Lecture 01

Number Systems

CSE115: Computing Concepts

Introduction to Numbering Systems

- Base: The number of fundamental symbols in a numbering system (e.g. 0, 1, 2 etc.)

We are all familiar with the decimal number system (Base 10). Some other number systems that we will work with are:

- **Binary $\rightarrow$ Base 2**
- **Octal $\rightarrow$ Base 8**
- **Hexadecimal $\rightarrow$ Base 16**

Common Number Systems

System	Base	Symbols	Used by humans?	Used in computers?
Decimal	10	0, 1, … 9	Yes	No
Binary	2	0, 1	No	Yes
Octal	8	0, 1, … 7	No	No
Hexa-decimal	16	0, 1, … 9, A, B, … F	No	No

Lets do some counting (1 of 3)

Decimal	Binary	Octal	Hexa-decimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7

Lets do some counting (2 of 3)

Decimal	Binary	Octal	Hexa-decimal
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Lets do some counting (3 of 3)

Decimal	Binary	Octal	Hexa-decimal
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17

Etc.

Bits and Bytes

A single binary digit is called a **bit**.
A collection of 8 bits is called a **byte**.

There are 10 fundamental digits in the binary number system. One of them is 0 and the other is 1.

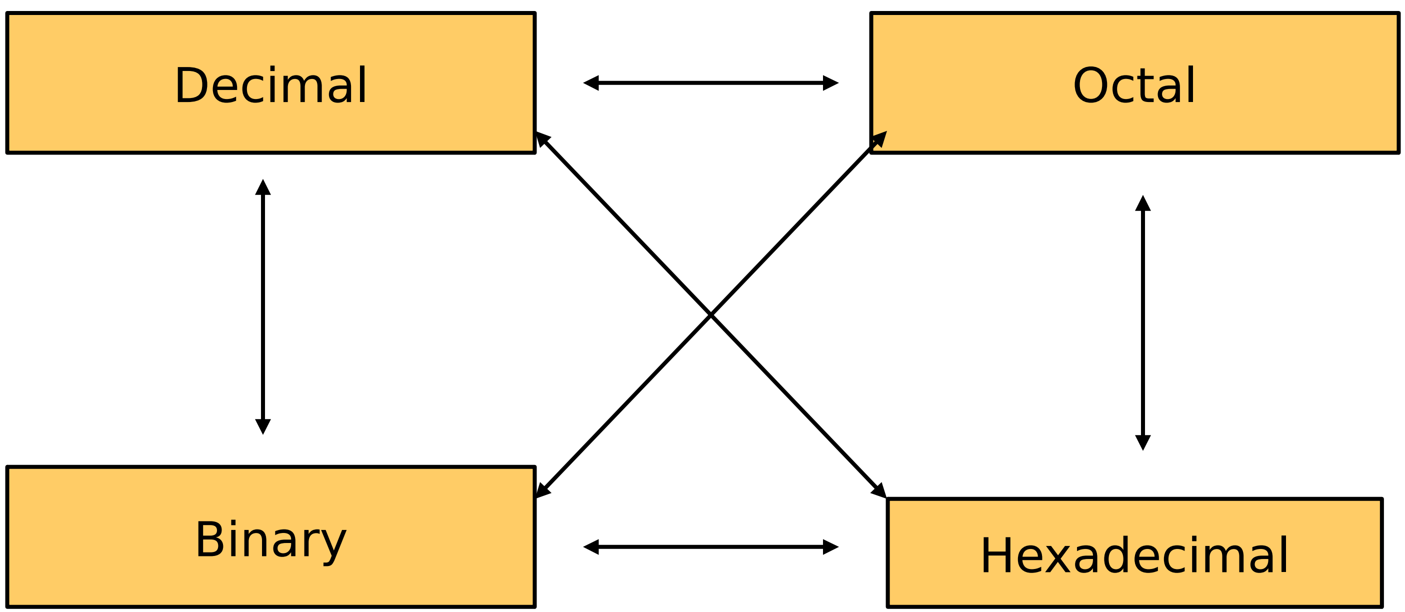
Write the base as subscript

2510 = 110012 = 318 = 1916

Base

Conversion Among Bases

- The possibilities:



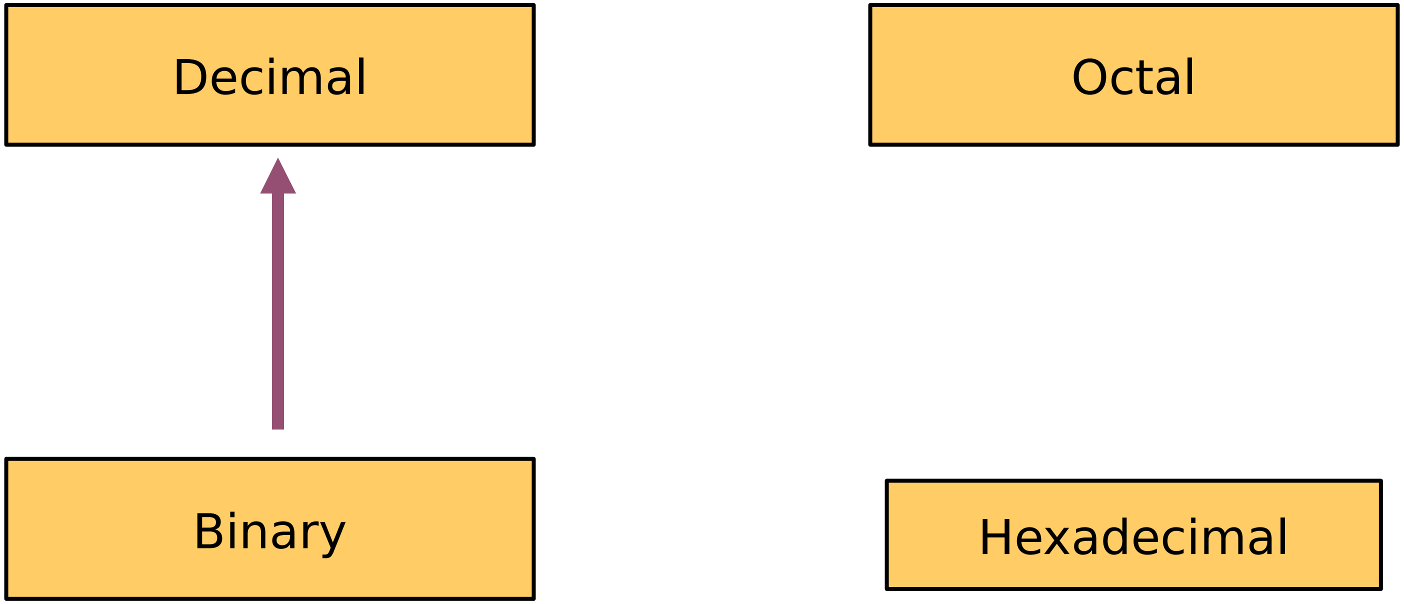
A closer look at a decimal number

Weight

```
12510 =>  5 x 100   =   5
        2 x 101   =  20
        1 x 102   = 100
              125
```

