;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

;

;

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AREA IEEEtoTNS, CODE

SWI\_write EQU &0 ;print r0

SWI\_exit EQU &11 ;exit the program

ENTRY

Main

ADR r1, IEEE ;Store IEEE

LDR r1, [r1] ;Load register

ADR r2, TNS ;Load TNS

LDR r2, [r2] ;Load register

ADR r7, signMask ;Load sign mask for IEEE to TNS

LDR r7, [r7] ;load register

ADR r8, IEEXPMask ;Load exponent mask for IEEE to TNS

LDR r8, [r8] ;load register

ADR r9, IESigMask ;Load significand mask for IEEE to TNS

LDR r9, [r9] ;load register

ADR r3, zeroSet ;Load zero hex

LDR r3, [r3] ;load register

MOV r4, r1 ;upload IEEE to register for sign mask

AND r4, r4, r7 ;mask sign bit

MOV r5, r1 ;upload IEEE to register for EXP mask

AND r5, r5, r8 ;mask exponent bits

MOV r6, r1 ;upload IEEE to register for Significand mask

AND r6, r6, r9 ;mask significand bits

MOV r5, r5, LSR #23 ;shift exponent to right place

SUB r5, r5, #127 ;subtract 127 from exponent

ADD r5, r5, #256 ;add 256 to exponent for right excess

MOV r6, r6, LSL #8 ;shift significand by 8 bits

ORR r3, r3, r4 ;unpack sign bit

ORR r3, r3, r5 ;unpack exponent

ORR r3, r3, r6 ;unpack significand

CMP r3, r2 ;compare with converted value

ADREQ r12, Good2 ;if equal load string

BEQ PrintMessage1 ;if equal go to print string

BNE notEqual ;if not equal go to notEqual

PrintMessage1 LDRB r0, [r12], #1 ;load next byte of string

CMP r0, #0 ;check null terminator

SWINE SWI\_write ;if not equal print character

BNE PrintMessage1 ;if not equal go to PrintMessage1

AddNewLine1 MOV r13, #0 ;load r13 with 0

ADR r13, NewLine ;load string

LDRB r0, [r13], #1 ;load next byte

SWI SWI\_write ;print character

Print MOV r11,#0 ;load with 0

MOV r11,#8 ;set number or nibbles to 8

LOOP MOV r0,r3,LSR #28 ;set top nibble and store it

CMP r0, #9 ;hexanumber 0-9 or A-F

ADDGT r0,r0, #"A"-10 ;ASCII alphabetic

ADDLE r0,r0, #"0" ;ASCII numeric

SWI SWI\_write ;print character

MOV r3,r3,LSL #4 ;shift left one nibble

SUBS r11,r11, #1 ;decrement nibble count

BNE LOOP ;loop back if more nibbles

B ClearBits ;return

TNStoIEEE ADR r7, signMask ;Store sign mask for IEEE to TNS

LDR r7, [r7] ;Load sign mask

ADR r8, TNSEXPMask ;Store exponent mask for IEEE to TNS

LDR r8, [r8] ;Load exponent mask

ADR r9, TNSSigMask ;Store significand mask for IEEE to TNS

LDR r9, [r9] ;Load significand mask

ADR r3, zeroSet ;Store zeroSet

LDR r3, [r3] ;Load zeroSet

MOV r4, r2 ;Load IEEE to register for sign mask

AND r4, r4, r7 ;Mask sign bit

MOV r5, r2 ;upload IEEE to register for EXP mask

AND r5, r5, r8 ;mask exponent bits

MOV r6, r2 ;upload IEEE to register for Significand mask

AND r6, r6, r9 ;mask significand bits

MOV r6, r6, LSR #8 ;shift significand by 8 bits

SUB r5, r5, #256 ;subtract 256 from exponent

ADD r5, r5, #127 ;add 127 to exponent for right excess

MOV r5, r5, LSL #23 ;shift exponent to right place

ORR r3, r3, r4 ;unpack sign bit

ORR r3, r3, r5 ;unpack exponent

ORR r3, r3, r6 ;unpack significand

CMP r3, r1 ;compare with converted IEEE to original

ADREQ r12, Good ;if euqal load string

BEQ AddNewLine2 ;if equal go to AddNewLine1

BNE notEqual ;if not equal go to not equal

AddNewLine2 MOV r13, #0 ;load 0

ADR r13, NewLine ;load string

LDRB r0, [r13], #1 ;load next byte

SWI SWI\_write ;print character

PrintMessage LDRB r0, [r12], #1 ;load next byte

CMP r0, #0 ;check for null terminator

SWINE SWI\_write ;if not equal print character

BNE PrintMessage ;if not equal go to PrintMessage

Print2 MOV r12, #0 ;empty register

ADR r12, NewLine ;store string

LDRB r0, [r12], #1 ;load next byte

SWI SWI\_write ;print character

MOV r0, #0 ;empty register

MOV r13,#8 ;set number or nibbles to 8

LOOP2 MOV r0,r3,LSR #28 ;set top nibble and store it

CMP r0, #9 ;hexanumber 0-9 or A-F

ADDGT r0,r0, #"A"-10 ;ASCII alphabetic

ADDLE r0,r0, #"0" ;ASCII numeric

SWI SWI\_write ;print character

MOV r3,r3,LSL #4 ;shift left one nibble

SUBS r13,r13, #1 ;decrement nibble count

BNE LOOP2 ;loop back if more nibbles

B Exit ;go to exit

notEqual MOV r13, #0 ;load 0

ADR r13, NewLine ;load string

LDRB r0, [r13], #1 ;load next byte

SWI SWI\_write ;print character

MOV r12, #0

ADR r12, wrong

endMessage LDRB r0, [r12], #1 ;load next byte

CMP r0, #0 ;check for null terminator

SWINE SWI\_write ;if not equal print character

BNE endMessage ;if not equal go to PrintMessage

BEQ Exit

ClearBits MOV r3, #0 ;reset bits

MOV r4, #0 ;reset bits

MOV r5, #0 ;reset bits

MOV r6, #0 ;reset bits

MOV r7, #0 ;reset bits

MOV r8, #0 ;reset bits

MOV r9, #0 ;reset bits

BL TNStoIEEE ;go to TNStoIEEE

Exit SWI SWI\_exit ;exit program

IEEE DCD 0x42144000 ;37.0625 in IEEE

TNS DCD 0x14400105 ;37.0625 in TNS

zeroSet DCD 0x00000000

signMask DCD 0x80000000

IEEXPMask DCD 0x7f800000

IESigMask DCD 0x007fffff

TNSEXPMask DCD 0x000001ff

TNSSigMask DCD 0x7ffffc00

NewLine DCD &0a, 0

Good DCB "The TNS to IEEE conversion is good!",0

Good2 DCB "The IEEE to TNS conversion is good!",0

wrong DCB "This conversion is wrong.", 0

END