LOW-POWER SYNTHESIS FLOW FOR REGULAR PROCESSOR DESIGN

R Woods[+], G Lightbody[*], A Cassidy[++], G Keane[**] and J Spanier[+++]

Abstract

The paper presents the development of a low-power synthesis flow for the development of dedicated silicon circuits for data-dominated applications such as DSP systems. The work was carried out as part of a European ESPRIT low power action and a collaborative "low-power" project involving the universities of Liverpool, Manchester and Sheffield. The design flow is briefly described and some results are presented for multiplier implementations and their use in the development of a Discrete Cosine Transform (DCT) circuit

1. Introduction

In data-dominated applications, the performance capabilities of customized solutions are attractive particularly in applications such as video compression where functionality is fixed or has been standardized. For high volumes, customized VLSI solutions offer the superior system performance in terms of both area and speed. For example, a processing performance of 100 $GOPs/cm^2Ws$ is achievable for computationally complex DSP algorithms in 0.35μ standard cell CMOS technology. The performance gain comes from the process of developing an architecture that allows the algorithm to be efficiently mapped to hardware. This results in a very efficient solution as it is often possible to exploit characteristics of the algorithm in such a way to allow a highly optimised implementation to be realized. There are a number of different design flows one of which is described in this paper. The resulting designs are characterized by high area utilization, high levels of locality (preserving power) and efficient memory utilization.

2. Low-power Regular Processors

Broadly speaking, DSP processor development can be classified as two design styles.

- The first approach involves the development of an instruction set processor where suitable processing units are synthesized to achieve the required performance. Once a sensible architecture has been determined, the majority of the design activity is involved in scheduling, memory optimisation and code transformation. As the underlying architecture is well understood and highly predictable, accurate power performance is achievable.

- The second approach attempts to synthesize an application specific architecture which closely matches the application functionality that it implements. This allows the designer to apply a number of performance efficient optimisations that would not be available in the previous design flow. In addition, the designer has access to hardware transformations and different number systems which can offer advantages in some applications.

The material described in this paper concentrates on the synthesis of solutions which fall into the second classification and closely follows the work described in the book produced from this work[1]. This latter approach allows a more power-efficient solution as the architecture can be customized to

+Electrical and Electronic Engineering, Queen's University of Belfast, Ashby Building, Stranmillis Road, Belfast
* Integrated Silicon Systems, Malone Road, Belfast
++ Cypress Irish Design Centre, Mahon Technology Park, Cork, Ireland
** Edinburgh House, Cambridge, England
+++ PCM-Sierra, Vancouver, Canada

meet an area, speed and power budget. On the other hand, this flexibility presents greater challenges in delivering low power solutions as there are a greater number of degrees of freedom.

3. Low-Power design approach

Generally speaking, power reduction techniques either act to minimize the switched capacitance (C*F) or trade off performance with reduced supply voltage[2]. In voltage minimization techniques, transformations are used to speed up the system's throughput beyond that which is necessary. The voltage is then reduced, slowing up performance until the required throughput rate is met but at a lower power consumption budget. However, in the semi-custom approach, the voltage of the technology is pre-determined by the silicon foundry and cannot be altered by the user to reduce power. Typically, the voltage will have already been developed for low power operation and the user will access the design though library phantoms which the user cannot modify. Therefore, the "reduced voltage" techniques cannot be employed.

Low power design therefore must be targeted at reducing the switched capacitance. This is the summation of the products of the toggling occurring on each node in the circuit by the capacitance of that node. It is important to consider switched capacitance as opposed to toggling or circuit capacitance independently. For example, a circuit may have a large capacitative net which has a low switching activity and which will not contribute greatly to power consumption[3]. Conversely, low capacitance nets may have a lot of switching activity[3]. For these reasons, minimization of switched capacitance in order to reduce power consumption is adopted for the regular processor meta flow.

4. Regular Processor design flow for signal processing modules

The starting point of the regular processor "meta" design flow is an algorithmic description of the functionality that has to be mapped to the processor. This algorithmic description is then refined using the design flow illustrated in figure 1, resulting finally in a circuit level solution. The various stages are described here in more detail.

algorithmic: This involves the generation of a correct algorithmic description for the specific custom processor e.g. a full MPEG encoder. At this stage, the main focus is to develop and validate the algorithmic operation. High level modelling tools such as Cadence SPW^{TM} and Matlab^{TM} can be used to assist in this purpose.

data type refinement: At the algorithm stage, the development will have been carried out using abstract data types, such as floating point arithmetic. In DSP design, a fixed point or a limited range floating point solution is preferable as it usually results in a more hardware efficient solution. In many cases, the choice of fixed or floating point arithmetic is not straightforward and can have a dramatic effect on the system performance. This will include the consideration of finite wordlength issues such as truncation and saturation.

