LOW POWER SYSTEM ON CHIP IMPLEMENTATION SCHEME OF DIGITAL FILTERING CORES

E. P. Zwyssig, A.T. Erdogan, T. Arslan

Department of Electronics and Electrical Engineering, University of Edinburgh,
Kings Buildings, Mayfield Road, Edinburgh, EH9 3JL, Scotland U.K.
Phone: ++44 131 650 1000; Fax: ++44 131 650 5155
erich.zwyssig@ieee.org, {arslan, ate}@ee.ed.ac.uk

1. Abstract

The paper describes a scheme for the implementation of low power cores for hearing aid applications. Power saving features of the scheme are two fold. The first due to the utilisation of a macro-component framework which allows the rapid assembly of the cores on easy-to-verify hierarchical plug-in basis. The second is due to the system-on-chip strategy. The cores are embedded within an ARM-based system-on-chip platform. For this VSIA compliant wrapper interfaces have been investigated which allow the cores operation in multiple clock and bus domains. This provides effective power manipulation by power management strategies. The paper demonstrates the scheme with a number of FIR filtering cores. The cores are assembled with a number of different macro-components from the wide range available within a rich library which contains a flexible and synthesisable set of components for computation, memory, decision, and BIST functions.

Keywords: *Low Power, IP framework, Digital Filter, Design for Testability (DfT)*

2. Motivation

Digital hearing aids today perform a variety of advanced digital signal processing algorithms, such as noise reduction and echo cancelling. For this reason hearing aids have challenging power and area requirements compared to other portable devices.

Work reported in the literature on the development of digital signal processing for hearing aid applications is either based on a full customised design of a DSP processor [1] [2] or makes use of a generic DSP core [3]. Products available on the market today are all based on customised DSP cores since the power dissipation in these devices could be managed more efficiently. Example work in this area is that presented by Neuteboom et. all. [1] and Moller et. all. [2]. These consider performance issues regarding the multiply-accumulate unit (MAC) alone without analysing the overall system power dissipation.

Our investigations reveal that it is important to consider the impact of the complete DSP architecture on the overall performance of the hearing aid application system. As an example consider Eq. 1 which describes a FIR Filter.

$$y_n = \sum_{m=0}^{M-1} b_m \cdot x_{n-m} \qquad \text{Eq. 1}$$

From the equation a number of key components/operations can be identified. These are as follows:

- a multiply-accumulator
- several forms of memory such as those required for the sample values x_{n-m} (RAM) and the coefficient values b_m (ROM)
- a storage cell for the filter output value y_n
- a control function which schedules the different components.

Figure 2-1 illustrates the principle data flow between the above components in a block diagram.

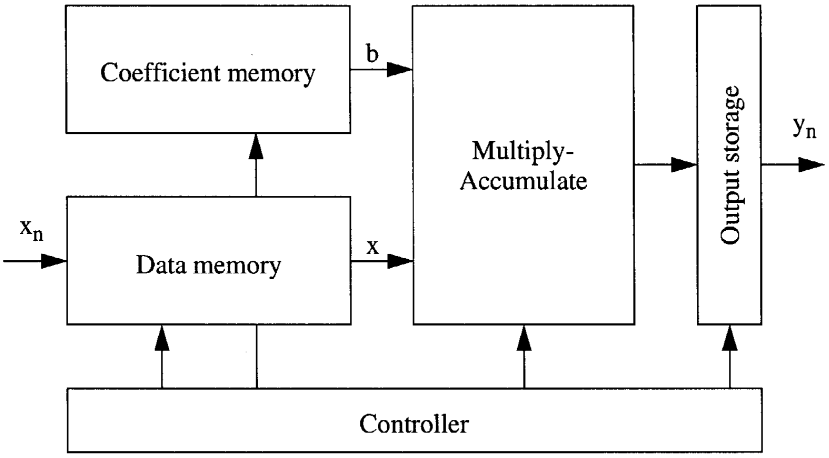

Figure 2-1 System Architecture of FIR Filter

When analysing other DSP functions (e.g. IIR, Lattice) or Fast Fourier Transform (FFT) modules, similar components can be identified.

