetection is only way for set of format type, if the control program runs as resident - during download is current format indicated by character B (binary), I (Intel HEX) or M (Motorola) on first line of program window by blinking attribute. The program erases SIMEPROM before all HEX download automatically. You can suppress this action in BATCH version by using of **/NOERASE** parameter.

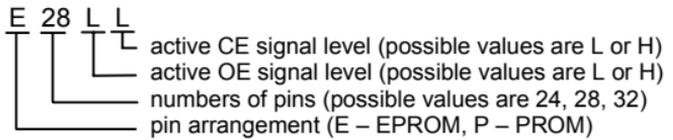
After the start, activate the program by depressing the following combination of keys: **<Ctrl+Alt+BackSpace>**. The following window appears on the display:



Select the individual program functions by means of the following function keys:

- <F1>** - brief HELP
- <F5>** - EPROM type selection, and its description in terms of the arrangement pins, the number of pins, and the active level of OE\ and CE\ signals. Use keys **<↑>**

<↓><↔>, and <⇒> to select the required EPROM type. In addition, you can describe it by the above mentioned attributes. The <F1> key will provide you an updated HELP in the following form:



- Acknowledge the setting by depressing the **<Enter>** key. When using several simulators this function applies to the selected simulator only (**<F9>**).
- <F7>** - RESET signal activation. The active state is indicated by ACTIVE in the RESET line. When using several simulators this function applies to the selected simulator only (**<F9>**).
  - <F8>** - RESET signal deactivation. The inactive state is indicated by PASSIVE in the RESET line. When using several simulators this function applies to the selected simulator only (**<F9>**).
  - <F9>** - Switching over to an active simulator; this function is used when using several simulators. Select the required simulator (Byte configuration), or a couple of simulators (WORD configuration) by consecutively depressing the **<F9>**. If only one simulator is connected, the setting is on SIMEPROM#1.
  - <F10>** - Switching between the BYTE and the WORD configurations. The selected configuration type is displayed in the top line. This function is designed for operating several simulators.
  - <F4>** - Setting/change of the file default menu for download. When using several simulators this function applies to the selected simulator only (**<F9>**).
  - <F3>** - Download. This function activates the cursor in the highlighted FILE line: Acknowledge the file name by **<Enter>**, or key in the file name you wish to load into the simulator, including possible paths, if the file does not reside in the present directory. Depress **<Enter>** to execute download; the latter is indicated by displaying the line being loaded. The termination is indicated by an acoustic signal (if switched ON). The file data loaded into the simulator are read from the first address. If the size of a data file exceeds that of the simulated EPROM, excessive data are ignored. If a file is smaller, missing data will be replaced by the 00H value.
  - <F6>** - Acoustic indication ON/OFF. The ON state is indicated in the upper left corner by "♪".



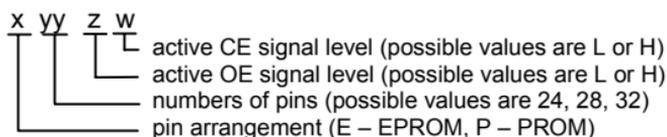
- <F2>** - Storing the set program parameters. When the program is restarted the required parameters needn't be readjusted.

## BATCH Version

Syntax for applying the SIM02R in the batch file:

### SIM02R [options] [filename]

- filename** - The file name whose contents is to be sent to the simulator. If the file has an .HEX extension, its contents is interpreted by the SIM02BR program as in the Intel HEX format; otherwise it is taken to be a binary file.
- /S<type>** - EPROM type, <type> = 2716, 2732, 2764, 27128, 27256, 27512, 27010, 27020, (27040)
- /CBx** - BYTE configuration, x = 1,2,3,4 for the corresponding HW simulators
- /CWxy** - WORD configuration, xy = 21, 43 for the corresponding HW simulators, where x designates the high byte and y the low byte
- /B, /H** - Download format - binary/Intel HEX. This selection is necessary if, for a file in HEX, you use an extension different from the .HEX, or if you wish to download a file with the .HEX extension as a binary one.
- /NOERASE** To suppress erase SIMEPROM-02aLV before HEX loading.
- /NOMSG** - To suppress all messages writing without error reports in BATCH version for SIMEPROM-02aLV.
- /Rx, /Nx** - RESET signal activation/deactivation; x = HW simulator number(s) (e.g. /R1 for RESET HW #1, /N123 for NORESET HW #1, #2, and #3).
- /P<seq>** - Download sequence, <seq> -  
\*L. = Reset ON, Download, Reset OFF  
L. = Download, Reset OFF  
\*L = Reset ON, Download  
L = Download
- /Txyyzw** - (E)PROM descriptor



