COMPUTER AND COMMUNICATION TECHNOLOGY

DATA REPRESENTATION

BINARY NUMBER SYSTEM

BIT

- A binary digit is called a bit.
- ➤ It is usually expressed as 0 or 1, the two numbers of binary numbering system.
- ➤ A bit is a smallest unit of information a computer can use.
- ➤ A 16 bit computer would process a series of 16 bits such as 0100111101011000 in one go, repeating the process thousands or millions of times per second.
- ➤ Reading a series of bits is very difficult and to make the process easier they are often displayed in a group of 4 bits 0100 1111 0101 1000.
- This grouping is quite interesting in that a group of four bits can be replaced by a single hexadecimal digit, two groups of 4 bits can be replaced by 2 hexadecimal digits and 4 hexadecimal digits are required to replace 16 bits.

| Binary | 0100 | 1111 | 0101 | 1000 |
|-------------|------|------|------|------|
| hexadecimal | 4 | F | 9 | 8 |

BYTE

- A group of 8 bits is called a byte.
- With 8 bits, there are 256 possible decimal combinations.
- ➤ 1 byte can store one alphabetical letter, single digit or single character symbol such as #.
- Large number of bytes can be expressed in kilobytes and megabytes.

KILOBYTE

- ➤ The value of a kilobyte is 1024.
- \triangleright It is worked out as 2^{10} .
- Normally a kilo refers to a thousand but in computing a kilobyte is equal to 1024.

MEGABYTE

- Likewise 1024 kilobytes is referred to as a megabyte.
- Normally a Mega refers to a million.
- ➤ In computing a megabyte is 1,048,576 bits.
- \triangleright It is worked out as 2^{20} .
- A byte of memory can normally hold:
 - i. a single alphabet letter (upper or lower case).
 - ii. A single number 1 9.
- iii. A symbol $(, _{-}, +, $, #, > etc.$
- iv. A further 127 alternate characters. These could be letters used in foreign languages, lines to produce a box etc.

NUMBER SYSTEM

- The everyday number system we use is denary or decimal number system.
- ➤ In computing three number system are commonly used, binary, hexadecimal and to a lesser extent octal.
- ➤ In denary, binary, octal and hexadecimal systems, the value of any digit depend on its position within the number i.e. which column it is in

How do Numbering Systems Work

- To understand this we will examine the denary system in more detail.
- ➤ Because you are used to the denary system and because it is very easy to multiply, 10 100 etc, you calculate the number in your head.
- Let us use the number 256 as an example.
- ➤ The calculation is automatically done in the following:
 - The most important calculation to do is to work out the position value for that system.
 - o The position value is based on the powers of the number system base value.

| Power of the base | 10^{4} | 10^{3} | 10^{2} | 10^1 | 10^{0} |
|-------------------|-------------|----------|----------|--------|----------|
| calculation | 10x10x10x10 | 10x10x10 | 10x10 | 10 | 1 |
| =position value | 10,000 | 1000 | 100 | 10 | 1 |

- > Write down the position value for the number system you are using so for denary we would write
- Position value 10,000 1000 100 10 1.
- > Underneath the correct value, write down your number
- > Position value 10,000 1000 100 10 1
- Enter number 2 5 6
- > The calculation that is done is
- ➤ Position value 10,000 1000 100 10 1
- Enter number 2 5 6
- Required calculation 2x100 5x10 6x1
- ➤ This equals 200 50 6
- Add the three results 200 + 50 + 6 = 256
- You can convert any number to denary system using this calculation.
- > Ensure you use the positional number of the system you are using.
- A byte of memory can store number in the range of 0 to 255.
- ➤ Denary number system uses numbers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 thus 10 numbers.
- ➤ Hence base 10
- Adding 1 to 9 we must introduce another column to the left i.e. 10.

