Problem 2.82 Solution:

This problem helps students think about fractional binary representations.

A. Letting V denote the value of the string, we can see that shifting the binary point k positions to the right gives a string $y.yyyyyy\cdots$, which has numeric value $Y+V$, and also value $V \times 2^k$. Equating these gives $V = \frac{Y}{2^k-1}$.

B. (a) For $y = 001$, we have $Y = 1$, $k = 3$, $V = \frac{1}{7}$.
   (b) For $y = 1001$, we have $Y = 9$, $k = 4$, $V = \frac{9}{15} = \frac{3}{5}$.
   (c) For $y = 000111$, we have $Y = 7$, $k = 6$, $V = \frac{7}{63} = \frac{1}{9}$.

Problem 2.85 Solution:

This exercise is of practical value, since Intel-compatible processors perform all of their arithmetic in extended precision. It is interesting to see how adding a few more bits to the exponent greatly increases the range of values that can be represented.

Description	Extended precision	
	Value	Decimal
Smallest pos. denorm.	$2^{-63} \times 2^{-16382}$	3.65×10^{-4951}
Smallest pos. norm.	2^{-16382}	3.36×10^{-4932}
Largest norm.	$(2-\epsilon) \times 2^{16383}$	1.19×10^{4932}

Problem 2.87 Solution:

This problem tests a lot of concepts about floating-point representations, including the encoding of normalized and denormalized values, as well as rounding.

Format A		Format B		Comments
Bits	Value	Bits	Value	
`1 01110 001`	$\frac{-9}{16}$	`1 0110 0010`	$\frac{-9}{16}$	
`0 10110 101`	208	`0 1110 1010`	208	
`1 00111 110`	$\frac{-7}{1024}$	`1 0000 0111`	$\frac{-7}{1024}$	Norm $\rightarrow$ denorm
`0 00000 101`	$\frac{5}{131072}$	`0 0000 0001`	$\frac{1}{1024}$	Smallest positive denorm
`1 11011 000`	-4096	`1 1110 1111`	-248	Smallest number $> -\infty$
`0 11000 100`	768	`0 1111 0000`	$+\infty$	Round to ∞.