Word-Length Oriented Multiobjective Optimization of Area and Power Consumption in DSP Algorithm Implementation

A. Ahmadi, M. Zwoliński

Abstract – The word-length of Functional Units (FU) has a great impact on design costs. This paper addresses the problem of choosing different word-lengths for each FU while considering circuit area and power consumption. A high-level synthesis tool is used to minimize the circuit area and power consumption by selecting an optimal word-length for each FU in the system. Our results demonstrate that by customizing word lengths to non-standard sizes, savings can be made in the overall area and power without losing accuracy.

I. INTRODUCTION

One of the problems in implementing signal processing algorithms on digital hardware is choosing an appropriate word length for arithmetic units. Traditionally this problem is solved by making a worst-case assumption and choosing a single word length for all arithmetic units. Using a word-length less than this worst-case assumption at different points in the algorithm would, however, save both area and power.

Several pieces of work have focused on finding an optimal word-length for the algorithm as the first step and then designing or optimizing the system within that constraint, [2]. The word-length is not considered in the subsequent optimization process. In other studies in which word-length has been considered during optimization [9], signals have been categorized into a few groups to constrain the word-length in all functional blocks, or only one objective has been considered in addition to digital noise [1, 8].

High Level Synthesis (HLS) has been considered to be a key factor in reducing the distance between the initial specification and target design [3]. Because of the variety of possible applications, domain-specific HLS tools are needed to achieve an optimal solution.

In this work, we present a multi-objective optimization method to optimize circuit area and power consumption by choosing optimum word-lengths for each FU. Cost functions for area, power consumption and digital noise are discussed in section 2, sections 3 gives a short description of the implemented design tools and section 4 is devoted to the GA method which has been applied. Results are explained in section 5.

A. Ahmadi and M. Zwoliński are with the Electronic System Design Group, School of Electronics and Computer Science, University of Southampton Southampton, UK, E-mail {aa03r, mz}@ecs.soton.ac.uk

II. COST FUNCTION

From a high level synthesis point of view, both the total area and power consumption of a system can be divided into three parts: data paths; controllers; and interconnections. Having focused on word length optimization, area and power costs should be considered as functions of the functional unit word length. Depending on the implementation methodology, word lengths have different impacts on each part (datapath, controller and interconnections) of the metrics. In our method, changing the word length does not change the controller area, so that is considered as a constant value in the cost function. In addition, since this methodology is bus oriented, interconnection costs are only marginally affected by word length compared to the changes in the datapath. Accordingly our cost models assume the datapath costs are variable and others are constant, equation (1).

$$F_{Total}(\vec{W}) = F_{Controller} + F_{Intercon} + F_{Datapath}(\vec{W}) \quad (1)$$

F is the cost function and $\vec{W}$ are the word lengths for the functional units. In the following section, we present a brief description of the cost models for circuit area, power consumption and digital noise.

A. Area Cost Function

Since the area of the controller (A_C) does not change with word length and the interconnection area (A_B) only slightly depends on it, the total area of the datapath is evaluated by adding up the sub-block and FUs areas ($A_{FU}(\vec{W})$). Thus as an approximation, the area of building blocks such as sequential multipliers, adders, registers, buffers and switches can be assumed to have a proportional relationship to word length while the area of a combinational multiplier can be modeled by a second order relationship with its word length. Design implementation results confirm this assumption as depicted in figure (1). Equation (2) gives the area cost function for a system.

$$F_A(\vec{W}) = A_C + A_{FU}(\vec{W}) + A_B \quad (2)$$

B. Power consumption Cost Function

Knowing that the changing word length of the FUs does not affect the controller activity and structure, the

power consumption of controllers (P_C) is a fixed term in the estimated power consumption. In addition, because a bus-oriented approach is used in this study, interconnection power consumption (P_B) only depends on the maximum word length in the shared bus; therefore, ignoring the P_B dependency on W is acceptable at this level of abstraction. Equation (3) shows the general model of power consumption with these approximations.

$$F_P(\vec{W}) \approx P_{FU}(\vec{W}) + P_B + P_C \quad (3)$$

Fig. 1 Dependency of area on word length for basic cells (registers, adder and multiplier).

