Binary and Hexadecimal Numbers

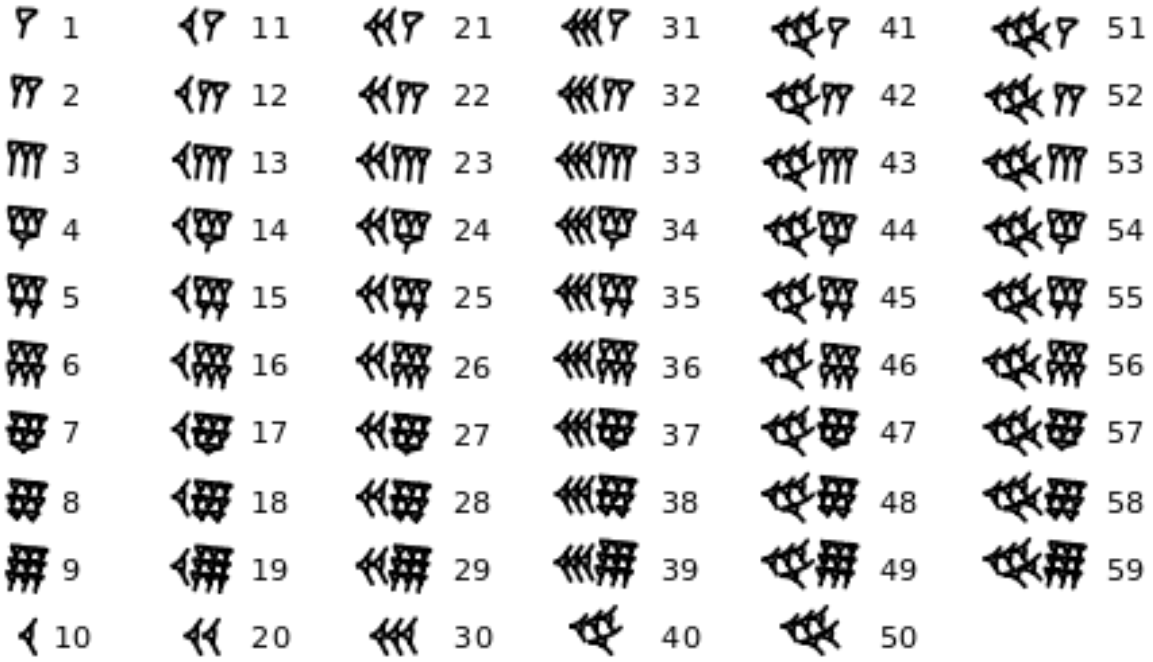
CSE 1310 – Introduction to Computers and Programming
Vassilis Athitsos
University of Texas at Arlington

Decimal Numbers

- A decimal number is an integer written the way we are used to write numbers, using digits from 0 to 9.
- Why do we use 10 digits, and not 2, or 20, or 60, or 150?
- It is a result of:
  - Having 10 fingers in our hands, which makes it easier to count up to 10.
  - Cultural convention.

Babylonian Numbers

- Babylonians used 60 digits instead of 10.
  - Make sure you know them for the exam :)



Babylonian Numbers

- Babylonians used 60 digits instead of 10.
  - Make sure you know them for the exam :)
  - In case you are reading this at home: the line above is a joke!

Babylonian Numbers

- Babylonians used 60 digits instead of 10.
- While we do not use that system for writing numbers today, it has survived in our modern culture. Where?
- In the way we measure time.
  - 24 hours.
  - 60 minutes.
  - 60 seconds.
- Also in the way we measure angles.
  - 360 degrees.
  - 60 minutes.
  - 60 seconds.

Binary Numbers

- Computers, in their electronic circuits, essentially use two digits: 1 and 0.
- 1 corresponds to a certain electrical (or magnetic, or optical) signal.
- 0 corresponds to another electrical (or magnetic, or optical) signal.
- This means that we need to represent all numbers using those two digits.
  - Numbers represented using 1 and 0 are called **binary numbers.**
- Java (like any modern programming language) allows us to use decimal numbers.
- Inside the computer, before the program gets executed, all decimal numbers are translated to binary.

Binary Numbers

- Since 1310 is our introductory course, it is a good place for you to learn some basics about binary numbers.
- We will do a simplified version.
- We will not worry about representing:
  - Floating point numbers.
  - Negative numbers.
- We will learn how to represent positive integers in binary.
- We will learn how to translate from binary to decimal and the other way around.

