ENGR (XMUT) 101
Engineering Technology

A/Prof. Pawel Dmochowski

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 10 Lecture 1

- Main topics
  - Number system – part 2
    - Conversion between binary to decimal; octal to decimal and hexadecimal to decimal

Convert an Integer from Binary *to* Decimal

Convert the binary number $(1001)_2$ to decimal number.

Highest order digit

Lowest order digit

a) Binary number	**1**	**0**	**0**	**1**
b) Base 2	2^3	2^2	2^1	2^0

Convert an Integer from Binary *to* Decimal

Convert the binary number $(1001)_2$ to decimal number.

Highest order digit

Lowest order digit

a) Binary number	**1**	**0**	**0**	**1**
b) Base 2	2^3	2^2	2^1	2^0
c) Decimal equivalent	8	4	2	1

Convert an Integer from Binary *to* Decimal

Convert the binary number $(1001)_2$ to decimal number.

Highest order digit → (leftmost digit); Lowest order digit → (rightmost digit)

a) Binary number	**1**	**0**	**0**	**1**
b) Base 2	2^3	2^2	2^1	2^0
c) Decimal equivalent	8	4	2	1
a) x c)	1 x 8 = 8	0 x 4 = 0	0 x 2 = 0	1 x 1 = 1

Convert an Integer from Binary *to* Decimal

Convert the binary number $(1001)_2$ to decimal number.

a) Binary number	**1**	**0**	**0**	**1**
b) Base 2	2^3	2^2	2^1	2^0
c) Decimal equivalent	8	4	2	1
a) x b)	1 x 8 = 8	0 x 4 = 0	0 x 2 = 0	1 x 1 = 1

$$(1001)_2 = (8 + 0 + 0 + 1)_{10} = (9)_{10}$$

Convert an Integer *from* Binary *to* Decimal

Highest order digit

Lowest order digit

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal Digit Value	256	128	64	32	16	8	4	2	1

Convert an Integer *from* Binary *to* Decimal

Highest order digit

Lowest order digit

Decimal Digit Value

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1
					1	0	0	1

First digit

Last digit

Convert an Integer *from* Binary *to* Decimal

Highest order digit

Lowest order digit

Decimal Digit Value

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Convert binary number 1001 to decimal

Convert an Integer *from* Binary *to* Decimal

Highest order digit

Lowest order digit

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Decimal Digit Value

Convert binary number 1001 to decimal

$$(1\ 0\ 0\ 1)_2 = (1\text{x}8) + (0\text{x}4) + (0\text{x}2) + (1\text{x}1) = (9)_{10}$$

Convert an Integer *from* Binary *to* Decimal

Exercise 4.1

Convert the following binary numbers to decimal:

a) $(1\ 0\ 1\ 1)_2$

b) $(1\ 0\ 1\ 0\ 1\ 0)_2$

c) $(1\ 1\ 1\ 1\ 0\ 1)_2$

d) $(1\ 1\ 0\ 0\ 0\ 1\ 0)_2$

5 minutes to convert these 4 binary numbers to decimal!!

Convert an Integer *from* Binary *to* Decimal

Exercise 4.1

Convert the following binary numbers to decimal:

a) $(1\ 0\ 1\ 1)_2$ $= (11)_{10}$

b) $(1\ 0\ 1\ 0\ 1\ 0)_2$ $= (42)_{10}$

c) $(1\ 1\ 1\ 1\ 0\ 1)_2$ $= (61)_{10}$

d) $(1\ 1\ 0\ 0\ 0\ 1\ 0)_2$ $= (98)_{10}$

Convert an Integer *from* Binary *to* Octal

1) First convert the **binary number** to decimal

2) Convert the decimal number to **octal**

Convert an Integer *from* Binary *to* Octal

Most Significant Bit

Least Significant Bit

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Decimal Digit Value

1) Convert binary number **1001** to decimal

$$(1\ 0\ 0\ 1)_2 = (1x8) + (0x4) + (0x2) + (1x1) = (9)_{10}$$

Convert an Integer *from* Binary *to* Octal

Most Significant Bit → Least Significant Bit

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal Digit Value	256	128	64	32	16	8	4	2	1

1) Convert binary number 1001 to decimal

$(1\ 0\ 0\ 1)_2 = (1x8) + (0x4) + (0x2) + (1x1) = (9)_{10}$

2) Convert $(9)_{10}$ to octal: $(9)_{10} \rightarrow 9/8 = 1$ remainder 1

$(9)_{10} \rightarrow (11)_8$

Convert an Integer *from* Binary *to* Octal

Exercise 4.2

Convert the following binary numbers to octal:

a) $(1\ 0\ 1\ 1)_2$

b) $(1\ 0\ 1\ 0\ 1\ 0)_2$

c) $(1\ 1\ 1\ 1\ 0\ 1)_2$

d) $(1\ 1\ 0\ 0\ 0\ 1\ 0)_2$

5 minutes to convert these 4 binary numbers to octal numbers!!

Convert an Integer *from* Binary *to* Octal

Exercise 4.2

Convert the following binary numbers to octal:

a) $(1\ 0\ 1\ 1)_2$ $= (13)_8$

b) $(1\ 0\ 1\ 0\ 1\ 0)_2$ $= (52)_8$

c) $(1\ 1\ 1\ 1\ 0\ 1)_2$ $= (75)_8$

d) $(1\ 1\ 0\ 0\ 0\ 1\ 0)_2$ $= (142)_8$

Convert an Integer *from* Binary *to* Hexadecimal

1) First convert the **binary number** to decimal

2) Convert the decimal number to **hexadecimal**

Convert an Integer *from* Binary *to* Hexadecimal

Most Significant Bit

Least Significant Bit

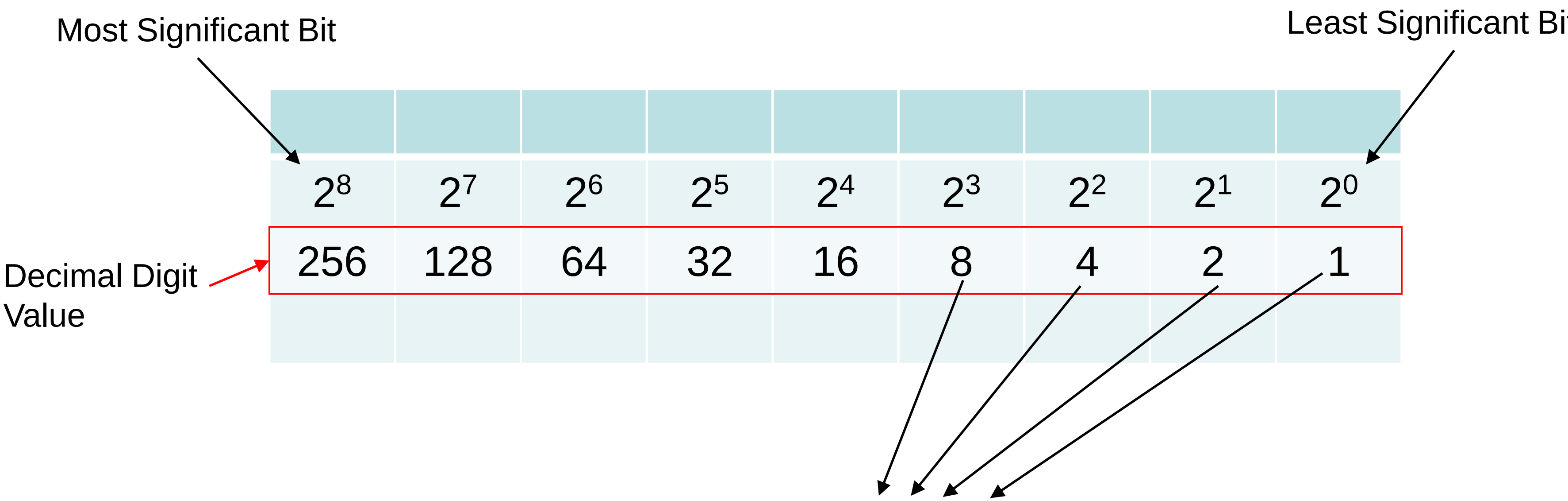
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Decimal Digit Value

Convert binary number 1001 to decimal

$$(1\ 0\ 0\ 1)_2 = (1x8) + (0x4) + (0x2) + (1x1) = (9)_{10}$$

Convert an Integer *from* Binary *to* Hexadecimal



Convert binary number 1001 to decimal

$(1\ 0\ 0\ 1)_2 = (1x8) + (0x4) + (0x2) + (1x1) = (9)_{10}$

Convert $(9)_{10}$ to hexadecimal:

9/16 = 0 remainder 9

$(9)_{10} \rightarrow (9)_{16}$

Converting between Base 16 and Base 2

- Conversion is easy!
  - Determine 4-bit value for each hex digit
- Note that there are $2^4 = 16$ different values of four bits
- Easier to read and write in hexadecimal.
- Representations are equivalent!

Converting between Base 16 and Base 2

$(3A9F)_{16} = (\underline{0011}\ \ \underline{1010}\ \ \underline{1001}\ \ \underline{1111})_2$

0011	1010	1001	1111
↑	↑	↑	↑
3	A	9	F

Converting Between Base 16 and Base 8

1. Convert from Base 16 to Base 2
2. Regroup bits into groups of three starting from right side
3. Ignore leading zeros
4. Each group of three bits forms an octal digit.

Converting Between Base 16 and Base 8

$(3A9F)_{16} = (0011\ 1010\ 1001\ 1111)_2$

1. Convert from Base 16 to Base 2
2. Regroup bits into groups of three starting from right side
3. Ignore leading zeros
4. Each group of three bits forms an octal digit.

Converting Between Base 16 and Base 8

$3A9F_{16} = (0011\ 1010\ 1001\ 1111)_2$

Start from right side

1. Convert from Base 16 to Base 2
2. Regroup bits into groups of three starting from right side
3. Ignore leading zeros
4. Each group of three bits forms an octal digit.

Converting Between Base 16 and Base 8

$3A9F_{16}$ = 0011 1010 1001 1111_2

3 A 9 F

011 101 010 011 111_2

3 5 2 3 7

1. Convert from Base 16 to Base 2
2. Regroup bits into groups of three starting from right
3. Ignore leading zeros
4. Each group of three bits forms an octal digit.

Converting Between Base 16 and Base 8

$3A9F_{16} = \underline{0011}\ \underline{1010}\ \underline{1001}\ \underline{1111}_2$

3 A 9 F

$35237_8 = \underline{011}\ \underline{101}\ \underline{010}\ \underline{011}\ \underline{111}_2$

3 5 2 3 7

1. Convert from Base 16 to Base 2
2. Regroup bits into groups of three starting from right
3. Ignore leading zeros
4. Each group of three bits forms an octal digit.

Week 10 Lecture 2

- Main topics
  - Number system
    - Binary Arithmetic

Binary Addition

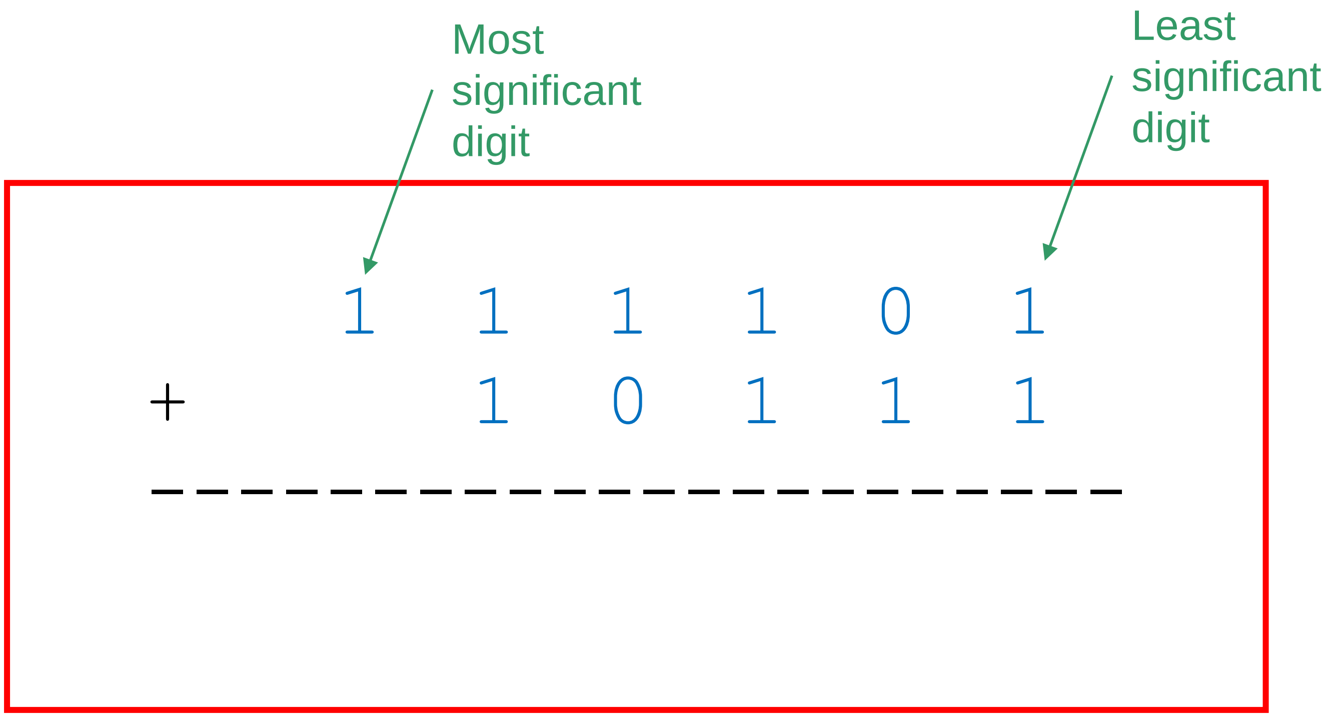
- Binary addition is similar to decimal addition.

