

Caml Virtual Machine — Instruction set

Document version: 1.4

<http://cadmium.x9c.fr>

Copyright © 2007-2010 Xavier Clerc – cadmium@x9c.fr
Released under the LGPL version 3

February 6, 2010

Abstract: This document describes the instruction set of the Caml¹ 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 *long* value corresponding to the `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 *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 *code offset* value, relative to code start (the code value of a block is the code offset value stored at index 0 in this block).

Along with the stack, the virtual machine contains seven registers:

pc program counter, that is next instruction to interpret;

sp stack pointer, that is top of the stack;

accu accumulator;

trapSp stack pointer of highest exception handler;

extraArgs number of extra arguments to function application;

env environment;

global global data.

¹The official Caml website can be reached at caml.inria.fr and contains the full development suite (compiler, tools, virtual machine, *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.

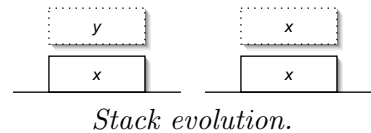
ACC0 (opcode: 0)

parameters: –

synonym: –

related to: [accumulator stack](#)

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



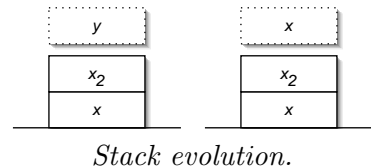
ACC1 (opcode: 1)

parameters: –

synonym: –

related to: [accumulator stack](#)

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



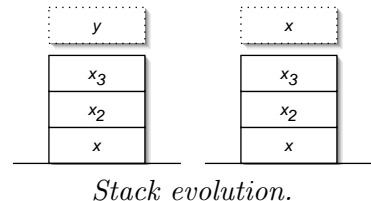
ACC2 (opcode: 2)

parameters: –

synonym: –

related to: [accumulator stack](#)

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



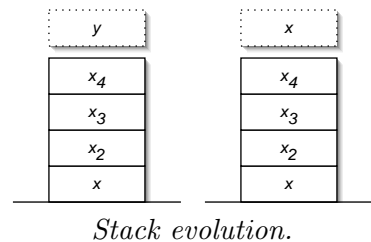
ACC3 (opcode: 3)

parameters: –

synonym: –

related to: [accumulator stack](#)

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



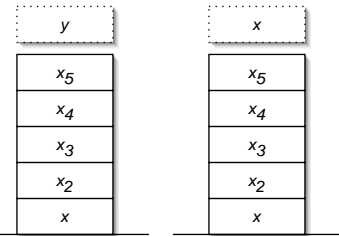
ACC4 (opcode: 4)

parameters: –

synonym: –

related to: [accumulator stack](#)

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



Stack evolution.

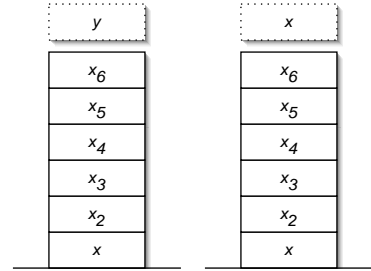
ACC5 (opcode: 5)

parameters: –

synonym: –

related to: [accumulator stack](#)

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



Stack evolution.

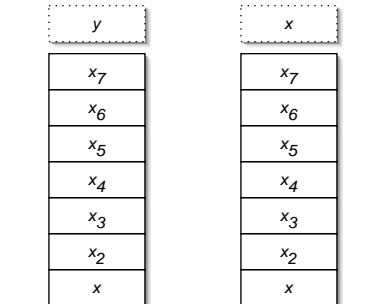
ACC6 (opcode: 6)

parameters: –

synonym: –

related to: [accumulator stack](#)

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



Stack evolution.

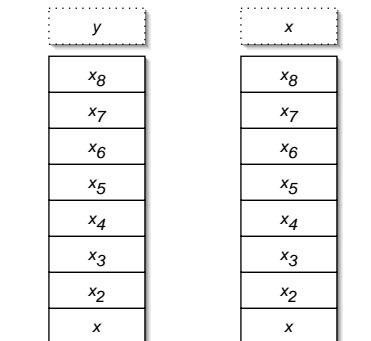
ACC7 (opcode: 7)

parameters: –

synonym: –

related to: [accumulator stack](#)

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



Stack evolution.

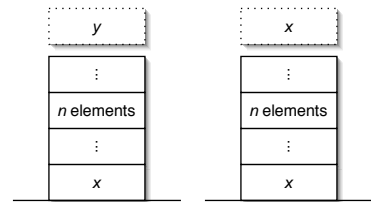
ACC (opcode: 8)

parameters: n

synonym: –

related to: [accumulator stack](#)

description: Peeks the $n+1$ -th element of the stack and puts it into the accumulator.



Stack evolution.

PUSH (opcode: 9)

parameters: –

synonym: [PUSHACC0](#)

related to: [accumulator stack](#)

description: Pushes the accumulator onto the stack.



Stack evolution.

PUSHACC0 (opcode: 10)

parameters: –

synonym: [PUSH](#)

related to: [accumulator stack](#)

description: Pushes the accumulator onto the stack.
Equivalent to [PUSH](#) then [ACC0](#).



Stack evolution.

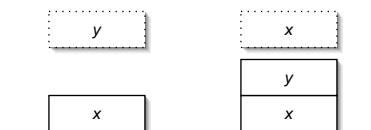
PUSHACC1 (opcode: 11)

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](#).



Stack evolution.

