```
; Author: Shuaib Shameem
; UIN: 673551999
; ECE 367 -Microprocessor-Based Design
; Experiment 4 - 24 Second Shot Clock
; 02/16/2012
;
; Purpose: Design a 24 second count down timer similar to the shot
; clock used in NBA basketball games. The reset button resets the
; count. The pause/stop button pauses and starts the counter.

;PAY ATTENTION TO THE ALIGNMENT BELOW
;Labels start in the first column  (left most column = colunm 1)
;OP CODES are at column 9
;COMMENTS follow a ";" symbol
;Blank lines are allowed (Makes the code more readable)

; Define symbolic constants
PortT   EQU  $240        ;Define Register Locations
PortM   EQU  $250
DDRT    EQU  $242
DDRM    EQU  $252
INITRG  EQU  $11
INITRM  EQU  $10
CLKSEL  EQU  $39
PLLCTL  EQU  $3A
CRGFLG  EQU  $37
SYNR    EQU  $34
REFDV   EQU  $35
COPCTL  EQU  $3C
TSCR1   EQU  $46
TSCR2   EQU  $4D
TIOS    EQU  $40
TCNT    EQU  $44
TC0     EQU  $50
TFLG1   EQU  $4E
;
; The ORG statment below would normally be followed by variable definitions
; There are no variables needed for this project.
;
        ORG  $3800         ; Beginning of RAM for Variables
COUNT:  EQU  $3800
BUT:    EQU  $3802
;
; The main code begins here. Note the START Label
;
       ORG   $4000         ; Beginning of Flash EEPROM
START  LDS   #$3FCE     ; Top of the Stack
       SEI                     ; Turn Off Interrupts
        MOVB  #$00, INITRG ; I/O and Control Registers Start at $0000
             MOVB  #$39, INITRM   ; RAM ends at $3FFF
;
; We Need To Set Up The PLL So that the E-Clock = 24MHz
;
       BCLR  CLKSEL,$80      ; disengage PLL from system
       BSET  PLLCTL,$40      ; turn on PLL
       MOVB  #$2,SYNR        ; set PLL multiplier
       MOVB  #$0,REFDV       ; set PLL divider
       NOP                     ; No OP
       NOP                        ; NO OP
plp     BRCLR CRGFLG,$08,plp  ; while (!(crg.crgflg.bit.lock==1))
       BSET  CLKSEL,$80      ; engage PLL
;
;
;
       CLI                     ; Turn ON Interrupts
       LDAA  #$FF      ; Make PortT Outbound
       STAA  DDRT
       LDAA  #$03      ; Make PortM pins 1 and 2 Outbound
       STAA  DDRM
;
; Initial Reset Location
;
AGAIN   BCLR  COUNT,$FF
        BSET  BUT,  $FF
;
       LDY   #$03        ; Load #10 on Index Register Y
;
       LDAA  TABLE, Y
        STAA  PortT       ; Output the value 0 to PortT
;
        BSET  PortM, $02  ; Set PortM to 10
        NOP               ; No Operation
        NOP
        BCLR  PortM, $FF  ; Clear all the bits of Port M
        NOP
        NOP
        BRA   FIRST       ; if the program is running the first time
                          ; branch to this label
;
;
; Reset the Units Place
;
UNITST  LDX   #$0A        ; Load #10 on Index Register X
        BRA   UNITS       ; Branch to "UNITS"
;
; Units Place Counter
;
FIRST:  LDX   #05         ; Load initial Values
        LDY   #02

UNITS:  LDAA  TABLE, X    ; Loads the value from memory location $5000 with the
offset X
        STAA  PortT       ; Output the value to PortT
;
        BSET  PortM, $01  ; Set PortM to 01
        NOP
        NOP
        BCLR  PortM, $FF  ; Clear all the bits of Port M
        NOP
;
        JSR   DELAY       ; Delay of 1 Second
;
       DEX                    ; Decrement the count
       BNE      UNITS            ; Do again unless the count = 102
