

OVERVIEW AND DESIGN DIRECTIONS FOR LOW-POWER CIRCUITS AND ARCHITECTURES FOR DIGITAL SIGNAL PROCESSING

T. Arslan, D.H. Horrocks, and A.T. Erdogan¹

1. Introduction

Power dissipation is becoming a limiting factor in the realisation of VLSI systems. The principal reasons for this are maximum operating temperature and, for portable applications, battery life. Because of the relatively great complexity, the power dissipation in Digital Signal Processing (DSP) applications is of special significance, and low-power design techniques are now emerging.

This paper provides an overview of these techniques and aims to serve as a bibliography of the key papers relevant to low-power DSP design. In addition, the paper presents indications for potential design directions bearing in mind the architectural complexity and the speed requirements of today's systems.

CMOS logic is assumed since this is currently the most commonly used VLSI technology due to its high degree of integration which is in turn allowed by its scaling properties and low power dissipation.

2. Power Dissipation in Digital CMOS Circuits

The main sources of power dissipation, in a typical CMOS digital circuit (Figure 1), are given by the following equation:

$$P_{ave} = P_s + P_{sc} + P_l \quad (1)$$

where P_{ave} is the average power dissipated by the circuit, P_s is the switching component of the power caused by charging/discharging of the circuit output load capacitance C_L , and P_{sc} and P_l reflect the power dissipated due to short-circuit and leakage currents respectively (I_{sc} and I_l). Equation (1) could be expanded in order to reveal the basic circuit parameters contributing to each of the above power components as follows:

$$P_{ave} = k \cdot C \cdot V_{dd}^2 \cdot f + I_{sc} \cdot V_{dd} + I_l \cdot V_{dd} \quad (2)$$

where V_{dd} is the supply voltage, f is the clock frequency, C is the physical capacitance of the circuit, and k is the *transition activity factor* which is defined as the average number of times the circuit makes a power consuming ($0 \rightarrow 1$) transition in a single clock cycle. The parameters k and C are often lumped together in a single parameter, C termed the *effective capacitance* of a design.

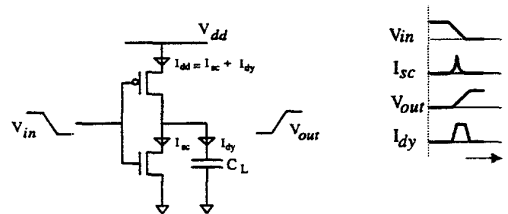


Figure 1 : Typical CMOS circuit power consumption parameter profiles

¹The authors are with the University of Wales, Cardiff

By employing appropriate design techniques both P_{sc} and P_l could be reduced to a negligible level leaving P_s as the dominant factor of power consumption. For this reason the rest of this paper will consider ways of reducing the effect of the constituent parameters of P_s (see equation 2) which can be exploited by the DSP engineer. This paper does not consider issues at the digital circuit level, such as the choice of static/dynamic logic or symmetric/non-symmetric organisation of gates to overcome hazards, since these are covered in greater depth in papers which specifically target logic design (e.g., [1]).

3. Transformation Techniques for Reduced Operation Voltage

It can be seen from equation (2), that the power consumed is a quadratic function of the operating voltage. This dependence on supply voltage has been verified for a number of CMOS logic circuits [2]. Hence reducing the operation voltage for such circuits could significantly reduce the consumed switching power. However, as verified in [2], reducing the operating voltage will increase the delay of the different logic and memory elements in the circuit. The above implies that operation under significantly reduced voltages is associated with an increase in delay and therefore reduced throughput. This reduction in throughput could be compensated through the application of a number of high level transformation techniques [3] which could be applied successively or individually to the DSP system under consideration for low power design [4]. The transformations could be applied to a DSP system at a hardware-level and/or a higher algorithmic-level, represented using a Data-Flow Graph (DFG) [5]. Individual transformation techniques have specific advantages. However they may have side-effect disadvantages, such as an increase in area, which could be overcome by applying another subsequent transformation technique. The rest of this section will list the main transformation techniques with a brief description accompanying each technique [3,4]

Pipelining [5]: is a powerful transformation which can improve the performance of both general purpose and special purpose DSP systems. It involves the insertion of delay elements at specific points of a DFG [3] of an algorithm's structure. The aim being to increase the amount of concurrency. The application of pipelining to synchronous hardware architectures can allow operation with a faster system clock. However, pipelining increases both system latency and the number of delay elements in a system.

Parallel Processing [5]: is similar to pipelining in that it exploits parallelism in a system, however, here this is achieved by duplicating hardware sections in order to perform a number of similar tasks concurrently. The use of both pipelining and parallel processing in maintaining throughput under reduced supply voltage, hence saving power, is demonstrated in [1] and [3].

