1.2 Unsigned binary numbers

Counting in binary

Humans have ten fingers so humans use a base ten number system. Ex: 452 means $4\times10^2 + 5\times10^1 + 2\times10^0$. Digital systems have two-valued signals (high, low) so digital systems use a base two number system. Ex: 1101 means $1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$. A number in base ten is called a ***decimal*** number (from Latin "decem" meaning ten), while a number in base two is called a ***binary*** number (from Latin "bini" meaning two together).

Base ten has ten symbols for a digit: 0, 1, ..., 9. When counting up and reaching 9, the digit resets to 0 and a 1 carries to the next digit. Ex: 008, 009, 010, 011, or 098, 099, 100, 101. Base two has only two symbols for a digit: 0 and 1. So counting up results in frequent carries. Ex: 000, 001, 010, 011, 100, 101, 110, 111. Each digit in a binary number is called a ***bit***, short for "binary digit".

PARTICIPATION ACTIVITY 1.2.1: Counting up in decimal and in binary.

1 2 3 4 5 6 2x speed

-	0		0	
*	1		1	
* *	2		10	Reset, carry
* * *	3		11	
* * * *	4		100	Reset, carry and reset, carry
* * * * *	5		101	
* * * * * *	6		110	Reset, carry
* * * * * * *	7		111	
* * * * * * * *	8		1000	Reset, carry and reset, carry and re
* * * * * * * * *	9		1001	
* * * * * * * * * *	10	Reset digit to 0, carry 1	1010	Reset, carry
* * * * * * * * * * *	11		1011	
	...			
	98			
	99			
	100	Reset digit to 0, carry 1 to next digit which is 9 so reset to 0, carry 1 to next digit		

Resets and carries happen on every other count.

Captions

1. Various quantities.
2. In base ten, the first digit goes up to 9, then the digit is reset to 0 and a 1 is carried to the next digit.
3. If the carry is to a digit that is already 9, then that digit is also reset to 0 and a 1 is carried to the next digit.
4. In base two, the first digit goes up to 1, then the digit is reset to 0 and a 1 is carried to the next digit.
5. If the carry is to a digit that is already 1, then that digit is also reset to 0 and a 1 is carried to the next digit.
6. Resets and carries happen on every other count.

Feedback?

Note: This section only covers unsigned binary numbers. An ***unsigned binary*** number can only represent non-negative values, such as a 4-bit binary number being 0000 (0), 0001 (1), 0010 (2), ..., 1111 (15). In contrast, a signed binary number uses the leftmost bit to represent whether a number is positive or negative.

PARTICIPATION ACTIVITY 1.2.2: Counting up in binary.

Type the next higher 3-digit binary number.

1) 000

001

Check Show answer

Correct

001

Counting up always starts by trying to add 1 to the rightmost digit.

2) 001

010

Check Show answer

Correct

010

1 can't be added to the rightmost digit due to being at the max of 1, so that digit is reset to 0, and a 1 carried to the next digit.

3) 010

011

Correct

011

Check Show answer

Counting up always starts by trying to add 1 to the rightmost digit, from 0 to 1. The rightmost digit is currently 0, so becomes 1.

4) 011

100

Check Show answer

Correct

100

1 can't be added to the rightmost bit, so that bit is reset to 0 and a carry generated to the second bit.
1 can't be added to the second bit, so that bit is reset to 0 and a carry generated to the third (leftmost) bit.

5) 111 (type a 4-bit answer)

1000

Check Show answer

Correct

1000

The rightmost bit is reset to 0 and a carry generated for the next bit. That bit is also reset to 0, and a carry generated for the next bit. That bit is also reset to 0 and a carry generated to the next bit. That bit isn't shown initially but is assumed 0, so becomes 1. (Akin to 99 incrementing to 100).

Feedback?

CHALLENGE ACTIVITY 1.2.1: Counting up with 3 bits.

Can you count from 000 to 111 in binary in 20 seconds?

547404.4091098.qx3zqy7

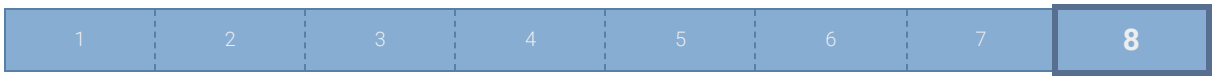
Jump to level 1

Decimal	Binary (3 bits)
0	000
1	001
2	010
3	011
4	100
5	101

1 2 3 4 5 6 7

6	110
7	111

8



Check Next **Done**. Click any level to practice more. Completion is preserved.

