

COMENTARIO TÉCNICO

Buceando en los MCUs Freescale.....



Por Ing. Daniel Di Lella
Dedicated Field Application Engineer
EDUDEVICES

www.edudevices.com.ar
dilella@arnet.com.ar



www.edudevices.com.ar

“Matemática de Punto Flotante”

Por el CETAD – UNLP
Coordinador Ing. Jose Rapallini
Participantes:

Sebastián Ledesma
e-mail: sledesma@barcala.ing.unlp.edu.ar

Federico Costantino,
e-mail: fcostant@barcala.ing.unlp.edu.ar

Jorge R. Osio,
e-mail: josio@gioia.ing.unlp.edu.ar



3era. Parte.

Nota: La presente serie de artículos está basada en el trabajo realizado por el Centro de Técnicas Analógicas – Digitales (CETAD) de la universidad Nacional de La Plata, y constituye una Nota de Aplicación para los MCU's de 8 Bits de las familias HC908 y HC9S08 de Freescale Semiconductor.

Listado de materiales de la placa de prueba.

- 1 Microcontrolador de la Familia HC908.
- 1 conversor RS-232.
- 4 capacitores de 1 uF.
- 2 capacitores de 0,1 uF.
- 2 capacitores de 68 pF
- 1 resistencia de 10 MΩ.

- Un cristal de 10 Mhz.
- 1 conector DB9.
- 1 fuente de 5 V.

Código Fuente en lenguaje Assembler.

El código fuente fue desarrollado en gran parte en base a la información provista por el Ing. Gabriel Dubatti en [5]. En las rutinas de suma, resta, multiplicación y división se incluye la subrutina de comunicación SCI como un archivo INCLUDE.

Subrutina de suma y resta:

```

$BASE 10T
pta      Equ    $00
ptb      EQU    $01
DDRb     EQU    $05
DDRa     EQU    $04
         ORG    $80
caracter RMB    1
cuenta_bit RMB  1
aux      RMB    1
VAR      FCB    $00
FLOTA    RMB    4      ;Número A en punto flotante
EXPOA    DS     1
FLOTB    RMB    4      ;Número B en punto flotante
EXPOB    DS     1
OPERA    FCB    0      ;0 ES SUMA ;1 ES RESTA
FLOTR    DS     4      ;resultado de la suma
FLOTBX   DS     4      ;Auxiliar para la resta

         ORG    $EE00

         clr    $80
         clr    clrx
         lda    #$FE    ;setea el pin 0 del Ptb como
         sta    DDRb    ;entrada y el pin 1 como salida
         lda    #$01    ;setea el pin cero del Pta como
         sta    DDRa    ;salida
         sta    ptb     ;pone un 1 en pin 0 de Ptb
                     ;y un cero en los demás

         lda    #$00
         sta    pta
         bclr  1,pta,leedato ; si el pin 1 de pta es 0 hay suma
         lda    1       ;si es 1 hay una resta
         sta    OPERA
leedato   jsr    lee_caracter ;lee el 1° byte del 1° sumando
         lda    caracter
         sta    FLOTA    ; guarda primer byte en FLOTA
         jsr    lee_caracter ; lee segundo byte
         lda    caracter
         sta    FLOTA+1  ; guarda segundo byte en FLOTA+1
         jsr    lee_caracter ; lee tercer byte
         lda    caracter
         sta    FLOTA+2  ;guarda tercer byte en FLOTA+2
         jsr    lee_carácter ; lee cuarto byte
         lda    caracter
         sta    FLOTA+3  ;guarda cuarto byte en FLOTA+3
         bclr  0,pta
         jsr    lee_caracter ;lee el 1° byte del 2°sumando
         lda    caracter  ;guarda primer byte en FLOTB
         sta    FLOTB
         jsr    lee_caracter ; lee segundo byte
         lda    caracter
         sta    FLOTB+1  ; guarda segundo byte en FLOTB+1
         jsr    lee_caracter ; lee tercer byte
         lda    caracter
         sta    FLOTB+2  ; guarda tercer byte en FLOTB+2
         jsr    lee_carácter ; lee cuarto byte
         lda    caracter
         sta    FLOTB+3  ; guarda cuarto byte en FLOTB+3
         bclr  0,pta