Due to the demand for fast time-to-market, there is a continuous need for fast prototyping of algorithms for DSP systems with complex performance requirements. In this work, we have developed a specially tailored macro component library which can be manipulated within a unified framework for the fast development of DSP systems. Macro components are fundamental units which are heavily used within certain application domains, in this case dedicated hearing aid applications. These components are designed with a high degree of flexibility and parametrisability, hence making them highly reusable. An important aspect of the components is that they are technology independent and can be easily ported for use with different foundary libraries. We have developed a complete framework which allows rapid design and verification of complete DSP systems as an Intellectual Property (IP) with embedded DfT features.

The paper describes the library and emphasises the role of the framework in the development of DSP systems which are characterised by the following:

- low power dissipation
- low area consumption
- flexibility through use of HDL (including synthesisability)
- parametrisability
- Design for Testability (DfT)

In addition the paper also describes a strategy which allows embedding the developed cores in a system-on-chip architecture allowing a trade off between power consumption and area on one hand and reusability on the other.

3. Macro component library

Currently, the macro-components in the library can be categorised as follows:

- Clock (e.g. clock gate)
- Counter
- Multiplexer
- Memory, Storage
- Arithmetic
- Decision
- Conversion

- DfT

With regard to the first category, a number of clock gating units have been developed. These range from those using primitive gates such as AND only to those which incorporate an edge triggered register with the aim of reducing glitch activity. Clock gating forms the backbone of low power design strategy in hearing aid devices.

A design strategy which has been employed with the sequential macro components in the library is the use of a global enable signal in conjunction with the clock. This results in a toggling activity only when the sequential component is enabled (activated). The approach presented above has for example been used in the design of a number of parametrisable up/down counters with features such as gated clock, enable and scan path.

Strong emphasis has been put on the design of multiplexer and demultiplexer structures with the focus on small area usage, low power dissipation and Design for Testability (DfT). The macro component library contains three different multiplexer types: the discrete multiplexer [4], our low power multiplexer and a sea-of-gates type multiplexer.

Our low power multiplexer utilises a special address decoder generating individual enable signals for every multiplexer leaf, with the aim of reducing the toggle activity. A sea-of-gates multiplexer is implemented using the Verilog *casex* construct for address decoding and is processed by automatic synthesis tools.

The design of multiplexer and demultiplexer is especially important because they are also the building blocks of memories.

The demultiplexer and the multiplexer are major building blocks of the memory cells and it is helpful to have a set of different sizes available for building different sizes of memory. We therefore developed a multitude of different generator cells using a behavioural Verilog description which, after setting several parameters, generates an RTL Verilog description of blocks of these blocks.

A major part of the macro component library is dedicated to arithmetic components. This includes basic functions like increment/decrement, add/subtract and multiply as well as more complex multiply-accumulate units. Various types of computation techniques have been employed including carry-ripple, carry-sum and carry-look-ahead [5]. The multiply-accumulate units have the following features:

- different data representations such as twos complement, sign-magnitude and mixed. For this reason a number of data type converters have also been incorporated into the library.
- multipliers with pre-addition units in order to cope with folded structures [2]
- multiple accumulators in order to cope with transpose structures
- multiplier encoding techniques which include unsigned and Booth [6]
- multiplier partial product reduction trees such as carry save, redundant binary arithmetic and mixed [6]

Hearing aid applications also require functions such as Min(a,b) or Equal(a,b) which have been assembled in a category named *decision* and other functions like y=Round(a) or y=Shift(a,n) which have been put into a category named *conversion*. These macro components have been redesigned for low power dissipation with pre-computation [8] circuits. Novel low power features have been incorporated into the above cells using simple data gating instead of pre-computation, resulting in additional power savings. Table 1 shows power and area results with a selected number of decision circuitry. Each block of three entries in the table, refers to the core decision function, a registered version of the function, and a low power version implemented using pre-computation.

