The Binary Number System

Robert B. Heckendorn
University of Idaho

January 17, 2022

Numbers are said to be represented by a **place value system**. Where the value of a symbol depends on where it is... its place. For instance an 8 in right most position in a number means 8 but in the third place from the right means 800. That is, each position from the right side of the number has an associated value. Each position to the left is worth 10 more than the next position to the right. For example: in the number 7654, the rightmost place is worth 1, the next to the right most position is worth 10, next position is worth 100 and the left most is worth 1000. The number 7654 is therefore: $7 * 1000 + 6 * 100 + 5 * 10 + 4 * 1$ which is 7654 in decimal.

Base 10 or decimal numbers

The base 10 number system, numbers are represented by a list of symbols of which there are 10 kinds of symblos: $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$. The position or place values are powers of 10: $10^0, 10^1, 10^2, 10^3, ...$ or $1, 10, 100, 1000, ...$

Long ago computing machines used mechanical mechanisms and decimal represention but now with fast electricity based computing devices a new way to encode numbers and data in general is needed.

1 Binary

In **binary**, the base of the number system is 2.

Base 2 or binary numbers

Each position to the left is worth 2 more than the next position to the right. For example: in the number 1101, the rightmost position is worth 1, the next to the right most position is worth 2, next position is worth 4 and the left most is worth 8. The number 1101 is therefore: $1 * 8 + 1 * 4 + 0 * 2 + 1 * 1$ which is 13 in decimal. In short converting from binary to decimal is as easy as just adding up the binary digits times their place values.

Converting binary to decimal

The base 2 number system, numbers are represented by a list of symbols of which there are 2 kinds: $0, 1$. The position values are powers of 2: $2^0, 2^1, 2^2, 2^3, ...$ or $1, 2, 4, 8, 16, ...$

Counting from 0 to 10 in binary is: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010. See how the numbers from 0 to 7 can be represented by 3 bits and at 8 you have to go to 4 bits.

Counting in binary

Because any nonnegative integer can be represented in binary, those numbers can be represented as a string of 1's/0's, electricity on/electricity off, north magnetic field/south magnetic field, etc. So this is how numbers are represented inside modern

Why binary is useful.

computers which use electric components and magnetic fields. Music on CDs is stored the same way, as 1's/0's on the CD surface, etc.

Converting decimal to binary is not as easy. Let's look at a 4 bit example. Here is how to convert a number between 0 and 15 inclusive into a 4 bit binary number:

Converting decimal to binary

```
is it >=8?
      if yes write 1 and subtract 8
      if no write 0

is it >=4?
      if yes write 1 and subtract 4
      if no write 0

is it >=2?
      if yes write 1 and subtract 2
      if no write 0

is it >=1?
      if yes write 1 and subtract 1
      if no write 0
```

Pretty easy, eh? How would you extend this to 5 bits? Hint: the 5^{th} place in a binary number is worth 16.

It is important to see that it is not completely trivial to convert decimal to binary. It requires knowing the powers of 2, asking a question for each digit and subtracting off the power of two if the answer is yes. It requires answering a yes/no question for every power of 2 up to the size of the number you want to convert. Let's run through the above algorithm with the decimal number 13:

Example conversion of decimal to binary.

```
is 13 >= 8?
      yes: write 1 and subtract off 8 so we look at the number 5.

is 5 >= 4?
      yes: write 1 and subtract off 4 so we look at the number 1.

is 1 >= 2?
      no: write 0

is 1 >= 1?
      yes: write 1 and subtract off 1 so we look at the number 0.
```

So the number 13 in decimal is 1101 in binary. What is the decimal number 23 in binary[1]? Hint: you have to add a test for 16 to the above approach.

Each **binary digit**, or **bit** for short, represents the quantity of information that can be determined by answering a yes or no question. You can see this in the 4 bit conversion routine above. Four questions are asked. Bits are the fundamental unit

The quantity of information is measured in bits.

[1] The answer is 10111

of information!

A **byte** is 8 bits and is enough to contain a simple encoding of a character. **ASCII** is one such encoding standard.

