Hexadecimal Arithmetic Using a Binary/Octal Digit Code

Hexadecimal arithmetic can be simplified using a combined binary/octal code to represent each hexadecimal digit. In this code, each hexadecimal digit occupies three rods: the right-most of the three (on the soroban this will be a marked units rod) contains the octal number which is the modulo eight remainder of the hexadecimal digit while the next rod to the left is binary and holds a zero if the hexadecimal number is less than eight or a one if it is greater than or equal to eight. The third rod to the left is simply a spacer and never used in the calculations. Only three earth beads are used on the octal rod and the heaven bead counts for four, not the usual five. The octal rod only holds counts up to seven, and the binary rod is normally either zero or one, but may temporarily hold a greater count before being normalized.

Hexadecimal	Binary/Octal Code	Binary rod	Bead setting on octal rod
0	000	0	H = 0, 0 earth beads set
1	001	0	H = 0, 1 "
2	002	0	H = 0, 2 "
3	003	0	H = 0, 3 "
4	004	0	H = 1, 0 "
5	005	0	H = 1, 1 "
6	006	0	H = 1, 2 "
7	007	0	H = 1, 3 "
8	010	1	H = 0, 0 earth beads set
9	011	1	H = 0, 1 "
A	012	1	H = 0, 2 "
B	013	1	H = 0, 3 "
C	014	1	H = 1, 0 "
D	015	1	H = 1, 1 "
E	016	1	H = 1, 2 "
F	017	1	H = 1, 3 "

In addition, when the count on the octal rod exceeds seven a carry is propagated to the binary left neighbor rod. If this increment causes the binary rod to exceed one, the rod must be normalized by subtracting from it the largest possible multiple of two and adding one half this amount to the octal rod of the left neighboring hexadecimal digit, thus restoring the binary rod to a legitimate value of either zero or one.

In subtraction, when a number is subtracted from a smaller number on the octal rod, there must be a borrow from the binary left neighbor rod and eight added to the octal rod. If the binary rod is zero then the borrow must come from the octal rod of the left neighbor hexadecimal digit – this rod will be decremented by one and two added to the binary rod from which we needed to borrow. Then the borrow will be executed by subtracting one from the binary rod and adding eight to the octal rod.

Addition example: EC + BD = 1A9

Convert to binary/octal code:
0 000 016 014
+ 000 013 015
----

0 000 031 031 add (remember, octal addition)
0 000 032 011 normalize – subtract 2 from 3 in 031, add 1 to left neighbor octal rod
0 001 012 011 normalize again
=   1   A   9

Subtraction example: D4 – A9 = 2B

Convert to binary/octal code:
0 000 015 004
- 000 012 011

?
here we must borrow from the left neighbor digit, so decrement the 5 and add 2 to the binary rod
0 000 014 024
- 000 012 011
----
0 000 002 013
=   0   2   B

This method can be extended to multiplication in an obvious manner, but we must be aware that the product of a binary rod times a binary rod will produce either a zero or a four onto an octal rod; a binary rod product is either 0 x 0, 0 x1, 1 x 0, or 1 x 1. The 1 x 1 product represents 8 x 8 = 64 = 4 x 16, hence four added to the left neighbor octal rod. The product of an octal rod times an octal rod or an octal rod times a binary rod yields an ordinary octal product. As in addition, use octal addition on the rightmost (octal) rod of each hexadecimal digit and add carries, if any, into the center rod. Then normalize in order to get the center (binary) rod back to a valid binary number; i.e., zero or one. Remember that when normalizing, you must subtract the largest even number possible from the center rods and add one-half that number to the rightmost (octal) rod of the next digit to the left.

Simple multiplication example: 2A x 3B = 9AE

Write the multiplicand and multiplier in binary/octal format:

	002 012	
x	003 013	
	----------	
	006 036	multiply by octal 3 of right-hand multiplier digit
	007 016	normalize 036 (subtract 2 from 3, add 1 to 6)
+	2	add binary 1 times octal 2 of right-hand digit
	----------	
	007 036	
	010 016	normalize 036 (and increment octal 007 to octal 010)
+	4	add binary 1 times binary 1 of right-hand digit (=4 !)

```
           014 016
   +         2             add binary 1 times octal 2 of left-hand digit
           ------------
           034 016
       001 014 016         normalize 034
   +           6           add octal 3 times octal 2 of right-hand digit
      -----------------
       001 022 016         (recall that 4 + 6 = 12 in octal)
       002 002 016         normalize 022
   +         3             add octal 3 times binary 1 of right-hand digit
       -----------------
        002 032 016
        003 012 016        normalize 032
   +      6                add octal 3 times octal 2 of left-hand digit
       ------------------
        011 012 016        (octal 3 + 6 = octal 11)
          9    A    E      the hexadecimal form of the solution
```