Binary Number System (base 2)

- Uses a 0 and a 1
- ➤ Hence base 2.
- ➤ Binary numbering is the numbering system that is used by computers.
- ➤ The positional values are shown below.
- \triangleright Power of the base 2^4 2^3 2^2 2^1 2^0
- ➤ Positional values 16 8 4 2 1

- A byte of memory can store a number in the range of 00000000 to 111111111.
- Numbers are often displayed in groups of 4s, as follows, to make them easier to read: 0000 0000 to 1111 1111.

How can you Convert from Binary to Denary

For example convert 1011 binary to denary.

➤ Position value 16 8 4 2 1

 \triangleright Enter number 1 0 1 1

➤ Required calculation 8x1 4x0 2x1 1x1

 \triangleright This equals 8 0 2 1

Add the three results 8+0+2+1=11.

ightharpoonup Therefore $1011_2 = 11_{10}$

Hexadecimal Numbering System (Base 16)

- Uses numbers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
- ➤ Hence 16 numbers and base 16.
- \triangleright By using letters A F, a single digit requires a single position (column)

Power of the base 16^4 16^3 16^2 16^1 16^0

Positional value 65536 4096 256 16 1

- ➤ A byte of memory can store the number in the range of 00 to FF.
- ➤ Hexadecimal number are of displayed in the group of two to make them easier to read.
- ➤ For example 10 AF 3C 9F.

Octal (Base 8)

- > Uses numbers 0, 1, 2, 3, 4, 5, 6, 7,
- > Thus 8 numbers
- ➤ Hence base 8.
- ➤ A byte of memory can store an octal number in the range of 0 to 377.

 \triangleright Power of the base 8^4 8^3 8^2 8^1

➤ Position value 4096 512 64 1

Binary Addition

$$0 + 0 = 0$$
,

$$0+1=1$$
,

$$1 + 0 = 1$$
,

$$1 + 0 = 1$$
, $1 + 1 = 10$.

Consider the following:

+01011101010

11111

Exercise:

Perform the following binary additions

+111011

Subtracting Using Complement

- > Subtraction in any number system can be done through the use of a complement.
- A complement is a number that is used to represent the negative of a number.
- When two numbers are subtracted, the complement of a subtrahend can be added to a minuend to obtain the difference.
- When this method is used, the addition will produce a higher order (left most) one in the result (a "carry") which must be dropped.
- > This is how a computer performs subtraction.
- > To understand the complements consider a mechanical register such as mileage indicator being rotated backwards.
- A five digit register approaching and passing through 0 would read as follows
- > 00005
- > 00004
- **>** 00003
- > 00002
- > 00001
- > 99999
- > 99998
- > 99997
- > Etc

- ➤ It should be clear that the number 99998 correspond to -2.
- Further if we add
- > 00005
- **>** +99998
- **>** 100003
- And ignore carry to the left, we have effectively formed an operation of subtraction 5-2=3.
- The number 99998 is called a tens complement of 2.
- ➤ The tens complement of any decimal number may be formed by subtracting each digit from the number 9, then adding 1 to the least significant digit of the number formed.
- ➤ In the example above:
 - 1. First each digit of subtrahend was subtracted from 9 (this preliminary value is called nines complement of the subtrahend

| 9 | 9 | 9 | 9 | 9 |
|-----------|----|----|----|----|
| <u>-0</u> | -0 | -0 | -0 | -2 |
| 9 | 9 | 9 | 9 | 8 |

2. We now add 1 to the nines complement to get the tens complement.

| 9 | 9 | 9 | 9 | 7 |
|---|---|---|---|---|
| + | | | | 1 |
| 9 | 9 | 9 | 9 | 8 |

3. The tens complement was added to the minuend giving 100003. The leading carry was dropped, effectively performing subtraction 00005 - 00002 = 00003.

$$00005$$
 $+99998$
 100003

- \triangleright Another example consider 4589 322.
- 1. First compute the nines complement

| 9 | 9 | 9 | 9 | |
|-----------|----|----|----|--|
| <u>-0</u> | -3 | -2 | -2 | |
| 9 | 6 | 7 | 7 | |

2. Add 1 to nines complement

3. Add tens complement of subtrahend to minuend giving

Drop the leading 1 and the answer is 4267.

