Resolución de tarea de Electrónica Digital II

Valenzuela Gabriel

Universidad Nacional de Córdoba - FCEFyN

31 de marzo de 2019

Ejercicio 2.6

A continuación se adjunta el código fuente, cabe aclarar que el mismo se modifica para añadirse aquí, es por ello que al final se deja un enlace a un repositorio de GitHub.

```
;*******************************************************************************
; TITLE : EXERCISE 2.6
; PROFESSOR: MARTIN DEL BARCO
; AUTHOR: VALENZUELA GABRIEL EMANUEL
; SUBJET: DIGITAL ELECTRONICS II
; INSTITUTE : FEFYN - UNC
; DATE: 03-30-2019
; VERSION: 1.0.0
;
; RESUME:
; THIS PROGRAM GENERATE A DELAY OF 1[s] USING NESTED LOOPS WITH A ERROR OF
; 1.06%
;*******************************************************************************
#include "p16f887.inc"  ;HEADER THAT CONTAINS ALL ADDRESS MEMORY DEFINED BY MI-
                        ;CROCHIP
    LIST P=16F887       ;DEFINE THE MICROCONTROLLER TO BE USED.
;*******************************************************************************
; CONFIG1

; __config 0x3FF1
 __CONFIG _CONFIG1, _FOSC_XT & _WDTE_OFF & _PWRTE_OFF & _MCLRE_ON & _CP_OFF
  & _CPD_OFF & _BOREN_ON & _IESO_ON & _FCMEN_ON & _LVP_ON
; CONFIG2
; __config 0x3FFF
 __CONFIG _CONFIG2, _BOR4V_BOR40V & _WRT_OFF
;===============================================================================
;                           VARIABLE & CONSTANT DEFINITION
COUNTER_ADD EQU 0X20
OCOUNT_ADD EQU 0X21
ICOUNT_ADD EQU 0X22
;===============================================================================
NESTED_OC SET D'243'
NESTED_IC SET D'243'
COUNTER_VAL SET D'7'
;===============================================================================

;===============================================================================
;                           ORIGIN PRORGRAM
    ORG 0X00
    GOTO PROGRAM
;===============================================================================

;===============================================================================
```

```
;                                   MAIN PROGRAM
PROGRAM ORG 0X05
        CLRW
        CLRF COUNTER_ADD
        CLRF OCOUNT_ADD
        CLRF ICOUNT_ADD

        ;CLEAR ALL

        MOVLW COUNTER_VAL
        MOVWF COUNTER_ADD

        MOVLW NESTED_OC
        MOVWF OCOUNT_ADD

        MOVLW NESTED_IC
        MOVWF ICOUNT_ADD

        CALL DELAY              ;CALL SUBRUTINE
        GOTO PROGRAM            ;START AGAIN THE HOLE PROGRAM

        ;TWELVE LINES OF CODE...
        ;CLEAR ALL = 4IC I.E 4uS
        ;THE SETTER, THAT MEANS, THE MOV, EACH ONE HAS
        ;1IC SO, IT'S HAS 6IC EQUIVALENT TO 6uS
        ;CALL AND GOTO SUMS 4IC I.E 4uS
        ;THEREBEFORE, THE MAIN PROGRAM CONSUMES
        ;14IC

DELAY   DECFSZ COUNTER_ADD,1
        GOTO OUTER              ;GO TO OUTER COUNT COUNTER TIMES
        CLRF OCOUNT_ADD         ;CLEAR
        CLRF ICOUNT_ADD         ;CLEAR
        RETURN                  ;RETURN TO MAIN PROGRAM


OUTER
        DECFSZ OCOUNT_ADD,1     ;DECREMENT THE COUNTER, SKIP IF ZERO
        GOTO INNER              ;GO TO INNER COUNT NESTED_OC TIMES
        MOVLW NESTED_OC         ;IF IT'S ZERO, I MUST RESTART THE VALUE
        MOVWF OCOUNT_ADD        ;SET THE OUTER COUNTER
        GOTO DELAY              ;COME BACK TO DELAY SUBRUTINE
        ;FROM THE EQUATION:
        ;766 IC = 3*(NESTED_OC-1) + 4 I.E NESTED_OC = 255

INNER   DECFSZ ICOUNT_ADD       ;DECREMENT THE COUNTER, SKIP IF ZERO
        GOTO INNER              ;IF NOT ZERO, START AGAIN
        MOVLW NESTED_IC         ;IF IT'S ZERO, I MUST RESTART THE VALUE
        MOVWF ICOUNT_ADD        ;SET THE INNER COUNTER
        GOTO OUTER              ;RETURN TO OUTER
        ;FROM THE EQUATION:
        ;768 IC = 3*(NESTED_IC-1) + 6 I.E NESTED_IC = 255
        END
```