default = /S27512 /CB1 /PL /TE28LL

The program erases SIMEPROM before all HEX download automatically. You can suppress this action in BATCH version by using of **/NOERASE** parameter. Use this parameter, if you



wish to download simulator from two or more HEX files. We recommend the following sequence:

```
SIM02R /R1
SIM02R file1.hex
SIM02R file2.hex /NOERASE
.
.
.
SIM02R /N1
```

The SIM02R control program version 2.00 can be only used with a logical simulator in the BYTE or the WORD configuration, with constraints imposed on the assignment of individual HW simulators to logical set-ups of the WORD type:

- LS#1 can be set up by HW#1 and HW#2 only
- LS#2 can be set up by HW#3 and HW#4 only.

The SIM02R control program does not transform segments for an extended HEX format of a file. The only way to calculate the address for download is as follows: (segment shl 4)+offset.





---

# *Specification*

---



# **SIMEPROM-01B specifications**

## **Supported devices**

- EPROM/PROM: NMOS/CMOS, family 27xxx and 27Cxxx from 2KByte (2716) to 64Kbyte (27512) with 8 bit data bus
- EEPROM: NMOS/CMOS, family 28xxx, 28Cxxx, 27EExxx from 2KByte (2816) to 32KByte (28256) with 8 bit data bus
- EPROM/PROM/EEPROM from 64KByte to 128KByte with 16 bit data buss (2x SIMEPROM-01B with converter DIL40/DIL2x32 W-EPROM)

## **Compatibility**

- response on the address changed : max. 120 ns
- response on the CE\ signal going to L : max. 120 ns
- response on the OE\ signal going to L : max. 60 ns
- response on the either OE\ or CE\ going to H (data float): max. 10 ns
- input compatibility : 74HCT
- output compatibility: 74HC
- Low voltage EPROM support from 2.5V if powered from external power supply. This is valid for SIMEPROM-01B, serial number 29641 or above.

**Relationship between supply voltage of EPROM in target device and equivalent speed.**

Supply voltage	speed of Simeprom-01B
4.5V - 5.5V	120ns
3.5V - 4.5V	160ns
2.5V - 3.5V	200ns

## **SIMEPROM-01B operations**

- download
- RESET control in target system

## **Buffer operations**

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

## **Download time**

- EPROM type 27512, PC386/DX33 : 4 seconds

## ***Supported file formats***

- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-SPACE-HEX

## ***PC system requirements***

- Processor: IBM PC AT or above, free 512 KB RAM
- Hard disk with 2 MB free space
- Operating system: DOS 3.2 or higher, Windows 3.x/95/98/Me/NT/2000/XP
- Free parallel (printer) port for connection of simulator

## ***General***

- operating voltage: 7..15V DC, max.100mA or 5V +/-10% from target system
- power consumption: 80mA max.
- RESET forcing circuit: max. 20V, max. 100mA
- dimensions: 132x66x30 [mm]
- weight: about 100g
- environment: office/laboratory
- temperature: 5 to 40°C
- humidity 20%..80%, non condensing

## ***Package included***

- SIMEPROM-01B simulator
- connection cable to PC
- user manual
- CD with software
- registration card
- transport case

## ***Simulator's price included too:***

- free technical support (hot line)
- free lifetime software update via Web site



# SIMEPROM-02aLV specifications

## Versions

- SIMEPROM-02aLV ver. 020/120, equivalent EPROM/EEPROM/PROM 27C020-120ns
- SIMEPROM-02aLV ver. 040/85, equivalent EPROM/EEPROM/PROM 27C040-85ns