- Adding 2 binary numbers:

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

Binary Addition

- Adding 2 binary numbers: 111101 + 10111



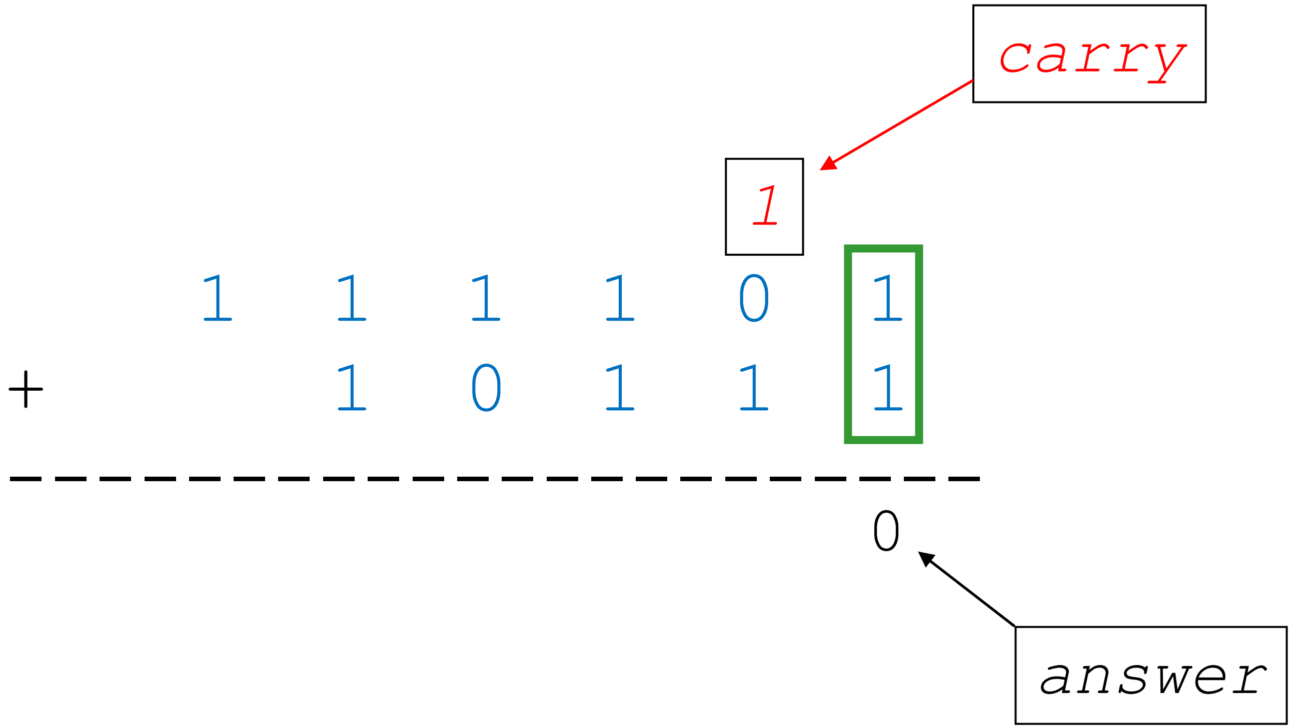
Binary Addition

- Adding 2 binary numbers: 111101 + 10111

```
        1   1   1   1   0  [1]
+           1   0   1   1  [1]
----------------------------
                            0
```

Binary Addition

- Adding 2 binary numbers: 111101 + 10111



Binary Addition

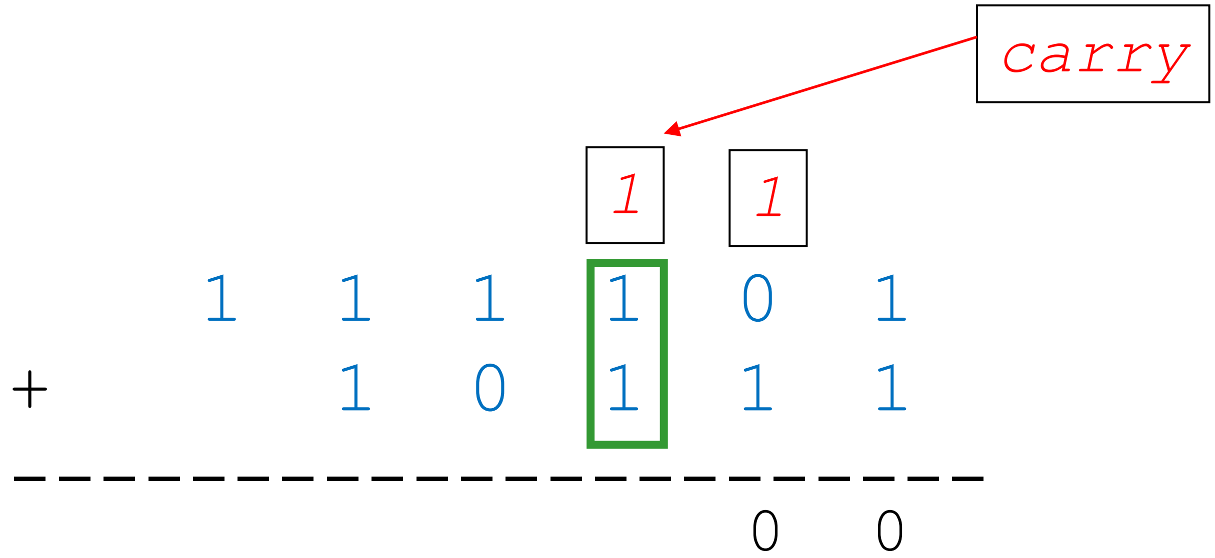
- Adding 2 binary numbers: 111101 + 10111

