1.2 Decimals and Fractions

In this section we will continue our exploration of numbers in binary format. The specific task in this section is to determine how to represents decimals and fractions in binary and hexadecimal format, a skill we need to master before learning how to store real numbers in floating point format on computers.

Decimals in Binary

Recall that the decimal 0.1485 can be expanded in powers of 10 as follows.

$$0.1485 = 1 \cdot 10^{-1} + 4 \cdot 10^{-2} + 8 \cdot 10^{-3} + 5 \cdot 10^{-4}$$

In similar fashion, we can expand a binary decimal in powers of 2.

$$\begin{aligned}(0.1101)_2 &= 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} \\ &= \frac{1}{2} + \frac{1}{4} + \frac{0}{8} + \frac{1}{16} \\ &= \frac{13}{16}\end{aligned}$$

In base ten, multiplying by 10 will move the decimal one place to the right.

$$\begin{aligned}0.1485 \cdot 10 &= (1 \cdot 10^{-1} + 4 \cdot 10^{-2} + 8 \cdot 10^{-3} + 5 \cdot 10^{-4}) \cdot 10 \\ &= 1 \cdot 10^{0} + 4 \cdot 10^{-1} + 8 \cdot 10^{-2} + 5 \cdot 10^{-3} \\ &= 1.485\end{aligned}$$

This fact gives us an algorithm for determining the coefficients of a decimal expansion in powers of 10.

In **Table 1.1**, begin by placing 0.1485 in the third column of the first row. Multiply by 10 and place the result in the second column of the second row. Strip the digit to the left of the decimal point in the result and place it in the fourth column of the second row. Place the remainder in the third column of the second row. Iterate until no more digits exist to the right of the decimal point.

10		0.1485	
10	1.485	0.485	1
10	4.85	0.85	4
10	8.5	0.5	8
10	5.	0.	5

Table 1.1. Repeated multiplication by 10 shifts the coefficients to the left of the decimal point, one-by-one.

[1] Copyrighted material. See: http://msenux.redwoods.edu/Math4Textbook/

Finally, the digits in the fourth column, read from top to bottom, will be the coefficients in the expansion in powers of 10.

This process might not terminate. For example, we follow the same procedure to find the coefficients for the decimal expansion of 1/7 in powers of ten in **Table 1.2**.

10		$1/7$	
10	$10/7 = 1 + 3/7$	$3/7$	1
10	$30/7 = 4 + 2/7$	$2/7$	4
10	$20/7 = 2 + 6/7$	$6/7$	2
10	$60/7 = 8 + 4/7$	$4/7$	8
10	$40/7 = 5 + 5/7$	$5/7$	5
10	$50/7 = 7 + 1/7$	$1/7$	7
10	$10/7 = 1 + 3/7$	$3/7$	1

Table 1.2. Sometimes the process produces a repeating decimal.

Note that in the last row of **Table 1.2**, we have entries identical to the second row. Hence, the pattern $1/7 = 0.142857142857\ldots$ repeats indefinitely.

In base two, multiplying by 2 will move the decimal one place to the right.

$$\begin{aligned}(0.1101)_2 \cdot 2 &= (1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4}) \cdot 2 \\ &= 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ &= (1.101)_2\end{aligned}$$

This means that the algorithm demonstrated in **Tables 1.1** and **1.2** will work equally well for base two. We simply multiply repeatedly by two instead of ten.

▶ **Example 1.** *Find the binary representation of the base ten decimal* 0.875.

Place 0.875 in the third column of row one in **Table 1.3**. Multiply by 2 and place the result in the second column of row two. Strip off the digit to the left of the decimal point of this result and place it in the fourth column of row two. Place the remainder in the third column of row two. Iterate.

Note that further multiplication by 2 will only produce zeros, which we can ignore. Hence, $0.875 = (0.111)_2$.

The result is easily checked by expanding in powers of two.

2		0.875	
2	1.75	0.75	1
2	1.5	0.5	1
2	1.	0.	1

Table 1.3. Repeated multiplication by 2 shifts the coefficients to the left of the decimal point, one-by-one.

$$\begin{aligned}(0.111)_2 &= 1\cdot 2^{-1} + 1\cdot 2^{-2} + 1\cdot 2^{-3}\\ &= \frac{1}{2}+\frac{1}{4}+\frac{1}{8}\\ &= \frac{7}{8}\end{aligned}$$

If you divide 7 by 8 (or repeat the base ten algorithm of **Table 1.1**), you will see that $7/8 = 0.875$. Of course, you could also check in Matlab.

```
>> 7/8
ans =
    0.8750
```

As with base ten, some decimals have binary expansions that will repeat.

▶ **Example 2.** *Find a binary expansion for the decimal* 0.1.

In **Table 1.4**. After repeatedly multiplying by 2, a pattern emerges.

2		0.1	
2	0.2	0.2	0
2	0.4	0.4	0
2	0.8	0.8	0
2	1.6	0.6	1
2	1.2	0.2	1
2	0.4	0.4	0
2	0.8	0.8	0
2	1.6	0.6	1

Table 1.4. Sometimes the process produces a repeating decimal.

