Abstract: This document describes the instruction set of the Caml\textsuperscript{1} Virtual Machine (sometimes referred to as the Zinc machine), covering its “3.11.2” version. This document is structured in three parts: the first one lists the instructions sorted by ascending opcode with complete description, the second part is a table of instructions sorted by ascending mnemonic, and the third and last part lists the instructions per group (a group gathering instructions related to the same purpose). Hyperlinks allow easy navigation between the different parts of the document.

Introduction

The virtual machine executing a caml bytecode is stack-centered. The caml stack, augmented with an accumulator, stores values. A caml value can be either:

- a \textit{long} value corresponding to the \texttt{int} type, that is a signed integer of either 31-bit long (on 32-bit architectures) or 63-bit long (on 64-bit architectures);
- a \textit{block} value, composed of an header (indicating the type and size of the block) and a memory address that point to actual data (which can consist of values indexed by an integer);
- a \textit{code offset} value, relative to code start (the code value of a block is the code oofset value stored at index 0 in this block).

Along with the stack, the virtual machine contains seven registers:
\texttt{pc} program counter, that is next instruction to interpret;
\texttt{sp} stack pointer, that is top of the stack;
\texttt{accu} accumulator;
\texttt{trapSp} stack pointer of highest exception handler;
\texttt{extraArgs} number of extra arguments to function application;
\texttt{env} environment;
\texttt{global} global data.

\footnote{The official Caml website can be reached at \texttt{caml.inria.fr} and contains the full development suite (compiler, tools, virtual machine, \textit{etc.}) as well as links to third-party contributions.}
Specification of instructions

This section lists the instructions sorted by ascending opcode, with parameters and complete description. For each instruction, a figure representing stack evolution is provided. These schemas have two parts: the left one represents the state of the stack before instruction execution whereas the right one represents the state of the stack after instruction execution. In both parts, the stack (growing from bottom up) is depicted with its elements drawn in solid boxes and the accumulator is drawn in a dashed box.

ACC0 (opcode: 0)

parameters: –

synonym: –

related to: accumulator stack

description: Peeks the top of the stack and puts it into the accumulator.

ACC1 (opcode: 1)

parameters: –

synonym: –

related to: accumulator stack

description: Peeks the second element of the stack and puts it into the accumulator.

ACC2 (opcode: 2)

parameters: –

synonym: –

related to: accumulator stack

description: Peeks the third element of the stack and puts it into the accumulator.

ACC3 (opcode: 3)

parameters: –

synonym: –

related to: accumulator stack

description: Peeks the fourth element of the stack and puts it into the accumulator.
ACC4 (opcode: 4)
parameters: –
synonym: –
related to: accumulator stack
description: Peeks the fifth element of the stack and puts it into the accumulator.

ACC5 (opcode: 5)
parameters: –
synonym: –
related to: accumulator stack
description: Peeks the sixth element of the stack and puts it into the accumulator.

ACC6 (opcode: 6)
parameters: –
synonym: –
related to: accumulator stack
description: Peeks the seventh element of the stack and puts it into the accumulator.

ACC7 (opcode: 7)
parameters: –
synonym: –
related to: accumulator stack
description: Peeks the eighth element of the stack and puts it into the accumulator.
ACC (opcode: 8)
parameters: \( n \)
synonym: –
related to: accumulator stack
description: Peeks the \( n+1 \)-th element of the stack and puts it into the accumulator.

PUSH (opcode: 9)
parameters: –
synonym: PUSHACC0
related to: accumulator stack
description: Pushes the accumulator onto the stack.

PUSHACC0 (opcode: 10)
parameters: –
synonym: PUSH
related to: accumulator stack
description: Pushes the accumulator onto the stack. Equivalent to PUSH then ACC0.

PUSHACC1 (opcode: 11)
parameters: –
synonym: –
related to: accumulator stack
description: Pushes the accumulator onto the stack and then peeks the second element of the stack into the accumulator. Equivalent to PUSH then ACC1.

PUSHACC2 (opcode: 12)
parameters: –
synonym: –
related to: accumulator stack
description: Pushes the accumulator onto the stack and then peeks the third element of the stack into the accumulator. Equivalent to PUSH then ACC2.
PUSHACC3 (opcode: 13)
parameters: –
synonym: –
related to: accumulator stack
description: Pushes the accumulator onto the stack and then peeks the fourth element of the stack into the accumulator.
Equivalent to PUSH then ACC3.

PUSHACC4 (opcode: 14)
parameters: –
synonym: –
related to: accumulator stack
description: Pushes the accumulator onto the stack and then peeks the fifth element of the stack into the accumulator.
Equivalent to PUSH then ACC4.

PUSHACC5 (opcode: 15)
parameters: –
synonym: –
related to: accumulator stack
description: Pushes the accumulator onto the stack and then peeks the sixth element of the stack into the accumulator.
Equivalent to PUSH then ACC5.

PUSHACC6 (opcode: 16)
parameters: –
synonym: –
related to: accumulator stack
description: Pushes the accumulator onto the stack and then peeks the seventh element of the stack into the accumulator.
Equivalent to PUSH then ACC6.
PUSHACC7 (opcode: 17)

parameters: –
synonym: –
related to: accumulator stack
description: Pushes the accumulator onto the stack and then peeks the eighth element of the stack into the accumulator. Equivalent to PUSH then ACC7.

PUSHACC (opcode: 18)

parameters: \( n \)
synonym: –
related to: accumulator stack
description: Pushes the accumulator onto the stack and then peeks the \( n + 1 \)-th element of the stack into the accumulator. Equivalent to PUSH then ACC.

POP (opcode: 19)

parameters: \( n \)
synonym: –
related to: stack
description: Pops \( n \) elements from the stack.

ASSIGN (opcode: 20)

parameters: \( n \)
synonym: –
related to: accumulator stack
description: Sets the element of index \( n \) in the stack (0 being the top) to the value of the accumulator. Then sets the accumulator to the unit value.
ENVACC1 (opcode: 21)
parameters: –
synonym: –
related to: accumulator environment
description: Sets the accumulator to the field of index 1 of the environment.

ENVACC2 (opcode: 22)
parameters: –
synonym: –
related to: accumulator environment
description: Sets the accumulator to the field of index 2 of the environment.

