VLSI Implementation of the SIFT Algorithm for Pitch Detection

B. Brichta[1], M. Franke[2], A.Th. Schwarzbacher[3], J. Timoney[4] and B. Hoppe[1]

[1]*University of Applied Sciences Darmstadt, Germany*
E-mail: bjoern.brichta@web.de
E-mail: hoppe@fh-darmstadt.de

[2]*University of Applied Sciences Merseburg, Germany*
E-mail: marco.franke@et.fh-merseburg.de

[3]*Dublin Insitute of Technology, Ireland*
E-mail: andreas.schwarzbacher@dit.ie

[4]*National University of Ireland Maynooth, Ireland*
E-mail: jtimoney@cs.may.ie

Abstract -- **Speech voicing classification and pitch detection are fundamental techniques in speech analysis. Voicing information provides valuable insights into the nature of the excitation source used in speech production, and the pitch information is useful to many speech processing applications. In 1972 John Markel developed a technique which combines the benefits of inverse linear predictive (LPC) analysis and simple short-time autocorrelation to extract essential speech parameters. The research resulted in the simplified inverse filter tracking (SIFT) algorithm to make voiced/unvoiced classification of speech signals and determine the pitch period [1]. Up until now this algorithm was used in various software algorithms only. However, this paper describes a real-time CMOS hardware implementation of this algorithm small enough to be implemented into various mobile communications equipment.**

Keywords – **SIFT Algorithm, CMOS Design, VLSI Design, Pitch Detection**

I INTRODUCTION

The simplified inverse filter tracking (SIFT) [1] is an algorithm for classification of the voicing of speech segments and to estimate the pitch period of the speech labelled as voiced. Since both time- and frequency-domain approaches are used for the actual pitch detection; the SIFT algorithm is referred to as a hybrid pitch detector. This means that this algorithm influences the spectral properties using the inverse filter and extracts the pitch period information from the short-time autocorrelation. The general processing stages required for the extraction system are presented in Figure 1.

The algorithm commences by dividing the input speech signal into frames and then decimating each one. Each 32ms segment long frame of a digitised speech signal, $\{s_n\}$, is input to a tenth order FIR low pass filter. A FIR filter is used, since these structures are less complex to implement into hardware. The FIR filter bandlimits the frame to a bandwidth of 800Hz before it is downsampled to a sampling rate of 2kHz to give the signal denoted as $\{w_n\}$.

This stage is followed by another filtering procedure. This is intended to flatten the spectrum of the speech frame, thereby minimising the magnitude of the formant peaks. This will reduce their influence on the computed frame's autocorrelation function, assisting the peak detection function used in the determination of the pitch. The spectral flattening is achieved using an inverse filter derived from a Linear Prediction Coefficient (LPC) description of the spectrum of the speech frame. The inverse filter is a fourth order filter and its coefficients are calculated using the autocorrelation method of linear predictive analysis. This is computed by the autocorrelation equation

$$\sum_{i=1}^{\overline{M}} a_i p_{i-j} = -p_j, \qquad i = 1,2,...,\overline{M} \tag{1}$$

where $\{p_i\}$ is determined as the autocorrelation of the sequence $\{w_n\}$ from

$$p_j = \sum_{n=0}^{N-1-j} w_n w_{n+j}, \qquad j = 0,1,...,\overline{M} \tag{2}$$

