การใช้ Fixed-point Math Library

1. Download library source code (00617.zip) จากหน้าเวปของรายวิชา
2. Unzip file ออก จะพบกลุ่มไฟล์ library สำหรับการหาร ซึ่งจะมีชื่อว่า **FXD***xy***.A16** โดย x และ y เป็นตัวเลข และ กลุ่มไฟล์ library สำหรับการคูณ ซึ่งจะมีชื่อว่า **FXM***xy***.A16** โดย x และ y เป็นตัวเลข ซึ่ง x และ y จะบอกจำนวนบิตของ operand ได้แก่ (2 = 22 บิต, 4 = 24 บิต, 6 = 16 บิต, 8 = 8 บิต)

การใช้งานทำได้ 2 วิธี คือ

1. include ไฟล์ที่ต้องการเข้ามาใน code ที่ต้องการใช้งาน หรือ
2. คัดลอกเอาเฉพาะ subroutine ที่จะเรียกใช้ เข้ามาใน code ที่ต้องการ

วิธีแรกทำได้โดยระบุ include file ที่ส่วนต้นของ code ดังนี้

```
#include <p16f628a.inc>
#include “math16.inc”          ;ต้องทำการ edit content ในไฟล์นี้ให้ตรงกับ MCU ที่ใช้
#include “fxd__.a16”            ;ถ้าต้องการใช้การหาร (เลือกขนาดของ operand ตามต้องการ)
#include “fxm__.a16”            ;ถ้าต้องการใช้การคูณ (เลือกขนาดของ operand ตามต้องการ)
```

โดยไฟล์ math16.inc, fxd__.a16 และ fxm__.a16 จะต้องอยู่ใน folder เดียวกันกับ code ที่เขียน
วิธีที่สองทำได้โดยคัดลอกเฉพาะ subroutine ที่ต้องการเข้ามารวมอยู่กับ code ที่จะเรียกใช้งาน ซึ่งจะมีข้อดีกว่าคือ code ของ library ที่ไม่ได้ใช้ก็ไม่จำเป็นต้องถูก assembler แปล ทำให้ได้ .hex file ที่มีขนาดเล็กกว่า โดยจะต้องคง math16.inc ไว้ในส่วนของการ include โดยใน comment ของแต่ละ subroutine จะมีคำอธิบายวิธีเรียกใช้ไว้ให้แล้ว

การใช้งานทั้งสองวิธี ต้องทำการ edit ไฟล์ math16.inc ดังนี้

1. ลบข้อความตั้งแต่บรรทัดที่ 43 “IF (P16_MAP1)” ไปจนถึงบรรทัดที่ 138 “IF (P16_MAP2)”
2. ลบข้อความตั้งแต่บรรทัดที่ 287 “ ;24 BIT FLOATING POINT CONSTANTS “ ไปจนจบไฟล์

ตัวอย่างการนำ code จาก 8 บิต library มาใช้งานเป็นดังนี้

```
#include <p16f628a.inc>
#include "math16.inc"

                org 0x000
                ;Your program
                movlw   0x0f     ; Dividen = 15
                movwf   AARGB0
                movlw   0x04     ; Divisor = 4
                movwf   BARGB0
                call    FXD0808U ; result is at 0x27 and remainder is at 0x23
        loop    goto loop

; THE FOLLOWING CODE IS OBTAINED FROM FXD88.A16 WITH SOME SUBROUTINE REMOVED
;*****************************************************************************************

;       08/08 BIT Division Macros

SDIV0808L       macro
;       Max Timing:     3+5+2+5*11+10+2 = 77 clks
;       Min Timing:     3+5+2+5*11+10+2 = 77 clks
;       PM: 22                                  DM: 4

                MOVF            BARGB0,W
                SUBWF           REMB0, F
                RLF             AARGB0, F
                RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                ADDWF           REMB0, F
                RLF             AARGB0, F
                MOVLW           6
                MOVWF           LOOPCOUNT
LOOPS0808A      RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W

                BTFSC           AARGB0,LSB
                SUBWF           REMB0, F
                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F
                RLF             AARGB0, F

                DECFSZ          LOOPCOUNT, F
                GOTO            LOOPS0808A

                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F

                endm

UDIV0808L  macro

;       Max Timing: 2+7*12+11 = 97 clks
;       Min Timing: 2+7*11+10 = 89 clks
;       PM: 13                                  DM: 4
                MOVLW           8
                MOVWF           LOOPCOUNT
LOOPU0808A       RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                SUBWF           REMB0, F
                BTFSC           _C
                GOTO            UOK88A
                ADDWF           REMB0, F
                BCF             _C
UOK88A           RLF             AARGB0, F
                DECFSZ          LOOPCOUNT, F
                GOTO            LOOPU0808A
                endm

UDIV0807L       macro
```

