

## File Menu



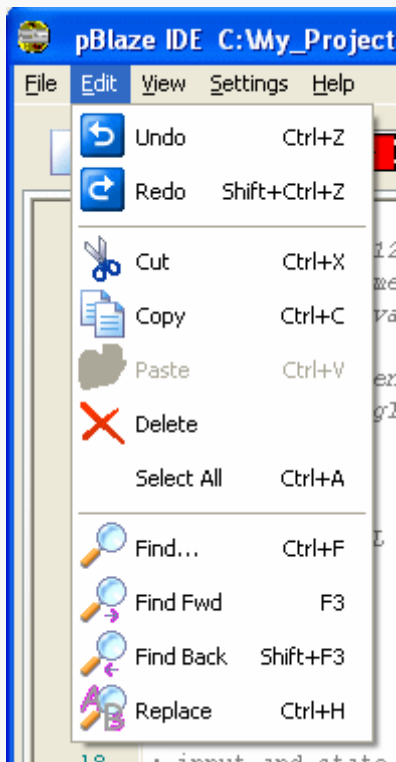
- **New** creates a new document to enter a new source file
- **Open** allows to open an existing source file
- **Recent Files** lists the 5 most recently used files
- **Import** creates a new document and imports a Xilinx (KCPSMBLE) source
  - [This](#) is an example KCPSMBLE source
  - [This](#) is the unedited result after importing
- **Export to HTML** creates a new file formatted as HTML for publication

Note:  
All the code fragments on this site are processed by exporting to HTML.
- **Save** saves the open document to the related file
- **Save As** save the open document to a new file
- **Close** closes the currently open document
- **Close All Files** closes all currently open documents
- **Preview** shows the first page as it will appear on the printer
- **Print** will print the open document on the printer
- **Exit** closes all open documents and closes pBlazIDE

Equivalent tool buttons:



## Edit Menu

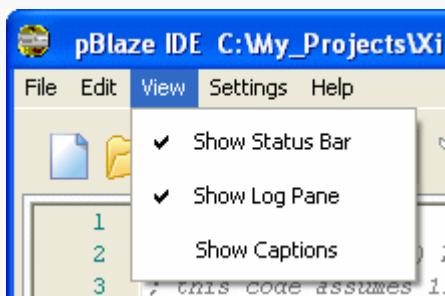


- **Undo** will undo all edit actions on the open document in reverse order
- **Redo** will redo all undo actions
- **Cut** cuts all selected parts of the open document and puts it on the clipboard
- **Copy** copies all selected parts of the open document to the clipboard
- **Paste** pastes the text contents of the clipboard in the open document
- **Delete** deletes all selected parts of the document
- **Select All** selects all text in the open document
- **Format** format the open document using the syntax rules for a PSM document
- **Find** find a to be given string through the open document
- **Find Fwd** find the given string forward through the open document
- **Find Back** find the given string backward through the open document
- **Replace** replace a given string in the open document by a new string

Equivalent tool buttons:

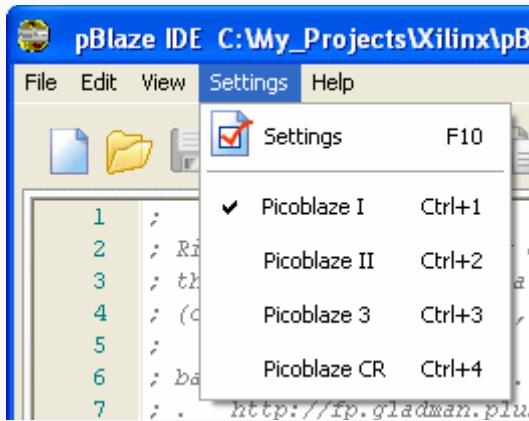


## View Menu



- **Show Status Bar** shows or hides the status bar
- **Show Log Pane** shows or hides the log
- **Show Captions** shows or hides the captions of the tool buttons

## Setting Menu



- **Settings** opens the [settings dialog](#)
- **Picoblaze I** selects the Picoblaze-I (for Spartan-II(e) and Virtex) mode
- **Picoblaze II** selects the Picoblaze-II (for Virtex-II) mode
- **Picoblaze 3** selects the Picoblaze-3 (for Spartan-3) mode
- **Picoblaze CR** selects the Coolblaze (for Coolrunner) mode