Retiming [6]: is the process of moving delays around a DFG of an algorithm such that the overall computation is unaltered. It aims to move a computation in an attempt to reduce the *critical path* time, the path with the longest computation time without delays [7]. Optimal retiming of a DFG produces the minimum critical path time.

Unfolding/Folding [7]: is used to unravel the hidden concurrency in a DFG. The iteration period of an N unfolded DFG is $1/N$ times the critical path length of the unfolded DFG. By exploiting inter-iteration concurrency, unfolding can lead to a lower iteration period for a programmable multi-processor implementation. Unfolding could also be used to improve processor utilisation in time-constrained synthesis of DSP systems [3]. Folding is the reverse of unfolding, but is more complex than unfolding since different transformations can result in different folded DFGs.

Other transformation techniques include *Loop-Unrolling*, *look-ahead*, *Distributivity*, and *Constant Propagation*. All of the above techniques are described in a greater detail in [4] and [3]. In the case of non-recursive structures, such as FIR filters, one or more of the above transformations could be used for critical path time reduction. The order in which the transformations are applied is significant in achieving a reduced critical path

time. However, in the case of recursive structures, such as IIR filters, either of *Loop-Unrolling/Look-ahead* could be used in order to apply the rest for critical path time reduction (as in the non-recursive case above). The use of multiple transformations is demonstrated for a simple IIR filter in Figure 2.

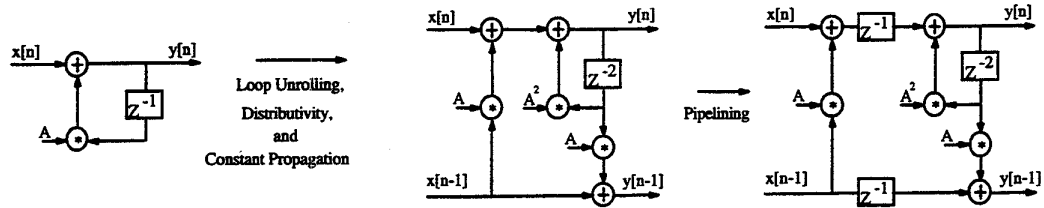


Figure 2: The Application of Multiple transformations.

4. Techniques for Reducing the Effective Capacitance

As mentioned in section 2, the parameters contributing to the effective capacitance of a DSP system are the activity factor k and the physical capacitance C of the system. This section considers a number of techniques which aid the DSP system designer at the architectural levels and the circuit levels of the design.

4.1. Reduced architectural complexity

Such techniques aim to reduce the number of complex area-consuming elements in a DSP system hence reducing its overall capacitance. For example, the use of *primitive operators* [8] has proved effective in providing a significant reduction in computational complexity for digital filters. The above paper also provides references to other published work on reduced complexity techniques.

Transformation techniques such as *Linear Transformation* [9], can be used for reducing the number of multipliers required for FIR filter design. This usually operates on the rows and columns of matrices in a state-space representation of the filter in order to reduce the numbers of additions and multiplications.

4.2. Reduced switching activity

This section considers ways of reducing the switching activity at higher level in the design hierarchy. The choice of data representation for the DSP system is one way of influencing the switching activity. For example, in two's complement representation the signal transition from positive to negative occurs frequently. This transition toggles all the bits and hence significantly contributes to the overall switching activity of the system. This is less so for the sign magnitude representation [1]. The choice of coding, wherever required in a DSP system, is important. The work in [10] demonstrates that gray-coded instruction addressing results in a considerable reduction in switching activity.

Another way of influencing the switching activity is by the choice of an appropriate ordering of the operations in a design. The work in [1] demonstrates that the switching activity, during an operation of multiplication by a constant which is common in DSP, can be reduced by 30% by appropriate reordering. A number of transformations exist [11] that can effect the power consumption by reducing the average transition activity.

Switching activity could also be influenced at a much higher level in the design hierarchy. An example is the choice of point-to-point data buses in favour of multiplexing [12].

4.3. Dedicated DSP processor-based design

Programmability in general purpose DSP processors can be the key to the cost effectiveness but comes at a high energy cost relative to dedicated DSP chips. However, power saving techniques are emerging for both the DSP processor designer and its programmer. For example work at AT&T laboratories [13] has shown that CPU savings of up to 40% could be achieved by reordering the instruction sequence of the Intel 486DX2 processor. It was also shown that memory access instructions consume much more power than those for register access. Hence reducing the number of memory operations can lead to power savings.

For the DSP processor designer, instruction set optimisation is one way of saving power. An example is to provide special data path for often executed instructions in order to reduce the capacitance switched for each execution of the instruction [14]. This approach has been utilised for modem and voice coder processors [15].

5. Power Estimation Techniques

There is a need for accurate power estimates at all levels of the design to provide the DSP designer with accurate accounts of area, speed, and power.

