High Level Transformations using Taylor Expansion Diagrams

D. Gomez-Prado, Q. Ren, M. Ciesielski
University of Massachusetts, ECE Department
Amherst , MA 01003
dgomezpr@ecs.umass.edu

J. Guillot, E. Boutillon
LESTER, Université de Bretagne Sud
BP 92116, 56321 Lorient, France
jeremie.guillot@univ-ubs.fr

http://tango.ecs.umass.edu/TED/Doc/html

Abstract: *TED Express is a CAD software to generate and optimize Taylor Expansion Diagrams (TED), a novel, canonical, graph-based representation for arithmetic expressions of data flow designs. In addition to equivalence checking of high-level design specifications, TEDs can be used for optimization of mathematical expressions derived from DSP designs and other high-level design descriptions. TED adds behavioral transformation level to architectural and RTL synthesis.*

CONTEXT: All high-level synthesis tools rely on a *fixed* data flow graph (DFG), derived directly from the initial specification, with no capability to transform the design flow into another DFG. In order to explore another solution the user must rewrite the original specification, from which another fixed DFG is derived and subjected to architectural optimization.

There have been several attempts to add behavioral level transformations on top of high level synthesis that would modify the DFG. These techniques rely on the application of basic algebraic properties, (such as associativity, commutativity, distributivity, etc.) but they are applied largely in an *ad-hoc* manner. TED Express solves this problem by offering a *systematic* method to perform and optimize behavioral transformations, based on a canonical TED representation. In particular, TED can optimize an initial algorithmic description of DSP transforms to minimize the amount of hardware resources.

USAGE: The TED package has been implemented in C++ and runs on a Linux platform. TED Express is called with the shell command *./ted*. The list of available commands, shown in Fig. 1, is available by typing *help*. The package takes as input a set of mathematical expressions describing the design flow, or read the file in the *gc* format, obtained from a high-level synthesis tool GAUT; it will then construct the corresponding TED representation. The ordering of TED variables is obtained using several static and dynamic ordering methods. The functionality of two designs can be compared with a help of the *verify* command.

Several DSP transforms have been built into the TED package, including DCT, DFT, WHT, DWT, and others. The TED structure is automatically generated for those transforms by using the command *tr*. The expressions encoded in the TED can be optimized to reduce the number of operations (adders, multipliers) using several optimization options, such as *sub* and *factor*. The TED graph can be displayed at any time using command *show*. Fig. 2 shows the TED graph of the initial WHT transform, and Fig. 3 shows the transform after factorization and variable ordering.

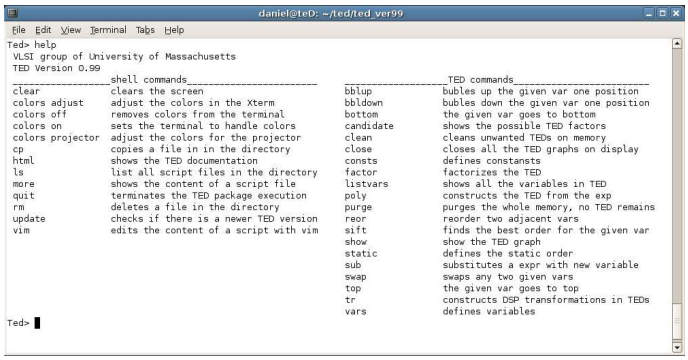

Figure 1. Help file of TED Express

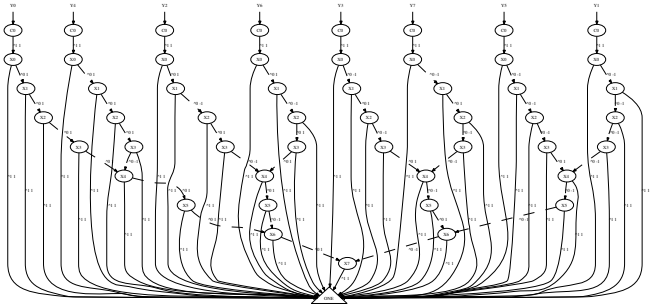

Figure 2. Initial TED for WHT3x3 Transform

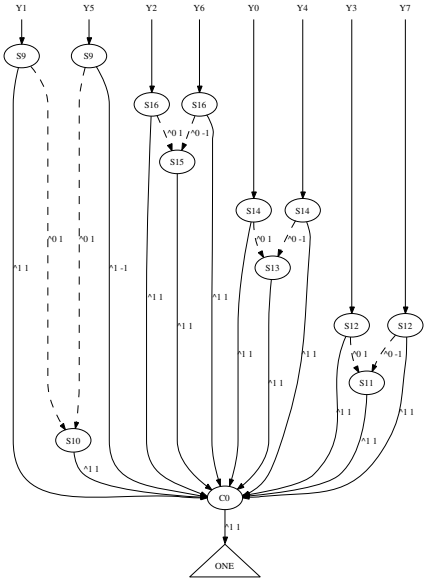

Figure 3. TED of WHT3x3 after factorization