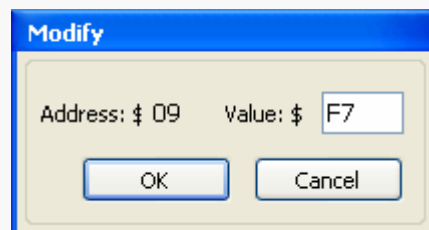
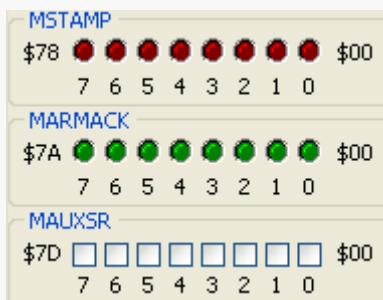
## Simulation Control



- **Assemble** assembles the open document and starts the simulator. If there are any errors the simulator will not be started. Pressing this button again will switch back to edit mode.
- **Reset** resets the simulator, the PC and SP are initialized, the registers are not.
- **Run** starts full speed simulation. Only if the reset, pause or step buttons are pressed the execution simulation is halted. If a run-time error, like a stack-overflow occurs, the simulation is aborted.
- **Step** steps the simulator; only one instruction is simulated.
- **Step Over** steps simulator; only one instruction is simulated; if this is a **JUMP, CALL, RET or RETI**, the simulation will end when the PC is again at the current PC + 1. This function is similar to setting a breakpoint after the current instruction.
- **Step till cursor** runs the simulator until the PC is at the line where the text cursor is pointing to.
- **Pause** stops a running simulation, but does not alter PC or SP. The simulation can be continued in any way.
- **Toggle breakpoint** sets or removes a breakpoint at the line where the text cursor is at. If the PC reaches a line with a breakpoint the simulation is paused.
- **Remove all breakpoints** removes all breakpoints.

## Simulated I/O

The IO panels come in three variations, a pure output panel, specified by **DSOUT**; a pure input panel specified by **DSIN** and an in/output panel, specified by **DSIO**. The value of the whole byte can be changed for the input and in/output panels by double clicking on the byte value; a modify dialog will appear.



The RAM and ROM panels are 16 byte, distributed RAM simulations; they are specified by **DSRAM** and **DSROM**

repectively. The Scratch Pad simulation is for the Picoblaze-3 core for the Spartan-3 series and will automatically appear if **FETCH** or **STORE** instructions are used. The value of the bytes can be changed for the panels by double clicking on the byte value; a modify dialog will appear.

## Directives

pBlazIDE sources can contain the following directives:

```
;
; sample file to show the use of the available directives in pBlazIDE
; (c) 2003 Henk van Kampen, www.mediatronix.com
;
; ORG
; sets the instruction pointer of the assembler to a new address in the Code map
           ORG      $FF                ; set the instruction pointer to $FF
; EQU
; equates a value to a symbol
; can be used to assign a name to a register too
maxcount      EQU      128 / 8
counter       EQU      s8
; DS, DSIN, DSOUT, DSIO
; define I/O ports
temp          DS       $80                ; a I/O port at 80 hexadecimal
; following ports will be presented during simulation
switches     DSIN     $10                ; define an inputport at portaddress $10
LEDs         DSOUT    $11                ; specify an output port at $11
; specify an output port which can be read back at the same address
backup       DSIO     $12
; DSROM and DSRAM can be used to define RAM16 blocks used as RAM or ROM
; the first value is the base address (can only be a multiple of $10)
lookup       DSROM    $00, 0, 2, 4, 6, 8, 10, 12, 14, 1, 3, 5, 7, 9, 11, 13, 15
buffer       DSRAM    $20
; specify include file(s)
; include files are processed after the 1st pass of the main file
; only basic definitions should be used in an include file
           INCL      "defs.inc", "io.inc"
; the following functions will be performed after all assembly functions:
; specify an VHDL template and target file and a entity name

; tempalet file contains all VHDL code, some replace ments will
; be made based on the code .the template should start with: {begin template}
; the following strngs will be replaced:
; {name} -> replaced by the entity name
; {pico} -> replaced by pbtI, pbtII, pbt3 and pbtC depending on Mode
```

```

; {INIT...} replaced by configuration data representing the instruction code
        VHDL      "template.vhd", "target.vhd", "ROM"
; build a COE coefficient file
; specify the target file and word width (only for pb-I)
        COE       "target.coe", 8
        COE       "target.coe", 16
        COE       "target.coe"
; build a MEM memory file usable by Data2BRAM/Data2MEM
        MEM       "target.mem"
; define a lookup table based on a BlockRAM
; first value is the port address of the lookup address (lower byte)
; the upper byte (if any) is placed at the buddy location of the address
; the second value is the size in bytes of the BlockRAM used/necessary
; the last part is the name of the file used to initialize the ROM
lookup   BROM     $7E, 1024, "test.mem"; the upper address is at $7F
; execute a command line
; the specified command will be executed after all assembly functions
; only one command can be executed, however, this can be a batch file
        EXEC     "impact -batch bat_file.cmd"

```

## Error Messages

### Error messages of pBlazIDE

Errors which occur in the 2nd pass of the assembler are recorded in the gutter of the source window and in the log.