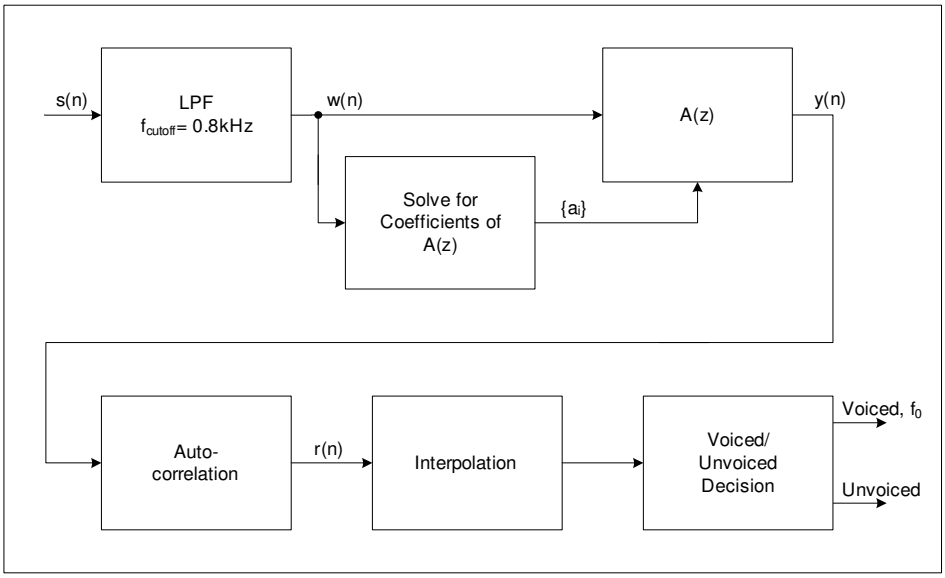
and the linear prediction coefficients are denoted by $\{a_i\}$.

Equation (2) is solved by use of the Levinson-Durbin recursion method [2] to obtain the coefficient values $\{a_i\}$.

The decimated speech signal $\{w_n\}$ is then inverse filtered by the FIR filter with transfer function

$$A(z) = 1 + \sum_{i=1}^{\bar{M}} a_i z^{-i} \tag{3}$$

The z-transform of the resulting output $Y(z)$, when applying the decimated speech frame $W(z)$ to $A(z)$, can be written as [2]



$$Y(z) = W(z)A(z) \tag{4}$$

In the sampled data-time domain this equation is equivalent to

$$y_n = w_n + \sum_{i=1}^{4} a_i w_{n-i} \tag{5}$$

where $\{y_n\}$ is the spectrally flattened prediction error signal.

To determine the voicing and then if voiced the pitch of the frame, the autocorrelation function $\{r_n\}$ is computed from $\{y_n\}$. If the frame is voiced, the location of the largest peak of the autocorrelation function indicates the pitch period of the frame. This short-time autocorrelation sequence is defined by the equation

$$r_n = \begin{cases} \sum_{j=0}^{N-1-n} y_j \cdot y_{j+n} & n = 0,1,\ldots,M/2-1 \\ 0 & n = M/2 \\ r_{M-n} & n = M/2+1, M/2+2,\ldots,M-1 \end{cases} \tag{6}$$

Since the fundamental frequency of human speech varies in the range of about 60Hz to 500Hz [3], the search of the autocorrelation sequence for the most significant peak can be reduced. Markel himself defined the frequency range where reliable results can be obtained with the algorithm from about 50Hz to 250Hz in a later publication [4]. However, the theoretically possible search bandwidth due to the 8kHz sampling rate of the system is approximately 58.82Hz to 500Hz.

To provide a better estimate of the pitch period, the autocorrelation function is interpolated in the vicinity of its the largest peak. Since any peak detected will define a real peak only within a range of ±1 sample, it is possible to apply the interpolation across the three samples, that is the detected peak itself and the surrounding samples on either side. A better estimate of the location of the true peak can be made using a simplified set of interpolation equations derived from a trigonometric interpolation formulation as shown in (7) that are applied to the scaled autocorrelation values

$$\begin{bmatrix} \gamma_{+3/4} & \gamma_{-3/4} \\ \gamma_{+1/2} & \gamma_{-1/2} \\ \gamma_{+1/4} & \gamma_{-1/4} \end{bmatrix} = A \begin{bmatrix} \gamma_{+1} & \gamma_{-1} \\ \gamma_0 & \gamma_0 \\ \gamma_{-1} & \gamma_{+1} \end{bmatrix} \tag{7}$$

where

$$A = \begin{bmatrix} 0.879124 & 0.321662 & -0.150534 \\ 0.637643 & 0.636110 & -0.212208 \\ 0.322745 & 0.878039 & -0.158147 \end{bmatrix} \tag{8}$$