Reading a Binary Number

- In general, a binary number is represented like this:

  $b_{n-1}\ b_{n-2}\ \ldots\ b_2\ b_1\ b_0$

- Each digit b_i is either 1 or 0.
- For example, consider binary number 10011.
- What is n?
- What is b_0?
- What is b_1?
- What is b_2?
- What is b_3?
- What is b_4?

Reading a Binary Number

- In general, a binary number is represented like this:

  $b_{n-1}\ b_{n-2} \ldots b_2\ b_1\ b_0$

- Each digit b_i is either 1 or 0.
- For example, consider binary number 10011.
- What is n? 5
- What is b_0? 1
- What is b_1? 1
- What is b_2? 0
- What is b_3? 0
- What is b_4? 1

Reading a Binary Number

- In general, a binary number is represented like this:

  $b_{n-1}\ b_{n-2}\ \ldots\ b_2\ b_1\ b_0$

- This binary number can be translated to decimal using this formula:

$b_{n-1}*2^{n-1} + b_{n-2}*2^{n-2} + \ldots + b_2 * 2^2 + b_1 * 2^1 + b_0 * 2^0.$

- As a quick review:
  - What is 2^0?
  - What is 2^1?
  - What is 2^2?
  - What is 2^3?

Reading a Binary Number

- In general, a binary number is represented like this:

  $b_{n-1}\ b_{n-2}\ \ldots\ b_2\ b_1\ b_0$

- This binary number can be translated to decimal using this formula:

$b_{n-1}*2^{n-1} + b_{n-2}*2^{n-2} + \ldots + b_2 * 2^2 + b_1 * 2^1 + b_0 * 2^0.$

- As a quick review:
  - What is 2^0? 1
  - What is 2^1? 2
  - What is 2^2? 4
  - What is 2^3? 8

Translating from Binary to Decimal

- In general, a binary number is represented like this:

  $b_{n-1}\ b_{n-2}\ \ldots\ b_2\ b_1\ b_0$

- This binary number can be translated to decimal using this formula:

$b_{n-1}*2^{n-1} + b_{n-2}*2^{n-2} + \ldots + b_2 * 2^2 + b_1 * 2^1 + b_0 * 2^0.$

- So, how do we translate binary number 10011 to decimal?

Translating from Binary to Decimal

- In general, a binary number is represented like this:

  $b_{n-1}\ b_{n-2}\ \ldots\ b_2\ b_1\ b_0$

- This binary number can be translated to decimal using this formula:

$b_{n-1}*2^{n-1} + b_{n-2}*2^{n-2} + \ldots + b_2 * 2^2 + b_1 * 2^1 + b_0 * 2^0.$

- So, how do we translate binary number 10011 to decimal?

  $1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 16 + 2 + 1 = 19.$

Translating from Binary to Decimal

- In general, a binary number is represented like this:

  $b_{n-1}\ b_{n-2}\ \ldots\ b_2\ b_1\ b_0$

- This binary number can be translated to decimal using this formula:

$b_{n-1}*2^{n-1} + b_{n-2}*2^{n-2} + \ldots + b_2 * 2^2 + b_1 * 2^1 + b_0 * 2^0.$

- How do we translate binary number 101000 to decimal?

Translating from Binary to Decimal

- In general, a binary number is represented like this:

  $b_{n-1}\ b_{n-2}\ \ldots\ b_2\ b_1\ b_0$

- This binary number can be translated to decimal using this formula:

$b_{n-1}*2^{n-1} + b_{n-2}*2^{n-2} + \ldots + b_2 * 2^2 + b_1 * 2^1 + b_0 * 2^0.$

- How do we translate binary number 101000 to decimal?

  $1*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 0*2^0 = 32 + 8 = 40.$

binaryToDecimal Function

- Let's write a function binaryToDecimal that translates a binary number into a decimal number.
- What arguments should the function take?
- What type should the function return?

binaryToDecimal Function

- Let's write a function binaryToDecimal that translates a binary number into a decimal number.
- What arguments should the function take?
  - The binary number can be represented as a string.
  - It can also be represented as an int.
  - We will go with the string choice.
- What type should the function return?
  - The decimal number should be an int.

binaryToDecimal Function

```
public static int binaryToDecimal(String text)
{
   int result = 0;
   for (int i = 0; i < text.length(); i++)
   {
     String c = text.substring(i, i+1);
     if ("01".indexOf(c) == -1)
     {
       System.out.printf("Error: invalid binary number %s,
  exiting...\n", text);
       System.exit(0);
     }
     int digit = Integer.parseInt(c);
     int power_of_2 = (int) (Math.pow(2, text.length() - i - 1));
     result = result + digit * power_of_2;
   }
   return result;
}
```

Translating from Decimal to Binary

- How can we translate a decimal number, such as 37, to binary?

