**Presentation Notes**

1. Number systems used in Computer Science
   1. List the main features of the Decimal System  
      Digits: 0,1,2,3,4,5,6,7,8,9

Used for communicating with human users

* 1. List the main features of the Binary System

Digits: 0,1 (On or Off)

Binary 10 == Decimal 2

Used by internal CPU and Memory circuits

* 1. List the main features of the Octal System

Digits: 0,1,2,3,4,5,6,7 (No digits 8 & 9)

Octal 10 == Decimal 8

Used by Computer Scientists for groupings of 3 binary digits

* 1. List the main features of the Hexadecimal System

Digits: 0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F (Uses extra letters)

Hex F == Decimal 15

Hex 10 == Decimal 16

Used by Computer Scientists for groupings of 4 binary digits

1. Compare and contrast the Decimal and Binary systems

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Decimal System** | **Binary System** |
| Digits  Used | 0, 1, 2, 3, 4, 5, 6, 7 , 8, 9 | 0, 1 (True, False) (+, -) |
| Addition Example | 0 + 1 = 1  1 + 1 = 2  99 + 1 = 100 | 0 + 1 = 1  1 + 1 = 10  11 + 1 = 100 |
| Powers of  Base | 100 = 1  101 = 10  102 = 100  Etc. | 20 = 1  21 = 10 (or 2 decimal)  22 = 100 (or 4 decimal)  Etc. |
| Value of 111 | 111 = 102 + 101 + 100  (100 + 10 + 1) | 111 = 22 + 21 + 20  (Decimal : 4 + 2 + 1 = 7) |

1. Convert the following binary numbers to decimal:  
   1. 11 binary = 3 decimal
   2. 101 binary = 5 decimal
   3. 1010 binary = 10 decimal

1. Convert the following decimal numbers to binary:  
   1. 6 decimal = 110 binary
   2. 13 decimal = 1101 binary
2. Add the following binary numbers. (verify your answers using decimal)

|  |  |
| --- | --- |
| a) 0101 (Decimal 5)  +  0010 (Decimal 2)  0111 | b) 0101 (Decimal 5)  +  1010 (Decimal 10)  1111 |
| c) 0011 (Decimal 3)  +  0010 (Decimal 2)  0101 | d) 0110 (Decimal 6)  +  0011 (Decimal 3)  1001 |

1. List the main features of the following Computer Memory Structures:
   1. Bit

1 binary digit

Used for Boolean data type

Building Block for All computer data and memory

* 1. Byte

8 binary digits

Largest value: 1111 1111 (28 – 1 = 255 Decimal)

Used for Char (character) data type

26 lower case letters + 26 uppercase letters  
+ 10 number symbols + punctuation marks+ Other Stuff  
equal about 130 distinct characters

* 1. Word

16 binary digits (2 bytes)

Largest value: 1111 1111 1111 1111   
(216 – 1 = 65,535 Decimal)

* 1. Integer Data Type

Is 1 Word (16 bits)

But must represent both Positive (+) and Negative (-)

Range: +32767 to -32768

Larger or smaller numbers require a different data type

* 1. Double Word

32 binary digits (4 bytes or 2 words)

Largest value: 232 – 1 = 4 billion approx)

**Research Questions**

1. The Intel 8085 microprocessor was a first generation processor that was used in many early game systems and personal computers. Google “8085 microprocessor architecture” to answer these questions.
   1. Year Introduced

1976

* 1. Size of data bus (in bits)

8-bits

* 1. Largest data number (in binary and decimal)

16 Binary and 65,535 Decimal

* 1. Size of address bus (in bits)

8-bits

* 1. Largest memory address (in binary and decimal)

16 Binary and 65,535 Decimal

1. The Intel 8086 microprocessor was the processor used in the first IBM PCs running the DOS operating system. Google “8086 microprocessor architecture” to answer these questions.
   1. Year Introduced

1979

* 1. Size of data bus (in bits)

16-bits

* 1. Largest data number (in decimal)

1,048,576

* 1. Size of address bus (in bits)

20-bit

* 1. Largest memory address (in decimal)

1,048,576

1. The Intel 80286 microprocessor a common processor used in IBM PCs running the Windows operating system. Google “80286 microprocessor architecture” to answer these questions.
   1. Year Introduced

1982

* 1. Size of data bus (in bits)

16-bits

* 1. Largest data number (in decimal)

65,535

* 1. Size of address bus (in bits)

24-bits

* 1. Largest memory address (in decimal)

65,535

1. The modern PCs run either a 32 bit or 64 bit Windows operating system. Google “32 vs 64 bit” to answer these questions.
   1. How do these systems differ in data capacity? (explain using bits)

64-bit processors are more capable than a 32-bit processor, cause they can handle more data.

* 1. How do these systems differ in memory capacity? (explain using bits)

64-bit processor of storing more computational values, including memory addresses.

* 1. How do these systems differ in hardware requirements?

32-bit processors are perfectly capable of handling a limited amount of RAM (in Windows, 4GB or less) and a 64-bit processor is capable of utilizing much more. In the case of Microsoft Windows, the basic versions of the operating systems put software limitations on the amount of RAM that can be used by applications, but even in the ultimate and professional version of the OS, 4GB is the maximum usable memory the 32-bit version can handle. The latest versions of a 64-bit operating system can increase the capabilities of a processor drastically.

1. Research and explain how negative (-) numbers are represented using bits and how they are stored in computer memory.  
   Negative numbers in any base are represented by prefixing them with a minus (-) sign.

Negative numbers in any base are represented by prefixing them with a minus (-) sign. In computer, numbers are represented only as sequences as bits.

1. Research and explain how floating point (decimal) numbers are represented using bits and how they are stored in computer memory.

Eight digits are used to represent a floating point number : two for the exponent and six for the mantissa. The sign of the mantissa will be represented as + or -, but in the computer it is represented by a bit: 1 means negative, 0 means positive. This representation makes it easy to compare numbers. In memory, a floating point number is represented similarly: One bit has the sign, some bits form the factor as a fixed-precision number (“mantissa”), the remaining bits form the exponent. The exact size of each part depends on the exact floating-point standard you are using.

Sample Program

number = input ( "Enter a 4 digit decimal number" )

index=0

for z in number:

index += 1

print ("Digit", index, "is : ", bin(int(z)))

number = input ( "Enter a 4 digit decimal number" )

index=0

for z in number:

index += 1

print ("Digit", index, "is : ", oct(int(z)))