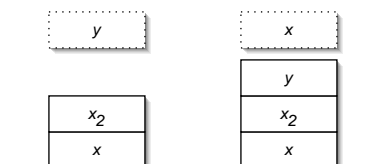
PUSHACC2 (opcode: 12)

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](#).



Stack evolution.

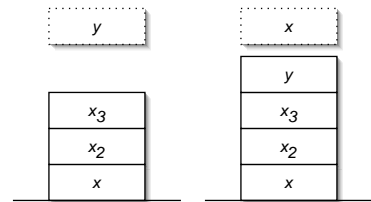
PUSHACC3 (opcode: 13)

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](#).



Stack evolution.

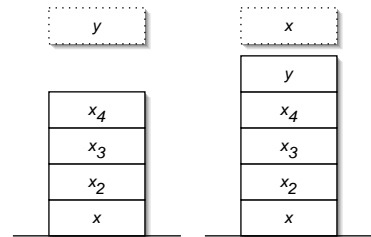
PUSHACC4 (opcode: 14)

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](#).



Stack evolution.

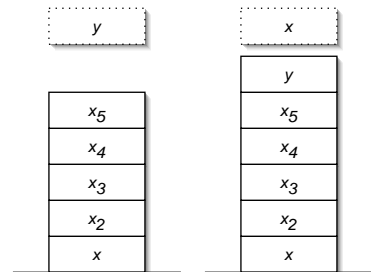
PUSHACC5 (opcode: 15)

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](#).



Stack evolution.

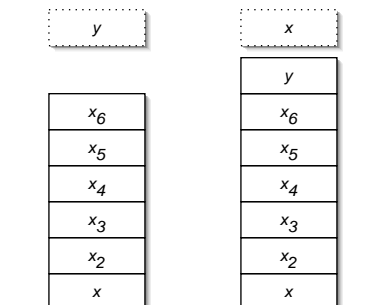
PUSHACC6 (opcode: 16)

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](#).



Stack evolution.

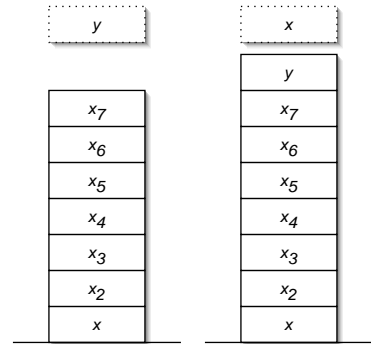
PUSHACC7 (opcode: 17)

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](#).



Stack evolution.

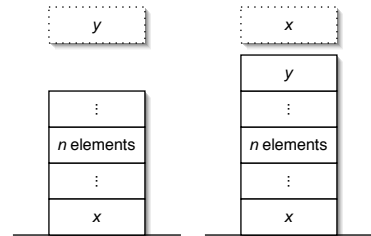
PUSHACC (opcode: 18)

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](#).



Stack evolution.

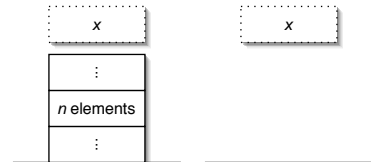
POP (opcode: 19)

parameters: n

synonym: –

related to: [stack](#)

description: Pops n elements from the stack.



Stack evolution.

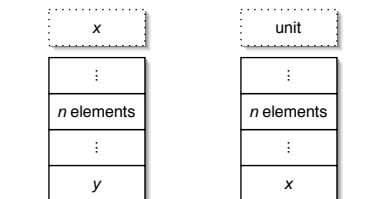
ASSIGN (opcode: 20)

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.



Stack evolution.

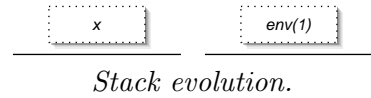
ENVACC1 (opcode: 21)

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.



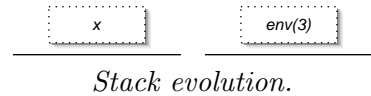
ENVACC3 (opcode: 23)

parameters: –

synonym: –

related to: [accumulator environment](#)

description: Sets the accumulator to the field of index 3 of the environment.



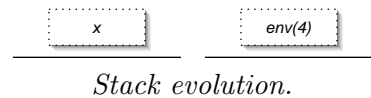
ENVACC4 (opcode: 24)

parameters: –

synonym: –

related to: [accumulator environment](#)

description: Sets the accumulator to the field of index 4 of the environment.



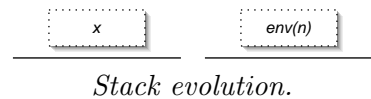
ENVACC (opcode: 25)

parameters: n

synonym: –

related to: [accumulator environment](#)

description: Sets the accumulator to the field of index n of the environment.



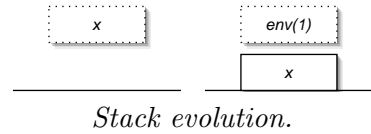
PUSHENVACC1 (opcode: 26)

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](#).



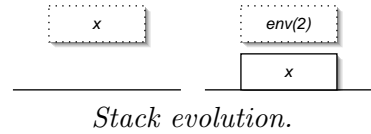
PUSHENVACC2 (opcode: 27)

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](#).



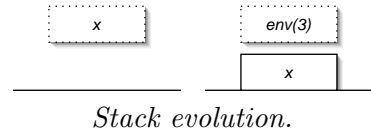
PUSHENVACC3 (opcode: 28)

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](#).



