A 660-μW 50-Mops 1-V DSP for a Hearing Aid Chip Set

Philippe Mosch, Gerard van Oerle, Stefan Menzl, Nicolas Rougnon-Glasson, Koen Van Nieuwenhove, and Mark Wezelenburg, *Member, IEEE*

***Abstract*—This article presents the flow and techniques used to design a low-power digital signal processor chip used in a hearing aid system implementing multiband compression in 20 bands, pattern recognition, adaptive filtering, and finescale noise cancellation. The pad limited 20 mm^2 chip contains 1.3 M transistors and operates at 2.5 MHz under 1.05-V supply voltage. Under these conditions, the DSP consumes 660 μW and performs 50 million 22-bit operations per second, therefore achieving 0.013 mW/Mops (milliwatts per million operations), which is a factor of seven better than prior results achieved in this field. The chip has been manufactured using a 0.25-μm 5-metal 1-poly process with normal threshold voltages. This low-power application-specific integrated circuit (ASIC) relies on an automated algorithm to silicon flow, low-voltage operation, massive clock gating, LP/LV libraries, and low-power-oriented architectural choices.**

***Index Terms*—Algorithm optimization, digital signal processing, gated clock techniques, low power design.**

I. Introduction

THE HEARING AID of which this DSP chip (Fig. 1) is a part does not replace an existing analog counterpart. It is designed from the beginning to be a high-end digital processing system with features which are only feasible with this specific technology. Examples of these features are multiband compression, pattern recognition, adaptive filtering, and noise cancellation.

The hearing aid system contains three chips which are physically stacked one on the top of the other. On the top of the structure, a standard 64-kB EEPROM stores all the fitting parameters. In the middle, the analog front-end ASIC [2] implements the programmable gain amplifier (PGA), analog-to-digital converters (ADCs), and serial interface of the signal path as well as a frequency-shift keying (FSK) receiver for the remote control of the system and support blocks such as the oscillator, booster, regulator, and references. At the bottom is the DSP chip which is presented in this article. One must notice that this chip, for assembly reasons, can not be smaller than 20 mm^2, which allows a high gate count.

Manuscript received April 4, 2000; revised June 16, 2000.

P. Mosch and N. Rougnon-Glasson are with Xemics SA, 2000 Neuchâtel, Switzerland (e-mail: philippe.mosch@xemics.com; nicolas.rougnonglasson@xemics.com).

G. van Oerle and S. Menzl are with Phonak AG, 8712 Staefa, Switzerland (e-mail: gerardo@phonak.ch; stefanm@phonak.ch).

K. Van Nieuwenhove and M. Wezelenburg are with Frontier Design, 3001 Leuven, Belgium (e-mail: koen_vannieuwenhove@frontierd.com; mark_wezelenburg@frontierd.com).

Publisher Item Identifier S 0018-9200(00)09420-8.

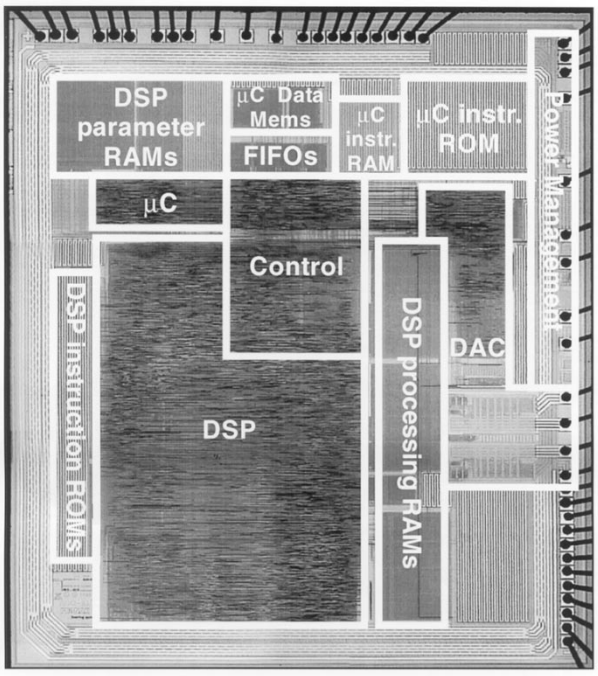


Fig. 1. Chip micrograph (parts of the design are hidden by metal filling structures).

The products based on this chipset must fit in the ear. Volume limitations for such devices are thus important and batteries have to be very small. The power consumption then becomes a limiting factor. Therefore, power consumption is the main concern of this article.

Target performances for the DSP chip were 50 million operations per second (Mops) with less than 1-mW power consumption under 1.2 V. Test coverage in production had to be over 97%, with an ability to perform a self-test in the application for a final test in the assembly line.

The article starts with an introduction in which the final hearing aid system and its constraints are presented. Then, as a reminder, well-known power reduction principles are enumerated. In the next six sections, it will be shown how these principles are applied at different levels of the design, from algorithm optimization down to gate-level implementation. The final chip performances are then discussed followed by a conclusion on the chosen approach.