Exercise:

Try the following problems:

Binary Subtraction

- ➤ We use complement method to compute subtraction in binary.
- > Steps:
 - 1. Compute 1s complement subtract each digit of subtrahend from 1. A short cut for doing this is simply to reverse the digits.
 - 2. Add 1 to the 1s complement of the subtrahend to get 2s complement.
 - 3. Add 2s complement of subtrahend to minuend and drop the higher order 1. This is your difference.

> Example : compute 11010101₂ - 1001011₂

1. 1 1 1 1 1 1 1 1
$$\frac{-0}{1}$$
 $\frac{-0}{1}$ $\frac{-1}{1}$ $\frac{-0}{1}$ $\frac{-1}{1}$ $\frac{-0}{1}$ $\frac{-1}{1}$ 0 1 0 0

| 2. | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
|----|-----------|---|---|---|---|----|---|---|
| | + | | | | | | | 1 |
| | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 3. | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| | <u>+1</u> | 0 | 1 | 1 | 0 | 11 | 0 | 1 |
| | 1 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

- \triangleright Example 1 1 1 1 10 1 1₂ 1 1 0 0 0 0 0 1₂
- 1. Original number 1 1 0 0 0 0 0 1 1s complement 0 0 1 1 1 1 1 0
- 2. Add 1 to get the 2s complement

3. Add 2s complement of the subtrahend to minuend.

4. Ignore the carry the answer is 0 0 1 1 1 0 1 0

Exercise: carry out the following subtractions

$$a. \quad 1 \; 1 \; 1 \; 1 \; 1 \; 0 \; 1 \; 1_2 - 1 \; 0 \; 1 \; 0 \; 1 \; 0 \; 1 \; 0_2 \qquad \qquad b. \qquad \qquad 1 \; 0 \; 0 \; 1 \; 0 \; 0_2 - 1 \; 1 \; 1 \; 0 \; 1_2$$

Octal Addition

➤ Octal addition is performed like decimal addition

> Example

$$543_8 + 121_8$$

> 543

> Example $7652_8 + 4574_8$ > 7 5 2 6 7 5 **>** <u>+4</u> 4 4 4 **>** 1 4 6 12/8 12/8 12/8

Exercise:

a.
$$5430_8 + 3241_8$$
 b. $6405_8 + 1234_8$

Octal Subtraction

Example: compute 7526₈ - 3142₈

> 7526

→ <u>-3142</u>

> 4364

> Example: compute 545₈ - 14₈

> 545

> <u>- 14</u>

> 531

Hexadecimal Addition

➤ One consideration is that when the result of addition is between 10 and 15, a corresponding letter A through F must be written in the result.

 \triangleright Example: $195_{16} + 319_{16}$

▶ 195

→ +3 1 9

> 4 A E

Example:

 $3A2_{16} + 41C_{16}$

> 3 A(10) 2

> +4 1 C(12)

> 7 B Ε

 \triangleright

Example:

 $DEB_{16} + 10E_{16}$

➤ D(13) E(14) B(11)

> +1 0 E(14)

> E F 9

Example:

 $8F97_{16} - D54C_{16}$

> 8 F(15) 9 7

 \rightarrow + D(13) 5 4 C(12)

▶ 16 4 E 3

Exercise: compute the following:

a.
$$BED_{16} + 2A9_{16}$$

b.
$$DEED_{16} + BEEF_{16}$$

Hexadecimal Subtraction

Example:

compute ABED₁₆ – 1FAD₁₆

➤ A(10) B(11) E (14) D(13)

F(15) A(10) D(13)

