

PES, Section 2.4
Hexadecimal

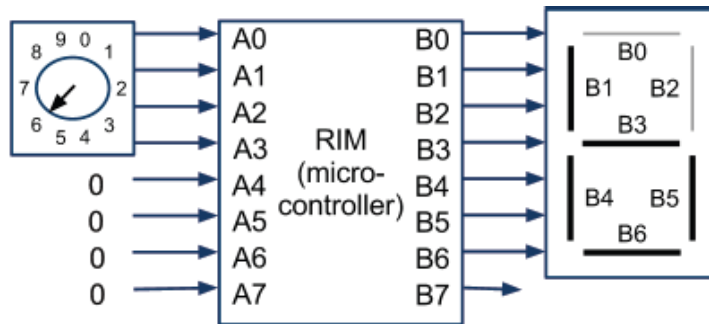
1. Rewrite the following single-line RIMS-compatible C statement to use hexadecimal.

`B = 247;`
`B = 0xF7;`

2. Write a single statement for RIMS that sets B0 to 1 if A3-A0 are all 1s, and A7-A4 are all zeroes. (Use a hex constant).

`B0 = (A == 0x0F);`

3. Consider the following embedded system with a dial that can set A3..A0 to binary 0 to 9, and a 7-segment display.



On the next page is a (partial) RIMS C program that appropriately sets the display for the given dial position.

A new company has created a new dial that can represent all 16 4-bit combinations, from 0-F. Extend the partial RIMS C program to display the six hexadecimal characters larger than 9 (pay attention to upper/lowercase):

A
b
C
d
E
F

```

#include "RIMS.h"
void main()
{
    while (1) {
        switch( A ) {
            case 0 : B = 0x77; break;    // 0111 0111 (0)
            case 1 : B = 0x24; break;    // 0010 0100 (1)
            case 2 : B = 0x5d; break;    // 0101 1101 (2)
            //...
            case 9 : B = 0x6f; break;    // 0110 1111 (9)

            case 10 : B = 0x3f; break;    // 0011 1111 (A)
            case 11 : B = 0x7A; break;    // 0111 1010 (b)
            case 12 : B = 0x53; break;    // 0101 0011 (C)
            case 13 : B = 0x7C; break;    // 0111 1100 (d)
            case 14 : B = 0x5B; break;    // 0101 1011 (E)
            case 15 : B = 0x1B; break;    // 0001 1011 (F)

            // 1000 0000 (Activate B7 to indicate an error)
            default: B = 0x80; break;
        }
    }
}

```