A set of designs was used to evaluate the functional unit dependency on word length and the results are presented in Figure (2). In this figure, the average power consumption for basic cells, with random input data is shown with respect to word length. In these simulations, the Nominal Low Leakage ST 0.12 μ m technology file is used. From this, we can see that power consumption can be modelled as a linear function of the word length.

On the other hand, power consumption is a combination of static and dynamic parts; accordingly, in each FU it is a sum of static and dynamic parts as in Equation (4).

$$P_k = P_{k,Dynamic} + P_{k,Static} \quad (4)$$

Here P_k is the power consumption of the k^{th} FU.

In general, dynamic and static power consumption are data dependent [5] but in this study, to estimate power consumption in the optimization procedure, static power consumption is considered proportional to the total power, Equation (5).

$$P_{k,Static} = \lambda_k \cdot P_k \quad (5)$$

λ_k is the leakage power factor. Simulations verify this assumption for basic blocks for different word lengths.

Another assumption used to reduce the evaluation complexity is a time slot approximation [10]. In this approximation the total power consumption of a functional unit is calculated in two parts: *activation* time slots and *standby* time slots. During functional operation, power consumption is the sum of dynamic and static power whereas in standby, only the leakage power is taken in account. Based on this approximation, the total power consumption for each functional unit is given in Equation (6).

$$F_P(\vec{W}) = \frac{1}{T} \cdot \sum_{k=1}^{N_F} (t_k P_k + (T - t_k) \cdot P_k \cdot \lambda_k) \quad (6)$$

F_P is the average power consumption of the system, P_k is the average power consumption of the k^{th} functional unit, t_k is its activation time and T is the total system operation time.

Fig. 2 Dependency of Power Consumption on word length for basic cells (Register, Adder and Multiplier)

C. Digital Noise Cost Function

In practice, digital signal processing systems can only offer a finite number of binary digits to represent the signals to be processed. Fitting real values in these limited containers causes effects which can be categorized in several different ways. From a mathematical point of view, using a limited number of bits to represent a real number always means adding or removing indeterminate information at the input, which is usually considered as an error or noise. To model this problem in our tool and to evaluate its impact; there are two problems to consider: first is a noise model for computational errors and second is a model of noise propagation. A number of models of digital noise have been proposed in [7].

To provide a noise propagation model, it must be recalled that many DSP algorithms can be considered as Linear Time Invariant (LTI) systems. This assumption allows us to use superposition of independent noise sources to compute the noise effect on the system output, [2], [6]. The effects of noise sources on the output can be approximated using Equation (7).

$$E_{Output} = \sum_{k=1}^M \sigma_k^2 \cdot (L_2\{H_k(z)\})^2, \quad (7)$$

$H_k(z)$ is the Z-transform of the transfer function ($h[n]$) from the k^{th} noise source to the output and $L_2\{\}$ is the L-Norm [6], given by Equation (8).

$$L_m\{H(z)\} = \left[\sum_{n=0}^{\infty} |Z^{-1}\{H(z)\}[n]|^m \right]^{\frac{1}{m}} \quad (8)$$

σ_k can be found from Equation (9), [2],

$$\sigma_k^2 = \frac{1}{12} 2^{2p} (2^{-2n_2} - 2^{-2n_1}). \quad (9)$$

n_1 is the present arithmetic unit word length and n_2 is the next arithmetic unit word length and p is the position of the decimal point.

III. IMPLEMENTATION

The system design methodology starts from a hierarchical specification of the target system and is based on three parts: the functional unit data base; the target architecture; and the synthesizer-optimizer. The target architecture is built on a partitioned shared bus with distributed controller which makes the target design very flexible to match a variety of DSP algorithms as well as being very modular and manageable for the synthesizer and optimizer [1]. From a synthesis point of view, on the other hand, this target architecture is a restriction in that it forces the synthesizer to map every design to a pre-defined structure which dominates the feasible solution space in favour of the optimizer.

The functional unit database is a library of functions and sub-systems. There are four kinds of sub-system in our method: algorithm executors, interfaces, memories and controllers, which each might contain further functional units and/or sub-systems. In addition to implementation information, this database provides the required information for the design optimizer cost functions including: area, accuracy, delay and power consumption.