A more complicated example: A6B x 7D9 = 51C1B3

Write in binary/octal format:

```
            012 006 013
   x        007 015 011
            ----------------
            012 006 013   multiply by octal 1 of right-hand multiplier digit
   +                  3   add binary 1 times octal 3 of right-hand digit
          -----------------
            012 006 043
            012 010 003   normalize 043 (and add 2 to 006 = 010 octal)
   +              4       add binary 1 times binary 1 of right-hand digit (=4)
          -----------------
            012 014 003
   +            6         add binary 1 times octal 6 of middle digit
          -----------------
            012 074 003
            015 014 003   normalize 074 (subtract 6 from 7 and add 3 to 012)
   +          0           add binary 1 times binary 0 of middle digit (= 0)
          ------------------
            015 014 003
   +         2            add binary 1 times octal 2 of left-hand digit
           -----------------
        000 035 014 003
        001 015 014 003   normalize 035
   +      4               add binary 1 times binary 1 of left-hand digit (= 4)
       ----------------------
```

```
       005 015 014 003
  +              17       add octal 5 times octal 3 of right-hand digit (=17 octal)
      ----------------------
       005 015 033 003    (octal 4 + 7 = 13)
       005 016 013 003    normalize 033
  +               5       add octal 5 times binary 1 of right-hand digit
      ----------------------
       005 016 063 003
       005 021 003 003    normalize 063 (and 016 + 3 = 021 octal)
  +          36           add octal 5 times octal 6 of middle digit (= 36 octal)
      ----------------------
       005 057 003 003
       007 017 003 003    normalize 057
  +          0            add octal 5 times binary 0 of middle digit
       ----------------------
       007 017 003 003
  +      12               add octal 5 times octal 2 of left-hand digit (= 12 octal)
      -----------------------
       021 017 003 003
   001 001 017 003 003    normalize 021
+        5                add octal 5 times binary 1 of left-hand digit
-----------------------------
   001 051 017 003 003
   003 011 017 003 003    normalize 051
+                 3       add binary 1 times octal 3 of right-hand digit
-----------------------------
   003 011 017 033 003
   003 011 020 013 003    normalize 033
   003 012 000 013 003    normalize 020
+             4           add binary 1 times binary 1 of right-hand digit (= 4)
-----------------------------
   003 012 004 013 003
+            6            add binary 1 times octal 6 of middle digit
-----------------------------
   003 012 064 013 003
   003 015 004 013 003    normalize 064
+         0               add binary 1 times binary 0 of middle digit (= 0)
 ----------------------------
   003 015 004 013 003
+        2                add binary 1 times octal 2 of left-hand digit
 -----------------------------
   003 035 004 013 003
   004 015 004 013 003    normalize 035
+     4                   add binary 1 times binary 1 of left-hand digit (= 4)
-----------------------------
   010 015 004 013 003
+            25           add octal 7 times octal 3 of right-hand digit (= 25 octal)
-----------------------------
  010 015 031 013 003
```

```
       010 016 011 013 003   normalize 031
+                  7         add octal 7 times binary 1 of right-hand digit
      ------------------------------
       010 016 081 013 003
       010 022 001 013 003   normalize 081 (and 16 octal + 4 = 22 octal)
+             52             add octal 7 times octal 6 of middle digit (= 52 octal)
      -------------------------------
       010 074 001 013 003
       013 014 001 013 003   normalize 074
+              0             add octal 7 times binary 0 of middle digit
      -------------------------------
       013 014 001 013 003
+        16                  add octal 7 times octal 2 of left-hand digit (= 16 octal)
      -------------------------------
       031 014 001 013 003   (octal 6 + 3 = 11)
   001 011 014 001 013 003   normalize 031
+         7                  add octal 7 times binary 1 of left-hand digit
     --------------------------------
   001 081 014 001 013 003
   005 001 014 001 013 003   normalize 081, and done!
     5   1   C   1   B   3   the hexadecimal form of the solution
```

Hexadecimal division is also possible, combining the subtraction and multiplication methods shown above, in an obvious manner. This could, however, be rather complicated and prone to error for all but the simplest examples, so may not be as useful as the above techniques.

I worked these examples from right to left rather than the normal abacus method of left to right as it seemed easier to keep my place in the calculation.

Steve Treadwell
October, 2014