```
* Project2 Part A - Sinusoidal tone generator
*             This code plays the musical octave continuously (ie.
from middle C to upper C)
* Written by:Thusitha Mabotuwana
*         9790416
* 22/10/2002
*-------------------------------------------------------------------
--------------------------------------------------------

      ORG     $4100            ; Look up table for tones where
$82=Middle C,$7B=C#,$74=D,$6D=D+ etc.
        FCB     $82,$7B,$74,$6D,$67,$61,$5B,$56
        FCB     $51,$4D,$49,$42,$41

      ORG     $4000            ; Given sampling values to generate one
cycle of the sine wave
        FCB     $80,$9F,$CB,$E8,$FA,$FF,$FA,$E8,$CB
        FCB     $9F,$80,$59,$35,$19,$6,$0,$6,$18,$34,$58


*-------------------------------------------------------------------
--------------------------------------------------------

N       EQU     $14              ;20 samples
TOC1    EQU     $1016            ;Time Output Compare 1
TMSK1   EQU     $1022            ;Timer Mask1
TFLG1   EQU     $1023            ;Timer Flag1
FREQ    EQU     $3900            ;The tone to pick up is stored in this
memory location.
LKUP    EQU     $4000            ;Start address of sample lookup table
CYCLES  EQU     $4300            ;The number of times the counter has
access the whole of memory is sorted in this address
LED   EQU   $A201              ;address of LED2
JP8   EQU    $A400              ;Port direction
PORT1   EQU     $A600            ;PLD Port1 address

*-------------------------------------------------------------------
--------------------------------------------------------

      ORG     $3CB             ;Start address of ISR. Interrupt occurs
when TCNT=TOC1.
        JMP     ISR

      ORG    $2000             ;Start of the program
          CLRA                   ;Initialisation of regiters to make
them 0 at first
      LDX   #0          ;Need to make acc LED displays zero
        STAA    FREQ
          STAA  TFLG1
      STX   LED         ;Displays 0000 at the start
      LDAB  #01         ;Initial display should be 0001.Need manual 01
since this is 00 in machine
      STAB  $A201
        STAA    JP8              ;Set PLD PORT1 direction as OUTPUT
        STAA    CYCLES           ;Initially CYCLES is zero
        LDAA    #$80             ;Write #1000 0000 to TMSK1 to enable
TOC1
          STAA  TMSK1          ;Store $80 in TMSK1 at address 1022
```

```
        CLI                     ;Enable interrupts to occur
BEGIN   LDX     #N              ;Load the number of samples
        LDY     #LKUP           ;Y Points to the first SAMPLE in the
Lookup table
LOOP1   LDAA    0,Y             ;LOAD A with CURRENT sample from Lookup
table
        STAA    PORT1           ;OUTPUT DATA to PORT1
        INY                     ;INC index Y to be ready for NEXT
SAMPLE
        JSR     T1              ;Call DELAY subroutine
        DEX                     ;DEC X
        BEQ     BEGIN           ;IF X=0, ONE Cycle finishes and next
cycle starts
        JMP     LOOP1           ;Otherwise, OUTPUT next SAMPLE


*----------------------------------------------------------------------
-----------------------------------------------------

* DELAY SUBRUTINE:
T1      PSHY                    ;Record Y in a Stack so the register
can be used in this subroutine
        LDY     #$4100          ;Load Y with the starting address of
the tone lookup table
        LDAB    FREQ            ;Load ACCB with tone number
        ABY                     ;Add it to Y to get the current address
of the tone lookup table
        LDAA    $0,Y            ;Get current tone repeat value
L1      DECA                    ;start time delay loop to get the
correct time delay
        BNE     L1
        PULY                    ;Recall Y from Stack before returning
to main program
        RTS                     ;Return from Subroutine


*----------------------------------------------------------------------
-----------------------------------------------------

* Interrupt Service Routine

ISR     LDD     #$FFFF          ;It takes 0.016384 seconds to access
all memory locations
      ADDD  TOC1        ;add $FFFF to TOC1, so after 0.016384 sec
TCNT=TOC1 causing interrupt
          STD   TOC1        ;Stored in TOC1 so as to check when next
whole cycle finishes
        LDAA    CYCLES          ;ACCA has the number of times whole of
memory (65535) has been accessed
        INCA                    ;Increase number of times when
interrupt occurs
        STAA    CYCLES          ;Updated value stored
        CMPA    #$F4            ;Whole of memory need to be accessed
244 times for a 4s delay.
        BNE     DONE            ;Less than 244 times. Therefore should
play the same note
        LDAA    FREQ            ;Memory accessed 244 times. Need to
move to the next tone
      INCA                ;Updated frequency
```

```
        INCA                    ;Initially 1 was loaded(to avoid displaying
0000)thus when cycle is 2, 2 has to be displayed.
        STAA  LED               ; and display result
        DECA                    ;Restore value in ACCA
          STAA    FREQ          ;Updated frequency stored
          CMPA    #13           ;Check if all 13 noted have been played
          BNE     START         ;Should start playing notes from the
beginning. (ie again from mm $4100)
        CLRA                    ;Start from the beginning
          STAA    FREQ          ;Updated tone stored
        LDAA  #01
        STAA  LED       ;Display 0000 on LED's
START   CLRA                    ;End of cycles for 1 tone. Start from
the beginnning for next tone
        STAA    CYCLES          ;Updated value of CYCLES stored
DONE    LDX     #$1000          ;Needed for BCLR
        BCLR    TFLG1,X $7F     ;This need to be cleared to enable
interrupt in the next cycle
        RTI               ;Return from ISR
        END
```