Table 1: Analysis results of decision cells

Cell	Area	Power
	[u]	[uW]
Greater	96	210
greater_reg	336	653
greater_lp	361	423
Min	118	290
min_reg	360	734
min_lp	393	653
Max	123	297
max_reg	370	761
max_lp	394	642

Pattern generators (e.g. RAM BIST [9]) and data compressor (e.g. BIST signature) cells have been implemented so that power is dissipated in test mode only. These components have been included in the DfT category of the macro component library.

A strategy followed during the design of the macro components framework was to employ parameterisability whenever possible in order to avoid an extensive set of cells with different size and width. Complex cells with high repetition of code, such as RAM, ROM or carry-save adder, are automatically designed using a generator cell from which cells with variable width and depth can be generated.

4. IP framework

Currently the development of different systems is based around UNIX's shell scripting language which invokes synthesis or simulations in a hierarchical design oriented directory framework. Synthesis and simulation scripts have a simple and uniform format. Assembling a new/modified DSP system will simply require instantiation of individual macro component blocks and developing the appropriate control circuit.

Each macro component is processed through a number of verification stages which interrogate aspects related to issues such as functionality, area usage, circuit delay and power dissipation. Low power dissipation and area usage are analysed after the functionality has been verified. Dedicated test circuitry is inserted within critical macro components in order to enhance fault coverage during manufacturing test.

For a given DSP system, the appropriate functionality has to be guaranteed to fulfil the specification, and an acceptable fault coverage has to be achieved for manufacturing. The test versus verification process is illustrated in Figure 4-1.

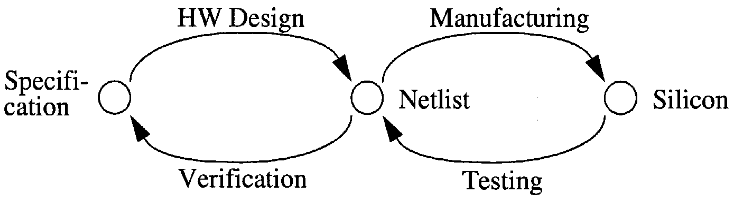

Figure 4-1 Testing versus Verification

4.1 Verification

The macro components are verified at different levels. The overall verification procedure is illustrated in Figure 4-2.

Designing a new cell commences by developing a high-level model of the required functionality. We have adopted Matlab for this purpose due to the DSP intensive nature of hearing aid applications.

The input stimuli (input.dat) for the verification of the high-level model is generated and tested using the model within Matlab. Random input data and/or typical filter data are used for the verification of the different macro components.

- Random input data (White noise, uniformly distributed, minimum value = -1, maximum value = $1-2^{-(\text{data\_width}-1)}$)
- Typical filter data (Noisy sine, $f = f_s/5$, SNR = 1)[1]

Once the Matlab model has been properly verified, a functional[2] Verilog model is developed. The functional model is verified using the generated input stimuli from the Matlab simulation and the output response of the Matlab model (output.ref) is compared with the output response of the functional Verilog model (output.dat). A netlist can be generated if the two models show identical behaviour.

Adequate performance analysis is only possible after the selection of a vendor's library. The RTL model of the different cells can then be synthesised and the same input stimuli (input.dat) is used for checking the proper functionality of the generated netlist. The netlist simulation is further used for generating a switching activity report with which the power dissipation is estimated using a power analysis tool.

Accurate timing analysis of the macro component requires the generation of adequate timing information based on pre-layout SDF back-annotation. In this case this is also useful in performing accurate power analysis.

