Base Conversions

Computer Science Department
University of Central Florida

COP 3502 – Computer Science I

Counting Systems – Basic Info

- Regular Counting System
  - Known as Decimal
  - also known as base 10
  - Do you know why it is called base 10?
    - If you said, "because it has ten counting digits":
      - 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9
    - You are right!
    - To count in base ten, you go from 0 to 9
    - Then you count in combinations of two digits starting with 10 all the way to 99
    - After 99 comes three-digit combinations from 100 – 999, etc.

Counting Systems – Basic Info

- Regular Counting System
  - Let's examine a decimal number:

2713

Thousands place | Hundreds place | Tens place | Ones place

  - When we break down this number, we have:
    - 2 "thousands" + 7 "hundreds" + 1 "tens" + 3 "ones
    - 2000 + 700 + 10 + 3
  - Let's see, in detail, how we get this

Counting Systems – Basic Info

- Regular Counting System
  - The decimal number 2713:
  - When we break down this number, we have:
    - 2000 + 700 + 10 + 3
  - Where does the 2000 come from?
    - How do we get 2000?
  - Mathematically,
    - We said this means we have two "thousands"
    - A thousand is 1000
    - How do we represent 1000, in terms of 10? 10^3
    - So 2000 is the same as $2 \times 10^3 = 2 \times 1000 = 2000$

Counting Systems – Basic Info

- Regular Counting System
  - The decimal number 2713:
  - Similarly,
    - The next digit, 7, means that we have 7 “hundreds”
      - We have 7, “100”s
    - Mathematically, how do we represent 100 in terms of 10?
      - 10^2
    - So 700 comes from $7 \times 10^2 = 7 \times 100 = 700$

Counting Systems – Basic Info

- Regular Counting System
  - The decimal number 2713:
  - Next:
    - The next digit, 1, means that we have 1 "ten"
      - We have 1, "10"
      - Mathematically, we represent this as 10^1
    - So 10 comes from $1 \times 10^1 = 1 \times 10 = 10$
  - Finally:
    - The last digit, 3, means that we have 3 "ones"
      - We have 3, "1"s
      - How do we represent 1 in terms of 10? As 10^0.
    - So 3 comes from $3 \times 10^0 = 3 \times 1 = 3$

Counting Systems – Basic Info

- Regular Counting System
  - The decimal number 2713:
  - Putting this all together,
    - $2713_{10} = 2 \times 10^3 + 7 \times 10^2 + 1 \times 10^1 + 3 \times 10^0$
  - What we learn from this:
    - Each <u>digit's value is determined by the place it is in</u>
    - Each <u>place</u> is a perfect power of the base
    - With the least significant at the end
    - Counting up, by 1, as you go through the number from right to left

Counting Systems – Basic Info

- Other Counting Systems
  - At first glance, it may seem that this would be the only possible number system
    - That is, using 10 digits (0 – 9)
  - Turns out, the number of digits used is arbitrary
  - We could have chosen to use only 5 digits
    - 0 – 4 (base 5 system)
  - Look at how we determine the value of a number:
    - $314_5 = 3 \times 5^2 + 1 \times 5^1 + 4 \times 5^0 = 84_{10}$
  - Guess what???
    - We just converted from base 5 to base 10

Counting Systems – Basic Info

- **<u>CONVERT from ANY base to base 10</u>**
  - This example illustrates how we can convert from a different base to base 10
  - In general, we write the conversion as follows:
    - $d_{n-1}d_{n-2}\ldots d_2d_1d_{0\text{ (in base b)}} = d_{n-1}xb^{n-1} + d_{n-2}xb^{n-2} + \ldots + d_2xb^2 + d_1xb + d_0$
  - Note:
    - b based to the 1 and 0 powers were simplified above
  - Couple quick examples:
    - $781_9 = 7x9^2 + 8x9^1 + 1x9^0 = 640_{10}$
    - $1110101_2 = 1x2^6 + 1x2^5 + 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 117_{10}$
    - This last one was the very common base 2 (binary)

Counting Systems – Basic Info

- Binary (aka base 2)
  - MOST common in computer science
  - Why?
    - Cuz all your computer "innards" are represented in binary
      - All software ultimately boils down to a binary representation
  - So here's a little binary chart to get you going:

Decimal	Binary	Decimal	Binary
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111
8	1000	16	10000

Counting Systems – Basic Info

- Hexadecimal
  - The most common base with more than 10 digits
    - Aka base 16
    - Meaning there are **<u>16 counting digits</u>**
    - WAIT!!!
    - But we <u>only</u> have 10 possible digits to use!
      - 0 through 9
    - So that means we are six digits short!
      - That is correct.
    - It was decided to use the following six additional "digits":
      - A, B, C, D, E, and F