Base

Binary to Decimal



Binary to Decimal

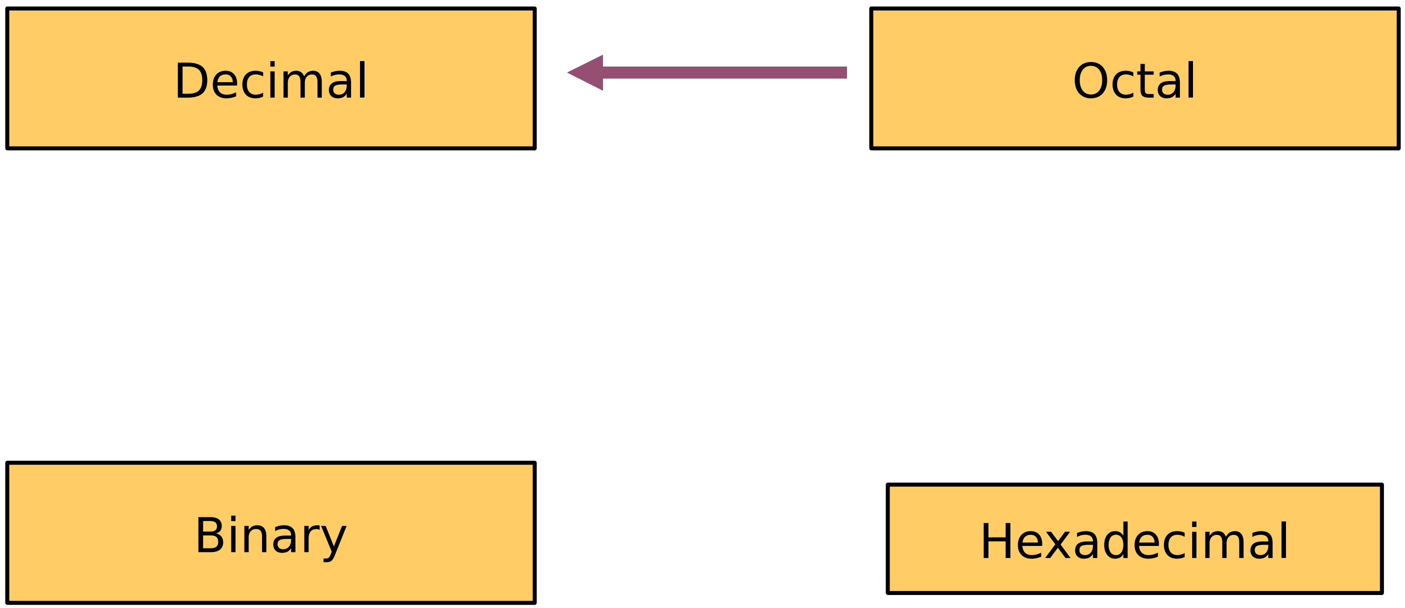
- Technique
  - Multiply each bit by 2^n, where *n* is the “weight” of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

Bit “0”

```
1010112 =>   1 x 20 =   1
             1 x 21 =   2
             0 x 22 =   0
             1 x 23 =   8
             0 x 24 =   0
             1 x 25 =  32
                       ----
                        4310
```

Octal to Decimal



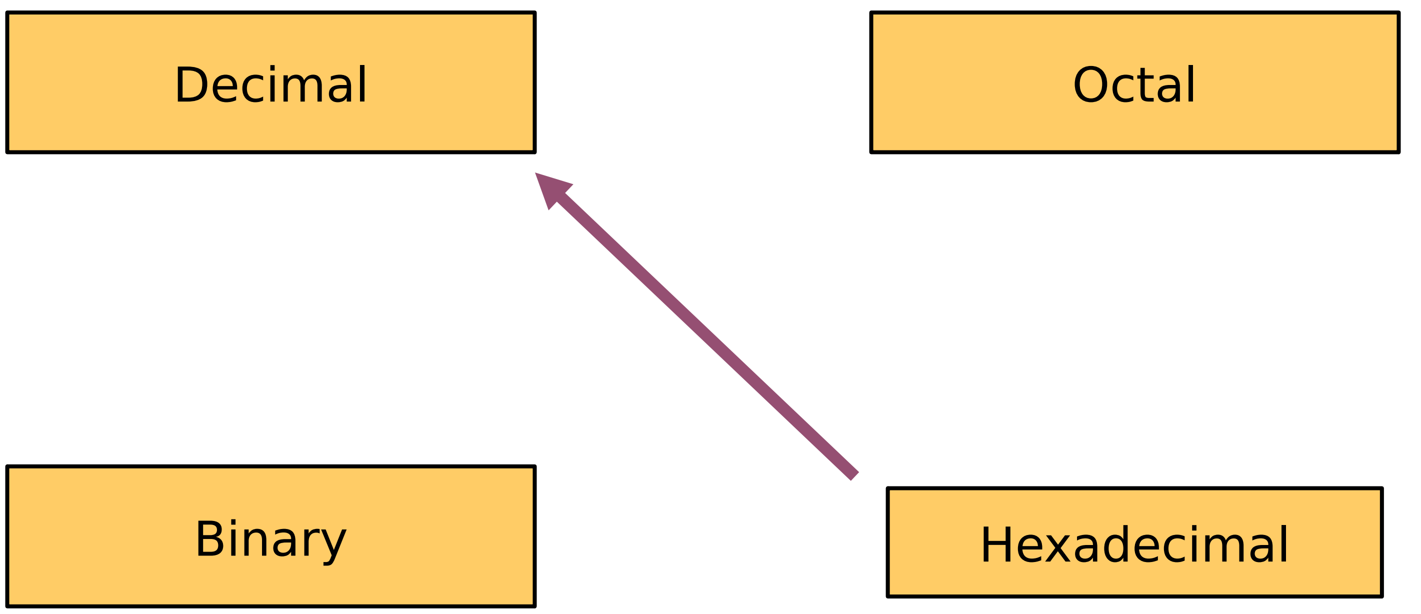
Octal to Decimal

- Technique
  - Multiply each bit by 8^n, where *n* is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

```
7248 =>   4 x 80 =     4
     2 x 81 =   16
     7 x 82 =  448 ______
          46810
```