```

;Parte Inicial: Testeo de casos especiales y obtención de los ;exponentes

```

MOV      FLOTB,FLOTBX
MOV      FLOTB+1,FLOTBX+1
MOV      FLOTB+2,FLOTBX+2
MOV      FLOTB+3,FLOTBX+3
LDA      OPERA
SUB      #1
BNE      AUXILIAR
LDA      FLOTBX
ADD      #128
STA      FLOTBX
AUXILIAR LDA      FLOTA      ;DE ACÁ PARA ABAJO CAMBIAR FLOTB CON FLOTBX
LDX      FLOTA+1
ROLX
ROLA
STA      EXPOA      ;Tengo en A el expoA
BNE      NORMAL      ;Salta si no es cero
LDA      FLOTA+1
AND      #127      ;Borro el LSB del exponente
ADD      FLOTA+2
ADC      FLOTA+3
BCS      NORMAL
BNE      NORMAL      ; A es cero por lo que la suma será B
LDHX     #FLOTBX      ; Si no es cero, A es un número
                        ; desnormalizado

NORMAL   JSR      COPIAR
JMP      FIN
ADD      #1      ; Pruebo si es $FF
BCC      NOFF      ; Salta si no hubo carry
LDHX     #FLOTA
JSR      COPIAR
INC      VAR
LDA      FLOTA+1
AND      #127      ;Borro el LSB del exponente
ADD      FLOTA+2
ADC      FLOTA+3
BCS      CASIFIN
BNE      CASIFIN      ;A es NaN, la suma será NaN, o sea A
                        ;Si no es cero, a es infinito

NOFF     LDA      FLOTBX
LDX      FLOTBX+1
ROLX
ROLA      ;Tengo en A el expoB
STA      EXPOB
BNE      NORMAL2      ; Salta si no es cero
LDA      FLOTBX+1
AND      #127
ADD      FLOTBX+2
ADC      FLOTBX+3
BCS      NORMAL2
BNE      NORMAL2
LDHX     #FLOTA
JSR      COPIAR
JMP      FIN

NORMAL2  ADD      #1      ; Pruebo si es $FF
BCC      NOFF2      ; Salta si no hubo carry
LDA      FLOTBX+1
AND      #127
ADD      FLOTBX+2
ADC      FLOTBX+3
BCS      COPIARB
BNE      COPIARB      ; B es NaN
                        ; Si no es cero, b es infinito

DEC      VAR
BNE      COPIAR      ;(Tengo que chequear operación para ver
                        ; si cambio el signo de b)
                        ; Si no saltó es porque ambos son infinito
BRA      NOSALTA      ; Estos dos saltos están porque no
                        ; llego desde
CASIFIN  BRA      FIN      ; mas arriba hasta el final.