It takes about $3\frac{1}{3}$ bits to represent each decimal digit. That means a 10 bit number is about 3 decimal digits. The number $1,000,000$ in decimal is about 20 bits long! In fact: $1,000,000$ in decimal is 11110100001001000000 in binary. That is a lot of writing to express a number.

Binary is not a very compact representation for a number.

2 Octal and Hex

Computer scientists use two shortcut bases to make writing binary easier. The first is base 8 or **octal**. The second is base 16 or **hexadecimal**. The reason they use these is it is insanely easy to convert from binary to octal and back! Same for hexadecimal.

Octal and hexadecimal are compact and easy ways to write binary.

Let's do octal to binary and back. What makes octal so easy is, unlike a decimal digit, there are **exactly** 3 binary bits in each octal digit! So for every octal digit I can translate that into 3 bits. This is because $2^3 = 8$. For example: the octal number: 3705 is 4 octal digits. This should become 12 binary digits: 3 is 011 in binary, 7 is 111 in binary, 0 is 000 in binary, 5 is 101 in binary. So $3705_8 = 011111000101_2$. Note the use of subscripts to denote the base of the number. Yes, it is that straight forward. If you know the 3 bit values for each of the 8 octal digits you are practically done.

Octal to binary conversion.

What is 11110100001001000000_2 in octal? First divide the number in groups of 3 starting on the right: 11 110 100 001 001 000 000. Then simply read off the octal digits: 3641100_8.

Example conversion of binary to octal.

Hexadecimal numbers are base 16. Hexadecimal is sometimes simply referred to as **hex**. Each position to the left is worth 16 more than the next position to the right. The base 16 number system, numbers are represented by a list of symbols of which there are 16 kinds: $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, a, b, c, d, e, f. The position values are powers of 16: $16^0, 16^1, 16^2, 16^3, ...$ or $1, 16, 256, 4096, ...$

Hexadecimal numbers.

For example: in the hex number 1fab, the rightmost position is worth 1, the next to the right most position is worth 16, next position is worth 256 and the left most is worth 4096. The number 1fab is therefore: $1 * 4096 + 15 * 256 + 10 * 16 + 11 * 1$ which is 8107 in decimal. But it is super easy to convert to binary one hex digit at a time because like octal each hex digit is worth **exactly** 4 bits: $1\text{fab}_{16} = 0001\ 1111\ 1010\ 1011_2 = 0001111110101011_2$

Compare the conversion of hexadecimal to decimal to the conversion of hexadecimal to binary.

Table 1: A table of counting in different bases.

Hexadecimal	Decimal	Octal	Binary	6-bit Binary
0	0	0	0	000000
1	1	1	1	000001
2	2	2	10	000010
3	3	3	11	000011
4	4	4	100	000100
5	5	5	101	000101
6	6	6	110	000110
7	7	7	111	000111
8	8	10	1000	001000
9	9	11	1001	001001
a	10	12	1010	001010
b	11	13	1011	001011
c	12	14	1100	001100
d	13	15	1101	001101
e	14	16	1110	001110
f	15	17	1111	001111
10	16	20	10000	010000
11	17	21	10001	010001
12	18	22	10010	010010
13	19	23	10011	010011
14	20	24	10100	010100
15	21	25	10101	010101
16	22	26	10110	010110
17	23	27	10111	010111
18	24	30	11000	011000
19	25	31	11001	011001
1a	26	32	11010	011010
1b	27	33	11011	011011
1c	28	34	11100	011100
1d	29	35	11101	011101
1e	30	36	11110	011110
1f	31	37	11111	011111

The key ideas I want you to know:

The key ideas.

- Current day computers use electricity and so the binary number system has become a convenient way to represent information.
- As an example, binary is a great way to represent integers.
- If a Computer Scientist needs to talk about the detail of what the representation of a number they may use binary, octal, or hexadecimal.
- The **bit** is the fundamental unit to measure the quantity of information. It represents the answer to a yes or no question.

