Pascal Byte Code Dispatch

```
; check for process switch
    tst.b   pswitch(SB)                 12 + 3
    BEQ.S   1$                          10 + 1
    JSR.L   checkprocessSwitch
1$:
; fetch next byte code, increment IP
    movea.l ActiveContext(SB), A0       16 + 4
    clr.l   D0                           6 + 1
    move.w  IP(SB), D0                  12 + 3
    clr.l   D1                           6 + 1
    move.b  0(A0, D0.L), D1             14 + 3
    move.b  D1, this_bytecode(SB)       13 + 3
    add.L   D1, D0                       8 + 1
    move.w  D0, IP(SB)                  13 + 3
;
; dispatch
    move.l  D1, -(SP)                   16 + 3
    pea     k36                         18 + 4
    JSR     _dcase.L                    22 + 5
                                       166 + 35
_dcase                                 324 + 73
                                       490 + 107
```

≈ 122.5 μsec ≈ 8150 bc/sec

Assuming that Dispatching is ≈ 10%
of an average bytecode then
10K bc/sec requires at most 10 μsec
of interpretation overhead per bytecode

```
executeLoop:
        Moveq   #0, D0          4+1
        MOVE.B  (IP)+, D0       8+2
        ADD.W   D0, D0          4+1
        MOVE.W  Dispatch(PC, D0.w), A0   14+3
        JMP     (A0)            14+3
bytecodeExit:
        JMP     0(A6, D6)       14+3
                                66+13
```

≈ 8.75 μsec

Assumes

IP is "address" in register
Dispatch table holds 16-bit absolute addresses
(all bytecode routines in 1st 32K)
Offset of "executeLoop" (relative to
object table) in Data Register.