

CSARCH2 Simulation Project
S13 Group 3
BACOSA, JAWALI, MARISTELA, REJANO, SERRATO

Analysis Writeup on IEEE-754 Decimal-64 floating-point converter

I. Problems Encountered

A. Handling the special case involving NaN

The function `get_densely_packed_bcd()` which was initially designed for normal values, encountered difficulties when handling the special case of "NaN" (Not a Number). For instance, if a user input contained characters alongside digits (e.g., "sqrt8"), the function `get_densely_packed_bcd()` would try to process it as a numeric value, leading to potential errors. To address this issue, the group modified the implementation. When processing each character in the input, if the character is a digit, '-', or '.', the modified code assigns the value of 0 to the respective variable. This change ensures that even in cases where non-numeric characters are present in the input, the code can continue execution without raising exceptions, thereby successfully executing the specific function for this case to output the correct values for NaN.

B. Handling the special case involving Zero

The function `convert_four_binary_to_hexadecimal()` manages inputs adhering to the expected format of four binary digits for accurate binary-to-hexadecimal conversion. However, when encountering the special case of 0, the team opted not to engage in normalization or preprocessing. Instead, a separate function was specifically made to yield the correct output for the special case of 0.

II. Screenshots of the test cases used

A. Normal Input (MSD is between 0-7)

Decimal-64 Floating Point Converter

Decimal: 123456789876543

10 raised to: 99

Rounding Type: truncate

Convert

Save Output

Normalized Decimal: 123456789876543
Normalized Exponent: 99
Exponent Prime: 497
Exponent Prime Binary: 0111110001
Sign Bit: 0
Combination Field: 01001
Exponent Extension: 11110001
Coefficient Continuation: 0100110100 1011100111 0001111111 1101111100 1011000011
Binary Output: 0 01001 11110001 0100110100 1011100111 0001111111 1101111100 1011000011
Hexadecimal Output: 27C534B9C7FDF2C3

B. Normal Input (MSD is 8 or 9)

Decimal-64 Floating Point Converter

Decimal: 9876543211234567

10 raised to: 121

Rounding Type: truncate

Convert

Save Output

Normalized Decimal: 9876543211234567
Normalized Exponent: 121
Exponent Prime: 519
Exponent Prime Binary: 1000000111
Sign Bit: 0
Combination Field: 11101
Exponent Extension: 00000111
Coefficient Continuation: 1101111100 1011000011 0100010001 0100110100 1011100111
Binary Output: 0 11101 00000111 1101111100 1011000011 0100010001 0100110100 1011100111
Hexadecimal Output: 741F7CB0D114D2E7

C. Positive Infinity

Decimal-64 Floating Point Converter

Decimal: 123456789876543

10 raised to: 700

Rounding Type: truncate

Convert

Save Output

Normalized Decimal: inf
Normalized Exponent: 700
Exponent Prime:
Exponent Prime Binary:
Sign Bit: 0
Combination Field: 11110
Exponent Extension: 00000000
Coefficient Continuation: 0000000000 0000000000 0000000000 0000000000 0000000000
Binary Output: 0 11110 00000000 0000000000 0000000000 0000000000 0000000000 0000000000
Hexadecimal Output: 7800000000000000

D. Negative Infinity

Decimal-64 Floating Point Converter

Decimal: -9876543211234567

10 raised to: 700

Rounding Type: truncate

Convert

Save Output

Normalized Decimal: -inf
Normalized Exponent: 700
Exponent Prime:
Exponent Prime Binary:
Sign Bit: 1
Combination Field: 11110
Exponent Extension: 00000000
Coefficient Continuation: 0000000000 0000000000 0000000000 0000000000 0000000000
Binary Output: 1 11110 00000000 0000000000 0000000000 0000000000 0000000000 0000000000
Hexadecimal Output: F800000000000000

E. NaN

Decimal-64 Floating Point Converter

Decimal: sqrt(-8)

