INTEGRATED CIRCUITS FOR AUDITORY SPEECH PROCESSING

W. Nebel, M. Brucke, M. Cakir

Carl von Ossietzky University of Oldenburg
Department of Computing Science
[nebel,brucke,cakir]@uni-oldenburg.de

ABSTRACT

Deriving an implementable application specific integrated circuit (ASIC) from the functional specification of an auditory speech processing algorithm requires several refinement steps. These include design decisions on the data representation, the detailed scheduling of the operations and the resources allocated. We give an overview of these steps and illustrate them with an example, the ASIC implementing an auditory perception model. We demonstrate the optimization potential of custom integration and the need for objective quality measures to assess the impact of design decisions. The paper concludes with a summary of the results for the example circuit.

1. INTRODUCTION

This paper presents in a tutorial style an introduction to the design of application specific integrated circuits (ASICs) for auditory speech processing. Observing the continuous increase in processor and DSP performance one might ask why should we bother to think about application specific circuits rather than utilizing the available processing power of programmable processors. These processors are available off the shelf together with comfortable development systems for SW design. In most cases they have enough computing power to solve even complex signal processing problems. For example a Texas Instruments TMS 320C62xx clocked with 250 MHz reaches a peak performance of 2000 MIPS for the cost of about 100$. Further they can be easily reprogrammed to meet changed requirements or fix bugs in the software. However, looking closer at the efficiency of the computations performed on a processor compared to a dedicated circuit one immediately realizes that the programmable processor needs to execute a number of sequential operations to perform a single data manipulation, e.g. the addition of a digital filter:

1. Fetch the instruction from the program memory or I-cache.
2. Decode and interpret which instruction it is and what further actions to take.
3. Calculate the addresses of the operands.
4. Make them available to the execution unit and execute the arithmetical, logical or control flow instruction.
5. Write back the results.

Many of these steps are an unnecessary overhead especially for signal processing algorithms. The sequence of instructions is exactly known in advance from the algorithm, no data dependent conditional branches in the instruction sequence are to be expected and even the source and destination of the data is known far ahead. Hence a classical relatively small finite state machine can organize the execution control of the data manipulations avoiding phases 1-3 and 5. The gain in throughput is not only due to eliminating four out of five phases, but also by avoiding slow access to the program memory, which is typically one order of magnitude slower than the processor itself. In consequence to implement an algorithm in SW on a DSP requires additional resources to perform these steps. These resources consume area and power. Hence in applications where area or power are constraint by the physical environment, e.g. hearing aids, implants, mobile communication devices, an application specific integrated circuit provides factors in integration density and power reduction. Further in case of cost limited systems the semiconductor area reduction of an ASIC, which basically determines its fabrication cost, may enable new products to penetrate the market. In Section 2 of this paper we will first describe a generic ASIC design from the specification of the algorithm to the implementable structural description of a semiconductor device. The design flow will be illustrated by a practical example, the silicon implementation of the Oldenburg Perception Model in Section 3. A Framework for the estimation of internal number representations will be shown in Section 4. We will conclude with the results of this example in Section 5.

2. THE ASIC DESIGN FLOW

The task of the IC-design is to bridge the gap between a functional specification of the requirements and the fabrication instructions for the final physical device. The specification of signal processing algorithms typically comes in form of MatLab or C-programs, or even mathematical descriptions, of the functions to be implemented. This input more or less describes the data dependencies of the operands and operations of the algorithms. Although the programs seem to suggest a sequential order of the operations - assuming an execution on a sequential processor - this is an overspecification, because in many cases the execution can be performed in another order without changing the results. Further the sequential program seems to prescribe that exactly one execution step is performed at a time, which is again an over-specification because typically several of the operations could also be performed in parallel or concurrently without any effect on the result. On the other hand looking at the program code, it seems to be obvious, that each operation is performed in one step. Again this is not true, because we might, e.g. perform a multiplication in a bit sequential fashion as long as we meet the throughput constraints. Finally the operands and operations are typically coded as floating point data in the algorith-

mic specification. This is an idealization of the cost optimized hardware implementation, which will map data and instructions to the smallest possible bit-width. As you can see, ASIC design is the task to:

- Define the data representation of numbers and operations as fixed point bit vector data.
- Allocate the minimum set of hardware resources to implement the required functionality while meeting the timing and throughput constraints.
- Design a detailed schedule of the operations on the given resources.
- Design a communication structure, which ensures that the data can be passed to the operators.
- Design a controller, which ensures that the right data are available at the correct operator in the required time instance.
- Design the physical layout for the fabrication.

