

ELEC 2220 - Computer Systems
Homework #11
Due: Wednesday, June 17

PROGRAM 1

To exercise the logical operations, write a program that works with three one-byte data variables labeled STATUS, CONTROL and PERIPH. The program is to perform the following operations.

- If bit 0 of STATUS is 1, set bit 6 of CONTROL to 1, otherwise set bit 0 of CONTROL to 0.
- If bit 7 of STATUS is 0 and bit 3 of STATUS is 0, set bit 2 of CONTROL to 1, otherwise set bit 2 of CONTROL to 0.
- Replace bits 5-2 of PERIPH with the hex digit 5, without changing the other four bits of PERIPH.

Data are stored in “little endian” format (bits are numbered 7 down to 0, from left to right, within each byte.)

Test the program by initializing STATUS to the value 0x43, CONTROL to 0x00, and PERIPH to 0x77. Display these values in a memory window (as bytes), and capture that window to show the final values of STATUS, CONTROL, and PERIPH.

Repeat for STATUS initialized to 0x88, CONTROL to 0xff, and PERIPH to 0xb5.

PROGRAM 2 on next page.

PROGRAM 2

Eight ASCII character codes, representing eight decimal digits as might be entered from a keyboard, are stored eight consecutive bytes of memory beginning at address STRG in RAM. Write a program to convert this string of characters to a *packed BCD* (Binary-Coded Decimal) number and store it in the 32-bit word at data address PACK. (A program might need to do this when numbers are entered from a keyboard.) “Packed BCD” format uses four bits to represent each decimal digit, with two BCD digits packed into one byte (less-significant digit in the low half of the byte).

For example, if you were to type the number 13587609 on the keyboard, the characters would be stored in memory as follows:

```
STRG   => '1' (0x31) - Most significant digit (10,000,000's)
STRG+1 => '3' (0x33) - Next most significant digit (1,000,000's)
STRG+2 => '5' (0x35) - Next most significant digit (100,000's)
STRG+3 => '8' (0x38) - Next most significant digit (10,000's)
STRG+4 => '7' (0x37) - Next most significant digit (1,000's)
STRG+5 => '6' (0x36) - Next most significant digit (100's)
STRG+6 => '0' (0x30) - Next most significant digit (10's)
STRG+7 => '9' (0x39) - Least significant digit (1's)
```

Your program is to convert this string of characters to packed BCD format. The packed BCD number corresponding to the above would be stored in one 32-bit word of memory (least significant byte first) as follows:

Word: PACK => 0x13587609

Equivalent to bytes:

```
PACK: 0x09,  PACK+1: 0x76,  PACK+2: 0x58,  PACK+3: 0x13
      (LSB)                                     (MSB)
```

Note that this 32-bit word also corresponds to the size of a register. You will likely need shift operations in this program.

Demonstrate your program in the debugger by initializing the eight bytes of STRG as shown above, and the 32-bit value of PACK to 0. Capture and submit the debug memory window showing the final values of these memory locations after executing the program.

Consider also testing your program on one additional set of data, of your choosing (not required).