Correct.

Feedback?

Converting from binary to decimal

Software and hardware developers benefit from being able to quickly convert between binary and decimal numbers.

Given a binary number, each digit's weight is summed to form a decimal number. Ex: 1101 = $1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 = 8 + 4 + 0 + 1 = 13$.

PARTICIPATION ACTIVITY 1.2.3: Binary to decimal tool.

Reset Each value ranges 0 to 1

1	1	1	1	1	1	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

$1\cdot128 + 1\cdot64 + 1\cdot32 + 1\cdot16 + 1\cdot8 + 1\cdot4 + 1\cdot2 + 1\cdot1 = 255$ (decimal value)

Feedback?

PARTICIPATION ACTIVITY 1.2.4: Converting from binary to decimal.

Convert from binary to decimal. Use the fewest decimal digits possible. Recall that four-bit binary digit weights are 8, 4, 2, and 1.

1) 0001

1

Correct

Check Show answer

1

1 × 1 = 1

2) 0010

2

Check Show answer

Correct

2

1 × 2 = 2

3) 0111

7

Check Show answer

Correct

7

4 + 2 + 1 = 7 (omits the "1 ×" for simplicity)

4) 1001

9

Check Show answer

Correct

9

8 + 1 = 9

5) 1111

15

Check Show answer

Correct

15

8 + 4 + 2 + 1 = 15

Feedback?

Converting from decimal to binary

Given a decimal number, starting from the leftmost binary digit (greater than the decimal number), a 1 is placed in each digit as long as the resulting binary number doesn't exceed the decimal number.

PARTICIPATION ACTIVITY 1.2.5: Converting from decimal to four-bit binary.

Type a four-bit answer: 0101, not 101. Four-bit binary digit weights: 8, 4, 2, 1.

1) 3

0011

Correct

0011

Check Show answer

Starting from left:

```
1 _ _ _   8 is too much
0 1 _ _   4 is too much
0 0 1 _   2 is less
0 0 1 1   2 + 1 = 3
```

2) 4

0100

Check Show answer

Correct

0100

Starting from left:

```
1 _ _ _   8 is too much
0 1 0 0   4 is equal
```

3) 5

0101

Check Show answer

Correct

0101

Starting from left:

```
1 _ _ _   8 is too much
0 1 _ _   4 is less
0 1 1 _   6 is too much
0 1 0 1   4 + 1 = 5 is equal
```

4) 13

1101

Check Show answer

Correct

1101

Starting from left:

```
1 _ _ _   8 is less
1 1 _ _   8 + 4 = 12 is less
1 1 1 _   8 + 4 + 2 = 14 is too much
1 1 0 1   8 + 4 + 1 = 13 is equal
```

Feedback?

PARTICIPATION ACTIVITY 1.2.6: Binary-to-decimal converter.

Binary		Decimal
11	⇄	3
Binary numbers have 0s and 1s		Value ranges from 0 to 65535

Give me credit

Feedback?

Bits required to represent a number

A computer has a limited amount of storage available to represent numbers as bits. The number of bits available to store a number defines the range of numbers that can be represented. Ex: If 2 bits are available, then the decimal numbers 0 (00_2), 1 (01_2), 2 (10_2), and 3 (11_2) can be represented, and the range is 0-3.

PARTICIPATION ACTIVITY 1.2.7: Number of bits required to represent a decimal number.

Determine the minimum number of bits required to represent each decimal number.

1) 1

1

Check Show answer

Correct

1

Since $1 = 1_2$, 1 bit is required to represent 1.

2) 5

3

Check Show answer

Correct

3

Since $5 = 101_2$, 3 bits are required to represent 5.

3) 15

4

Check Show answer

Correct

4

Since $15 = 1111_2$, 4 bits are required to represent 15.

4) 16

5

Check Show answer

Correct

5

Since $16 = 10000_2$, 5 bits are required to represent 16.

Feedback?