Translating from Decimal to Binary

- How can we translate a decimal number, such as 37, to binary?
- Answer: we repeatedly divide by 2, until the result is 0, and use the remainder as a digit.
- 37 / 2 = 18, with remainder ???

Translating from Decimal to Binary

- How can we translate a decimal number, such as 37, to binary?
- Answer: we repeatedly divide by 2, until the result is 0, and use the remainder as a digit.
- 37 / 2 = 18, with remainder 1. So, b_0 is 1.
- 18 / 2 = 9, with remainder ???

Translating from Decimal to Binary

- How can we translate a decimal number, such as 37, to binary?
- Answer: we repeatedly divide by 2, until the result is 0, and use the remainder as a digit.
- 37 / 2 = 18, with remainder 1. So, b_0 is 1.
- 18 / 2 = 9, with remainder 0. So, b_1 is 0.
- 9 / 2 = 4, with remainder ???

Translating from Decimal to Binary

- How can we translate a decimal number, such as 37, to binary?
- Answer: we repeatedly divide by 2, until the result is 0, and use the remainder as a digit.
- 37 / 2 = 18, with remainder 1. So, b_0 is 1.
- 18 / 2 = 9, with remainder 0. So, b_1 is 0.
- 9 / 2 = 4, with remainder 1. So, b_2 is 1.
- 4 / 2 = 2, with remainder ???

Translating from Decimal to Binary

- How can we translate a decimal number, such as 37, to binary?
- Answer: we repeatedly divide by 2, until the result is 0, and use the remainder as a digit.
- 37 / 2 = 18, with remainder 1. So, b_0 is 1.
- 18 / 2 = 9, with remainder 0. So, b_1 is 0.
- 9 / 2 = 4, with remainder 1. So, b_2 is 1.
- 4 / 2 = 2, with remainder 0. So, b_3 is 0.
- 2 / 2 = 1, with remainder ???

Translating from Decimal to Binary

- How can we translate a decimal number, such as 37, to binary?
- Answer: we repeatedly divide by 2, until the result is 0, and use the remainder as a digit.
- 37 / 2 = 18, with remainder 1. So, b_0 is 1.
- 18 / 2 = 9, with remainder 0. So, b_1 is 0.
- 9 / 2 = 4, with remainder 1. So, b_2 is 1.
- 4 / 2 = 2, with remainder 0. So, b_3 is 0.
- 2 / 2 = 1, with remainder 0. So, b_4 is 0.
- 1 / 2 = 0, with remainder ???

Translating from Decimal to Binary

- How can we translate a decimal number, such as 37, to binary?
- Answer: we repeatedly divide by 2, until the result is 0, and use the remainder as a digit.
- 37 / 2 = 18, with remainder 1. So, b_0 is 1.
- 18 / 2 = 9, with remainder 0. So, b_1 is 0.
- 9 / 2 = 4, with remainder 1. So, b_2 is 1.
- 4 / 2 = 2, with remainder 0. So, b_3 is 0.
- 2 / 2 = 1, with remainder 0. So, b_4 is 0.
- 1 / 2 = 0, with remainder 1. So, b_5 is 1.
- So, 37 in binary is ???

Translating from Decimal to Binary

- How can we translate a decimal number, such as 37, to binary?
- Answer: we repeatedly divide by 2, until the result is 0, and use the remainder as a digit.
- 37 / 2 = 18, with remainder 1. So, b_0 is 1.
- 18 / 2 = 9, with remainder 0. So, b_1 is 0.
- 9 / 2 = 4, with remainder 1. So, b_2 is 1.
- 4 / 2 = 2, with remainder 0. So, b_3 is 0.
- 2 / 2 = 1, with remainder 0. So, b_4 is 0.
- 1 / 2 = 0, with remainder 1. So, b_5 is 1.
- So, 37 in binary is 100101.

decimalToBinary Function

- Let's write a function decimalToBinary that translates a decimal number into a binary number.
- What arguments should the function take?