;
       CPY   #$00        ; Compare to check if Y = 0
       BEQ   FLASH      ; Branch if Y = 0
;
; Tens Place Counter
;
TENS:   LDAA  TABLE, Y    ; Loads the value from memory location $5000 with the
offset X
        STAA  PortT       ; Output the value to PortM
;
        BSET  PortM, $02  ; Set Bits of PortM to 100
        NOP
        NOP
        BCLR  PortM, $FF  ; Clear all the bits of PortM
;
        DEY               ; Decrement the count
        BRA   UNITST      ; Branch to "UNITST"
;
; Flash the display with the interval of 1 Secs
;
FLASH   BCLR  PortT, $FF  ; Clears out PortT
        JSR   OPENB       ; Opens both M1 and M2 for Latch Enable
        JSR   DELAY       ; Delays the program for one second

        BSET  PortT, $3F  ; Sets the value $3F in portT
        JSR   OPENB       ; Opens both M1 and M2 for Latch Enable
        JSR   DELAY       ; Delays the program for one second

        BRA   FLASH      ; Branch to "FLASH" for infinite loop
;
; Open both ports
;
OPENB   BSET  PortM, $03  ; Set Bits of PortM to 11
        NOP
        NOP
        BCLR  PortM, $FF  ; Clear all the bits of PortM
        NOP
        RTS
;
; We use the CPU clock cycles to create a delay
;
; Delay of about 1 Sec with the switching control
;
DELAY:  PSHY
        LDAA  #100                ; Outer Loop Counter – 1 clock cycle
;
L1:     LDY   #12000              ; Inside Loop Counter 2 clock cycles
;
L0:     BRCLR PortM, $04, L3      ; Branch to L3 if the button is pressed
        BRSET BUT,   $FF, L0      ; Branch if 'BUT' is set to hex $FF
        BRSET COUNT, $01, L4      ; Branch if 'COUNT' is set to hex $01
        BRA   L0                  ; Branch always to L0

L3:     JSR   SDELAY              ; Delay for button Debounce
L5:     BRCLR PortM, $04, L5      ; Branch to L5 is the button hasnt been
released
        JSR   SDELAY              ; Delay for button Debounce

        COM   BUT                 ; Compliment 'BUT'
        BRSET COUNT, $01, L0      ; Branch if 'COUNT' is set to hex $01
        BSET  COUNT, $01          ; Bit set 'COUNT'
;
L4:     DEY                       ; Decrement IndexY
        LBNE  L0                  ; 3 clock cycles
        DECA                      ; Decrement AccA,
        BNE   L1                  ; 3 clock cycles
        PULY
        RTS                       ; Return from subroutine - 5 clock cycles
;
; Short Delay of 5 mSecs
;
SDELAY: PSHY
        LDY #15000   ; Loop counter = 15000 - 2 clock cycles
A0:     LBRN A0      ; 3 clock cycles \
        DEY          ; 1 clock cycles | 8 clock cycles in loop
        LBNE A0      ; 4 clock cycles / Time = 8*<Y>/(24*10**6) + 2 =
;                    ; [8X15000 + 2]/24000000 ~= 5msec
        PULY
        RTS
;
; End of counter code
```

```
        ORG $5000
;
; Table of 7 segment LED values as bits 7-0 as 0gfedcba.
;
TABLE:  DC.B $00, $3F, $06, $5B, $4F, $66, $6D, $7D, $07, $7F, $67
;
;    Order  :off, 0, 1, 2, 3, 4, 5 ,6 ,7 ,8 ,9


; Define Power-On Reset Interrupt Vector

; AGAIN - OP CODES are at column 9

        ORG $FFFE ; $FFFE, $FFFF = Power-On Reset Int. Vector Location
        FDB START ; Specify instruction to execute on power up

; End of Interrupt code

        END        ; (Optional) End of source code
```