These subtasks are highly interdependent and can be classified as refinement steps for data, timing and structure. It is the responsibility of the ASIC designer to perform these tasks to optimize the circuit with respect to cost and power without neglecting functional, performance and quality requirements. In particular in case of auditory signal processing, methods and tools are needed to allow for an early assessment of the implication of design decisions on the quality of the final design. Since the entire design typically takes several months and the semiconductor fabrications requires further weeks, it is very important to detect any kind of design problems and errors at the earliest possible instant. This requires sophisticated EDA (electronic design automation) tools for the verification and assessment of design decisions. Further, due to the enormous possible design space tool support is needed to efficiently explore this space for feasible solutions and to help the designer in optimizing these in an automatic way. In the following part of this paper an ongoing project is used to show the necessary steps when implementing an auditory speech processing algorithm in hardware.

3. THE PERCEPTION MODEL

The Medical Physics Group at the University of Oldenburg has been working on the field of psychoacoustical modeling, speech perception and processing, audiological diagnostics and digital hearing aids for several years (see [7] for a review). One part of the work is the development of a psychoacoustical preprocessing model and the demonstration of its applicability as a preprocessing algorithm for speech in, e.g. automatic speech recognition, objective speech quality measurement, noise cancellation and digital hearing aids. In interaction with the "Graduate School in Psychoacoustics" the processing model has been improved and optimized. The model describes the "effective" signal processing in the human auditory system and provides the appropriate internal representation of acoustic signals. The VLSI group at the University of Oldenburg and the IMA group at the University of Hamburg are now working on a transformation of this signal processing algorithm into a set of integrated digital circuits [1],[9].

3.1. The Preprocessing algorithm

Figure 1 outlines the structure of the psychoacoustical perception model. The model combines several stages of processing simulating spectral properties of the human ear (spectral masking, frequency-dependent bandwidth of auditory filters) as well as dynamical effects (nonlinear compression of stationary or dynamic signals, temporal masking). The appropriateness of this approach was shown in several psychoacoustical experiments by Dau [3]. The gammatone filter bank (GFB) represents the first processing stage and simulates the frequency-place transformation on the basilar membrane. It is built up from 30 bandpass filters with center frequencies from 73 Hz to 6.7 kHz equidistant on the ERB scale. The bandwidth of the filters grows with increasing frequency. The output of each channel of the gammatone filter bank is halfwave rectified and lowpass filtered at 1 kHz to preserve the envelope of the signal for high carrier frequencies. This is motivated by the limited phase locking of auditory nerve fibers at higher frequencies.

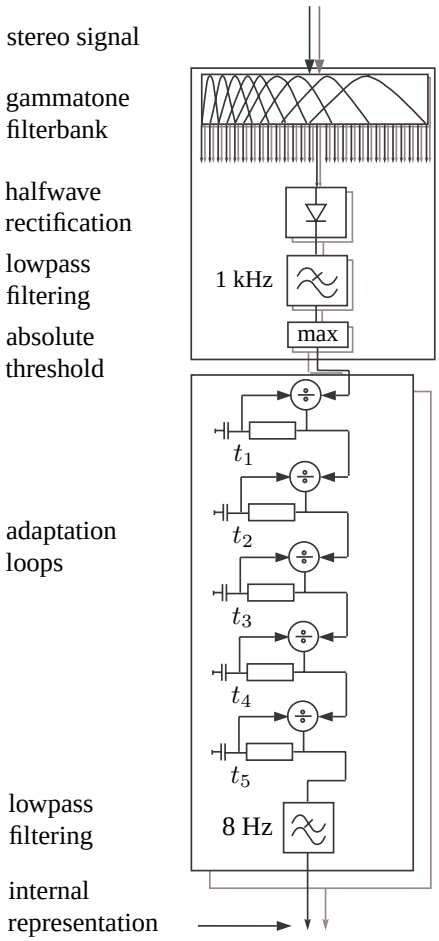

Figure 1: Signal processing scheme of the binaural perception model. Transformation of the input sound signal into its internal representation according to [3]

