

Z80 Simulator IDE

Help Topics

Table Of Contents:

[General info](#)

[File menu](#)

[Clear Memory](#), [Load Program](#), [Save Memory](#),

[Simulation menu](#)

[Start](#), [Step](#), [Stop](#),

[Rate menu](#)

[Step By Step](#), [Slow](#), [Normal](#), [Fast](#), [Extremely Fast](#), [Ultimate](#),

[Tools menu](#)

[Memory Editor](#), [Disassembler](#), [Peripheral Devices](#), [I/O Ports Editor](#), [Memory Editor 2](#), [Memory Editor 3](#), [Assembler](#), [Breakpoints Manager](#), [BASIC Compiler](#), [IEEE 754 Conversion Tool](#), [Simulation Log Viewer](#), [Interactive Assembler Editor](#), [Watch Variables](#), [External Modules](#),

[Options menu](#)

[Enable Logging](#), [Shortcuts Panel Configuration](#), [HALT Stops Simulation](#), [FF Power On Defaults](#), [Refresh Memory Editor](#), [Refresh Breakpoints Manager](#), [Save Positions](#), [Save Always On Top](#), [Auto Start Options](#), [Change Clock Frequency](#), [Change Starting Address](#), [Change Offset Address For OBJ Files Loading](#), [Enable Unofficial Instructions](#), [Prompt For Value Before IN Instruction](#), [Enable IN/OUT Instructions Logging](#), [Show Confirmation Boxes](#), [Change Registers Arrangement](#), [Change Ultimate Rate Refresh Interval](#), [Editor Setup](#), [Change Color Theme](#),

[Help menu](#)

[Help Topics](#), [BASIC Compiler Reference Manual](#), [External Modules Manual](#), [Check For Updates](#), [Bug Report Interface](#), [About](#), [View License Information](#),

[Interrupts info](#)

[Assembler info](#)

• [General info](#)

Z80 Simulator IDE is powerful application that supplies Z80 microprocessor users and educators with user-friendly graphical development environment for Windows with integrated simulator (emulator), z80 basic compiler, assembler, disassembler and debugger for the Zilog Z80 8-bit microprocessor.

The main application window shows Z80 microprocessor internal registers (and flags) structure, mnemonics of the last executed instruction, mnemonics of the next instruction that will be executed, clock cycles and instructions counter and interrupt interface.

• [File menu](#)

- [Clear Memory](#)

This command restores the initial state of the 64K memory with address range 0000H-FFFFH assumed to be attached to the CPU. The whole memory is filled with NOPs with opcode 00H.

- [Load Program](#)

This command loads program file into CPU memory. The program file must be in Intel HEX format or binary image (OBJ extension) of the memory starting from address 0000H with maximum length of 64K. Once loaded program file can be quickly reloaded by clicking on its location field on the main program interface.

- [Save Memory](#)

This command saves the contents of the memory to a file.

• [Simulation menu](#)

- [Start](#)

Z80 Simulator IDE enters simulation mode and begins the execution of instructions starting from the memory location that corresponds to starting address parameter. The default value for this parameter is 0000H and this value can be changed using Change Starting Address command from the Options menu.

- [Step](#)

This command is enabled only when Step By Step simulation rate is selected. Next instruction is executed on every click on keyboard shortcut F2.

- [Stop](#)

Z80 Simulator IDE exits simulation mode and presents the information about the total number of executed instructions, duration of the simulation and real-time duration of the simulation in clock cycles.

• [Rate menu](#)

It enables user to change the simulation rate. It is accessible during the simulation, also.

- [Step By Step](#)

The interval between consecutive instructions is at user will. When the simulator is in Step By Step mode, it is possible to change the values in all CPU internal registers, by clicking on the appropriate fields on the program interface. When this simulation rate is selected new main menu item 'STEP' will appear on the program interface. That will enable an easy access to Step command from the Simulation menu.

