Example 2

```
 TYPE LOG2.MCT

+NOTRACE.
+LET(

LOG2(1,0);
LOG2(2,1);
LOG2(4,2);
LOG2(8,3);
LOG2(16,4);
LOG2(32,5);
LOG2(64,6);
LOG2(128,7);
LOG2(256,8);
LOG2(512,9);
LOG2(1024,10);
LOG2(2048,11);
LOG2(4096,12);
LOG2(8192,13);
LOG2(16384,14);
LOG2(32768,15);
LOG2(65536,16);
LOG2(131072,17)

).-COMPILE.-STOP.

.TYPE LOG2.OUT
```

```
 C1:  MOVE T,@1(A)
   TLNN T,MSKMAS
   JSP C,UATOM
   CAME T,A2
   JRST FAIL
   HRLM FL,0(V)
   JRST 2(A)
 C2:  MOVE T,@1(A)
   TLNN T,MSKMAS
   JSP C,UATOM
   CAME T,A1
   JRST FAIL
   HRLM FL,0(V)
   JRST 2(A)
 C3:  MOVE T,@1(A)
   TLNN T,MSKMAS
   JSP C,UATOM
   CAME T,A3
   JRST FAIL
   HRLM FL,0(V)
   JRST 2(A)
 C4:  MOVE T,@1(A)
   TLNN T,MSKMAS
   JSP C,UATOM
   CAME T,A6
   JRST FAIL
   HRLM FL,0(V)
   JRST 2(A)
 C5:  MOVE T,@1(A)
   TLNN T,MSKMAS
```

Illustrates the improved scheme for indexing clauses. Clauses are accessed "associatively" according to the first argument of the positive literal.

The code is rather bulky in this case, as the compiler generally aims at speed rather than compactness. It is planned to provide a user-option for selected predicates to be stored more compactly.

[This example was suggested by the fact that even in machine code there doesn't seem to be a simple trick for determining whether a number is a power of 2.]

```
        JSP C,UATOM
        CAME T,A4
        JRST FAIL
        HRLM FL,0(V)
        JRST 2(A)
C6:     MOVE T,@1(A)
        TLNN T,MSKMAS
        JSP C,UATOM
        CAME T,A9
        JRST FAIL
        HRLM FL,0(V)
        JRST 2(A)
C7:     MOVE T,@1(A)
        TLNN T,MSKMAS
        JSP C,UATOM
        CAME T,A11
        JRST FAIL
        HRLM FL,0(V)
        JRST 2(A)
C8:     MOVE T,@1(A)
        TLNN T,MSKMAS
        JSP C,UATOM
        CAME T,A13
        JRST FAIL
        HRLM FL,0(V)
        JRST 2(A)
C9:     MOVE T,@1(A)
        TLNN T,MSKMAS
        JSP C,UATOM
        CAME T,A5
        JRST FAIL
        HRLM FL,0(V)
        JRST 2(A)
C10:    MOVE T,@1(A)
        TLNN T,MSKMAS
        JSP C,UATOM
        CAME T,A16
        JRST FAIL
        HRLM FL,0(V)
        JRST 2(A)
C11:    MOVE T,@1(A)
        TLNN T,MSKMAS
        JSP C,UATOM
        CAME T,A18
        JRST FAIL
        HRLM FL,0(V)
        JRST 2(A)
C12:    MOVE T,@1(A)
        TLNN T,MSKMAS
        JSP C,UATOM
        CAME T,A20
        JRST FAIL
        HRLM FL,0(V)
        JRST 2(A)
C13:    MOVE T,@1(A)
        TLNN T,MSKMAS
        JSP C,UATOM
        CAME T,A22
        JRST FAIL
        HRLM FL,0(V)

        JRST 2(A)
```

```
C14:  MOVE T,@1(A)
  TLNN T,MSKMAS
  JSP C,UATOM
  CAME T,A24
  JRST FAIL
  HRLM FL,0(V)
  JRST 2(A)
C15:  MOVE T,@1(A)
  TLNN T,MSKMAS
  JSP C,UATOM
  CAME T,A26
  JRST FAIL
  HRLM FL,0(V)
  JRST 2(A)
C16:  MOVE T,@1(A)
  TLNN T,MSKMAS
  JSP C,UATOM
  CAME T,A28
  JRST FAIL
  HRLM FL,0(V)
  JRST 2(A)
C17:  MOVE T,@1(A)
  TLNN T,MSKMAS
  JSP C,UATOM
  CAME T,A7
  JRST FAIL
  HRLM FL,0(V)
  JRST 2(A)
C18:  MOVE T,@1(A)
  TLNN T,MSKMAS
  JSP C,UATOM
  CAME T,A31
  JRST FAIL
  HRLM FL,0(V)
  JRST 2(A)
P1:  JSP C,INTRO
  JSP C,SSECTO
  WD P1+81
  WD 0
  JRST FAILB
  WD 31
  WD @P1+7(R1)
  WD P1+39
  WD P1+66
  WD P1+69
  WD @P1+4
  WD P1+72
  WD @P1+4
  WD @P1+4
  WD @P1+4
  WD P1+75
  WD @P1+4
  WD @P1+4
  WD @P1+4
  WD @P1+4
  WD @P1+4
  WD @P1+4
  WD @P1+4
  WD @P1+4
  WD P1+78
  WD @P1+4
  WD @P1+4
  WD @P1+4
```