ENVACC3 (opcode: 23)
parameters: –
synonym: –
related to: accumulator environment
description: Sets the accumulator to the field of index 3 of the environment.

ENVACC4 (opcode: 24)
parameters: –
synonym: –
related to: accumulator environment
description: Sets the accumulator to the field of index 4 of the environment.

ENVACC (opcode: 25)
parameters: \(n\)
synonym: –
related to: accumulator environment
description: Sets the accumulator to the field of index \(n\) of the environment.
PUSHENVACC1 (opcode: 26)
parameters: –
synonym: –
related to: accumulator stack environment
description: Pushes the accumulator onto the stack and then sets the accumulator to the field of index 1 of the environment. Equivalent to PUSH then ENVACC1.

PUSHENVACC2 (opcode: 27)
parameters: –
synonym: –
related to: accumulator stack environment
description: Pushes the accumulator onto the stack and then sets the accumulator to the field of index 2 of the environment. Equivalent to PUSH then ENVACC2.

PUSHENVACC3 (opcode: 28)
parameters: –
synonym: –
related to: accumulator stack environment
description: Pushes the accumulator onto the stack and then sets the accumulator to the field of index 3 of the environment. Equivalent to PUSH then ENVACC3.

PUSHENVACC4 (opcode: 29)
parameters: –
synonym: –
related to: accumulator stack environment
description: Pushes the accumulator onto the stack and then sets the accumulator to the field of index 4 of the environment. Equivalent to PUSH then ENVACC4.
PUSHENVACC (opcode: 30)
parameters: \(n\)
synonym: –
related to: accumulator stack environment
description: Pushes the accumulator onto the stack and then sets the accumulator to the field of index \(n\) of the environment. Equivalent to PUSH then ENVACC.

PUSH-RETADDR (opcode: 31)
parameters: \(ofs\)
synonym: –
related to: stack call
description: Pushes \(extraArgs\), then the environment, and then \(pc + ofs\).

APPLY (opcode: 32)
parameters: \(args\)
synonym: –
related to: accumulator environment call
description: Sets \(extraArgs\) to \(args - 1\). Sets \(pc\) to the code value of the accumulator. Then sets the environment to the value of the accumulator.

APPLY1 (opcode: 33)
parameters: –
synonym: –
related to: accumulator stack environment call
description: Pops one argument from the stack and pushes \(extraArgs\), environment, \(pc\) and argument back. Then \(pc\) is set to code value of accumulator, environment to accumulator and \(extraArgs\) to 0.
APPLY2 (opcode: 34)

parameters: –
synonym: –
related to: accumulator stack environment call
description: Pops two arguments from the stack and pushes $extraArgs$, environment, $pc$ and arguments back. Then $pc$ is set to code value of accumulator, environment to accumulator and $extraArgs$ to 1.

APPLY3 (opcode: 35)

parameters: –
synonym: –
related to: accumulator stack environment call
description: Pops three arguments from the stack and pushes $extraArgs$, environment, $pc$ and arguments back. Then $pc$ is set to code value of accumulator, environment to accumulator and $extraArgs$ to 2.

APPTERM (opcode: 36)

parameters: $n, s$
synonym: –
related to: accumulator stack environment call
description: Slides the $n$ top elements from the stack towards bottom of $s - n$ positions. Then sets $pc$ to the code value of the accumulator, the environment to the accumulator, and increases $extraArgs$ by $n - 1$.

APPTERM1 (opcode: 37)

parameters: $n$
synonym: –
related to: accumulator stack environment call
description: Peeks the top element from the stack as $arg$, pops $n - 1$ elements from the stacks, and pushes back $arg$. Then sets $pc$ to the code value of the accumulator, and the environment to the accumulator.
APPTERM2 (opcode: 38)

parameters: \( n \)

synonym: –

related to: accumulator stack environment call

description: Peeks the top elements from the stack as \( arg_1 \) and \( arg_2 \), pops \( n - 2 \) elements from the stacks, and pushes back \( arg_1 \) and \( arg_2 \). Then sets \( pc \) to the code value of the accumulator, the environment to the accumulator, and increments \( extraArgs \).

Stack evolution.

APPTERM3 (opcode: 39)

parameters: \( n \)

synonym: –

related to: accumulator stack environment call

description: Peeks the top elements from the stack as \( arg_1, arg_2 \) and \( arg_3 \), pops \( n - 3 \) elements from the stacks, and pushes back \( arg_1, arg_2 \) and \( arg_3 \). Then sets \( pc \) to the code value of the accumulator, the environment to the accumulator, and adds two to \( extraArgs \).

Stack evolution.

RETURN (opcode: 40)

parameters: \( n \)

synonym: –

related to: accumulator stack environment call

description: Pops \( n \) elements from the stack. If \( extraArgs \) is strictly positive then it is decremented, \( pc \) is set to the code value of the accumulator, and the environment is set to the value of the accumulator. Otherwise, three values are popped from the stack and assigned to \( pc \), environment and \( extraArgs \).

Stack evolution.
RESTART (opcode: 41)

parameters: –
synonym: –
related to: stack environment

description: Computes \( n \), the number of arguments, as the size of the environment minus 2. Then pushes elements of the environment from index \( n - 1 \) to 2 onto the stack. Environment is set to the element of index 1 of the environment and \( \text{extraArgs} \) is increased by \( n \).

GRAB (opcode: 42)

parameters: \( n \)
synonym: –
related to: accumulator stack environment call

description: If \( \text{extraArgs} \) is greater than or equal to \( n \), then \( \text{extraArgs} \) is decreased by \( n \). Otherwise, creates a closure of \( \text{extraArgs} + 3 \) elements in the accumulator. Code of this closure is set to \( pc - 3 \), element of index 1 is set to the environment and other elements are set to values popped from the stack. Then \( pc \), environment, and \( \text{extraArgs} \) are popped from the stack.