- [Slow](#)

The interval is 1500 ms.

- [Normal](#)

The interval is 250 ms.

- [Fast](#)

The interval is around 50 ms.

- [Extremely Fast](#)

The interval is very short and is linearly dependent on the overall computer performance.

- [Ultimate](#)

The main simulator window is not continuously refreshed after every simulated instruction that significantly improve the simulation performance. The refresh interval can be changed using Change Ultimate Rate Refresh Interval command in Options menu.

• [Tools menu](#)

- [Memory Editor](#)

This is access to the graphical interface for the Z80 Simulator CPU 64K memory. It has a scroll bar and is very easy to use. The value in the specific memory location can be changed by clicking on it. When the entered value is confirmed by pressing Enter key, edit box will automatically move to the next memory location. Edit box can be moved freely over the memory table using arrow keys and it can be closed by Esc key.

- [Disassembler](#)

Z80 Simulator IDE has very powerful internal disassembler that is started by this command.

Disassembler is independent from the simulator and it has its own program memory. So, it is necessary to load the program file (Intel HEX file or binary image) into disassembler memory first.

The disassembling process is initiated by an appropriate command from the disassembler menu.

The disassembler will always start from the address 0000H. After the operation is completed disassembler will display the output listing file. The generated listing can be saved to disk. User will be prompted to enter the name for the output file. Default extension is LST.

- [Peripheral Devices](#)

This is useful tool to monitor and control IN and OUT instructions. Up to four basic I/O peripheral devices can be setup and there is also one output terminal useful for viewing ASCII characters sent to one of the ports. The bytes sent with OUT instructions are displayed graphically showing individual bits. If the device is configured as an input device, the value it supplies can be set either by entering it directly after click on value label or by toggling individual bits of graphical representation.

- [I/O Ports Editor](#)

This is access to the graphical interface for the control over the complete I/O ports range. The value at the specific I/O port can be changed by clicking on it. When the entered value is confirmed pressing Enter key, edit box will automatically move to the next I/O port. Edit box can be moved freely over the I/O ports table using arrow keys and it can be closed by Esc key. Information in the I/O Ports Editor window is consistent with the Peripheral Devices window.

- Memory Editor 2

This command starts another Memory Editor tool with the same features.

- Memory Editor 3

This command starts yet another Memory Editor tool with the same features. So, it is possible to simultaneously view/edit three different memory ranges during the simulation.

- Assembler

This command starts integrated assembler. Assembler source files can be edited, directly assembled and finally loaded into memory in the same graphical environment. Default extension is ASM. After the successful assembly process two new files are generated. One with OBJ extension that is binary image of the program and that can be directly loaded into CPU memory and the other with LST extension that is assembler listing used with the debugger. If Generate HEX File Also option is selected then program file in Intel HEX format will also be generated. It is not only possible to use the internal assembler - this window can also be used as the graphical interface for the external TASM assembler. It is only necessary to locate the executable file of the TASM assembler before the first use. If TASM assembler is used from the command line, -80 -b options will generate OBJ file.

- Breakpoints Manager

This command starts integrated debugger that can be used to debug and monitor the program execution. If the assembler listing file of the program in memory do not exist useable alternative listing will be generated by the internal disassembler. It is possible to define up to 10 breakpoints by clicking on individual lines in the loaded program listing. When the simulation starts in faster rate modes it will automatically switch to Step By Step mode when reaching any of these breakpoints. The breakpoints are marked by red circles, and the current value of the PC register is marked by yellow arrow. There is an option to keep the PC pointer in focus during the simulation.

- BASIC Compiler

Integrated BASIC compiler editor window will be opened. More information available in BASIC Compiler Reference Manual. It can be accessed from the Help menu of the main application window or from the BASIC compiler editor window.

- IEEE 754 Conversion Tool

This command starts the tool that converts real numbers to their four byte IEEE 754 single precision floating point representation. That representation is used for 'Single' data type by BASIC compiler.

