Published in 7th International Symposium on IC Technology, Systems & Applications (ISIC-97), 649-652, 1997
which should be used for any reference to this work

AN AUTOMATED DESIGN METHODOLOGY FOR THE MAPPING OF DSP ALGORITHMS INTO LOW POWER VLSI ARCHITECTURES

Alexandre HEUBI, Peter BALSIGER and Fausto PELLANDINI

Institute of Microtechnology, University of Neuchâtel, Rue A.-L. Breguet 2, CH-2000 Neuchâtel, Switzerland

Phone: ++41 32 718 34 11, Fax: ++41 32 718 34 02, E-Mail: alexandre.heubi@imt.unine.ch

Abstract. A design methodology suitable for an effective low power VLSI implementation of a large class of digital signal processing algorithms is presented, which shows to be particularly well-adapted to fulfil the requirements of portable and autonomous microsystems. Starting with the precise specifications of the application algorithms, an appropriate scheduling method is first applied to optimize the data flow, followed by an assignment method which produces the detailed architecture. The actual VLSI implementation is then performed, resorting to commercial logic synthesis and place&route tools. As an example, the implementation of an algorithm suitable for all-digital hearing aids is discussed. The resulting silicon area is less than 20 mm^2 for a 1 µm CMOS process, and the measured power consumption at a sampling rate of 16 kHz is about 300 µW at 1.2 V.

1. Introduction

In the fields of telecommunications, hearing aids, and electronic instrumentation, there is an important demand for new generations of high performance portable applications. The requirements of such applications are manifold, and can be expressed in terms of improved functionalities, enhanced miniaturization, and reduced power consumption to extend the system autonomy.

For flexibility and efficiency reasons, the signal processing facilities to be incorporated are more and more realized digitally on dedicated ASICs. However, in order to achieve optimal solutions with respect to the power consumption, the designs should be performed coherently and carefully at all levels, including the algorithmic, architectural, logical, and layout levels.

This paper proposes a contribution to the design and optimization of low power VLSI architectures suitable for effective implementations of various digital signal processing (DSP) algorithms such as FIR and IIR filters, adaptive filters, and Fourier transforms, with a particular strength on the power consumption which should be kept as low as possible.

The second part of this paper presents the principles for low power and the selected architecture. The third part is devoted to the top-down methodology where the special optimisation tools are explained. The fourth part concerns an application example with measurements results and a last part presents some conclusions.

2. Processor architecture

For digital CMOS circuits, the power consumption is known to be essentially determined by the dynamic consumption, which can be expressed as [Pigu92]:

$$P = f \cdot C_{eq} \cdot Vdd^2 \qquad \text{(EQ 1)}$$

where f is the clock rate, C_{eq} the equivalent capacitance, and Vdd the supply voltage of the considered circuit. All three quantities should be kept as low as possible.

Substantial savings can be achieved with respect to the equivalent capacitance C_{eq} by strictly limiting the processing activities to the resources providing a direct contribution to the required computations. Hence, the next principles were applied:

a) Idle modules should be set into a power-saving STAND-BY mode.
b) The processor architecture and modules should be organized in a way to limit the overall data transfer to the strict minimum, local data traffic being preferred versus global traffic.
c) Larger memories should be split into a set of smaller memories, where a single one is active at a time.

Also, the structural regularity of most DSP algorithms should be utilized to simplify both the scheduling and the hardware implementation.

Finally, there is a global trade-off to be determined between f, C_{eq}, and Vdd in order to obtain the optimal power consumption. Indeed, reducing the supply voltage lowers the maximum achievable clock rate due to the increased signal propagation delay. Hence, in order to cope with the required computation throughput, it is usually necessary to extend the parallelism of the architecture [Chan92], which in turn will have an effect on the equivalent capacitance! In this context, the architecture should be flexible enough to let the degree of parallelism be best fitted to the considered application.

2.1 Hardware model

The architecture determines the type of hardware used for the implementation of the algorithms. It has to be defined in such a way that it can be used for a large class of algorithms. The architecture must also be well suited for the associated optimisation tools (next section) and enable low power solutions.