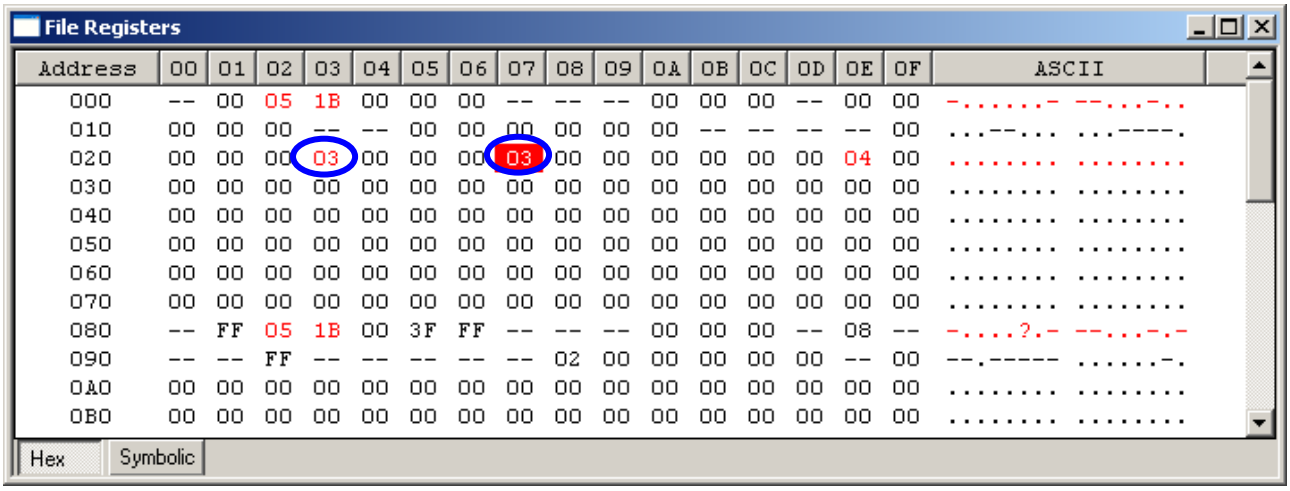
```
;       Max Timing:     7+6*11+10+2 = 85 clks
;       Min Timing:     7+6*11+10+2 = 85 clks
;       PM: 19                                  DM: 4

                RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                SUBWF           REMB0, F
                RLF             AARGB0, F
                MOVLW           7
                MOVWF           LOOPCOUNT
LOOPU0807       RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                BTFSC           AARGB0,LSB
                SUBWF           REMB0, F
                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F
                RLF             AARGB0, F
                DECFSZ          LOOPCOUNT, F
                GOTO            LOOPU0807
                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F
                endm

UDIV0707L       macro

;       Max Timing:     3+5+2+5*11+10+2 = 77 clks
;       Min Timing:     3+5+2+5*11+10+2 = 77 clks
;       PM: 22                                  DM: 4

                MOVF            BARGB0,W
                SUBWF           REMB0, F
                RLF             AARGB0, F
                RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                ADDWF           REMB0, F
                RLF             AARGB0, F
                MOVLW           6
                MOVWF           LOOPCOUNT
LOOPU0707       RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                BTFSC           AARGB0,LSB
                SUBWF           REMB0, F
                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F
                RLF             AARGB0, F
                DECFSZ          LOOPCOUNT, F
                GOTO            LOOPU0707
                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F
                endm

SDIV0808        macro
;       Max Timing:     3+5+6*8+2 = 58 clks
;       Min Timing:     3+5+6*8+2 = 58 clks
;       PM: 58                                  DM: 3

                variable i

                MOVF            BARGB0,W
                SUBWF           REMB0, F
                RLF             AARGB0, F
                RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                ADDWF           REMB0, F
                RLF             AARGB0, F
                i = 2
                while i < 8
                RLF             AARGB0,W
```

```
                RLF             REMB0, F
                MOVF            BARGB0,W
                BTFSC           AARGB0,LSB
                SUBWF           REMB0, F
                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F
                RLF             AARGB0, F
                i=i+1
                endw
                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F
                endm

UDIV0808  macro

;       restore = 9 clks,  nonrestore = 8 clks
;       Max Timing: 8*9 = 72 clks
;       Min Timing: 8*8 = 64 clks
;       PM: 72                                      DM: 3

                variable         i
                i = 0
                while i < 8

                RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                SUBWF           REMB0, F

                BTFSC           _C
                GOTO            UOK88#v(i)
                ADDWF           REMB0, F
                BCF             _C
UOK88#v(i)      RLF             AARGB0, F
                i=i+1
                endw
                endm

UDIV0807        macro

