# TM57FLA80

# ROM Page Switch When Interrupt Occurs

# Application Note

tenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. tenx does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. tenx products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.

# AMENDMENT HISTORY

| Version | Date | Description |
|---|---|---|
| V1.0 | May, 2012 | New release |

# CONTENTS

## Description

1.The 8K ROM storage in TM57FLA80 is divided into two 4K-page: Page0 and Page1.

2.The address in ROM is divided into: Page0: 000H～FFFH; Page1: 1000H～1FFFH.

3.During the usage, when routine uses more than 4K byte address, it has to be noticed that whether page is switched to another page when interrupt occurs.

4.The command goto in interrupt vector is decided by IC hardware which can only jump to Page0 address (regardless of the routine is interrupted from Page0/1), therefore the interrupt subroutine entry address must be inside Page0 address.

Below samples are used to describe how the settings in the 3 conditions are:

Routine:

```
;-*-*-*-*-*-Register Definition-*-*-*-*-*-
TM0             equ     01H     ;;F-Plane  Timer0 register
STATUS          equ     03H     ;;F-Plane  Status register
INTE0           equ     08H     ;;F-Plane  Interrupt Enable register
INTF0           equ     09H     ;;F-Plane  Interrupt Flag register
TM0CTL          equ     02H     ;;R-Plane
MFR10           equ     10H     ;;R-Plane
;;
Tm0DatBuf   equ   40h           ;;F-Plane   TM0 restore buf value
;-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
; Condition 1: All routine is in Page0, page switch is not necessary in
; the interrupt subroutine Page, routine can proceed goto/call operation normally.
;-*-*-*-*-*- Program starts -*-*-*-*-*-
        org     000h
        goto    Main
        org     001h
        goto    Tm0IntPro       ;; Interrupt vector
        ..
        org     00Ah
Tm0IntPro:                      ;; Interrupt subroutine entry address
        btfss   INTF0,4
        goto    Int_End
        movfw   Tm0DatBuf
        addwf   TM0             ;; Reload TM0
        movlw   11101111b       ;; Clear TM0 interrupt flag
        movwf   INTF0
Int_End:
        RETI                    ;; Exit interrupt
;;-----------
Main:
        movlw   00010000b       ;; AutoSave open, auto save and restore W and
```

```
                                ;; STATUS value automatically when interrupt occurs
        movwr MFR10
        movlw 00001000b
        movwr TM0CTL            ;; Timer TM0 related setting
        movlw .1
        movwf TM0
        movwf Tm0DatBuf
        clrf  INTF0             ;; Clear interrupt flag
        movlw 00010000B
        movwf INTE0             ;; Enable TM0 interrupt
Loop:                           ;; Enter loop wait for interrupt occurs
        nop
        goto  Loop
;-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
; Condition 2: Interrupt occurs when routine executes in Page1.
;-*-*-*-*-*- Program starts -*-*-*-*-*-
        org   000h
        goto  Main
        org   001h
        goto  Tm0IntPro         ;; Interrupt vector; this goto command is decided by IC
                                ;; hardware which can only jump to Page0 address
                                ;; (regardless of the routine is interrupted from Page0/1),
                                ;; this command decides that interrupt subroutine entry
                                ;; address must be inside Page0 address. Besides, after
                                ;; entering interrupt subroutine, the page is switched
                                ;; according to destination address in goto/call command.
                                ;; Please refer to the following discussion:
        ..
        org   00Ah
Tm0IntPro:                      ;; Interrupt subroutine entry address
        bcf   STATUS,7          ;; At this point, the page is switched to Page0 because
                                ;; the routine may execute the following "goto Int_End"
                                ;; statement, because the Int_End address is in Page0.
        btfss INTF0,4
        goto  Int_End           ;; go toPage0
        movfw Tm0DatBuf
        addwf TM0               ;; reload TM0
        movlw 11101111b         ;; clear TM0 interrupt flag
        movwf INTF0
Int_End:
        RETI                    ;; exit the interrupt (Note: no need to switch the page to
                                ;; Page0/1 before exit the interrupt, because AutoSave