```

```

NOSALTA      LDHX      #FLOTA
              LDA FLOTBX
ERASUMA      EOR        ,X
              ASLA
              BCC FIN      ;Si los signos son iguales el resultado
                          ; es infinito
              BSET      6,FLOTR+1 ;Si son distintos es QNaN -->busco
              INC      VAR      ; que la mantisa sea distinta de cero

NOFF2        DEC VAR      ;Ya tengo FLOTA en FLOTR, si era infinito
              BEQ FIN      ;ese es el resultado --> salgo

*****
;Segunda Parte: preparativos para realizar la suma o la resta
*****

              LDA      EXPOB
              BNE BUENO
              LDHX      #FLOTBX
              JSR COPIAR
              LDA EXPOA
              BEQ DOSDESN
              LDA      EXPOB
BUENO        LDX      EXPOA
              BEQ NOSE
              SUB      EXPOA      ;expB-expA
              BLS      MENORIGUAL ;expB<=expA
              INCA
NOSE         DECA
              STA      VAR
              SUB      #24      ;expB>expA ---> veo si la dif es mayor a 23
              BHS      COPIARB   ;Salta si la resta es menor que cero
              LDHX      #FLOTA
              JSR COPIAR
              LDA EXPOA
              BEQ NOSET
NOSET        BSET      7,FLOTR+1
              CLC

              JSR DESPLA      ;Después de desplazar puedo sumar
              LDHX      #FLOTBX
              JSR      TESTSIG ;Testeo los signos para ver si son
                          ;diferentes

              JSR      MODEXP
              BRA      FIN
COPIARB      LDHX      #FLOTBX
              JSR      COPIAR ; Copio FLOTB en FLOTR
              BRA      FIN
MENORIGUAL  NEGA
              STA      VAR
              SUB      #24      ;expA>expB -> veo si la dif es mayor a 23
              BHS      COPIARA
              LDHX      #FLOTBX
              JSR COPIAR      ; Si la diferencia es menor

              BSET      7,FLOTR+1
              LDA VAR
              BEQ DOSDESN
              CLC
              JSR DESPLA      ; Después de desplazar puedo sumar
DOSDESN:    LDHX      #FLOTA
              JSR      TESTSIG
              JSR      MODEXP
              BRA      FIN
COPIARA:    LDHX      #FLOTA
              JSR COPIAR      ; Copio FLOTA en FLOTR
FIN:        lda      FLOTR      ; se envía el resultado de la operación
              sta      caracter ; por puerto serie.
              jsr      saca_caracter
              lda      FLOTR+1
              sta      caracter
              jsr      saca_caracter
              lda      FLOTR+2
              sta      caracter
              jsr      saca_caracter
              lda      FLOTR+3

```

```

sta    character
jsr    saca_caracter
SWI

```

```

*****
;Subrutinas empleadas en la operación matemática
*****

```

```

COPIAR:      MOV     X+,FLOTR      ;Rutina para copiar
              MOV     X+,FLOTR+1
              MOV     X+,FLOTR+2
              MOV     X+,FLOTR+3
              RTS

SUMA:        LDA     3,X          ;Rutina de suma y resta
              ADD     FLOTR+3
              STA     FLOTR+3
              LDA     2,X
              ADC     FLOTR+2
              STA     FLOTR+2
              LDA     1,X
              ORA     #128
              ADC     FLOTR+1
              STA     FLOTR+1

              ;Acá podría chequear si la resta da cero
              ;pero como pienso incluir la subrutina de resta
              ;lo dejo para después

              DEC     VAR          ; Tengo que establecer si fue suma o resta
              BNE     SALE
              CLC
              RTS

SALE

COMP2:       LDA     FLOTR+1
              COMA
              STA     FLOTR+1      ;Rutina de complemento a 2
              LDA     FLOTR+2
              COMA
              STA     FLOTR+2
              LDA     FLOTR+3
              COMA
              ADD     #$1
              STA     FLOTR+3
              LDA     FLOTR+2
              ADC     #$0
              STA     FLOTR+2
              LDA     FLOTR+1
              ADC     #$0
              STA     FLOTR+1
              RTS

DESPLA:      LDA     VAR
              BEQ     NO
              ROR     FLOTR+1      ;Rutina para desplazar mantisa
              ROR     FLOTR+2      ;La primera no es LSR porque uso la subrutina
              ROR     FLOTR+3      ;para acomodar el exponente
              CLC
              DEC     VAR
              BNE     DESPLA

NO

TESTSIG:     CLR     VAR          ;Si los signos son iguales hago la suma
              LDA     FLOTR      ;y coloco el signo de uno de los dos
              EOR     ,X
              ASLA
              BCC     IGUAL        ; Si son diferentes hago la resta y el signo
              MOV     #1,VAR      ;será el del mayor exponente
              JSR     COMP2

IGUAL        JSR     SUMA
              RTS