and the scaled values are given by

$$\gamma_a = r_{\hat{n}+a} \Big/ r_0 \tag{9}$$

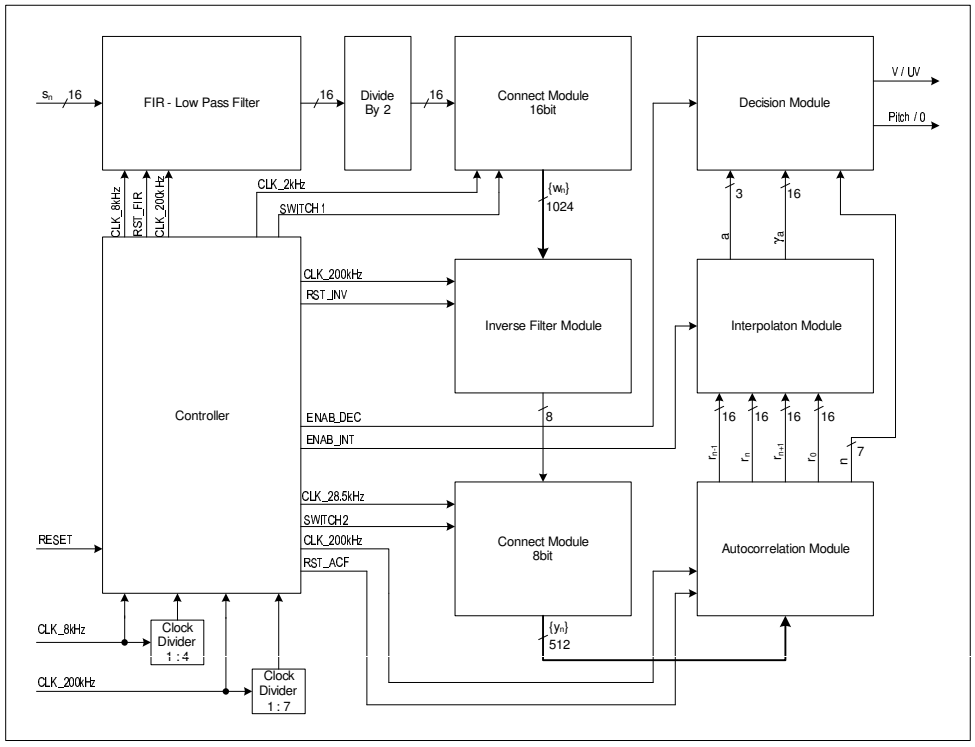
Based upon the magnitude of this interpolated peak, the decision block determines whether a speech signal is voiced or unvoiced. When a segment is defined as being voiced the reciprocal of the peak

location defines the fundamental frequency, F_0, in kHz.

II IMPLEMENTATION

The system was implemented using the VHDL and synthesised with the Synopsys Design Compiler [5] without any design constrains. The implementation technology used for synthesis is the Europractice ES2 ECPD 0.7µm CMOS technology [6].

The overall structure the system was split into five main functional components to facilitate the coding, testing and verification of its individual parts. The VHDL code had to be implemented to allow the synthesis of the code in an energy efficient and area saving way [7]. Therefore, signed magnitude number representation and add and shift algorithms for multiplication instead of two's complement and parallel multipliers were implemented. To store the output frames of the low pass filter and the inverse filter, two additional pipeline blocks had to be realised and integrated in the top level design as shown in Figure 2.



The input speech signal of the overall system is an 8bit quantised speech signal, sampled at a frequency of 8kHz to match the requirements of modern telecommunication standards. The system has four input ports: one for the speech input, another for a clock of operating frequency of 200kHz, another for a clock of at sampling frequency of 8kHz, and lastly one for a reset signal,. Since the voiced/unvoiced classification and the pitch period are to be extracted from the incoming speech signal, two output ports were defined. The decision whether a frame is voiced/unvoiced is displayed by a single bit. Concurrently the pitch information for voiced classified frames or zeros for unvoiced frames is shown at the second output port.