CLOSURE (opcode: 43)

parameters: \( n, ofs \)
synonym: –
related to: accumulator stack

description: If \( n \) is greater than zero then the accumulator is pushed onto the stack. A closure of \( n + 1 \) elements is created into the accumulator. The code value of the closure is set to \( pc + ofs \). Then, the other elements of the closure are set to values popped from the stack.
CLOSURERE (opcode: 44)
parameters: $f, v, o, t$
synonym: –
related to: accumulator stack
description: If $v$ is greater than 0 then the accumulator is pushed onto the stack. A closure of $2f - 1 + v$ elements is created into the accumulator. The code value of the closure is set to $pc + o$. The $v$ last elements are set to values popped from the stack. Then, the element of index 0 is set to $pc + o$. The last $2f$ remaining elements are set to created infix blocks whose values are read from $t$, each of these infix block being pushed onto the stack (each infix block takes two consecutive fields: the first one being the header, the second one being the actual value).

OFFSETCLOSUREM2 (opcode: 45)
parameters: –
synonym: –
related to: accumulator environment
description: Sets the accumulator to the value of the closure preceding the environment.

OFFSETCLOSURE0 (opcode: 46)
parameters: –
synonym: –
related to: accumulator environment
description: Sets the accumulator to the value of the environment.

OFFSETCLOSURE2 (opcode: 47)
parameters: –
synonym: –
related to: accumulator environment
description: Sets the accumulator to the value of the closure following the environment.
OFFSETCLOSURE (opcode: 48)

parameters: \( n \)

synonym: –

related to: accumulator environment

description: Sets the accumulator to the value of the \( n \)-th closure relatively to the environment.

PUSHOFFSETCLOSUREM2 (opcode: 49)

parameters: –

synonym: –

related to: accumulator stack environment

description: Pushes the accumulator onto the stack. Then, sets the accumulator to the value of the closure preceding the environment.

Equivalent to PUSH then OFFSETCLOSUREM2.

PUSHOFFSETCLOSURE0 (opcode: 50)

parameters: –

synonym: –

related to: accumulator stack environment

description: Pushes the accumulator onto the stack. Then, sets the accumulator to the value of the environment.

Equivalent to PUSH then OFFSETCLOSURE0.

PUSHOFFSETCLOSURE2 (opcode: 51)

parameters: –

synonym: –

related to: accumulator stack environment

description: Pushes the accumulator onto the stack. Then, sets the accumulator to the value of the closure following the environment.

Equivalent to PUSH then OFFSETCLOSURE2.
PUSHOFFSETCLOSURE (opcode: 52)

parameters: $n$

synonym: –

related to: accumulator stack environment

description: Pushes the accumulator onto the stack. Then, sets the accumulator to the value of the $n$-th closure relatively to the environment.
Equivalent to PUSH then OFFSETCLOSURE.

GETGLOBAL (opcode: 53)

parameters: $n$

synonym: –

related to: accumulator

description: Sets the accumulator to the field of index $n$ of global data.

PUSHGETGLOBAL (opcode: 54)

parameters: $n$

synonym: –

related to: accumulator stack

description: Pushes the accumulator onto the stack. Then, sets the accumulator to the field of index $n$ of global data.
Equivalent to PUSH thenGETGLOBAL.

GETGLOBALFIELD (opcode: 55)

parameters: $n, p$

synonym: –

related to: accumulator

description: Sets the accumulator to the field of index $p$ of the field of index $n$ of the global data.
PUSHGETGLOBALFIELD (opcode: 56)

parameters: \(n, p\)

synonym: –

related to: accumulator stack

description: Pushes the accumulator onto the stack. Then, sets the accumulator to the field of index \(p\) of the field of index \(n\) of the global data.

Equivalent to PUSH then GETGLOBALFIELD.

SETGLOBAL (opcode: 57)

parameters: \(n\)

synonym: –

related to: accumulator

description: Sets the field of index \(n\) of the global data to the value of the accumulator. Then, sets the accumulator to the unit value.

ATOM0 (opcode: 58)

parameters: –

synonym: –

related to: accumulator

description: Sets the accumulator to the value of the 0-th atom.

ATOM (opcode: 59)

parameters: \(n\)

synonym: –

related to: accumulator

description: Sets the accumulator to the value of the \(n\)-th atom.
PUSHATOM0 (opcode: 60)

parameters: –
synonym: –
related to: accumulator stack
description: Pushes the accumulator onto the stack. Then sets the accumulator to the value of the 0-th atom. Equivalent to PUSH then ATOM0.

PUSHATOM (opcode: 61)

parameters: $n$
synonym: –
related to: accumulator stack
description: Pushes the accumulator onto the stack. Then sets the accumulator to the value of the $n$-th atom. Equivalent to PUSH then ATOM.

MAKEBLOCK (opcode: 62)

parameters: $n, t$
synonym: –
related to: accumulator stack
description: Creates a block of $n$ elements, with tag $t$. The element of index 0 of the block is set to the value of the accumulator, the $n - 1$ other elements are popped from the stack. Then the accumulator is set to the created block.

MAKEBLOCK1 (opcode: 63)

parameters: $t$
synonym: –
related to: accumulator stack
description: Creates a block of 1 element, with tag $t$. The element of index 0 of the block is set to the value of the accumulator. Then the accumulator is set to the created block.
MAKEBLOCK2 (opcode: 64)
parameters: $t$
synonym: –
related to: accumulator stack
description: Creates a block of 2 elements, with tag $t$. The element of index 0 of the block is set to the value of the accumulator, the other element being popped from the stack. Then the accumulator is set to the created block.

MAKEBLOCK3 (opcode: 65)
parameters: $t$
synonym: –
related to: accumulator stack
description: Creates a block of 3 elements, with tag $t$. The element of index 0 of the block is set to the value of the accumulator, the other elements being popped from the stack. Then the accumulator is set to the created block.

MAKEFLOATBLOCK (opcode: 66)
parameters: $n$
synonym: –
related to: accumulator stack
description: Creates a block of $n$ float values. The element of index 0 of the block is set to the value of the accumulator, the other elements being popped from the stack. Then the accumulator is set to the created block.

GETFIELD0 (opcode: 67)
parameters: –
synonym: –
related to: accumulator
description: Sets the accumulator to the value of the field of index 0 of the accumulator.
GETFIELD1 (opcode: 68)
parameters: –
synonym: –
related to: accumulator
description: Sets the accumulator to the value of the field of index 1 of the accumulator.