Counting Systems – Basic Info

- Hexadecimal
  - base 16: use 16 counting digits
    - It was decided to use the following six additional “digits”:
      - A, B, C, D, E, and F
    - **A** represents the value 10, **B** is 11, **C** is 12, **D** is 13, **E** is 14, and **F** is 15
    - So here is the single digit sequence for base 16:
      - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Counting Systems – Basic Info

- Hexadecimal
  - Benefit of Hexadecimal:
    - Everything internally (in a computer) is stored in base 2
      - binary
    - However, when we view contents of memory
    - Or when we assign values
      - Such as RGB values for colors
    - We often view numbers in hexadecimal
  - So it is important to be familiar with hexadecimal
  - Also important to be able to convert to and from hexadecimal to other bases

Base Conversion Methods

- Conversion from Hexadecimal to Decimal
  - This is done EXACTLY the same as shown previously
  - $A3D_{16} = \text{A}x16^2 + 3x16^1 + \text{D}x16^0$

    $= 10x16^2 + 3x16^1 + 13x16^0 = 2621_{10}.$

    $= 2621_{10}.$

Base Conversion Methods

- Conversion from Hexadecimal to Binary
  - Note:
    - 16, as in "base 16", is a PERFECT power of 2
    - This makes conversion to base 2 (binary) very EASY
    - Why?
    - Each hexadecimal digit is perfectly represented by 4 binary digits
    - Does that make sense?
    - A base 16 digit can be up to F (which is 15)
    - So, in order to represent, up to 15, in binary
      - We MUST have 4 binary digits
      - From the chart earlier, we know that 15_{10} is 1111_2

Base Conversion Methods

- Conversion from Hexadecimal to Binary
  - Note:
    - This allows us to make the following “purty” chart showing the conversions from hexadecimal to binary:

Hex:	0	1	2	3	4	5	6	7
Bin:	0000	0001	0010	0011	0100	0101	0110	0111
Hex:	8	9	A	B	C	D	E	F
Bin:	1000	1001	1010	1011	1100	1101	1110	1111

    - Using this, we can easily convert from base 16 to base 2
    - $A3D_{16} = 1010\ 0011\ 1101_2$
    - $F4BC72_{16} = 1111\ 0100\ 1011\ 1100\ 0111\ 0010\ 0001\ 0110_2$

Base Conversion Methods

- **<u>CONVERT from ANY base to base 10</u>**
  - We already went over this one
  - In general, the conversion is as follows:
    - $d_{n-1}d_{n-2}\ldots d_2d_1d_{0\text{ (in base b)}} = d_{n-1}xb^{n-1} + d_{n-2}xb^{n-2} + \ldots + d_2xb^2 + d_1xb^1 + d_0xb^0$
  - Some quick examples:
    - $246_7 = 2x7^2 + 4x7^1 + 6x7^0 = 132_{10}$
    - $781_9 = 7x9^2 + 8x9^1 + 1x9^0 = 640_{10}$
    - $30122_4 = 3x4^4 + 0x4^3 + 1x4^2 + 2x4^1 + 2x4^0 = 794_{10}$
    - $1110101_2 = 1x2^6 + 1x2^5 + 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 117_{10}$
    - This last one was the very common base 2 (binary)

Base Conversion Methods

- **Conversion from Decimal to Binary**
  - Given the number 27_{10}
    - Convert it to binary
  - Basically, we start by dividing 27 by 2
    - Integer Division!
      - Remember, 27/2 would be 13
    - So, 27/2 is 13 with a remainder of 1
  - We then divide 13 by 2
    - 13/2 is 6 with a remainder of 1
  - Continue this process until you get 1
    - At that point, you will have 1/2 is 0 with a remainder of 1

Base Conversion Methods

- Conversion from Decimal to Binary
  - Convert 27_{10} to binary

27/2 = 13	with a remainder of	1
13/2 = 6	with a remainder of	1
6/2 = 3	with a remainder of	0
3/2 = 1	with a remainder of	1
1/2 = 0	with a remainder of	1

So, 27_{10} is the same as 11011_2

- You stop when you get 0 as an answer
  - Of course, the final remainder will be 1
- Now, how do you determine the equivalent binary # ?
  - Read the remainders from bottom to top!

