The main kick routine used for CBM 8032 AV

Robert Henke, February 14 2025

```
* = $2000

wait_init                   = $f4  ;initial wait
current_wait                = $f5  ;wait counter
current                     = $f0  ;variable needed for DSP code, see below
previous                    = $f1  ;variable needed for DSP code, see below
period_count                = $f2  ;total number of wave periods before stop
pitch_decay_amount          = $f3  ;we add this to wait_init after each period
init_phase                  = $f6  ;starting point of the wave

NMIFlag                     = $a2  ;raised by the NMI when new note did arrive
currentNote                 = $0ff3  ;current note, set by NMI routine

; --- reset NMI re-trigger flag & disable interrupt---
.resetNMIflag
LDA #$00
STA NMIFlag
SEI

.drawBrightID
LDX #$01
JSR DrawID

.grabNoteParameters
LDA currentNote
AND #%00000111    ;8 variations
TAX

LDA routine_select_table,x  ;which algorithm to select
STA choose_routine+1
LDA hop_size_table,x  ;wavetable readout increment
STA hop+3
LDA pitch_decay_table,x ;how fast is pitch going down
STA pitch_decay_amount
LDA period_count_table,x  ;how many periods before stop
STA period_count
LDA init_pitch_table,x ;initial pitch
STA wait_init
LDA init_phase_table,x  ;initial phase
STA init_phase

.setFilter
; --- set filter ---
LDA filter_table,x
TAX
mSetFilter

; --- copy sine table to temporary lookup ---
LDY #$00              ; CPU cycles needed is 2
.copywave
LDA SineWaveTable,y   ; CPU cycles needed is 4
STA TempWaveTable,y   ; CPU cycles needed is 5
INY ; CPU cycles needed is 2
BNE copywave  ; CPU cycles needed is 3 (2 if no jump)


LDY #$80  ; initialise (n_1) for DSP to 0.0 Volt

.resetphase
 LDX init_phase
```

```
; //////// MAIN DSP LOOP /////////
.mainDSPloop
LDA TempWaveTable,x

.choose_routine
JMP versionA  ; !!! MODIFIED !!!

;--------  Medium Slow Damping  n = n * 4/8 + (n-1) * 3/8  + 1/8
.versionA
LSR A ; divide by 2
STA current
TYA   ; get previous output
LSR A ; divide previous by 2
LSR A ; divide previous by 2
STA previous
LSR A ; divide previous by 2
ADC previous ; add previous
ADC current
ADC #$20
JMP reentry

;--------  Very Fast Damping  n = n *1/4 + (n-1) * 3/4
.versionB
LSR A ; divide by 2
LSR A ; divide by 2
STA current
TYA ; get previous output
LSR A ; divide previous by 2
STA previous
LSR A
ADC previous
ADC current
JMP reentry

; --------  Medium Damping n = n * 1/2 + (n_1) *1/2
.versionC
LSR A ; divide by 2
STA current
TYA ; get previous output
LSR A ; divide previous by 2
ADC current
JMP reentry

; --------  Slow Damping n = n *3/4  + (n_1) * 1/4
.versionD
lsr a ; divide by 2
sta current
lsr a
adc current
sta current
tya
lsr a
lsr a
adc current

.reentry
STA TempWaveTable,x
STA DAC ; physical location of the DAC.
TAY

.hop
TXA
CLC
ADC #01  ; !!! MODIFIED !!!
TAX
BCC enterwait  ; if we did not reach end of current period we continue
```

```
; --- end of period calculations ---
DEC period_count
BEQ noteEnd ; exit if we reach last period

; ---  increment wait timer ----
LDA wait_init
CLC
ADC pitch_decay_amount
STA wait_init

.enterwait
; ---- waiting timer ----
LDA wait_init ; grab wait time
.wait
SBC #01
BNE wait

; --- check if new note arrived and we need to exit ---
.checkNewNote
LDA NMIFlag
BNE new_jump  ;if flag > 0 we exit
JMP mainDSPloop

.new_jump
JMP jumpAddress ; exit to new sound routine

; ////// END DSP //////////
.noteEnd
; --- draw dim note number ---
LDX #$00
JSR DrawID
; --- exit to draw moving things ---
JMP DrawBars

; -----   lookup tables ----
.routine_select_table;   !!!! this changes if code length changes !!!!!!!
