ECED 3204 Microprocessor

Assignment #2 Reference Solution

http://www.jasongu.org/3204/assignments.html

Assignment #2 contains the following problems:

E3.3 Invoke the Setpointer macro defined in Section 3.5.8 to load the address 0x3000 into Y register.

```
        setPointer  YL,YH,0x3000
```

E3.7 Write a program to subtract the 16 bit number stored at 0x200-0x201 from the 16 bit number stored at 0x202-0x203, and store the difference at 0x204-0x205.

Subtraction is proceeded from lsb toward msb.

```
        .include <m2560def.inc>
        .cseg
        .org    0x00
        rjmp    start
        .org    0xF6
start:  lds     r1,0x202
        lds     r0,0x200
        sub     r1,r0
        sts     0x204,r1
        lds     r1,0x203        ; fetch the high byte of minuend
        lds     r0,0x201        ; fetch the high byte of subtrahend
        sbc     r1,r0           ; subtract with borrow
        sts     0x205,r1        ; save the high byte of the difference
// End of program
```

E3.11 Write a program to swap the upper 4 bis with the lower 4 bits of every element of an array of 8-bit integers. The initial array is stored in the program memory ad you must copy this arrary to the data memory for testing purpose.

We can use the Z register to point to the array in the program memory and use the register X to point to the buffer to receive the array after swapping the upper and lower nibbles. The program is as follows:

```
        .include    <m2560def.inc>
```

```
            .equ    NN=30
            .dseg
            .org    0x200
buf1:       .byte   NN              ; buffer to hold the array
            .def    lpcnt = r17
            .cseg
            .org    0x00
            rjmp    start
            .org    0xF6
start:      ldi     XL,low(buf1)    ; set up pointer to buf1
            ldi     XH,high(buf1)   ;    "
            ldi     ZL,low(array<<1) ; set up pointer to array
            ldi     ZH,high(array<<1)   ;        "
            ldi     lpcnt,NN        ; set up loop count to copy array
cloop:      lpm     r0,z+           ; loop to copy array to data memory
            swap    r0              ; swap the upper and lower nibbles
            st      x+,r0           ; store the resultant element in data memory
            dec     lpcnt
            brne    cloop
here:       rjmp    here
array:      .db     0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19,0x20
            .db     0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29,0x30
            .db     0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x40
// end of program
```

E3.15 write program to set the bit 7 and bit 0 of all the elements of an array to 1.

The following program sets bit 7 and bit 0 of the array elements in program memory and store the result in a data memory buffer:

```
            .include <m2560def.inc>
            .macro  setPointer
            ldi     @0,low(@2)
            ldi     @1,high(@2)
            .endmacro
            .def    lpcnt=r16       ; loop count for performing left-shifting
            .def    tmp = r17
            .equ    NN=20
            .dseg
            .org    0x200
buffer:     .byte   30
            .cseg
            .org    0x00
            rjmp    start
            .org    0xF6
start:      ldi     lpcnt,NN
```

```
        setPointer ZL,ZH,(tstdat<<1) ; Z is the pointer to test data
        setPointer XL,XH,buffer
again:  lpm     tmp,z+      ; read one array element
        ori     tmp,0x81    ; set bit 7 and bit 0 to 1
        st      x+,tmp      ; store the result in destination
        dec     lpcnt
        brne    again
        nop
tstdat: .db     0x12,0x14,0x16,0x26,0x35,0x33,0x18,0x46,0x56,0x38
        .db     0x2A,0x3B,0x4A,0x5C,0x6C,0x78,0x78,0x82,0x6A,0x5C
// end of the program
```

E3.19 Write a program to copy a block of memory locations from one area to another area in data memory.

The program first copies an array of test data from program memory to data memory. The program then sets up pointer to that test data and copies it to another area. The program is as follows:

```
        .include <m2560def.inc>
        .macro  setPointer
        ldi     @0,low(@2)
        ldi     @1,high(@2)
        .endmacro
        .dseg
buf1:   .byte   20
buf2:   .byte   20
        .def    lpcnt=r16
        .equ    NN = 20
        .cseg
        .org    0x00
        rjmp    start
        .org    0xF6
start:  setPointer ZL,ZH,(tstDat<<1)
        setPointer XL,XH,buf1
        ldi     lpcnt,NN        ; set up loop for transferring test data
cploop: lpm     r0,z+           ; copy test data to buf1
        st      x+,r0
        dec     lpcnt
        brne    cploop
        setPointer XL,XH,buf1   ; use X as pointer to buf1
        setPointer YL,YH,buf2   ; use Y as pointer to buf2
        ldi     lpcnt,NN        ; set up loop count for data transfer
; block transfer from buf1 to buf2
tloop:  ld      r0,x+           ; read one byte
        st      y+,r0           ; store one byte
```

```
        dec     lpcnt
        brne    tloop
        nop
tstDat: .db     0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19,0x20
        .db     0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29,0x30
// end of program
```