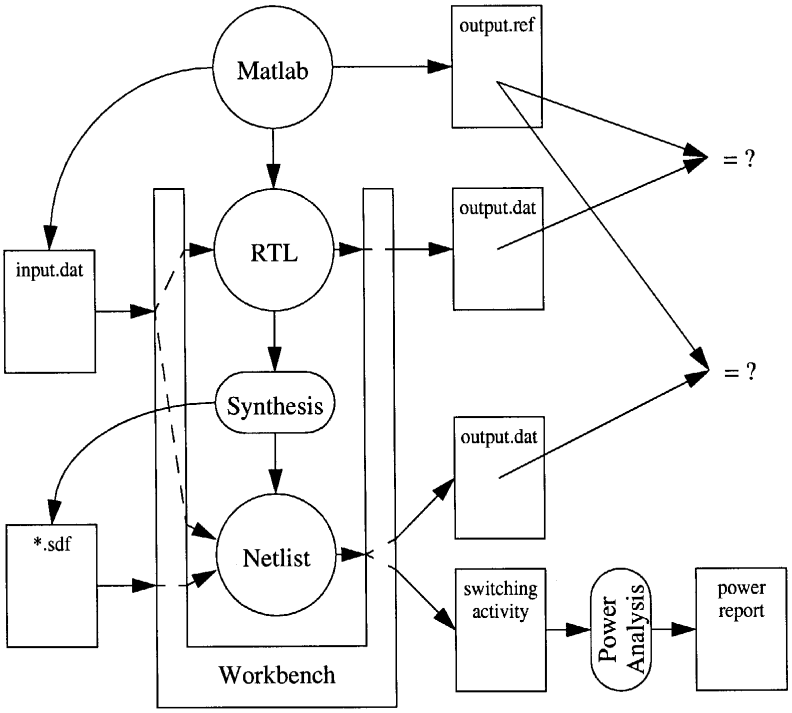

Figure 4-2 Verification process of the macro components

The above verification process is used to qualify the low area usage and power dissipation of macro component cells such as counters or multiplexer, complex memory cells as well as complete filter structures.

---

[1] *f* is the signal frequency, f_s is the sample frequency and *SNR* is the Signal to Noise Ratio.

[2] We refer to a functional Verilog model as an synthesisable RTL (Register Transfer Level) description of a cell.

4.2 Test

Once the functional verification is completed the different components also need to be verified for their Design for Testability (DfT). Figure 4-3 illustrates the test process used for checking the fault coverage and testability of the macro component cells and IPs.