The hardware is composed of mainly four types of building blocs (figure 1):

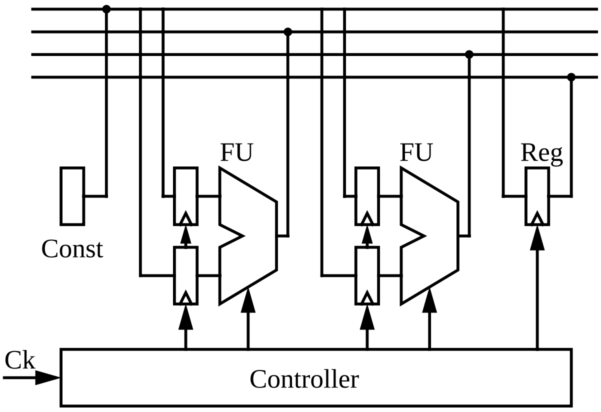

FIGURE 1. Hardware model

- The functional units to perform the operations of the algorithm to implement. Each operation must have a corresponding functional unit able to perform the calculation. Most common operations in digital signal processing are covered by an optimized low power functional blocs library. For specific needs it is possible to extend this library.
- The memories to store temporary data, longer term data and the constants. Normal registers, register files or Random Access Memories (RAM) can be used.
- The interconnections to transfer the data between functional units and memories. In order not to be limited by data transfers, each functional unit and each register has its own output bus that can be connected to any input. Of course, after optimisation, only the necessary connections will be implemented.
- The controller to generate all signals for data transfers and functional units. As the target is to implement data independent signal flow graphs, the controller can be realised with a simple state machine.

Input and outputs are considered in this model as simplified functional units. Inputs look like constants (figure 1) and outputs like an outputless single input functional unit.

The gated clock concept is used throughout the design to limit the activity of each part to the strict minimum. Registers are placed at the input(s) of the functional units keeping stable data when they are idle. This concept has also the advantage that registers can store data over multiple cycles without any input multiplexer as it is the case for pure synchronous designs. To guaranty proper work with gated clocks, it is necessary to take special care at the logic synthesis phase by introducing delay constrains on the clock signals.

3. Top-down design methodology

Considering the complexity of implementing DSP (Digital Signal Processing) algorithms onto parallel architectures on silicon, it is necessary to resort to a design methodology supported by a CAD environment suited to the specificities of the problem.

Therefore, a design methodology for low power microsystems has been developed. It follows a top-down approach (fig. 2) allowing first to specify the DSP algorithm at high level and to simulate it. In a second step, the implementation is optimized by the scheduling and the assignment of the algorithm’s operations. Finally, the circuit can be realized in an ASIC (Application Specific Integrated Circuit) or on an FPGA (Field Programmable Gate Array).

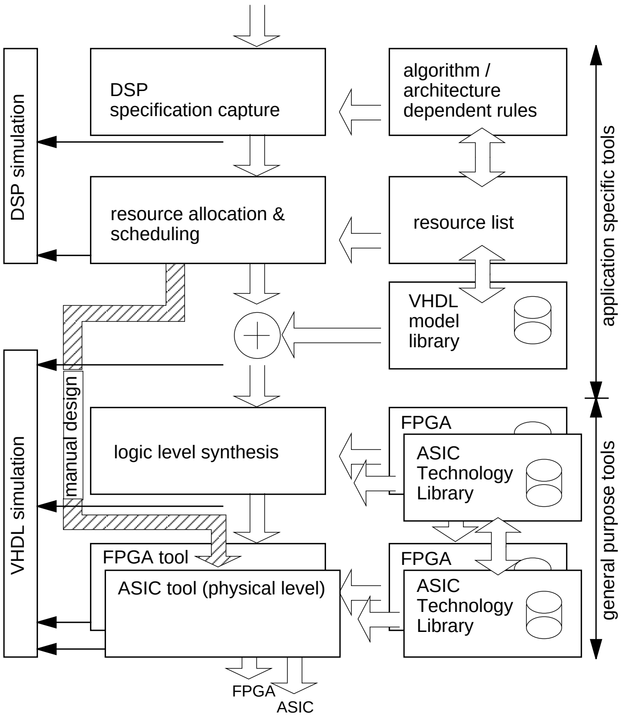

FIGURE 2. Top-down design methodology

Commercial tools are used for the specification, the simulations, the logic synthesis and the place and route. The resource allocation, defining on which type of functional unit each operation has to be performed, has to be done by the user.

Special tools, suited for low power architecture have been developed for the scheduling and assignment steps. They use optimized functional units, selected from the resource list, and the target architecture described in the previous section.

The methodology and it’s related tools allows in particular:

- To reduce the clock frequency to save energy and/or to meet the specification at very low supply voltages (around 1 Volt),

- To explore several design alternatives thank to a fast turn around time to,
- To design maximum throughput architectures,
- To produce a cost effective solution for medium to high production volumes,
- To reuse previously developed blocs.

3.1 Scheduling

The scheduling process is particularly important to optimize the throughput or, for voltage scaling, to maintain a given throughput and simultaneously to minimize the amount of resources and so the silicon area.

