ENGR 101
Engineering Technology

Dr. Kerese Manueli

School of Engineering and Computer Science
Victoria University of Wellington

Victoria
UNIVERSITY OF WELLINGTON
Te Whare Wānanga
o te Ūpoko o te Ika a Māui
CAPITAL CITY UNIVERSITY

Week 3 Lecture 4b

- Main topics
  - Introduction to Engineering Technology
  - Number system
  - Logic Gates
  - Boolean Algebra

- Course web page:
  https://ecs.wgtn.ac.nz/Courses/XMUT101_2021T1/

- kerese@ecs.vuw.ac.nz

Binary Addition

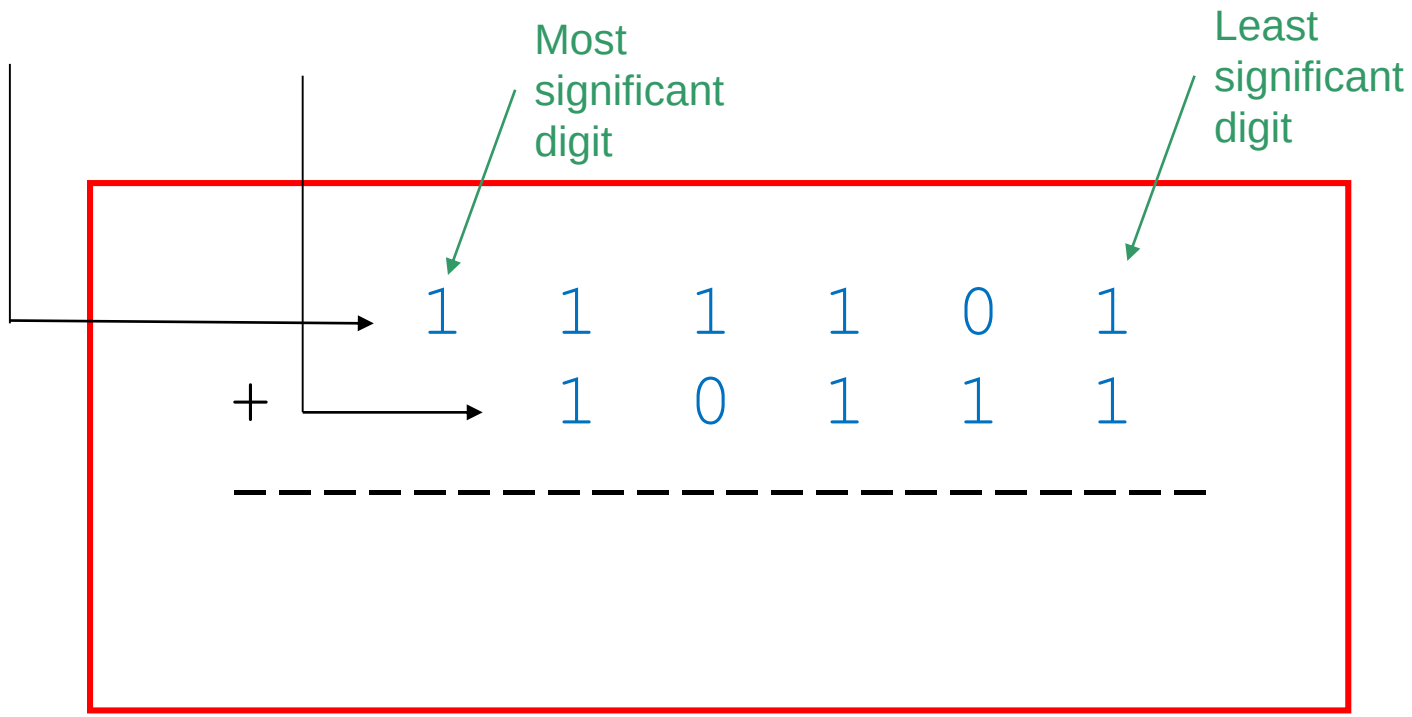
- Binary addition is similar to decimal addition.

- Adding 2 binary numbers:

1 1 1 1 0 1 + 1 0 1 1 1

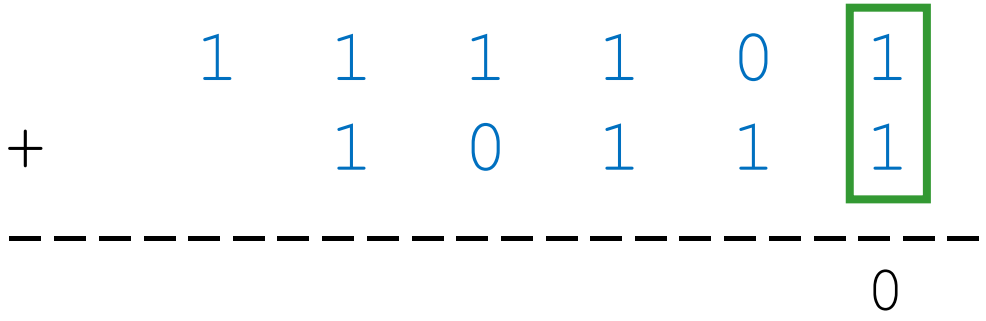
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



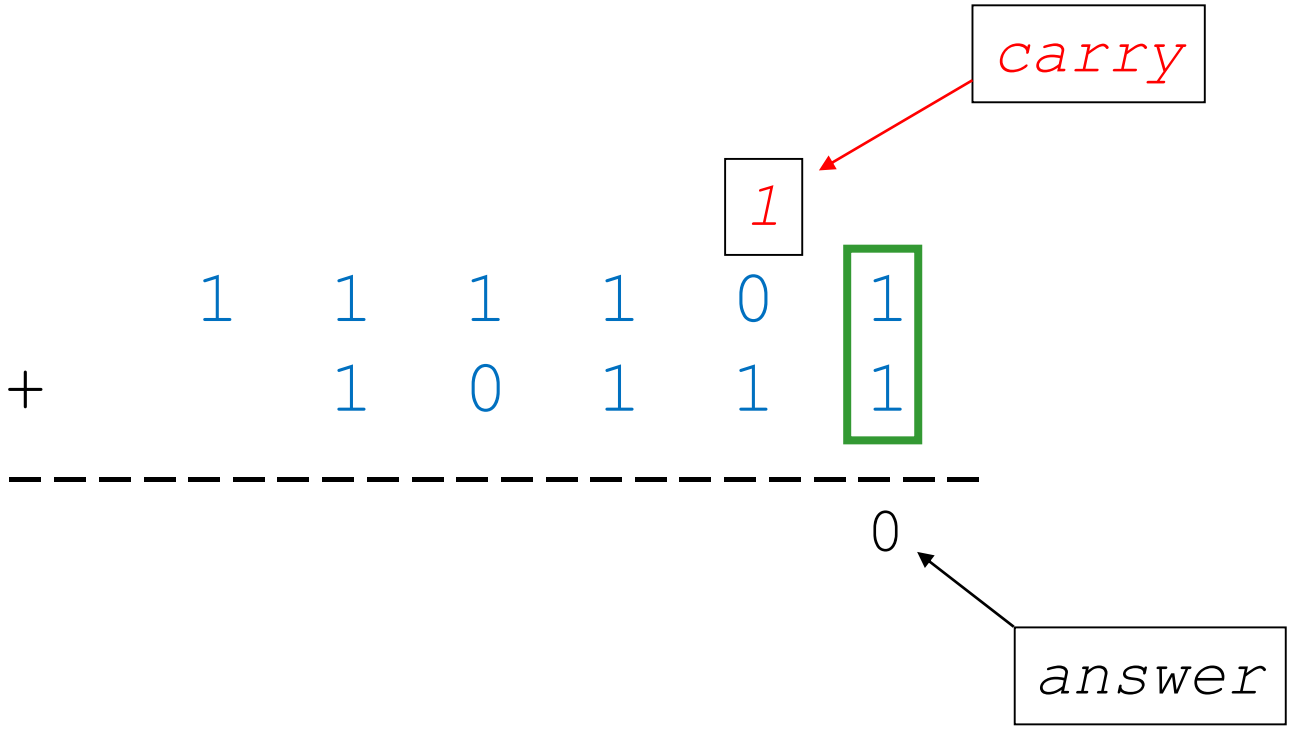
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