- What type should the function return?

decimalToBinary Function

- Let's write a function decimalToBinary that translates a decimal number into a binary number.
- What arguments should the function take?
  - The decimal number should be an int.
- What type should the function return?
  - Again, the most simple choice is to represent the binary number as a string.

decimalToBinary Function

```
public static String decimalToBinary(int number)
{
   String result = "";
   while(true)
   {
     int remainder = number % 2;
     String digit = Integer.toString(remainder);
     result = digit + result;
     number = number / 2;
     if (number == 0)
     {
       break;
     }
   }
   return result;
}
```

Hexadecimal Numbers

- Another very popular way to represent numbers among programmers is **hexadecimal numbers**.
- Hexadecimal numbers use 16 digits:

0123456789abcdef

- a stands for 10.
- b stands for 11.
- c stands for 12.
- d stands for 13.
- e stands for 14.
- f stands for 15.

Reading a Hexadecimal Number

- A hexadecimal number is represented like this:

  $h_{n-1}\ h_{n-2}\ \ldots\ h_2\ h_1\ h_0$

- Each digit h_i is one of the sixteen digits:
  - 0123456789abcdef.
- For example, consider hexadecimal number 5b7.
- What is n?
- What is h_0?
- What is h_1?
- What is h_2?

Reading a Hexadecimal Number

- A hexadecimal number is represented like this:

  $h_{n-1}\ h_{n-2}\ \ldots\ h_2\ h_1\ h_0$

- Each digit h_i is one of the sixteen digits:
  - 0123456789abcdef.
- For example, consider hexadecimal number 5b7.
- What is n? 3
- What is h_0? 7
- What is h_1? b
- What is h_2? 5

Reading a Hexadecimal Number

- A hexadecimal number is represented like this:

  $h_{n-1}\ h_{n-2}\ \ldots\ h_2\ h_1\ h_0$

- This hexadecimal number can be translated to decimal using this formula:

$b_{n-1}*16^{n-1} + b_{n-2}*16^{n-2} + \ldots + b_2 * 16^2 + b_1 * 16^1 + b_0*16^0.$

- Note: if b_i is one of abcdef, use the corresponding number (10, 11, 12, 13, 14, 15) when you plug it into the formula.
- This looks like the formula for binary numbers, except that here we use powers of 16 instead of powers of 2.

From Hexadecimal to Decimal

- A hexadecimal number is represented like this:

  $h_{n-1}\ h_{n-2}\ \ldots\ h_2\ h_1\ h_0$

- This hexadecimal number can be translated to decimal using this formula:

$b_{n-1}*16^{n-1} + b_{n-2}*16^{n-2} + \ldots + b_2 * 16^2 + b_1 * 16^1 + b_0*16^0.$

- So, how do we translate hexadecimal number 5b7 to decimal?

From Hexadecimal to Decimal

- A hexadecimal number is represented like this:

  $h_{n-1}\ h_{n-2} \ldots h_2\ h_1\ h_0$

- This hexadecimal number can be translated to decimal using this formula:

$b_{n-1}*16^{n-1} + b_{n-2}*16^{n-2} + \ldots + b_2 * 16^2 + b_1 * 16^1 + b_0*16^0.$

- So, how do we translate hexadecimal number 5b7 to decimal? (note that we plug in 11 for b).

  $5*16^2 + 11*16^1 + 7*16^0 = 5*256 + 11*16 + 7 = 1463.$

From Hexadecimal to Decimal

- A hexadecimal number is represented like this:

  $h_{n-1}\ h_{n-2}\ \ldots\ h_2\ h_1\ h_0$

- This hexadecimal number can be translated to decimal using this formula:

$b_{n-1}*16^{n-1} + b_{n-2}*16^{n-2} + \ldots + b_2 * 16^2 + b_1 * 16^1 + b_0*16^0.$

- So, how do we translate hexadecimal number **bad** to decimal?

From Hexadecimal to Decimal

- A hexadecimal number is represented like this:

  $h_{n-1}\ h_{n-2}\ \ldots\ h_2\ h_1\ h_0$

- This hexadecimal number can be translated to decimal using this formula:

$b_{n-1}*16^{n-1} + b_{n-2}*16^{n-2} + \ldots + b_2 * 16^2 + b_1 * 16^1 + b_0*16^0.$

- So, how do we translate hexadecimal number **bad** to decimal?