The synthesizer's input is a high level specification of the algorithm in the form of difference equations. Basically there is a pre-defined hierarchical architecture to which the target system must be mapped. The starting specification of the system and the final implementation are both represented by a digraph. A set of library files is used to produce Intermediate Code (ICD) files which are a more compact form of the initial specification of the target system. The library files contain the basic blocks of the system and their cost relationships (noise, area, power, and delay) as functions of word length. These cost parameters can be used in a cost evaluation program after scheduling, allocation and binding to optimize the design.

IV. OPTIMIZATION

A GA is utilized in this study for design optimization. The genetic operators are extracted from standard GA procedure which includes selection by roulette wheel, crossovers, mutation [4] and brand new randomly produced genes. Rates for crossovers, mutation and imported genes are chosen as shown in Table (1). In Table (1) M is the number of FUs, $p(x)$ is a randomly generated value and K_1, K_2, K_3, K_4, K_5 are constant values dependent on M and the number of the iterations in the algorithm.

According to the target architecture, one word length (w) has to be assigned to each functional unit. Therefore, we define a vector of word lengths for the FUs in data paths as in Equation (10) and this vector is used as the gene in the GA optimization algorithm.

TABLE I
GENETIC ALGORITHM PARAMETERS

Parameter	Value
Number of Individuals in the Population	$K_1 \cdot M$
Number of crossovers	$K_2 \cdot M \cdot P_2(x)$
Number of brand new Individuals	$K_3 \cdot M \cdot P_3(x)$
Number of Increment/decrement Mutations	$K_4 \cdot M \cdot P_4(x)$
Number of Generations (Iterations)	$K_5 \cdot M$

$$\vec{W} = [w_1 \quad w_2 \quad w_3 \quad \dots \quad w_M] \quad (10)$$

An optimization problem must then be solved, with multiple objectives and constraints taken into consideration. A standard technique for Multi-objective Optimization is to minimize a positively weighted convex sum of the objectives, as shown in Equation (11).

$$F_A(\vec{W}) = \left(K_A \frac{F_A(\vec{W}) - F_{A,MIN}}{F_{A,MAX} - F_{A,MIN}} + K_P \frac{F_P(\vec{W}) - F_{P,MIN}}{F_{P,MAX} - F_{P,MIN}} + K_N \frac{F_N(\vec{W}) - F_{N,MIN}}{F_{N,MAX} - F_{N,MIN}} \right) \times \frac{1}{K_A + K_P + K_N} \quad (11)$$

F_A , F_P and F_N are cost functions for area, power consumption and digital noise respectively, as given in the previous sections; and MIN and MAX indicate minimum and maximum values of the functions. K_A , K_P and K_N are constants as weighting factors for costs.

V. RESULTS

Four case studies were implemented in ST 1.2 μ m technology. Design I is an order-10 difference equation, Design II is an order-18 difference equation, Design III is a Filter (FIR-25) and Design IV is a DCT 4x4.

In most practical implementations, there are known constraints which must be satisfied and therefore, other costs must be optimized with respect to them. Comparison of the results in Figures (1) and (2) and equation (11), suggests that by freezing one of the costs and taking it as a design constraint during optimization; it is possible to achieve the same required objective with minimum costs for the other two. To illustrate this, a set of constrained optimizations was performed with constrained accuracy. Table (2) provides the results of such design optimizations.

Several examples are given in Table (2) for each design. At first, all the FUs in the design were assigned to a fixed word-length. Four basic cases ($W=8, 16, 24$ and 32) were implemented and their design costs (Area, Power Consumption and Digital Noise) were calculated as the reference values. In the second step, three optimization approaches were applied for each design in each case. Optimizations were based on freezing one of the costs and optimizing two others. Clearly, in all cases design costs are reduced by our methodology however this improvement is dependent on design and accuracy constraints.

VI. CONCLUSIONS

This study presents a methodology for implementing DSP algorithms which uses models of power consumption,

circuit area and output noise and their relationship to word-length. Investigation of basic designs shows a considerable improvement in costs when optimizations are employed.