GETFIELD2 (opcode: 69)
parameters: –
synonym: –
related to: accumulator
description: Sets the accumulator to the value of the field of index 2 of the accumulator.

GETFIELD3 (opcode: 70)
parameters: –
synonym: –
related to: accumulator
description: Sets the accumulator to the value of the field of index 3 of the accumulator.

GETFIELD (opcode: 71)
parameters: n
synonym: –
related to: accumulator
description: Sets the accumulator to the value of the field of index n of the accumulator.

GETFLOATFIELD (opcode: 72)
parameters: n
synonym: –
related to: accumulator
description: Sets the accumulator to the value of the field of index n of the accumulator (being a float).
SETFIELD0 (opcode: 73)
parameters: –
synonym: –
related to: accumulator stack
description: Sets the field of index 0 of the block in the accumulator to the value popped from the stack. Then sets the accumulator to the unit value.

SETFIELD1 (opcode: 74)
parameters: –
synonym: –
related to: accumulator stack
description: Sets the field of index 1 of the block in the accumulator to the value popped from the stack. Then sets the accumulator to the unit value.

SETFIELD2 (opcode: 75)
parameters: –
synonym: –
related to: accumulator stack
description: Sets the field of index 2 of the block in the accumulator to the value popped from the stack. Then sets the accumulator to the unit value.

SETFIELD3 (opcode: 76)
parameters: –
synonym: –
related to: accumulator stack
description: Sets the field of index 3 of the block in the accumulator to the value popped from the stack. Then sets the accumulator to the unit value.
SETFIELD (opcode: 77)

parameters: \( n \)

synonym: –

related to: accumulator stack

description: Sets the field of index \( n \) of the block in the accumulator to the value popped from the stack. Then sets the accumulator to the unit value.

SETFLOATFIELD (opcode: 78)

parameters: \( n \)

synonym: –

related to: accumulator stack

description: Sets the field of index \( n \) of the block in the accumulator to the (double) value popped from the stack. Then sets the accumulator to the unit value.

VECTLENGTH (opcode: 79)

parameters: –

synonym: –

related to: accumulator

description: Sets the accumulator to the size of the block in the accumulator.

GETVECTITEM (opcode: 80)

parameters: –

synonym: –

related to: accumulator stack

description: Pops an element from the stack, \( n \). Then sets the accumulator to the field of index \( n \) of the block in the accumulator.
SETVECTITEM (opcode: 81)
parameters: –
synonym: –
related to: accumulator stack
description: Pops two elements from the stack, \( n \) then \( v \). Then sets the field of index \( n \) of the block in the accumulator to \( v \). The accumulator is then set to the unit value.

GETSTRINGCHAR (opcode: 82)
parameters: –
synonym: –
related to: accumulator stack
description: Pops an element from the stack, \( n \). Sets the accumulator to the character of index \( n \) of the string contained in the accumulator.

SETSTRINGCHAR (opcode: 83)
parameters: –
synonym: –
related to: accumulator stack
description: Pops two elements from the stack, \( n \) then \( v \). Then sets the character of index \( n \) of the string in the accumulator to \( v \). The accumulator is set to the unit value.

BRANCH (opcode: 84)
parameters: \( ofs \)
synonym: –
related to: call
description: Performs an unconditional jump by adding \( ofs \) to \( pc \).

BRANCHIF (opcode: 85)
parameters: \( ofs \)
synonym: –
related to: accumulator call
description: Performs an conditional jump by adding \( ofs \) to \( pc \) if the accumulator is not zero.
BRANCHIFNOT (opcode: 86)

parameters: ofs

synonym: –

related to: accumulator call

description: Performs a conditional jump by adding ofs to pc if the accumulator is zero.

SWITCH (opcode: 87)

parameters: n, tab

synonym: –

related to: accumulator call

description: Given that $n = (\text{sizeTag} \ll 16) + \text{sizeLong}$, it defines the lookup tables tab that contains sizeTag + sizeLong entries. If the accumulator contains a long value, then pc is incremented by the value of tab of index idx, idx being the value of the accumulator. Otherwise, pc is incremented by the value of tab of index idx + sizeLong, idx being the tag of the accumulator block value.

BOOLNOT (opcode: 88)

parameters: –

synonym: –

related to: accumulator

description: Performs a boolean not on the accumulator.

PUSHTRAP (opcode: 89)

parameters: ofs

synonym: –

related to: stack environment

description: Pushes extraArgs, environment, trapSp and pc + ofs onto the stack. Then sets trapSp to the current sp value.
POPTRAP (opcode: 90)

parameters: –
synonym: –
related to: stack
description: Pops an element, then trapSp, then two other elements.

RAISE (opcode: 91)

parameters: –
synonym: –
related to: stack environment call
description: If no stack frame is defined, stops the execution printing the exception. Otherwise, restores sp from trapSp, then pc, trapSp, environment and extraArgs from four elements popped from stack.

CHECK-SIGNALS (opcode: 92)

parameters: –
synonym: –
related to: miscellaneous
description: Handles signals, if any.

C-CALL1 (opcode: 93)

parameters: p
synonym: –
related to: accumulator stack environment call
description: Pushes the environment, calls the primitive (originally in C) subroutine of identifier p with one parameter (the accumulator) and sets the accumulator to the return value of the primitive. Then pops the environment back.
C-CALL2 (opcode: 94)

parameters:  $p$

synonym: –

related to: accumulator stack environment call

description: Pushes the environment, calls the primitive (originally in C) subroutine of identifier $p$ with two parameters (the accumulator and the second element of the stack) and sets the accumulator to the return value of the primitive. Then pops the environment back and also pops one element.

C-CALL3 (opcode: 95)

parameters:  $p$

synonym: –

related to: accumulator stack environment call

description: Pushes the environment, calls the primitive (originally in C) subroutine of identifier $p$ with three parameters (the accumulator and the second and third elements of the stack) and sets the accumulator to the return value of the primitive. Then pops the environment back and also pops two elements.