Hexadecimal to Decimal



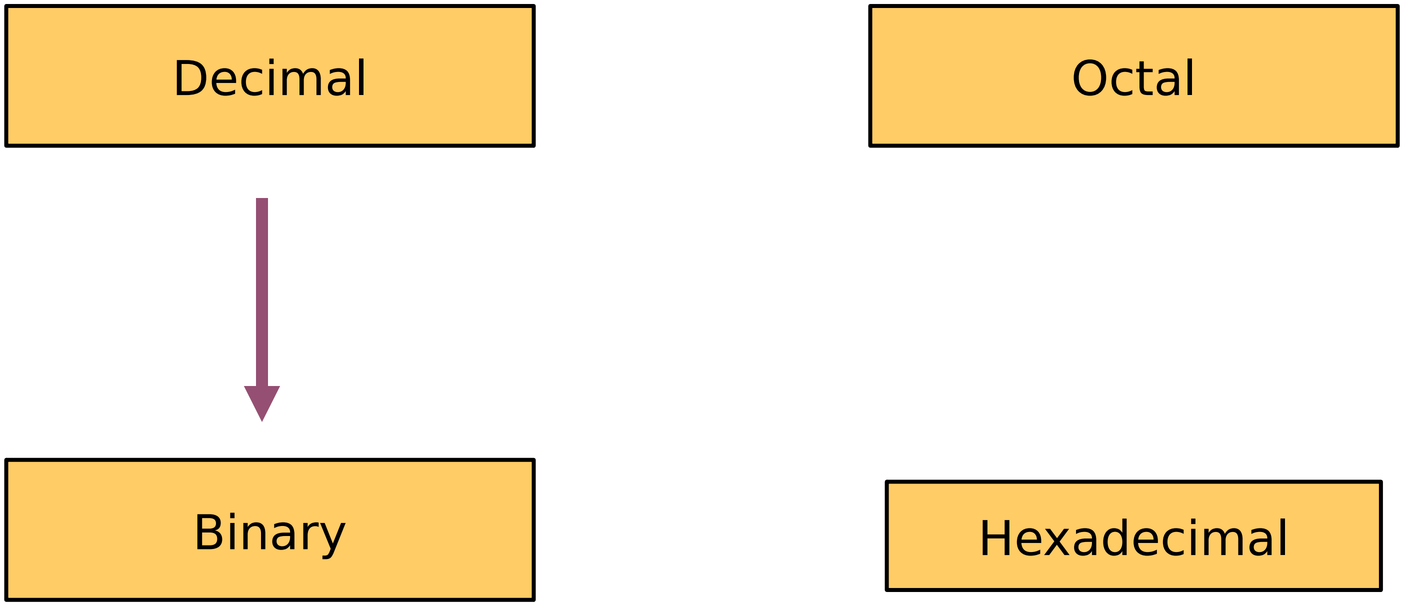
Hexadecimal to Decimal

- Technique
  - Multiply each bit by 16^n, where *n* is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

```
ABC16 =>  C x 160 = 12 x   1 =   12
          B x 161 = 11 x  16 =  176
        A x 162 = 10 x 256 = 2560
                             ________
                             274810
```

Decimal to Binary



Decimal to Binary

- Technique
  - Divide by two, keep track of the remainder
  - First remainder is bit 0 (LSB, least-significant bit)
  - Second remainder is bit 1
  - Etc.