 \mathbf{C} > 8 4 0

> Example:

compute FEED₁₆ – DAF3₁₆

> F(15) E(14) E(14) D(13)

 \rightarrow -D(13) A(10) F(15) 3

> 2 3 F

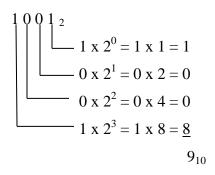
 \triangleright Exercise: compute: a. 98AE₁₆ – 1FEE₁₆

b. $B6A1_{16} - 8B12_{16}$

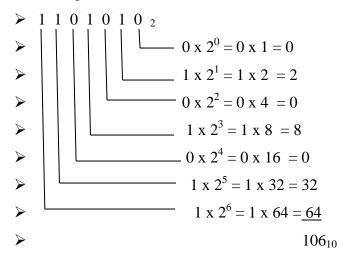
Number System Conversion

Converting a binary number to a decimal number.

Example: convert 1001₂ to decimal.



> Example: convert 1101010 to decimal



Exercise: convert a. 1100110_2 b. 11111001_2 to decimal.

- > Converting decimal number to binary using remainder method.
 - 1. Divide the decimal number by the base (in case of binary divide by 2)
 - 2. Indicate the remainder to the right.
 - 3. Continue dividing each quotient (indicating the remainder) until the divide operation produces a zero quotient.

 \triangleright Example: convert 99₁₀ to binary.

The answer is bottom up 1100011₂

> Example convert 13₁₀ to binary

$$13_{10} = 1101_2$$

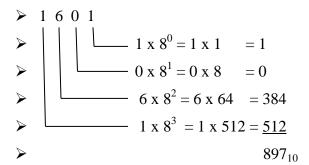
Exercise: convert the following numbers to binary: a. 49_{10} b. 21_{10}

- > Converting octal number to decimal number
- > Example: convert 367₈ to decimal

>
$$367$$

> $6x8^{0} = 7x1 = 7$
> $6x8^{1} = 6x8 = 48$
> $3x8^{2} = 3x64 = 192$
> 247_{10}

> Example: convert 1601₈ to decimal.



Exercise: convert the following numbers to decimal: a. 536₈ b. 1163₈

- > Converting decimal number to octal number.
 - 1. Divide the decimal number by the base (in this case of octal, divide by 8).
 - 2. Indicate the remainder to the right.
 - 3. Continue dividing into each quotient (and indicating the remainder) until the division operation produces a zero quotient.
- > The base 8 number is the numeric remainder reading from the last division to the first.

 \triangleright Example: convert 465₁₀ to octal.

$$ightharpoonup 465_{10} = 741_8$$

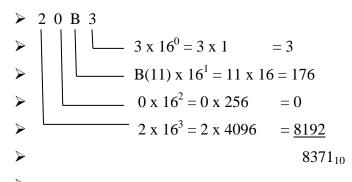
Example: convert 2548₁₀ to octal.

r

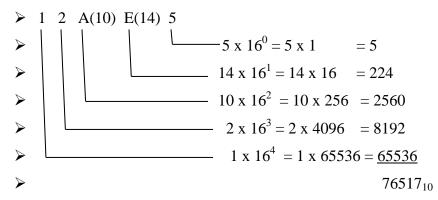
$$\triangleright$$
 2548₁₀ = 4764₈

Exercise: convert the following decimal numbers to octal: a. 3002 b. 6512.

- > Converting hexadecimal number to decimal number.
- Example: convert 20B3₁₆ to decimal



> Example: convert 12AE5₁₆ to decimal.



Exercise: Convert the following numbers to decimal: a. $243F_{16}$ b. BEEF₁₆

- > Converting decimal number to hexadecimal
- ➤ Example convert 9263₁₀ to hexadecimal

$$\triangleright$$
 9263₁₀ = 242F₁₆

Example; convert 4259₁₀ to hexadecimal.