C-CALL4 (opcode: 96)

parameters:  $p$

synonym: –

related to: accumulator stack environment call

description: Pushes the environment, calls the primitive (originally in C) subroutine of identifier $p$ with four parameters (the accumulator and the second, third and fourth elements of the stack) and sets the accumulator to the return value of the primitive. Then pops the environment back and also pops three elements.
C-CALL5 (opcode: 97)

parameters: $p$

synonym: –

related to: accumulator stack environment call

description: Pushes the environment, calls the primitive (originally in C) subroutine of identifier $p$ with five parameters (the accumulator and the second, third, fourth and fifth elements of the stack) and sets the accumulator to the return value of the primitive. Then pops the environment back and also pops four elements.

C-CALLN (opcode: 98)

parameters: $n, p$

synonym: –

related to: accumulator stack environment call

description: Pushes the accumulator and the environment, calls the primitive (originally in C) subroutine of identifier $p$ with $n$ parameters from the second element of the stack (towards bottom) and sets the accumulator to the return value of the primitive. Then pops the environment back and also pops $n$ elements.

CONST0 (opcode: 99)

parameters: –

synonym: –

related to: accumulator

description: Sets the accumulator to 0.

CONST1 (opcode: 100)

parameters: –

synonym: –

related to: accumulator

description: Sets the accumulator to 1.
CONST2 (opcode: 101)
parameters: –
synonym: –
related to: accumulator
description: Sets the accumulator to 2.

CONST3 (opcode: 102)
parameters: –
synonym: –
related to: accumulator
description: Sets the accumulator to 3.

CONSTINT (opcode: 103)
parameters: $n$
synonym: –
related to: accumulator
description: Sets the accumulator to $n$.

PUSHCONST0 (opcode: 104)
parameters: –
synonym: –
related to: accumulator stack
description: Pushes the value of the accumulator onto the stack. Then sets the accumulator to 0. Equivalent to PUSH then CONST0.

PUSHCONST1 (opcode: 105)
parameters: –
synonym: –
related to: accumulator stack
description: Pushes the value of the accumulator onto the stack. Then sets the accumulator to 1. Equivalent to PUSH then CONST1.
PUSHCONST2 (opcode: 106)

parameters: –
synonym: –
related to: accumulator stack
description: Pushes the value of the accumulator onto the stack.
Then sets the accumulator to 2.
Equivalent to PUSH then CONST2.

PUSHCONST3 (opcode: 107)

parameters: –
synonym: –
related to: accumulator stack
description: Pushes the value of the accumulator onto the stack.
Then sets the accumulator to 3.
Equivalent to PUSH then CONST3.

PUSHCONSTINT (opcode: 108)

parameters: \( n \)
synonym: –
related to: accumulator stack
description: Pushes the value of the accumulator onto the stack.
Then sets the accumulator to \( n \).
Equivalent to PUSH then CONSTINT.

NEGINT (opcode: 109)

parameters: –
synonym: –
related to: accumulator
description: Changes the value of the accumulator to its opposite.
ADDINT (opcode: 110)

parameters: –

synonym: –

related to: accumulator stack

description: Sets the accumulator to the sum of the accumulator and the value popped from the stack.

---

SUBINT (opcode: 111)

parameters: –

synonym: –

related to: accumulator stack

description: Sets the accumulator to the difference between the accumulator and the value popped from the stack.

---

MULINT (opcode: 112)

parameters: –

synonym: –

related to: accumulator stack

description: Sets the accumulator to the product of the accumulator by the value popped from the stack.

---

DIVINT (opcode: 113)

parameters: –

synonym: –

related to: accumulator stack

description: Sets the accumulator to the division of the accumulator by the value popped from the stack. Raises 'zero divide' exception if the value popped from the stack is equal to 0.
MODINT (opcode: 114)

parameters: –
synonym: –
related to: accumulator stack
description: Sets the accumulator to the modulo of the accumulator by the value popped from the stack. Raises 'zero divide' exception if the value popped from the stack is equal to 0.

\[ x \mod y \]

Stack evolution.

ANDINT (opcode: 115)

parameters: –
synonym: –
related to: accumulator stack
description: Performs a logical 'and' between the accumulator and the value popped from the stack. This value is stored into the accumulator.

\[ x \text{ and } y \]

Stack evolution.

ORINT (opcode: 116)

parameters: –
synonym: –
related to: accumulator stack
description: Performs a logical 'or' between the accumulator and the value popped from the stack. This value is stored into the accumulator.

\[ x \text{ or } y \]

Stack evolution.

XORINT (opcode: 117)

parameters: –
synonym: –
related to: accumulator stack
description: Performs a logical 'xor' between the accumulator and the value popped from the stack. This value is stored into the accumulator.

\[ x \text{ xor } y \]

Stack evolution.
LSLINT (opcode: 118)
parameters: –
synonym: –
related to: accumulator stack
description: Performs a logical ‘left shift’ of the accumulator by the value popped from the stack (does not preserve sign).

LSRINT (opcode: 119)
parameters: –
synonym: –
related to: accumulator stack
description: Performs a logical ‘right shift’ of the accumulator by the value popped from the stack (do not preserve sign).

ASRINT (opcode: 120)
parameters: –
synonym: –
related to: accumulator stack
description: Performs an arithmetic ‘right shift’ of the accumulator by the value popped from the stack (do preserve sign).

EQ (opcode: 121)
parameters: –
synonym: –
related to: accumulator stack
description: Sets the accumulator to a non-zero value or to zero whether the accumulator is equal to the value popped from the stack or not.
NEQ (opcode: 122)
parameters: –
synonym: –
related to: accumulator stack
description: Sets the accumulator to a non-zero value or to zero whether the accumulator is different from the value popped from the stack or not.

LTINT (opcode: 123)
parameters: –
synonym: –
related to: accumulator stack
description: Sets the accumulator to a non-zero value or to zero whether the accumulator is lower than the value popped from the stack or not.

LEINT (opcode: 124)
parameters: –
synonym: –
related to: accumulator stack
description: Sets the accumulator to a non-zero value or to zero whether the accumulator is lower than or equal to the value popped from the stack or not.

GTINT (opcode: 125)
parameters: –
synonym: –
related to: accumulator stack
description: Sets the accumulator to a non-zero value or to zero whether the accumulator is greater than the value popped from the stack or not.
GEINT (opcode: 126)
parameters: –
synonym: –
related to: accumulator stack
description: Sets the accumulator to a non-zero value or to zero whether the accumulator is greater than or equal to the value popped from the stack or not.