Base Conversion Methods

- Conversion from Decimal to Binary
  - Another example: Convert 117_{10} to binary

117/2 = 58	with a remainder of	1
58/2 = 29	with a remainder of	0
29/2 = 14	with a remainder of	1
14/2 = 7	with a remainder of	0
7/2 = 3	with a remainder of	1
3/2 = 1	with a remainder of	1
1/2 = 0	with a remainder of	1

So, 117_{10} is the same as 1110101_2

- You stop when you get 0 as an answer
- Read the remainders from bottom to top to get binary #

Base Conversion Methods

- **Conversion from Decimal to Any Other Base**
  - The previous example worked great for base 2
  - Turns out that this method is not specific to base 2
  - Meaning, the same logic can be applied to convert from decimal to ANY other base!
  - Let's look at a couple of examples…

Base Conversion Methods

- Conversion from Decimal to Any Other Base
  - Convert 381_{10} to base 16 (hexadecimal)

381/16 = 23	with a remainder of	13	(D)
23/16 = 1	with a remainder of	7	
1/16 = 0	with a remainder of	1	

So, 381_{10} is the same as $17D_{16}$

- Start by dividing 381 by the BASE (to convert to)
- SAME idea: you stop when you get 0 as an answer
  - The final remainder could be anything 1 through 15 (F)
- Now, how do you determine the equivalent **base 16** # ?
  - Read the remainders from bottom to top!

Base Conversion Methods

- Conversion from Decimal to Any Other Base
  - Convert 175_{10} to base 3 (ternary)

175/3 = 58	with a remainder of	1
58/3 = 19	with a remainder of	1
19/3 = 6	with a remainder of	1
6/3 = 2	with a remainder of	0
2/3 = 0	with a remainder of	2

So, 175_{10} is the same as 20111_3

- Again, start by dividing 175 by the BASE (to convert to)
- SAME idea: you stop when you get 0 as an answer
  - In this case, the final remainder could be 1 or 2
- Now, how do you determine the equivalent **base 3** # ?
  - Read the remainders from bottom to top!

Brief Interlude: Bike Fail (non-UCF)

UCF Daily Bike Fail

Courtesy of
David Levy

Base Conversion Methods

- Generic Conversion Process
  - Convert from ANY base (call it B1)
  - To ANY to other base (call it B2)
    - where NEITHER of the bases are base 10
  - This is a two step process:
  1) <u>Convert from B1 to base 10</u>
  2) <u>Convert from base 10 to B2</u>
  - How to do this should be straightforward:
    - You simply utilize <u>**both**</u> of the methods already shown

Base Conversion Methods

- Generic Conversion Process
  - Convert 125_7 to base 4
  - This is a two step process:
  1) Convert 125_7 to base 10

  - Solution:
    - $125_7 = 1x7^2 + 2x7^1 + 5x7^0 = 68_{10}$

    - Refer to slide 17 for a reminder of how to do this step if there is confusion

Base Conversion Methods

- Generic Conversion Process
  - Convert 125_7 to base 4
  - This is a two step process:
  2) Now, convert 68_{10} to base 4
  - Solution:

68/4 = 17	with a remainder of	0
17/4 = 4	with a remainder of	1
4/4 = 1	with a remainder of	0
1/4 = 0	with a remainder of	1

Final Answer: 125_7 converts to 1010_4

So, 125_7 is the same as 68_{10}, which is the same as 1010_4

Base Conversion Methods

- Generic Conversion Process
  - If you are converting between two bases (B1 & B2) that are BOTH a perfect power of 2
  - You can use the method we just showed.
  - But the following process works more quickly:

1) <u>Convert from B1 to base 2</u>
2) <u>Convert from base 2 to B2</u>

  - Part 1 should be straightforward:
  - We just need to briefly look at Part 2

Base Conversion Methods

- Generic Conversion Process
  - Convert $A3D_{16}$ to base 8 (octal)
    - Notice they are both perfect powers of 2
  - This is a two step process:

1) Convert $A3D_{16}$ to base 2

  - Solution:
    - For this part, we just put the binary equivalent of each digit
    - $A3D_{16} = 1010\ 0011\ 1101_2$

Base Conversion Methods

- Generic Conversion Process
  - Convert $A3D_{16}$ to base 8 (octal)
    - Notice they are both perfect powers of 2
  - This is a two step process:

  2) Now, convert $1010\ 0011\ 1101_2$ to base 8

  - Solution:
    - Think:
      - How many possible counting digits are there in base 8?
      - DUH!
      - There are 8! Hence base 8! They are 0 through 7.

Base Conversion Methods

- Generic Conversion Process
  - Convert $A3D_{16}$ to base 8 (octal)
    - Notice they are both perfect powers of 2
  - This is a two step process:

2) Now, convert $1010\ 0011\ 1101_2$ to base 8

  - Solution:
    - Think:
      - Now, how many binary digits does it take to perfectly represent one octal (base 8) digit?
      - Three!
      - Why? Cuz $8 = 2^3$

Base Conversion Methods

- Generic Conversion Process
  - Convert $A3D_{16}$ to base 8 (octal)
    - Notice they are both perfect powers of 2
  - This is a two step process:

  2) Now, convert $1010\ 0011\ 1101_2$ to base 8

  - Solution:
    - So group the binary digits, in SETS OF THREE
      - From right to left
    - Then convert each set of three binary digits to its octal equivalent