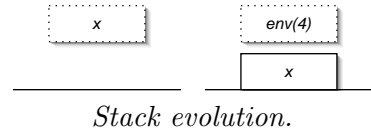
PUSHENVACC4 (opcode: 29)

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](#).



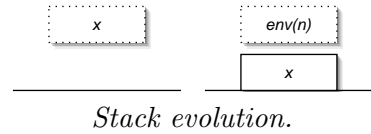
PUSHENVACC (opcode: 30)

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](#).



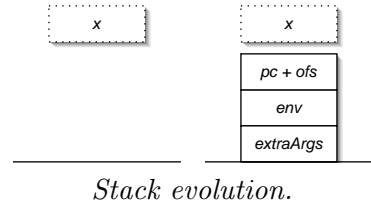
PUSH-RETADDR (opcode: 31)

parameters: ofs

synonym: –

related to: [stack call](#)

description: Pushes $extraArgs$, then the environment, and then $pc + ofs$.



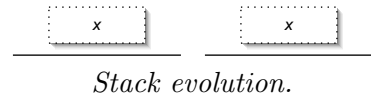
APPLY (opcode: 32)

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.



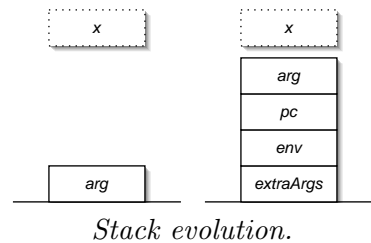
APPLY1 (opcode: 33)

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.



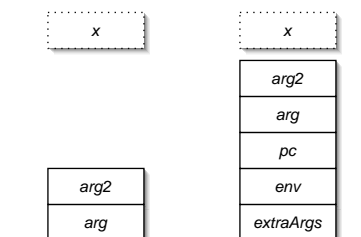
APPLY2 (opcode: 34)

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.



Stack evolution.

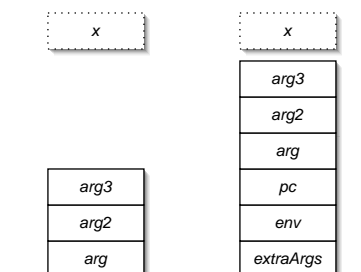
APPLY3 (opcode: 35)

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.



Stack evolution.

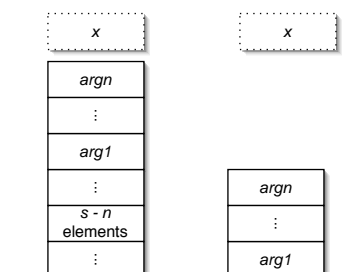
APPTERM (opcode: 36)

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 $n - 1$.



Stack evolution.

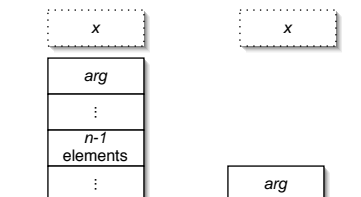
APPTERM1 (opcode: 37)

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.



Stack evolution.

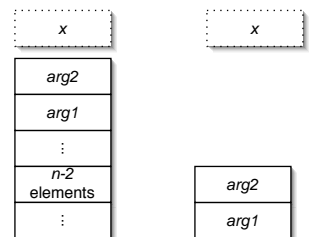
APPTERM2 (opcode: 38)

parameters: n

synonym: –

related to: [accumulator stack environment call](#)

description: Peeks the top elements from the stack as $arg1$ and $arg2$, pops $n-2$ elements from the stacks, and pushes back $arg1$ and $arg2$. Then sets pc to the code value of the accumulator, the environment to the accumulator, and increments $extraArgs$.



Stack evolution.

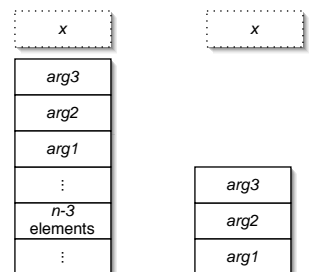
APPTERM3 (opcode: 39)

parameters: n

synonym: –

related to: [accumulator stack environment call](#)

description: Peeks the top elements from the stack as $arg1$, $arg2$ and $arg3$, pops $n-3$ elements from the stacks, and pushes back $arg1$, $arg2$ and $arg3$. Then sets pc to the code value of the accumulator, the environment to the accumulator, and adds two to $extraArgs$.



Stack evolution.

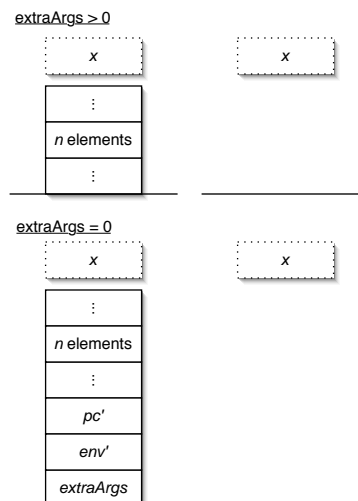
RETURN (opcode: 40)

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)

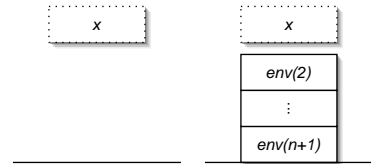
parameters: –

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 *extraArgs* is increased by n .



Stack evolution.