When the SIFT system is started and the central RESET occurs, the internal counter is started, enabling the modules to work. The first module is the FIR low pass filter which works continuously on the 8bit input signal and reduces the frequency content of the speech signal to the desired bandwidth. The speech frame length was chosen to be 32ms or 256 samples, and thus at the decimation stage following the FIR filter only every fourth is switched through to the following connect module. The shift-register in the connect module has a working frequency of 2kHz and the remaining modules have a time slot of 32ms to compute the SIFT result.

The next section contained the LPC-based inverse filter computation stage. It was found that for some input frames with high amplitude, the values of the LPC coefficients exceeded the defined bitwidth of 16bit for the inverse filter module. Further investigation lead to the conclusion that for the later stages of processing the amplitude height did not matter, once the relative proportions of the amplitude values in the frame are retained. Thus, amplitude

scaling by a factor of 0.5 was performed to alleviate this problem.

When the 64 samples of the decimated FIR output are computed, frame $\{w_n\}$ is handed to the inverse filter at the positive clock edge of the SWITCH1 signal. Then the inverse filtering is activated on the current frame by resetting the module and enabling a 200kHz clock. To suspend the operation of the inverse filter the clock is switched off when it is not needed for processing. The calculation of each sample of the inverse filter output requires seven clock cycles. Hence, the following shift-structure of the second connect module has to store a new sample every 35µs, giving an operating frequency of 28.57kHz. The second switch signal, SWITCH2, enables the frame to be applied to the short-time autocorrelation. As for the inverse filter, the autocorrelation block is set off by a reset impulse and the enabling of the 200kHz clock. The calculation of the values r_0 to r_3 requires 11.22ms. Each of the following two modules is then triggered by a separate enable signal by the next two clock cycles after the autocorrelation. Once the prefiltering sections have computed a complete frame of 32ms duration, the estimation of the pitch information takes approximately 15.2ms in total.

III RESULTS

After the top-level design of the SIFT algorithm was implemented, simulations using real speech signals were applied to evaluate the system performance of the whole structure. The test signals used for this evaluation were speech signals from a male and a female speaker as described in Table 1. These test signals were distorted with White Gaussian Noise with different Signal to Noise Ratios. For the benchmarking, the reference signal against which the output values of the hardware detector are compared was the corresponding output found by inputting the clean signal to a reference software model. Thus, it could be ensured that the correct pitch period as determined by the algorithm was found and that the voiced and unvoiced frames were labelled correctly.

Speech Sample	***Utterance***	***Voiced Content / %***
Female	"Elderly people are often excluded"	61.0
Male	"Are you looking for employment?"	73.2

Table 1: Real Speech Files Used for the SIFT Performance Tests

Figure 3 and **Figure 4** show an example from the simulation results for both speech signals where the input has a SNR of 25dB. Each figure shows the clean speech signal as analysed by the detector in the time domain. Below this graph, the reference pitch period contour returned from the software model is presented. The third graph shows an example of the pitch detection result performed by the SIFT hardware pitch detector for a SNR level of 25dB. It can be seen that for the male speech signal the pitch is detected accurately and the extracted fundamental frequencies correspond to those determined by the reference model. Only a single frame, that is frame 24, is erroneously defined unvoiced. It is probable that this difference is due to the computational method for finding the short-time autocorrelation peaks by the hardware model when compared to the software model.

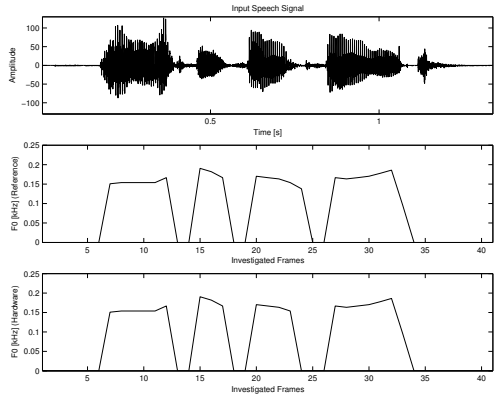


Figure 3: Results of Pitch Detection Benchmarking, File: Male, SNR =25dB

For the female speaker the pitch is sometimes detected to be twice as high or twice as low as expected. This phenomenon is known as frequency doubling/ halving, and an example of halving appears in Figure 4 for several frames in the vicinity of frame 20. The reference model shows less occurrence of this phenomenon but in general has the same problems as the hardware implementation. Although this problem should be avoided because of the spectral flattening action of the inverse filter this error often occurs when the frame has a low-power fundamental harmonic combined with a strong second or subharmonic which is detected in place of the fundamental.