- Simulation Log Viewer

This command starts integrated graphical tool that will log all simulated instructions together with Z80 registers and flags status. This is very useful tool for debugging process.

- Interactive Assembler Editor

This command starts integrated graphical tool that will enable beginners to write their first assembler routines interactively without having to memorize the mnemonics of individual instructions from Z80 rich instruction set. This is a great tool for educational purposes.

- Watch Variables

During the simulation of programs written using integrated basic compiler, this tool can be used to watch the current values of all variables declared in the simulated basic program. It is also possible to add user defined variables to the list to monitor other memory locations of interest during the simulation. This feature is useful for memory monitoring for simulated program files, that are not compiled with the integrated basic compiler. User added variables will be remembered between sessions as long as the same program file is loaded in the simulator. Variables from the watch list can be easily removed with the Delete Variable command, so the list can contain variables of special interest only. Other commands and options include: Change Variable Value (can be also started by a single-click on the variable from the list), Display HEX Values, Confirm Delete.

- External Modules

This tool should be used to establish automation interface with up to five external client/server modules. It is required to enter the class name supplied by external device in the form ApplicationName.ObjectName in order to establish connection with it. External client/server applications will be started and terminated automatically with Z80 Simulator IDE. More information available in External Modules Manual. It can be accessed from the Help menu of the main application window.

- **Options menu**

- **Enable Logging**

This option will force the simulator to log to LOG.TXT file in the application folder all simulated instructions together with Z80 registers and flags status. This option do not interfere with the integrated graphical Simulation Log Viewer.

- **Shortcuts Panel Configuration**

This option command opens an easy to use interface for turning on and customizing the fancy-looking shortcuts panel on the main IDE window, for an easy access to the most frequently used menu commands. The panel can contain up to three lines of menu item shortcuts. All main IDE window menu items are available to be placed on the panel.

- **HALT Stops Simulation**

If this option is selected the simulation will automatically stop when HALT instruction is reached. If it is not selected, the simulator will, just as real Z80 CPU does, execute this instruction repeatedly until it receives an interrupt. After the return from interrupt the execution will continue with next instruction.

- **FF Power On Defaults**

Toggleing this option will switch between 00H and FFH initial values for Z80 CPU registers.

- **Refresh Memory Editor**

If this option is selected and Memory Editor is started, the displayed memory range will be refreshed after every simulated instruction, in all simulation rate modes. This is useful if you want to monitor what is going on on the stack or elsewhere in the memory during the simulation.

- **Refresh Breakpoints Manager**

If this option is selected and Breakpoints Manager is started, it will be refreshed after every simulated instruction. If Hold PC In Focus option is also selected, that will enable user to watch live presentation of the program execution.

- **Save Positions**

With this option selected, the positions of the windows on the screen will be remembered.

- **Save Always On Top**

With this option selected, the Always On Top setting for all of the windows with this feature will be remembered.

- **Auto Start Options**

With this utility users can define actions that will be performed on the application startup. These actions include automatic opening of various tools and simulation interfaces from the Tools menu and automatic loading of the last used files in the simulator, assembler and basic compiler.

- **Change Clock Frequency**

This command allows user to change the frequency parameter that is used for the calculation of the real-time duration of the simulation. The entered value in MHz is remembered for the future sessions. The default value is 4 MHz.

- **Change Starting Address**

This command allows user to change the starting address for the simulation. The entered value is remembered for the future sessions. The default value is 0000H.

- **Change Offset Address For OBJ Files Loading**

Allows user to change the starting address that will be used for binary image files (OBJ extension) loading into the 64K memory buffer. The entered value is remembered for the future sessions. The default value is 0000H. This parameter is ignored for HEX files loading.

- **Enable Unofficial Instructions**

This option will turn on the simulation of documented unofficial Z80 instructions.