carry

				1	*1*	
	1	1	1	1	0	1
+		1	0	1	1	1
					0	0

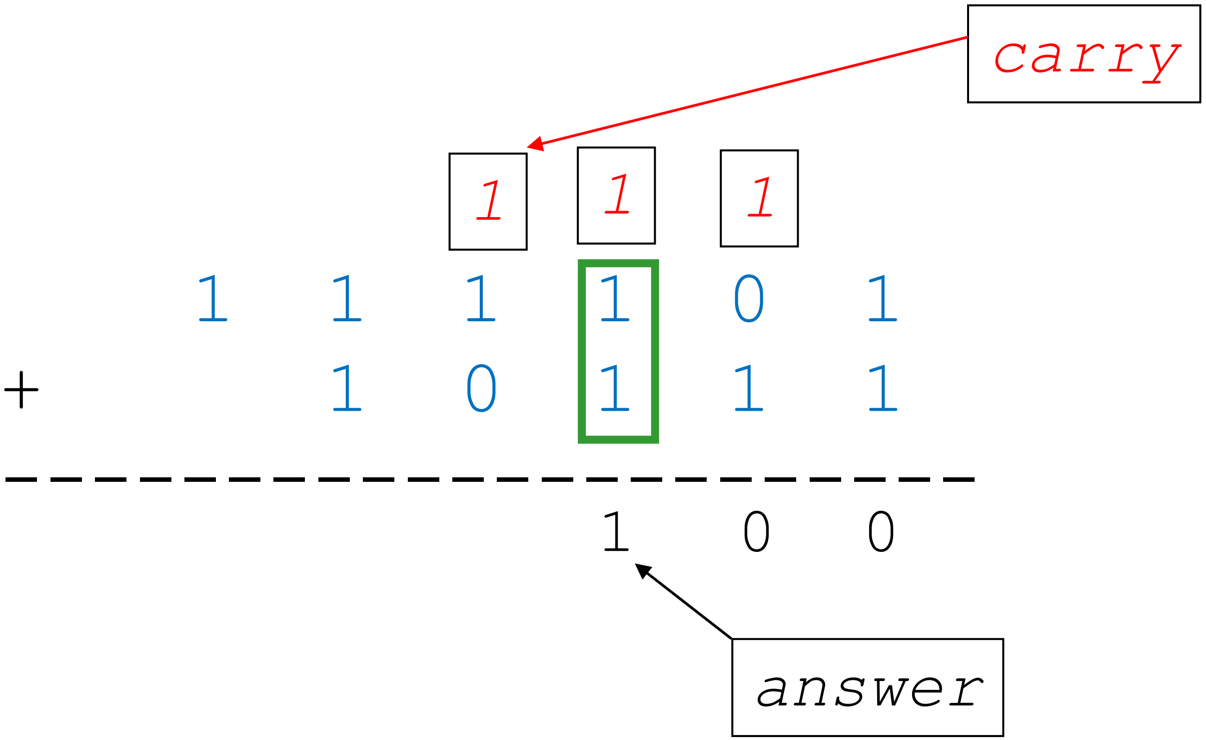
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



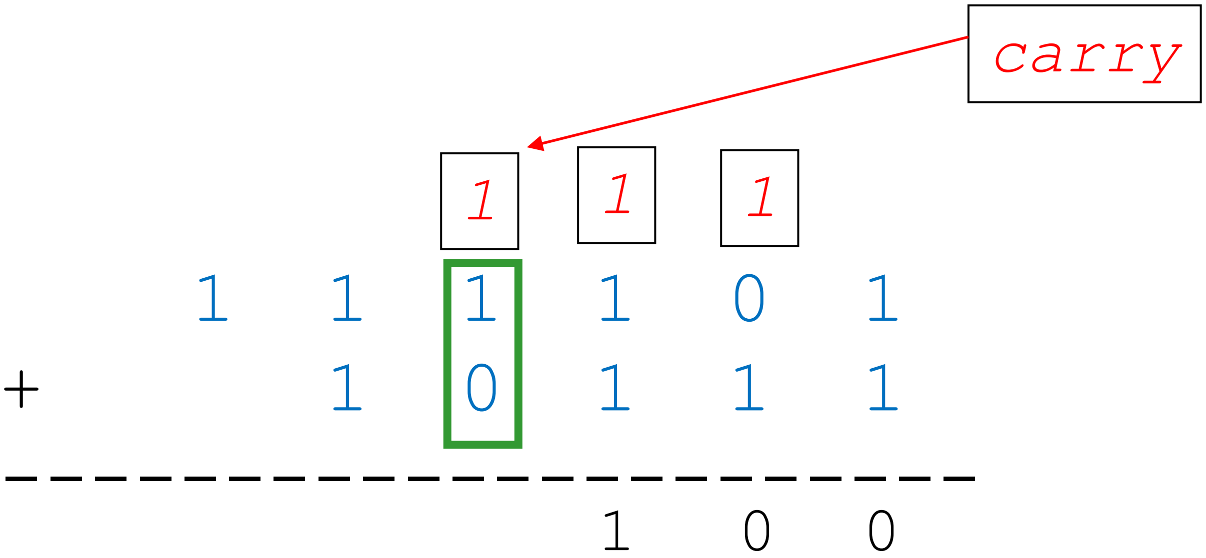
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



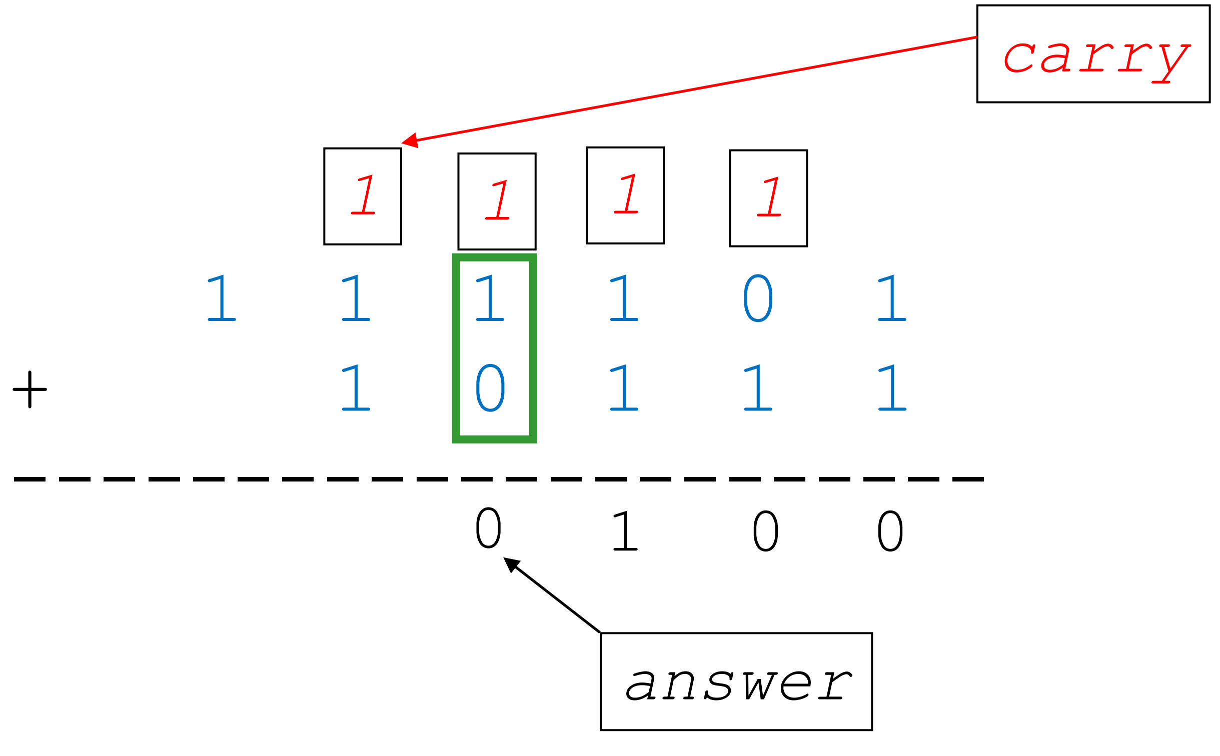
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



