

```

1 ; BCA/L/E DACTEST.BCA; RESUME SMALL.BOOT
2 ; Authors: Bruce Horn and Kim McCall
3 ; Last changed: February 6, 1979 10:53 PM by BH
4 ; This test uses the DAC to produce tones
5 ; Waveform = Sine, with Sampling technique
6 ; by Saunders
7 ;
8
9
10 .predefine "8086Predefs.sr"
11
12 ;Ports:
13 dacout = 0C0
14 dacstatus = 044
15 dacsampleselect = 060
16
17 ;constants:
18 fifoready = 4 ;FIFO input ready--output rdy=
19 UseChannelA = C000
20 UseChannelB = 8000 ;Must OR all arguments for output to 60H
21 TableLength = 200 ;# of values in the table (512 decimal)
22 DurMult = 178. ;cycles per sixtieth of quarter note
23
24 ;DAC clock rate:
25 S16Khz = 0
26 S8Khz = 0400
27 SaKhz = 0800
28 SbKhz = 0C00
29 ScKhz = 1000
30 SdKhz = 1400
31 SeKhz = 1800
32 S1Khz = 1C00
33 ClockedAt= 16000. ;16000 samples/sec is clock rate
34
35 ;Output levels:
36 Low = 07FFF ;Lowest level (0 volts out)
37 MidLevel = 04000 ;Mid level, i.e. zero line (5v)
38 High = 00000 ;Highest level (10 volts out)
39
40 .Loc 1000
41 DacTest:
42 jmp Beep
43
44 duration: .blk 2 ;in units of 1 sixtieth of a quarter note
45 pitch: .blk 2 ;a code # saying what pitch this is
46 intDelt: .blk 2 ;for holding the whole part of the delta
47 fracDelt: .blk 2 ;for holding the fractional part of the
48
49 ValueTable: .GetNoList "SineWaveTbl.bca"
50
51 DeltaTable: .GetNoList "DeltaTable.bca"
52
53
54 Beep:
55 mov ax,#S16khz ;select 16khz sampling
56 outw dacsampleselect
57 mov ax,duration ;in 90ths of a second
58 mov cx,#DurMult
59 mul cx ;gives # of samples for this note
60 mov cx,ax ;store cycle counter in cx
61 mov ax,pitch ;load pitch
62 mov dx,#MidLevel
63 cmp ax,#0
64 jne SetPitch
65
66 WaitMore:
67 call WaitReady
68 mov ax,dx ;just keep dac at midlevel
69 shl ax
70 outw dacout ;send midlevel to dac
71 loopnz WaitMore
72 ret ;return to caller
73
74 SetPitch:
75 mov si,ax ;set offset to pitch
76 dec si ;correct for 0 = rest

```

```

152B D1 E6      77    shl    si    ;multiply by 4 to
152D D1 E6      78    shl    si    ;index the delta table
152F 8B 84 0D 14 79    mov    ax,DeltaTable!si ;find delta for stepping thru
                        80    ; sine table
1533 A3 07 10    81    mov    intDelt,ax ;store int part
1536 FF C6      82    inc    si    ;increment offset
1538 FF C6      83    inc    si    ;
153A 8B 84 0D 14 84    mov    ax,DeltaTable!si ;get and store frac part
153E A3 09 10    85    mov    fracDelt,ax
1541 2B FF      86    sub    di,di ;init i part offset
1543 2B DB      87    sub    bx,bx ;init frac part holder
                        88
                        89    CycleAgain:
1545 E8 25 00    90    call  WaitReady ;wait for the dac fifo to unload
1548 8B C2      91    mov    ax,dx ;get starting voltage
154A D1 E7      92    shl    di    ;mult by 2 for word indexing
154C 03 85 0B 10 93    add    ax,ValueTable!di ;offset voltage by sine
1550 D1 E0      94    shl    ax
1552 E7 C0      95    outw  dacout ;send to dac
1554 D1 EF      96    shr    di    ;correct for earlier shl
1556 03 1E 09 10 97    add    bx,fracDelt ;augment frac part holder
155A 13 3E 07 10 98    adc    di,intDelt ;augment i part holder (? carry)
155E 81 FF 00 02 99    cmp    di,#TableLength ;have we looped around table?
1562 7C 04      100   jl    DecDur ;if not, jump one line
1564 81 EF 00 02 101   sub    di,#TableLength ;if so reset i-part offset
                        102   DecDur:
1568 FF C9      103   dec    cx
156A 75 D9      104   jnz   CycleAgain ;decrement duration count
156C C3        105   ret    ;return to caller
                        106
                        107   WaitReady:
156D E5 44      108   inw   dacstatus ;get status word
156F A9 04 00    109   test  ax,#fifoready ;is dac ready?
1572 75 F9      110   jnz   WaitReady ;no, try again, otherwise...
1574 C3        111   ret    ;return to caller
                        112
                        113
                        114
                        115   .END

```