```python
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 28 22:29:51 2024

@author: dnick
"""
import numpy as np


#
# embedding_generator.py
#

def embedding(z_vals, precision, bits, max_len):
    z_prime_mags = np.abs((10**(precision)*z_vals)).astype(np.int16)
    bses = (z_vals < 0).astype(np.int8)
    embedded_reps = []
    for z, bs, b in zip(z_prime_mags, bses, bits):
        b_z = f'{z:b}'.zfill(max_len)
        b_bin = f'{b}' + f'{bs}' + b_z
        embedded_reps.append(int(b_bin,2))
    return embedded_reps


#
# Get the maximum possible length of the binary representations.
def get_max_len(s,precision):
    s *= 10**(precision)
    z_mag_max = f'{int(s):b}'
    return len(z_mag_max)

if __name__=='__main__':
    s = 9.999
    precision = 3
    max_len = get_max_len(s,precision)
    z_vals = np.array([9.999,-9.999,1.234,1.235,1.234,1.235,0,0])
    bits = np.array([1,1,1,1,0,0,1,0])
    embedded_representations = embedding(z_vals, precision, bits, max_len)

    for b, z, e in zip(bits, z_vals, embedded_representations):
        print(f'b = {b}, z = {z} --> b_embed = {e}')
```