Luego de ejecutar la simulación se obtuvo:

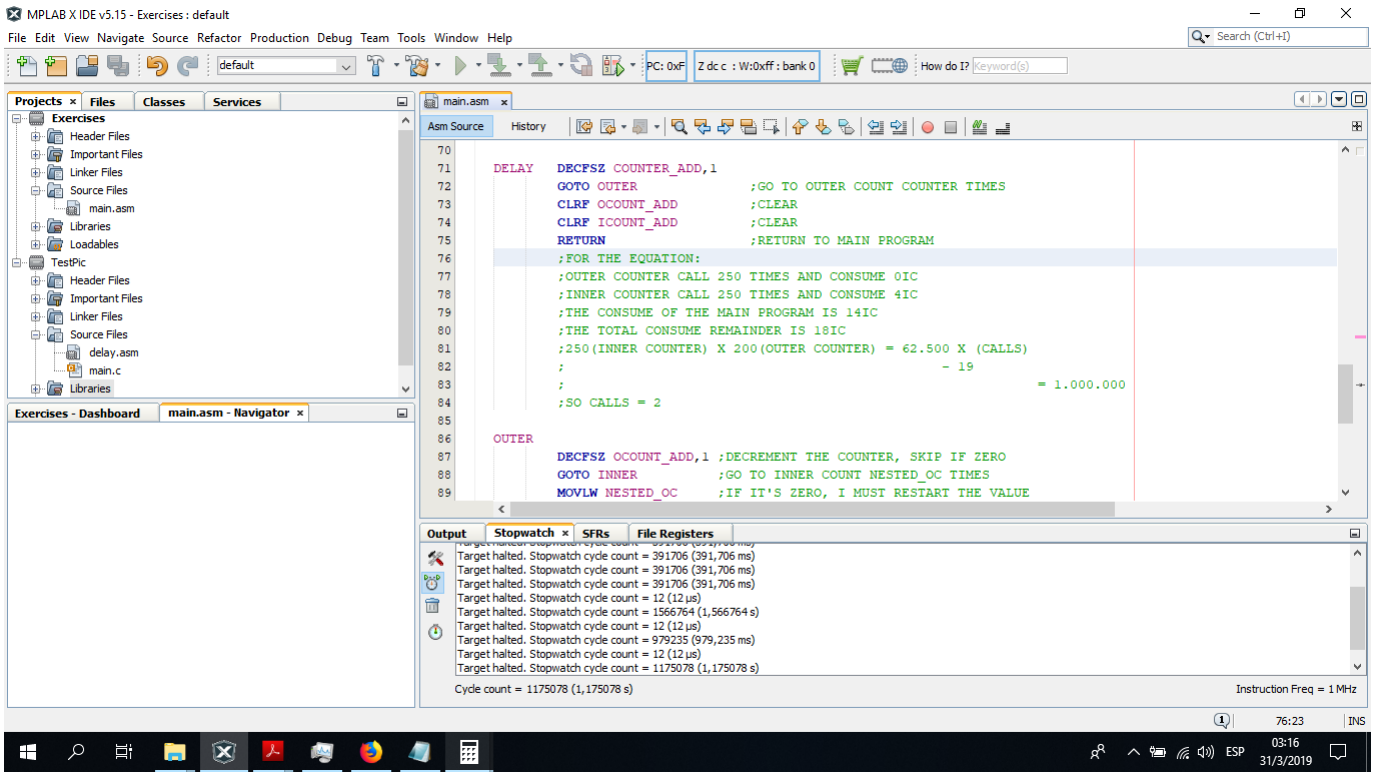

Figura 1: Primera simulación sin ajuste

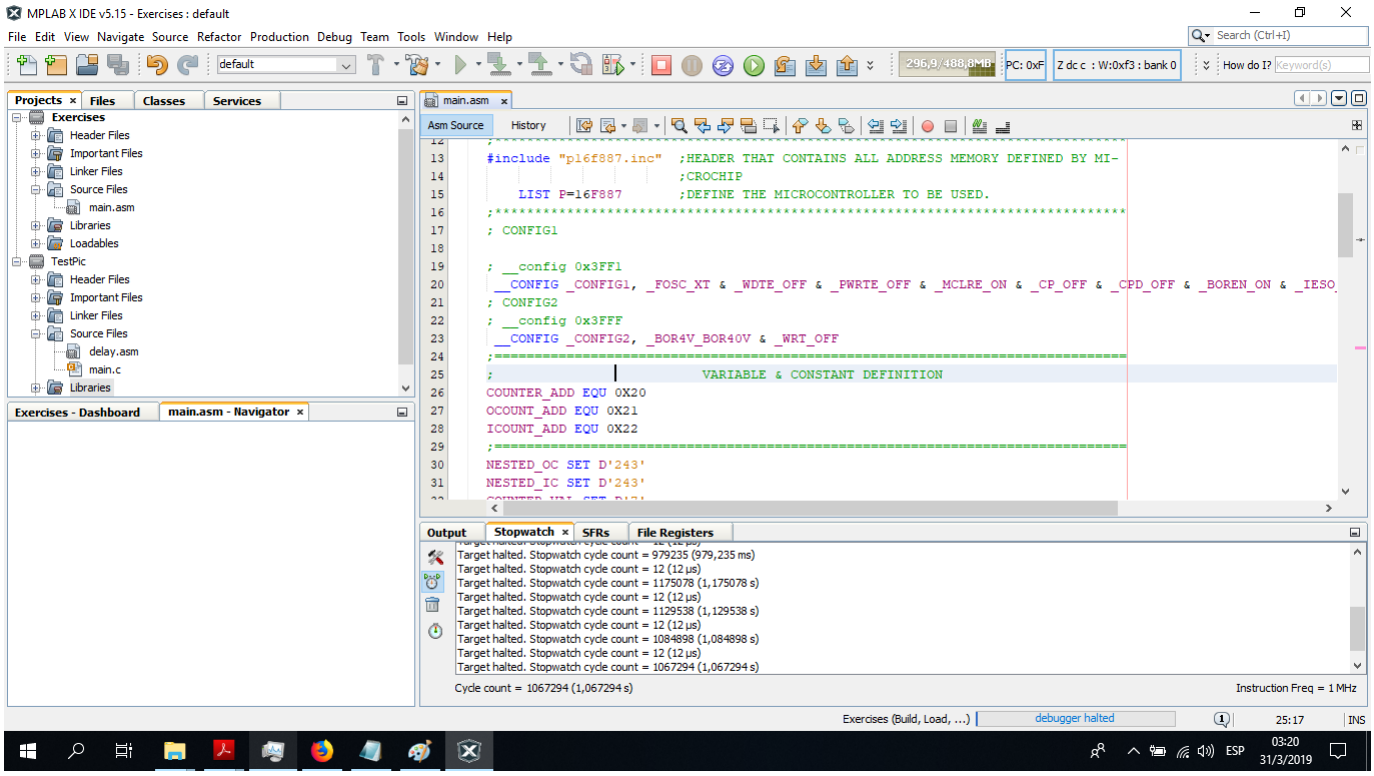

Figura 2: Segunda simulación con ajuste

Ejercicio 2.11

A continuación se adjunta el código fuente, cabe aclarar que el mismo se modifica para añadirse aquí, es por ello que al final se deja un enlace a un repositorio de GitHub.

```
;*******************************************************************************
; TITLE : EXERCISE 2.11
; PROFESSOR: MARTIN DEL BARCO
; AUTHOR: VALENZUELA GABRIEL EMANUEL
; SUBJET: DIGITAL ELECTRONICS II
; INSTITUTE : FEFYN - UNC
; DATE: 03-31-2019
; VERSION: 1.0.0
;
; RESUME:
; THIS PROGRAM TAKES A NUMBER ON ASCII FORMAT TO CONVERT IT ON HEXADECIMAL
; FORMAT UNPACKED. IT ASSUME THAT THE UNPACKED FORMAT CONSIST ON THE NUMBER
; REPRESENTATION USING 4 BITS, AND THE LAST 4 BITS ARE USE TO THE SIGN, HERE
; THE SIGN IT'S ALWAYS POSITIVE. FOR MORE INFORMATION (PP 38)
; @SEE: http://weblidi.info.unlp.edu.ar/catedras/organiza/descargas/clase1.pdf
;*******************************************************************************
#include "p16f887.inc"  ;HEADER THAT CONTAINS ALL ADDRESS MEMORY DEFINED BY MI-
                        ;CROCHIP
    LIST P=16F887       ;DEFINE THE MICROCONTROLLER TO BE USED.
;*******************************************************************************
; CONFIG1

; __config 0x3FF1