II. Power Reduction Principles

The starting point of the optimization for power is (1), where V_{DD} is the supply voltage, $f_{\rm ck}$ the operating clock frequency, C

the sum of the capacitances of all the nodes in the circuit, and a the average activity of the circuit.

$$P_{\text{Dynamic}} = \frac{1}{2} {V_{DD}}^2 f_{\text{ck}} C a. \quad (1)$$

This equation is valid for conventional non-adiabatic digital CMOS designs and neglects the direct switching and static power consumptions. It is a reasonable starting point for first-order power consumption estimations and optimization strategies in applications where the power budget is dominated by the dynamic operating current.

Several principles for low-power design are derived from (1). Here, a nonexhaustive list is given for the best known of them.

1) Supply voltage reduction can be achieved through:
   a) parallelism [3];
   b) process optimization (low V_t).
2) Activity reduction:
   a) algorithmic complexity reduction;
   b) activity control at architectural level;
   c) gated clocks;
   d) memory partitioning.
3) Capacitance reduction:
   a) gate level optimizations;
   b) low-power libraries.

For our design, the decision has been made at the beginning of the project to optimize as much as possible at the higher levels of the design: algorithm, system, and architecture. The project had to rely on a standard automated flow and commercially available tools, libraries, and processes to ensure the portability and durability of the design. This removed a number of optimization possibilities, for instance, asynchronous design methodologies as in [4], special low V_t processes, or hand-optimized gate-level netlists and layouts. But even so, enough degrees of freedom remain available to achieve a very power-efficient implementation.

III. Algorithm Optimization

The algorithm optimization directly affects the activity of the design by the number of operations to be performed as well as by the width of the operands.

The operands' width has been reduced to 22 bits by performing an appropriate scaling of the algorithm. This width cannot be reduced further due to numerically sensitive feedback loops in the algorithm and to the extension of the dynamic range of the signal in the frequency domain in which most processing is performed. The gain compared to the original unscaled algorithm is of roughly 30%.

The input data are processed by frames of 32 samples. Instead of having all the processing performed during one frame, the computing of slowly varying variables has been spread over several frames. This multirate processing technique reduces the operation count by up to 60%.

The original algorithm uses the minimum required number of constants resulting from its functional definition. But some of the values calculated during the algorithm processing depend only on these constants, without any signal or state dependency. In the final algorithm, these calculations have been replaced by reading precalculated additional constants stored in memory. The constants' number increases therefore by 20% in our algorithm but the resulting operation count gain is around 15%.

Finally, for decision steps in the algorithm, estimators were used instead of exact results. The sum of the absolute values instead of the sum of the squares for a power estimation is a trivial example for such an approach. Such an optimization leads to an additional operation count gain of about 15%.

The overall optimizations result in a gain factor of more than five measured in operation counts between the original and the final bit-true algorithm, without sacrificing performance.

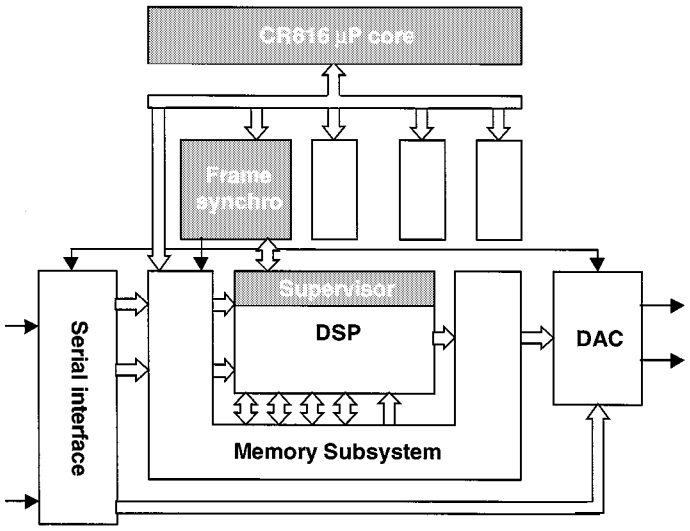


Fig. 2. Simplified block diagram with a data flow control distribution example.

IV. Distributed Control

At the chip level, a well-mastered activity management has to be enforced to bring all unnecessary functions to a near-zero power consumption when not used. One of our approaches to handle this issue was to distribute the control over the system. We implemented several small independent state machines controlling the different functions of the chip. The interactions between state machines are minimized by a proper partitioning and by turning off their clocks until a start condition is detected.

An example of that is the three levels of control used for the operation of the DSP (Fig. 2). The highest level is an 8-bit microprocessor which supervises the whole hearing aid system operation. Its power dissipation is around 200 μW in operation. However, the processor is only needed to handle exceptional events: startup and selftest, user requests either from remote control or from manual switches, fitting program changes due to new environment detection, and finally, error conditions.

The second level, the so-called frame synchronizer, performs two functions: data stream control in the signal path and communication interface between the DSP supervisor and the microprocessor. This is a very small state machine, running all the time. It provides control signals to the serial interface, to the memory subsystem, to the DSP and to the DAC to synchronize the data stream between them. At the same time it provides algorithm execution parameters to the DSP supervisor and restarts the microprocessor if special events are issued by the DSP.