3 A Binary Card Trick

Below are the cards for the **Binary Card Trick**. Cut them out. Ask person to think of a number between 1 and 31 inclusive but don't tell you the number. Now hand the cards to the person and tell them to select all the cards that have their secret number on them. When they hand you the selected cards simply add the numbers in the upper left corner and that is the number they are thinking of.

Binary Card Trick.

Why does this work? On the card with a 1 in the upper left are all the numbers that have a 1 in right most place in the binary representation of the number. That is the 1's place. The card with a 2 in the upper left are all the numbers that have a 1 in next to last position in the binary representation of the number. That is the 2's place. The 4 card is for the 4's place, etc. By handing you that card they are answering a simple yes/no question and giving you one bits worth of information. The question for the card with a 1 in the upper left is "What is the right most digit in the binary representation of the number". When they had you all the selected cards you have the binary for the number and you simply add the place value of each of the 1 bits. That place value is the number in the upper left. That is because that is always a number that looks like a 1 followed by some number of 0's. Note also that it must be the case that each secret number will cause a different set of cards to be chosen? Why? Because each number has a unique binary number. What would the cards look like if the secret number was from 1 to 63 inclusive?

How does the trick relate to binary numbers and bits?

1	**3**	**5**	**7**
9	**11**	**13**	**15**
17	**19**	**21**	**23**
25	**27**	**29**	**31**

2	**3**	**6**	**7**
10	**11**	**14**	**15**
18	**19**	**22**	**23**
26	**27**	**30**	**31**

4	**5**	**6**	**7**
12	**13**	**14**	**15**
20	**21**	**22**	**23**
28	**29**	**30**	**31**

8	**9**	**10**	**11**
12	**13**	**14**	**15**
24	**25**	**26**	**27**
28	**29**	**30**	**31**

16	**17**	**18**	**19**
20	**21**	**22**	**23**
24	**25**	**26**	**27**
28	**29**	**30**	**31**

4 Converting decimal directly to octal and hexadecimal

Converting to octal is based on powers of 8: 1, 8, 64, 512, 4096, 32768, ... Here is an algorithm for the first 4 octal digits.

```
is it >=512?
      if yes write integer portion of number/512 and subtract that many 512's
      if no write 0

is it >=64?
      if yes write integer portion of number/64 and subtract that many 64's
      if no write 0

is it >=8?
      if yes write integer portion of number/8 and subtract that many 8's
      if no write 0

is it >=1?
      if yes write integer portion of number/1 and subtract that many 1's
      if no write 0
```

For example: What is 666 in octal?

- 512 goes into 666 1 time with a remainder of 154 so the first octal digit is 1.
- 64 goes into 154 2 times with a remainder of 26 so the second octal digit is 2.
- 8 goes into 26 3 times with a remainder of 2 so the third octal digit is 3.
- 1 goes into 2 2 times with a remainder of 0 so the third octal digit is 2.
  the answer is 1232_8.

Similarly the algorithm for converting a number into as many as 4 hex digits is:

```
is it >=4096?
      if yes write integer portion of number/4096 and subtract that many 4096's
      if no write 0

is it >=256?
      if yes write integer portion of number/256 and subtract that many 256's
      if no write 0

is it >=16?
      if yes write integer portion of number/16 and subtract that many 16's
      if no write 0
```

```
is it >=1?
      if yes write integer portion of number/1 and subtract that many 1's
      if no write 0
```