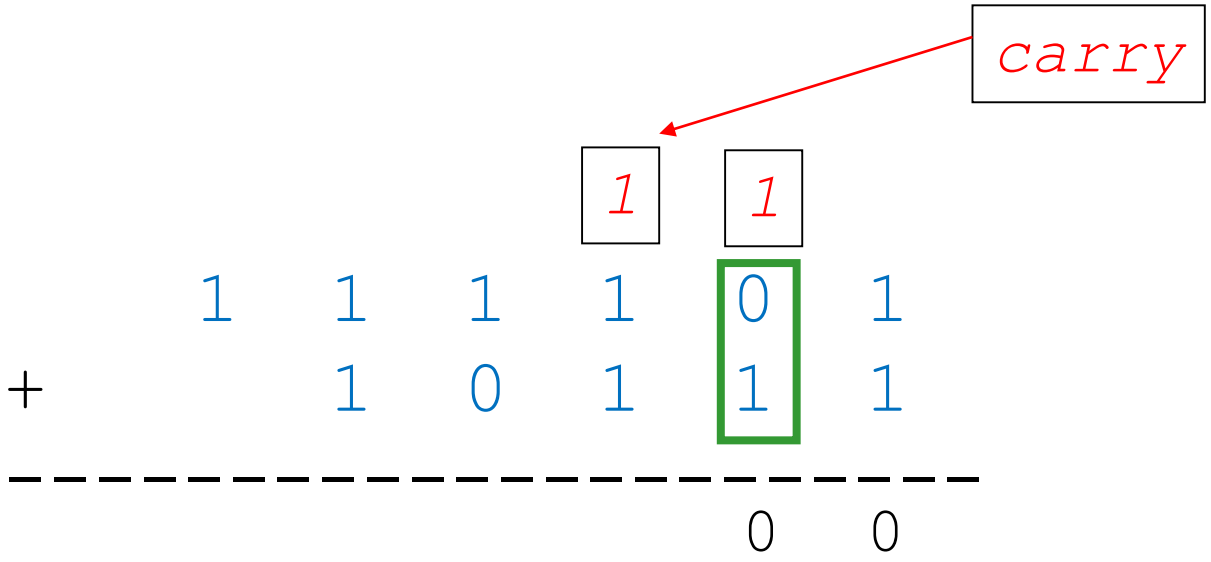
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



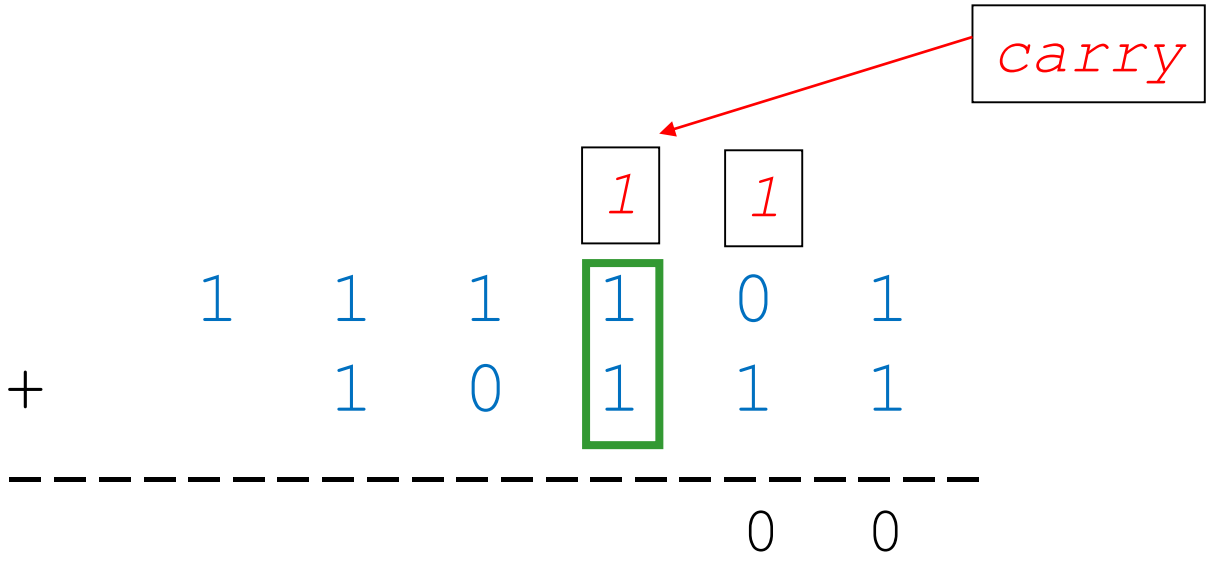
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



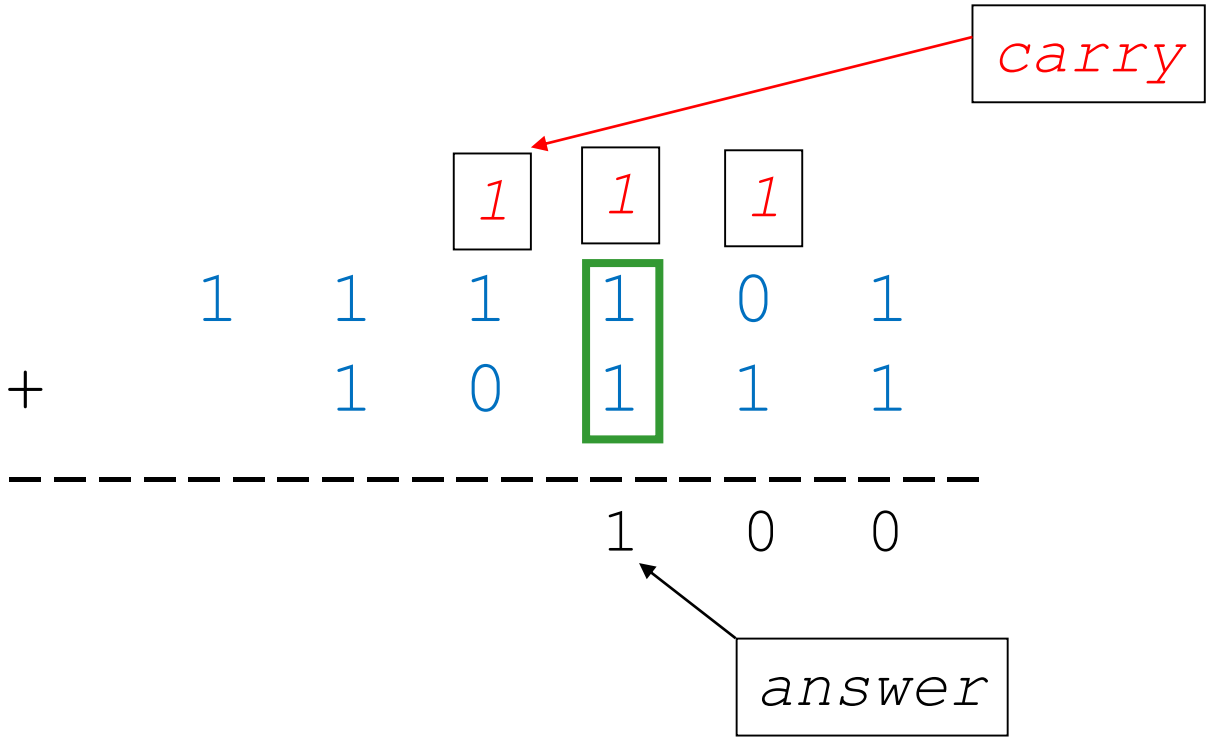
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



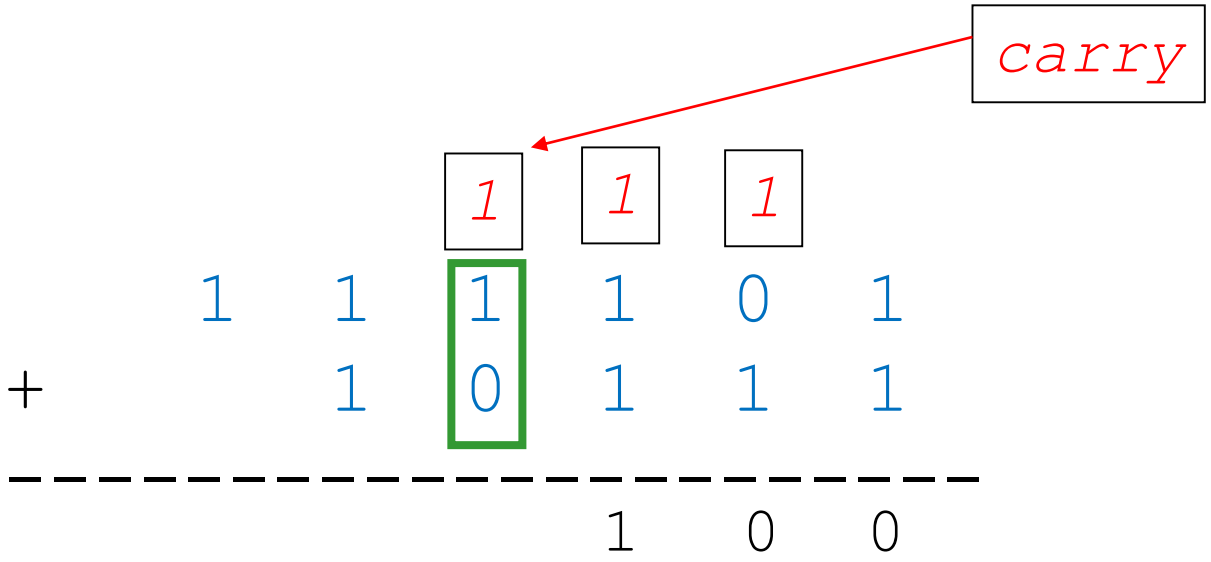
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