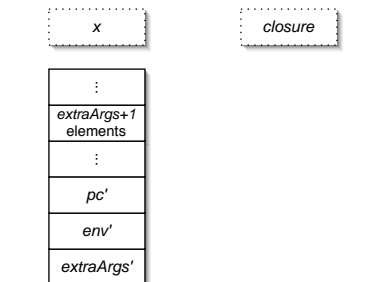
GRAB (opcode: 42)

parameters: n

synonym: –

related to: [accumulator stack environment call](#)

description: If *extraArgs* is greater than or equal to n , then *extraArgs* is decreased by n . Otherwise, creates a closure of $extraArgs + 3$ 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 *extraArgs* are popped from the stack.



Stack evolution.

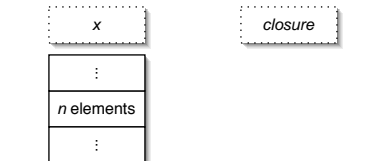
CLOSURE (opcode: 43)

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.



Stack evolution.

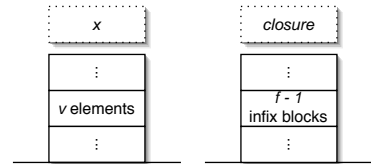
CLOSUREREC (opcode: 44)

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).



Stack evolution.

OFFSETCLOSUREM2 (opcode: 45)

parameters: –

synonym: –

related to: [accumulator environment](#)

description: Sets the accumulator to the value of the closure preceding the environment.



Stack evolution.

OFFSETCLOSURE0 (opcode: 46)

parameters: –

synonym: –

related to: [accumulator environment](#)

description: Sets the accumulator to the value of the environment.



Stack evolution.

OFFSETCLOSURE2 (opcode: 47)

parameters: –

synonym: –

related to: [accumulator environment](#)

description: Sets the accumulator to the value of the closure following the environment.



Stack evolution.

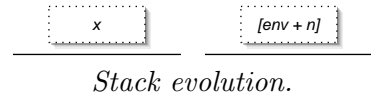
OFFSETCLOSURE (opcode: 48)

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)

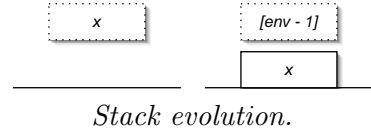
parameters: –

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)

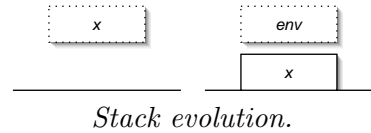
parameters: –

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)

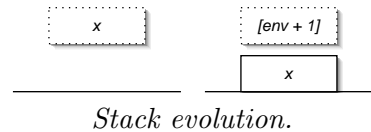
parameters: –

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](#).



Stack evolution.

GETGLOBAL (opcode: 53)

parameters: n

synonym: –

related to: [accumulator](#)

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



Stack evolution.

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](#) then [GETGLOBAL](#).



Stack evolution.

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.



Stack evolution.

PUSHGETGLOBALFIELD (opcode: 56)

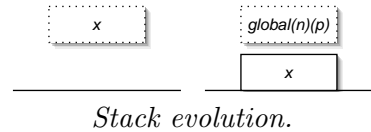
parameters: n, p

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](#).



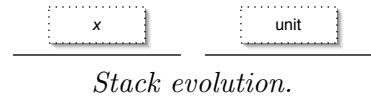
SETGLOBAL (opcode: 57)

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.



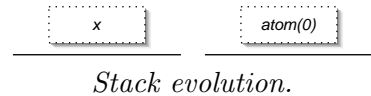
ATOM0 (opcode: 58)

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](#).



Stack evolution.

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](#).



Stack evolution.

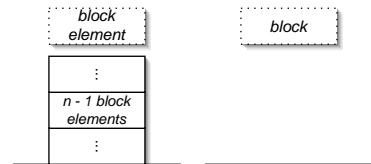
MAKEBLOCK (opcode: 62)

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.



Stack evolution.

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.



Stack evolution.

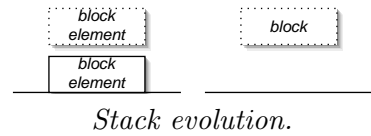
MAKEBLOCK2 (opcode: 64)

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.



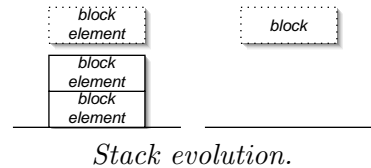
MAKEBLOCK3 (opcode: 65)

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.



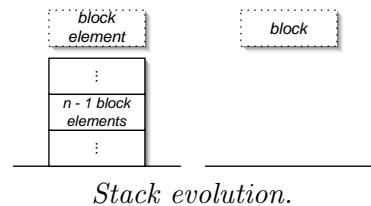
MAKEFLOATBLOCK (opcode: 66)

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.



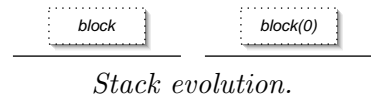
GETFIELD0 (opcode: 67)

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.



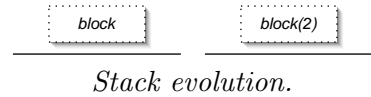
GETFIELD2 (opcode: 69)

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.



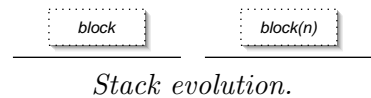
GETFIELD (opcode: 71)

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).



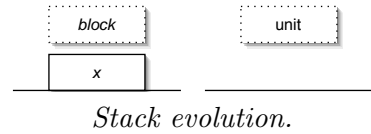
SETFIELD0 (opcode: 73)

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.