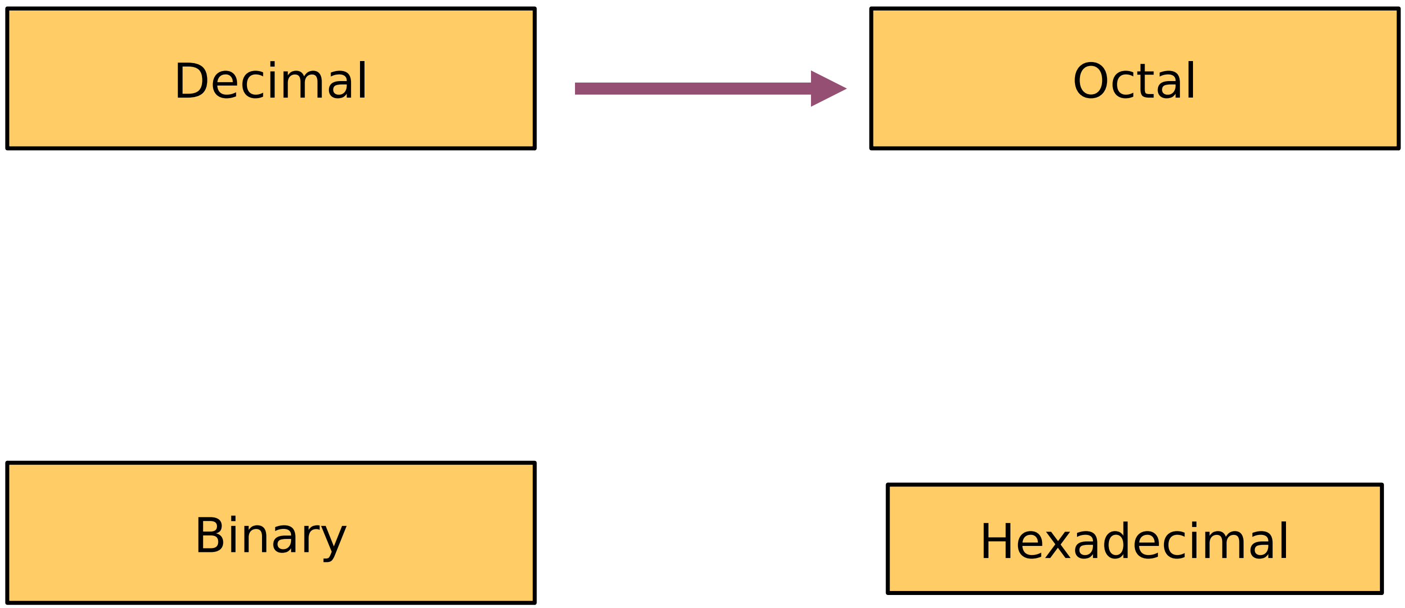
Example

12510 = ?2

2	125	
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	1

12510 = 11111012

Decimal to Octal



Decimal to Octal

- Technique
  - Divide by 8
  - Keep track of the remainder

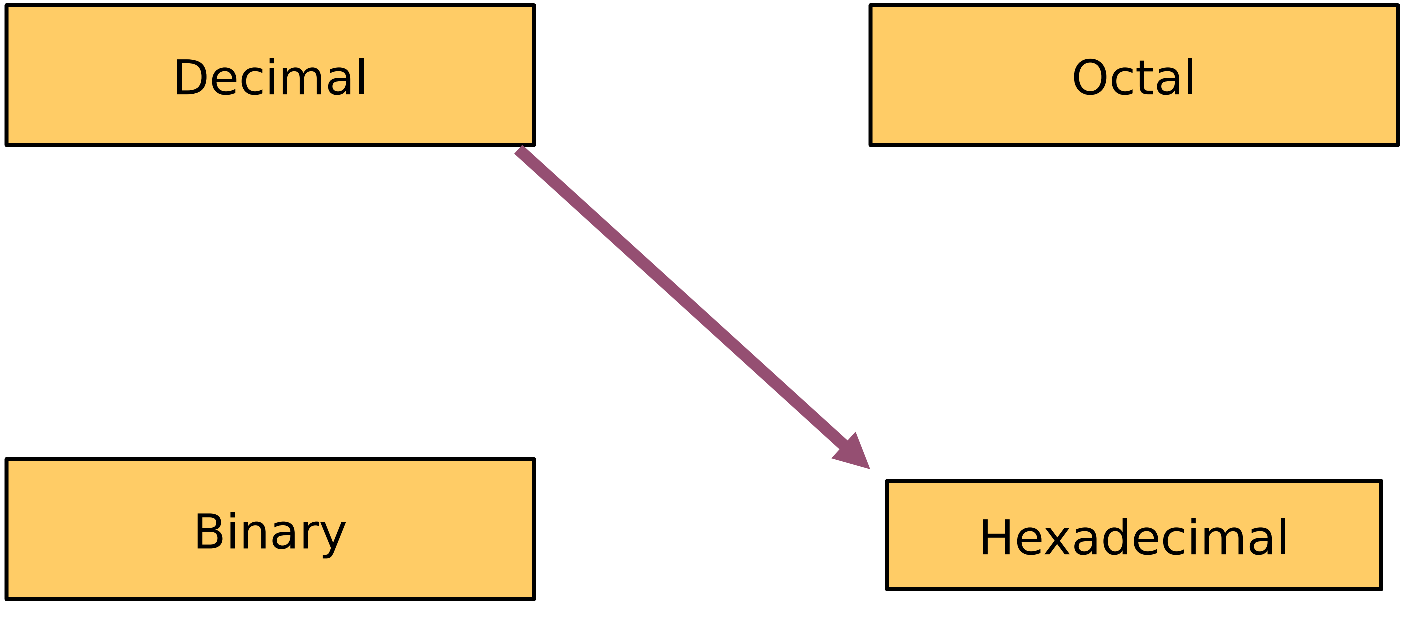
Example

$1234_{10} = ?_8$

8	1234	
8	154	2
8	19	2
8	2	3
	0	2

$1234_{10} = 2322_8$

Decimal to Hexadecimal



Decimal to Hexadecimal

- Technique
  - Divide by 16
  - Keep track of the remainder

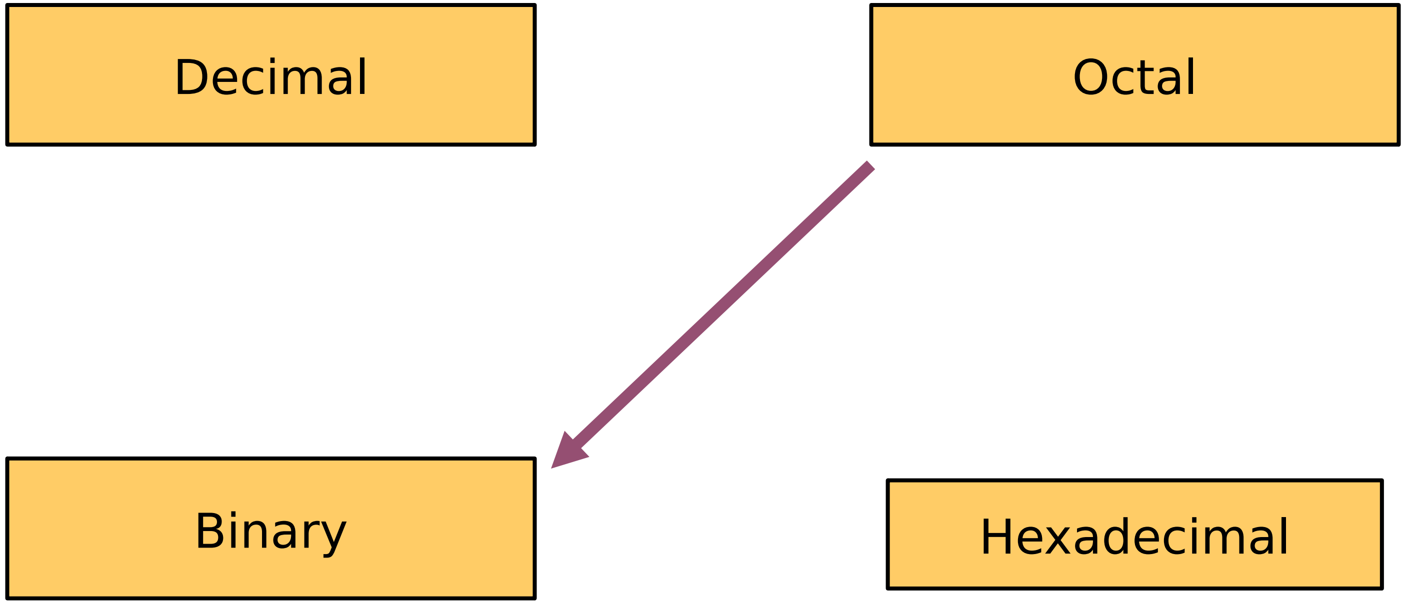