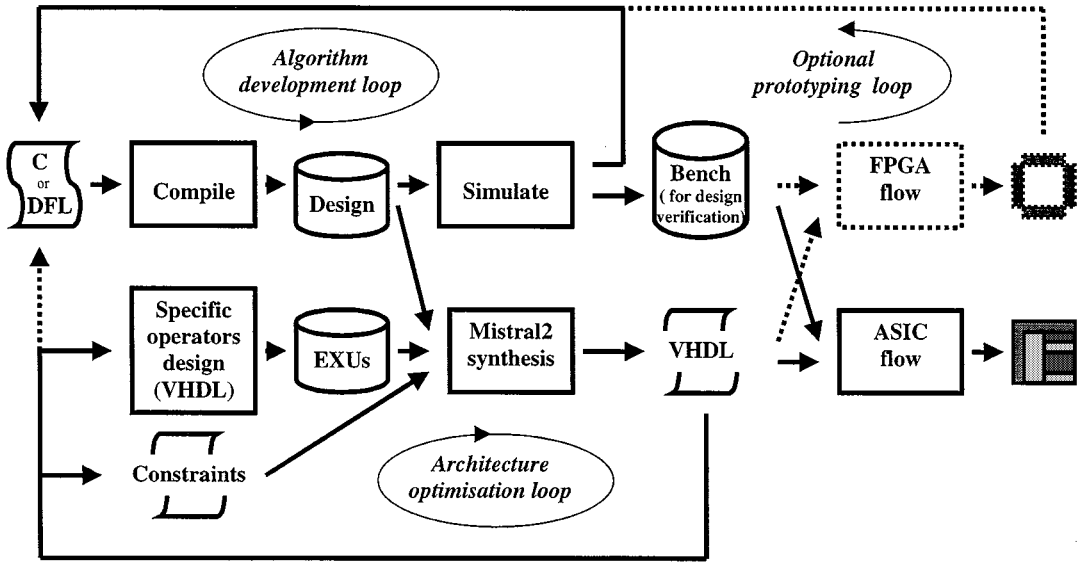


Fig. 3. Algorithm to silicon flow with Mistral2.

The third level is another small state machine which controls the algorithm execution on the DSP. The DSP can execute basic functions of the complete algorithm. According to the fitting of the hearing aid and to the particular environment, different parts of the algorithm are enabled or disabled. The DSP supervisor launches the basic functions according to the particular program running at one time. The state machine implements the control in a more efficient way than the DSP would. Also, once the processing for a group of data is finished, the DSP's clocks can be turned off, until new data become available.

This proper control distribution enables several savings in power consumption, namely:

1) to completely stop the microcontroller during the normal operation of the hearing aid;
2) to have less than 1 μW consumed in the state machine controlling the whole data stream between the serial audio input and the DAC;
3) to reduce by 10 μW the power consumption of the DSP processor using an external algorithm execution supervisor.

V. DSP Architecture Optimization

A systematic architectural solution to reduce power consumption is the usage of massively parallel processing. For a given computational performance, one can reduce the operating frequency and therefore the operating voltage and the power consumption.

The DSP processor has been generated by means of Mistral2 [5], a high-level architectural synthesis tool which uses a DFL or C description of the algorithm as input. This tool is integrated in a consistent design environment which provides support for high-level and bit-true simulation, architectural synthesis, optimization, and validation (Fig. 3).

The generic architecture of the generated processor is shown in Fig. 4. It can be noticed that there is a complete freedom on the number of execution units and on the way they are interconnected. Therefore, one can add any kind of hardware resources which are needed to enhance the processing bandwith. The exploration of the design space is very fast, so one can experiment a new architecture and its effects on performances in a couple of minutes.

The optimizations performed at the DSP level were to configure the datapath to always have the longest possible pipeline to execute a specific loop of the algorithm. This has two advantages.

1) The DSP operating frequency is reduced by parallelism.
2) The number of memory accesses is drastically reduced.

An example of the joint optimization performed on both dataflow and function level is a 128-point real-data fast Fourier transform (FFT) implementation. Typical algorithms are usually optimized for the number of operations, but rarely in terms of memory accesses, which is equally relevant from a power point of view. The implemented FFT algorithm instead minimizes memory accesses through the choice of a high radix butterfly and by integrating data windowing and filtering operations within the pipeline. This modified radix-8 implementation of a dedicated real-data FFT algorithm (derived from [6], Fig. 5) results in a seven-stage pipeline implementation (Fig. 6). The high radix not only guarantees a high throughput at low operating frequencies (158 cycles to process the 128-real-points FFT) but minimizes the number of memory accesses and operation count as well. In particular, the number of power costly general complex multiplications is limited to a mere 64 per execution of the FFT. The complete FFT processing results in 3296 22-bits operations (640 memory accesses, 544 multiplications, and 2112 additions) performed in 158 cycles; therefore the DSP executes on average over 20 operations per clock cycle.

