Constraint Satisfaction for Storage Files with Fifos or Stacks during Scheduling

Carlos A. Alba Pinto, Bart Mesman, Koen van Eijk, and Jochen Jess
Eindhoven University of Technology, Design Automation Section
P.O. Box 513, 5600 MB Eindhoven, The Netherlands. E-mail: alba@ics.ele.tue.nl

1 Introduction

This paper presents a method that, during scheduling of DSP algorithms, handles constraints of storage files with fifos or stacks together with resource- and timing constraints. Constraint analysis techniques and the characteristics of the exact coloring of conflict graphs are used to identify values that are bottlenecks for storage assignment with the aim of ordering their accesses. This is done with pairs of values until it can guarantee that all constraints will be satisfied.

2 Problem statement and global approach

Problem Definition 1 *Constrained Storage Assignment and Operation Scheduling Problem. Given a data flow graph, resource constraints, a binding of values to storage files, for each storage file a number of addresses with a specific access behavior (fifo or stack) and a unit size, an initiation interval, and a latency. Find an assignment of storages to values and a schedule that satisfy all constraints.*

In Figure 1, the *constraint analyzer* translates the given resource constraints into additional precedence constraints. These additional precedences refine the *distance matrix* providing a more accurate estimate of the set of feasible start times. A conflict graph is constructed and colored to compute the worst-case number of required addresses. If this number exceeds the one of available addresses, some selected pair of values are access ordered so as reduce the number of conflicts. Once the number of addresses is satisfied, the worst-case size requirement is computed. If this number exceeds the available unit size, some value lifetimes are serialized or extra conflicts are added between them to force them to be stored in different addresses. Finally, the result of partial ordering is passed to a conventional scheduler and storage assignment is performed to complete the process. Figure 2 shows a dataflow graph example and conflict graphs for fifo and stack address satisfaction.

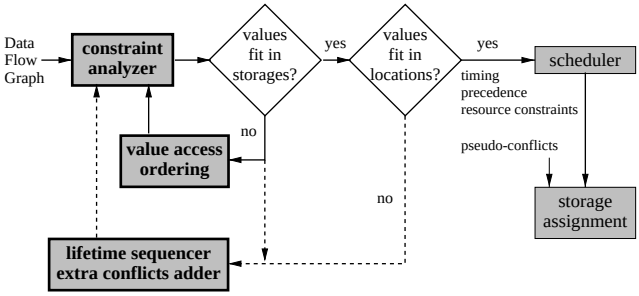

Figure 1. Global approach

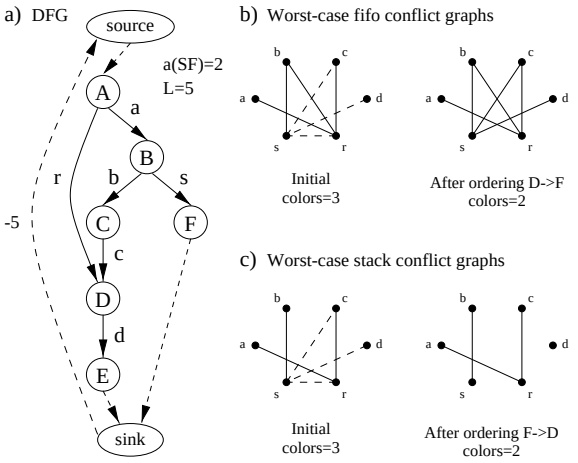

Figure 2. Precedence and conflict graphs

3 Experimental Results

All experiments are run on a 350MHz Pentium II processor machine.

DFG	ops/II/lat	SF addr	SF size	time(s)	mobility
fft	30/2/11	8 fifos	5	1.12	$2.2 \rightarrow 0.1$
ifft	73/14/23	8 fifos	3	5.88	$6.3 \rightarrow 0.5$
		9 stacks	2	3.10	$6.3 \rightarrow 0.4$
loef	56/10/10	8 fifos	2	1.73	$2.4 \rightarrow 0.0$
		12 stacks	2	7.01	$2.4 \rightarrow 0.3$