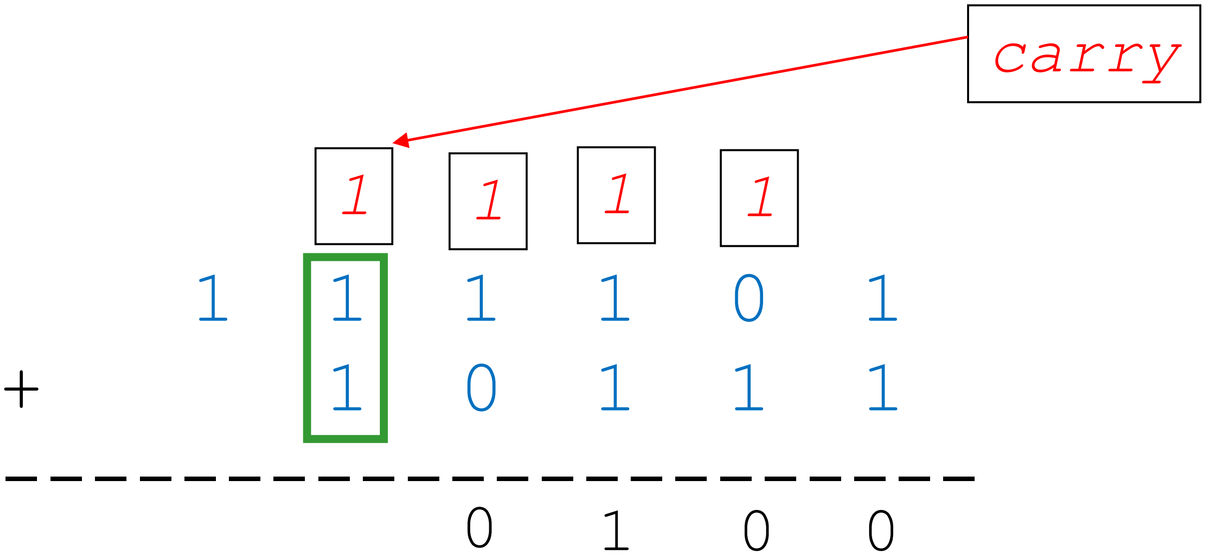
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



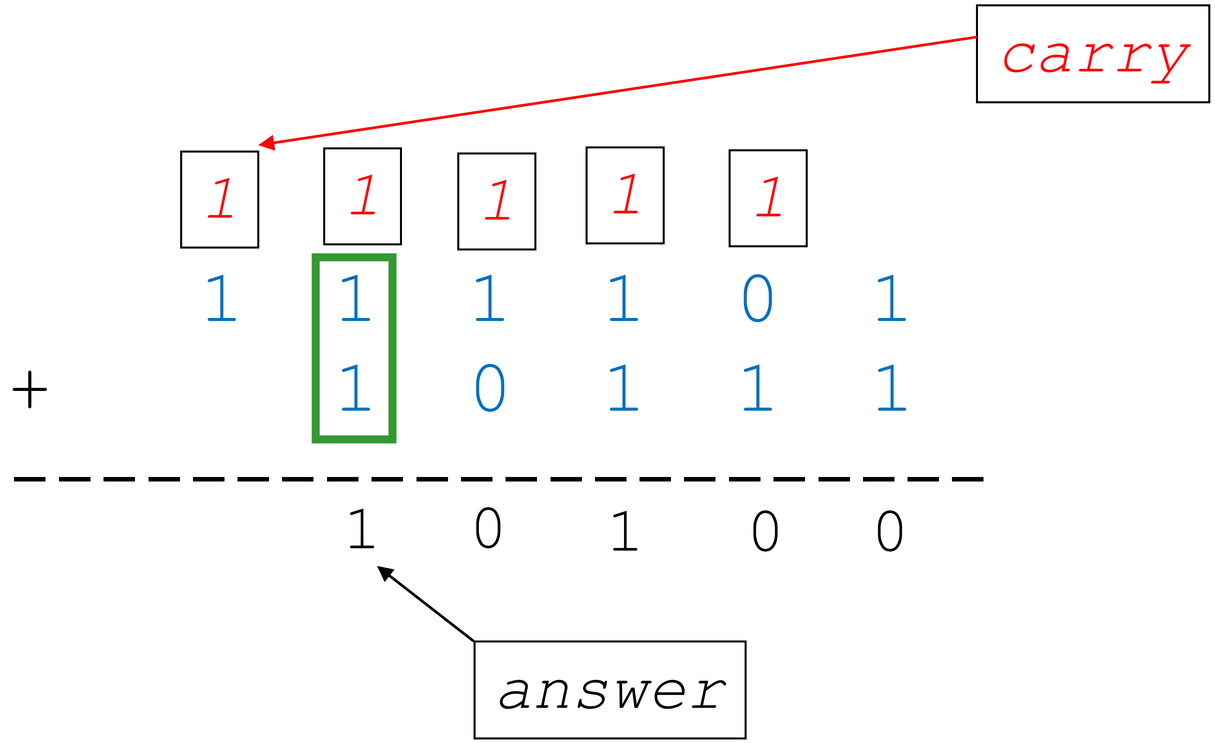
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