 \rightarrow 4259₁₀ = 10A3₁₆

Exercise: convert the following numbers to hexadecimal: a. 69498_{10} b. 114267_{10}

- ➤ Converting binary to hexadecimal or vice versa
- Four binary digits are equivalent to one hexadecimal digit
- > Divide the binary number into groups of 4 starting from the right.
- ➤ If the left most group has less than 4 digits, put necessary number of leading zeros on the left.
- For each group, write equivalent hexadecimal digit

Example: convert 1101001101110111₂

> 1011 0011 0111 0111

➤ D 3 7 7 (Hex)

Example: convert 101101111₂ to hexadecimal

> 0001 0110 1111

➤ 1 6 F (Hex)

 \triangleright

Example: convert 1BE9₁₆ to binary.

► Hex 1 B E

> Binary 001 1011 1110 1001

Data representation

- Example: convert B0A₁₆ to binary
 Hex B 0 A
 Binary 1011 0000 1010
- Converting from octal to binary or from binary to octal.
- > Three binary digits are equivalent to one octal digit.
- > To convert from binary to octal, divide the binary number into group of three digits starting from the right.
- ➤ If the left most group has less than three digits, put necessary leading zeros.
- For each group of three bits, write a single corresponding octal digit.

- \triangleright Example; convert 1101001101110111₂ to octal.
- > Binary 001 101 001 101 110 111
- > Octal 1 5 1 5 6 7

- Example: convert 10110111₂ to octal.
- > Binary 101 101 111
- ➤ Octal 5 5 7

- > Example: convert 1764₈ to binary.
- ➤ Octal 1 7 6 4
- > Binary 001 111 110 0100

- \triangleright Example: convert 731₈ to binary.
- ➤ Octal 7 3 1
- > Binary 111 011 001

ASCII

- ➤ ASCII represents American standard code for information interchange.
- This is a character encoding scheme originally based on the English alphabet.
- ➤ ASCII codes represent text in computers, communication equipment and other devices that use text.

- ➤ Most modern characters schemes are based on ASCII, though they support many additional characters.
- ➤ ASCII include the definition of 128 characters (many now obsolete) that affect how text or space are processed and 95 printable characters including the space.

Table of ASCII codes

EBCDIC

- ➤ EBCDIC stands for extended binary coded decimal interchange code
- ➤ It is an 8 bit character encoding used mainly on IBM mainframe and IBM midrange computer operating system.
- ➤ It descends from the codes used with punch cards and the corresponding six bit binary coded decimal code, used with most of the IBM computer peripherals of late 1950s and 1960s.
- ➤ It is also employed on various non IBM platforms such as Fujistu.-Siemens BS2000/OSD/ix and the Unisys MCP.
- ➤ It was created to extend the existing binary coded decimal (BCD) interchange code.
- ➤ While IBM was a proponent of ASCII, it did not have time to prepare for ASCII peripherals.

UNICODE

- ➤ Unicode provides a unique number for every character.
- ➤ It developed because of limitations of existing text encoding schemes such as ASCII.
- ➤ ASCII can accommodate up to 256 characters because it uses only a byte for storage of information.
- ➤ ASCII was sufficient when only English characters were encoded.
- ➤ If you consider characters for all other languages in the world, ASCII does not have capacity to accommodate them.
- ➤ Hence UNICODE comes in to solve this problem.
- > Unicode provides a unique number for every character:
 - o No matter what the platform
 - o No matter what the program
 - o No matter what the language

- ➤ Before UNICODE was invented, there were hundreds of different of encoding schemes for assigning these numbers.
- > The encoding schemes also conflicted with each other.
- > That is two encoding schemes can use the same number for two different characters or use different numbers for the same character.
- ➤ UNICODE uses up to four bytes for storage of these numbers that represent the characters.

