Abstract: The time scale modification (TSM) of speech is concerned with the compressing or expanding of audio signals in the time domain without affecting the signals pitch or naturalness. Conversely, the frequency scale modification (FSM) of speech is concerned with altering the pitch and formants of a signal without changing the signal duration.

This paper describes a hardware implemented and optimized TSM/FSM system. Biomedical speech related applications for such a system include accelerated aural reading for the blind and improved speech recognition – In a voice controlled robotic system for the disabled, the speech can be effectively “slowed down” to improve the recognition rate. Other applications of the system include speech synthesis, foreign language learning, audio typing, and voice transformation.

Keywords: TSM, FSM, VLSI

I. INTRODUCTION

Time-Scale Modification (TSM) of speech consists of modifying the speed of the speech segment without affecting its naturalness or pitch. Conversely, Frequency-Scale Modification (FSM) of speech consists of modifying the pitch of the speech without changing the duration of the speech segment. Much research has been done in this type of speech processing since the early twentieth century and so a variety of algorithms exist. It is recognized however that some types of speech are more easily modified than others. Voiced speech segments are quasi-periodic in the time domain and in the frequency domain possess clearly defined pitch and harmonics. This is due to the vibration of the vocal cords while air is forced through the glottis. Typical voiced sounds are vowel sounds and broad consonant sounds such as ‘y’. In contrast, unvoiced speech is spectrally noisy since there is no vocal cord vibration and the sound is instead produced in the oral cavity with the aid of the teeth and lips. Examples of unvoiced sounds are ‘s’ sounds and ‘t’ sounds. TSM/FSM algorithms exist to preserve the periodicity (continuity) and hence quality of voiced speech types and indeed music. The noisy nature of unvoiced speech means it is therefore unnecessary to employ algorithms for time-scale or frequency-scale modification. Distinctions between voiced speech and unvoiced speech may be based upon signal energy content and upon the signal’s zero-crossings rate (the number of sign changes in a given period).

Algorithms for TSM and FSM fall broadly into three categories: time-domain techniques; frequency-domain techniques; parametric techniques. The level of output quality across the three categories is similar, however the time domain category is the most efficient in terms of computational burden [1]. By far the most widely used algorithm within this category is synchronized overlap-add (SOLA)[2] and its close relation, pitch synchronized overlap-add (PSOLA) [3]. However, the adaptive overlap-add (AOLA) algorithm due to Lawlor achieves similar quality with a saving in computational burden of an order of magnitude less [1]. Hence, this algorithm was selected over the others for implementation, since power consumption in a CMOS device is a strong function of switching activity and as such, the number of operations should be kept to minimum.

TSM and FSM are intrinsically related. If, for example, a speech segment is time-scale modified by a factor of two, the resultant speech segment is twice as long as the original segment. Playing this segment at double speed results in a speech segment that is the same duration as the original segment but its frequency content has doubled.

The possible applications for TSM/FSM algorithms are broad ranging. Possible speech related applications include speech synthesis, foreign language learning, audio typing, accelerated aural reading for the blind, voice conversion, improved speech recognition, film/speech synchronisation, audio compression and noise reduction.

II. METHODOLOGY

For the modification of voiced speech the AOLA algorithm is used. The algorithm uses a fixed length rectangular stepping window and a simple peak alignment criterion to perform the overlap-add. Adjusting the overlap distance has the effect of increasing or decreasing the amount of expansion or compression. Overlap-adding in this way results in a local natural expansion factor or natural scaling factor. This factor is given by the ratio of the lengths of the original waveform and the newly formed synthetic segment and shall be denoted \( \alpha_{uv} \).
Figure 1: Steps in the AOLA algorithm.