The adaption loops model the dynamic compression of the input signal. Stationary signals are compressed almost logarithmically whereas fast fluctuating signals are transformed linearly [6]. The adaptation stage is a chain of five consecutive feedback loops with different time constants range from 5 to 500 ms. Each

feedback loop consists of a divider and a lowpass filter. The divisor determines the "charging state" of the capacitor low pass. Thus the system has some kind of "memory" determining the compression of the current signal level based on the signal history which accounts for the ability to correctly model temporal masking effects.
In the next stage the signal is filtered by an 8 Hz first order modulation lowpass. This accounts for decreasing modulation detection at higher frequencies found for many broadband carrier signals.

3.2. Implementation of the Model

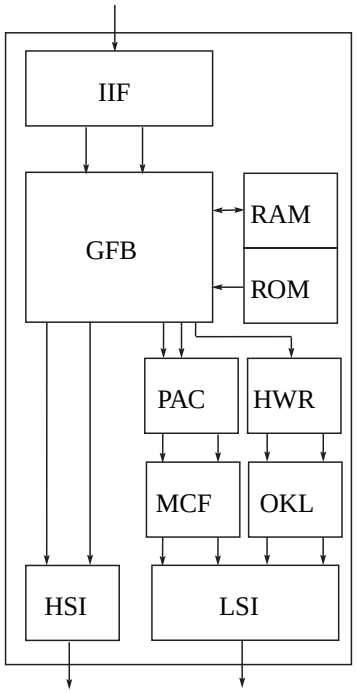

Figure 2: Internal structure of ASIC 1

The design was partitioned to split up the necessary computations on two ASICs. The first ASIC (ASIC 1) being developed in Oldenburg contains the gammatone filter bank, the halfwave rectification and the 1 kHz lowpass which work in stereo (i.e., two independent channels used for binaural signal processing). Furthermore a module is incorporated which calculates the phase difference and the amplitude quotient between the two stereo channels for each auditory filter [2]. The second ASIC (ASIC 2) being developed in Hamburg processes the output data of ASIC 1 and applies the dynamic compression of the adaptation loops and the modulation lowpass filtering on the signal. [8]. Due to its complexity it works monaurally and has to be used twice in the target environment.
The internal structure of ASIC 1 is shown in Fig. 2. The serial input interface (IIF) reads the data from an ADC or DSP. The filter bank (GFB) calculates the complex valued output for all 30 channels. The ROM serves as a memory for the filter constants for all channels, the RAM for saving temporary values. ASIC 1 has two output interfaces: the highspeed serial interface (HSI) transmits the pure output data of the GFB with a rate of about 30 MBit/s. This data can be used for a resynthesis of the input signal or a digital hearing aid algorithm. The real parts of the output data of the GFB are halfwave rectified (HWR) and filtered by an 1 kHz lowpass (OKL). Simultaneously the amplitude quotient and the phase difference are calculated for each channel (PAC) and averaged by a lowpass-and-decimation filter (MCF). The data of the two paths are joined with the real parts of the filter outputs of each channel and are available at the lowspeed serial interface (LSI).