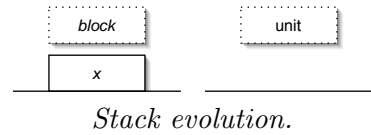
SETFIELD1 (opcode: 74)

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.



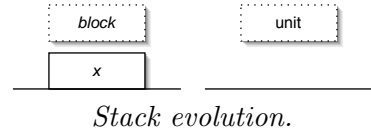
SETFIELD2 (opcode: 75)

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.



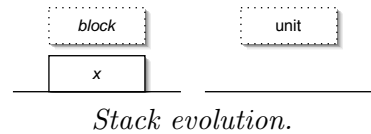
SETFIELD3 (opcode: 76)

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.



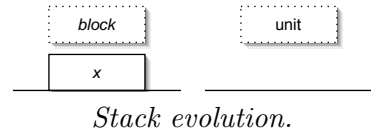
SETFIELD (opcode: 77)

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.



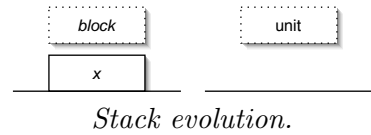
SETFLOATFIELD (opcode: 78)

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.



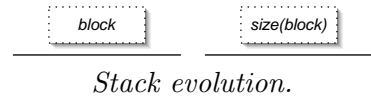
VECTLENGTH (opcode: 79)

parameters: –

synonym: –

related to: [accumulator](#)

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



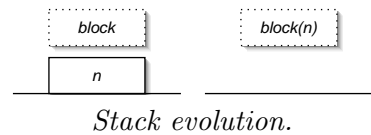
GETVECTITEM (opcode: 80)

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.



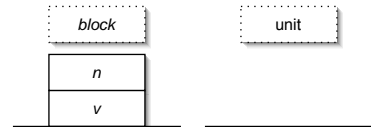
SETVECTITEM (opcode: 81)

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.



Stack evolution.

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.



Stack evolution.

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.



Stack evolution.

BRANCH (opcode: 84)

parameters: ofs

synonym: –

related to: [call](#)

description: Performs an unconditional jump by adding ofs to pc .



Stack evolution.

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.



Stack evolution.

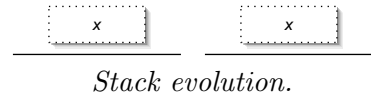
BRANCHIFNOT (opcode: 86)

parameters: ofs

synonym: –

related to: [accumulator call](#)

description: Performs an 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 $n = (sizeTag \ll 16) + sizeLong$, 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.



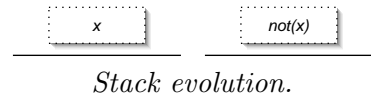
BOOLNOT (opcode: 88)

parameters: –

synonym: –

related to: [accumulator](#)

description: Performs a boolean not on the accumulator.



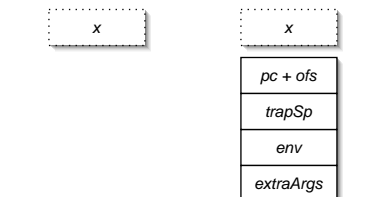
PUSHTRAP (opcode: 89)

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.



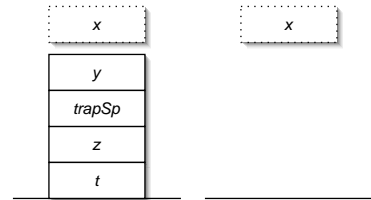
POPTRAP (opcode: 90)

parameters: –

synonym: –

related to: [stack](#)

description: Pops an element, then *trapSp*, then two other elements.



Stack evolution.

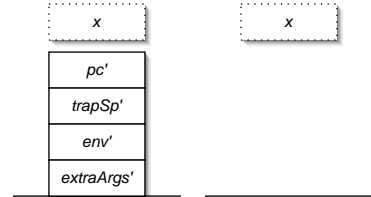
RAISE (opcode: 91)

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.



Stack evolution.

CHECK-SIGNALS (opcode: 92)

parameters: –

synonym: –

related to: [miscellaneous](#)

description: Handles signals, if any.



Stack evolution.

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.



Stack evolution.

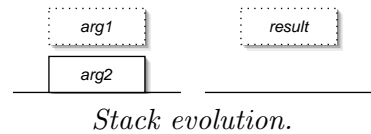
C-CALL2 (opcode: 94)

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.



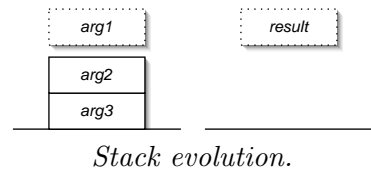
C-CALL3 (opcode: 95)

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.



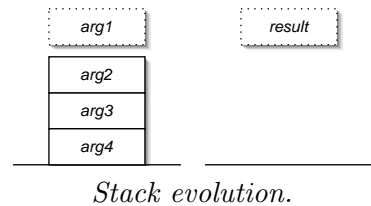
C-CALL4 (opcode: 96)

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.



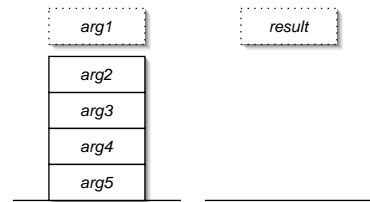
C-CALL5 (opcode: 97)

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.



Stack evolution.

