Custom Reduction of Arithmetic in Linear DSP Transforms

Smarahara Misra James C. Hoe Markus Püschel*

Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA

Introduction

In multiplier-less hardware implementations of DSP transforms, multiplication-by-constants are implemented as a network of (wire-)shifts and additions. The number of additions required can be reduced by approximating the multiplicative constants using lower precision fixed-point representations, but the loss of precision increases the numerical error in the implementation. This trade-off can be leveraged to reduce the hardware area, critical path and power/energy while maintaining the perceptible quality of a signal processing application (e.g., MPEG-4). This paper describes an automatic approach to minimize the number of additions subject to a given quality measure, or, vice-versa, to maximize the quality subject to a given number of available additions. Our automatic approach can handle linear DSP transforms in general and includes optimizations over the space of algorithm design. A Verilog backend generates synthesizable descriptions of the final variable-width fixed-point implementations.

Approach

We consider the following two optimization problems for a given linear DSP transform: (1) Given a quality threshold Q, find the multiplierless implementation with the least arithmetic cost C that satisfies Q; (2) Given an arithmetic cost threshold C, find the multiplierless implementation with the highest quality Q. Our proposed system automatically solves this problem in the following steps. We consider problem (1); problem (2) is analogous.

Given is a formally specified linear DSP transform T (e.g., a DCT of size 8) and the quality threshold Q (e.g., the maximum allowed error of the output).

Step 1: Generating a Fast Algorithm. First, we generate a fast algorithm for T represented as a formula in a mathematical notation using SPIRAL[1]. The formula is built from few constructs and primitives such as the Kronecker product '$\otimes$', permutation matrices, or 2×2 rotations R_α. For example, one out of many possible formulas for the DCT of size 8 looks like

$$\begin{aligned}\mathrm{DCT}_8 \;=\; & [(2,5)(4,7)(6,8),8] \cdot (\mathrm{diag}(1, \tfrac{1}{\sqrt{2}}) \oplus \mathrm{R}_{\frac{3}{8}\pi} \oplus \mathrm{R}_{\frac{15}{16}\pi} \oplus \mathrm{R}_{\frac{21}{16}\pi}) \\ & \cdot[(2,4,7,3,8),8] \cdot ((\mathrm{F}_2 \otimes \mathbf{1}_3) \oplus \mathbf{1}_2) \cdot (\mathbf{1}_4 \oplus \mathrm{R}_{\frac{3}{4}\pi} \oplus \mathbf{1}_2) \cdot [(2,3,4,5,8,6,7),8] \\ & \cdot(\mathbf{1}_2 \otimes ((\mathrm{F}_2 \oplus \mathbf{1}_2) \cdot [(2,3),4] \cdot (\mathbf{1}_2 \otimes \mathrm{F}_2))) \cdot [(1,8,6,2)(3,4,5,7),8].\end{aligned}$$

Step 2: Manipulation for Numerical Stability. In the second step, we formally manipulate the formula to increase its numerical stability, which determines how quick the quality of T degrades when implemented in low precision. In particular, we expand the formula into lifting steps using ideas from[2].

*The work of Markus Püschel was supported by DARPA through research grant DABT63-98-1-0004 administered by the Army Directorate of Contracting and by NSF through award 9988296

[1] J. Moura, J. Johnson, R. Johnson, D. Padua, V. Prasanna, M. Püschel, B. Singer, M. Veloso, and J. Xiong. Generating Platform-Adapted DSP Libraries using SPIRAL. In *Proc. HPEC*, 2001. `http://www.ece.cmu.edu/~spiral`.

[2] J. Liang and T.D. Tran. Fast Multiplierless Approximations of the DCT With the Lifting Scheme. In *IEEE Transactions on Signal Processing*, Vol.49, No.12, Dec 2001, pages 3032-3044.