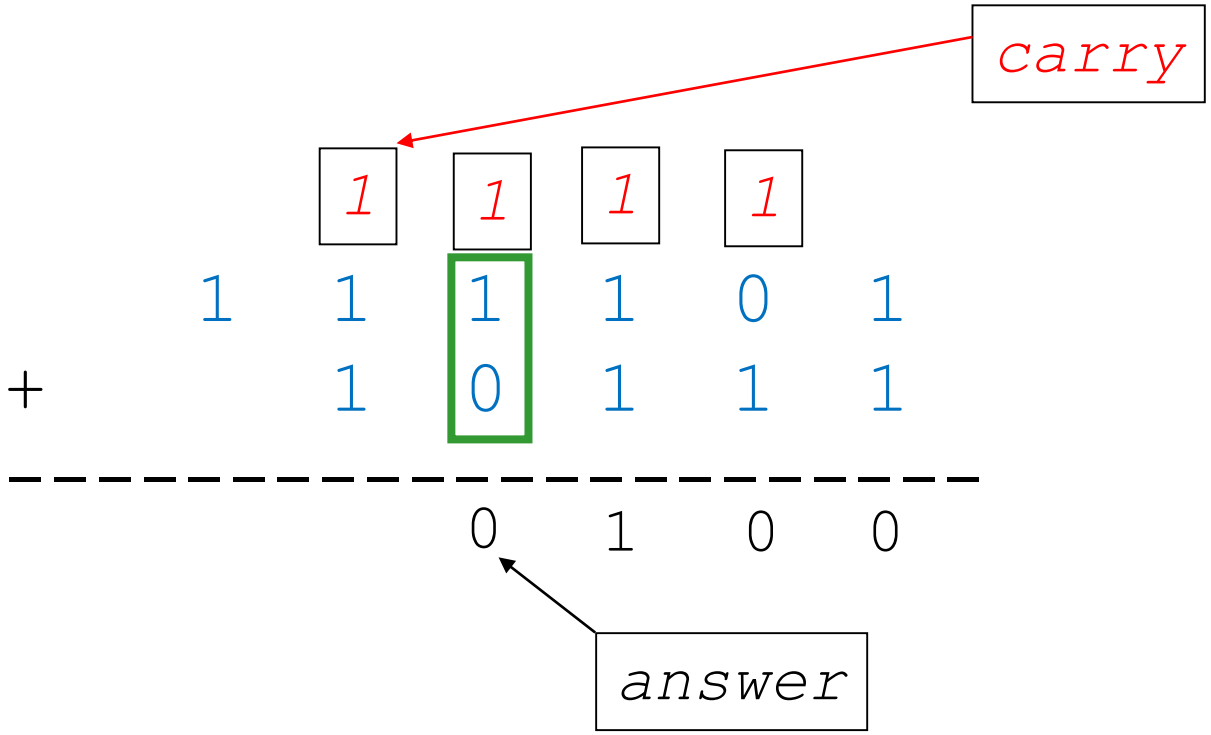
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



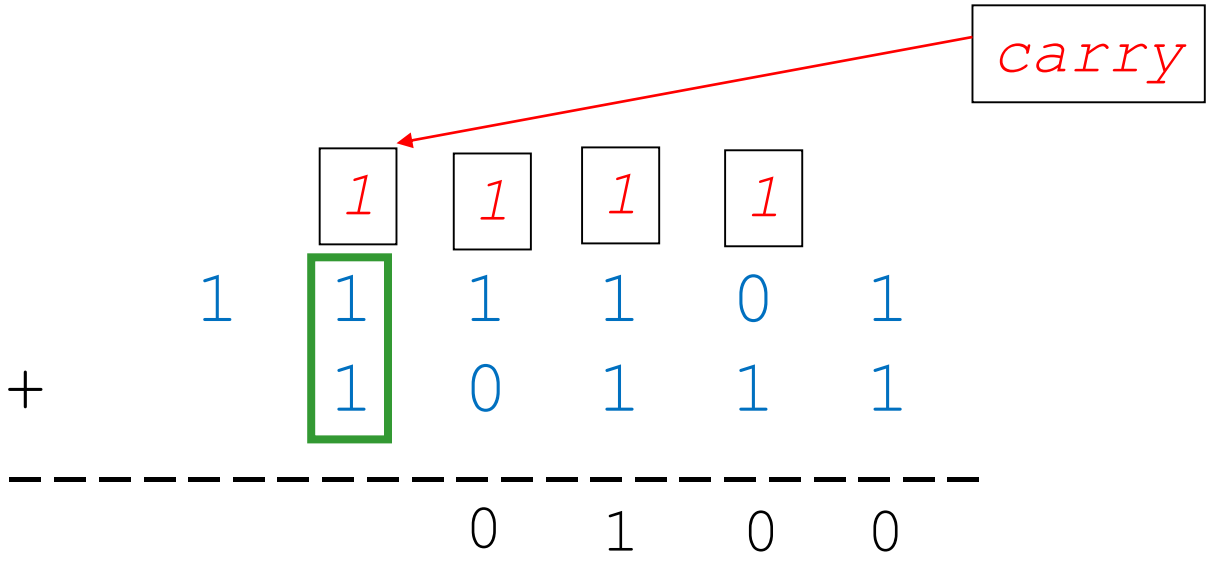
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



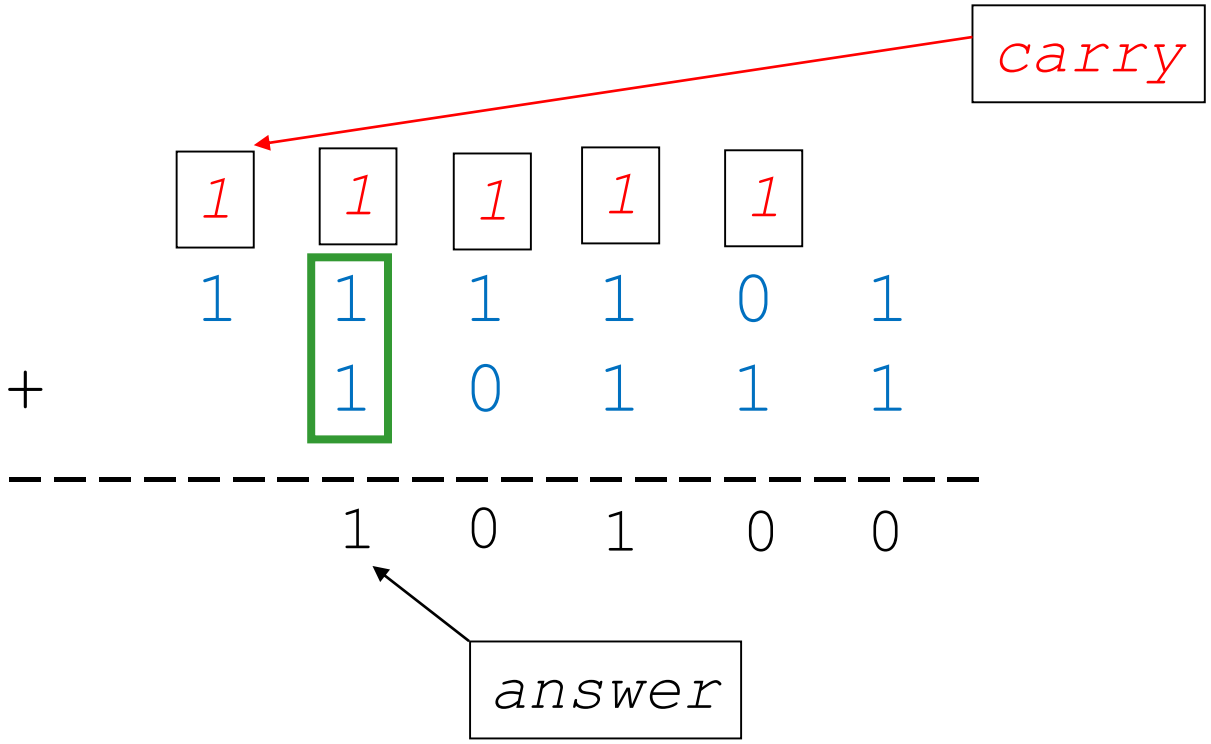
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



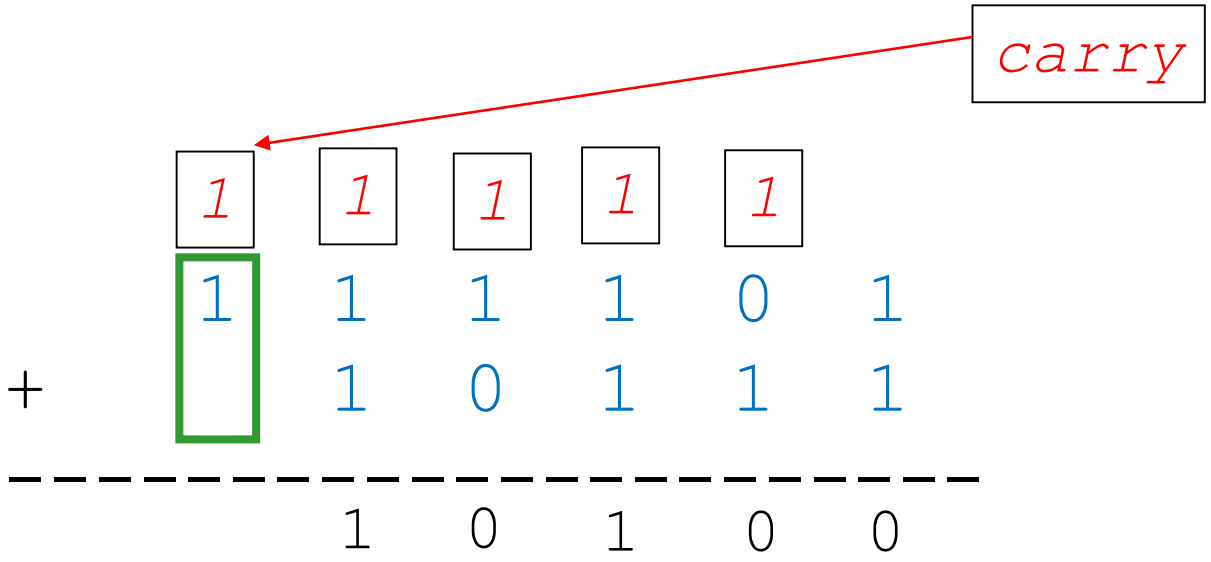
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



