CS1800
Fall 2026
Representation of Numbers

Intro & Opening Joke
Binary Numbers (Base Two)
    Convert from binary to decimal
    Convert from decimal to binary -- division method
Converting to and from Other Bases (General Method)
    Converting with Bases that are Powers of 2

Intro & Opening Joke

Binary numbers (aka “base two”) can represent any number using two digits. Computer scientists also care about other bases, including octal numbers (base eight) and hexadecimal numbers (base 16). Octal and hex are used as abbreviations for binary numbers, because we can specify bytes easily; a computer’s memory address is usually 32 or 64 bits and easily represented with base-8 or base-16 numbers.

> *Q: Why do computer scientists celebrate Christmas on Halloween?*
>
> *A: Because Oct-31 is the same as Dec-25!!!!!!*

In this handout, our numbers will be *unsigned*, that is, we will not indicate whether the numbers are positive or negative. We can have signed numbers as well, and those are covered in the two’s complement handout.

Binary Numbers (Base Two)

Base two is the counting system that computers understand. Why? Because computers have microchips, and microchips have transistors. A transistor can have electricity flowing through it, or not -- so they can understand on/off, yes/no, 1/0.

The only possible digits in base 2 are 0 and 1. Let’s assume numbers are *unsigned* for now -- no need to represent a positive or a negative sign. A digit’s position is linked to its value, all in powers of two.

Convert from binary to decimal

Consider the binary number 10010110_2.

Digit	1	0	0	1	0	1	1	0
Value	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

We can translate this number into base-10 by multiplying each digit by its value and summing the result:

$$(1 \cdot 2^7) + (0 \cdot 2^6) + (0 \cdot 2^5) + (1 \cdot 2^4) + (0 \cdot 2^3) + (1 \cdot 2^2) + (1 \cdot 2^1) + (0 \cdot 2^0)$$

We computer scientists come to know the powers of two pretty well; if you can't rattle them off from the top of your head yet, you will! Let's finish off the math:

$$(1 \cdot 128) + (0 \cdot 64) + (0 \cdot 32) + (1 \cdot 16) + (0 \cdot 8) + (1 \cdot 4) + (1 \cdot 2) + (0 \cdot 1)$$
$$= (1 \cdot 128) + (1 \cdot 16) + (1 \cdot 4) + (1 \cdot 2)$$
$$= 128 + 16 + 4 + 2$$
$$= 150$$

Now we can say that $10010110_2 = 150_{10}$.

Convert from decimal to binary -- division method

Now let's go the other way, and start with 150_{10}, with the goal of representing the same value in base two. We can convert from decimal to base 2 by continuously dividing by the base and tracking the remainders. We keep on dividing until we see a quotient of zero, and then the remainders are our answer.

$$150 \div 2 = 75\, r\, 0$$
$$75 \div 2 = 37\, r\, 1$$
$$37 \div 2 = 18\, r\, 1$$
$$18 \div 2 = 9\, r\, 0$$
$$9 \div 2 = 4\, r\, 1$$
$$4 \div 2 = 2\, r\, 0$$
$$2 \div 2 = 1\, r\, 0$$
$$1 \div 2 = 0\, r\, 1$$

Reading from bottom to the top, we get 101010110as the unsigned binary for 150_{10}. Which we already know is correct -- so woo hoo, it worked!

Converting to and from Other Bases (General Method)

The two methods we described above are generalized; they can work to convert from decimal to any other base, and from any other base to decimal. Here's a quick example in base 7 so we can see it in action.

Convert from base 7 to decimal

Consider the number 3521_7. Convert it to decimal using the same approach as above:

Digit	3	5	2	1
Value	7^3	7^2	7^1	7^0

And so we get $3521_7 = (3 \cdot 7^3) + (5 \cdot 7^2) + (2 \cdot 7^1) + (1 \cdot 7^0)$

$= 1029 + 245 + 14 + 1$

$= 1289_{10}$

Convert from decimal to base 7

Consider the number $= 1289_{10}$. Let's convert back to base 7 using the division method:

$1289 \div 7 = 184\, r\, 1$

$184 \div 7 = 26\, r\, 2$

$26 \div 7 = 3\, r\, 5$

$3 \div 7 = 0\, r\, 3$

Reading from bottom to the top, we get 3521 as the unsigned base-7 for 1289_{10}. Which we already know is correct -- so woo hoo, it worked!

Converting with Bases that are Powers of 2

The conversion methods we've seen so far are general and will work no matter what base we're talking about. But we can also take fun shortcuts converting from base 2 to base 16 or base 8. The bases are all powers of two, making our life easier and more fun!

Hexadecimal (Base 16, a.k.a, Base 2^4) Lookup Table

Decimal	**Hex**	**4-bit Binary**	**Decimal**	**Hex**	**4-bit Binary**
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	10	A	1010
3	3	0011	11	B	1011
4	4	0100	12	C	1100

5	5	0101	13	D	1101
6	6	0110	14	E	1110
7	7	0111	15	F	1111

Now, if we want to convert from base 16 to base 2, or vice versa, all we need to do is use the lookup table!

Convert from base 16 to base 2

Let's say we want to convert $A4F3_{16}$ to binary. All we need is to see that:

- $A = 1010$
- $4 = 0100$
- $F = 1111$
- $3 = 0011$

We string them together and boom --- base 2! $A4F3_{16} = 1010010011110011_2$