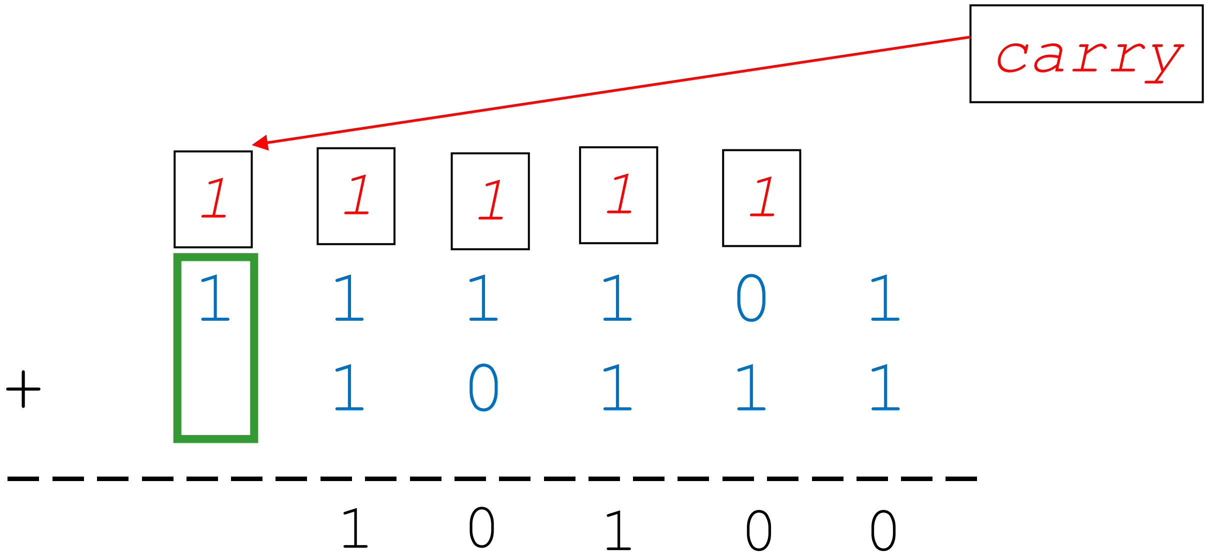
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



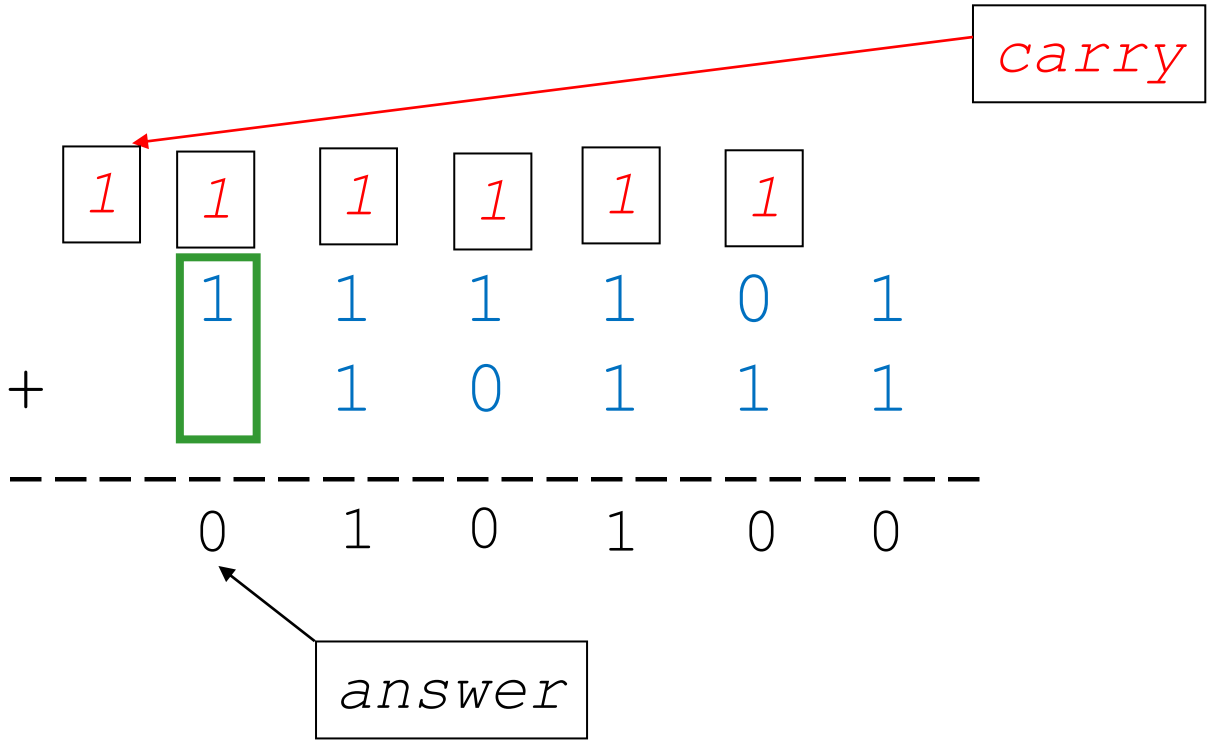
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



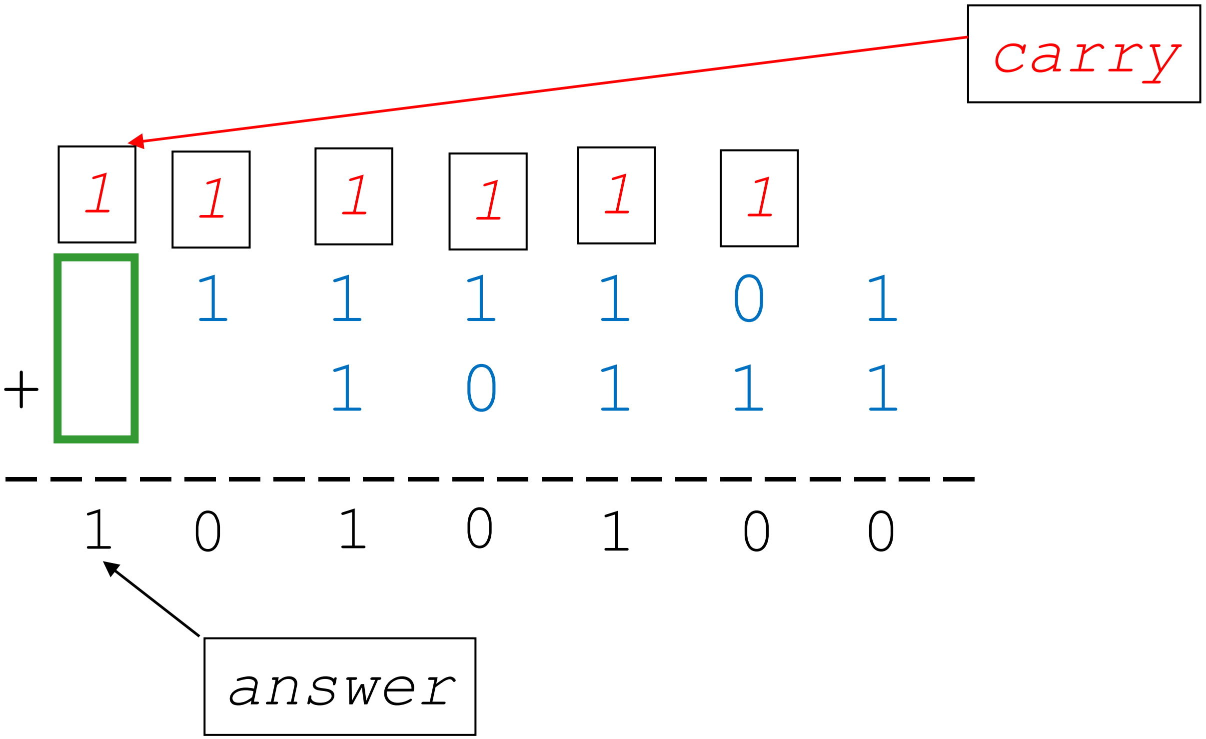
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