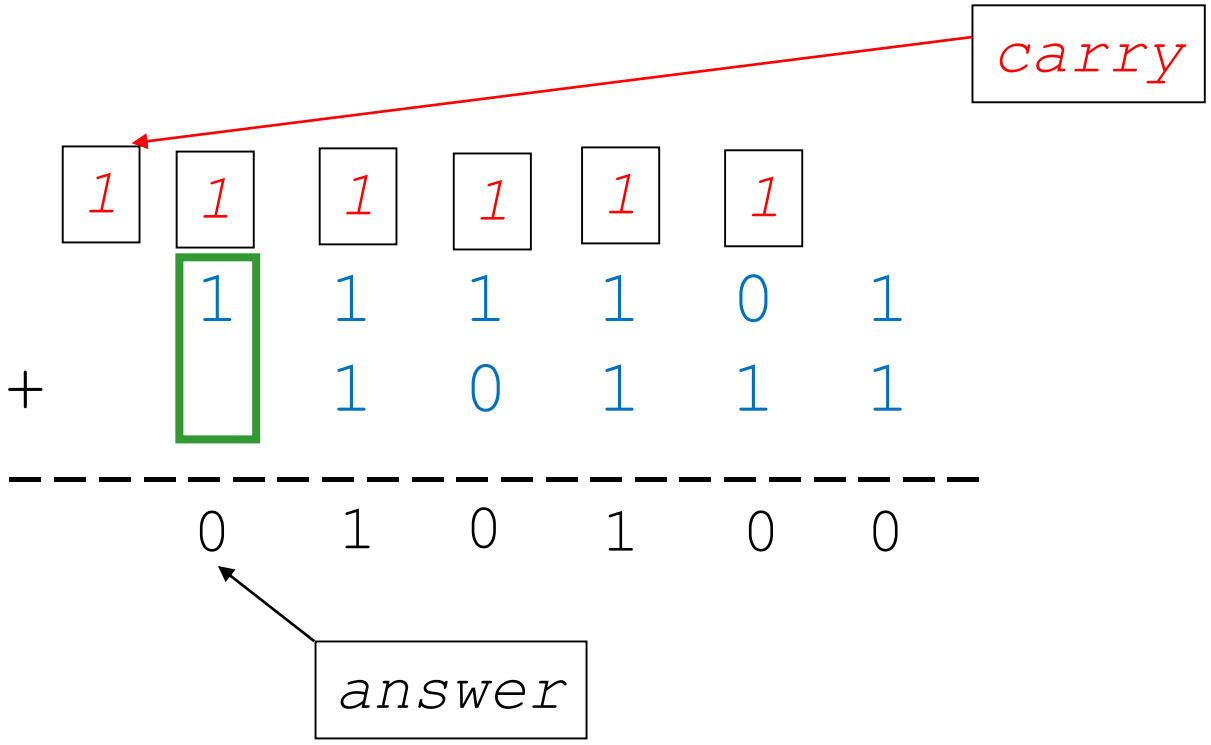
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



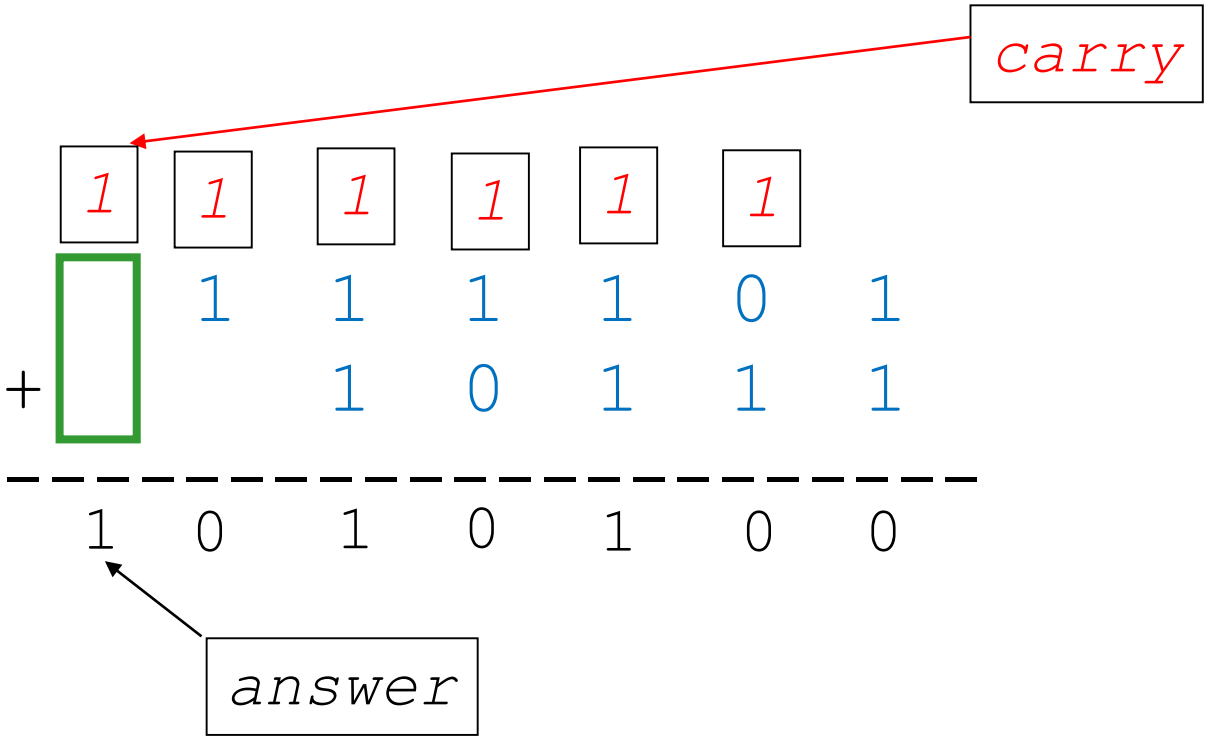
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



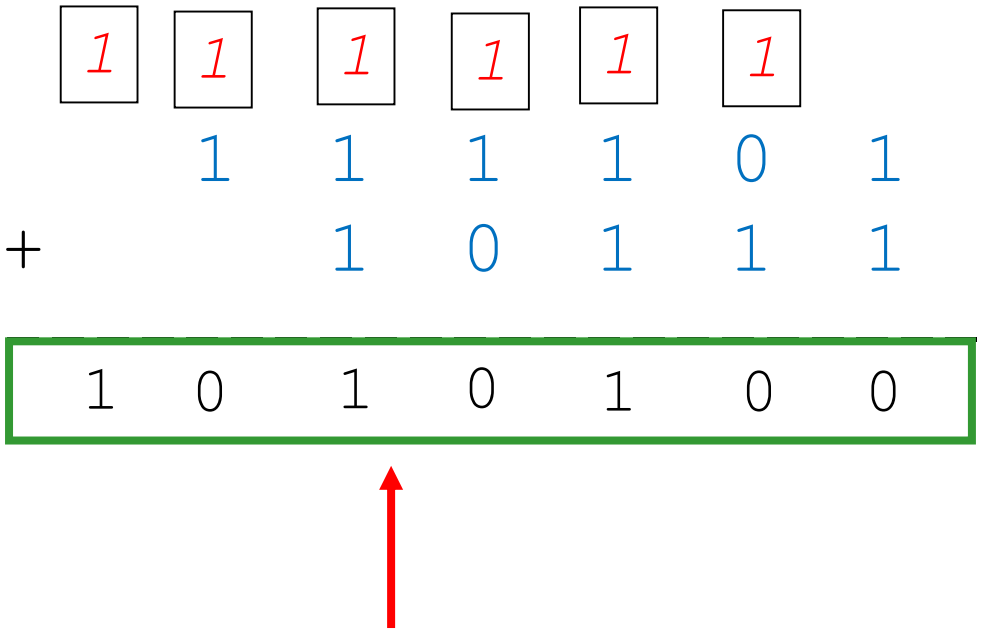
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



Binary Addition

- Adding 2 binary numbers: 111101 + 10111



Binary Addition

- Adding 2 binary numbers: 111101 + 10111

$(1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$
$= (1 \times 32) + (1 \times 16) + (1 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1)$
$= 32 + 16 + 8 + 4 + 0 + 1$
$= 48 + 13$
$= 61$