| ASCII | DEC | Hex | Octal | Binary | ASCII | Dec | Hex | Octal | Binary | ASCII | Dec | Hex | Octal | Binary |
|-------|-----|-----|-------|-----------|----------|------------|----------|-------|-----------|-------|------------|----------|------------|-----------|
| NULL | 000 | 00 | 000 | 0000 0000 | + | 043 | 2B | 053 | 0010 1011 | ٧ | 086 | 56 | 126 | 0101 0110 |
| SOH | 001 | 01 | 001 | 0000 0001 | <u> </u> | 044 | 2C | 054 | 0010 1100 | W | 087 | 57 | 127 | 0101 0111 |
| STX | 002 | 02 | 002 | 0000 0010 | - | 045 | 2D | 055 | 0010 1101 | Х | 088 | 58 | 130 | 0101 1000 |
| ETX | 003 | 03 | 003 | 0000 0011 | | 046 | 2E | 056 | 0010 1110 | Υ | 089 | 59 | 131 | 0101 1001 |
| EOT | 004 | 04 | 004 | 0000 0100 | / | 047 | 2F | 057 | 0010 1111 | Z | 090 | 5A | 132 | 0101 1010 |
| ENQ | 005 | 05 | 005 | 0000 0101 | 0 | 048 | 30 | 060 | 0011 0000 | [| 091 | 5B | 133 | 0101 1011 |
| ACK | 006 | 06 | 006 | 0000 0110 | 1 | 049 | 31 | 061 | 0011 0001 | ١ | 092 | 5C | 134 | 0101 1100 |
| BEL | 007 | 07 | 007 | 0000 0111 | 2 | 050 | 32 | 062 | 0011 0010 |] | 093 | 5D | 135 | 0101 1101 |
| BS | 800 | 08 | 010 | 0000 1000 | 3 | 051 | 33 | 063 | 0011 0011 | ^ | 094 | 5E | 136 | 0101 1110 |
| HT | 009 | 09 | 011 | 0000 1001 | 4 | 052 | 34 | 064 | 0011 0100 | | 095 | 5F | 137 | 0101 1111 |
| LF | 010 | 0A | 012 | 0000 1010 | 5 | 053 | 35 | 065 | 0011 0101 | ` ` | 096 | 60 | 140 | 0110 0000 |
| VT | 011 | 0B | 013 | 0000 1011 | 6 | 054 | 36 | 066 | 0011 0110 | а | 097 | 61 | 141 | 0110 0001 |
| FF | 012 | 0C | 014 | 0000 1100 | 7 | 055 | 37 | 067 | 0011 0111 | b | 098 | 62 | 142 | 0110 0010 |
| CR | 013 | 0D | 015 | 0000 1101 | 8 | 056 | 38 | 070 | 0011 1000 | С | 099 | 63 | 143 | 0110 0011 |
| S0 | 014 | 0E | 016 | 0000 1110 | 9 | 057 | 39 | 071 | 0011 1001 | d | 100 | 64 | 144 | 0110 0100 |
| SI | 015 | 0F | 017 | 0000 1111 | : | 058 | 3A | 072 | 0011 1010 | е | 101 | 65 | 145 | 0110 0101 |
| DLE | 016 | 10 | 020 | 0001 0000 | ; | 059 | 3B | 073 | 0011 1011 | f | 102 | 66 | 146 | 0110 0110 |
| DC1 | 017 | 11 | 021 | 0001 0001 | a | 060 | 3C | 074 | 0011 1100 | g | 103 | 67 | 147 | 0110 0111 |
| DC2 | 018 | 12 | 022 | 0001 0010 | = | 061 | 3D | 075 | 0011 1101 | h | 104 | 68 | 150 | 0110 1000 |
| DC3 | 019 | 13 | 023 | 0001 0011 | > | 062 | 3E | 076 | 0011 1110 | i | 105 | 69 | 151 | 0110 1001 |