10 raised to: 23

Rounding Type: nearest

Convert

Save Output

Normalized Decimal:
Normalized Exponent:
Exponent Prime:
Exponent Prime Binary:
Sign Bit: 0
Combination Field: 11111
Exponent Extension: 00000000
Coefficient Continuation: 00000000 00000000 00000000 00000000 00000000
Binary Output: 0 11111 00000000 00000000 00000000 00000000 00000000 00000000
Hexadecimal Output: 7C00000000000000

F. Zero case

Decimal-64 Floating Point Converter

Decimal: 0

10 raised to: 0

Rounding Type: nearest

Convert

Save Output

Normalized Decimal: 0000000000000000
Normalized Exponent: 0
Exponent Prime: 398
Exponent Prime Binary: 0110001110
Sign Bit: 0
Combination Field: 01000
Exponent Extension: 10001110
Coefficient Continuation: 00000000 00000000 00000000 00000000 00000000
Binary Output: 0 01000 10001110 00000000 00000000 00000000 00000000 00000000
Hexadecimal Output: 2238000000000000

G. Zero case (exponent is non-zero)

The screenshot shows the 'Decimal-64 Floating Point Converter' window. The 'Decimal' input field contains '0'. The '10 raised to:' input field contains '99'. The 'Rounding Type' dropdown is set to 'nearest'. The 'Convert' button is highlighted. Below the input fields, the output fields are populated with the following values:

```
Normalized Decimal: 0000000000000000
Normalized Exponent: 0
Exponent Prime: 398
Exponent Prime Binary: 0110001110
Sign Bit: 0
Combination Field: 01000
Exponent Extension: 10001110
Coefficient Continuation: 00000000 00000000 00000000 00000000
Binary Output: 0 01000 10001110 00000000 00000000 00000000 00000000
Hexadecimal Output: 2238000000000000
```

H. Less than 16 digits input

The screenshot shows the 'Decimal-64 Floating Point Converter' window. The 'Decimal' input field contains '123456789'. The '10 raised to:' input field contains '24'. The 'Rounding Type' dropdown is set to 'truncate'. The 'Convert' button is highlighted. Below the input fields, the output fields are populated with the following values:

```
Normalized Decimal: 0000000123456789
Normalized Exponent: 24
Exponent Prime: 422
Exponent Prime Binary: 0110100110
Sign Bit: 0
Combination Field: 01000
Exponent Extension: 10100110
Coefficient Continuation: 0000000000 0000000000 0010100011 1001010110 1111001111
Binary Output: 0 01000 10100110 0000000000 0000000000 0010100011 1001010110 1111001111
Hexadecimal Output: 2298000000A395BCF
```

I. More than 16 digits input (tests the rounding feature)

- Truncate

The screenshot shows the 'Decimal-64 Floating Point Converter' window. The 'Decimal' input field contains '1234567899876543999'. The '10 raised to:' input field contains '56'. The 'Rounding Type' dropdown is set to 'truncate'. The 'Convert' button is highlighted. Below the input fields, the output fields are populated with the following values:

```
Normalized Decimal: 1234567899876543
Normalized Exponent: 59
Exponent Prime: 457
Exponent Prime Binary: 0111001001
Sign Bit: 0
Combination Field: 01001
Exponent Extension: 11001001
Coefficient Continuation: 0100110100 1011100111 0001111111 1101111100 1011000011
Binary Output: 0 01001 11001001 0100110100 1011100111 0001111111 1101111100 1011000011
Hexadecimal Output: 272534B9C7FDF2C3
```

- Floor

Decimal-64 Floating Point Converter

Decimal:1234567899876543999

10 raised to:56

Rounding Type:floor

Convert

Save Output

Normalized Decimal:1234567899876543

Normalized Exponent:59

Exponent Prime:457

Exponent Prime Binary:0111001001

Sign Bit:0

Combination Field:01001

Exponent Extension:11001001

Coefficient Continuation:0100110100 1011100111 0001111111 1101111100 1011000011

Binary Output:0 01001 11001001 0100110100 1011100111 0001111111 1101111100 1011000011