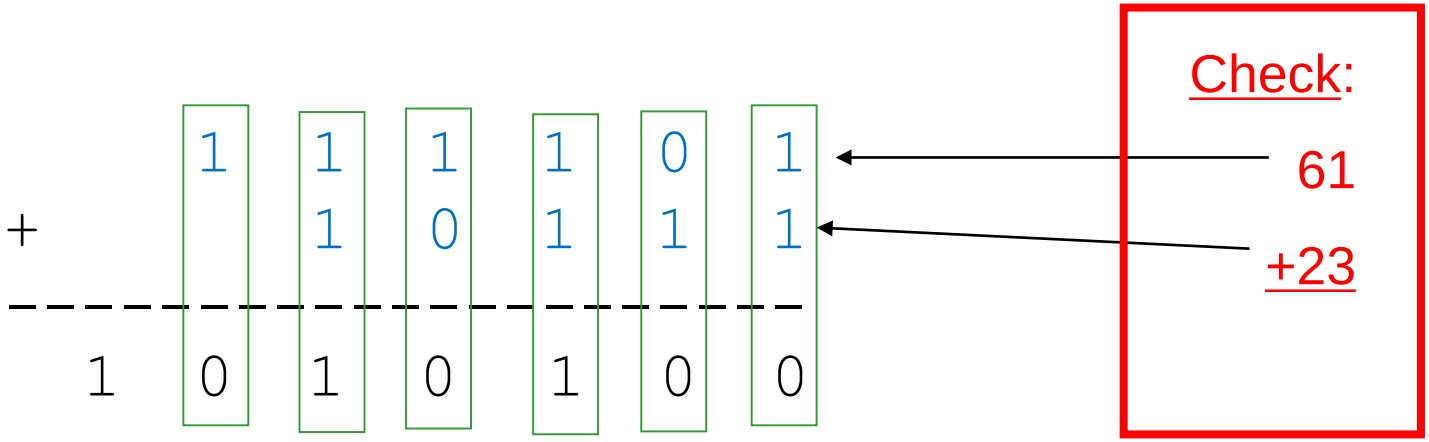
Check:

61

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

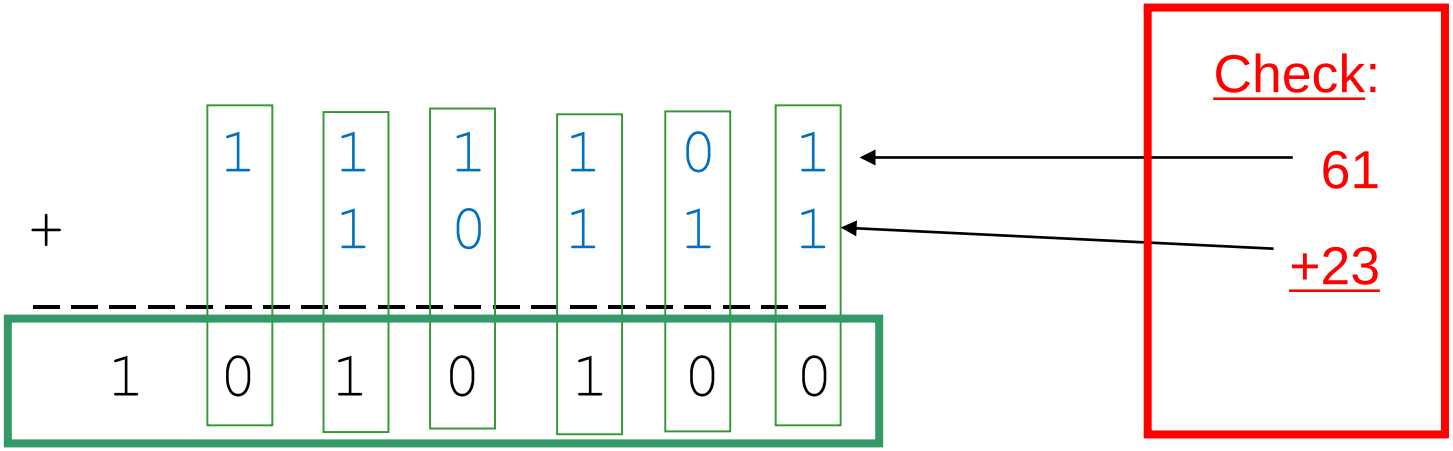
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



Binary Addition

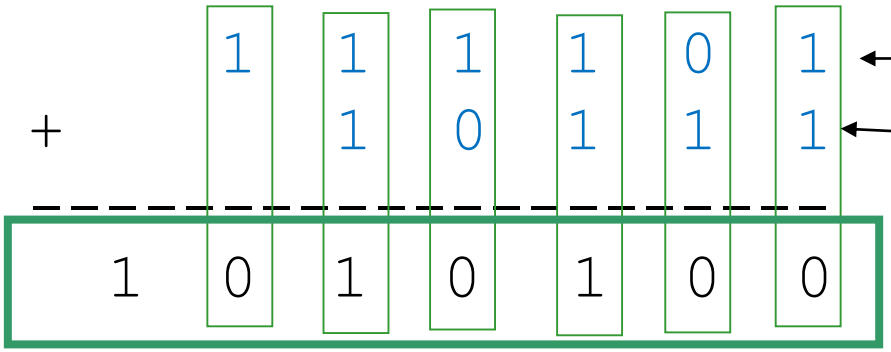
- Adding 2 binary numbers: 111101 + 10111



Binary Addition

- Adding 2 binary numbers: 111101 + 10111

$(1 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0)$
$= (1 \times 64) (0 \times 32) + (1 \times 16) + (0 \times 8) + (1 \times 4) + (0 \times 2) + (0 \times 1)$
$= 64 + 0 + 16 + 0 + 4 + 0 + 0$
$= 64 + 20$
$= 84$



Check:

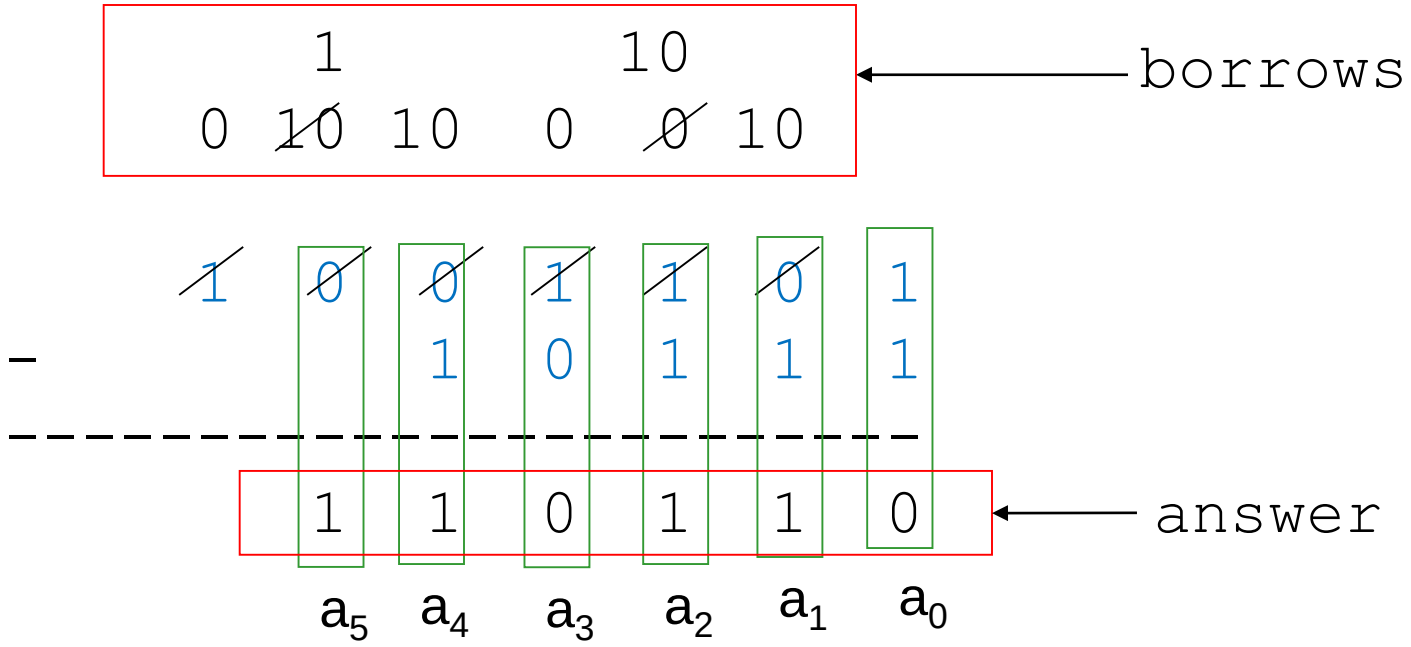
61

+23

84

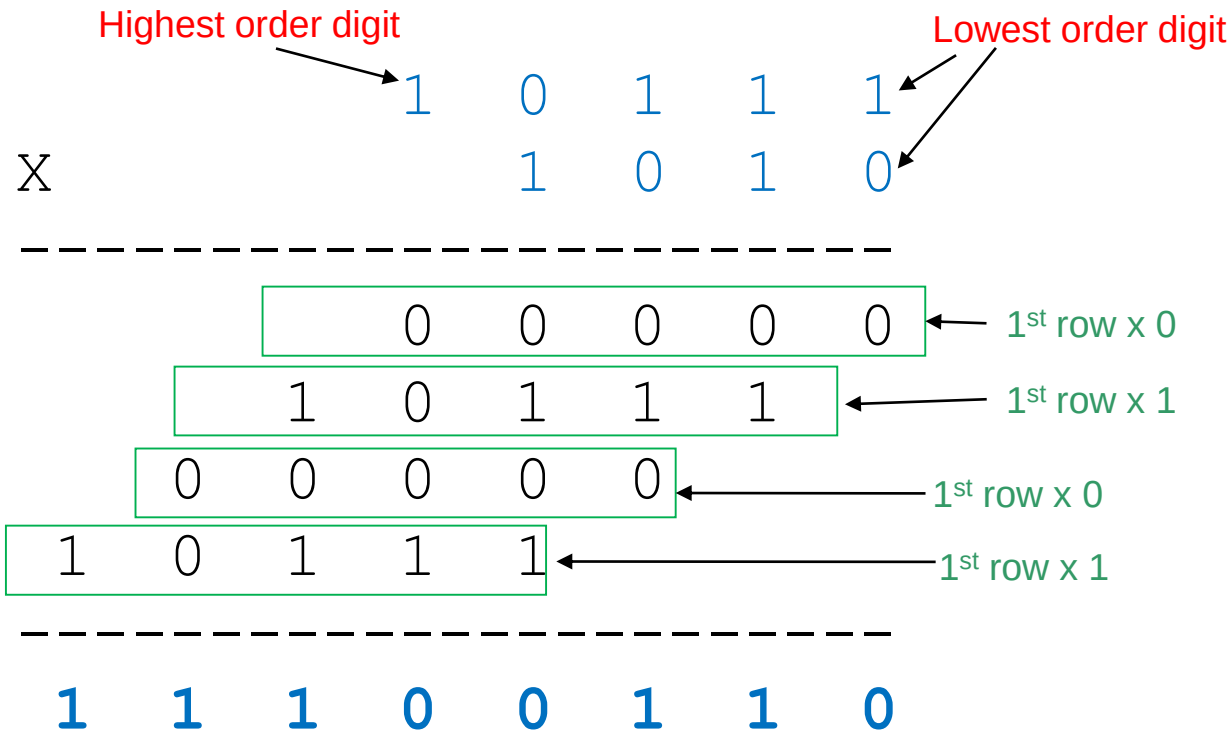
Binary Subtraction

- We can also perform subtraction (with borrows in place of carries).
- Let's subtract $(10111)_2$ from $(1001101)_2$



Binary Multiplication

- Binary multiplication is much the same as decimal multiplication, except that the multiplication operations are much simpler…



How To Represent Signed Numbers?

- Plus (+) and minus (-) signs are used for decimal numbers: 25 (or +25), -16, etc.

How To Represent Signed Numbers

- Plus and minus sign are used for decimal numbers: 25 (or +25), -16, etc.

- For computers, desirable to represent everything as *bits.* (Bit – Binary digit)

How To Represent Signed Numbers

- Plus and minus sign are used for decimal numbers:

  25 (or +25), -16, etc.

- For computers, desirable to represent everything as *bits*. (Bit – Binary digit)

- Three types of signed binary number representations:

  signed magnitude, 1's complement, 2's complement.

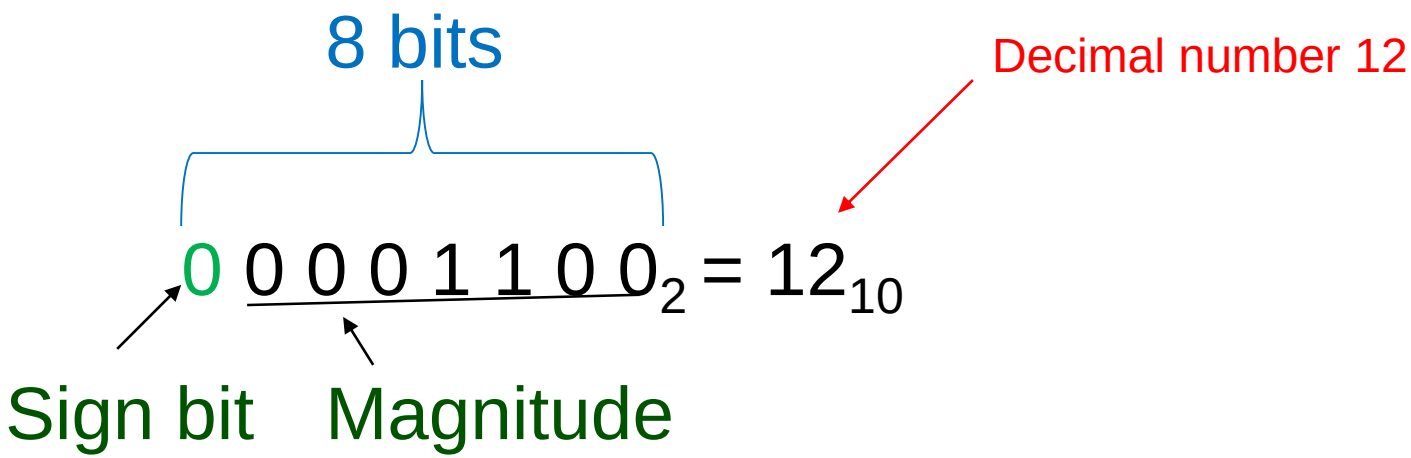
How To Represent Signed Numbers

- Plus and minus sign used for decimal numbers: 25 (or +25), -16, etc.
- For computers, desirable to represent everything as *bits*.
- Three types of signed binary number representations: signed magnitude, 1's complement, 2's complement.
- In each case: left-most bit indicates the sign;

  positive (0) or negative (1).

1) Signed magnitude numbers

Signed magnitude example:



1) Signed magnitude numbers

Consider ***signed magnitude***:

$$00001100_2 = 12_{10}$$

Sign bit Magnitude

$$10001100_2 = -12_{10}$$

Sign bit Magnitude

2) One’s Complement Representation

- One’s complement of a binary number involves inverting all bits.

2) One’s Complement Representation

- One’s complement of a binary number involves inverting all bits.

1. 1’s comp of 00110011 is 11001100
2. 1’s comp of 10101010 is 01010101

2) One's Complement Representation

- The one's complement of a binary number involves inverting all bits.
  - 1's comp of 00110011 is 11001100
  - 1's comp of 10101010 is 01010101
- For an n bit number N the 1's complement is $(2^n-1) - N$.

2) One's Complement Representation

- The one's complement of a binary number involves inverting all bits.
  - 1's comp of 00110011 is 11001100
  - 1's comp of 10101010 is 01010101
- For an n bit number N the 1's complement is $(2^n-1) - N$.
- Called diminished radix complement by Mano since 1's complement for base (radix 2).

2) One's Complement Representation

- The one's complement of a binary number involves inverting all bits.
  - 1's comp of 00110011 is 11001100
  - 1's comp of 10101010 is 01010101
- For an n bit number N the 1's complement is $(2^n-1) - N$.
- Called diminished radix complement by Mano since 1's complement for base (radix 2).
- To find negative of 1's complement number take the 1's complement.

$0\underline{0001100}_2 = 12_{10}$ $1\underline{1110011}_2 = -12_{10}$

Sign bit Magnitude Sign bit Magnitude

3) Two’s Complement Representation

- The two’s complement of a binary number involves inverting all bits and adding 1.

3) Two's Complement Representation

- The two's complement of a binary number involves inverting all bits and adding 1.

1. 2's comp of 00110011 is 11001101
2. 2's comp of 10101010 is 01010110

3) Two's Complement Representation

- The two's complement of a binary number involves inverting all bits and adding 1.
  - 2's comp of 00110011 is 11001101
  - 2's comp of 10101010 is 01010110

- For a n-bit number N the 2's complement is:

$$(2^n-1) - N + 1$$

3) Two's Complement Representation

- The two's complement of a binary number involves inverting all bits and adding 1.
  - 2's comp of 00110011 is 11001101
  - 2's comp of 10101010 is 01010110
- For an n bit number N the 2's complement is
  $(2^n-1) - N + 1$

- Called radix complement by Mano since 2's complement for base (radix 2).

3) Two's Complement Representation

- The two's complement of a binary number involves inverting all bits and adding 1.
  - 2's comp of 00110011 is 11001101
  - 2's comp of 10101010 is 01010110
- For an n bit number N the 2's complement is $(2^n-1) - N + 1$.
- Called radix complement by Mano since 2's complement for base (radix 2).
- To find negative of 2's complement number take the 2's complement.

$0\underline{0001100}_2 = 12_{10}$

Sign bit Magnitude

$1\underline{1110100}_2 = -12_{10}$

Sign bit Magnitude

Two's Complement Shortcuts

Algorithm 1 – Simply complement each bit and then add 1 to the result.

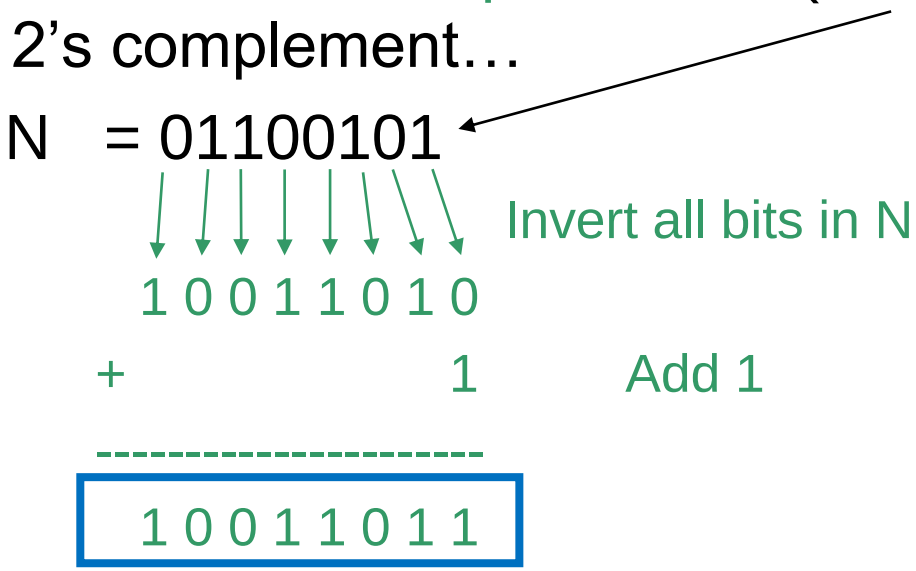
Two's Complement Shortcuts

Algorithm 1 – Simply complement each bit and then add 1 to the result.