Example

123410 = ?16

16	1234		
16	77	2	
16	4	13	= D
	0	4	

123410 = 4D216

Octal to Binary



Octal to Binary

- Technique
  - Convert each octal digit to a 3-bit equivalent binary representation

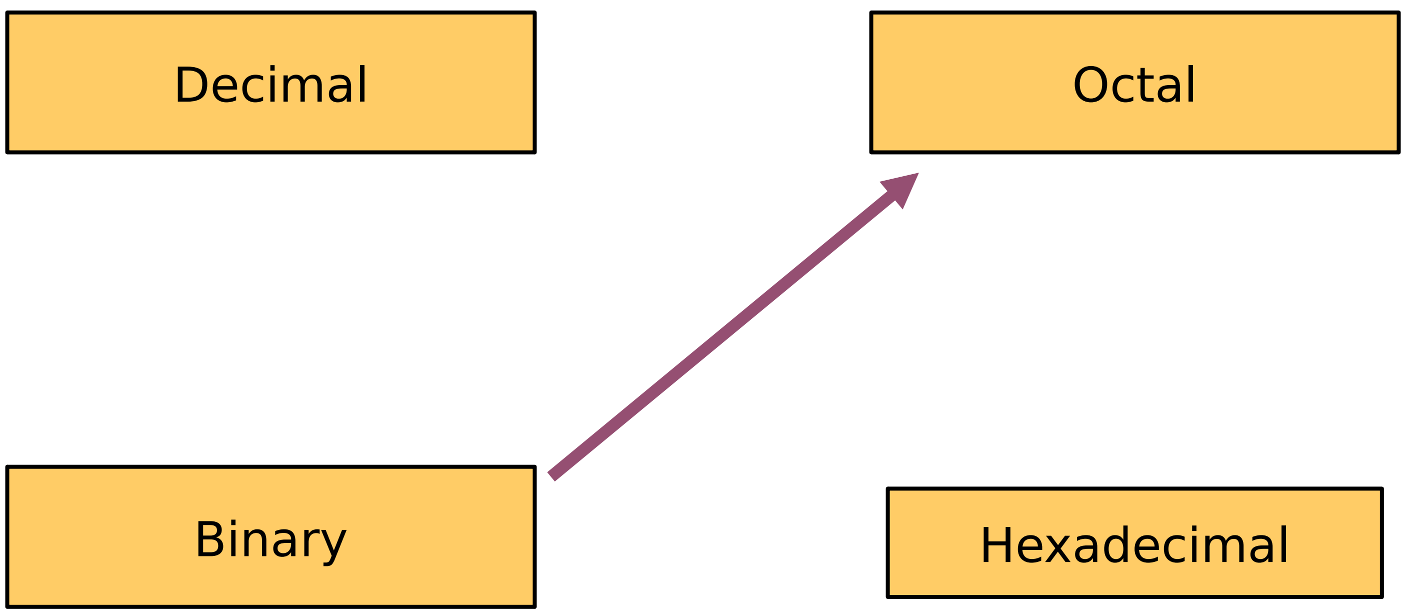
Example

$705_8 = ?_2$

$$\begin{array}{ccc} 7 & 0 & 5 \\ \downarrow & \downarrow & \downarrow \\ 111 & 000 & 101 \end{array}$$

$705_8 = 111000101_2$

Binary to Octal



Binary to Octal

- Technique
  - Group bits in threes, starting on right
  - Convert to octal digits