| DC4 | 020 | 14 | 024 | 0001 0100 | ? | 063 | 3F | 077 | 0011 1111 | j | 106 | 6A | 152 | 0110 1010 |
| NAK | 021 | 15 | 025 | 0001 0101 | @ | 064 | 40 | 100 | 0100 0000 | k | 107 | 6B | 153 | 0110 1011 |
| SYN | 022 | 16 | 026 | 0001 0110 | Α | 065 | 41 | 101 | 010 00001 | | 108 | 6C | 154 | 0110 1100 |
| ETB | 023 | 17 | 027 | 0001 0111 | В | 066 | 42 | 102 | 0100 0010 | m | 109 | 6D | 155 | 0110 1101 |
| CAN | 024 | 18 | 030 | 0001 1000 | С | 067 | 43 | 103 | 0100 0011 | n | 110 | 6E | 156 | 0110 1110 |
| EM | 025 | 19 | 031 | 0001 1001 | D | 068 | 44 | 104 | 0100 0100 | 0 | 111 | 6F | 157 | 0110 1111 |
| SUB | 026 | 1A | 032 | 0001 1010 | E | 069 | 45 | 105 | 0100 0101 | р | 112 | 70 | 160 | 0111 0000 |
| ESC | 027 | 1B | 033 | 0001 1011 | F | 070 | 46 | 106 | 0100 0110 | q | 113 | 71 | 161 | 0111 0001 |
| FS | 028 | 1C | 034 | 0001 1100 | G | 071 | = | 107 | 0100 0111 | r | 114 | \vdash | 162 | 0111 0010 |
| GS | 029 | 1D | 035 | 0001 1101 | Н | 072 | 48 | 110 | 0100 1000 | S | 115 | 73 | 163 | 0111 0011 |
| RS | 030 | 1E | 036 | 0001 1110 | I | 073 | == | 111 | 0100 1001 | t | 116 | 74 | 164 | 0111 0100 |
| US | 031 | 1F | 037 | 0001 1111 | J | 074 | | 112 | 0100 1010 | u | 117 | 75 | 165 | 0111 0101 |
| space | 032 | 20 | 040 | 0010 0000 | K | 075 | 4B | 113 | 0100 1011 | V | 118 | 76 | 166 | 0111 0110 |
| ! | 033 | 21 | 041 | 0010 0001 | L | 076 | == | 114 | 0100 1100 | W | 119 | 77 | 167 | 0111 0111 |
| " | 034 | 22 | 042 | 0010 0010 | M | 077 | = | 115 | 0100 1110 | X | 120 | 78 | 170 | 0111 1000 |
| # | 035 | 23 | 043 | 0010 0011 | N | 078 079 | 4E | 116 | 0100 1111 | У 7 | 121 | 79 74 | 171 | 0111 1001 |
| \$ | 036 | 24 | 044 | 0010 0100 | 0 | | H | 117 | 0100 1111 | Z | 122 | 7A | 172 | |
| % | 037 | 25 | 045 | 0010 0101 | P | 080 | 50 | 120 | 010 10000 | { | 123 124 | 7B 7C | 173 174 | 0111 1011 |
| & | 038 | 26 | 046 | 0010 0110 | Q R | 081 | 52 | 121 | 0101 0001 | 1 | 124 | 7C 7D | 174 | 0111 1100 |
| ' | 039 | 27 | 047 | 0010 0111 | S | 082 | 53 | 123 | 0101 0010 | } | 126 | 7E | 176 | 0111 1101 |
| (| 040 | 28 | 050 | 0010 1000 | 5 T | 083 | 54 | 123 | 0101 0011 | DEL | 126 | 7E 7F | 176 | 0111 1110 |
|) | 041 | 29 | 051 | 0010 1001 | U | 085 | \vdash | 125 | 0101 0100 | DLL | 14/ | _ / 1 | 1// | 0111 1111 |
| * | 042 | 2A | 052 | 0010 1010 | | 003 | رر | 143 | 2101 0101 | | | | | |