2. Previous work

Embedded speech recognition has become more and more popular, hence there are plenty related work. Brian Delaney et al. outline some optimization techniques to extract features [1]. They demonstrate that fixed-point arithmetic is an important step to reduce the computational complexity and the power consumption on embedded systems.

Laura Miyakawa and Lee Hetherington also emphasize the importance of fixed-point arithmetic [2], but they leave the feature extraction on the embedded system and the recognizer on a back-end server. On the other hand, our study leaves both on the embedded system.

There are other researches dealing with this issue on hardware designs [3, 4, 5, 6]. Although we focus on software designs in this research, they still give us valuable information. For example, we need to use some trigonometric and logarithmic functions in order to extract speech features. Building look-up tables [4] is an alternative method to perform these functions. The function values are stored directly in memory. The function argument addresses a memory location, and the contents yield the function value. Though it has the advantages of simple structure and high speed of data access, exponential growth in memory requirements is the drawback. Therefore a hybrid table is used to solve this problem, and the detail would be discussed in Sec. 3.2.

A particular class developed in the C++ language could be used to achieve fixed-point computations [5]. We define a class, called Int, to do the similar job. It could help us to detect where overflow occurs. Decreasing the amount of computation and memory requirement in decoding process of speech recognition is also a critical part [6]. This research adopts some techniques to fulfill it. The system concept is

approximate to [7] but more complicated since the extracted features are 39-dimensions MFCC in this research.