```

```

MODEXP:      MOV    #2,VAR    ;Tengo que volver a normalizar el número
             ROL    VAR      ; y modificar el exponente
             LDA    1,X      ; Levanto el byte con el LSB del
                               ; exponente

             STA    FLOTR
             LSL    FLOTR
             LDA    ,X      ;Levanto el byte que contiene el resto del exp
                               ;y el signo

             ROLA
             BCC    POS
             BSET   1,VAR    ;Si es negativo
POS:         LSR    VAR
             BCS    YA
             INCA
NUEVO:      SUB    #1        ; El número es menor que uno
             BEQ    RECUPSIG ; Desplazo hacia izquierda y
                               ; decremento exponente
             ASL    FLOTR+3  ; Si al decrementar obtengo exp
                               ;cero es desnorm.

             ROL    FLOTR+2
             ROL    FLOTR+1
             BCC    NUEVO
             BRA    RECUPSIG
YA:         ADD    #2        ; El exp FF está reservado
             BCC    ETIQUETA ; Si C=1 --> infinito
             CLR    FLOTR+1  ; Borro la mantisa para formar el infinito
             CLR    FLOTR+2
             CLR    FLOTR+3
ETIQUETA    DECA          ; Tengo el exponente
RECUPSIG    LSR    VAR      ; Agrego el signo
             RORA
             STA    FLOTR
             BRA    DESPLA  ; Inserto el LSB del exponente a FLOTR+1
                               ;En VAR tengo 1

```

\$INCLUDE "SCI_TXTR.INC";inclusión del archivo de comunicación serie

```

Org    $FFFA
dw     $EE00 ;IRQ
dw     $EE00 ;SWI
dw     $EE00 ;Reset

```

Subrutina de Multiplicación:

\$BASE 10T

```

part    org    $80
part    ds    1
part1   ds    1
part2   ds    1
part3   ds    1
expl    ds    1
exp2    ds    1
nbytes  ds    1
nbytes2 ds    1
numA    db    $00,$7F,$FF,$FF
numB    db    $40,$49,$0F,$DA
partalt ds    1
acum    ds    1
resp    ds    7
temp    ds    1

```

```

        org      $EE00

clrresp  ldx      #6          ;Pone a cero todo el espacio para la mantisa
otra     clr      resp,x
        dbnzx    otra

        mov      #3,nbytes
        mov      #3,nbytes2
        mov      #6,part3
        mov      #5,part2
        mov      #4,part1

*****
;1° Detección de cifras desnormalizadas
*****

inicio   clr      acum
        clr      temp
        mov      numA,exp1
        lda      (numA+1)
        rola
        rol      exp1
        lda      exp1
        bne      nodes1A
        mov      #$FF,acum      ; numA es desnormalizado
nodes1A  lda      exp1
        bpl      esnegat1      ; N=0 => el exp es negativo, N=1 => el exp es
positivo
        inc      temp
        bra      esposit1
        esnegat1 dec      temp
        esposit1 mov      numB,exp2
        lda      (numB+1)
        rola
        rol      exp2
        lda      exp2
        bne      nodes1B
        clc
        lda      acum
        add      #1          ; numB es desnormalizado
        bcs      alceros      ; Ambos números son desnormalizados => resultado=0
        sta      acum
nodes1B  bpl      esnegat2      ;N=0 =>el exp es negativo, N=1 =>el exp es positivo
        inc      temp
        bra      first
        esnegat2 dec      temp

*****
;2° Deteccion de los QNaN, SNaN, Cero e Infinito
*****

first    clr      part
        bclr     7,numA+1
        bclr     7,numA+1
        clc
        lda      numA+3
        add      numA+2
        adc      numA+1
        bcs      sigo3
        bne      sigo3
        lda      exp1
        beq      sigo
        cmp      #$FF
        beq      sigo2
        bra      second
sigo     inc      part          ; numA=0
        bra      second
sigo2   dec      part          ; numa = infinito
        bra      second
sigo3   lda      exp1
        cmp      #$FF
        beq      SNaN