Thus,

$$0.1 = (0.00011001100110011001\ldots)_2.$$

Note further that because of its infinite repeating nature, it is not possible to store the binary number $(0.0001100110011001\ldots)_2$ **exactly** in a computer that uses finite base two storage.

Decimals in Hex

We can expand a hexadecimal in powers of 16. Recall that working with base 16, $A = 10$, $B = 11$, $C = 12$, $D = 13$, $E = 14$, and $F = 15$. So, for example,

$$\begin{aligned}(0.2A8F)_{16} &= 2 \cdot 16^{-1} + A \cdot 16^{-2} + 8 \cdot 16^{-3} + F \cdot 16^{-4} \\ &= \frac{2}{16} + \frac{10}{16^2} + \frac{8}{16^3} + \frac{15}{16^4} \\ &= \frac{2 \cdot 16^3 + 10 \cdot 16^2 + 8 \cdot 16 + 15}{16^4} \\ &= \frac{10895}{65536}.\end{aligned}$$

In the next few examples, we convert binary decimals into hexadecimals.

▶ **Example 3.** *Convert the binary decimal* $(0.111011110011)_2$ *into hexadecimal format.*

We can expand the binary decimal $(0.111011110011)_2$ in powers of 2.

$$\begin{aligned}&1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4} \\ &\quad + 1 \cdot 2^{-5} + 1 \cdot 2^{-6} + 1 \cdot 2^{-7} + 1 \cdot 2^{-8} \\ &\quad + 0 \cdot 2^{-9} + 0 \cdot 2^{-10} + 1 \cdot 2^{-11} + 1 \cdot 2^{-12}\end{aligned}$$

This can be rewritten as follows.

$$\begin{aligned}&(1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0) \cdot 2^{-4} \\ &\quad + (1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0) \cdot 2^{-8} \\ &\quad + (0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0) \cdot 2^{-12}\end{aligned}$$

This is equivalent to the following expression.

$$13 \cdot (2^{-4})^1 + 15 \cdot (2^{-4})^2 + 3 \cdot (2^{-4})^3$$

However, $2^{-4} = (2^4)^{-1} = 16^{-1}$, so this last expression can be written as

$$13 \cdot (16^{-1})^1 + 15 \cdot (16^{-1})^2 + 3 \cdot (16^{-1})^3,$$

which is equivalent to

$$13 \cdot 16^{-1} + 15 \cdot 16^{-2} + 3 \cdot 16^{-3}.$$

Finally, $D = 13$ and $F = 15$ in hex, so

$$(0.111011110011)_2 = (DF3)_{16}.$$

In pracice, we don't have to work so hard. We can simply group the binary digits in groups of four, then use the mappings in **Table 1.5**.

Binary	Hex	Binary	Hex
0	0	1000	8
01	1	1001	9
10	2	1010	A
11	3	1011	B
100	4	1100	C
101	5	1101	D
110	6	1110	E
111	7	1111	F

Table 1.5. Binary-Hex conversions.

Thus,

$$(0.111011110011)_2 = (0.1110 - 1111 - 0011)_2 = (DF3)_{16}.$$

Let's look at another example.

▶ **Example 4.** *In* **Example 2**, *we found a binary decimal for the base ten decimal 0.1. Convert that expansion to hexadecimal.*

In **Example 2**, we found

$$0.1 = (0.00011001100110011001\ldots)_2.$$

We can group the binary digits in groups of four,

$$(0.0001 - 1001 - 1001 - 1001 - 1001 - \ldots)_2,$$

then use the conversions from **Table 1.5** to make the change to hexadecimal.

$$(0.19999\cdots)_{16}$$

1.2 Exercises

For each of the base ten decimals in **Exercises 1-4**, use hand calculations to find an equivalent binary decimal. Check your result by expanding in powers of two and simplifying, then use Matlab to check that your fractional result is equivalent to the original decimal.

1. 0.3125

2. 0.53125

3. 0.6171875

4. 0.232421875

In **Exercises 5-8**, use hand calculations find a repeating binary expansion for the given base ten decimal.

5. 0.3

6. 0.9

7. 0.05

8. 0.15

In **Exercises 9-14**, use hand calculations to place the given binary decimal in hexadecimal format.

9. $(0.10110111)_2$

10. $(0.111110110011)_2$

11. $(0.0111011111101101)_2$

12. $(0.0111011111001111)_2$

13. $(0.0111011101110111\ldots)_2$

14. $(0.0011011000000011000000110\ldots)_2$

In **Exercises 15-18**, use hand calculations to place the given base ten decimal in hexadecimal format.

15. 0.259765625

16. 0.88671875

17. 0.8

18. 1/3

1.2 Answers

1. $(0.0101)_2$

3. $(0.1001111)_2$

5. $(0.0100110011001\ldots)_2$

7. $(0.0000110011001100\ldots)_2$

9. $(0.B7)_{16}$

11. $(0.77ED)_{16}$

12. $(0.7777\ldots)_{16}$

14. $(0.428)_{16}$

16. $(0.110011001100\ldots)_{16}$