A number of researchers have already attempted to model power consumption for digital VLSI chips. For example, [16] describes a method of power modelling of CMOS VLSI chips in which the power consumption estimate is based on gate count, memory size, logic, and layout styles. The estimation process is fast but less accurate than gate-level simulation.

The work in [17] describes a method of estimating power consumption for VLSI array processors using constant technology dependant values for the different array elements (multipliers, static memory, and I/O). Finally, the authors in [14] suggest guidelines for power estimation, at a number of levels in the design hierarchy, by analysing the main factors contributing to power loss (equation 2). The paper also provides useful references for a number of other publications on gate-level power estimation.

6. Future Design Directions

In this paper we have highlighted the main work which has been carried out to date in low-power DSP system design. However interest in this area is recent and significant future developments are both needed and can be expected. Some directions that these developments could take are as follows :

1. The full exploration of the transformation techniques, introduced in section 3, to realistic DSP applications.
2. The optimal use of the various low-power design techniques is a problem that is combinatorial in nature and thus enables the use of AI-based techniques, such as Genetic Algorithms [18], and heuristics.
3. The techniques of reordering general purpose CPU instructions can be expanded to dedicated DSP processors.
4. Full investigation of coding and signal-representation techniques wherever applicable in the DSP system.
5. Extending the number of power estimation models, such as [16], to cover the full range of DSP functional operators.

References

1. Chandrakasan A.P., Brodersen R.W.: "Minimizing Power Consumption in Digital CMOS Circuits", Proc. IEEE, Vol 83, No 4, pp 498-523, Apr 1995.
2. Chandrakasan A.P., Sheng S., Brodersen R.W.: "Low-Power CMOS Digital Design", IEEE J. Solid-State Circuits, Vol 27, No 4, pp 473-484, Apr 1992.

3. Parhi K.K.: "High-Level Algorithm and Architecture Transformations for DSP Synthesis", J. VLSI Signal Processing, 9, pp 121-143, 1995.
4. Chandrakasan A.P., Potkonjak M., Mehra R., Rabey J., Brodersen R.W.: "Optimizing Power Using Transformations", IEEE Trans. Computer-Aided Design, Vol 14, No 1, pp 12-31, Jan 1995
5. Hwang K., Briggs F.F.: "Computer Architecture and Parallel Processing", McGraw-Hill, 1985.
6. Leiserson C.E., Rose F., Saxe J.: "Optimizing Synchronous Circuitry by Retiming", Proc. of the Third Caltech Conference on VLSI, pp 87-116, Pasadena, CA, March 1983.
7. Lucke L.E., Parhi K.K.: "Data-Flow Transformations for Critical Path Time Reduction in High-Level DSP Synthesis", IEEE Trans. Computer-Aided Design, Vol 12, No 7, pp 1063-1068, July 1993.
8. Bull D.R., Horrocks D.H.: "Primitive Operator Digital Filters", IEE Proceedings-G, Vol 138, No 3, pp 401-412, June 1991.
9. Wintermantel M., Lueder E.: "Increasing the speed and saving multipliers in block parallel digital filters by a linear transformation", Proc. IEEE ISCAS, pp 81-84, London, 1994.
10. Wuytack S., Cathoor F., Franssen F., Nachtergaele L., DeMan H.: "Global communication and memory optimizing transformations for low power systems", Int. Workshop Power Design, Napa Valley, CA, Apr 1994.
11. Brodersen R., Chandrakasan A., Sheng S.: "Low-power Signal Processing Systems", VLSI Signal Processing, V (Yao K., Jain R., Przytula W., Rabaey J., eds.), pp 3-13, IEEE 1992.
12. Blair G.M.: "Designing Low-Power Digital CMOS", Electronics & Communication Eng. Journal, pp 229-236, Oct 1994.
13. Bursky D.: "Power-reduction schemes promise cool digital ICs", Electronic Design Journal, pp 51-65, Jan 1995.
14. Singh D., Rabey J.M., Pedram M., Cathoor F., Rajgopal S., Sehgal N., Mozden T.J.: "Power Conscious CAD Tools and Methodologies: A perspective", Proc. IEEE, Vol 83, No 4, pp 570-593, Apr 1995
15. Be'ery Y., Berger S., Ovadia B.: "An application specific DSP for portable applications", Proc. VLSI Signal Proc. Workshop, Veldhoven, The Netherlands, pp 48-56, 1993.
16. Liu D., Svensson C.: "Power Consumption Estimation in CMOS VLSI Chips", IEEE J. Solid-State Circuits, Vol 29, No 6, pp 663-670, June 1994.
17. Chau P.M., Powel S.C.: "Power Dissipation of VLSI Array Processing Systems", J. VLSI Signal Processing, Vol 4, pp 199-212, 1992
18. Golberg, D.E.: "Genetic algorithms in search, optimisation and machine learning," Addison Wesley, Reading, 1989.