EQUB 88,132,132,122, 106,106,106,106
.init_pitch_table  ; initial value for pitch wait timer
EQUB 3,6,10,12,  2,2,4,2
.hop_size_table  ; step size through 1 block sine table
EQUB 14,8,4,4,  4,2,12,4
.pitch_decay_table ; how fast we do lower pitch
EQUB 16,7,6,1  ,5,1,8,5
.period_count_table ; total number of periods
EQUB 6,13,13,240, 10,20,20,10
.init_phase_table
EQUB 0,64,64,1, 64,1,1,64
.filter_table
EQUB 1,1,1,1, 0,0,1,1


.resetNMIflag
     2000   A9 00      LDA #$00
     2002   85 A2      STA $A2
     2004   78         SEI
.drawBrightID
     2005   A2 01      LDX #$01
     2007   20 00 13   JSR $1300
.grabNoteParameters
     200A   AD F3 0F   LDA $0FF3
     200D   29 07      AND #$07
     200F   AA         TAX
     2010   BD E9 20   LDA $20E9,X
     2013   8D 56 20   STA $2056
     2016   BD F9 20   LDA $20F9,X
```

```
     2019   8D 9B 20   STA $209B
     201C   BD 01 21   LDA $2101,X
     201F   85 F3      STA $F3
     2021   BD 09 21   LDA $2109,X
     2024   85 F2      STA $F2
     2026   BD F1 20   LDA $20F1,X
     2029   85 F4      STA $F4
     202B   BD 11 21   LDA $2111,X
     202E   85 F6      STA $F6
.setFilter
     2030   BD 19 21   LDA $2119,X
     2033   AA         TAX
Macro mSetFilter:
     2034   BD 80 1F   LDA $1F80,X
     2037   8D 0B 90   STA $900B
     203A   8D FF 0F   STA $0FFF
     203D   BD 84 1F   LDA $1F84,X
     2040   8D 80 87   STA $8780
End macro mSetFilter
     2043   A0 00      LDY #$00
.copywave
     2045   B9 00 70   LDA $7000,Y
     2048   99 00 7D   STA $7D00,Y
     204B   C8         INY
     204C   D0 F7      BNE $2045
     204E   A0 80      LDY #$80
.resetphase
     2050   A6 F6      LDX $F6
.mainDSPloop
     2052   BD 00 7D   LDA $7D00,X
.choose_routine
     2055   4C 58 20   JMP $2058
.versionA
     2058   4A         LSR A
     2059   85 F0      STA $F0
     205B   98         TYA
     205C   4A         LSR A
     205D   4A         LSR A
     205E   85 F1      STA $F1
     2060   4A         LSR A
     2061   65 F1      ADC $F1
     2063   65 F0      ADC $F0
     2065   69 20      ADC #$20
     2067   4C 91 20   JMP $2091
.versionB
     206A   4A         LSR A
     206B   4A         LSR A
     206C   85 F0      STA $F0
     206E   98         TYA
     206F   4A         LSR A
     2070   85 F1      STA $F1
     2072   4A         LSR A
     2073   65 F1      ADC $F1
     2075   65 F0      ADC $F0
     2077   4C 91 20   JMP $2091
.versionC
     207A   4A         LSR A
     207B   85 F0      STA $F0
     207D   98         TYA
     207E   4A         LSR A
     207F   65 F0      ADC $F0
     2081   4C 91 20   JMP $2091
.versionD
     2084   4A         LSR A
     2085   85 F0      STA $F0
     2087   4A         LSR A
```

```
     2088   65 F0      ADC $F0
     208A   85 F0      STA $F0
     208C   98         TYA
     208D   4A         LSR A
     208E   4A         LSR A
     208F   65 F0      ADC $F0
.reentry
     2091   9D 00 7D   STA $7D00,X
     2094   8D 0C 90   STA $900C
     2097   A8         TAY
.hop
     2098   8A         TXA
     2099   18         CLC
     209A   69 01      ADC #$01
     209C   AA         TAX
     209D   90 0B      BCC $20AA