- **Prompt For Value Before IN Instruction**

This option will force the program to always prompt user to manually enter every incoming byte on all ports. If it is off the value will be taken from Peripheral Devices or I/O Ports Editor window.

- **Enable IN/OUT Instructions Logging**

When this option is enabled the program will log all IN and OUT instructions in IO.TXT file located in application folder. Every IN and OUT instruction will append a new line in that file.

- **Show Confirmation Boxes**

When this option is enabled the confirmation boxes showing results of operations will be displayed and will require user response to be closed.

- **Change Registers Arrangement**

This command will change the arrangement of labels on the main and alternate registers panels. There are two different arrangements available.

- **Change Ultimate Rate Refresh Interval**

This command allows user to change the refresh interval (in milliseconds) for the main simulation

interface when the simulation is running at Ultimate rate. Its value however does not affect the simulation performance considerably. The default value is 500ms.

[- Editor Setup](#)

With this setup tool it is possible to change various properties of basic compiler and assembler code editors.

[- Change Color Theme](#)

This command will open a dialog with the rich list of available color themes, so that user can change application appearance.

• [Help menu](#)

[- Help Topics](#)

This command will display Help Topics. This help file contains general information about the application with description of all menu items.

Help viewer window features navigation panel showing topics and subtopics of the displayed help file. Right-click on the navigation panel will show popup menu with Show All Subtopics and Hide All Subtopics commands. Single-click on the item from the navigation panel will move focus on the display panel to the appropriate position. Double-click on the topic item will show/hide its subtopics. The display panel shows the content of the loaded help file. Right-click will display popup menu containing various options and commands including: Copy, Copy RTF, Copy HTML, Print, Font Increase, Font Decrease, Font Reset, Always On Top. Help viewer window is resizable and will remember both its position and size. The vertical separator between navigation and display panels is moveable and its position will also be saved after the viewer is closed.

[- BASIC Compiler Reference Manual](#)

BASIC Compiler Reference Manual will be displayed in the help viewer.

[- External Modules Manual](#)

External Modules Manual will be displayed in the help viewer.

[- Check For Updates](#)

This tool will enable user to establish connection with OshonSoft.com web site to check out if there is a new software release available for download. Version log file will be displayed after the response from the web site has been received.

[- Bug Report Interface](#)

This interface should be used to send the reports about possible bugs in software to OshonSoft.com. In addition to the user written part the full report will contain a part that is generated by software (system report).

[- About](#)

This command will display the basic information about the software package.

[- View License Information](#)

This command will display the information about the installed license for the software.

• [Interrupts info](#)

All possible interrupts are supported: non-maskable interrupt (NMI) and three modes of maskable interrupts (INT mode 0, INT mode 1, INT mode 2). NMI jumps to restart location 0066H. INT mode 0 prompts user to enter the RST instruction supplied by the peripheral device and then initiates a call to the selected one of eight restart locations. INT mode 1 jumps to the restart location 0038H. INT mode 2 prompts user to enter the 8-bit vector supplied by the peripheral device, and then forms a pointer using this byte as the lower 8 bits and the contents of the I register as the upper 8 bits. This points to an entry in the table of addresses for interrupt service routines, and the simulator jumps to the routine at that address. Interrupts are triggered by pressing NMI and INT buttons on the interrupt interface. The simulation can be reset by pressing the RESET button.

• [Assembler info](#)

Internal assembler is excellent solution for assembling source files with size up to 20K. For larger files the assembly process can take some time, but there are no limits in file size. Its limits are that labels can be a maximum of 32 characters long and that only .ORG (ORG), .EQU (EQU), .DB (DB, .DEFB, DEFB), .DW (DW, .DEFW, DEFW), .DS (DS, .DEFS, DEFS) and .END (END) assembler directives are supported. If better performance is needed, the graphical interface for TASM assembler can be used. Shareware version of the TASM assembler can be downloaded from <http://home.comcast.net/~tasm/>