OFFSETINT (opcode: 127)
parameters: ofs
synonym: –
related to: accumulator
description: Adds ofs to the accumulator.

OFFSETREF (opcode: 128)
parameters: ofs
synonym: –
related to: accumulator
description: Adds ofs to the field of index 0 of the block in the accumulator. Then sets the accumulator to the unit value.

ISINT (opcode: 129)
parameters: –
synonym: –
related to: accumulator
description: Sets the accumulator to one or zero, whether the accumulator contains a long value or not.

GETMETHOD (opcode: 130)
parameters: –
synonym: –
related to: accumulator object
description: Peeks the top element of the stack as \( x \) and gets the field of index 0 from \( x \) as \( y \). Then sets the accumulator to the value of the \( z \)-th field of \( y \), \( z \) being the value of the accumulator.
BEQ (opcode: 131)
parameters: \(val, ofs\)
synonym: –
related to: accumulator call
description: Increments \(pc\) by \(ofs - 1\) if \(val\) is equal to the accumulator.

BNEQ (opcode: 132)
parameters: \(val, ofs\)
synonym: –
related to: accumulator call
description: Increments \(pc\) by \(ofs - 1\) if \(val\) is not equal to the accumulator.

BLTINT (opcode: 133)
parameters: \(val, ofs\)
synonym: –
related to: accumulator call
description: Increments \(pc\) by \(ofs - 1\) if \(val\) is lower than the accumulator.

BLEINT (opcode: 134)
parameters: \(val, ofs\)
synonym: –
related to: accumulator call
description: Increments \(pc\) by \(ofs - 1\) if \(val\) is lower than or equal to the accumulator.

BGTINT (opcode: 135)
parameters: \(val, ofs\)
synonym: –
related to: accumulator call
description: Increments \(pc\) by \(ofs - 1\) if \(val\) is greater than the accumulator.
BGEINT (opcode: 136)

parameters: \(val, ofs\)

synonym: –

related to: accumulator call

description: Increments \(pc\) by \(ofs - 1\) if \(val\) is greater than or equal to the accumulator.

ULTINT (opcode: 137)

parameters: –

synonym: –

related to: accumulator stack

description: Sets the accumulator to a non-zero value or to zero whether the accumulator is lower than the value popped from the stack or not (unsigned comparison).

UGEINT (opcode: 138)

parameters: –

synonym: –

related to: accumulator stack

description: Sets the accumulator to a non-zero value or to zero whether the accumulator is greater than or equal to the value popped from the stack or not (unsigned comparison).

BULTINT (opcode: 139)

parameters: \(val, ofs\)

synonym: –

related to: accumulator call

description: Increments \(pc\) by \(ofs - 1\) if \(val\) is lower than the accumulator (unsigned comparison).
BUGEINT (opcode: 140)
parameters: val, ofs
synonym: –
related to: accumulator call
description: Increments pc by ofs – 1 if val is greater than or equal to the accumulator (unsigned comparison).

GETPUBMET (opcode: 141)
parameters: tag, cache
synonym: –
related to: accumulator stack object
description: Pushes the accumulator (containing the object) onto the stack. Then gets the method corresponding to the given tag. The object is peeked at the top of the stack. The accumulator is set to the requested method. Method offset can be cached (implementation-dependent).

GETDYNMET (opcode: 142)
parameters: –
synonym: –
related to: accumulator object
description: Gets a method from a class. The object is peeked at the top of the stack. The method tag is in the accumulator. The accumulator is set to the requested method.

STOP (opcode: 143)
parameters: –
synonym: –
related to: accumulator
description: Stops the execution of the program. Returns the value of the accumulator to the caller.
EVENT (opcode: 144)
parameters: –
synonym: –
related to: debugger
description: Sends an 'event' message to the debugger.

BREAK (opcode: 145)
parameters: –
synonym: –
related to: debugger
description: Sends a 'break' message to the debugger.

Table of instructions

The following table lists the instructions sorted by ascending mnemonic, with opcode as well as categories an instruction belongs to. These categories (as the last seven columns) are:

accu. whether the accumulator is modified by the instruction;
stack whether the stack is modified by the instruction;
env. whether the environment is modified by the instruction;
call whether the instruction may produce a branch or call;
misc. whether the instruction fits in no other category;
obj. whether the instruction is related to objects;
debug. whether the instruction is related to debugger.

Table 1: Table of instructions (alphabetical order).

<table>
<thead>
<tr>
<th>mnemonic</th>
<th>opcode</th>
<th>accu.</th>
<th>stack</th>
<th>env.</th>
<th>call</th>
<th>misc.</th>
<th>obj.</th>
<th>debug.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>8</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ACC0</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ACC1</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ACC2</td>
<td>2</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ACC3</td>
<td>3</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ACC4</td>
<td>4</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ACC5</td>
<td>5</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ACC6</td>
<td>6</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ACC7</td>
<td>7</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td>.</td>
</tr>
</tbody>
</table>