Figure 1 [1] shows the alignment and output waveform synthesis procedures of AOLA for time-scale expansion. In the figure, the frame boundaries are marked by the dashed lines, \( x(n) \) is the input waveform and \( y(n) \) is the output waveform. Figure 1 (a) is the original segment to be expanded and is windowed with a rectangular window length \( w \). In Figure 1 (b) the original segment is duplicated and the peak alignment procedure described earlier is performed about the dashed line. The result is shown in Figure 1 (c). It should be noted that this segment has been expanded by the natural expansion factor \( \alpha_{ne} \), and the length of the segment is now \( w \cdot \alpha_{ne} \). In Figure 1 (d) the input window is now advanced by a time step. Where this step ends coincides with the end of the next window to be expanded as indicated in Figure 1 (e). The segment preceding this new window is considered as expanded already and can be output. In (f), the expanded window \( w \cdot \alpha_{ne} \) is shown to be the accumulation of \( A \) expanded steps. From this the following equation is derived:

\[
\text{step} = w \cdot \frac{1 - \alpha_{ne}}{1 - \alpha_{de}} \quad (1)
\]

There may be a discrepancy between the natural scaling factor \( \alpha_{ne} \) and the desired scaling factor \( \alpha_{de} \). Therefore, \( \text{step} \) has to be updated for every advance step of the analysis window. The whole process repeats iteratively until the desired scaling factor is met. The AOLA algorithm accurately adapts to the local signal characteristics and ensures the signal is expanded by the desired scaling factor \( \alpha_{de} \).

For time-scale compression the approach is similar. In this case the peaks or troughs are aligned as before but the signal to the left and right of the central overlapping region are discarded leaving a compressed segment. If the input segment has a natural compression factor of \( \alpha_{nc} \) and the desired compression factor of \( \alpha_{dc} \), (5) becomes:

\[
\text{step} = w \cdot \frac{1 - \alpha_{nc}}{1 - \alpha_{dc}} \quad (3)
\]

The algorithm can be recapitulated in the following three steps: 1. Isolate appropriate peaks; 2. Perform the overlap and determine the natural scaling factor; 3. Adapt as necessary and repeat.

The modification of unvoiced speech is a far simpler task. To achieve compression, the speech segment can simply be truncated as desired. As the frame boundaries are noisy, there will be no loss of continuity. In the case of expansion, a window of suitable length may be copied and appended to the end of the frame. As before, the integrity of the frame boundaries is preserved.

To ensure accuracy and efficiency, the system must discern between unvoiced speech and voiced speech. This distinction is based upon the short-term energy content and the zero-crossings rate mentioned earlier. In the case of short-term energy, a calculation is made of the energy content within a signal. Generally this energy content will be greater for a voiced speech segment than for an unvoiced segment of similar length. The total energy in a frame is given by the equation:

\[
\sum_{n=1}^{N} s(n)^2 \quad (4)
\]

Where \( N \) is the number of samples in the frame. Once the energy in a frame is known, it is compared with a reference value to decide if the energy present is indicative of voiced or unvoiced speech.

Since there will also be more energy in a voiced phrase that is louder than in the same phrase uttered softly, the zero-crossings decision mechanism is necessary. Unvoiced speech is spectrally noisy and will cross the time-domain origin a far greater number of times than voiced speech for a given segment. For a 20ms clean speech segment the crossing rate was found to be approximately 26 for voiced speech and more than 100 for unvoiced. These figures are used to determine whether the segment is voiced or not. Using both the methods outlined above, a more accurate decision is made.

III. IMPLEMENTATION

The system was coded and tested using VHDL. All VHDL code was synthesized and tested in the Synopsys Design environment. The system can be broken down into three major blocks of circuitry: 1. AOLA circuit; 2. unvoiced modification circuit; 3. decision circuit.
In addition there is an input RAM structure, a RAM structure to hold the window of operation (W RAM) and a Controller module which synchronizes the system and controls system resets. The RAM structures are implemented as latch-multiplexer structures. These structures are more easily customizable, and are preferred to the RAM structures available from the existing libraries.