task level: The algorithm is broken down into a number of separate tasks. For an MPEG encoder, the various tasks will include the DCT and motion estimation. Typically, the final solution or the computationally complex parts of the system, will be implemented as a custom processor. A key issue is to ensure that the minimal number of processors are used in the task allocation process. In a parameterizable IP cores, the task allocation will have been pre-determined in the design process for the core.

parallel-array/data level issues: At this stage, the data parallel functions will be transformed into an underlying structure of processors. For DSP algorithms, this will involve the development of a VLSI architecture from a Dependence Graph (DG) representation. A considerable body of work exists into the transformation of commonly used functions such as the DCT and FFT into efficient structures where the number of multiplications and additions are reduced. A key issue will be an efficient use of hardware for the given application requirements.

operation/instruction level issues: A number of choices are available at this stage including the structure of the processor blocks, arithmetic representation and data control. For example, the multipliers and adders can be implemented as separate processors or as a multiply-accumulator structure. The choice of data type is highly important as the regular-processor designer has access to a wide choice of options e.g. fixed point and different types of arithmetic e.g. redundant arithmetic and residue arithmetic representations. The choice of the different arithmetic representations other than two's complement, will be dictated by specific performance advantages gained. For example, redundant arithmetic can be used in the development of recursive structures such as recursive digital filters and in arithmetic functions that are naturally most significant bit first, namely division and square root.

circuit level: The circuit level implementation involves preserving regularity in the generation of the circuit level description and use of pipelining registers in the datapath.

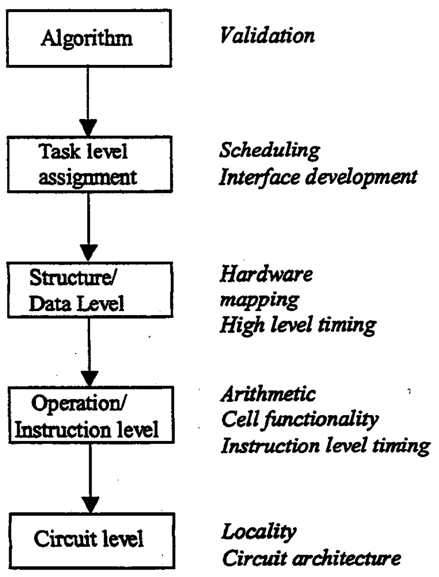

Figure 1. Regular processor design flow for signal processing modules

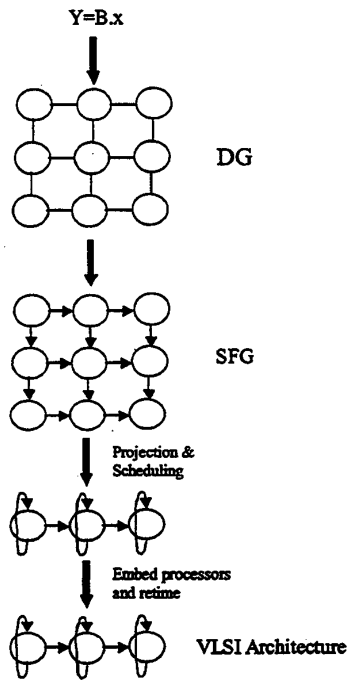

Figure 2: Development of VLSI architecture

5. Algorithmic issues

There have been a wide range of techniques for developing a mathematical representation from an algorithmic description. The process in figure 2 shows the general design flow for development of highly regular, VLSI solutions for DSP algorithms. The initial specification is given as a mathematical relationship which is translated into a DG representation. A DG representation shows the data dependencies between different dataflows. At this stage, it is possible to develop a Signal Flow Graph (SFG) representation and a number of projections can be applied to develop SFG alternatives that match the algorithmic requirements. The bit-level processors are then embedded into the SFG. These may have detailed timing requirements which may mean retiming the SFG to ensure correct operation[4]. From a low power perspective, the final VLSI solution should ideally meet some or all of the following criteria:

- Circuit representation with localized connections.
- Solution with efficient wordlength representations.
- Optimized dataflow to minimize transitions on the internal circuit buses.

6. Operation/instruction level issues

At this stage in the design flow, the development of a candidate VLSI architecture will have been chosen. A simple approach is to instantiate a processor for each of the processing elements. There are a variety of options open to the designer ranging from the arithmetic used for the instruction level processors to the choice of an operation level processor. Issues addressed at this level include:

- **Choice of arithmetic style**: This includes the use of alternative number systems such as redundant and residue arithmetic representations. Residue arithmetic works well in applications where word growth is limited. Redundant arithmetic has particular low power advantages in that it breaks carry propagation chains in addition and therefore multiplication. This reduces the circuit switching.