 __CONFIG _CONFIG1, _FOSC_XT & _WDTE_OFF & _PWRTE_OFF & _MCLRE_ON
  & _CP_OFF & _CPD_OFF & _BOREN_ON & _IESO_ON & _FCMEN_ON & _LVP_ON
; CONFIG2
; __config 0x3FFF
 __CONFIG _CONFIG2, _BOR4V_BOR40V & _WRT_OFF
;===============================================================================
;                               STORE DEFINITION
    CBLOCK  0X21
    ADDATA_0
    ADDATA_1
    ADDATA_2
    ADDATA_3
    ADDATA_4
    ADDATA_5
    ADDATA_6
    ADDATA_7
    ENDC

    CBLOCK  0X31
    RDATA_0
    RDATA_1
    RDATA_2
    RDATA_3
    RDATA_4
    RDATA_5
    RDATA_6
    RDATA_7
    ENDC

;===============================================================================
;                               VARIABLES
ASCII_0 SET 'E'
ASCII_1 SET 'l'
```

```
ASCII_2 SET 'e'
ASCII_3 SET 'c'
ASCII_4 SET 't'
ASCII_5 SET 'r'
ASCII_6 SET 'o'
ASCII_7 SET '!'
;=================================================================================
;                                   ORIGIN PRORGRAM
    ORG 0X00
    GOTO PROGRAM
;=================================================================================
;                                   MACROINSTRUCTIONS
; MACRO CONVERT (HEX,ASCII) TAKES THE ADDRESS OF A HEX VALUE AND OPERATES TO
; CONVERT IT ON UNPACKED FORMAT
CONVERT MACRO HEX,ASCII
        MOVF HEX,W
        ANDLW b'11111100'
        MOVWF ASCII
        ENDM
;=================================================================================