A central element of the DSP is the complex multiplier, made out of three Wallace tree multipliers and five adders. This element is reused in several ways in different pipelines for the FFT, nonlinear functions (Figs. 6 and 7), or other parts of the algorithm. The possibility to create these different pipelines originates from the many different operations a specific

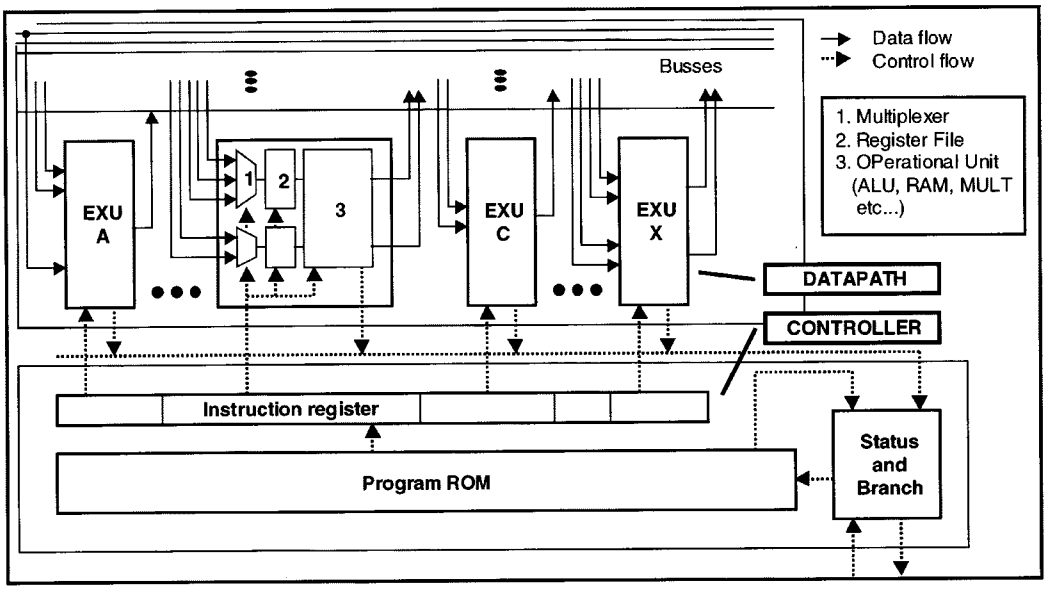


Fig. 4. Generic Mistral2-generated DSP architecture.

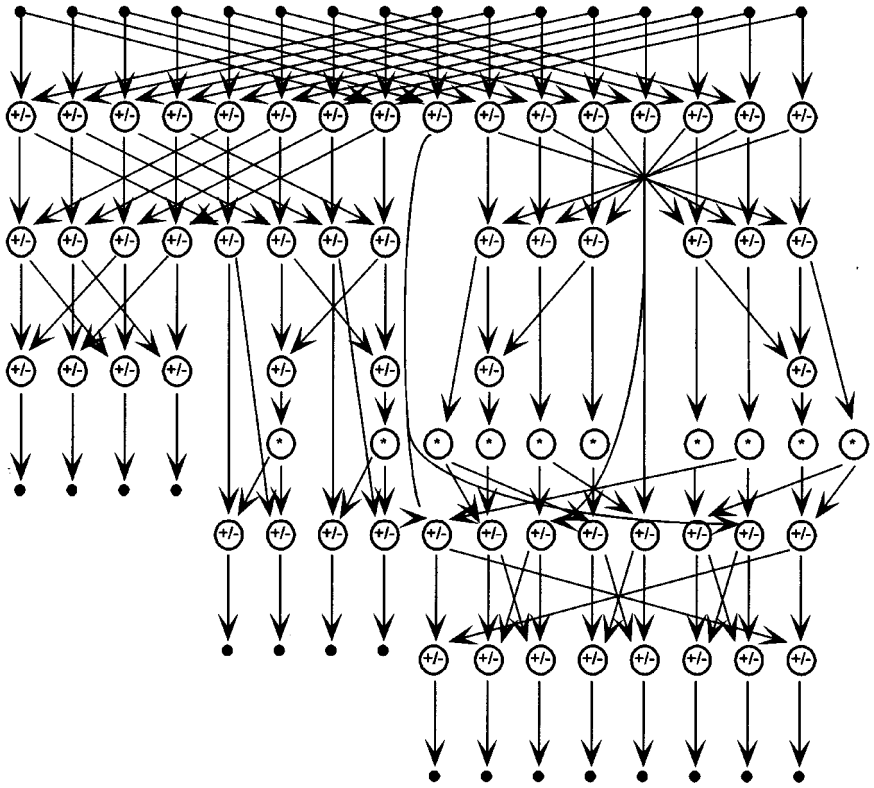


Fig. 5. Signal flow graph of one of the four radix-8 butterflies used to implement the 128-real-points FFT.

hardware unit can perform. There are twelve modes for the complex multiplier, which makes it versatile at a minimal hardware overhead cost compared to its basic function. The granularity of the operations is an important factor as well: simple general-purpose execution units can easily be reused in different pipelines. The nonlinear functions have for example been spread over different units which are combined in several ways in three-stages pipelines to compute the various nonlinear functions (Fig. 7).

Special care has been taken in the control part of the DSP to keep the inputs of the execution units stable when they are not used to perform an useful operation, therefore reducing the activity. A peephole optimization technique was implemented in the architectural synthesis tool for this purpose. The way the DSP accesses the memories has been enhanced as well, supporting now three kinds of operations, READ, WRITE, and NOP, thus taking full advantage of the almost-zero power consumption of the memories when they are not accessed.