This process is implemented using the so called "Tabu Search" optimisation technique [Glo93]. It is a time constrained algorithm; the control step number is fixed and the resource number is optimized. This process is able to handle inter-iteration parallelism by an implicit retiming (relocation of the delays of a data flow graph) in the same way as in [Hee92]. Rate optimal (defined by the critical loop of a given algorithm [Ger92]) architectures can thus be synthesised. The objective function to minimize in this step is given in equation 2.

$$C_{total} = \sum_i n_i \cdot c_i \qquad \text{(EQ 2)}$$

where: C_{total} is the total cost, n_i the number of resource of type i and c_i the cost of resource type i, given by the user.

This cost function includes also register and bus costs. It is also straightforward to cope with others costs in order to account for signal statistics for instance.

3.2 Assignment

The used assignment process is derived from [Mign91]. It's task is to bind the scheduled operations on specific functional units and to handle the memorization of intermediate data. It is an iterative process whose goal is to minimize the amount of interconnection resources.

At each control step, the cost of the assignment of each operation and register on all resources is computed, taking into account the global costs of the actual solution. Then an optimal permutation is found by resolving a matching in bipartite graph problem. The process is repeated over several iterations of the data flow graph.

The assignment tool is fast and especially well suited for handling regular structures which are frequent in digital signal processing.

Further power reductions can be achieved when signal statistics are considered [Cha96].

3.3 VHDL output

The previous optimization processes are followed by a VHDL writer which produces a synthesizable RTL (Register Transfer Level) description of the hardware. The multiplexers at the inputs of the functional units and registers can be implemented with standard static logic element or with tri-state drivers.

4. Application example

As an application example, the implementation of an adaptive speech processing algorithm suitable for all-digital hearing aids will be discussed. The considered algorithm aims at improving the speech intelligibility/ quality by spectral sharpening.

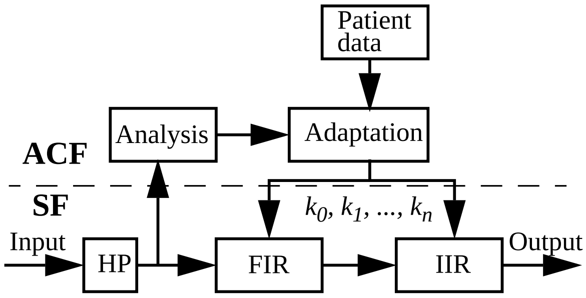

FIGURE 3. Block diagram of the digital hearing aid

The bloc diagram of the algorithm is shown in figure 3. The input signal goes first through a high-pass filter. Then an analysis combined with an adaptation considering patient data supplies a joined FIR-IIR filter with coefficients. The FIR-IIR filter is realized with lattice structures for their good numerical properties.

The complete data flow graph is made of some hundreds elements from which 68 are MAC (Multiply and ACcumulate) operations.

4.1 DSP implementation

This algorithm was first implemented and tested on a commercial DSP, the TMS320C50. The algorithm requires 783 instruction cycles for each input sample. This leads to 12.5 MIPS (Mega Instructions Per Second) or a clock rate of 25 Mhz for a 16 kHz sampling rate. The power consumption is then about 15 mW at 3.3 V.

4.2 ASIC implementation

The same algorithm is implemented in an ASIC using the presented methodology. To meet the throughput at voltages as low as 1 V, two MAC units are used. One MAC unit is devoted to the operations of the upper part (ACF on figure 3) and the other one for the signal path SF. Besides those units, an ALU and some specific functional blocs are used.

The scheduling can be done in 35 cycles per sample, resulting in a instruction frequency of only 560 kHz for a 16 kHz sampling rate.

The patient data are copied from an external EEPROM into an internal RAM at reset time. The circuit has also all necessary serial interfaces for the communication with the Analog to Digital and Digital to Analog converters.

The logic synthesis and the place and route are realised with the COMPASS tools with the low power standard cell library CSEL_LIB for ALP1lv, a 1 µm technology from EM Microelectronic Marin, Switzerland.

The final chip size is less than 19 mm^2 for the core and 22 mm^2 including pads (figure 4). The gate level simulations have shown a power consumption of about 250 µW at 1.2 V supply voltage.

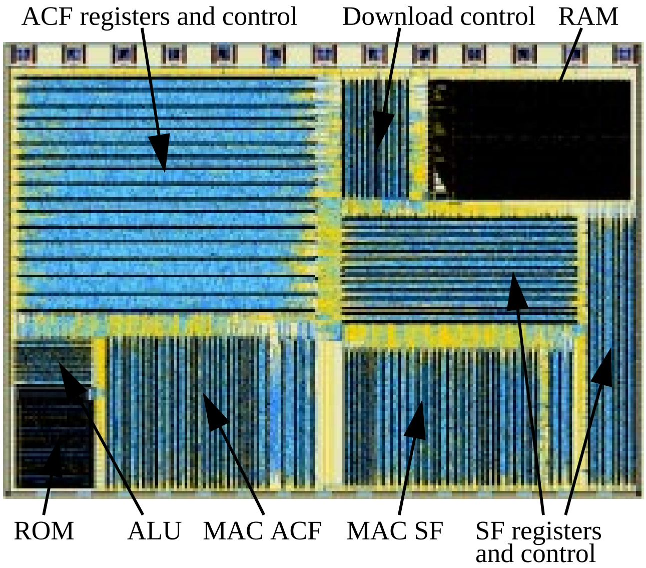

FIGURE 4. Chip layout

In an up to date 0.35 µm 5 metals technology, the silicon area of the core would be only about 2.5 mm^2.