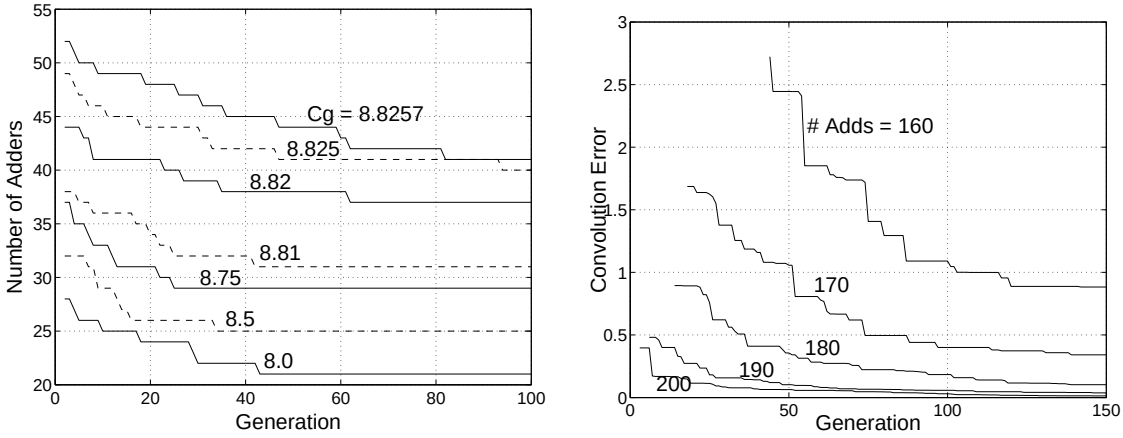

Figure 1: Evolutionary optimization. Left: for DCT_8 minimizing number of additions for various given coding gains (cg); right: for DFT_{16} optimizing convolution error for various given numbers of additions.

Step 3: Constant Reduction and Search. In this step the actual constrained optimization is performed using an automated search. The idea is to replace each occurring constant (multiplication) in the formula by a low-precision version, specified by the number of bits in $\{0, 1, \ldots b_{max}\}$. Doing so for every constant yields an approximation $\tilde{T}$ of the original transform T; $\tilde{T}$ has a lower cost C than the original, i.e., requires less additions. If the formula for T contains n multiplier constants $a_1, \ldots, a_n$, there are $(b_{max} + 1)^n$ many ways of approximation, which determines the search space for our optimization. Since an exhaustive search is infeasible we use evolutionary and greedy search techniques to find the approximation with the lowest cost (least number of additions) that still satisfies the quality threshold Q.

Step 4: Mapping to Verilog. In this final step we map the found (approximated) formula into Verilog.

We note that in the above the approach, the formula, i.e., algorithm chosen for the transform was fixed. The optimization can readily be extended to include the space of different possible formulas into the optimization using SPIRAL's formula generator.

Experimental Results

We show two examples for two different optimization problems for the discrete cosine transform (DCT) and discrete Fourier transform (DFT).

DCT, size 8. We chose as quality measure coding gain (cg) in dB, which for the exact (infinite precision) DCT is about 8.8259. We considered one formula for the DCT generated by SPIRAL (similar to the one above). A 10-bit multiplierless implementation for this formula requires 56 additions. After formula manipulation, we considered 9 constants in the formula for further approximation, which yields a search space of size 9^{10}. Figure 1 (left) shows the results of an evolutionary search for various cg thresholds. The abscissa shows the generations in this search, the ordinate the found solution with the least cost. For example, after 100 generations, for cg = 8.81 a solution with only 31 adders was found. The search took 30 minutes.

DFT, size 16. We chose as quality measure the convolution error (ce), which determines to what extent the DFT's convolution property is violated. The exact DFT has ce = 0. Again we considered one particular formula, whose 10-bit implementation requires 256 adders. Figure 1 (right) shows the results for fixing the number of additions and optimizing the achievable quality. For example, by allowing 170 adders, a solution with ce = 0.341 was found after 150 generations. The search took 2 hours.