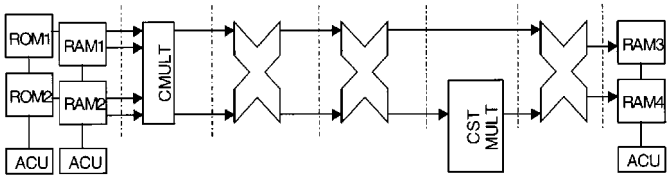


Fig. 6. Simplified radix-8 FFT pipeline (bypasses are not shown).

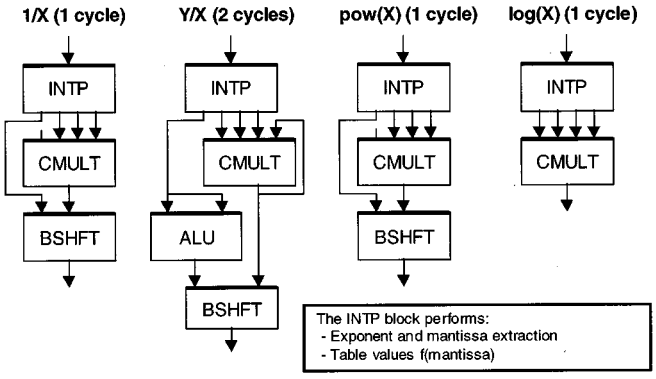


Fig. 7. Different pipelines for nonlinear functions using shared resources.

VI. DSP Instruction ROM Splitting

One has to notice that due to its general nature, the number of control signals of the datapath grows very quickly, since each execution unit has control signals to select the operation to be performed, to select the data sources and to control the read and write operations of each register file. In our design, we have 340 control signals in the datapath. The control, which includes the instruction memory and the instruction register, becomes therefore the most important contributor to the power consumption of the DSP.

Due to the small memory depth (less than 1500 words), decoder circuits that would allow an increase in the code density in the ROM would also have led to an increase in power consumption. Also, an efficient caching system would have required a depth such that its associated power consumption would have been higher than the expected instruction ROM power-consumption reduction. Therefore, no satisfactory and systematic approach has been found to reduce efficiently the power consumption of the control part of the DSP.

However, an opportunistic approach could be developed to partially solve the problem. It relies on the observation that it is possible, with a very low penalty in terms of execution cycles, to partition the datapath for two different kind of operations: FFT and non-FFT. By reducing the resources shared by the two kind of operations to the minimum (memories and complex multiplier), it is possible to partition effectively the instruction ROM in two parts, with only one ROM being accessed at each execution cycle (Fig. 8) and part of the instruction register being clocked. This solution reduces the activity of the circuit and also its equivalent capacitance since the sum of the sizes of the two memory partitions is smaller than the size of the initial memory.

The power consumption of the control part for the DSP was reduced by 40% in this way, leading to an overall reduction of 20%.

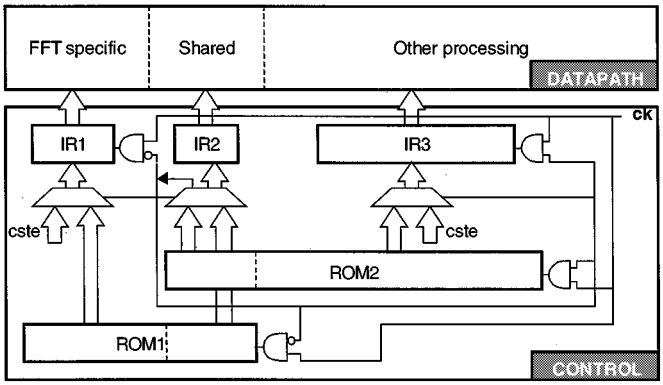


Fig. 8. Splitting of the instruction ROM of the DSP to reduce activity.

VII. Clock Gating

A systematic way to reduce power in synchronous designs is the usage of gated clocks. The gated clocks are a very well-known technique which becomes more and more popular since it is being supported in synthesis tools.

But despite the evolution of the tools, our implementation of gated clocks remains fully manual, each gating being explicitly described in the VHDL code. This gives us the ability to perform gating that tools still do not introduce automatically, namely:

1) hierarchical gating;
2) low-frequency clock domains;
3) stopping the clocks of state machines.

Hierarchical gating is essential to avoid power dissipation by the clock tree of an inactive part of the design. So each significant block of the design, such as the DSP or the 8-bit microcontroller, has its own gated clock domain.

Low-frequency clock domains are implemented in such a way that they do not require new clock definitions for the register transfer level (RTL) synthesis tool or for the clock tree generation. The principle is to generate clock pulses at the desired frequency through clock gating rather than obtaining lower-frequency clocks at 50% duty cycle through division of the main clock. This results in the fact that the design, from the RTL synthesis or clock tree generation tool point of view, remains with one unique clock root, making the definition of multiple clock domains unnecessary. This makes low-frequency clocks easy to implement and to manage. For this reason, they are extensively used.

As mentioned previously, inactive state machines are not clocked until a wake-up condition is detected. Finally, each data register of the DSP and of the microcontroller is individually clocked.

Using these specific gated clock techniques leads to 200 different clock domains in the chip, most of them related to the gating of the clock of each individual register in the DSP. Nevertheless, our methodology remains fully synchronous and easy to use in an automated design flow, with one unique root clock definition in the design. The gain in power consumption by the massive usage of gated clocks in our DSP is estimated to be around 150 μW.