4.3 Measurements results

Ten test chips were fabricated and encapsulated in DIL 24 packages. The functional test is made on a HP 16500 logic analysis mainframe with a vector generation card and an analysis card. From the ten chips, nine are functionally correct.

The average power dissipation is 295 µW at 1.2 V which is a slightly higher than simulated. This can come from the influence of the initialisation phase on the average simulated consumption. The computing part is in fact idle during the EEPROM download at reset time.

The measured behaviour of the power consumption versus supply voltage is shown in figure 5. This measure confirms the validity of equation 1 for the dynamic power consumption where $P = k \cdot Vdd^2$ with k constant.

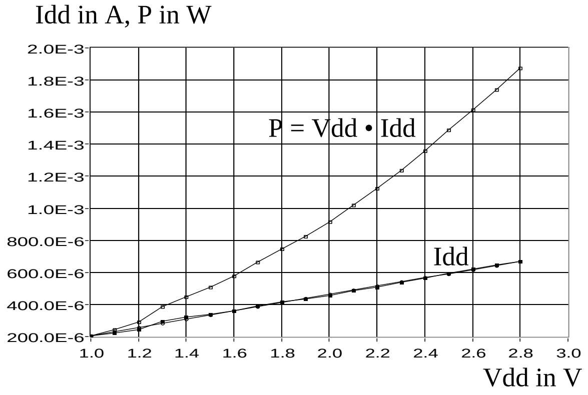

FIGURE 5. Power consumption versus supply voltage

The linear function of the power versus frequency has also been verified showing that the static power is negligible.

5. Conclusion

This paper proposes a complete top-down methodology for the low power VLSI implementation of DSP algorithms on hardwired architectures. The input algorithm is described with a data flow graph. Scheduling and assignment of the operations are performed automatically on a target architecture with dedicated tools, optimized for low power. Finally the architecture is translated in a Register Transfer Level synthesizable VHDL code.

This methodology is in particular intended to be used for portable and autonomous microsystems, and was applied as an example to the implementation of a spectral sharpening algorithm for digital hearing aids.

6. Acknowledgments

The authors would like to thank Michael Ansorge from the Institute of Microtechnology of Neuchâtel, Switzerland and Prof. Dr. Sabih Gerez from the University of Twente, Netherlands for their continuous support and valuable inputs.

References

[Chan92] A. P. Chandrakasan, S. Sheng, and R. Brodersen, “Low-Power CMOS Digital Design”, IEEE J. of Solid-State Circuits, Vol. SC-27, no. 4, April 1992, pp. 473-484.

[Cha96] Jui-Ming Chang and Massoud Pedram, “Module Assignment for Low Power”, Proc. European Design Automation Conference, Geneva, Switzerland, September 1996, pp. 376-381.

[Ger92] S. H. Gerez, S. M. Heemstra de Groot, O. E. Herrmann, “A Polynomial-Time Algorithm for the Computation of the Iteration-Period Bound in Recursive Data-Flow Graphs”, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Application, Vol. 39, no. 1, January 1992, pp. 49-52.

[Glo93] F. Glover, M. Laguna, E. Taillard, D. de Werra, “Tabu Search”, Annals of Operations Research, Volume 41, éd. J. C. Baltzer AG, Basel, Switzerland, May 1993, ISSN 0254 5330.

[Hee92] S. M. Heemstra de Groot, S. H. Gerez, O. E. Herrmann, “Range-Chart-Guided Iterative Data-Flow Graph Scheduling”, IEEE Transactions on Circuits and Systems, Vol. 39, no. 5, May 1992, pp. 351-364.

[Mign91] A. Mignotte, M. Crastes de Paulet, “Resource Assignment with Different Target Architectures”, Proceeding EURO ASIC’91, Paris, France, 1991, pp. 172-177.

[Pigu92] C. Piguet, V. von Kaenel, J.-M. Masgonty, J.-F. Perotto, R. Klootsema, “Basic Design Techniques for both Low-Power and High-Speed ASICs”, Proc. EUROASIC’92, Paris, France, 1992, pp. 220-225.