The AOLA algorithm is implemented in three modules corresponding to the three steps outlined earlier. The modules move samples as appropriate within the W RAM structure to perform the overlap-add, as well as performing the step calculations of the algorithm. These latter operations include a number of multiplications and divisions. The divider employed operates on a subtract-shift-divide basis. The multiplier used is small, and operates within a single clock cycle.

The unvoiced modification circuit consists of a multiplier to establish $\alpha_{de}$ (desired scaling factor) in terms of the amount of samples (framesize x $\alpha_{de}$), and a circular counter device which iteratively counts out the stored frame, $\alpha_{de}$ number of times.

The decision circuit consists of three modules, one for each of the decision mechanisms outlined earlier, and one to examine the results and make the decision. The modules operate on a running calculation basis. This allows a decision to be made at the input section as the input buffer is being filled with a reservoir of samples for working on. The input itself is a serial 8kHz sampled speech signal.

### IV. RESULTS

The system was tested with 8kHz quantised 8-bit speech samples. It was synthesized using the European Silicon Structures 0.7$\mu$m technology. The silicon area is shown in Figure 3. The total silicon area was 7518380$\mu$m$^2$ or 7.5mm$^2$, small enough for handheld devices.

![Figure 2: System block diagram.](image)

![Figure 3: Silicon area of individual modules.](image)

The following selected results show input and output waveforms for compression and expansion of both unvoiced and voiced speech. All inputs shown have a signal-to-noise ratio of 10. In the figure captions $\alpha_d$ is the desired scaling factor.

![Figure 4: Unvoiced compression input (47.5ms) and output (35.625 ms), $\alpha_d = 0.75$.](image)

![Figure 5: Unvoiced expansion input (47.5ms) and output (76 ms), $\alpha_d = 1.6$.](image)
The Circuit timing of the individual modules is shown in terms of propagation delays from input to output in the following table.

<table>
<thead>
<tr>
<th>Module</th>
<th>Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>2.08</td>
</tr>
<tr>
<td>Input RAM</td>
<td>2.10</td>
</tr>
<tr>
<td>Address counter</td>
<td>2.07</td>
</tr>
<tr>
<td>Decision circuit</td>
<td>2.08</td>
</tr>
<tr>
<td>Truncatenator</td>
<td>2.08</td>
</tr>
<tr>
<td>Peakpicker</td>
<td>1.98</td>
</tr>
<tr>
<td>OLA</td>
<td>2.09</td>
</tr>
<tr>
<td>AOLA</td>
<td>2.12</td>
</tr>
<tr>
<td>OLA RAM</td>
<td>2.14</td>
</tr>
</tbody>
</table>

Table 1: Propagation delays of individual modules.

The minimum operating clock frequency of the system is derived from the worst-case scenario time. This is when the maximum or minimum desired scaling factor is required and the minimum natural scaling factor occurs and the speech type is voiced. The time taken for the system to perform under these circumstances is approximately 10,000 clock cycles. Based on an 8kHz input signal the minimum operating frequency is therefore 80MHz. From the table above and based on the shortest route from input to output, the maximum allowable operating frequency was found to be approximately 159.75MHz.

V. CONCLUSION

The AOLA TSM/FSM algorithm was successfully implemented into hardware using high-level VLSI techniques and VHDL. In addition, a voiced/unvoiced decision circuit and an unvoiced speech modification circuit were also successfully implemented. Upon testing the system with various synthetic speech signals of varying signal-to-noise ratios (SNR), the circuit performed as expected. However for poorer SNR signals, the decision circuit occasionally made incorrect decisions. This problem may be overcome easily with a suitable adjustment of the reference threshold values for noisy environments.

The total silicon area was found to be 7.5mm² (based on the 0.7 µm library). This area is suitably small enough for handheld equipment such as mobile telephones, dictaphones or other portable speech processing equipment. However, it should be possible through additional optimization techniques to reduce this area further.

REFERENCES