Example

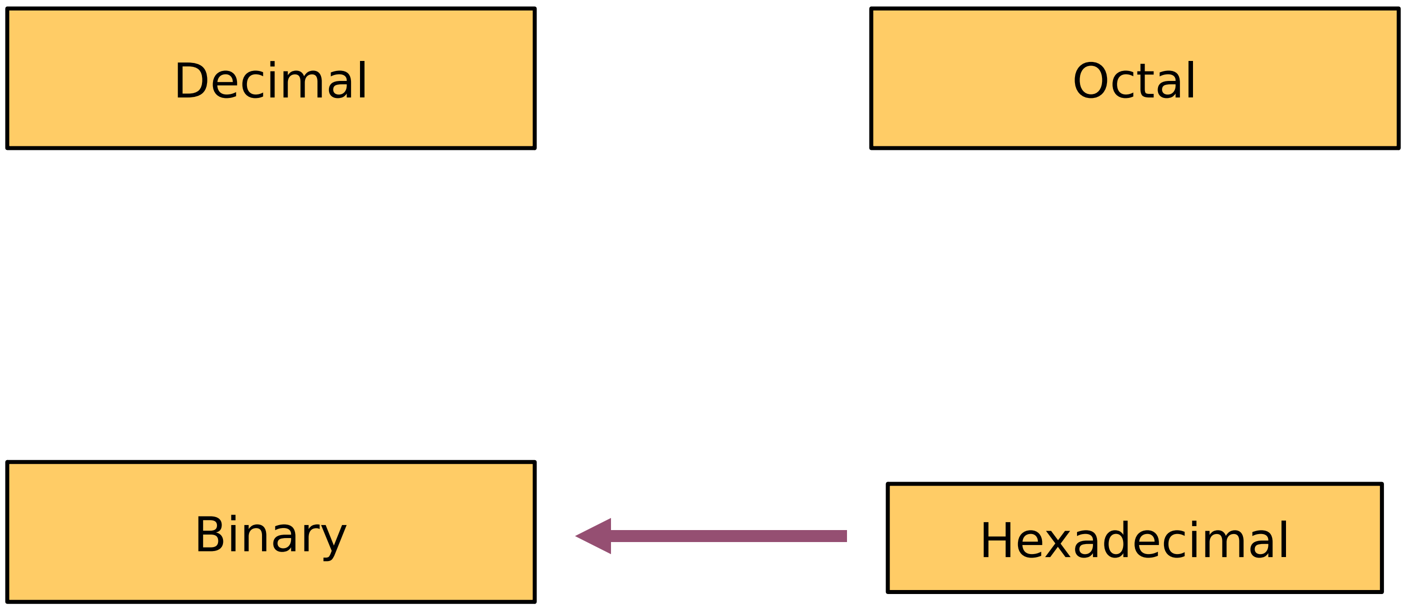
$1011010111_2 = ?_8$

1 011 010 111

1 3 2 7

$1011010111_2 = 1327_8$

Hexadecimal to Binary



Hexadecimal to Binary

- Technique
  - Convert each hexadecimal digit to a 4-bit equivalent binary representation

Example

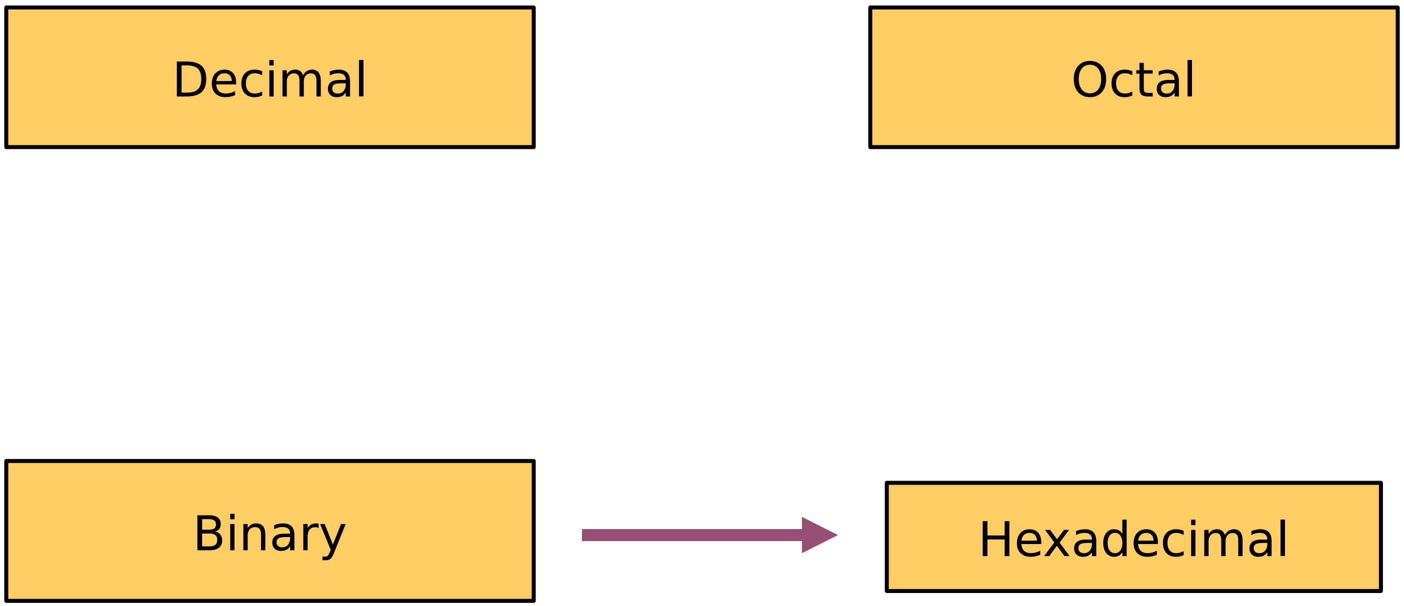
10AF16 = ?
2

1 0 A F

0001 0000 1010 1111

10AF16 =
00010000101011112

Binary to Hexadecimal



Binary to Hexadecimal

- Technique
  - Group bits in fours, starting on right
  - Convert to hexadecimal digits

