Midterm #1: Rework Problems 3 & 4 - 03/06/13

Problem 3

Write an ARM assembly language program to perform the following C "for loop":

```
for (I = 0; I<300; I=I+1)
      M[I+1] = (N[I] * 23.25) + M[I];
```

All numbers are to be in Q28.4 format. Variables are stored in the following data area. Arrays M and N have previously been loaded with data.

midterm_problem3.s

```
1  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2  ;; Brian Arnberg                                   ;;
3  ;; Midterm − Problem 3                             ;;
4  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
5  ;; for (I=0;I<300;I++)                             ;;
6  ;;    M[I+1] = (N[I]*23.25) + M[I];                ;;
7  ;; ; all numbers are Q28.4 format                  ;;
8  ;; ; M and N are already populated                 ;;
9  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
10          AREA variables ,DATA
11 M        SPACE 4*300        ;300 32−bit words
12 N        SPACE 4*300        ;300 32−bit words
13 I        SPACE 4            ;1 32−bit word
14 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
15          AREA for_loop ,CODE,READONLY
16          LDR r8 , =M            ;; initial address for M
17          LDR r9 , =N            ;; initial address for N
18          LDR r7 , =I            ;; address for I
19          MOV r2 , #0            ;; initialize I
20 loop
21          LDR r4 , [r8] , #4     ;; M: load r4 from r8, then index r8
22          LDR r5 , [r9] , #4     ;; N: load r5 from r9, then index r9
23          MOV r3 , #0x174       ;; 23.25 = 0x174 in Q28.4
24          MUL r5 , r5 , r3      ;; N[I]*23.5
25          LSR r5 , #4           ;; shift right 4 bits
26          AND r4 , r5          ;; calculate result
27          STR r4 , [r8]         ;; store to M[I+1]
28          ADD r2 , r2 , #1      ;; I = I + 1
29          CMP r2 , 300          ;; if (I < 300)
30          BLT loop              ;; branch to ``loop''
31          STR r2 , [r7]         ;; store I
32 h        B h                   ;; end
33          END
```

Problem 4

A digital to analog converter contains an 8-bit DATA register (address 0xE0000000) and an 8-bit STATUS register (address 0xE0000001). Conversion of an 8-bit datum from digital to analog form begins when the byte is written to the DATA register. As soon as the buye is written to DATA, the left-most bit of STATUS (called BUSY) is set to 1 and conversion begins. BUSY remains 1 while the conversion takes place and returns to 0 when the conversion is complete.

Using "busy-wait" output, write a short subroutine in ARM assembly language that will convert, from digital to analog, 200 bytes stored in an array beginning at address 0x40000000 in memory.

midterm_problem4.s

```
1  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2  ;; Brian Arnberg                                    ;;
3  ;; Midterm − Problem 4                              ;;
4  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
5  ;; 8−bit: STATUS 0xE0000001                         ;;
6  ;;         DATA   0xE0000000                         ;;
7  ;;         ARRAY  0x40000000 (200 bytes long)        ;;
8  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
9  DATA     EQU 0xE0000000
10 STATUS   EQU 0xE0000001
11 ARRAY    EQU 0x40000000
12
13         AREA CONVERT,CODE
14         EXPORT convert
15 convert
16         LDR  r0 , =STATUS         ;; address of STATUS byte
17         LDR  r1 , =DATA           ;; address of DATA byte
18         LDR  r2 , =ARRAY          ;; initial address of ARRAY
19         LDR  r5 , =ARRAY
20         ADD  r5 , r5 , #200       ;; calculate end of ARRAY
21
22 again   LDRB r3 , [r2 , #1]!      ;; load byte value from ARRAY, increment
23         STRB r3 , [r1]            ;; store byte to DATA for conversion
24 loop    LDRB r4 , [r0]            ;; load STATUS into r4
25         TST  r4 , #0x80           ;; test leftmost bit of STATUS
26         BEQ  loop                 ;; if leftmost bit of STATUS is 1, loop
27         CMP  r2 , r5              ;; if (r2 < r5) ; (at the end of the array
28         BLT  again                ;; loop back to ''again''
29         BX   r14                  ;; return from subroutine
30         END
```