FACTFILE: GCE DIGITAL TECHNOLOGY

AS2 DATA REPRESENTATION

Rewarding Learning

Data Representation

Learning Outcomes

Students should be able to:

- explain the terms bit, byte, kilobyte, megabyte, gigabyte and terabyte;
- demonstrate that 2^n different values can be represented with n bits (maximum n = 8);
- perform conversions from decimal to binary and from binary to decimal for a maximum of 8 bits;
- demonstrate how the two's complement system can represent positive and negative numbers in binary using 8 bits;
- demonstrate how ASCII and Unicode are used to represent characters;

Content in Data Representation Fact File

- ✔ Units of Storage
- ✔ The range of values that can be represented using n bits
- ✔ Conversions from decimal to binary and from binary to decimal for a maximum of 8 bits
- ✔ Using the two's complement system to represent positive and negative numbers in binary
- ✔ Representing characters using ASCII and Unicode

Units of Storage

Computers use a variety of memory techniques to store data. All data is stored in digital format using a number system known as binary. A Binary digIT (known as a BIT) is either a 0 or a 1. It is the smallest unit of storage. When bits are grouped together, typically eight, it is referred to as a byte. A single character (such as a letter or digit) is typically represented by a byte. The capacity of storage in a typical computer or peripheral is measure in bytes. Multiples of byes are referred to as:

1024 Bytes	=	1 Kilobyte
1024 Kilobytes	=	1 Megabyte
1024 Megabytes	=	1 Gigabyte
1024 Gigabytes	=	1 Terabyte

These terms are usually used to describe disk capacity, or data storage capacity, and system memory. Today, Terabyte is the common term being used to describe the capacity of a hard drive.

Question: Calculate the number of bytes in a 2 Gigabyte USB memory pen?

The range of values that can be represented using n bits

The number of bits used will determine the number of different values that can be represented. The greater the number of bits the greater the number of values. For example if we use 3 bits, this will allow eight different values as shown below.

2^2	2^1	2^0	Decimal Value
4	2	1	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Using unique patterns of 0 and 1, eight different combinations can be represented as shown in the table. The lowest value is zero whilst the highest value is 7. Therefore the range of values using 3 bits is 0–7.

- Using 4 bits will give a range 0–15
- Using 5 bits will give a range 0–31
- Using 6 bits will give a range 0–63

If we consider 8 bits the lowest value will be 0 and the highest value will be 255 as shown below. The bit position on the right hand side is referred to as the least significant bit (LSB) whereas the bit position on the left hand side is referred to as the most significant bit (MSB).

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	255

Therefore in general terms 2^n different values can be represented with n bits. Although 2^8 has a value of 256, the highest value is 255 because we include zero in the range of combinations of an 8 bit pattern. To calculate the range of values for n bits we can apply the following rule 0 – (2^n-1).

Question: Calculate the range of values using 7 bits?

Conversions from decimal to binary and from binary to decimal for a maximum of 8 bits;

In the decimal (sometimes referred to as denary) numbering system, each integer number column has values of units, tens, hundreds, thousands, etc. as we move along the number from right to left. Mathematically these values are written as index numbers starting from the right hand side 100, 101, 102, 103 etc. Then each position to the left of the decimal point indicates an increased positive power of 10.

…….	10^3	10^2	101	10^0
………	1000	100	10	1

The Binary Numbering System is used in all digital and computer based systems. Binary numbers follow the same set of rules as the decimal numbering system. The main difference is the decimal system uses powers of ten whereas the binary numbering system works on powers of two. Each binary place value can be converted to an equivalent decimal number.

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

To covert from decimal to binary we can use the "Repeated Division by 2" method. You write down the decimal number and continually divide-by-2 to give a result and a remainder of either a "1" or a "0" until the final result equals zero.

For example covert the decimal value 41 to binary.

41 divided by 2 = 20 remainder 1 (Least significant BIT)

20 divided by 2 = 10 remainder 0

10 divided by 2 = 5 remainder 0

5 divided by 2 = 2 remainder 1

2 divided by 2 = 1 remainder 0

1 divided by 2 = 0 remainder 1 (Most significant BIT)

When writing the binary equivalent start with the lowest remainder in your calculation i.e. the MSB.

Therefore 41 = 101001 in binary. Remember the LSB is the rightmost value and the MSB is the leftmost value.

Tip: An alternative way is to break the decimal number into binary place values.

Therefore 41 = 32+8+1 =100101_2

When converting from binary to decimal, the decimal number is equal to the sum of powers of 2 of the binary number '1' digits. Consider changing the binary value 11001110_2 to decimal.