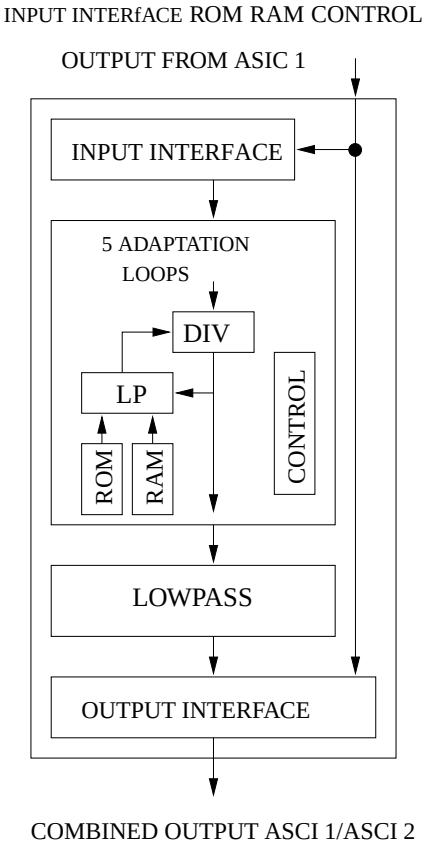

Figure 3: Internal structure of ASIC 2

Figure 3 shows the internal structure of ASIC 2. The INPUT INTERFACE converts the serial bitstream and passes the bit parallel data to the first adaptation loop. Each of the five adaptation loops contains a divider whose quotient is fed back by a 1st order IIR lowpass providing the divisor. This feedback and the necessary signal dynamic forces large fix point wordlengths or a logarithmic number format. Therefore the dividers are the most area expensive components. According to its time constant the lowpass follows with a certain time lag to a level change of the input signal. After the lowpass has been "charged" to values > 0 fast changes of the input signal are transmitted almost linearly to the output. However, the output of the fifth feedback loop is approximately the logarithm of the input signal for stationary signals: $y = x^{(1/2^5)} = x^{(1/32)} \cong \log(x)$. The output data stream of ASIC 2 contains the "internal representations" for each input sample and the phase difference and amplitude quotient between left and right stereo channel for each of the 30 filters of the gammatone filter bank.

4. TRANSFORMATION FROM FLOATING POINT TO FIXED POINT

A direct implementation of an IEEE 32 bit floating point arithmetic of the model is not possible due to limitations of area and power consumption. To gain an optimal implementation different methods are applied to the linear filter bank and the nonlinear adaptation loops respectively. The main problem when converting floating point arithmetic to fixed point arithmetic is the determination of the necessary numerical precision. Therefore the perception model was recoded in C++ using a self-developed scalable fixpoint data type. This data type takes the internal wordlength as a parameter and saves the values exactly in the same format as they would be saved in a register on an ASIC. So numerical effects of imprecise arithmetic can be simulated.
The necessary internal wordlength for the gammatone filter bank can be assessed in a straight-forward way, because the filters are linear time invariant systems where classical numerical parameters like SNR can be applied. It is sufficient to record the responses on a δ-pulse for each filter parameterized with different internal wordlengths. Figure 4 shows the mean square error between one of these implementations and the original specification with floating point arithmetic. The choice of a certain maximal square error (e.g. 10^{-3} for all channels) leads directly to the necessary internal wordlength.

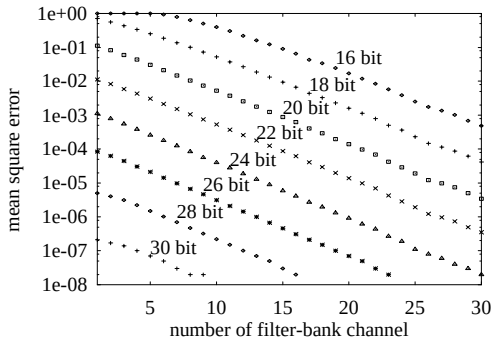

Figure 4: Mean square error between implementations with different wordlengths and the floating point implementation for each channel of the gammatone filter bank. Note, that due to increased analysis bandwidth the error for a given wordlength decreases with increasing frequency / channel number.

