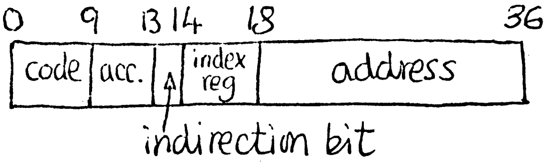
WARREN

Synopsis of the DEC System 10

word : 36 bits
address : 18 bits

instruction/pointer format:



symbolic forms:

CODE ACC, E
or CODE E (if ACC=0)

where E is either (i) ADDRESS
or (ii) ADDR (INDEXREG) *
or (iii) @ ADDRESS
or (iv) @ ADDR(INDEXREG) *;

for (i) and (ii) indirection bit is 0
for (i) and (iii) index reg is 0.

Calculation of the effective address E :–

E = ADDR + (contents of INDEXREG if INDEXREG ≠ 0)
if indirection bit is zero.

If the indirection bit is one, fetch the word from the above address and iterate the procedure until a word is encountered with a zero indirection bit.

fast registers: addresses 0 to 15_{10}.

* In the compiler clauses, the notation INDEXREG(ADDR) is used.

Summary of the Instructions used in Compiled Code

JRST	E	goto E
JUMPE	A,E	if $(A)=0$ then goto E
JUMPGE	A,E	if $(A)\geq 0$ then goto E
AOBJP	A,E	$A := (A)+(1,1)$; if $(A)\geq 0$ then goto E
JSP	A,E	$A := L$; goto E where L=addr. of next instr.
CAME	A,E	if $(A)=(E)$ then skip next instruction
TLNN	A,E	if $(A_L)\wedge E \neq 0$ then skip next instr.
MOVE	A,E	$A := (E)$
MOVEI	A,E	$A := E$
MOVEM	A,E	$E := (A)$
SETZM	E	$E := 0$
HRRZ	A,E	$A := (0,(E_R))$
HLRZ	A,E	$A := (0,(E_L))$
HRRI	A,E	$A_R := E$
HRLZI	A,E	$A := (E,0)$
HRLM	A,E	$E_L := (A_R)$
HRLZM	A,E	$E := ((A_R),0)$
HRRZS		
ADD	A,E	$A := (A)+(E)$
PUSH	A,E	$A := (A)+(1,1)$; if $(A_L)=0$ then interrupt; $(A_R) := (E)$ [push item onto a stack]
BLT	A,E	$(A_R), (A_R)+1, \ldots, E := ((A_L)), ((A_L)+1), \ldots$ [transfer a block of words]
EXP	E	data pointer E
XWD	X_1,X_2	data word (X_1,X_2)

where

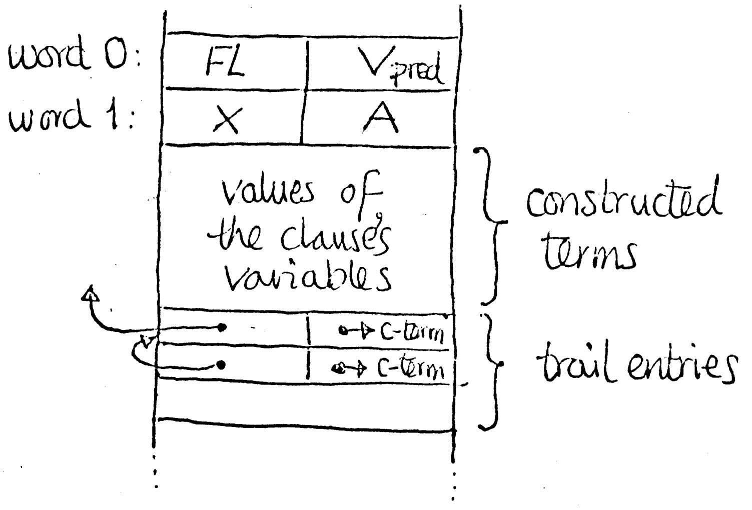
(X) = contents of X

X_L = left half of X

X_R = right half of X

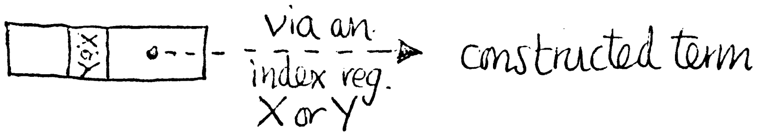
(X_1,X_2) = word with left-half X_1 and right-half X_2

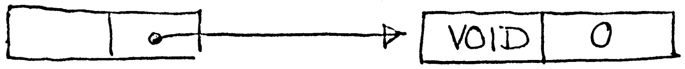
Format of an Environment

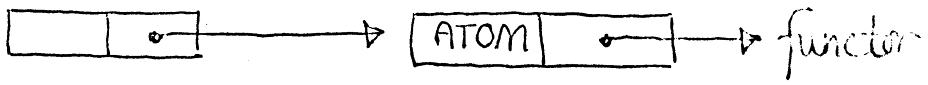


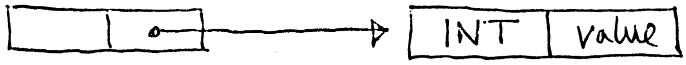
where Vpred is the address of the predecessor environment (for backtracking).

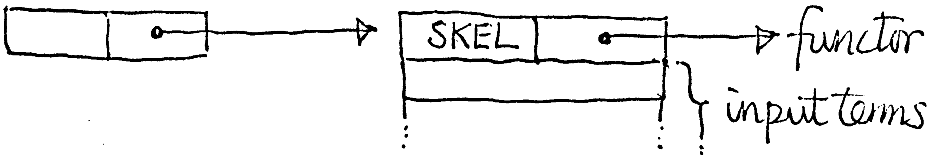
An input term is either.

(i) a variable: 

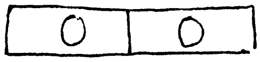
(ii) a void: 

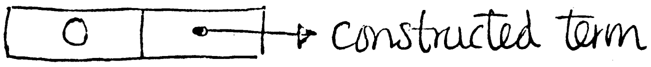
(iii) an atom 

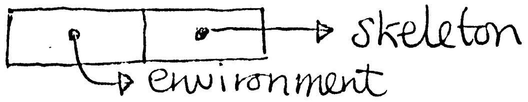
(iv) an integer: 

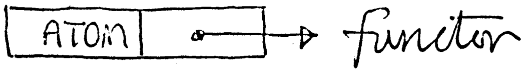
(v) a skeleton: 

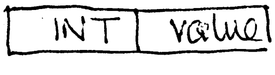
A constructed term is either

(i) unassigned: 

(ii) a reference: 

(iii) a molecule: 

(iv) an atom : 

(v) an integer: 

where
VOID = 1
SKEL = 2
ATOM = 4
INT = 5

Use of Fast Registers

ST : right-half = address of last word on the stack;
: left-half = − number of words left for allocation.

V : address of the environment for the current clause.

X : address of the environment of the parent clause.

A : right-half = address of the parent's argument list, followed by parent's continuation;
: left-half = right-half of X.

FL : the failure label = address of next clause for current predicate, or zero if there is none.

TR : address of the last trail entry (trail entries record variables which must be re-initialised on backtracking).

Y : address of an environment.

C : the link register (saves the return address for a run-time routine).

T, T1, B, B1 : terms (input or constructed) needed as parameters to run-time routines.

PDL : a push-down list pointer, similar to ST, used by the general unification routine.

PFN : the principal functor of the first argument of A (not yet used).

R1, R2 : registers for holding temporary results.

Masks

MASKMA : matches a molecule or atom.

MASKMAS : matches a molecule, atom or skeleton.