Hexadecimal Output:272534B9C7FDF2C3

- Ceiling

Decimal-64 Floating Point Converter

Decimal:1234567899876543999

10 raised to:56

Rounding Type:ceiling

Convert

Save Output

Normalized Decimal:1234567899876543

Normalized Exponent:59

Exponent Prime:457

Exponent Prime Binary:0111001001

Sign Bit:0

Combination Field:01001

Exponent Extension:11001001

Coefficient Continuation:0100110100 1011100111 0001111111 1101111100 1011000011

Binary Output:0 01001 11001001 0100110100 1011100111 0001111111 1101111100 1011000011

Hexadecimal Output:272534B9C7FDF2C3

- Nearest

Decimal-64 Floating Point Converter

Decimal:1234567899876543999

10 raised to:56

Rounding Type:nearest

Convert

Save Output

Normalized Decimal:1234567899876544

Normalized Exponent:59

Exponent Prime:457

Exponent Prime Binary:0111001001

Sign Bit:0

Combination Field:01001

Exponent Extension:11001001

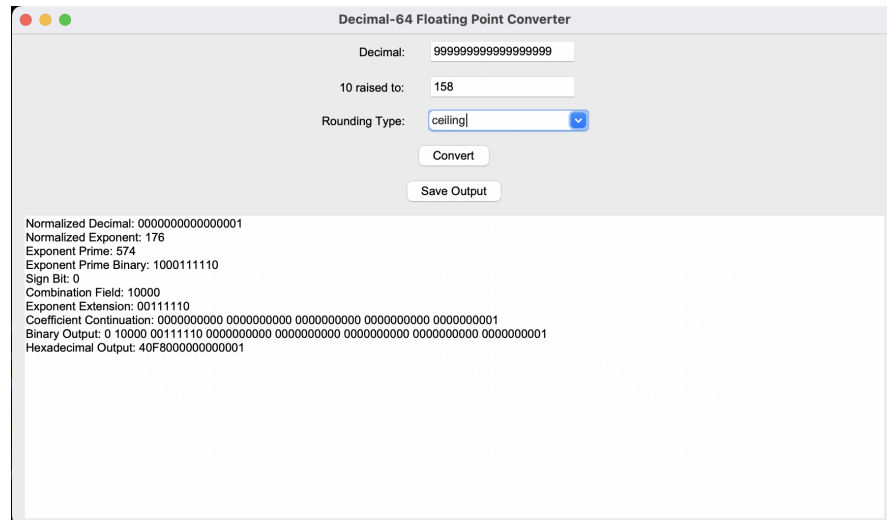
Coefficient Continuation:0100110100 1011100111 0001111111 1101111100 1011000100

Binary Output:0 01001 11001001 0100110100 1011100111 0001111111 1101111100 1011000100