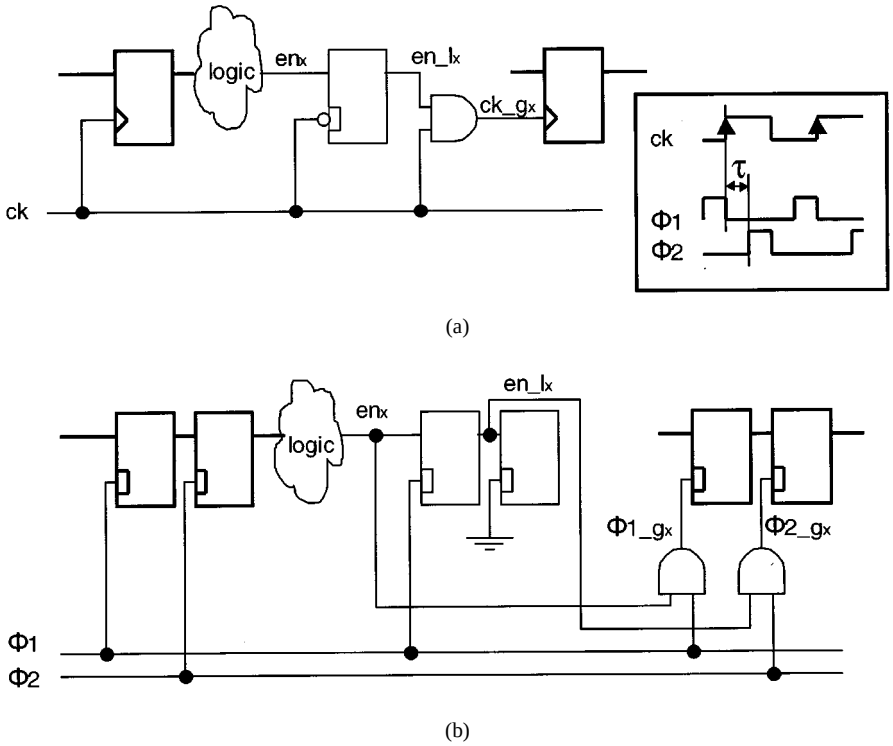


Fig. 9. Gated clock implementation using a master/slave design style. (a) Conventional gated clock implementation for clock domain x. (b) Equivalent master/slave gated clock implementation for clock domain x. Dummy latch, inactive in normal mode, is used in test mode to have all memory elements in a scan chain.

VIII. Low Voltage Operation

Supply voltage reduction is a very efficient way to reduce power consumption, but architectural optimization to reduce the requested operating frequency is not enough to perform voltage reduction. There are several issues which are to be solved for a power-efficient low-voltage operation:

1) efficient dc/dc conversion, from a varying battery supply voltage to the minimal requested operating voltage;
2) low-voltage robust design style;
3) libraries qualified for low-voltage operation.

The dc/dc conversion is handled at the hearing aid system level and is flexible enough to deliver to the DSP a fixed supply voltage which can be set between 1.05 and 1.25 V with a battery supply voltage varying between 0.9 and 1.5 V and with an overall efficiency over 85% in the whole operating domain [2].

Regarding design robustness, our clocking strategy is oriented toward skew-insensitive operation. This is required due to the use of standard tools which generate distributed balanced clock trees, to the low-voltage operation and to the difficulty to extract accurate parasitics in a 0.25-μm technology. Due to increasing mismatches in sub-micron technologies and to the high nonlinear sensitivity of the cells' speed to V_t and voltage fluctuations at low voltage, high margins have to be taken on hold times to ensure correct operation over the whole process and operating range. These margins are generally achieved by power-consuming buffers inserted in the fast paths of the design. Our approach, instead, was to use a master/slave design style with nonoverlapping phases (Fig. 9). The technique is more demanding in terms of VHDL coding, since it is actually not directly supported by the RTL synthesis tools when using a standard description. In order to support our approach, the Mistral2 libraries have been modified to generate automatically the design in the master/slave gated clock style and in a format proper to be used easily with the chain of tools. Notice that the chosen design style also enables an automatic scan chain insertion and the usage of automatic test pattern generation (ATPG) tools, the resulting chip test coverage actually exceeding 98%.

One must notice that despite the fact that two clock lines have to be routed through the whole design, the master/slave approach brings gains in terms of clock tree power consumption. This is due to the fact that the skew constraints can be much relaxed on master/slave-based designs, even to the point that clock tree insertion tools are not absolutely required. In terms of power consumption, we could observe a gain of 60 μW by using a clock tree generated by a synthesis tool, only fixing design rules like maximum load and maximum rise and fall times, instead of using the clock tree insertion of the place and route tool.

The design uses the commercially available CoolLib library which is composed of branch-based standard cells [7] (Fig. 10) and LP/LV memories with no static consumption between memory accesses due to their straightforward sense-amplifier free design. The RAMs use divided wordline techniques and the ROMs use simple partial swing and selective pre-charge (Fig. 10) techniques to reduce their power consumption.

The library is qualified to operate down to 0.9 V in all the corners of the process.

