

Quiz Preparation: Converting Values

1. Write the appropriate vocabulary word next to its definition.

a)		a set of 4 binary digits
b)		a set of 8 binary digits
c)		a name for the base-2 number system
d)		a name for the base-16 number system
e)		an operation that inverts (or “flips”) the bits of a binary number ($0 \leftrightarrow 1$)
f)		to change from one form to another, for example from binary to decimal

2. Given the binary number 1110_1001, make the conversions specified below. It is easy to make small mistakes, so show your work for partial marks.

a) Convert the value to decimal, assuming the byte represents an **unsigned number**.

b) Convert the value to decimal, assuming the byte represents a **sign and magnitude**.

c) Convert the value to decimal, assuming the byte represents a **two's complement** number.

d) Convert the value to **hexadecimal**.

Quiz Preparation: Converting Values

3. Again working with the same binary number, 1110_1001, complete the tables below. It is easy to make small mistakes, so show your work for partial marks.

- a) Perform a **left shift** by two bit positions, then convert the result to decimal, assuming the byte represents an **unsigned number**.

- b) A **left shift** by two bit positions should be equivalent to multiplication by four ($\times 4$); however the value calculated in part (3a) is obviously not even close to four times the value calculated in part (2a). What word describes what happened.

- c) Perform a **logical right shift** by two bit positions, then convert the result to decimal, assuming the byte represents an **unsigned number**.

- d) Perform a **logical right shift** by two bit positions, as was done in part (c), then convert the result to **hexadecimal**.

- e) What mathematical operation (operator and operand) is equivalent to a **logical right shift** by two bit positions, assuming the byte represents an **unsigned number**.

operand (binary)	operator	operand (decimal)
1110_1001		

Note: the reason this mathematical expression isn't exactly true is due to the **loss of precision** as bits are shifted out of the number, and lost.

- f) Perform an **arithmetic right shift** by two bit positions, then convert the result to decimal, assuming the byte represents a **two's complement** number.