;       Max Timing:     5+7*8+2 = 63 clks
;       Min Timing:     5+7*8+2 = 63 clks
;       PM: 63                                      DM: 3

                variable i
                RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                SUBWF           REMB0, F
                RLF             AARGB0, F
                i = 1
                while i < 8
                RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                BTFSC           AARGB0,LSB
                SUBWF           REMB0, F
                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F
                RLF             AARGB0, F
                i=i+1
                endw
                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F
                endm

UDIV0707        macro

;       Max Timing:     3+5+6*8+2 = 58 clks
;       Min Timing:     3+5+6*8+2 = 58 clks
;       PM: 58                                      DM: 3
                variable i
                MOVF            BARGB0,W
                SUBWF           REMB0, F
```

```
                RLF             AARGB0, F
                RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                ADDWF           REMB0, F
                RLF             AARGB0, F
                i = 2
                while i < 8
                RLF             AARGB0,W
                RLF             REMB0, F
                MOVF            BARGB0,W
                BTFSC           AARGB0,LSB
                SUBWF           REMB0, F
                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F
                RLF             AARGB0, F
                i=i+1
                endw
                BTFSS           AARGB0,LSB
                ADDWF           REMB0, F
                endm

;*****************************************************************************************************
;*****************************************************************************************************
;       8/8 Bit Signed Fixed Point Divide 8/8 -> 08.08
;       Input:  8 bit signed fixed point dividend in AARGB0
;               8 bit signed fixed point divisor in BARGB0
;       Use:    CALL    FXD0808S
;       Output: 8 bit signed fixed point quotient in AARGB0
;               8 bit signed fixed point remainder in REMB0

;       Result: AARG, REM  <--  AARG / BARG

;       Max Timing:     21+77+5 = 103 clks              A > 0, B > 0
;                       22+77+10 = 109 clks              A > 0, B < 0
;                       22+77+10 = 109 clks              A < 0, B > 0
;                       23+77+5 = 105 clks              A < 0, B < 0
;                      6 clks                            A = 0

;       Min Timing:     21+77+5 = 103 clks              A > 0, B > 0
;                       22+77+10 = 109 clks              A > 0, B < 0
;                       22+77+10 = 109 clks              A < 0, B > 0
;                       23+77+5 = 105 clks              A < 0, B < 0

;       PM: 23+22+9+25 = 79                 DM: 7

FXD0808S        CLRF            SIGN
                CLRF        REMB0                       ; clear partial remainder
                MOVF            AARGB0,W
                BTFSC           _Z
                RETLW           0X00
                XORWF           BARGB0,W
                MOVWF           TEMP
                BTFSC           TEMP,MSB
                COMF            SIGN,F
                CLRF            TEMPB3                  ; clear exception flag