  $11*16^2 + 10*16^1 + 13*16^0 = 11*256 + 10*16 + 13 = 2989.$

hexToDecimal Function

- Let's write a function hexToDecimal that translates a hexadecimal number into a decimal number.
- What arguments should the function take?
- What type should the function return?

hexToDecimal Function

- Let's write a function binaryToDecimal that translates a binary number into a decimal number.
- What arguments should the function take?
  - The hexadecimal number can be represented as a string.
- What type should the function return?
  - The decimal number should be an int.

hexToDecimal Function

```
public static String decimalToHex(int number)
{
   String result = "";
   String digits = "0123456789abcdef";

   while(true)
   {
     int remainder = number % 16;
     String digit = digits.substring(remainder, remainder+1);
     result = digit + result;
     number = number / 16;
     if (number == 0)
     {
       break;
     }
   }
   return result;
}
```

From Decimal to Hexadecimal

- How can we translate a decimal number, such as 540, to hexadecimal?

From Decimal to Hexadecimal

- How can we translate a decimal number, such as 540, to hexadecimal?
- Answer: we repeatedly divide by 16, until the result is 0, and use the remainder as a digit.
- If the remainder is 10, 11, 12, 13, 14, or 15, we use respectively digit a, b, c, d, e, or f.
- 540 / 16 = 33, with remainder 12. So, h_0 is ???

From Decimal to Hexadecimal

- How can we translate a decimal number, such as 540, to hexadecimal?
- Answer: we repeatedly divide by 16, until the result is 0, and use the remainder as a digit.
- If the remainder is 10, 11, 12, 13, 14, or 15, we use respectively digit a, b, c, d, e, or f.
- 540 / 16 = 33, with remainder 12. So, h_0 is c.
- 33 / 16 = ??? with remainder ???

From Decimal to Hexadecimal

- How can we translate a decimal number, such as 540, to hexadecimal?
- Answer: we repeatedly divide by 16, until the result is 0, and use the remainder as a digit.
- If the remainder is 10, 11, 12, 13, 14, or 15, we use respectively digit a, b, c, d, e, or f.
- 540 / 16 = 33, with remainder 12. So, h_0 is c.
- 33 / 16 = 2 with remainder 1. So, h_1 is 1.
- 2 / 16 = ??? with remainder ???

From Decimal to Hexadecimal

- How can we translate a decimal number, such as 540, to hexadecimal?
- Answer: we repeatedly divide by 16, until the result is 0, and use the remainder as a digit.
- If the remainder is 10, 11, 12, 13, 14, or 15, we use respectively digit a, b, c, d, e, or f.
- 540 / 16 = 33, with remainder 12. So, h_0 is c.
- 33 / 16 = 2 with remainder 1. So, h_1 is 1.
- 2 / 16 = 0 with remainder 2. So, h_2 is 2.
- So, 540 in hexadecimal is ???

From Decimal to Hexadecimal

- How can we translate a decimal number, such as 540, to hexadecimal?
- Answer: we repeatedly divide by 16, until the result is 0, and use the remainder as a digit.
- If the remainder is 10, 11, 12, 13, 14, or 15, we use respectively digit a, b, c, d, e, or f.
- 540 / 16 = 33, with remainder 12. So, h_0 is c.
- 33 / 16 = 2 with remainder 1. So, h_1 is 1.
- 2 / 16 = 0 with remainder 2. So, h_2 is 2.
- So, 540 in hexadecimal is 21c.

decimalToHexadecimal Function

- Let's write a function decimalToHexadecimal that translates a decimal number into a binary number.
- What arguments should the function take?

- What type should the function return?

decimalToHexadecimal Function

- Let's write a function decimalToHexadecimal that translates a decimal number into a binary number.
- What arguments should the function take?
  - The decimal number should be an int.
- What type should the function return?
  - The hexadecimal number should be a string.

decimalToHexadecimal Function

```
public static String decimalToHexadecimal(int number)
{
   String result = "";
   String digits = "0123456789abcdef";

   while(true)
   {
     int remainder = number % 16;
     String digit = digits.substring(remainder, remainder+1);
     result = digit + result;
     number = number / 16;
     if (number == 0)
     {
       break;
     }
   }
   return result;
}
```