← entry point for predicate LOG2

array of 32 "clause lists" corresponding to the value of the righthand 5 bits of the first argument

```
WD @P1+4
WD @P1+4
WD @P1+4
WD @P1+4
WD @P1+4
WD @P1+4
WD @P1+4
WD @P1+4
WD @P1+4
WD @P1+4
WD @P1+4
CAMN R2,A8
JRST C6
CAMN R2,A10
JRST C7
CAMN R2,A12
JRST C8
CAMN R2,A14
JRST C9
CAMN R2,A15
JRST C10
CAMN R2,A17
JRST C11
CAMN R2,A19
JRST C12
CAMN R2,A21
JRST C13
CAMN R2,A23
JRST C14
CAMN R2,A25
JRST C15
CAMN R2,A27
JRST C16
CAMN R2,A29
JRST C17
CAMN R2,A30
JRST C18
JRST @P1+4
CAMN R2,A1
JRST C1
JRST @P1+4
CAMN R2,A3
JRST C2
JRST @P1+4
CAMN R2,A4
JRST C3
JRST @P1+4
CAMN R2,A5
JRST C4
JRST @P1+4
CAMN R2,A7
JRST C5
JRST @P1+4
MOVE R1,A1
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C1
MOVE R1,A3
MOVE R2,-1(V)
MOVEM R1,0(R2)
```

```
JSP FL,C2
MOVE R1,A4
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C3
MOVE R1,A5
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C4
MOVE R1,A7
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C5
MOVE R1,A8
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C6
MOVE R1,A10
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C7
MOVE R1,A12
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C8
MOVE R1,A14
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C9
MOVE R1,A15
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C10
MOVE R1,A17
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C11
MOVE R1,A19
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C12
MOVE R1,A21
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C13
MOVE R1,A23
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C14
MOVE R1,A25
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C15
MOVE R1,A27
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C16
MOVE R1,A29
MOVE R2,-1(V)
MOVEM R1,0(R2)
JSP FL,C17
MOVE R1,A30
```

This code will only get invoked if the first argument in the call is a variable.

[Again, it would be easy to provide a user option to suppress it.]

Ex 2-6

```
  MOVE R2,-1(V)
  MOVEM R1,0(R2)
  JSP FL,C18
  WD 0
FNTAB:  XWD 0,NO
  XWD 2,N1
NO:  ASCIZ "NIL"
N1:  ASCIZ "."
A31:  XWD INT,17
A30:  XWD INT,131072
A29:  XWD INT,65536
A28:  XWD INT,15
A27:  XWD INT,32768
A26:  XWD INT,14
A25:  XWD INT,16384
A24:  XWD INT,13
A23:  XWD INT,8192
A22:  XWD INT,12
A21:  XWD INT,4096
A20:  XWD INT,11
A19:  XWD INT,2048
A18:  XWD INT,10
A17:  XWD INT,1024
A16:  XWD INT,9
A15:  XWD INT,512
A14:  XWD INT,256
A13:  XWD INT,7
A12:  XWD INT,128
A11:  XWD INT,6
A10:  XWD INT,64
A9:  XWD INT,5
A8:  XWD INT,32
A7:  XWD INT,16
A6:  XWD INT,3
A5:  XWD INT,8
A4:  XWD INT,4
A3:  XWD INT,2
A2:  XWD INT,0
A1:  XWD INT,1
A0:  XWD ATOM,0
   END START
```