Binary Powers	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Binary Value	1	1	0	0	1	1	1	0
Decimal Values	128	64	32	16	8	4	2	1
Sum the decimal values= 128 + 64 + 8 + 4 + 2 = 206								

Therefore 11001110_2 = 206

Questions:
Convert the following decimal number to binary using 8 bits
51
101
83

Convert the following binary numbers to decimal
$1011\ 0011_2$
01010101_2
00110011_2

Using the two's complement system to represent positive and negative numbers in binary
Consider the problem of representing both positive and negative integers over a given range in terms of only using ones and zeroes. This would mean we have to build in the sign as part of the binary representation.

We can represent a negative number in binary by making the most significant bit (MSB) a **sign bit**, which will tell us whether the number is positive or negative. In two's complement if the MSB has a value of 1 then the number will have a negative value and if the MSB is 0 then the number will have a positive value.

The column headings for an 8 bit two's complement number will look like this:

+/-128	64	32	16	8	4	2	1
MSB							LSB

The range of decimal values using 8 bits twos complement is -128 to +127. In general terms the range for n bits is ($-2^{n-1)}$ to (2^{n-1}-1).

To find the binary value of a negative decimal number we apply some simple rules. To demonstrate this consider the following question. Using two's complement represent the value -71 in binary

1. Convert 71 into binary

+/-128	64	32	16	8	4	2	1	
0	1	0	0	0	1	1	1	= 71

2. Change 0 to 1's and 1's to 0

+/-128	64	32	16	8	4	2	1
1	0	1	1	1	0	0	0

3. Add 1

+/-128	64	32	16	8	4	2	1	
1	0	1	1	1	0	0	0	
							1	
1	0	1	1	1	0	0	1	-71

We can now check the answer -128+ (32+16+8+1) = -128+57 = -71

Tip: An alternative way to find the two's complement of a binary value is:
Starting from the right, find the first '1' 01000111
Invert all of the bits to the left of that one 10111001

Questions
Convert the following decimal number to binary using two's complement and 8 bits
- – 51
- – 101
- – 83

Representing characters using ASCII and Unicode
A character set is a set of symbols that maybe represented in a computer at a particular time. These symbols, called characters, can be classified as letters, digits and punctuation marks. A character set also contains "control" characters which are non-printing and are used for special purposes such as an end of file marker. Two examples of character sets are ASCII and Unicode.

ASCII (American Standard Code for Information Interchange)
ASCII uses a 7 bit code and represents 128 characters including control characters. The characters are coded as consecutive numbers from 0 to 127. These characters and control codes are encoded as simple, unsigned, binary integers. Every character has an unique binary pattern as shown in the table below

Char	Dec	Binary	Char	Dec	Binary	Char	Dec	Binary
!	033	00100001	A	065	01000001	a	097	01100001
"	034	00100010	B	066	01000010	b	098	01100010
#	035	00100011	C	067	01000011	c	099	01100011
$	036	00100100	D	068	01000100	d	100	01100100
%	037	00100101	E	069	01000101	e	101	01100101
&	038	00100110	F	070	01000110	f	102	01100110
'	039	00100111	G	071	01000111	g	103	01100111
(	040	00101000	H	072	01001000	h	104	01101000
)	041	00101001	I	073	01001001	i	105	01101001
*	042	00101010	J	074	01001010	j	106	01101010
+	043	00101011	K	075	01001011	k	107	01101011
,	044	00101100	L	076	01001100	l	108	01101100
-	045	00101101	M	077	01001101	m	109	01101101
.	046	00101110	N	078	01001110	n	110	01101110
/	047	00101111	O	079	01001111	o	111	01101111
0	048	00110000	P	080	01010000	p	112	01110000
1	049	00110001	Q	081	01010001	q	113	01110001
2	050	00110010	R	082	01010010	r	114	01110010
3	051	00110011	S	083	01010011	s	115	01110011
4	052	00110100	T	084	01010100	t	116	01110100
5	053	00110101	U	085	01010101	u	117	01110101
6	054	00110110	V	086	01010110	v	118	01110110
7	055	00110111	W	087	01010111	w	119	01110111
8	056	00111000	X	088	01011000	x	120	01111000
9	057	00111001	Y	089	01011001	y	121	01111001

ASCII was designed to represent English-language text for an American user base, and is therefore insufficient for representing text in almost any language other than American English.

UNICODE

Unicode is a worldwide character standard. It allows for the interchange, processing, and display of the written texts of the majority of the diverse languages of the modern world. Whereas ASCII was limited to American English characters

Therefore it is a single-coded character set that incorporates characters from almost all the worlds' languages. To allow for the increase in representing characters Unicode is a 16 bit extension of ASCII code.

Question

1 Calculate the approximate number of characters that could be represented using 16 bits