Base Conversion Methods

- Generic Conversion Process
  - Convert $A3D_{16}$ to base 8 (octal)
    - Notice they are both perfect powers of 2
  - This is a two step process:

  2) Now, convert $1010\ 0011\ 1101_2$ to base 8

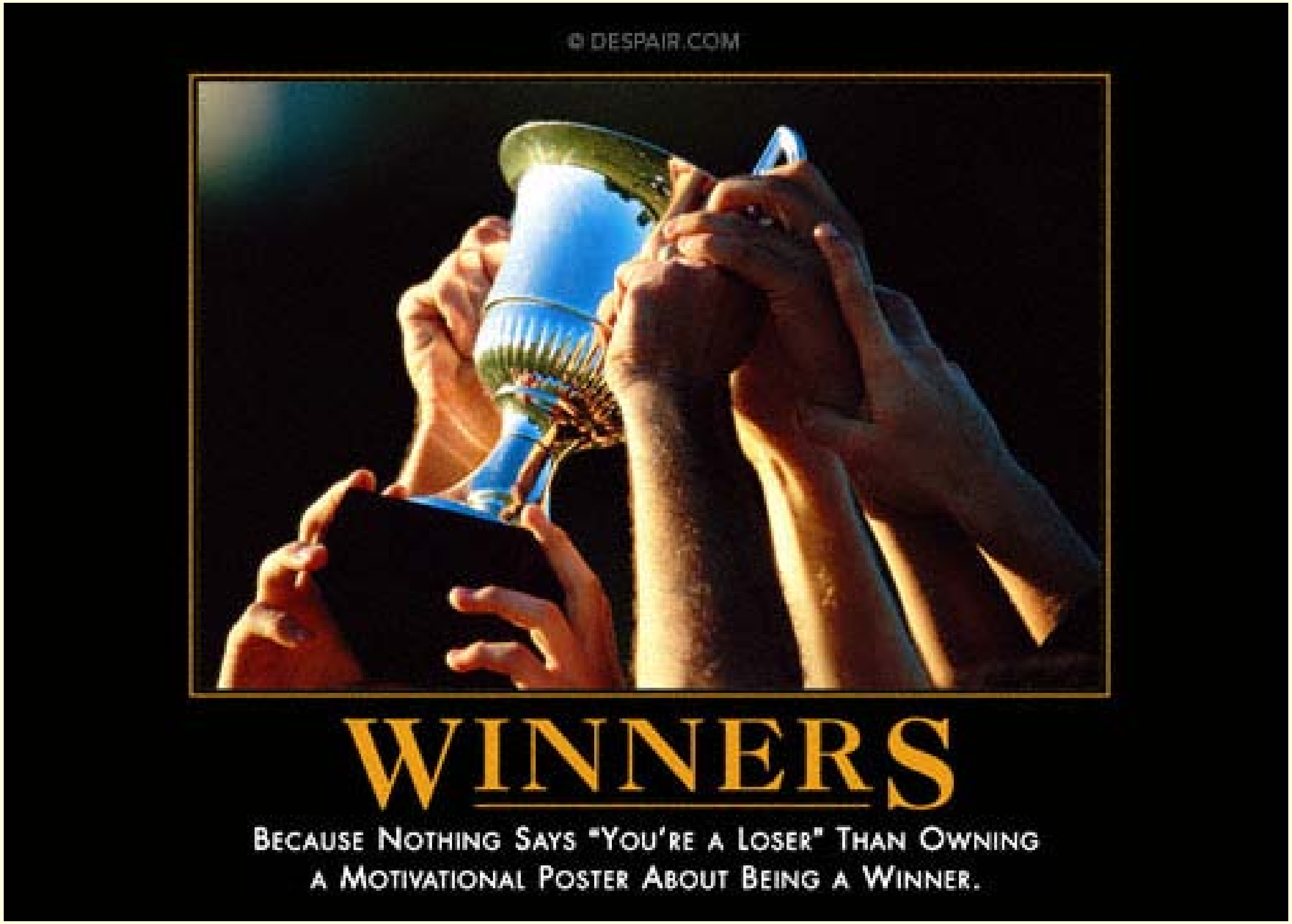
  - Solution:
    - $1010\ 0011\ 1101_2$
    - Just rewrite this with different spacing: $\underline{101}\ \underline{000}\ \underline{111}\ \underline{101}_2$
    - Convert each set of three digits:
    - 5075_8

Final Answer:
$A3D_{16}$ converts
to 5075_8

Base Conversions

We’re done!
WASN’T THAT STUPENDOUS!

Daily Demotivator



Base Conversions

Computer Science Department
University of Central Florida

COP 3502 – Computer Science I