For the realization of the nonlinear parts of the system this procedure is not applicable. It is a significant problem, that e.g. the divisor in the adaptation loops results from the quotient (feedback). Even the floating point implementation of the model needs a threshold for the divisor because otherwise strong overshoot effects may occur when dividing by very small numbers. The need to preserve a certain precision at the output of the last feedback stage would cause enormous wordlengths in the prior stages. This contrasts with the general demand for a minimum area at a given clock rate. To solve the problem of large wordlengths, the dynamic range of the signal must be limited by lower and upper bounds. This implies a change of the dynamic behavior between fixed point and floating point versions. The effects of this change to the behavior of the whole model had to be investigated. The only method to determine the optimal wordlengths and to validate the correct function of the model is to simulate various implementations with different wordlengths in a given target application and to observe the influence of the wordlength on the performance of the application.
One possible application of the model is the use as a preprocessing stage for the measurement of objective speech quality [4]. In this application, distorted speech signals are generated by low-bit-rate speech coding-decoding devices ("codecs") such as used in mobile telephony. These codecs produce a speech signal that is fully intelligible and allows almost normal speaker identification, compared to standard telephony. However, they exhibit a clearly reduced speech quality due to their highly nonlinear and/or time-variant algorithms. In listening experiments carried out by the research center of Deutsche Telekom, the speech quality has been subjectively rated by test subjects [4].

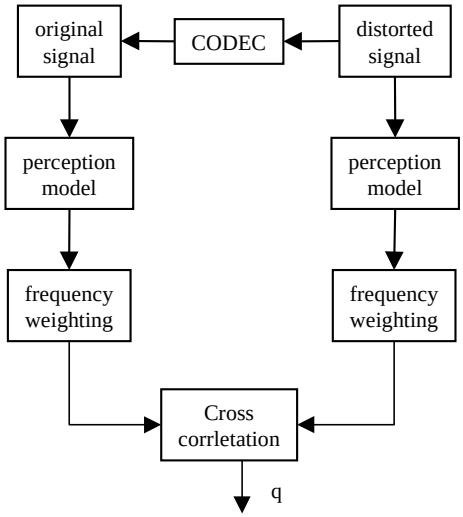

Figure 5: Structure of the speech quality measurement [5]

In objective speech quality measurement, the application of psychoacoustical preprocessing models is motivated by the assumption, that subjects are able to judge the quality of a test speech signal by comparing the "internal perceptual representation" of the test sound with that of a reference sound [4]. This representation is thought of as the information that is accessible to higher neural stages of perception. It should contain the perceptually relevant features of the incoming sound. Differences in this "internal representation" of input and output signal are expected to correspond to perceivable differences of the two signals and thus to indicate a decreased speech quality of the output signal. In subjective listening tests, typically sentences of different speakers are encoded using the same codec-condition and are rated individually by the subjects. For the objective method, these sentences of different speakers were concatenated and their average mean opinion score (MOS) were calculated in order to reduce the variability of the MOS due to the different voices of the speakers. The original and the distorted signal are then aligned with respect to overall delay and overall RMS. Both signals are then transformed to their internal representations. The objective-subjective speech quality data can be fitted by a monotonic function with only small deviation *sd*. A high correlation coefficient *r* is achieved and, in particular, no clusters of different codec types occur. This indicates that the individual signal degrada-

tions introduced by the different types of codecs are transformed in a perceptually "correct" way into the internal representations. Therefore, this method can be used to analyze effects of internal wordlength by observing the degradation of the correlation when decreasing internal arithmetical precision. Figure 5 shows the used framework.

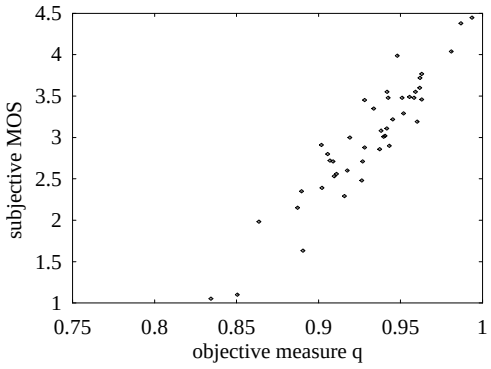

Figure 6: Results of the objective speech quality measurement with the auditory preprocessing model: Subjective quality (MOS) versus objective measure q for the ETSI Halfrate Selection Test: Floating point version of the model.