Binary Addition

- Adding 2 binary numbers: 111101 + 10111



Binary Addition

- Adding 2 binary numbers: 111101 + 10111

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

Is this correct?

Binary Addition

- Adding 2 binary numbers: 111101 + 10111

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

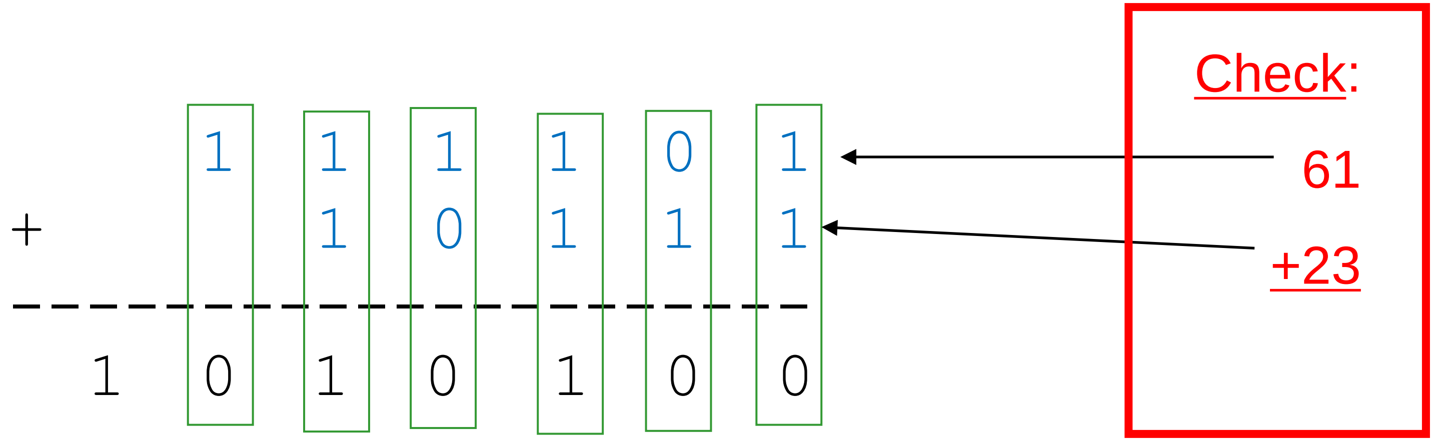
		1	1	1	1	0	1
+			1	0	1	1	1
	1	0	1	0	1	0	0

Check:

61

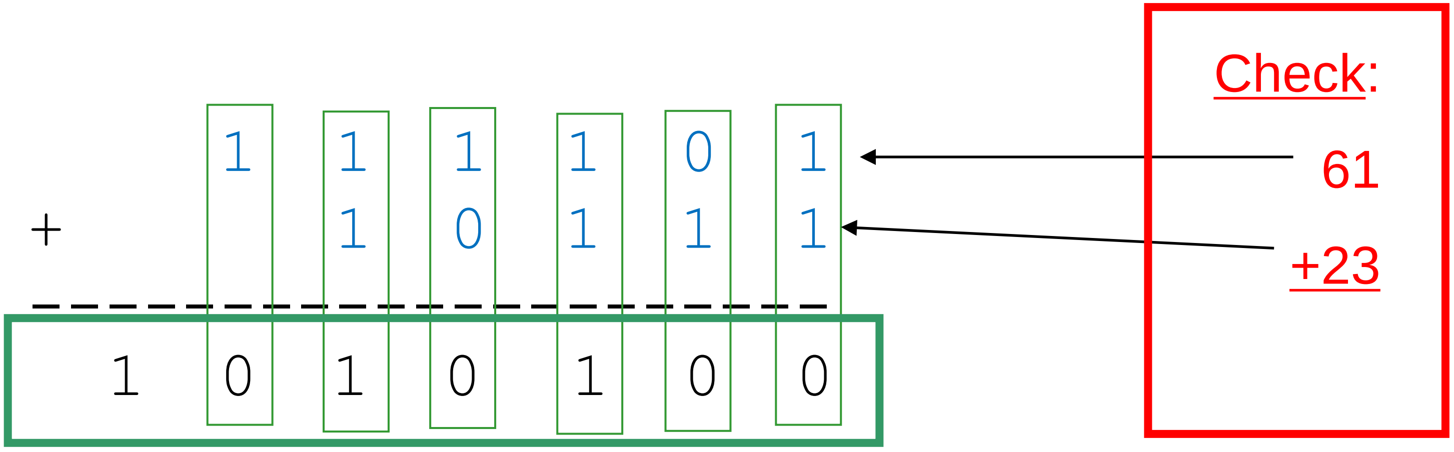
Binary Addition

- Adding 2 binary numbers: 111101 + 10111



Binary Addition

- Adding 2 binary numbers: 111101 + 10111



Binary Addition

- Adding 2 binary numbers: 111101 + 10111

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

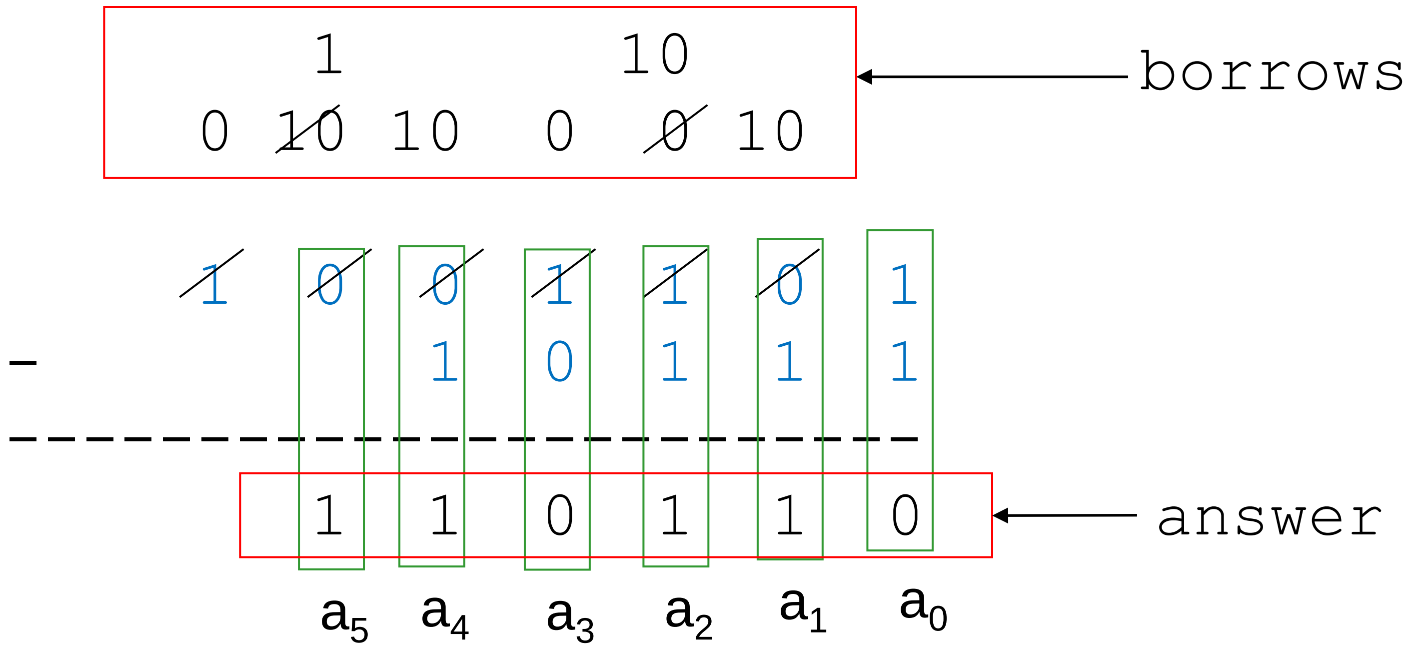
		1	1	1	1	0	1
+			1	0	1	1	1
	1	0	1	0	1	0	0

Check:

61
+23
84

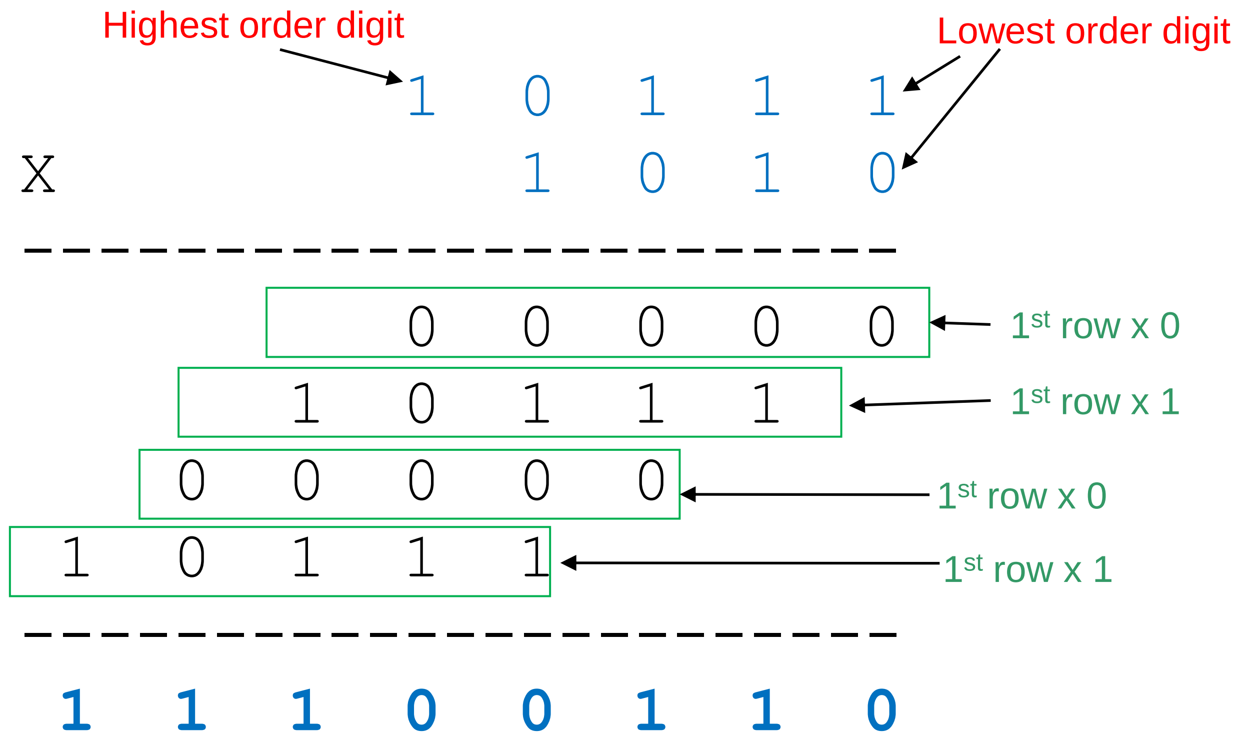
Binary Subtraction

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

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

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

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:

8 bits

Decimal number 12

$$0\ 0\ 0\ 0\ 1\ 1\ 0\ 0_2 = 12_{10}$$

Sign bit Magnitude

1) Signed magnitude numbers

Consider ***signed magnitude***:

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

Sign bit Magnitude

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

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.

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

Sign bit   Magnitude

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

Sign bit   Magnitude

2) One's Complement Representation

- The one's complement representation of an n-bit binary number can represent numbers in the range of $-(2^{N-1}-1)$ to $2^{N-1}-1$

- Example: an 8-bit binary number 1000 0000 represents

  128 if the system is unsigned OR

  -127 if the system is ones' complement

  because it is the negative of 0111 1111 =127

- Try it: express the following numbers using ones' complement representation:
  - $(-17)_{10}$
  - $(-32)_{10}$
  - $(-255)_{10}$

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.

$00001100_2 = 12_{10}$

Sign bit Magnitude

$11110100_2 = -12_{10}$

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…

N = 01100101

Invert all bits in N

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

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

```
  10011010
+        1
----------------
  10011011
```

[N] = 10011011

```
 01100100
+        1
---------------
 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
```