CSC488 Pseudo Machine Description

The Machine Components (See `Machine.java` for details)

main memory Main memory consists of *memorySize* 16-bit words, addressed from 0 to *memorySize* -1.
The value stored in a word may be interpreted in any of the following ways.

Integer	A two's complement integer value in the range -32767 to +32767.
Boolean	A boolean value. False is represented by the value MACHINE_FALSE. True is represented by the value MACHINE_TRUE.
character	A character value (one character per word). The representation is ASCII, extended to 16 bits by prefixing 8 zeros.
undefined	The undefined data value represented by the value UNDEFINED.
instruction	An operation code. The operations and codes are listed later.
address	A memory address, an integer value in the range 0 .. memorySize - 1

program counter A 16-bit program counter *pc* which contains the address of the instruction being executed.

mlp pointer A 16-bit memory mlp pointer *mlp* which contains the address of the last word in memory available for the run time stack.

msp pointer A 16-bit memory stack pointer *msp* which contains the address of the next free word in the run time stack.

display An array of 16-bit **display** registers, indexed from 0 to *displaySize* -1.
This array of registers is used to implement addressing into the run time stack.

See the *Code Generation Model* figure in the Code Generation handout for a description of how the machine is intended to be used. Like real hardware the machine interpreter is absolutely *paranoid* about checking for errors during instruction execution. It will raise an exception if any error is detected. The pseudo instructions, TRON, TROFF and ILIMIT, are provided as aids to assist in the debugging of the code generator.

Initial Machine State

At the start of execution, the following may be assumed:

a) The instructions of a program should occupy contiguous *memory* locations starting at address 0.
b) Register *msp* contains the address of the first word after the instructions.
c) *memory* between *msp* and *mlp* contains UNDEFINED.
d) The *display* is empty.
e) *pc* contains the address of the first instruction to be executed.

The area of memory between the initial *msp* and *mlp* is used as a runtime stack. Storage for the main program and all functions and procedures (activation records) are allocated here. Display entries created by the program are used to address this storage. The run stack space immediately above the last allocated storage is used as temporary space for evaluating expressions.

An instruction may occupy one, two, or three words of memory. The first word contains an op-code, specifying which machine operation to perform. The second and third words, if present, contain operands.

Machine Instructions

In this description, push, pop, and top refer to the run time stack whose top pointer is *msp*.
a, v, n, x and *y* are integer temporary variables local to the description. a is usually an address.

OpCode	Name	Args	Description
0	ADDR	LL ON	push(display[LL] + ON) % push stack address onto stack
1	LOAD		a:= top; pop; % load stack from memory if memory[a] = undefined then error; push(memory[a])
2	STORE		v:= top; pop; % store from stack to memory a:= top; pop; memory[a]:= v
3	PUSH	V	push(V) % push literal constant V
4	PUSHMT		push(msp) % push stack pointer onto stack
5	SETD	LL	display[LL]:= top; pop % set display entry from stack
6	POPN		n:= top; pop; pop^n % n-fold pop
7	POP		pop
8	DUPN		n:= top; pop; v:= top; pop; $push^n$(v) % n-fold duplicate
9	DUP		push(top)
10	BR		a:= top; pop; go to a % unconditional branch
11	BF		a:= top; pop; v:= top; pop; if v = FALSE then go to a
12	NEG		top := - top ;
13	ADD		For operations (**op**) 13-19:
14	SUB		y:= top; pop;
15	MUL		x:= top; pop;
16	DIV		n := x **op** y ;
17	EQ		if overflow or zero divide then error else push(n) ;
18	LT		Less than compare
19	OR		Boolean or
20	SWAP		x:= top; pop; y:= top; pop; push(x); push(y)
21	READC		One character of input is read and pushed.
22	PRINTC		Print top as character; pop
23	READI		The next integer value on the interpreters standard input is read in as an integer value n. push (n).
24	PRINTI		Print top as integer; pop
25	HALT		Halt

Pseudo Instructions

26	TRON		Start tracing the execution of machine instructions if tracing has been enabled.
27	TROFF		Stop tracing the execution of machine instructions. No-op if tracing not enabled.
28	ILIMIT	V	Set a limit on number of instructions to execute to V. V <= 0 sets the limit to ∞ Execution halts when V instructions have been executed after ILIMIT.