Errors occurring in the 1st pass are only recorded in the log.

?Number	Unexpected characters in the conversion of a number
?Closing	Missing closing quote of a char constant or a closing bracket in an expression
?Syntax	General syntax error
?Colon	After a label a colon is expected, could be a typo in an opcode
?Opcode	Using an opcode which is not supported by this core
?Register	Using a register which is not supported by this core
?Label	Unexpected label declaration, unexpected colon
?Operator	Operator expected in an expression
?Comma	Comma expected between parameters
?Double	A symbol or register name is already defined
?Reg Name	A symbol or register name is already defined
?Equate	No symbol given to equate
?Not found	Supplied file does not exist
?Address	Address value expected
?Undefined	Address value expected

?Register2	2nd register is illegal here
?Operand2	2nd operand is illegal here
?Phasing	Value equated to a symbol is different in the 2nd pass
?ROM Start	<b>DSROM</b> should start at a multiple of 16
?RAM Start	<b>DSRAM</b> should start at a multiple of 16
?I/O Map	Specified port address is already used
?Address	Value supplied is not an address
?Size	Value supplied is not a size value
?Not allowed	Multiple use of <b>VHDL</b> , <b>COE</b> , <b>MEM</b> , etc. directives
?Width	Problem with the width value in a <b>COE</b> directive
?> \$FF	Program exceeds the code space of 256 instructions
?> \$3FF	Program exceeds the code space of 1024 instructions
?Unexpected	pBlazIDE did not expect this condition, please report this to: <a href="mailto:pBlazIDE@mediatronics.com">pBlazIDE@mediatronics.com</a>
Please report any other error messages or just plain bugs to: <a href="mailto:pBlazIDE@mediatronics.com">pBlazIDE@mediatronics.com</a>	

## Screenshot

The screenshot displays the pBlaz IDE interface for a project named 'testbench-I.psm'. The main window shows assembly code for a testbench, including memory initialization for LEDs, switches, RAM, and ROM, and a test routine. The registers window shows the current state of registers s0x and s1x. The memory map shows the layout of RAM and ROM. The status bar at the bottom indicates the mode is Picoblaze-II, with 17 instructions, 48 cycles, and PC at \$004.

```

; pBlazIDE testbench
;
$00      leds          DSRWF  0
$01      switches     DSRIN  1
$10     ram           DSRAM  $10, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5,
$20     rom           DSRAM  $20, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
$30     rom1          DSRAM  $30, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5,
$40     rom1          DSRAM  $40, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,

$000 $00010 • TestStackNZ:  LOAD  s00, 16
$001 $00110 •              LOAD  s01, 16
$002 $0C001 • TestStackNZ10: SUB   s00, 1
$003 $37402 •              CALL  NZ, TestStackNZ10
$004 $0C101 •              SUB   s01, 1
$005 $25400 •              RET   NZ

$006 $000FF • TestStackC:   LOAD  s00, $FF
$007 $001FF •              LOAD  s01, $FF

```

Registers:

s0x	s1x
0	00 00
1	09 00
2	00 00
3	00 00
4	00 00
5	00 00

Memory Maps:

ram	rom	rom1	rom1
\$10 0F 0E 0D 0C 0B 0A 09 08 07 06 05 04 03 02 01 00	\$20 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	\$30 0F 0E 0D 0C 0B 0A 09 08 07 06 05 04 03 02 01 00	\$40 0C 0B 0A 09 08 07 06 05 04 03 02 01 00 0

Assembler Phase 1: building symbol table  
Assembler Phase 2: constructing opcodes  
Assembler Phase 3: building simulation objects  
Program is Paused  
Program is Started  
Program is Paused

Mode: Picoblaze-II    17: 1    Cycle: 48    PC: \$004    SP: 8 (\$08)    Stack: \$04 \$04 \$04 \$04 \$04 \$04 \$04 \$04