Hexadecimal Output:272534B9C7FDF2C4

J. More than 16 digits of 9s and testing the rounding

- Ceiling



Decimal-64 Floating Point Converter

Decimal: 9999999999999999

10 raised to: 158

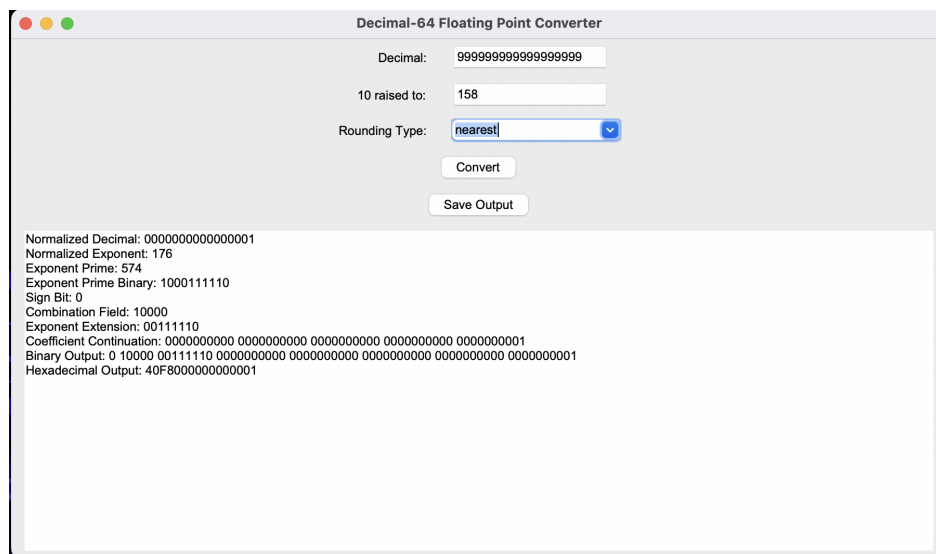
Rounding Type: ceiling

Convert

Save Output

Normalized Decimal: 0000000000000001
Normalized Exponent: 176
Exponent Prime: 574
Exponent Prime Binary: 1000111110
Sign Bit: 0
Combination Field: 10000
Exponent Extension: 00111110
Coefficient Continuation: 0000000000 0000000000 0000000000 0000000000 0000000000 0000000001
Binary Output: 0 10000 00111110 0000000000 0000000000 0000000000 0000000000 0000000001
Hexadecimal Output: 40F8000000000001

- Nearest



Decimal-64 Floating Point Converter

Decimal: 9999999999999999

10 raised to: 158

Rounding Type: nearest

Convert

Save Output

Normalized Decimal: 0000000000000001
Normalized Exponent: 176
Exponent Prime: 574
Exponent Prime Binary: 1000111110
Sign Bit: 0
Combination Field: 10000
Exponent Extension: 00111110
Coefficient Continuation: 0000000000 0000000000 0000000000 0000000000 0000000000 0000000001
Binary Output: 0 10000 00111110 0000000000 0000000000 0000000000 0000000000 0000000001
Hexadecimal Output: 40F8000000000001

K. Input with a decimal point

The screenshot shows a window titled "Decimal-64 Floating Point Converter". It has input fields for "Decimal:" (1233.458), "10 raised to:" (16), and a "Rounding Type:" dropdown menu set to "truncate". Below these are "Convert" and "Save Output" buttons. The output section displays the following information:

```
Normalized Decimal: 000000001233458
Normalized Exponent: 13
Exponent Prime: 411
Exponent Prime Binary: 0110011011
Sign Bit: 0
Combination Field: 01000
Exponent Extension: 10011011
Coefficient Continuation: 000000000 000000000 000000001 0100110011 1001011000
Binary Output: 0 01000 10011011 000000000 000000000 000000001 0100110011 1001011000
Hexadecimal Output: 226C0000014CE58
```

L. Denormalized input (exponent is less than -398)

The screenshot shows the same application window with "Decimal:" set to 12345678, "10 raised to:" set to -885, and "Rounding Type:" set to "truncate". The output section displays the following information:

```
Normalized Decimal: 000000000000000
Normalized Exponent: 0
Exponent Prime: 398
Exponent Prime Binary: 0110001110
Sign Bit: 0
Combination Field: 01000
Exponent Extension: 10001110
Coefficient Continuation: 000000000 000000000 000000000 000000000 000000000
Binary Output: 0 01000 10001110 000000000 000000000 000000000 000000000 000000000
Hexadecimal Output: 2238000000000000
```

M. Sample output saved in a text file

The screenshot shows a text file named "test1.txt" containing the following text:

```
Normalized Decimal: 1234567899876543
Normalized Exponent: 99
Exponent Prime: 497
Exponent Prime Binary: 0111110001
Sign Bit: 0
Combination Field: 01001
Exponent Extension: 11110001
Coefficient Continuation: 0100110100 1011100111 0001111111 1101111100 1011000011
Binary Output: 0 01001 11110001 0100110100 1011100111 0001111111 1101111100 1011000011
Hexadecimal Output: 27C534B9C7FDF2C3
```

III. Pertinent links

- A. [A short video showing program compilation is correct and shows all test cases that will cover the specifications](#)
- B. [GitHub Repository](#)