TABLE II
COST COMPARISONS BETWEEN UNIFIED WORD-LENGTH AND OPTIMIZED MULTIPLE WORD-LENGTH DESIGN METHODS. C=CONSTRAINED

Design	Costs	Case 1				Case 2			
		Unified W W=8	Optimized Multiple W			Unified W W=16	Optimized Multiple W		
Design I	Area	24392	C	23141	21343	48784	C	46421	45677
	Power	5.97918	5.97918	C	5.23178	11.9584	11.9584	C	11.1868
	Noise	7.39e-2	6.93e-2	4.33e-2	C	2.89e-4	9.23e-5	1.47e-4	C
Design II	Area	46928	C	41646	41062	93856	C	86211	84363
	Power	9.76618	9.76618	C	8.5454	19.5324	19.5324	C	17.9538
	Noise	8.80e-2	6.64e-2	7.17e-2	C	3.44e-4	2.19e-4	3.06e-4	C
Design III	Area	44512	C	34643	38948	89024	C	79711	83460
	Power	7.75508	7.75508	C	6.7857	15.5102	15.5102	C	14.5408
	Noise	7.57e-2	2.39e-2	6.93e-2	C	2.96e-4	9.35e-5	2.27e-4	C
Design IV	Area	106736	C	83384	93394	213472	C	190676	199736
	Power	17.0908	17.0908	C	14.9545	34.1817	34.1817	C	32.0082
	Noise	6.41e-1	5.11e-1	2.39e-1	C	2.51e-3	2.92e-4	8.66e-4	C
Design	Costs	Case 3				Case 4			
		Unified W W=24	Optimized Multiple W			Unified W W=32	Optimized Multiple W		
Design I	Area	73176	C	69979	69397	97568	C	94788	91368
	Power	17.9375	17.9375	C	17.1178	23.9167	23.9167	C	23.0729
	Noise	1.13e-6	1.31e-7	7.30e-7	C	4.40e-9	8.19e-10	3.24e-9	C
Design II	Area	140784	C	136614	131525	187712	C	186044	181846
	Power	29.2985	29.2985	C	28.3093	39.0647	39.0647	C	37.8439
	Noise	1.34e-6	7.55e-7	1.39e-6	C	5.25e-9	3.61e-9	5.91e-9	C
Design III	Area	133536	C	124640	127578	178048	C	169152	175997
	Power	23.2653	23.2653	C	22.2555	31.0203	31.0203	C	30.8386
	Noise	1.16e-6	3.65e-7	7.28e-7	C	4.51e-9	1.05e-9	3.39e-9	C
Design IV	Area	320208	C	296856	269246	426944	C	403731	384697
	Power	51.2725	51.2725	C	44.5662	68.3634	68.3634	C	62.6231
	Noise	9.79e-6	2.70e-6	3.48e-6	C	3.82e-8	1.04e-8	1.43e-8	C

REFERENCES

- [1] A. Ahmadi and M. Zvolinski, "Area Word-Length trade Off in DSP Algorithm Implementation and Optimization," presented at IEE/EURASIP Conference on DSPEnabledRadio, 2005.
- [2] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, *Synthesis and Optimization of DSP Algorithms (Fundamental Theories of Physics S.)*: Kluwer Academic Publishers, 2004.
- [3] G. De Micheli, *Synthesis and Optimization of Digital Circuits*: McGraw-Hill Education, 1994.
- [4] D. A. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* Addison-Wesley Professional 1989.
- [5] E. Macii, M. Pedram, and F. Somenzi, "High-level Power Modeling, Estimation, and Optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, pp. 1061 - 1079, 1998.
- [6] A. V. Oppenheim and C. J. Weinstein, "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform.," presented at IEEE Proceedings, 1972.
- [7] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*: Pearson US Imports & PHIPEs, 1998.
- [8] N. Sulaiman and T. Arslan, "A Multi-objective Genetic Algorithm for On-chip Real-time Optimisation of Word Length and Power Consumption in a Pipelined FFT Processor targeting a MC-CDMA Receiver," presented at 2005 NASA/DoD Conference on Evolvable Hardware, 2005.
- [9] W. Sung and K. Kum, "Simulation-based Word-length Optimization Method for Fixed-point Digital Signal Processing Systems," *IEEE Transactions on Signal Processing* vol. 43, pp. 3087 - 3090, 1995.
- [10] A. C. Williams, A. D. Brown, and M. Zvolinski, "Simultaneous Optimisation of Dynamic Power, Area and Delay in Behavioural Synthesis," *Computers and Digital Techniques, IEE Proceedings*, vol. 147, pp. 383 - 390, 2000.