```

```

second    clc
          lda  numB+3
          add  numB+2
          adc   numB+1
          bcs  sigo4
          bne  sigo4
          lda   exp2
          beq   sigo5
          cmp   #$FF
          bne   sumar
          dec  part
          beq  QNaN
          jmp  infin
sigo5     inc  part
          beq  QNaN
          jmp  cero
sigo4     lda  exp2
          cmp  #$FF
          beq  SNaN
          dec  part
          beq  alcerero
          inc  part
          inc  part
          beq  alinfin

*****
;3° Suma de los exponentes
*****

sumar     lda  exp1
          add  exp2
          sub  #127
          bpl  sumnegat
          sta  exp1           ; La suma es positiva
          lda  temp
          bmi  alcerero       ; El exponente es muy chico(o sea cero)
          bra  desdetec

*****

alcerero  jmp  cero

alinfin   jmp  infin

SNaN      mov  #0,resp         ; Uno o ambos números son no validos
          bset 0,resp+1
          jmp  fin

QNaN      mov  #0,resp         ; La operación da valor no valido
          bset 6,resp+1
          jmp  fin

desno2A   rol  numA,x
          dbnzz desno2A
          rts

desno2B   rol  numB,x
          dbnzz desno2B
          rts

infin     mov  #$FF,resp       ;El resultado es +/- infinito
          ror  resp
          ror  resp+1
          jmp  signo

cero      mov  #0,resp         ;El resultado es cero
          jmp  fin

*****

sumnegat  sta  exp1           ; La suma es negativa
          lda  temp
          beq  desdetec
          bpl  alinfin        ; El exponente es muy grande(+/- infinito)

```

;4° Se adecuan las cifras desnormalizadas

```

desdetec  mov     expl,resp
          bset   7,(numA+1)
          bset   7,(numB+1)
          clc
          ldx   #3
          lda   acum
          add   #1
          bcc   nodes2A
          clc
          jsr   desno2A
          bra   mulmant
nodes2A   dec   acum
          bne   mulmant
          clc
          bsr   desno2B

```

;5° Multiplicación de la mantisa

```

mulmant   ldx   nbytes
despacum  clr   partalt
          lda   numB,x
          sta   acum
          lda   part1-1,x
          sta   part
          pshx
          bsr   multacum
          pulx
          dbnzx despacum
          bra  despmant

multacum  ldx   nbytes2
          clc
seguir    lda   numA,x
          rolx
          pshx
          ldx   acum
          mul
          add   partalt
          bcc   noincre
          incx
noincre   stx   partalt
          pulx
          lsrx
          pshx
          ldx   part
          adc   resp,x
          sta   resp,x
          decx
          stx   part
          pulx
          decx
          bne   seguir
          bcc   noincre2
          inc   partalt
noincre2  ldx   part
          lda   partalt
          sta   resp,x
          rts

```

```
*****
;6° Arreglo de la presentación de la cifra
*****
```

```
despmant  mov    #2,part
           brclr  7,resp+1,multdeci
           inc   resp
           bra   yaincre

multdeci  lda    resp
           beq   signo
           brset  6,resp+1,sigue3
           ldx   #4
sigue4    rol   resp,x
           dbnzx sigue4
           dec   resp
           bra   multdeci

sigue3    clc
sigue2    ldx   #4
sigue     rol   resp,x
           dbnzx sigue

yaincre   dec   part
           bne   sigue2
           clr   clrx
           lda   #4
sigue5    ror   resp,x
           incx  deca
           bne   sigue5
           bra   signo
```

```
*****
;7° Se establece el signo
*****
```

```
signo     bclr   7,resp           ;Rutina que determina el signo
           lda   numA
           eor   numB
           and   #$80
           beq   fin
           bset  7,resp
fin       swi
```

Continuará

Nota de Redacción: El lector puede descargar este artículo y artículos anteriores de “*Buceando...*” desde la sección “*Artículos Técnicos*” en el sitio web de **EduDevices** (www.edudevices.com.ar)



WWW.EDUDEVICES.COM.AR