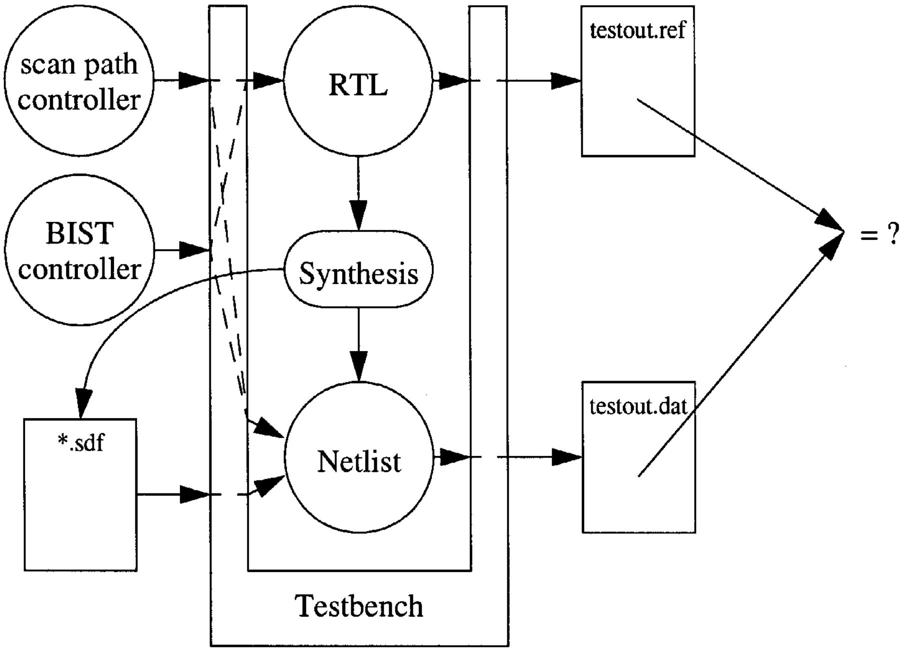

Figure 4-3 Test process of the macro components

The required fault coverage of the macro components is achieved using scan path and Built-In Self-Test (BIST). Built-In Self-Test is used for the test of data path components in hearing aid applications while scan path is used for controller related cells. The following BIST schemes have been implemented:

- RAM BIST [9]
- MAC BIST [10]
- Pseudo Random Bit Sequence (PRBS) BIST

All of the above BIST pattern generators have been implemented as parametrisable entities (e.g. data and address width). The schemes were selected for minimal area usage and maximum fault coverage while maintaining minimal test time and covering a large set of practical fault models.

The controlling of the BIST pattern generators and BIST data compressors is kept as simple as possible. This shifts a large proportion of the test control to the automatic test equipment (ATE). Models of the two controllers are used for checking the testability of the macro components and IPs as illustrated in Figure 4-3.

A test output reference response (testout.ref) is generated when verifying the RTL model of the macro components and compared with the actual output data (testout.dat) produced during netlist simulation.

5. System-on-Chip Strategy

The system on chip strategy proposed is targeted for an ARM based Platform such as the one shown in figure 5-1. The strategy connects any of the Filtering cores to both the Advanced High Performance Bus (AHB) as well as the Advanced Peripheral Bus (APB). Communication with the processor occurs using 2 handshake signals which allow fast transfer of data to/from the DSP. Whereas communication with the relatively low performance devices on the APB occur through a single handshake signal. This strategy has the advantage of allowing the DSP to operate in two different clock domains and the utilisation of power management signals such as sleep and wake-up.

In order to perform the above communication interface two sets of wrappers exist. The first allows for maximum re-usability of the DSP core and is implemented according to the Virtual Component interfacing standards of the VSIA (Virtual Socket Interface Alliance). Such an interface includes two sets of FSM type circuitry (Considering the AHB, for example, two sets of interfaces are required: the Virtual Component, in our case the DSP core, Initiator Interface and the AHB Target Bus Wrapper). For this reason additional power and area overhead are introduced for the added flexibility. The second set of wrappers allow for direct interconnection to the AMBA bus, hence only one set of interfaces is required. Power management signals are passed directly form the bus bridge.