## Supported devices

- EPROM/PROM: NMOS/CMOS, family 27xxx and 27Cxxx, from 2KByte (2716) to 512KByte (27040/274001) with 8 bit data bus in DIL or PLCC.
- EEPROM: NMOS/CMOS, family 28xxx, 28Cxxx, 27EExxx, from 2KByte (2816) to 512KByte (28040) with 8 bit data bus in DIL or PLCC.
- EPROM/PROM/EEPROM from 64KByte to 1MByte with 16 bit data bus in DIL or PLCC (2x SIMEPROM-02aLV with converter DIL40/DIL2x32 W-EPROM)

## Compatibility

- response on the address changed: max. 85 ns (max. 120 ns)
- response on the CE\ signal going to L: max. 85 ns (max. 120 ns)
- response on the OE\ signal going to L : max. 45 ns (max. 60 ns)
- response on the either OE\ or CE\ going to H (data float): max. 10 ns (max. 20 ns)
- inputs compatibility : 74F (74HCT)
- outputs : 74AC (74HC)
- low voltage EPROM support from 2.5V if powered from external power supply.

### Note:

*Values in parentheses are valid only for 020/120 version*

**Relationship between supply voltage of EPROM in target device and equivalent speed.**

Supply voltage	version 020/120	version 040/85
4.5V - 5.5V	120ns	85ns
3.5V - 4.5V	160ns	105ns
2.5V - 3.5V	200ns	125ns

## ***POD protection circuit parameters***

- all pins are protected against Electrostatic Discharge (ESD) and balancing current
- ESD protection rated to IEC1000-4-2 (15kV air, 8kV contact)
- protection against balancing current up to 2.5A

## ***SIMEPROM-02aLV operations***

- download
- RESET control in target system

## ***Buffer operations***

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

## ***Supported file formats***

- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-SPACE-HEX

## ***Download times***

- type 27020, parallel transfer, PC386/40MHz : 12 seconds
- type 27020, serial transfer, 115200 Bd, PC386/40MHz : 28 seconds

## ***PC system requirements***

- Processor: IBM PC AT or above, free 512 KB RAM
- CD drive, hard disk with 2 MB free space
- Operating system: DOS 3.2 or higher, Windows 3.x/95/98/Me/NT/2000/XP
- Free parallel (printer) port or free serial (RS-232) port

## ***General***

- operating voltage: 10..15V DC, max. 350mA or 5V  $\pm$ 10% from target system
- power consumption: max. 350 mA (max. 200 mA)
- RESET forcing circuit: max. 20V, max. 100mA
- galvanic isolation by serial line max. 250V DC
- dimensions: 132x66x30 [mm] without cables attached
- weight: about 150g
- environment: office/laboratory
- temperature: 5 to 40°C
- humidity 20%..80%, non condensing



**Note:**

*Values in parentheses are valid for 020/120 version only*

**Standard deliveries:**

- SIMEPROM-02aLV simulator
- SIMEPROM-02aLV, serial communication module (features optical isolation)
- connection cable to PC for connection through parallel port and serial port
- precise 32 pin socket, connected to simulator POD
- precise 28 pin socket
- precise 24 pin socket
- wall-plug adapter, 12V DC/500mA, unstabilized
- user manual
- CD with software
- registration card
- transport case

**Simulator's price included too**

- free technical support (hot line)
- free lifetime software update via Web site



---

# *Warranty*

---



The manufacturer gives a guarantee on failure-free operation of the device for the period of 36 months from the date of delivery as given in the delivery note. This guarantee bears on those SIMEPROM users who have registered with the manufacturer by filling and mailing a registration card (not later than four weeks after the date of purchase). The card is enclosed to the product sold through agents other than the ELNEC company. Provided you do not receive the registration card with your simulator, you can find its pattern in the Appendix. If you purchase the product directly from the manufacturer, you needn't send any registration card. The guarantee period for users who have not registered with the manufacturer is six months.