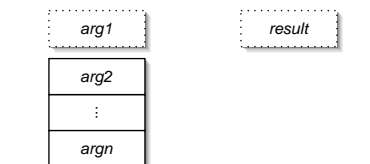
C-CALLN (opcode: 98)

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.



Stack evolution.

CONST0 (opcode: 99)

parameters: –

synonym: –

related to: [accumulator](#)

description: Sets the accumulator to 0.



Stack evolution.

CONST1 (opcode: 100)

parameters: –

synonym: –

related to: [accumulator](#)

description: Sets the accumulator to 1.



Stack evolution.

CONST2 (opcode: 101)

parameters: –

synonym: –

related to: [accumulator](#)

description: Sets the accumulator to 2.



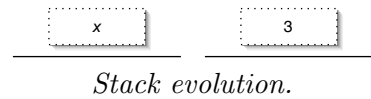
CONST3 (opcode: 102)

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: –

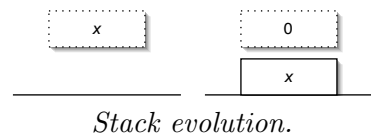
synonym: –

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: –

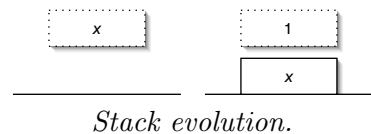
synonym: –

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](#).



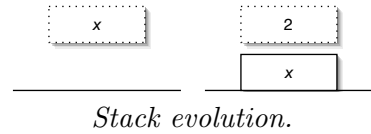
PUSHCONST2 (opcode: 106)

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](#).



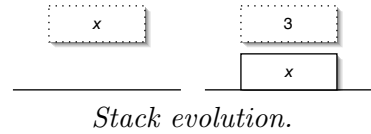
PUSHCONST3 (opcode: 107)

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](#).



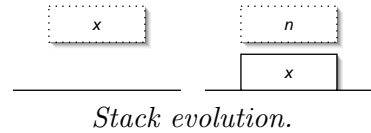
PUSHCONSTINT (opcode: 108)

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](#).



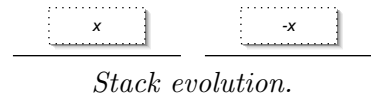
NEGINT (opcode: 109)

parameters: –

synonym: –

related to: [accumulator](#)

description: Changes the value of the accumulator to its opposite.



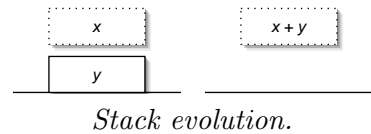
ADDINT (opcode: 110)

parameters: –

synonym: –

related to: [accumulator stack](#)

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



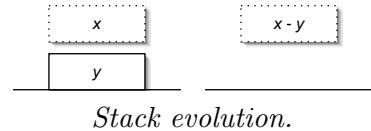
SUBINT (opcode: 111)

parameters: –

synonym: –

related to: [accumulator stack](#)

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



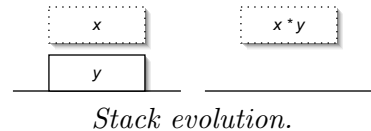
MULINT (opcode: 112)

parameters: –

synonym: –

related to: [accumulator stack](#)

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



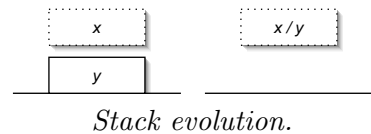
DIVINT (opcode: 113)

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.



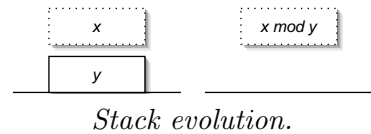
MODINT (opcode: 114)

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.



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.



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.



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.



Stack evolution.

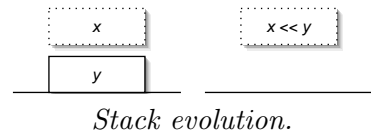
LSLINT (opcode: 118)

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).



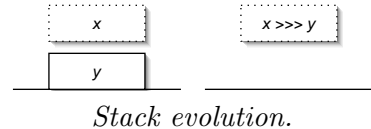
LSRINT (opcode: 119)

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).



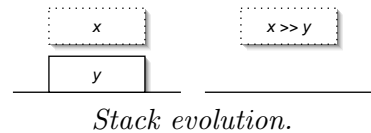
ASRINT (opcode: 120)

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).



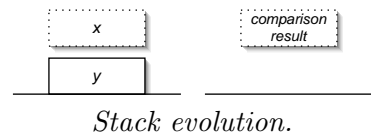
EQ (opcode: 121)

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.



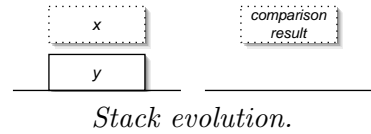
NEQ (opcode: 122)

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.



Stack evolution.

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.



Stack evolution.

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.



Stack evolution.

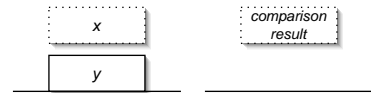
GTINT (opcode: 125)

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.



Stack evolution.

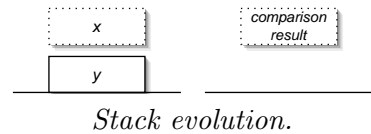
GEINT (opcode: 126)

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.



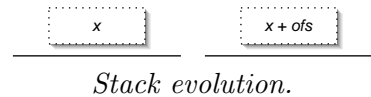
OFFSETINT (opcode: 127)

parameters: *ofs*

synonym: –

related to: [accumulator](#)

description: Adds *ofs* to the accumulator.