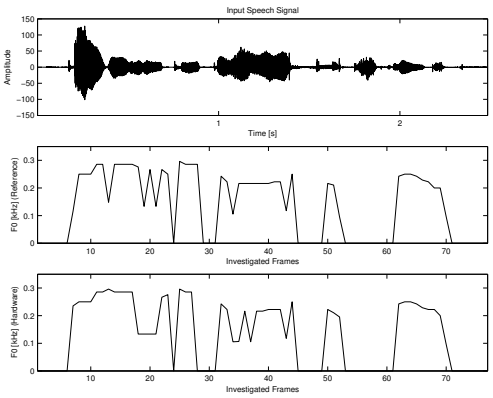


Figure 4: Results of Pitch Detection Benchmarking, File: Female, SNR =25dB

Since the male speech which varies below 200Hz has no deviations of that kind, it can be assumed that the pitch doubling problem for the SIFT algorithm is mostly caused by the higher fundamental frequency for the female utterances. This would explain the restricted frequency range for the SIFT algorithm from 50Hz to 250Hz as proposed by Markel in a later publication [4].

For the current hardware implementation it was found from the results of all the experiments that the pitch is detected well for low pitch speakers down to a distortion level of 5dB SNR. In case of the female utterances the gross error of pitch doubling/halving occurred several times. To overcome this problem it would be necessary to identify such a situation after the short-time autocorrelation and the force the peak finder to impose some continuity constraint on the peak selection process rather than just picking the largest each time. With such a correction mechanism embedded in the SIFT structure it may also be possible to increase the upper limit for reliable pitch estimation. The second problem that occurred during the simulation was the mistaken definition of voiced frames as unvoiced. A possible solution to overcome this problem would be to adjust the threshold settings for the hardware model. To obtain an improved threshold would require more extensive tests to provide the necessary statistical data for its determination.

IV CONCLUSION

The objective of this paper was to describe the implementation the SIFT pitch detecting algorithm in hardware. The work focussed on a high-level VLSI design using the hardware description language VHDL.

The hardware structure was tested using a variety of speech signals from a male and a female speaker with different levels of white noise distortion. Performance tests showed that the implemented system operates in real-time while fulfilling the algorithmic function of the SIFT correctly. It was found through the use of different levels of interfering noise that the detector works robustly for low pitch speakers down to a SNR of 5dB. For high pitch speakers, such as females, it was found that the algorithm is prone for pitch doubling.

In conclusion, this paper has presented the real time CMOS implementation of a pitch detector based on the SIFT algorithm. The developed structure has been tested and it has been shown that the system is suitable to classify the voicing of the speech segments into voiced or unvoiced, and to estimate the pitch period under noisy conditions.

REFERENCES

[1] J. D. Markel, *The SIFT Algorithm for Fundamental Frequency Estimation,* IEEE Transactions on Audio and Electroacoustics, Vol. AU-20, No. 5, December 1972

[2] L.R. Rabiner; R. W. Schafer; *Digital Processing of Speech Signals*, Prentice Hall, 1978

[3] Gunnar Fant, *Acoustic Theory of Speech Production*, Second Edition, Mouton, The Hague, Paris, 1970

[4] J.D. Markel, A.H. Grey, Jr., *Linear Prediction of Speech*, Springer Verlag 1976

[5] Synopsys, www.Synopsys.com (April 2005)

[6] European Silicon Structures, *ES2 ECPD07 Library Databook*, July 1995

[7] F. Gittel, *VLSI implementation and investigation of an adaptive noise canceller*, Final Year Report, Deutsche Telekom University of Applied Sciences & Dublin Institute of Technology, 2003