continued on next page
<table>
<thead>
<tr>
<th>mnemonic</th>
<th>opcode</th>
<th>accu.</th>
<th>stack</th>
<th>env.</th>
<th>call</th>
<th>misc.</th>
<th>obj.</th>
<th>debug.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDINT</td>
<td>110</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ANDINT</td>
<td>115</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>APPLY</td>
<td>32</td>
<td>x</td>
<td>.</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>APPLY1</td>
<td>33</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>APPLY2</td>
<td>34</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>APPLY3</td>
<td>35</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>APPTERM</td>
<td>36</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>APPTERM1</td>
<td>37</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>APPTERM2</td>
<td>38</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>APPTERM3</td>
<td>39</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>ASRINT</td>
<td>120</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>ASSIGN</td>
<td>20</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>ATOM</td>
<td>59</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>ATOM0</td>
<td>58</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BEQ</td>
<td>131</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BGEINT</td>
<td>136</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BGTINT</td>
<td>135</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BLEINT</td>
<td>134</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BLTINT</td>
<td>133</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BNEQ</td>
<td>132</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BOOLNOT</td>
<td>88</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BRANCH</td>
<td>84</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BRANCHIF</td>
<td>85</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BRANCHIFNOT</td>
<td>86</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BREAK</td>
<td>145</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>x</td>
</tr>
<tr>
<td>BUGEINT</td>
<td>140</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>BULTINT</td>
<td>139</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>CHECK-SIGNALS</td>
<td>92</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>x</td>
</tr>
<tr>
<td>CLOSURE</td>
<td>43</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>CLOSUREREC</td>
<td>44</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>CONST0</td>
<td>99</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>CONST1</td>
<td>100</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>CONST2</td>
<td>101</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>CONST3</td>
<td>102</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>CONSTINT</td>
<td>103</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>C-CALL1</td>
<td>93</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>C-CALL2</td>
<td>94</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>C-CALL3</td>
<td>95</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>C-CALL4</td>
<td>96</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>C-CALL5</td>
<td>97</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>C-CALLN</td>
<td>98</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>DIVINT</td>
<td>113</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>ENVACC</td>
<td>25</td>
<td>x</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>mnemonic</td>
<td>opcode</td>
<td>accu.</td>
<td>stack</td>
<td>env.</td>
<td>call</td>
<td>misc.</td>
<td>obj.</td>
<td>debug.</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>ENVACC1</td>
<td>21</td>
<td>x</td>
<td></td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ENVACC2</td>
<td>22</td>
<td>x</td>
<td></td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ENVACC3</td>
<td>23</td>
<td>x</td>
<td></td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ENVACC4</td>
<td>24</td>
<td>x</td>
<td></td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>EQ</td>
<td>121</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>EVENT</td>
<td>144</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
</tr>
<tr>
<td>GEINT</td>
<td>126</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GETDYNMET</td>
<td>142</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
</tr>
<tr>
<td>GETFIELD</td>
<td>71</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GETFIELD0</td>
<td>67</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GETFIELD1</td>
<td>68</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GETFIELD2</td>
<td>69</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GETFIELD3</td>
<td>70</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GETFLOATFIELD</td>
<td>72</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GETGLOBAL</td>
<td>53</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GETGLOBALFIELD</td>
<td>55</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GETMETHOD</td>
<td>130</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
</tr>
<tr>
<td>GETPUBMET</td>
<td>141</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
</tr>
<tr>
<td>GETSTRINGCHAR</td>
<td>82</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GETVECTITEM</td>
<td>80</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GRAB</td>
<td>42</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>GTINT</td>
<td>125</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ISINT</td>
<td>129</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>LEINT</td>
<td>124</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>LSLINT</td>
<td>118</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>LSRINT</td>
<td>119</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>LTINT</td>
<td>123</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>MAKEBLOCK</td>
<td>62</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>MAKEBLOCK1</td>
<td>63</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>MAKEBLOCK2</td>
<td>64</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>MAKEBLOCK3</td>
<td>65</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>MAKEFLOATBLOCK</td>
<td>66</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>MODINT</td>
<td>114</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>MULINT</td>
<td>112</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>NEGINT</td>
<td>109</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>NEQ</td>
<td>122</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>OFFSETCLOSURE</td>
<td>48</td>
<td>x</td>
<td></td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>OFFSETCLOSURE0</td>
<td>46</td>
<td>x</td>
<td></td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>OFFSETCLOSURE2</td>
<td>47</td>
<td>x</td>
<td></td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>OFFSETCLOSUREM2</td>
<td>45</td>
<td>x</td>
<td></td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>OFFSETINT</td>
<td>127</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>OFFSETREF</td>
<td>128</td>
<td>x</td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ORINT</td>
<td>116</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>mnemonic</td>
<td>opcode</td>
<td>accu.</td>
<td>stack</td>
<td>env.</td>
<td>call</td>
<td>misc.</td>
<td>obj.</td>
<td>debug.</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>POP</td>
<td>19</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>POPTRAP</td>
<td>90</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSH</td>
<td>9</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHACC</td>
<td>18</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHACC0</td>
<td>10</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHACC1</td>
<td>11</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHACC2</td>
<td>12</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHACC3</td>
<td>13</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHACC4</td>
<td>14</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHACC5</td>
<td>15</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHACC6</td>
<td>16</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHACC7</td>
<td>17</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHATOM</td>
<td>61</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHATOM0</td>
<td>60</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHCONST0</td>
<td>104</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHCONST1</td>
<td>105</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHCONST2</td>
<td>106</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHCONST3</td>
<td>107</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHCONSTINT</td>
<td>108</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHENVACC</td>
<td>30</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHENVACC1</td>
<td>26</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHENVACC2</td>
<td>27</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHENVACC3</td>
<td>28</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHENVACC4</td>
<td>29</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHTGLOBAL</td>
<td>54</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHTGLOBALFIELD</td>
<td>56</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHOFFSETCLOSURE</td>
<td>52</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHOFFSETCLOSURE0</td>
<td>50</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHOFFSETCLOSURE2</td>
<td>51</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHOFFSETCLOSUREM2</td>
<td>49</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHTRAP</td>
<td>89</td>
<td>.</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>PUSHT-READDR</td>
<td>31</td>
<td>.</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>RAISE</td>
<td>91</td>
<td>.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>RESTART</td>
<td>41</td>
<td>.</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>RETURN</td>
<td>40</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>SETFIELD</td>
<td>77</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>SETFIELD0</td>
<td>73</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>SETFIELD1</td>
<td>74</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>SETFIELD2</td>
<td>75</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>SETFIELD3</td>
<td>76</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>SETFLOATFIELD</td>
<td>78</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>SETGLOBAL</td>
<td>57</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>SETSTRINGCHAR</td>
<td>83</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>mnemonic</td>
<td>opcode</td>
<td>accu.</td>
<td>stack</td>
<td>env.</td>
<td>call</td>
<td>misc.</td>
<td>obj.</td>
<td>debug.</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>SETVECTITEM</td>
<td>81</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>STOP</td>
<td>143</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>SUBINT</td>
<td>111</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>SWITCH</td>
<td>87</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>UGEINT</td>
<td>138</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>ULTINT</td>
<td>137</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>VECTLENGTH</td>
<td>79</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>XORINT</td>
<td>117</td>
<td>x</td>
<td>x</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