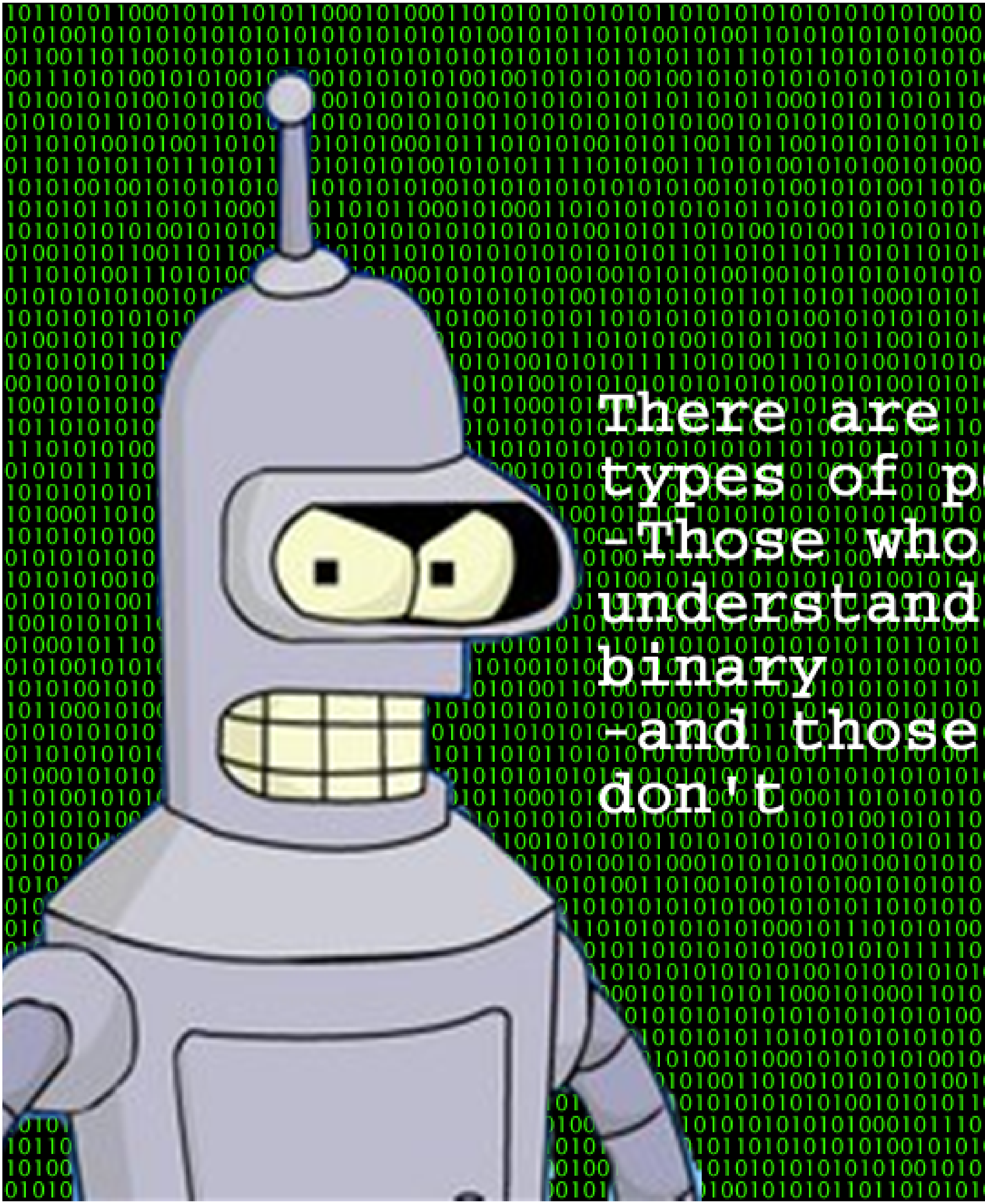
What is 43785 in hex?

- 4096 goes into 43785 10 times with a remainder of 2825 so the first hex digit is 10. In hex this is represented as an a. See the table for counting in different bases.
- 256 goes into 2825 11 times with a remainder of 9 so the second hex digit is 11 or b.
- 16 goes into 9 0 times with a remainder of 9 so the third hex digit is 0.
- 1 goes into 9 9 times with a remainder of 0 so the third hex digit is 9.

The answer is $ab09_{16}$. Notice that because base 16 is larger that 10 the number of digits to represent a number is less than or equal to the number of digits to represent a number in base 10.

5 Futurama and Binary

It turns out the writers of the TV show Futurama have degrees in math and put some math/computer humor into the show. Here are some examples can you explain why these numbers where chosen?


There are
types of
-Those who
understand
binary
-and those
don't