Example

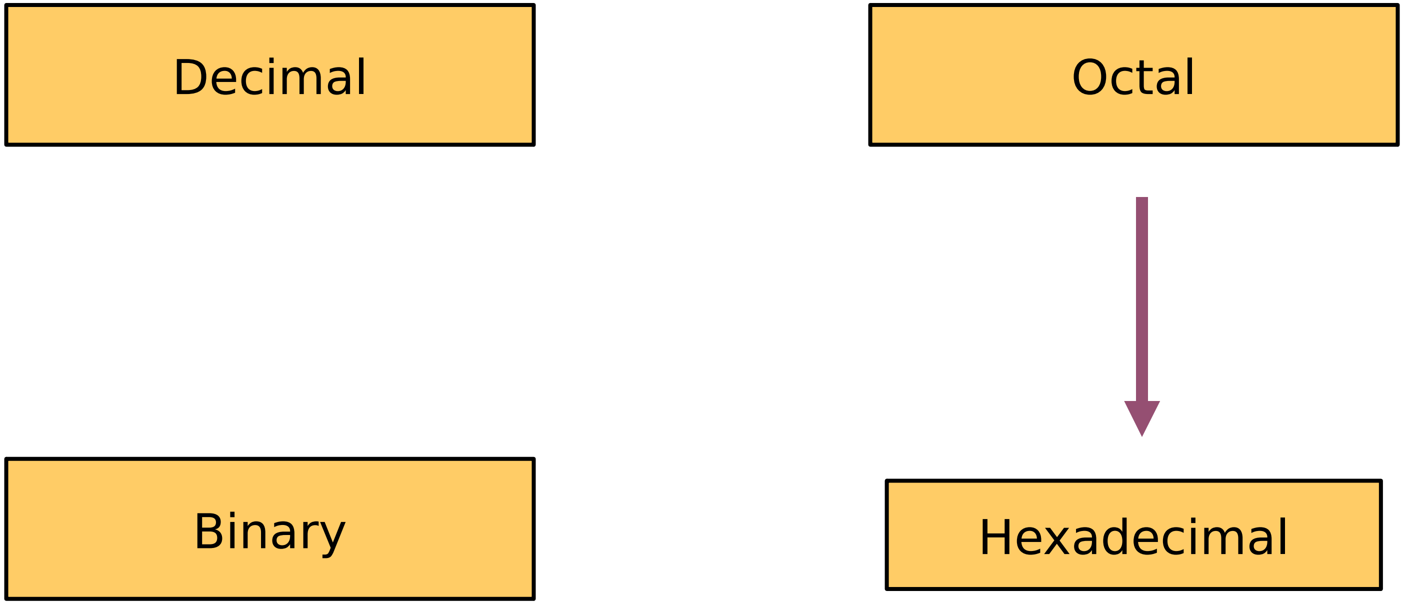
10101110112 = ?16

10 1011 1011

1 B B

10101110112 = 2BB16

Octal to Hexadecimal



Octal to Hexadecimal

- Technique
  - Use binary as an intermediary

Example

10768 = ?16

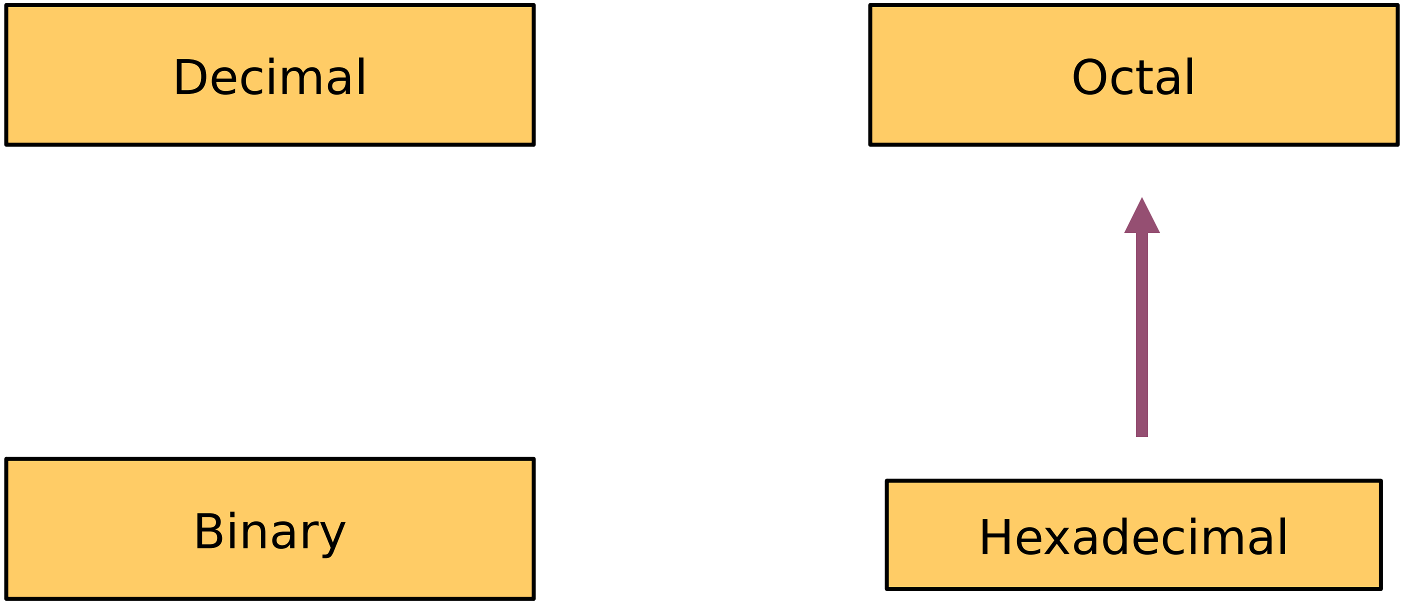
1 0 7 6

001 000 111 110

2 3 E

10768 = 23E16

Hexadecimal to Octal



Hexadecimal to Octal

- Technique
  - Use binary as an intermediary

Example

1F0C16 = ?8

1	F	0	C
0001	1111	0000	1100

1 7 4 1 4

1F0C16 = 174148

Conversion Exercise

Decimal	Binary	Octal	Hexa-decimal
33			
	1110101		
		703	
			1AF

Try not to use a calculator!

Conversion Exercise

Answer

Decimal	Binary	Octal	Hexa-decimal
33	100001	41	21
117	1110101	165	75
451	111000011	703	1C3
431	110101111	657	1AF

Common Powers (1 of 2)

- Base 10

Power	Preface	Symbol	Value
10-12	pico	p	.000000000001
10-9	nano	n	.000000001
10-6	micro	µ	.000001
10-3	milli	m	.001
103	kilo	k	1000
106	mega	M	1000000
109	giga	G	1000000000
1012	tera	T	1000000000000

Common Powers (2 of 2)