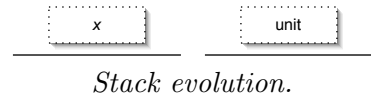
OFFSETREF (opcode: 128)

parameters: *ofs*

synonym: –

related to: [accumulator](#)

description: Adds *ofs* to to the field of index 0 of the block in the accumulator. Then sets the accumulator to the unit value.



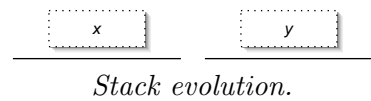
ISINT (opcode: 129)

parameters: –

synonym: –

related to: [accumulator](#)

description: Sets the accumulator to one or zero, whether the accumulator contains a long value or not.



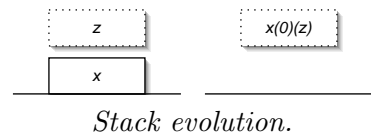
GETMETHOD (opcode: 130)

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.



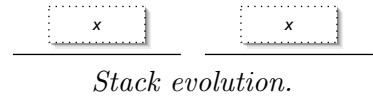
BEQ (opcode: 131)

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.



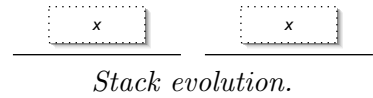
BLTINT (opcode: 133)

parameters: val, ofs

synonym: –

related to: [accumulator call](#)

description: Increments pc by $ofs - 1$ if val is lower than the accumulator.



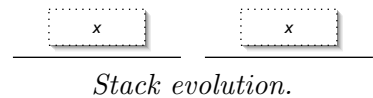
BLEINT (opcode: 134)

parameters: val, ofs

synonym: –

related to: [accumulator call](#)

description: Increments pc by $ofs - 1$ if val is lower than or equal to the accumulator.



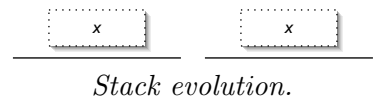
BGTINT (opcode: 135)

parameters: val, ofs

synonym: –

related to: [accumulator call](#)

description: Increments pc by $ofs - 1$ if val is greater than the accumulator.



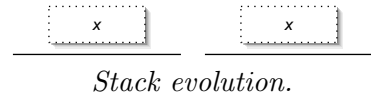
BGEINT (opcode: 136)

parameters: *val, ofs*

synonym: –

related to: [accumulator call](#)

description: Increments *pc* by *ofs* – 1 if *val* is greater than or equal to the accumulator.



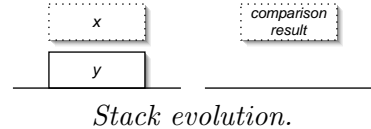
ULTINT (opcode: 137)

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).



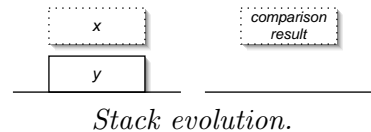
UGEINT (opcode: 138)

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).



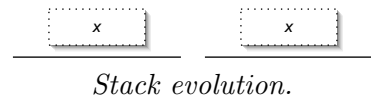
BULTINT (opcode: 139)

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).



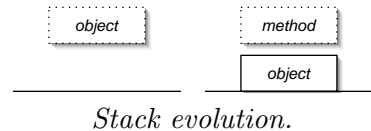
GETPUBMET (opcode: 141)

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).



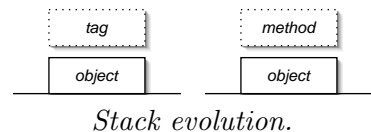
GETDYNMET (opcode: 142)

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.

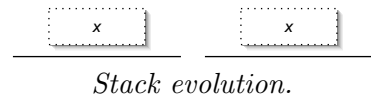


Table of instructions

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).

mnemonic	opcode	accu.	stack	env.	call	misc.	obj.	debug.
ACC	8	x	x
ACC0	0	x	x
ACC1	1	x	x
ACC2	2	x	x
ACC3	3	x	x
ACC4	4	x	x
ACC5	5	x	x
ACC6	6	x	x
ACC7	7	x	x

continued on next page

continued from previous page

mnemonic	opcode	accu.	stack	env.	call	misc.	obj.	debug.
ADDINT	110	x	x
ANDINT	115	x	x
APPLY	32	x	.	x	x	.	.	.
APPLY1	33	x	x	x	x	.	.	.
APPLY2	34	x	x	x	x	.	.	.
APPLY3	35	x	x	x	x	.	.	.
APPTERM	36	x	x	x	x	.	.	.
APPTERM1	37	x	x	x	x	.	.	.
APPTERM2	38	x	x	x	x	.	.	.
APPTERM3	39	x	x	x	x	.	.	.
ASRINT	120	x	x
ASSIGN	20	x	x
ATOM	59	x
ATOM0	58	x
BEQ	131	x	.	.	x	.	.	.
BGEINT	136	x	.	.	x	.	.	.
BGTINT	135	x	.	.	x	.	.	.
BLEINT	134	x	.	.	x	.	.	.
BLTINT	133	x	.	.	x	.	.	.
BNEQ	132	x	.	.	x	.	.	.
BOOLNOT	88	x
BRANCH	84	.	.	.	x	.	.	.
BRANCHIF	85	x	.	.	x	.	.	.
BRANCHIFNOT	86	x	.	.	x	.	.	.
BREAK	145	x
BUGEINT	140	x	.	.	x	.	.	.
BULTINT	139	x	.	.	x	.	.	.
CHECK-SIGNALS	92	x	.	.
CLOSURE	43	x	x
CLOSUREREC	44	x	x
CONST0	99	x
CONST1	100	x
CONST2	101	x
CONST3	102	x
CONSTINT	103	x
C-CALL1	93	x	x	x	x	.	.	.
C-CALL2	94	x	x	x	x	.	.	.
C-CALL3	95	x	x	x	x	.	.	.
C-CALL4	96	x	x	x	x	.	.	.
C-CALL5	97	x	x	x	x	.	.	.
C-CALLN	98	x	x	x	x	.	.	.
DIVINT	113	x	x
ENVACC	25	x	.	x