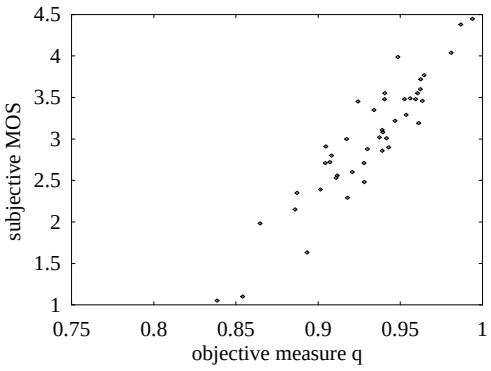

Figure 7: Results of the objective speech quality measurement with the auditory preprocessing model: Subjective quality (MOS) versus objective measure q for the ETSI Halfrate Selection Test: Fixed Point version of the model with "optimal" wordlength

Figure 6 and 7 show the results of the ETSI tests for a floatingpoint implementation and for a fixpoint implementation with "optimal" internal precission. Based on this information it is easy to decide whether the performance of the application is sufficient for a given wordlength. The results from Figure 6 and 7 are determined at middle signal levels. The derived wordlengths were validated by simulations with input data at different signal levels. This indicates that the deviations from the original floating point algorithm can be assessed quite well by observing the performance of the fixed point implementation in a typical application of the processing scheme.

5. CONCLUSION

In this paper we have presented the necessary steps for deriving an implementable application specific integrated circuit (ASIC) from the functional specification of an auditory speech processing algorithm. We have showed the ASIC design flow and the problems designers are faced when having to decide between different implementations. We demonstrated the optimization potential of custom integration and the need for objective quality measures to assess the impact of design decisions. An ongoing project – the hardware implementation of a quantitative model of the human auditory system – was used to show the framework for the estimation of internal wordlengths when converting a high level floating point description of the algorithm into a fixpoint implementation suitable for a hardware implementation.

6. REFERENCES

[1] M. Brucke, W. Nebel, M. Hansen, B. Kollmeier, A. Schwarz, and B Mertsching. Digital vlsi-implementation of a psychoacoustically and physiologically motivated speech preprocessor. In *Proceedings of the NATO Advanced Study Institute on Computational Hearing*, Il Ciocco (Tuscany), 1998.

[2] M Brucke, A. Schulz, and W. Nebel. Auditory signal processing in hardware: A linear gammatone filterbank design for a model of the auditory system. In *Field-Programmable Logic and Applications: Proceedings of the 9th International Workshop, FPL '99*, number 1673 in Lecture Notes in Computer Science, pages 11–20. Springer-Verlag, 1999.

[3] T. Dau, D. Püschel, and A. Kohlrausch. A quantitative model of the 'effective' signal processing in the auditory system: I. Model structure. *J. Acoust. Soc. Am.*, 99:3615–3622, 1996.

[4] M. Hansen and B. Kollmeier. Implementation of a psychoacoustical preprocessing model for sound quality measurement. In *Tutorial and Workshop on the Auditory Basis of Speech Perception*, pages 79–82, Keele, 1996. ESCA.

[5] M. Hansen and B. Kollmeier. Using a quantitative psychoacoustical signal representation for objective speech quality measurement. In *Proc. ICASSP-97*, page 1387, 1997.

[6] A. Kohlrausch, D. Püschel, and H. Alphei. *Temporal resolution and modulation analysis in models of the auditory system*, pages 85–98. Mouton de Gruyter, 1992.

[7] B. Kollmeier, T. Dau, M. Hansen, I. Holube, and M. Wesselkamp. An auditory model framework for psychoacoustics and speech perception - and its applications. In *Proc. Forum Acusticum*, volume 82, Suppl. 1, page 89, Antwerpen, 1996.

[8] A. Schwarz, B. Mertsching, M. Brucke, W. Nebel, J. Tschorz, and B. Kollmeier. Digital vlsi-implementation of a psychoacoustically and physiologically motivated speech preprocessor. In T. Dau, V Hohmann, and B. Kollmeier, editors, *Psychophysics, Physiology and Models of Hearing*, pages 11–20, Singapore, 1998. World Scientific. ISBN 981-02-3741-3.

[9] A. Schwarz, B. Mertsching, M. Brucke, W. Nebel, J. Tschorz, and B. Kollmeier. Implementing a quantitative model for the 'effective' signalprocessing in the auditory system on a detcated digital vlsi hardware. In *Proceedings of the 25th EUROMICRO Conference*, Milan, Italy, 1999.