     209F   C6 F2      DEC $F2
     20A1   F0 2A      BEQ $20CD
     20A3   A5 F4      LDA $F4
     20A5   18         CLC
     20A6   65 F3      ADC $F3
     20A8   85 F4      STA $F4
.enterwait
     20AA   A5 F4      LDA $F4
.wait
     20AC   E9 01      SBC #$01
     20AE   D0 FC      BNE $20AC
Macro CheckEsc:
     20B0   A9 02      LDA #$02
     20B2   8D 10 E8   STA $E810
     20B5   AD 12 E8   LDA $E812
     20B8   29 01      AND #$01
     20BA   D0 03      BNE $20BF
     20BC   4C C0 1C   JMP $1CC0
.continue
End macro CheckEsc
     20BF   EA         NOP
     20C0   EA         NOP
     20C1   EA         NOP
     20C2   EA         NOP
.checkNewNote
     20C3   A5 A2      LDA $A2
     20C5   D0 03      BNE $20CA
     20C7   4C 52 20   JMP $2052
.new_jump
     20CA   4C F0 0F   JMP $0FF0
.noteEnd
Macro EndWait:
     20CD   A0 50      LDY #$50
.exit_y_loop
     20CF   A2 53      LDX #$53
.exit_x_loop
     20D1   A5 A2      LDA $A2
     20D3   D0 03      BNE $20D8
     20D5   4C DB 20   JMP $20DB
.new_jump_post
     20D8   4C F0 0F   JMP $0FF0
.continue_exit
     20DB   CA         DEX
     20DC   D0 F3      BNE $20D1
     20DE   88         DEY
     20DF   D0 EE      BNE $20CF
End macro EndWait
     20E1   A2 00      LDX #$00
     20E3   20 00 13   JSR $1300
     20E6   4C 00 17   JMP $1700
```

```
.routine_select_table
     20E9   58
     20EA   84
     20EB   84
     20EC   7A
     20ED   6A
     20EE   6A
     20EF   6A
     20F0   6A
.init_pitch_table
     20F1   03
     20F2   06
     20F3   0A
     20F4   0C
     20F5   02
     20F6   02
     20F7   04
     20F8   02
.hop_size_table
     20F9   0E
     20FA   08
     20FB   04
     20FC   04
     20FD   04
     20FE   02
     20FF   0C
     2100   04
.pitch_decay_table
     2101   10
     2102   07
     2103   06
     2104   01
     2105   05
     2106   01
     2107   08
     2108   05
.period_count_table
     2109   06
     210A   0D
     210B   0D
     210C   F0
     210D   0A
     210E   14
     210F   14
     2110   0A
.init_phase_table
     2111   00
     2112   40
     2113   40
     2114   01
     2115   40
     2116   01
     2117   01
     2118   40
.filter_table
     2119   01
     211A   01
     211B   01
     211C   01
     211D   00
     211E   00
     211F   01
     2120   01
```

```
A9 00 85 A2 78 A2 01 20
00 13 AD F3 0F 29 07 AA
BD E9 20 8D 56 20 BD F9
20 8D 9B 20 BD 01 21 85
F3 BD 09 21 85 F2 BD F1
20 85 F4 BD 11 21 85 F6
BD 19 21 AA BD 80 1F 8D
0B 90 8D FF 0F BD 84 1F
8D 80 87 A0 00 B9 00 70
99 00 7D C8 D0 F7 A0 80
A6 F6 BD 00 7D 4C 58 20
4A 85 F0 98 4A 4A 85 F1
4A 65 F1 65 F0 69 20 4C
91 20 4A 4A 85 F0 98 4A
85 F1 4A 65 F1 65 F0 4C
91 20 4A 85 F0 98 4A 65
F0 4C 91 20 4A 85 F0 4A
65 F0 85 F0 98 4A 4A 65
F0 9D 00 7D 8D 0C 90 A8
8A 18 69 01 AA 90 0B C6
F2 F0 2A A5 F4 18 65 F3
85 F4 A5 F4 E9 01 D0 FC
A9 02 8D 10 E8 AD 12 E8
29 01 D0 03 4C C0 1C EA
EA EA EA A5 A2 D0 03 4C
52 20 4C F0 0F A0 50 A2
53 A5 A2 D0 03 4C DB 20
4C F0 0F CA D0 F3 88 D0
EE A2 00 20 00 13 4C 00
17 58 84 84 7A 6A 6A 6A
6A 03 06 0A 0C 02 02 04
02 0E 08 04 04 04 02 0C
04 10 07 06 01 05 01 08
05 06 0D 0D F0 0A 14 14
0A 00 40 40 01 40 01 01
40 01 01 01 01 00 00 01
01
```