CD4002/CN4002
Computer Systems and Networks

Week 2 – Number Systems

University of East London

Pioneering Futures Since 1898

Agenda

- Why do we need an understanding of number systems?
- The common features of number systems
- The decimal, binary and hexadecimal number systems
- Converting numbers from one base to another

University of East London
Pioneering Futures Since 1898

Why do we need an understanding of number systems?

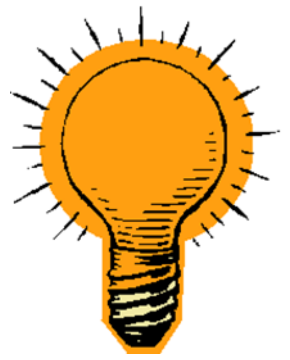
- Put simply, to help us understand how computers work and how they represent data

Binary numbers

- Form the basis of the Von Neumann architecture
- Natural relationship with on/off switches
- Used to represent both instructions and data

Hexadecimal numbers

- Closely related to binary numbers
- Used as a shorthand for binary numbers which makes life easier for humans
- Often used for debugging

	
On	Off
True	False
Yes	No
1	0

Base and Digits

Base: the number of different digits including zero that exist in the number system

Decimal number system

- Base = 10
- Number of digits = 10, 0 through 9

Binary number system

- Base = 2
- Number of digits = 2, 0 & 1

Hexadecimal number system

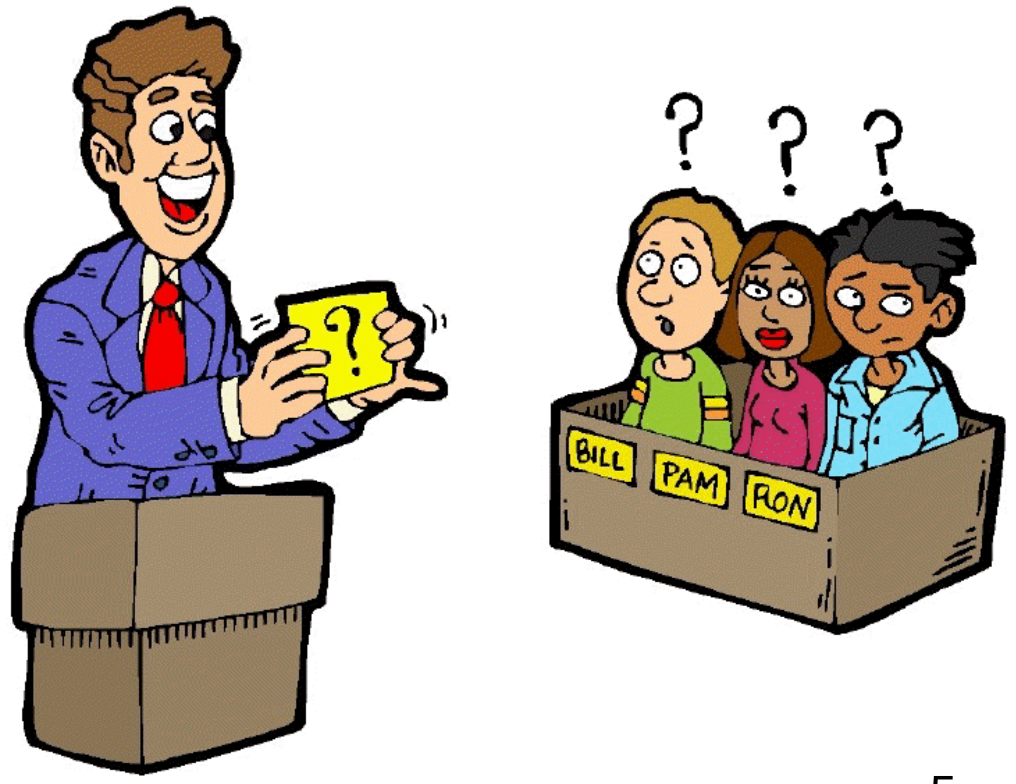
- Base = 16
- Number of digits = 16, 0 - 9, A - F (A equivalent to 10_{10}, B equivalent to 11_{10} etc.)