continued on next page

continued from previous page

mnemonic	opcode	accu.	stack	env.	call	misc.	obj.	debug.
ENVACC1	21	x	.	x
ENVACC2	22	x	.	x
ENVACC3	23	x	.	x
ENVACC4	24	x	.	x
EQ	121	x	x
EVENT	144	x
GEINT	126	x	x
GETDYNMET	142	x	x	.
GETFIELD	71	x
GETFIELD0	67	x
GETFIELD1	68	x
GETFIELD2	69	x
GETFIELD3	70	x
GETFLOATFIELD	72	x
GETGLOBAL	53	x
GETGLOBALFIELD	55	x
GETMETHOD	130	x	x	.
GETPUBMET	141	x	x	.	.	.	x	.
GETSTRINGCHAR	82	x	x
GETVECTITEM	80	x	x
GRAB	42	x	x	x	x	.	.	.
GTINT	125	x	x
ISINT	129	x
LEINT	124	x	x
LSLINT	118	x	x
LSRINT	119	x	x
LTINT	123	x	x
MAKEBLOCK	62	x	x
MAKEBLOCK1	63	x	x
MAKEBLOCK2	64	x	x
MAKEBLOCK3	65	x	x
MAKEFLOATBLOCK	66	x	x
MODINT	114	x	x
MULINT	112	x	x
NEGINT	109	x
NEQ	122	x	x
OFFSETCLOSURE	48	x	.	x
OFFSETCLOSURE0	46	x	.	x
OFFSETCLOSURE2	47	x	.	x
OFFSETCLOSUREM2	45	x	.	x
OFFSETINT	127	x
OFFSETREF	128	x
ORINT	116	x	x

continued on next page

continued from previous page

mnemonic	opcode	accu.	stack	env.	call	misc.	obj.	debug.
POP	19	.	x
POPTRAP	90	.	x
PUSH	9	x	x
PUSHACC	18	x	x
PUSHACC0	10	x	x
PUSHACC1	11	x	x
PUSHACC2	12	x	x
PUSHACC3	13	x	x
PUSHACC4	14	x	x
PUSHACC5	15	x	x
PUSHACC6	16	x	x
PUSHACC7	17	x	x
PUSHATOM	61	x	x
PUSHATOM0	60	x	x
PUSHCONST0	104	x	x
PUSHCONST1	105	x	x
PUSHCONST2	106	x	x
PUSHCONST3	107	x	x
PUSHCONSTINT	108	x	x
PUSHENVACC	30	x	x	x
PUSHENVACC1	26	x	x	x
PUSHENVACC2	27	x	x	x
PUSHENVACC3	28	x	x	x
PUSHENVACC4	29	x	x	x
PUSHGETGLOBAL	54	x	x
PUSHGETGLOBALFIELD	56	x	x
PUSHOFFSETCLOSURE	52	x	x	x
PUSHOFFSETCLOSURE0	50	x	x	x
PUSHOFFSETCLOSURE2	51	x	x	x
PUSHOFFSETCLOSUREM2	49	x	x	x
PUSHTRAP	89	.	x	x
PUSH-RETADDR	31	.	x	.	x	.	.	.
RAISE	91	.	x	x	x	.	.	.
RESTART	41	.	x	x
RETURN	40	x	x	x	x	.	.	.
SETFIELD	77	x	x
SETFIELD0	73	x	x
SETFIELD1	74	x	x
SETFIELD2	75	x	x
SETFIELD3	76	x	x
SETFLOATFIELD	78	x	x
SETGLOBAL	57	x
SETSTRINGCHAR	83	x	x

continued on next page

continued from previous page

mnemonic	opcode	accu.	stack	env.	call	misc.	obj.	debug.
SETVECTITEM	81	x	x
STOP	143	x
SUBINT	111	x	x
SWITCH	87	x	.	.	x	.	.	.
UGEINT	138	x	x
ULTINT	137	x	x
VECTLENGTH	79	x
XORINT	117	x	x

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
- ULTINT
- 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

- PUSHACC4
- PUSHACC5
- PUSHACC6
- PUSHACC7
- PUSHATOM
- PUSHATOM0
- PUSHCONST0
- PUSHCONST1
- PUSHCONST2
- PUSHCONST3
- PUSHCONSTINT
- PUSHENVACC
- PUSHENVACC1
- PUSHENVACC2
- PUSHENVACC3
- PUSHENVACC4
- PUSHGETGLOBAL
- PUSHGETGLOBALFIELD
- PUSHOFFSETCLOSURE
- PUSHOFFSETCLOSURE0
- PUSHOFFSETCLOSURE2
- PUSHOFFSETCLOSUREM2
- PUSHTRAP
- PUSH-RETADDR
- RAISE
- RESTART
- RETURN
- SETFIELD
- SETFIELD0

- 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