;                                   MAIN PROGRAM
PROGRAM ORG 0X05
        CLRW
        CALL SETTER
        CONVERT ADDATA_0, RDATA_0
        CONVERT ADDATA_1, RDATA_1
        CONVERT ADDATA_2, RDATA_2
        CONVERT ADDATA_3, RDATA_3
        CONVERT ADDATA_4, RDATA_4
        CONVERT ADDATA_5, RDATA_5
        CONVERT ADDATA_6, RDATA_6
        CONVERT ADDATA_7, RDATA_7
        GOTO PROGRAM

SETTER:                             ;MARK LABEL OF SUBRUTINE SETTER
;SUBRUTINE: SET ALL VALUES PREVIOUSLY DEFINED
SETTER
        MOVLW   ASCII_0
        MOVWF   ADDATA_0
        MOVLW   ASCII_1
        MOVWF   ADDATA_1
        MOVLW   ASCII_2
        MOVWF   ADDATA_2
        MOVLW   ASCII_3
        MOVWF   ADDATA_3
        MOVLW   ASCII_4
        MOVWF   ADDATA_4
        MOVLW   ASCII_5
        MOVWF   ADDATA_5
        MOVLW   ASCII_6
        MOVWF   ADDATA_6
        MOVLW   ASCII_7
        MOVWF   ADDATA_7
        RETURN
        END
```

1. Repositorio

Figura 3: Repositorio

Referencias

[1] Valdes-Perez, Fernando E., and Ramon Pallas-Areny. Microcontrollers: fundamentals and applications with PIC. CRC press, 2009.

[2] PIC16F882/883/884/886/887 Data Sheet

[3] Wilmshurst, Tim. Designing embedded systems with PIC microcontrollers: principles and applications. Elsevier, 2006.

[4] Assembly Programming for Mid-Range PIC. Slide of Martin Land. Subject Embedded Systems for Medical Devices `http://cs.hadassah.ac.il/staff/martin/embedded/slide05-1.pdf`