                                ;; function is enabled, when exit the interrupt, it will auto
```

```
                                ;; recover the original Page control bit).
;;-----------
Main:
        movlw   00010000b       ;; AutoSave is enabled, the W and STATUS value will be
                                ;; saved and recover automatically when interrupt occurs
        movwr   MFR10
        bsf     STATUS,7        ;; Switch Page to Page1, to make sure the "goto
                                ;; Label_1" jump is working properly.
        goto    Label_1         ;; Go to Page1
        ..
        ..
        org      1000H;; Page1: address 1000H
Label_1:
        movlw   00001000b
        movwr   TM0CTL          ;;Timer TM0 related settings
        movlw   .1
        movwf   TM0
        movwf   Tm0DatBuf
        clrf    INTF0           ;; Clear interrupt flag
        movlw   00010000b
        movwf   INTE0           ;; Enable TM0 interrupt
Loop_2:                         ;; Enter loop wait for interrupt occurs
        nop
        goto    Loop_2
;-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
; Condition 3: goto Page1 address in Interrupt subroutine; before executes
; the goto command, the page must be switched to Page1, otherwise it will go
; to the wrong address.
;-*-*-*-*-*- Program starts -*-*-*-*-*-
        org     000h
        goto    Main
        org     001h
        goto    Tm0IntPro       ;; Interrupt vector; this goto command is decided by IC
                                ;; hardware which can only jump to Page0 address
                                ;; (regardless of the routine is interrupted from Page0/1),
                                ;; this command decides that interrupt subroutine entry
                                ;; address must be inside Page0 address. Besides, after
                                ;; entering interrupt subroutine, the page is switched
                                ;; according to destination address in goto/call command.
                                ;; Please refer to the following discussion:
        ..
        org     00Ah
Tm0IntPro:                      ;; Interrupt subroutine entry address
```

```
        bcf     STATUS,7    ;; At this point, the page is switched to Page0 because
                            ;; the routine may execute the following "goto Int_End"
                            ;; statement, because the Int_End address is in Page0.
        btfss   INTF0,4
        goto    Int_End     ;; Go to Page0
        movfw   Tm0DatBuf
        addwf   TM0         ;; Reload TM0
        movlw   11101111b
        movwf   INTF0       ;; Clear TM0 interrupt flag
        bsf     STATUS,7    ;; At this point, the page is switched to Page1 because
                            ;; when the following "goto Pro_1" statement is executed,
                            ;; the routine will be switched from Page0 to Page1.
        goto    Pro_1       ;; Go to Page1
Int_End:
        RETI                ;; exit the interrupt (Note: no need to switch the page to
                            ;; Page0/1 before exit the interrupt, because AutoSave
                            ;; function is enabled, when exit the interrupt, it will auto
                            ;; recover the original Page control bit).
;;-----------
Main:
        movlw   00010000b   ;; AutoSave is enabled, the W and STATUS value will be
                            ;; saved and recover automatically when interrupt occurs
        movwr   MFR10
        bsf     STATUS,7    ;; Switch Page to Page1, to make sure the "goto
                            ;; Label_1" jump is working properly
        goto    Label_1     ;; Go to Page1
        ..
        ..
        org     1000H;; Page1: address 1000H
Label_1:
        movlw   00001000b
        movwr   TM0CTL      ;; Timer TM0 related settings
        movlw   .1
        movwf   TM0
        movwf   Tm0DatBuf
        clrf    INTF0       ;; Clear interrupt flag
        movlw   00010000b
        movwf   INTE0       ;; Enable TM0 interrupt
Loop_2:                     ;; Enter loop wait for interrupt occurs

        nop
        goto    Loop_2
        ;;-----
```

```
Pro_1:          ;; Page1, interrupt subroutine branch address
        nop
        RETI    ;; exit the interrupt (Note: no need to switch the page to
                ;; Page0/1 before exit the interrupt, because AutoSave
                ;; function is enabled, when exit the interrupt, it will auto
                ;; recover the original Page control bit).
```