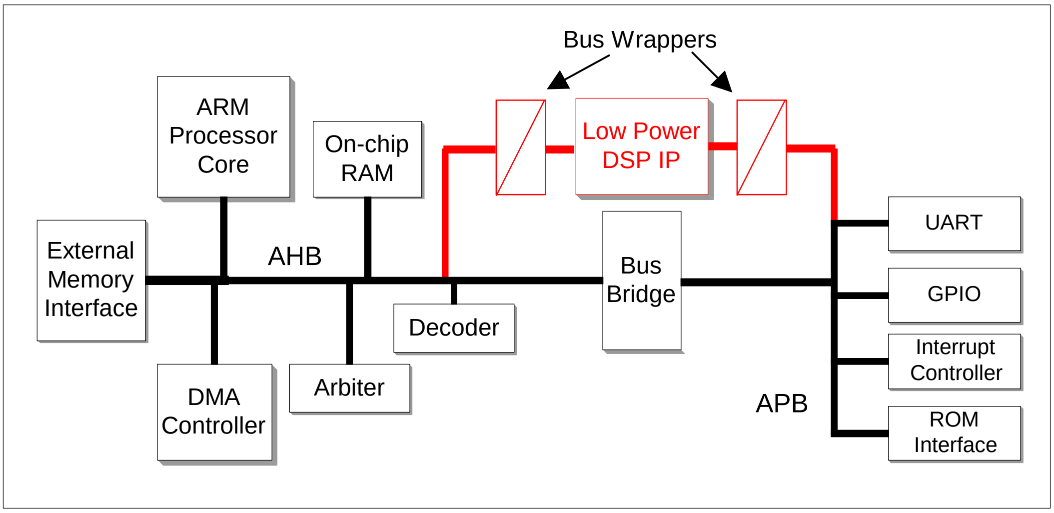

Figure 5-1: Target SoC platform and interface strategy

6. Results and Conclusions

In order to evaluate the framework, we have implemented several forms of FIR filter architectures based on hearing aid application requirements, see table 2. These filters vary in their realisation architecture and the types of macro-components employed. In summary, the following filters have been implemented:

1. Direct form FIR filter using 2s complement representation (fir_ref)
2. Direct form FIR filter using sign&magnitude representation (fir_sm)
3. Folded direct form FIR filter using 2s complement representation (fir_dfd)
4. Folded direct form FIR filter using sign & magnitude and 2s complement representation and Dadda sign & magnitude (unsigned) multiplication (fir_mixed_fdf).

We found that the design cycle for a filter could be reduced from days to hours using the macro component library framework. The macro component library is a source of high quality and highly optimised reusable cells. The library framework and the verification and test processes result in a faster design cycle without lacking important integration features such as DfT. Future work includes expanding the framework into the development of generic wireless platforms.

Table 2: Filter architecture analysis results

Filter Architecture	Performance	Fault coverage [%]	Area [u]	Power [mW]	Power/sample [nW]
fir_ref	10 MHz 24 tap 417 ksample/s	95.9	8511 (100%)	11.34	27.2 (100%)
fir_sm	10 MHz 24 tap 417 ksample/s	90.0	8282 (97%)	12.34	29.6 (109%)
fir_fdf	10 MHz 12 tap 833 ksample/s	95.6	9796 (115%)	12.96	15.5 (57%)
fir_mixed_fdf	10 MHz 12 tap 833 ksample/s	93.7	9125 (107%)	12.74	15.3 (56%)

7. Acknowledgements

The authors would like to thank Bernafon Ltd., Switzerland, for their support.

8. References:

[1] H. Neuteboom et. all., “A DSP based Hearing Instrument IC”, IEEE Journal of Solid State Circuits November 1997

[2] F. Moller et. all., “Algorithm and Architecture of a 1V Low Power Hearing Instrument DSP”, IEEE ISLPED 1999

[3] Y.-C. Park et. all., “Design of high-performance digital hearing aid processor”, Electronics Letter August 1998

[4] U. Narayanan et. all., “Low Power Multiplexer Decomposition”; IEEE Symposium on Low Power Electronics and Design (ISLPED), 1997

[5] R. Brent and H. Kung, “A regular layout for parallel adders”; IEEE Transactions on Computers, Vol. C-31 No. 3, March 1982

[6] E. E. Swartzlander Jr., “Computer Arithmetic“, Vol. 1, IEEE Computer Society Press 1990

[7] G. Keane et. all., “The Impact of Data Characteristics and Hardware Topology on Hardware Selection for Low Power DSP”; IEEE Symposium on Low Power Electronics and Design, 1998

[8] J. Monteiro et. all., “Optimization of Combinational Logic Circuits for Low Power Using Precomputation”; IEEE Conference on Advanced Research in VLSI, 1995

[9] R. Dekker et. all., "A realistic fault model and test algorithm for static random access memories”, IEEE Trans. on CAD 1990

[10] A. Paschalis et. all., “An Effective BIST architecture for Fast Multiplier Cores”, IEEE Conference on Design, Automation and Test in Europe (DATE), 1999

[11] J. Bergeron, “Writing Testbenches”, pp. 13 ff., “Testing vs. Verification“, Kluwer Academic Publishers, 2000, ISBN 0-7923-7766-4