- **Detailed architecture of the instruction level processor**: A hierarchical approach based on simple, processors can be used. Alternatively, it is possible to develop the circuit architecture using, for example, the Co-ordinate Rotation Digital Computer (CORDIC) transformation. The choice of bit level circuit architecture can also impact power performance.

7. Power Estimation of Processor Block

A number of studies have been carried out into adders structures[5,6] and multiplier structures[3]. Some of these results are shown in table 1. The multiplier structures presented all have different capabilities e.g. the Wallace-Tree structure can operate at much higher frequencies than the array structures. This factor was taken into consideration. The table clearly shows that preserving hierarchy at the circuit level allows a power saving. This is particularly observable at higher wordlengths where the increased wordlengths result in many more longer nets which make have a greater impact on power consumption.

8. Application to 1D DCT example

The DCT is a key DSP function that has been used extensively for image processing and some speech coding applications. The 2D version of the DCT is a separable function that can be split into two single DCT functions. This approach is commonly termed row-column decomposition. One approach to the implementation of the 1D DCT is to transform the expression into a transfer function resembling a second order IIR filter representation[8]. This gives a circuit comprising two multipliers and three adders which can be used to develop DCT implementations for a wide range of specifications. An analysis[9] has been carried out on a circuit with a 8 bit input, 11 bit output and a 12 bit internal datapath and that operates at 100 MHz. The two multipliers have coefficient wordlengths of 12 bits and 10 bits.

Name	8-Bit(mW)	16-bit(mW)	24-bit(mW)
Carry-Save array multiplier[6] synthesized as an array	5.99	23.19	56.92
Carry-Save array multiplier[6] with no layout hierarchy	3.23	25.94	67.96
Two's Complement Booth Encoded Wallace Tree	4.00	32.24	86.12

Table 1. Power consumption of multipliers processing random data at 20MHz

A number of approaches were investigated. The hierarchical approach involved the derivation of a solution where the individual processing blocks multipliers and adders are synthesized and implemented individually. An approach using carry-save multipliers was investigated where the circuit layout is generated without imposing any hierarchy i.e. flat. A hierarchical DCT approach using carry-save multipliers was also investigated. These results are shown in table 2.

	Power (mW)
Original DCT system (Wallace Tree)	64.7
Hierarchical DCT system (Wallace Tree)	58.9
Flat synthesis DCT system (Carry-save)	49.0
Hierarchical DCT system (Carry-save)	44.0

Table 2. Power consumption of multipliers processing random data at 20MHz

9. Conclusions

In this paper, a brief description of a low-power design flow has been presented. It is shown how a low-power design approach is achievable at an algorithm/architectural level without resorting to circuit or technology level techniques. Future works aims to target a design at each stage of the design flow to demonstrate more fully the design gains possible as the DCT example only demonstrates the impact at the operational/instructional and circuit levels.

10. References

1. Eds F Catthoor (Ed.), "Unified low-power design flow for data dominated multi-media and telecom applications", Kluwer Academic Publishers, Boston, USA, 2000.
2. A. Chandrakasan, S. Sheng, and R. Brodersen, R., "Low Power CMOS Digital Design", IEEE Journal of Solid-State Circuits, Vol. 27, No. 4, 1992, pp473-484.
3. G. Keane, J. R. Spanier, and R. Woods, "Low-Power design of Signal Processing systems using Characterization of silicon IP cores", 33rd Asilomar Conference on Signals, Systems and Computers, Asilomar, USA, Oct. 1999, invited paper, IEEE Computer Society, pp767-771.
4. D Trainor, R F Woods and J V McCanny, "Architectural Synthesis of Digital Signal Processing Algorithms using "IRIS"", Journal of VLSI Signal Processing, Vol. 16, No 1, May 1997, pp41-56.
5. A. M. Shams and M. Bayoumi, "A Structured approach for Designing Low Power Adders", 31st Asilomar Conference on Signals, Systems and Computers, IEEE Computer Society, Vol. 1., 1997, pp 757-761.
6. C. Nagendra, M-J. Irwin, and R. M. Owens, R.~M. "Area-Time-Power Tradeoffs in Parallel Adders", IEEE Transactions on Circuits and Systems, Vol. 43, No. 10, 1996, pp689-702.
7. J. Deverell, "Pipelined Iterative Arithmetic arrays" IEEE Trans. On Computers, C-24}, 1975, pp 317-322.
8. J. Hunter, "Rapid Design of Discrete Cosine Transform Cores for Multimedia VLSI Systems", PhD thesis, The Queen's University of Belfast 1999.
9. Spanier, J., Keane, G. and Woods R, "Low Power Implementation of a Discrete Cosine Transform IP core" *Proceedings of the Design, Automation and Test in Europe Conference (DATE-2000)*, Paris, France, 27-30 March 2000, Date User Forum, Session 8C, pp. 267-271.