3.1 5730

3.2 5730

3.3 0101 1110 1101 0100

The attraction is that each hex digit contains one of 16 different characters (0-9, A-E). Since with 4 binary bits you can represent 16 different patterns, in hex each digit requires exactly 4 binary bits.

And bytes are defined as 8 bits long, so two hex digits are all that are required to represent the contents of 1 byte.

3.4
$$\begin{array}{r} 4365 \\ -\ 3412 \\ \hline 753 \end{array} \rightarrow 753_{(8)}$$

3.5 4365 → -365

-365

-3412

(-3777)

111 111 111 11) → 7777

3.6 185
$\underline{-122}$
63 Neither up overflow nor down overflow.

3.7 185 → 10111001 → -57
"-"

185 + 122 = -57 + 122 = 65
Neither up-overflow nor down-overflow.

3.8 -57 - 122 = -179 < -128
Overflow

3.13 62 X 12

Step	Action	Multiplicand	Product/Multplier
0	Initial Vals	0110 0010	0 0000 0000 0001 0010
1	lsb=0	0110 0010	0 0000 0000 0001 0010
	Rshift	0110 0010	0 0000 0000 0000 1001
2	lsb=1	0110 0010	0 0110 0010 0000 1001
	Rshift	0110 0010	0 0011 0001 0000 0100
3	lsb=0	0110 0010	0 0011 0001 0000 0100
	Rshift	0110 0010	0 0001 1000 1000 0010
4	lsb=0	0110 0010	0 0001 1000 1000 0010
	Rshift	0110 0010	0 0000 1100 0100 0001
5	lsb=1	0110 0010	0 0110 1110 0100 0001
	Rshift	0110 0010	0 0011 0111 0010 0000
6	lsb=0	0110 0010	0 0011 0111 0010 0000
	Rshift	0110 0010	0 0001 1011 1001 0000
7	lsb=0	0110 0010	0 0001 1011 1001 0000
	Rshift	0110 0010	0 0000 1101 1100 1000
8	lsb=0	0110 0010	0 0000 1101 1100 1000
	Rshift	0110 0010	0 0000 0110 1110 0100

3.14

For hardware, it takes 1 cycle to do the add, 1 cycle to do the shift, and 1 cycle to decide if we are done. So the loop takes (3×A) cycles, with each cycle being B time units long.

For a software implementation, it takes 1 cycle to decide what to add, 1 cycle to do the add, 1 cycle to do each shift, and 1 cycle to decide if we are done. So the loop takes (5×A) cycles, with each cycle being B time units long.

$(3\times8)\times 4tu = 96$ time units for hardware.
$(5\times8)\times 4tu = 160$ time units for software.

3.27 $-1.5625\times10^{-1} = -0.15625\times10^{0}$

$= -0.00101\times2^{0}$

$= -1.01\times2^{-3}$

exponent : $-3+15 = 12$

fraction : $-0.0100\ 0000\ 00$

answer : 1 01100 0100 0000 00

$exp \in [-15, 16]$ accuracy : 10 bits

3.29

$2.6125 \times 10^1 + 4.150390625 \times 10^{-1}$

$2.612500000000 \times 10^{-1}$
0.041503906250
2.654003906250

$2.6125 \times 10^1 = 26.125 = 11010.001 = 1.101000|000 \times 2^4$
$4.150390625 \times 10^{-1} = 0.4150390625 = 0.0110\,1010\,0111$
$= 1.1010100111 \times 2^{-2}$

$1.101000|0000000000 \times 2^4$
$0.000001|10|0100111$
$1.1010|0001|0100111$
$= 11010.1000|0100111$
$= 26.5400390625$
$= 2.65400390625 \times 10^1$

3.30

$-8.0546875 \times -1.79931640625 \times 10^{+}$

$\rightarrow -1.0000000111 \times 2^{3}$

$-0.179931640625 \rightarrow -1.0111000010 \times 2^{-3}$

exp: $-3+3=0$ $0+16=16\ (10000)$

$$\begin{array}{r} 1.0000000111 \\ \times\ 1.0111000010 \\ \hline 1.0111001000001001110 \times 2^{0} \end{array}$$

answer: 0 10000 0111001100

$1.0111001100 = 1.44921876$

however: $-8.0546875 \times -1.79931640625$

$= 1.44929313659667968 75$

Some information was lost because the result did not fit into the available 10-bit field. Answer (only) off by 0.00007438659667985