Quiz Time

Q. Unlike humans, martians have 3 fingers on each hand and hence use the senary (base 6) number system.

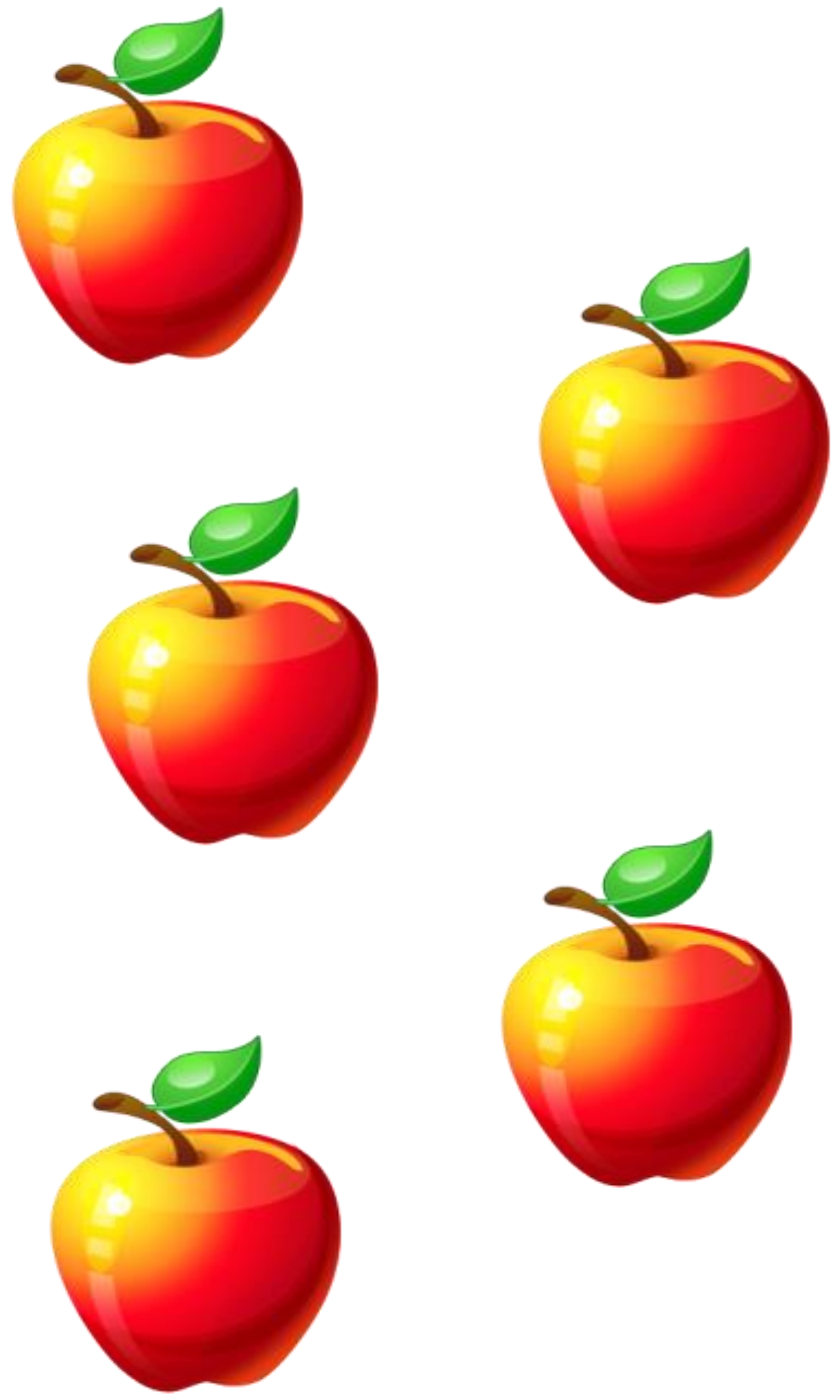
How many digits exist in this number system?

What are the digits?



Numbers: Physical Representation

- Different representation, same number of apples
  - Cave dweller: IIIII
  - Roman: V
  - You and I: 5_{10}
  - A digital computer: 101_2
  - A martian: 5_6



Modern Number Systems

- Based on positional notation (aka place value i.e. used to find the value)

1. Decimal system: system of **positional** notation based on powers of 10
   - i.e. 527_{10} is 5 lots of 100, 2 lots of 10 and 7 lots of 1
2. Binary system: system of **positional** notation based powers of 2
   - i.e. 111_2 is 1 lot of 4, 1 lot of 2 and 1 lot of 1
3. Hexadecimal system: system of **positional** notation based powers of 16
   - i.e. ABC_{16} is A (10) lots of 256, B (11) lots of 16 and C (12) lots of 1

The only thing that differentiates the above number systems is the base!