- Find the 2's complement of $(01100101)_2$ and of its 2's complement…

Two's Complement Shortcuts

- Algorithm 1 – Simply complement each bit and then add 1 to the result.
  - Find the 2's complement of $(01100101)_2$ and of its 2's complement…



Two's Complement Shortcuts

- Algorithm 1 – Simply complement each bit and then add 1 to the result.
  - Find the 2's complement of $(01100101)_2$ and of its 2's complement…

```
 N   = 01100101          [N] =   10011011
       10011010                  01100100
     +        1          +            1
     ----------          ----------
       10011011                  01100101
```

Two's Complement Shortcuts

- Algorithm 1 – Simply complement each bit and then add 1 to the result.
  - Finding the 2's complement of $(01100101)_2$ and of its 2's complement…

```
N   = 01100101        [N] =   10011011
      10011010              01100100
    +        1        +            1
    ----------------      ----------------
      10011011              01100101
```

- Algorithm 2 – Start with the least significant bit, copy all of the bits up to and including the first 1 bit and then complementing the remaining bits.

Two's Complement Shortcuts

- Algorithm 1 – Simply complement each bit and then add 1 to the result.
  - Finding the 2's complement of $(01100101)_2$ and of its 2's complement…

N = 01100101		[N] = 10011011
10011010		01100100
+ 1	+	1
10011011		01100101

- Algorithm 2 – Start with the least significant bit, copy all of the bits up to and including the first 1 bit and then complementing the remaining bits.

N = 0 1 1 0 0 1 0 1 ← least significant bit

[N] = 1 0 0 1 1 0 1 1

Finite Number Representation

- Machines that use 2's complement arithmetic can represent integers in the range:

  $-2^{n-1} <= N <= 2^{n-1}-1$

  where n is the number of bits available for representing N.

- Note that $2^{n-1}-1 = (011..11)_2$ and

  $-2^{n-1} = (100..00)_2$

Finite Number Representation

- Machines that use 2's complement arithmetic can represent integers in the range

  $$-2^{n-1} <= N <= 2^{n-1}-1$$

  where n is the number of bits available for representing N. Note that $2^{n-1}-1 = (011..11)_2$ and $-2^{n-1} = (100..00)_2$

- For 2's complement there are more negative numbers than positive.

Finite Number Representation

- Machines that use 2's complement arithmetic can represent integers in the range

  $$-2^{n-1} <= N <= 2^{n-1}-1$$

  where n is the number of bits available for representing N. Note that $2^{n-1}-1 = (011..11)_2$ and $-2^{n-1} = (100..00)_2$
- For 2's complement more negative numbers than positive.

- For 1's complement there are two representations for zero.

Finite Number Representation

- Machines that use 2's complement arithmetic can represent integers in the range

  $$-2^{n-1} <= N <= 2^{n-1}-1$$

  where n is the number of bits available for representing N. Note that $2^{n-1}-1 = (011..11)_2$ and $-2^{n-1} = (100..00)_2$
- For 2's complement more negative numbers than positive.
- For 1's complement two representations for zero.
- For a n bit number in base (radix) z there are z^n different unsigned values.

  $$(0, 1, \ldots z^{n-1})$$

1’s Complement Addition

- Adding 1’s complement numbers is easy.

1’s Complement Addition

- Adding 1’s complement numbers is easy.

- For example, to add $+(1100)_2$ and $+(0001)_2$.

1’s Complement Addition

- Adding 1’s complement numbers is easy.

- For example, to add $+(1100)_2$ and $+(0001)_2$.

- Let’s compute $(12)_{10} + (1)_{10}$.
  - $(12)_{10} = +(1100)_2 = 01100_2$ in 1’s comp.
  - $(1)_{10} = +(0001)_2 = 00001_2$ in 1’s comp.

1's Complement Addition

- Adding 1's complement numbers is easy.
- For example, to add $+(1100)_2$ and $+(0001)_2$.

- Let's compute $(12)_{10} + (1)_{10}$.
  - $(12)_{10} = +(1100)_2 = 01100_2$ in 1's comp.
  - $(1)_{10} = +(0001)_2 = 00001_2$ in 1's comp.

Step 1: Add binary numbers

Step 2: Add carry to low-order bit

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

1’s Complement Subtraction

- Subtracting 1’s complement numbers is also easy.

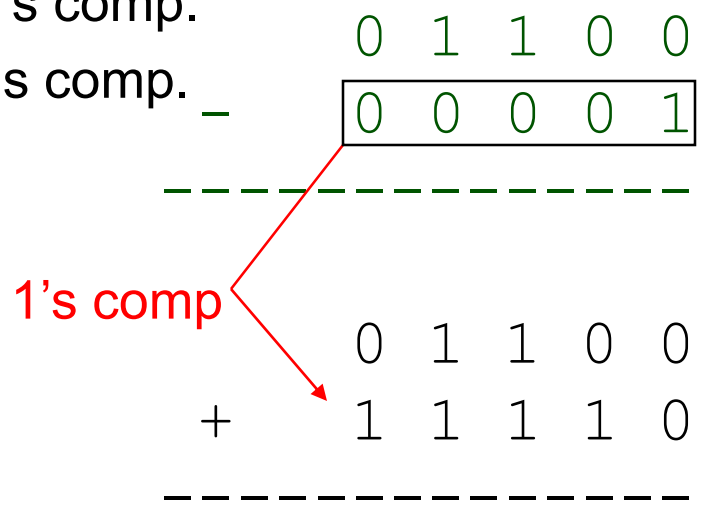
1’s Complement Subtraction

- Subtracting 1’s complement numbers is also easy.
- Let’s compute $(12)_{10}$ - $(1)_{10}$.
  - $(12)_{10}$ = +$(1100)_2$ = 01100_2 in 1’s complement
  - $(-1)_{10}$ = -$(0001)_2$ = 11110_2 in 1’s complement

1's Complement Subtraction

- Subtracting 1's complement numbers is also easy.
- For example, subtract $+(0001)_2$ from $+(1100)_2$.
- Let's compute $(12)_{10}$ - $(1)_{10}$.
  - $(12)_{10}$ = $+(1100)_2$ = 01100_2 in 1's comp.
  - $(-1)_{10}$ = $-(0001)_2$ = 11110_2 in 1's comp.

Step 1: Take 1's complement of 2nd operand.



1's Complement Subtraction

- Subtracting 1's complement numbers is also easy.
- For example, subtract $+(0001)_2$ from $+(1100)_2$.
- Let's compute $(12)_{10}$ - $(1)_{10}$.
  - $(12)_{10}$ = $+(1100)_2$ = 01100_2 in 1's comp.
  - $(-1)_{10}$ = $-(0001)_2$ = 11110_2 in 1's comp.

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

Step 1: Take 1's complement of 2nd operand.

Step 2: Add the binary numbers.

Add

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

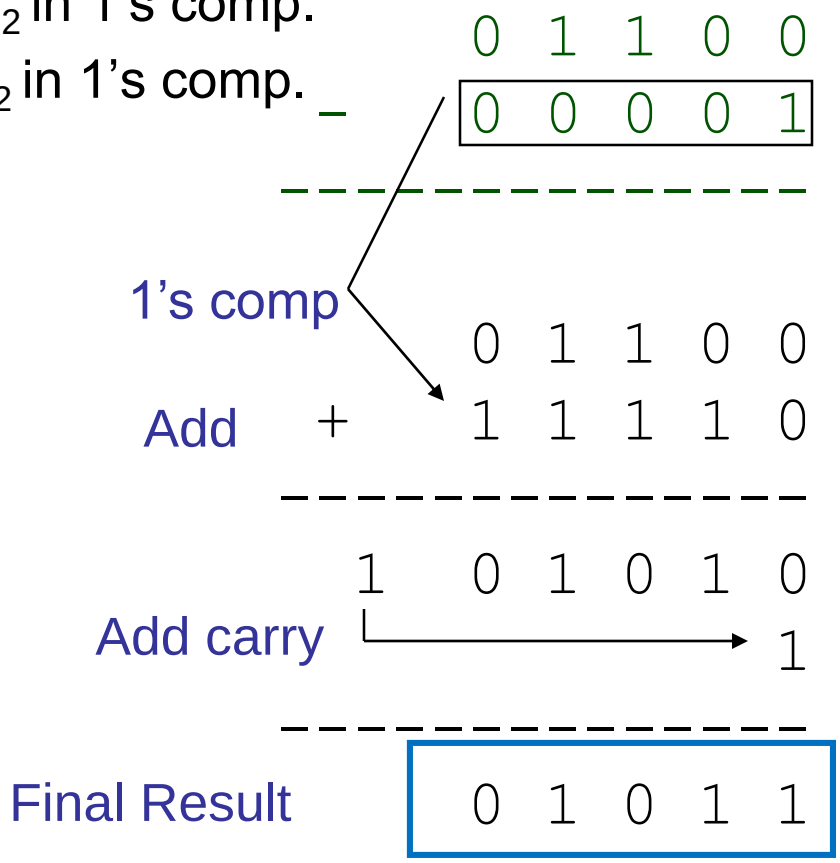
1's Complement Subtraction

- Subtracting 1's complement numbers is also easy.
- For example, subtract $+(0001)_2$ from $+(1100)_2$.
- Let's compute $(12)_{10}$ - $(1)_{10}$.
  - $(12)_{10}$ = $+(1100)_2$ = 01100_2 in 1's comp.
  - $(-1)_{10}$ = $-(0001)_2$ = 11110_2 in 1's comp.