List of instructions per group

Instructions related to accumulator

- ACC
- ACC0
- ACC1
- ACC2
- ACC3
- ACC4
- ACC5
- ACC6
- ACC7
- ADDINT
- ANDINT
- APPLY
- APPLY1
- APPLY2
- APPLY3
- APPTERM
- APPTERM1
- APPTERM2
- APPTERM3
- ASRINT
• ASSIGN
• ATOM
• ATOM0
• BEQ
• BGEINT
• BGTINT
• BLEINT
• BLTINT
• BNEQ
• BOOLNOT
• BRANCHIF
• BRANCHIFNOT
• BUGEINT
• BULTINT
• CLOSURE
• CLOSUREREC
• CONST0
• CONST1
• CONST2
• CONST3
• CONSTINT
• C-CALL1
• C-CALL2
• C-CALL3
• C-CALL4
• C-CALL5
• C-CALLN
• DIVINT
• ENVACC
• ENVACC1
• ENVACC2
• ENVACC3
• ENVACC4
• EQ
• GEINT
• GETDYNMET
• GETFIELD
• GETFIELD0
• GETFIELD1
• GETFIELD2
• GETFIELD3
• GETFLOATFIELD
• GETGLOBAL
• GETGLOBALFIELD
• GETMETHOD
• GETPUBMET
• GETSTRINGCHAR
• GETVECTITEM
• GRAB
• GTINT
• ISINT
• LEINT
• LSLINT
• LSRINT
• LTINT
• MAKEBLOCK
• MAKEBLOCK1
• MAKEBLOCK2
• MAKEBLOCK3
• MAKEFLOATBLOCK
• MODINT
• MULINT
• NEGINT
• NEQ
• OFFSETCLOSURE
• OFFSETCLOSURE0
• OFFSETCLOSURE2
• OFFSETCLOSUREM2
• OFFSETINT
• OFFSETREF
• ORINT
• PUSH
• PUSHACC
• PUSHACC0
• PUSHACC1
• PUSHACC2
• PUSHACC3
• PUSHACC4
• PUSHACC5
• PUSHACC6
• PUSHACC7
• PUSHATOM
• PUSHATOM0
• PUSHCONST0
• PUSHCONST1
• PUSHCONST2
• PUSHCONST3
- PUSHCONSTINT
- PUSHENVACC
- PUSHENVACC1
- PUSHENVACC2
- PUSHENVACC3
- PUSHENVACC4
- PUSHGETGLOBAL
- PUSHGETGLOBALFIELD
- PUSHOFFSETCLOSURE
- PUSHOFFSETCLOSURE0
- PUSHOFFSETCLOSURE2
- PUSHOFFSETCLOSUREM2
- RETURN
- SETFIELD
- SETFIELD0
- SETFIELD1
- SETFIELD2
- SETFIELD3
- SETFLOATFIELD
- SETGLOBAL
- SETSTRINGCHAR
- SETVECTITEM
- STOP
- SUBINT
- SWITCH
- UGEINT
- ULINT
- VECTLENGTH
- XORINT
Instructions related to stack

- ACC
- ACC0
- ACC1
- ACC2
- ACC3
- ACC4
- ACC5
- ACC6
- ACC7
- ADDINT
- ANDINT
- APPLY1
- APPLY2
- APPLY3
- APPTERM
- APPTERM1
- APPTERM2
- APPTERM3
- ASRINT
- ASSIGN
- CLOSURE
- CLOSUREREC
- C-CALL1
- C-CALL2
- C-CALL3
- C-CALL4
- C-CALL5
- C-CALLN
• DIVINT
• EQ
• GEINT
• GETPUBMET
• GETSTRINGCHAR
• GETVECTITEM
• GRAB
• GTINT
• LEINT
• LSLINT
• LSRINT
• LTINT
• MAKEBLOCK
• MAKEBLOCK1
• MAKEBLOCK2
• MAKEBLOCK3
• MAKEFLOATBLOCK
• MODINT
• MULINT
• NEQ
• ORINT
• POP
• POPTRAP
• PUSH
• PUSHACC
• PUSHACC0
• PUSHACC1
• PUSHACC2
• PUSHACC3
• SETFIELD1
• SETFIELD2
• SETFIELD3
• SETFLOATFIELD
• SETSTRINGCHAR
• SETVECTITEM
• SUBINT
• UGEINT
• ULTINT
• XORINT

Instructions related to environment
• APPLY
• APPLY1
• APPLY2
• APPLY3
• APPTERM
• APPTERM1
• APPTERM2
• APPTERM3
• C-CALL1
• C-CALL2
• C-CALL3
• C-CALL4
• C-CALL5
• C-CALLN
• ENVACC
• ENVACC1
• ENVACC2
• ENVACC3
• ENVACC4
• GRAB
• OFFSETCLOSURE
• OFFSETCLOSURE0
• OFFSETCLOSURE2
• OFFSETCLOSUREM2
• PUSHENVACC
• PUSHENVACC1
• PUSHENVACC2
• PUSHENVACC3
• PUSHENVACC4
• PUSHOFFSETCLOSURE
• PUSHOFFSETCLOSURE0
• PUSHOFFSETCLOSURE2
• PUSHOFFSETCLOSUREM2
• PUSHTRAP
• RAISE
• RESTART
• RETURN

Instructions related to call/branch
• APPLY
• APPLY1
• APPLY2
• APPLY3
• APPTERM
• APPTERM1
• APPTERM2
• APPTERM3
• BEQ
• BGEINT
• BGTINT
• BLEINT
• BLTINT
• BNEQ
• BRANCH
• BRANCHIF
• BRANCHIFNOT
• BUGEINT
• BULTINT
• C-CALL1
• C-CALL2
• C-CALL3
• C-CALL4
• C-CALL5
• C-CALLN
• GRAB
• PUSH-RETADDR
• RAISE
• RETURN
• SWITCH

Instructions related to objects
• GETDYNMET
• GETMETHOD
• GETPUBMET

Instructions related to debugger
• BREAK
• EVENT

Miscellaneous instructions
• CHECK-SIGNALS