IX. Performance

As a result of the combination of all the above techniques, the circuit exhibits a measured performance of 75 giga-operations per Watt, which is close to the "intrinsic computational

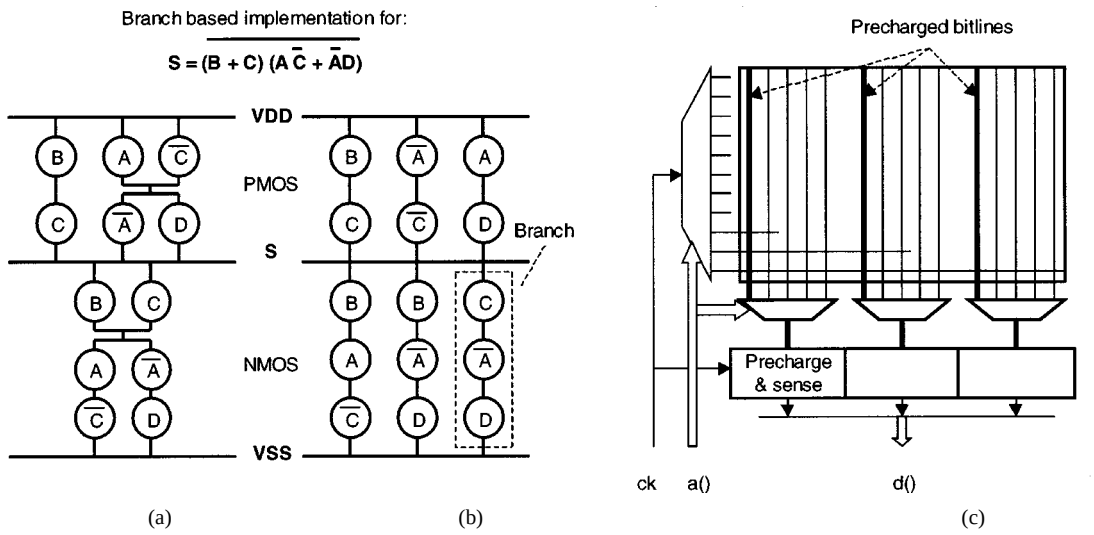


Fig. 10. Branch-based standard cells and low-power ROMs. (a) Conventional implementation: possibly uses pass gates; irregular layout. (b) Branch-based implementation: No pass gates, LV operation; regular layout, high density. (c) Bitline precharge through multiplexers easily enables selective precharge and swing limitation.

TABLE I
PERFORMANCE COMPARISON WITH THE DSP OF [1]

Performance criteria	Referenced DSP performance	Presented DSP performance	Gain ratio	Comment
Power [uW]	800	660	1.2	
MOPS *	12	50	4.2	For the comparison, it is assumed that 1 MAC corresponds to 3 Ops (1 memory access, 1 multiply, 1 add)
Data wordlength	16	22	1.4	If the algorithm had effectively to be implemented on a 16 bit DSP, the ratio would most probably be worse in terms of operations count
TOTAL			7.1	
* Number of 256 complex points FFTs processed per second	690	3125	4.5	First order cross validation of the chosen MAC/OPs ratio

efficiency of silicon" figure of [8] for the given technology and timeframe.

Table I gives a comparison in terms of power efficiency between the presented design and a recent programmable low-power DSP in the same field of application [1].

It is important to notice that, due to the implementation methodology, our design does not permit a reprogrammation of the algorithm. The flexibility of our DSP remains in the boundaries of what has been defined in the algorithm at the time of implementation. This is reasonable in the perspective of hearing aid systems in which implementing more complex algorithms requires first an increase in power efficiency of the hardware to become feasible.

It is worth mentioning that the drawbacks of the mainly non-programmable approach are compensated by the achieved performances and the use of a quick well-supported automated flow going from a high-level algorithm description down to the silicon implementation. This makes the design easily portable to new processes to benefit quickly from their resulting performance increases. Only six months are requested from the time the algorithm is defined to the time functional circuits become available for hearing aids assembly.

X. CONCLUSION

In this article, it has been demonstrated that outstanding power consumption performances can be achieved by using an automated algorithm to silicon flow. None of the presented techniques or tools are new. Much more than innovation, it is a combination of techniques at all levels which permitted the target to be achieved. The main optimization effort was spent on the higher levels of the design, especially on the algorithm, where the highest gains can be expected. At the lower levels, only straightforward, systematic, and tools-supported solutions were used.

Another important point is that a lot of effort was spent to get all teams involved in the project aware of the impact of their work on the power budget. This was not an easy task and has been supported through the project by having a constant tracking of the power budget, by means as simple as spreadsheets-based

calculations. The effects on power of algorithm changes, architecture changes, and implementation changes could quickly be evaluated based upon rough first-order estimations.

Finally, the proposed flow enabled quick improvement of the performance of the system by taking the benefit of the new available processes. Retargetting the design to a 0.18-μm technology with an estimated minimal operating voltage of 0.80 V will enable an additional power consumption reduction of 42%.

ACKNOWLEDGMENT

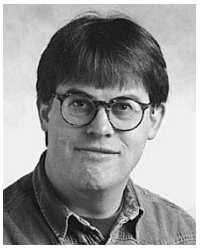
The authors would like to thank all the people from Xemics, Phonak, Frontier Design, CSEM, and TSMC who made this project feasible, and all the great people who worked in or supported this project. They are proud to be part of the large team who made this work possible. They also thank especially their families for their patient support of work which sometimes became an overnight and weekend activity.