Positional Notation: Base 10

Using positional notation to find values in decimal

$$527 = 5 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$$

100's place **10's place** **1's place**

Place	10^2	10^1	10^0
Value	100	10	1
Evaluate	5 x 100	2 x 10	7 x1
Sum	500	20	7

Positional Notation: Binary

Using positional notation to find values in decimal

$$1101\ 0110_2 = 214_{10}$$

Place	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	128	64	32	16	8	4	2	1
Evaluate	1 x 128	1 x 64	0 x 32	1 x16	0 x 8	1 x 4	1 x 2	0 x 1
Sum for Base 10	128	64	0	16	0	4	2	0

Try these yourself

- Using positional notation, convert the following binary numbers into decimal.
  - 1011_2
  - 10110100_2

Place	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	128	64	32	16	8	4	2	1
Evaluate								
Sum for Base 10								

Positional Notation: Hexadecimal

Using positional notation to find values in decimal

$\mathbf{6{,}704_{16}} = \mathbf{26{,}372_{10}}$

4,096's place **256's place** **16's place** **1's place**

Place	16^3	16^2	16^1	16^0
Value	4,096	256	16	1
Evaluate	6 x 4,096	7 x 256	0 x 16	4 x 1
Sum for Base 10	24,576	1,792	0	4

And try this one!

- Using positional notation, convert the following hexadecimal number into decimal.
  - $123A_{16}$

Place	16^3	16^2	16^1	16^0
Value	4,096	256	16	1
Evaluate				
Sum for Base 10				

University of East London
Pioneering Futures Since 1898

The Range of Possible Numbers

Range: determines the total number of values that can be represented by a number of a particular size.

The formula for finding the range is:

- $R = B^K$ where R = range, B = base, K = number of digits

- Example #1: Find the range of a 1 digit decimal number
  - $R = 10^1 =$ 10 different numbers (0…9)
- Example #2: Find the range of a 3 digit decimal number
  - $R = 10^3 =$ 1,000 different numbers (0…999)
- Example #3: Find the range of a 16 bit binary number
  - $R = 2^{16} =$ 65,536 or 64K different numbers (0…65,535)
  - A 16 bit integer can never store more than 64K different integers

Note: the largest number is always one less than the range.

University of East London
Pioneering Futures Since 1898

Decimal Range for Bit Widths

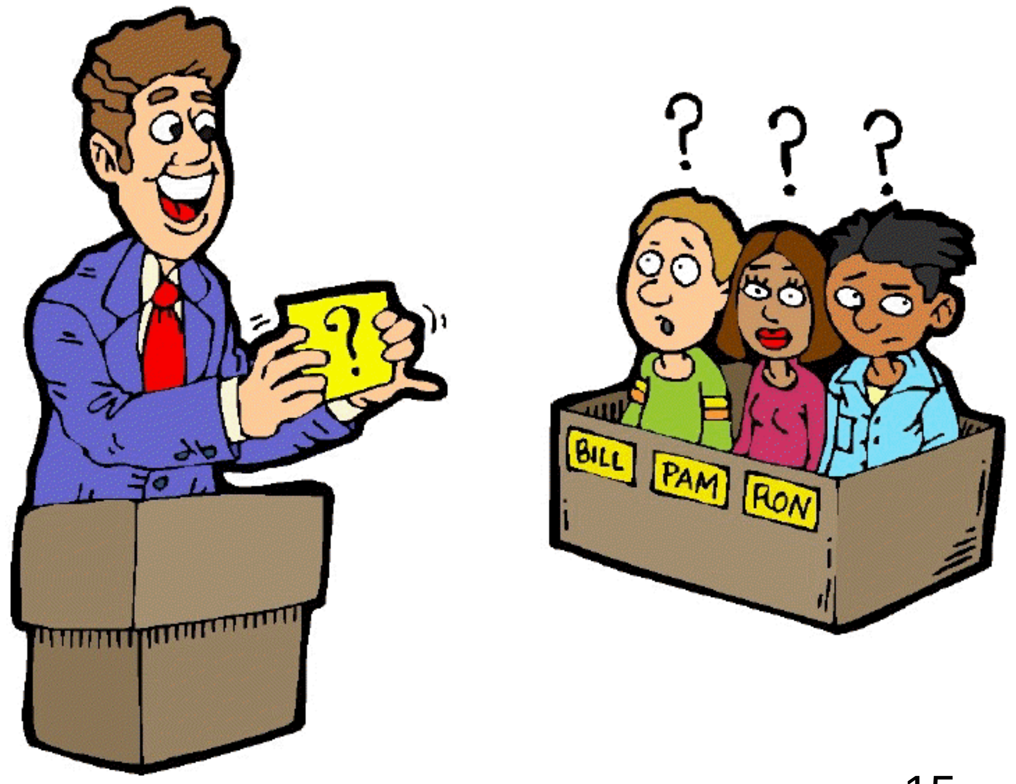
Bits	Digits	Range
1	0+	2 (0 and 1)
4	1+	16 (0 to 15)
8	2+	256
10	3	1,024 (1K)
16	4+	65,536 (64K)
20	6	1,048,576 (1M)
32	9+	4,294,967,296 (4G)
64	19+	Approx. 1.6×10^{19}
128	38+	Approx. 2.6×10^{38}

Try this example!