Adding binary numbers

For decimal numbers, adding by hand starts at the right and adds each digit, possibly carrying a 1 to the digit on the left. Adding binary numbers is identical.

PARTICIPATION ACTIVITY 1.2.8: Adding in binary.

1 2 3 4 2x speed

```
  1 1
  0 1 1 1
+ 0 1 1 0
-------
  1 1 0 1
```

(So 7 + 6 = 13)

Continue for the last digit: 1 + 0 + 0 = 1.

Captions

1. Add rightmost digit: 1 + 0 = 1.
2. Add next digit: 1 + 1 = 10, so carry 1.
3. Continue for next digit: 1 + 1 + 1 = 11. Carry 1.
4. Continue for the last digit: 1 + 0 + 0 = 1.

Feedback?

CHALLENGE ACTIVITY 1.2.2: Binary addition.

547404.4091098.qx3zqy7

Jump to level 1

Add 13 and 15 (Ignore 5th carry bit).

	1	1	1	0	Carry bits
	1	1	0	1	4-bit number
+	1	1	1	1	4-bit number
	1	1	0	0	Result number

1 2 3 4 5 6

Check Next

Done. Click any level to practice more. Completion is

preserved.

✔ Correct.

Feedback?

PARTICIPATION ACTIVITY 1.2.9: Adding binary numbers.

Note: A carry bit may cause the result to have five bits.

1)
```
  0010
+ 0010
---------
```

0100

Check Show answer

Correct

0100

```
  1
  0010
+ 0010
------
  0100
```

So 2 + 2 = 4

2)
```
  0110
+ 0010
---------
```

1000

Check Show answer

Correct

1000

```
  11
  0110
+ 0010
------
  1000
```

So 6 + 2 = 8
Multiple columns can carry a 1.

3)
```
  0101
+ 0111
---------
```

1100

Check Show answer

Correct

1100

```
  111
  0101
+ 0111
------
  1100
```

So 5 + 7 = 12

4) 1111
+ 0001

10000

Check Show answer

Correct

10000

```
 1111
  1111
+ 0001
------
 10000
```

So 15 + 1 = 16
Adding two 4-bit numbers may require a 5-bit result.

Feedback?

Overflow

Overflow occurs when the result of a binary operation is too large to fit in allowed number of bits. Ex: For four-bit numbers, 1111 + 0001 is 10000, which is too large for four bits. When adding two unsigned numbers, if the leftmost bit generates a carry bit, overflow has occurred.

PARTICIPATION ACTIVITY 1.2.10: Overflow for unsigned numbers.

Indicate which operations yield overflow for four-bit binary numbers.

1) 0001 + 0010
   - ○ Overflow
   - ◉ No overflow

Correct
Sum is simply 0011.

2) 0111 + 0111
   - ○ Overflow
   - ◉ No overflow

Correct
Sum is 1110. The leftmost digits did not generate a carry. (In base 10, 7 + 7 is 14, which is representable in 4 bits as 1110).

3) 1000 + 0111
   - ○ Overflow
   - ◉ No overflow

Correct
Sum is 1111. Close, but no overflow.

4) 1000 + 1000
   - ◉ Overflow
   - ○ No overflow

Correct
Leftmost digits generate a carry, meaning overflow. Sum would be 10000, requiring more than four bits. (In base 10, 8 + 8 is 16, which is 10000, requiring 5 bits).

5) 1100 + 0111

- (•) Overflow
- ( ) No overflow

Correct

First (rightmost) bit: 0 + 1 = 1
Second bit: 0 + 1 = 1
Third bit: 1 + 1 = 0, carry 1
Fourth bit: 1 + 0 + carry of 1 = 0, carry 1.
Leftmost digits generated a carry: Overflow. (In base 10, 12 + 7 is 19, which is 10011, requiring 5 bits).

6) 1111 + 1111

- (•) Overflow
- ( ) No overflow

Correct

Obviously! (In base 10, 15 + 15 is 30, which is 11110, requiring 5 bits).

7) In base 10, for 2 digit numbers:
50 + 70

- (•) Overflow
- ( ) No overflow

Correct

The leftmost digits of 5 + 7 yield 2 carry 1 (12). If only two digits are available, the resulting 120 cannot fit and thus is an overflow.

Feedback?

How was this section? **Provide section feedback**