Step 1: Take 1's complement of 2nd operand

Step 2: Add binary numbers

Step 3: Add carry to low order bit



2’s Complement Addition

- Adding 2’s complement numbers is easy.

2's Complement Addition

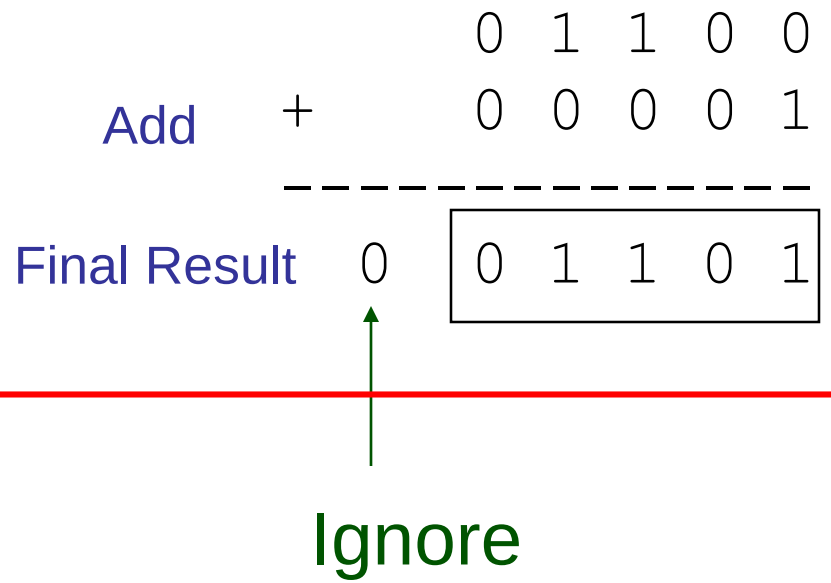
- Adding 2's complement numbers is easy.
- Let's compute $(12)_{10} + (1)_{10}$.
  - $(12)_{10} = +(1100)_2 = 01100_2$ in 2's complement
  - $(1)_{10} = +(0001)_2 = 00001_2$ in 2's complement

2's Complement Addition

- Adding 2's complement numbers is easy.
- Let's compute $(12)_{10} + (1)_{10}$.
  - $(12)_{10} = +(1100)_2 = 01100_2$ in 2's comp.
  - $(1)_{10} = +(0001)_2 = 00001_2$ in 2's comp.

Step 1: Add binary numbers

Step 2: Ignore carry bit



2's Complement Subtraction

- Follow the 3 steps for subtraction.
- Let's compute $(12)_{10}$ - $(1)_{10}$.
  - $(12)_{10}$ = +$(1100)_2$ = 0$1100_2$ in 2's comp.
  - $(-1)_{10}$ = -$(0001)_2$= 1$1111_2$ in 2's comp.

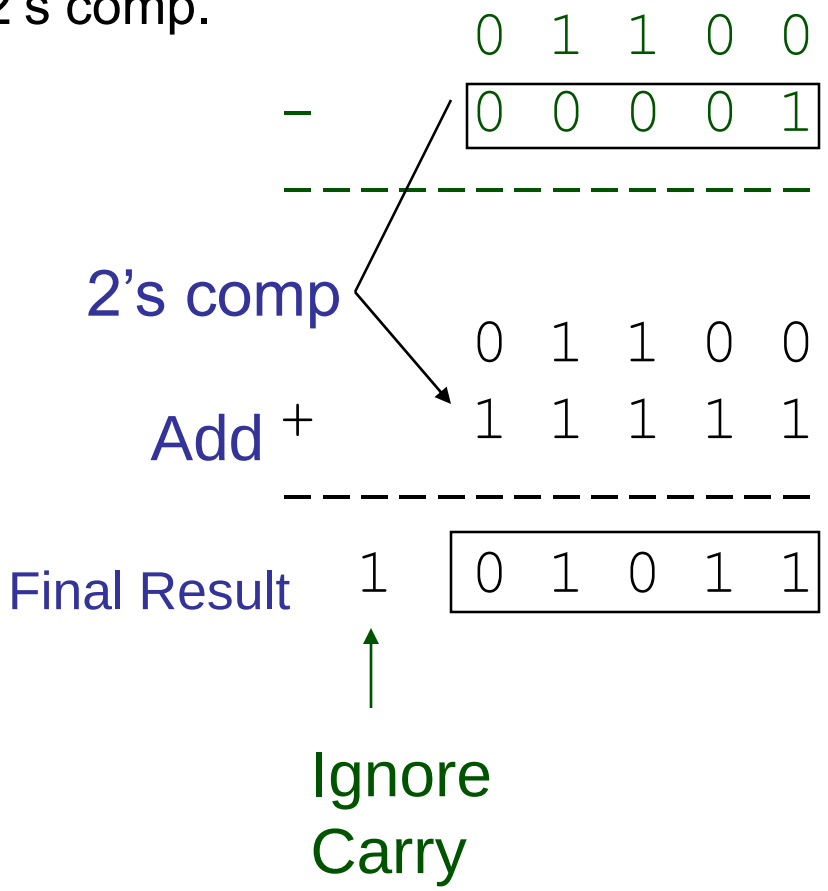
2's Complement Subtraction

- Follow the 3 steps for subtraction.
- Let's compute $(12)_{10}$ - $(1)_{10}$.
  - $(12)_{10}$ = +$(1100)_2$ = 01100_2 in 2's comp.
  - $(-1)_{10}$ = -$(0001)_2$= 11111_2 in 2's comp.

Step 1: Take 2's complement of 2nd operand

Step 2: Add binary numbers

Step 3: Ignore carry bit



2's Complement Subtraction: Exercise 1

Compute $(13)_{10} - (5)_{10}$ using the 2s complement form.

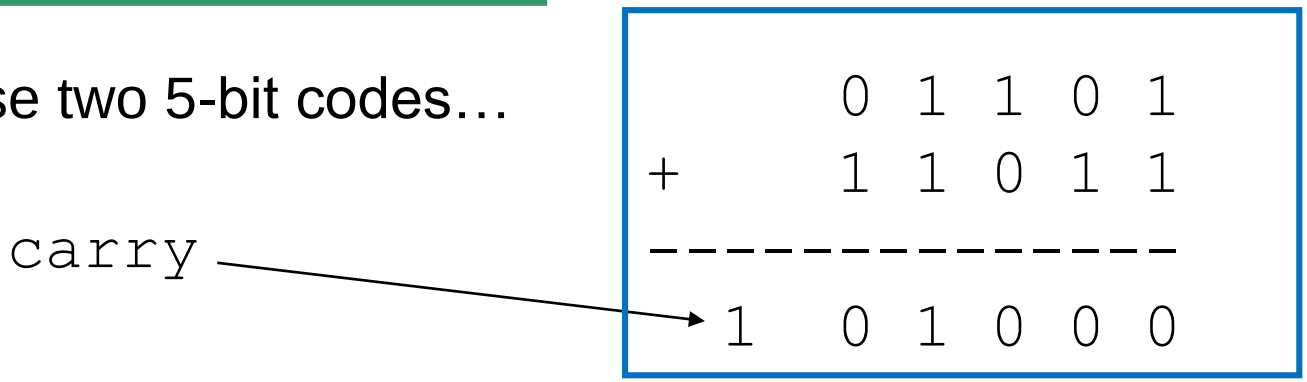
5 minutes
to complete this exercise!!

2's Complement Subtraction: Exercise 1

- Let's compute $(13)_{10} - (5)_{10}$.

$(13)_{10} = +(1101)_2 = (01101)_2$
$(-5)_{10} = -(0101)_2 = (11011)_2$

- Adding these two 5-bit codes…



- Discarding the carry bit, the sign bit is seen to be zero, indicating a correct result. Indeed,

$(01000)_2 = +(1000)_2 = +(8)_{10}$

2's Complement Subtraction: Exercise 2

Compute $(5)_{10} - (12)_{10}$

5 minutes
to complete this exercise!!

2's Complement Subtraction: Exercise 2

- Let's compute $(5)_{10} - (12)_{10}$

$(-12)_{10} = -(1100)_2 = (10100)_2$

$(5)_{10} = +(0101)_2 = (00101)_2$

- Adding these two 5-bit codes…

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

- Here, there is no carry bit and the sign bit is 1. This indicates a negative result, which is what we expect.

$(11001)_2 = -(7)_{10}$

Week 3 Lecture 4b

- Binary numbers arithmetic
- Signed binary numbers: signed magnitude, 1st complement and 2s complement
- 1s and 2s complement arithmetic

- Course web page:
  https://ecs.wgtn.ac.nz/Courses/XMUT101_2021T1/
- kerese@ecs.vuw.ac.nz