REFERENCES

[1] F. Møller, "Algorithm and architecture of a 1-V low-power hearing instrument DSP," in *Proc. ISLPED*, Aug. 1999, pp. 7–11.
[2] R. Klootsema *et al.*, "Battery-supplied low-power analog–digital front-end for audio applications," in *Proc. ESSCIRC*, Sept. 2000, pp. 156–159.
[3] A. P. Chandrakasan *et al.*, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473–484, Apr. 1992.
[4] L. Nielsen *et al.*, "An 85-μW asynchronous filter-bank for a digital hearing aid," in *Proc. ISSCC*, vol. 41, Feb. 1998, pp. 108–109.
[5] M. Van Canneyt *et al.*, "Power optimization of telecom baseband functions using architectural exploration," presented at the DSP'96, Copenhagen, Denmark.
[6] M. Wezelenburg, "General radix 2^n DCT and DST algorithms," in *Proc. ECCTD*, Aug. 1997, pp. 645–649.
[7] C. Piguet *et al.*, "Low-power low-voltage digital CMOS cell design," in *Proc. PATMOS, 4th Int. Workshop Power and Timing Modeling, Optimization and Simulation*, Oct. 1994, pp. 132–139.
[8] T. A. C. M. Claasen, "High speed: Not the only way to exploit the intrinsic computational power of silicon," in *Proc. ISSCC*, vol. 42, Feb. 1999, pp. 22–25.

Philippe Mosch was born in Zürich, Switzerland, in 1965. He received the M.S. degree in microengineering from the Swiss Federal Institute of Technology, Lausanne, Switzerland, in 1991.

In 1991, he joined the IC Design Division, Swiss Center for Electronics and Microtechnology (CSEM), Neuchâtel, Switzerland, where he worked on the realization of standard cells libraries and embedded memories and on the development of watch and medical application circuits. He joined Xemics SA, Neuchâtel, at the company creation in 1997. Since then, he has been in charge of the μC and DSP group as Design Group Manager. The group develops 8-bit and 32-bit microcontrollers as well as IP or custom DSP processor-based circuits. His fields of interest are methodologies and automated flows for low-power/low-voltage circuit design.



Gerard van Oerle was born on August 18, 1960, in Zaandam, the Netherlands. He did his studies at the Twente University of Technology, Enschede, the Netherlands. His practical training and his M.S.E.E. thesis work were done at Philips Semiconductors Zürich, Switzerland, from 1985 to 1987.

In 1987, he joined Philips Semiconductors Zürich, where he worked in several design groups. He joined Phonak Hearing Instruments, Staefa, Switzerland, in 1994, to work on ASICs for hearing instrument applications. Since 1995, he has been the Group Leader of the Systems Design Group. His main interests are low-power techniques, body-worn applications, and mixed-signal systems.

Stefan Menzl was born in St. Gallen, Switzerland, in 1971. He received the M.S. degree in electrical engineering from the Federal Institute of Technology, Zürich, Switzerland, in 1997.

He is currently an ASIC Design Engineer in the Research and Development Department of Phonak Hearing Systems, Staefa, Switzerland.

Nicolas Rougnon-Glasson was born in Pontarlier, France, in 1974. He graduated from the Ecole Superieure d'Electricite, Paris, France, in signal processing and electronics in 1996.

Since 1998, he has been with Xemics SA, Neuchâtel, Switzerland, as an IC Design Engineer, in the Microcontrollers and DSP Group.

Koen Van Nieuwenhove was born in Aalst, Belgium, on August 13, 1957. He received the M.S.M.E. and M.S.E.E. degrees from the Katholieke Universiteit Leuven, Belgium, in 1981 and 1982, respectively. He received the M.B.A. degree from the Katholieke Universiteit Leuven in 1995.

In 1983, he became a Project Engineer, active in the design of microprocessor-based measurement systems, at the Royal Military Academy, Brussels, Belgium. In 1985, he joined Silvar Lisco, where, as an Research and Development Software Engineer and later an Engineering Manager, he developed CAD tools for the design and layout of ICs. In 1990, he became Design Services Manager at EDC/Mentor Graphics. He currently holds the position of Vice President of Engineering at Frontier Design, Leuven, and is responsible for the Software Development and Design Service Groups. These groups are active in digital signal processing, high-level synthesis, and low-power design.

Mark Wezelenburg (M'97) was born in Akersloot, The Netherlands, in 1970. He received the M.Sc. degree in electrotechnical engineering from the Technical University, Delft, The Netherlands, in 1994.

From 1994 to 1998, he was with the Swiss Center for Electronics and Microtechnology, Neuchatel, Switzerland, as a Research and Development Engineer in the Department of IC Design and Data Communications, designing high-quality power-efficient algorithms for speech processing and enhancement. He participated in the European ACTS project working toward the definition of the new third-generation mobile telecommunication standards. He is currently a Design Consultant at Frontier Design, Leuven, Belgium, working on the design and design methodology of high-performance systems-on-a-chip. His interests and activities are the joint design and optimization of signal processing algorithms and their computer architecture for multimedia, robust adaptive filtering, speech recognition, and telecommunications.