Q. What is the range of a 3 digit hexadecimal number?

Q. What is the largest number that can be represented by 3 hexadecimal digits?

Use the formula $R = B^K$ where R = range, B = base and K = number of digits



University of East London
Pioneering Futures Since 1898

Number of Symbols vs. Number of Digits

- For a given number, the ***larger*** the base
  - the ***more*** symbols required (e.g. 16 symbols required in Base 16, 0 – 9, A - F)
  - but the ***fewer*** digits needed

- To prove that the above statement is true, look at the following examples:

- Example #1: Convert 65_{16} to decimal and binary
  - 101_{10} $110\ 0101_2$

- Example #2: Convert $11C_{16}$ to decimal and binary
  - 284_{10} $1\ 0001\ 1100_2$

Counting in Base 2

Binary Number	Equivalent				Decimal Number
	8's (2^3)	4's (2^2)	2's (2^1)	1's (2^0)	
0				0×2^0	0
1				1×2^0	1
10			1×2^1	0×2^0	2
11			1×2^1	1×2^0	3
100		1×2^2			4
101		1×2^2		1×2^0	5
110		1×2^2	1×2^1		6
111		1×2^2	1×2^1	1×2^0	7
1000	1×2^3				8
1001	1×2^3			1×2^0	9
1010	1×2^3		1×2^1		10

Arithmetic operations in Number Bases: Addition

Base	Problem	Largest Single Digit
Decimal	$\begin{array}{r} 6 \\ \underline{+3} \end{array}$	9
Hexadecimal	$\begin{array}{r} 6 \\ \underline{+9} \end{array}$	F
Binary	$\begin{array}{r} 1 \\ \underline{+0} \end{array}$	1

Addition with carry forward

Base	Problem	Carry	Answer
Decimal	6 +4	Carry the 10	10
Hexadecimal	6 +A	Carry the 16	10
Binary	1 +1	Carry the 2	10

Binary Addition

		1	1	0	1	1	0	1
+				1	0	1	1	0
	1	[1] 0	[1] 0	[1] 0	[1] 0	0	1	1

Hexadecimal Addition

	A	B	C
+	4	4	4
	[1]F	[1]0	0

University of East London
Pioneering Futures Since 1898

Now you try!

- Perform the following addition:
    - $1101_2 + 110_2$

Converting to and from Base 10

- Using Powers Table (e.g. 2^p and 16^p)

		Power								
		8	7	6	5	4	3	2	1	0
Base	2	256	128	64	32	16	8	4	2	1
	16					65,536	4,096	256	16	1

University of East London
Pioneering Futures Since 1898

Method 1: From Base 10 to Base 2

		Power						
		6	5	4	3	2	1	0
Base	2	64	32	16	8	4	2	1
			1	0	1	0	1	0
Integer			42/32 = 1	10/16 = 0	10/8 = 1	2/4 = 0	2/2 = 1	0/1 = 0
Remainder			10	10	2	2	0	0

$42_{10} = 101010_2$

Method 2: From Base 10 to Base 2
(Long division using the base and carry)

Base 10 42

Quotient

Remainder

2)	42	(**0**	Least significant bit
2)	21	(**1**	
2)	10	(**0**	
2)	5	(**1**	
2)	2	(**0**	
2)	**1**		Most significant bit

Base 2 **101010**

From Base 10 to Base 16

(Long division: using the base and carry)

Base 10 5,735

Remainder

Quotient

16)	5,735	(7	Least significant bit
16)	358	(6	
16)	22	(6	
16)	1	(1	Most significant bit
16)	0		

Base 16 **1667**

From Base 16 to Base 2 and vice-versa

- The nibble approach (i.e. group into 4 bits then convert to hexadecimal and vice versa)

Base 16	1	F	6	7
Base 2	0001	1111	0110	0111

Binary	Hexa-decimal	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

And one for you to try!

- Convert the binary number below to hexadecimal. (Watch out for the 2 most significant bits!)

Base 16				
Base 2	10	1010	0010	0110

Binary	Hexa-decimal	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Example of the use of ASCII code (base 16) in computing

LSD \ MSD	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	“	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACJ	SYN	&	6	F	V	f	v
7	BEL	ETB	‘	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

74_{16}

111 0100

Learning Objectives

- On completion of this topic, you will be able to:
  - Explain the relevance of the binary and hexadecimal number systems in computing
  - Convert decimal numbers to and from binary and hexadecimal
  - Convert binary numbers to and from hexadecimal
  - Perform simple additions in binary and hexadecimal

Reading

- Essential reading
    - Englander, Chapter 3
- Recommended reading
    - Stallings, Chapter 9