***If a simulator failure results from a defect in workmanship, we shall repair the device as if under guarantee.***

The device must be delivered for repair in the original packing, including all accessories, a copy of purchase document, and the description of failure in writing. In reasonable cases, please, supply the description of circumstances accompanying the failure:

- simulator control program version
- environment in view of possible interference (office, laboratory, industrial, and other environment types)
- description of your PC hardware: name, type, PC rate, printer port type (integrated on a HGC card, a multi I/O card, a disk controller card,...), I/O rate
- description of your PC software: type and version of your operating system, OS system (Shell, Commander, Windows,...), resident programs used.

If your claim is sent without failure description enclosed we shall have to render an account for the time needed for ascertaining these facts. Moreover, we do not guarantee to adjust your claim appropriately if the simulator is not delivered with all the accessories when the failure is outside the device itself. Unjustified claims will be charged.

The guarantee does not apply to common wear, failures due to unqualified handling, and/or rough mechanical damaging. No claim can be made in the case of any intervention into the device in question.

The manufacturer provides customer service within and after the guarantee period:

**Manufacturer:**

✉: ELNEC s. r. o., Post 5, P. O. Box 22, SK - 08005 Presov, Slovakia

☎: +42151/77 34 328, 77 31 007, fax 77 32 797

[www.elnec.com](http://www.elnec.com), [elnec@elnec.com](mailto:elnec@elnec.com)



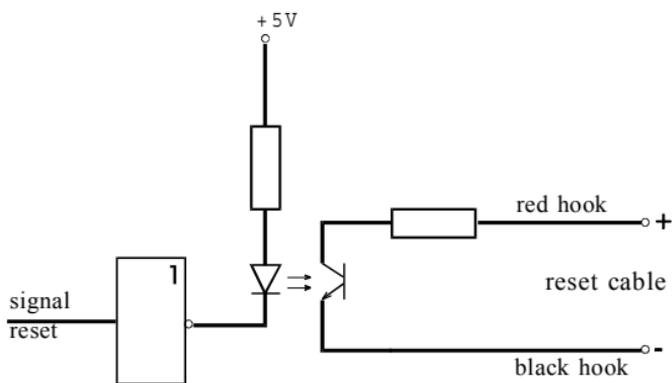
---

# *Appendix*

---



## Appendix A - reset circuit wiring



## **Appendix B - caution for ESD (electrostatic sensitive device)**

**Caution!** *Faults due to a user's failure to observe these precautions are not covered by the guarantee.*

Simulator handling safety precautions aimed at preventing it from electric damage.

Electrostatic discharge or transient current may damage the simulator. It is advisable to follow the following precautions, or equivalent rules:

- Prior to operating your simulator, touch a larger metal, or a grounded object;
- Never touch simulation base outlets if you are not certain about zero potential relative to the simulator. This also applies to all simulator-internal circuits and the connecting cable outlets.
- Do not remove the antistatic sponge protection from the simulation socket unless necessary.
- When connecting the simulator to another device, check the grounds of both units for zero potential. Ideally, the grounds should be connected by a thicker conductor.
- The simulator and the device to be connected must be OFF. This also holds for interconnecting the simulator and the PC.





## Appendix C - registration card

If the registration card is missing from your standard simulator delivery package please use a copy of the form below and send it to ELNEC. We remind you that without the "Extended warranty" document only the standard 6-month warranty is granted.

✂-----

**Device:** \_\_\_\_\_  
**Serial number:** \_\_\_\_\_  
**Name:** \_\_\_\_\_  
**Company:** \_\_\_\_\_  
**Department:** \_\_\_\_\_  
**Address:** \_\_\_\_\_  
**Post Code, City:** \_\_\_\_\_  
**Country:** \_\_\_\_\_  
**Phone/Fax/e-mail:** \_\_\_\_\_  
**Date of purchase:** \_\_\_\_\_  
**Purchase from:** \_\_\_\_\_

Complete, please:

- what type of computer are you using:  
AT/386   486   Pentium I-II   Pentium III-4
- operating system:  
MS-DOS   WINDOWS 3.x   WINDOWS 95/98  
WINDOWS NT/2000/XP
- how did you first hear about ELNEC ?  
advert.   dealer   other \_\_\_\_\_
- why did you choose this product?  
price   quality   recommendation  
features (please specify) \_\_\_\_\_
- comments: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_