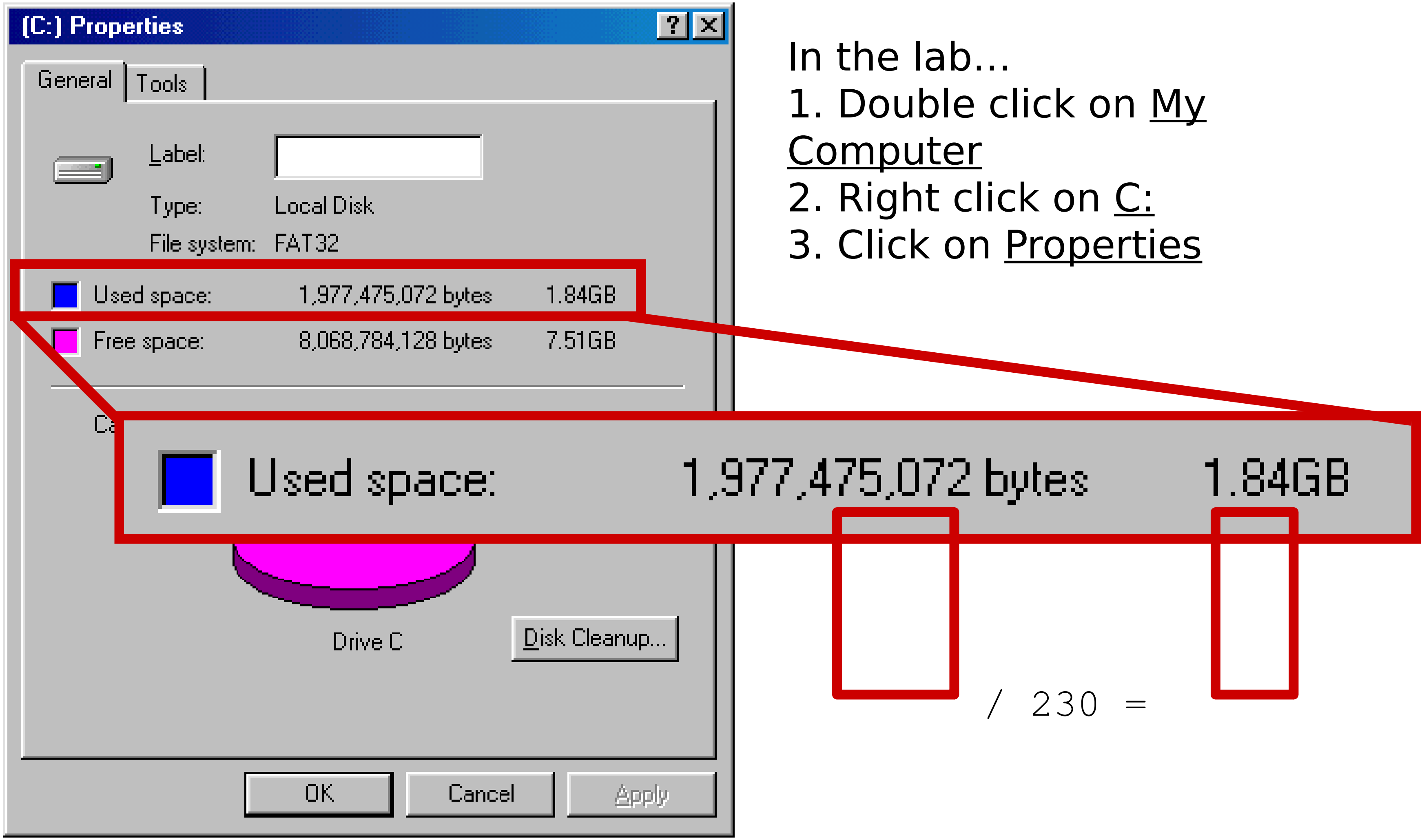
- Base 2

Power	Preface	Symbol	Value
210	kilo	k	1024
220	mega	M	1048576
230	Giga	G	1073741824

l What is the value of “k”, “M”, and “G”?

n computing, particularly w.r.t. memory,
the base-2 interpretation generally applies

Example



In the lab…

1. Double click on <u>My Computer</u>
2. Right click on <u>C:</u>
3. Click on <u>Properties</u>

/ 230 =

Binary Addition (1 of 2)

- Two 1-bit values

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	10

“two”

Binary Addition (2 of 2)

- Two *n*-bit values
  - Add individual bits
  - Propagate carries
  - E.g.,

```
  1 1
    11010        26
  + 11001      + 25
  -------      ----
   110011        51
```

Multiplication (1 of 3)

- Decimal (just for fun)

```
     35
x   105
-------
    175
   000
  35
-------
   3675
```

Multiplication (2 of 3)

- Binary, two 1-bit values

A	B	$A \times B$
0	0	0
0	1	0
1	0	0
1	1	1

Multiplication (3 of 3)

- Binary, two *n*-bit values
  - As with decimal values
  - E.g.,

```
       1110
     x 1011
   --------
       1110
      1110
     0000
    1110
   --------
   10011010
```

Complement

- Complement is the negative equivalent of a number.
- If we have a number N then complement of N will give us another number which is equivalent to –N
- So if complement of N is M, then we can say M = -N
- So complement of M = -M = -(-N) = N
- So complement of complement gives the original number

Types of Complement

- For a number of base r, two types of complements can be found
  - 1. r’s complement
  - 2. (r-1)’s complement
- Definition:
  - If N is a number of base r having n digits then
  - r’s complement of N = rn – N and
  - (r-1)’s complement of N = rn-N-1

Example

- Suppose N = (3675)10
- So we can find two complements of this number. The 10's complement and the 9's complement. Here n = 4
- 10's complement of (3675) = 104 - 3675
  = 6325
- 9's complement of (3675) = 104 - 3675 -1
  = 6324

Short cut way to find (r-1)’s complement

- In the previous example we see that 9’s complement of 3675 is 6324. We can get the result by subtracting each digit from 9.
- Similarly for other base, the (r-1)’s complement can be found by subtracting each digit from r-1 (the highest digit in that system).
- For binary 1’s complement is even more easy. Just change 1 to 0 and 0 to 1. (Because 1-1=0 and 1-0=1)

Example

- Find the (r-1)’s complement in short cut method.
  - (620143)8 Ans: 157634
  - (A4D7E)16 Ans: 5B281
  - (110100101)2 Ans: 001011010

Example

- Find the r's complement in short cut method.
  - $(8210)_{10}$ Ans: 1790
  - $(61352)_{10}$ Ans: 38648
  - $(6201430)_8$ Ans: 1576350
  - $(A4D7E0)_{16}$ Ans: 5B2820

Example for binary

- For binary: how to find 2's complement?
- start from rightmost bit
- Up to first 1 from right – no change.
- For rest of the bits toggle (Change 1 to 0 and 0 to 1)
  - $(11010010100)_2$ Ans: 00101101100
  - $(01101001011)_2$ Ans: 10010110101
  - $(1000000)_2$ Ans: 10000000

Use of Complement

- Complement is used to perform subtraction using addition
- Mathematically A-B = A + (-B)
- So we can get the result of A-B by adding complement of B with A.
- So A-B = A + Complement of (B)

Addition and Subtraction

- Two's complement addition follows the same rules as binary addition.

5 + (-3) = 2	0000 0101	=	+5
	+ 1111 1101	=	-3
	0000 0010	=	+2

- *Two… he min… ng a neg… ne).*

7 - 12 = (-5)	0000 0111	=	+7
	+ 0100	=	-12
	1011	=	-5