                BTFSS       BARGB0,MSB          ; if MSB set, negate BARG
                GOTO        CA0808S
                COMF        BARGB0, F
                INCF        BARGB0, F
CA0808S         BTFSS           AARGB0,MSB          ; if MSB set, negate AARG
                GOTO            C0808SX
                COMF            AARGB0, F
                INCF            AARGB0, F
C0808SX         MOVF            AARGB0,W
                IORWF           BARGB0,W
                MOVWF           TEMP
                BTFSC           TEMP,MSB
                GOTO            C0808SX1
C0808S          SDIV0808L
                BTFSC           TEMPB3,LSB              ; test exception flag
                GOTO            C0808SX4
C0808SOK        BTFSS           SIGN,MSB
                RETLW           0X00
```

```
                COMF            AARGB0, F
                INCF            AARGB0, F
                COMF            REMB0, F
                INCF            REMB0, F
                RETLW           0X00
C0808SX1        BTFSS           BARGB0,MSB              ; test BARG exception
                GOTO            C0808SX3
                BTFSC           AARGB0,MSB              ; test AARG exception
                GOTO            C0808SX2
                MOVF            AARGB0,W                ; quotient = 0, remainder = AARG
                MOVWF           REMB0
                CLRF            AARGB0
                GOTO            C0808SOK
C0808SX2        CLRF            AARGB0                  ; quotient = 1, remainder = 0
                INCF            AARGB0,F
                RETLW           0X00
C0808SX3        COMF            AARGB0,F                ; numerator = 0X7F + 1
                INCF            TEMPB3,F
                GOTO            C0808S
C0808SX4        INCF            REMB0,F                 ; increment remainder and test for
                MOVF            BARGB0,W                ; overflow
                SUBWF           REMB0,W
                BTFSS           _Z
                GOTO            C0808SOK
                CLRF            REMB0                   ; if remainder overflow, clear
                INCF            AARGB0,F                ; remainder, increment quotient and
                BTFSS           AARGB0,MSB              ; test for overflow exception
                GOTO            C0808SOK
                BSF             FPFLAGS,NAN
                RETLW           0XFF



;*****************************************************************************************
;*****************************************************************************************

;       8/8 Bit Unsigned Fixed Point Divide 8/8 -> 08.08

;       Input:  8 bit unsigned fixed point dividend in AARGB0
;               8 bit unsigned fixed point divisor in BARGB0

;       Use:    CALL    FXD0808U

;       Output: 8 bit unsigned fixed point quotient in AARGB0
;               8 bit unsigned fixed point remainder in REMB0

;       Result: AARG, REM  <--  AARG / BARG

;       Max Timing:     1+97+2 = 100 clks

;       Min Timing:     1+89+2 = 92 clks

;       PM: 1+13+1 = 15          DM: 4

FXD0808U        CLRF            REMB0
                UDIV0808L
                RETLW           0X00

                End
```

ผลการทำงานของโปรแกรมตัวอย่าง (จาก Simulator)



15 / 4 ผลลัพธ์ (0x27) = 3 เศษ (0x23) = 3

การตรวจสอบขนาด .hex file ที่ได้ ให้ทำการ load .hex เข้าไปในโปรแกรม WinPIC800 แล้วหาจุดสิ้นสุด (การดูขนาด .hex ไฟล์ใน windows จะไม่ถูกต้อง)