A TRANSFORMATIONAL APPROACH TO ASYNCHRONOUS HIGH-LEVEL SYNTHESIS

Ganesh Gopalakrishnan [1]
ganesh@cs.utah.edu,
University of Utah,
Dept. of Computer Science,
Salt Lake City, UT 84112, USA

Venkatesh Akella [2]
akella@eecs.ucdavis.edu,
University of California,
Dept. of EE and Computer Engg.,
Davis, CA 95616, USA

UUCS-93-017

Department of Computer Science
University of Utah
Salt Lake City, UT 84112, USA

July 21, 1993

Abstract

Asynchronous high-level synthesis is aimed at transforming high level descriptions of algorithms into efficient asynchronous circuit implementations. This approach is attractive from the point of view of the flexibility it affords in performing high level program transformations on users' initial descriptions, the faithfulness with which it supports the communicating process model of computation, and the ease with which it accommodates computations that have data dependent control-flow decisions as well as data dependent execution times. In this paper, we take the reader through the entire process of synthesizing two asynchronous circuits using our high level synthesis tool, SHILPA, starting from input descriptions in hopCP, emphasizing the *program transformation* techniques employed in the process. Specifically, we show how tail-recursive loops with accumulating parameters can be software pipelined, by evaluating the accumulating parameters in separate processes. We then show how the resulting hopCP flow graphs (HFGs) are transformed through *action refinement* resulting in *normal form HFGs* (NHFGs). NHFGs are then technology mapped onto an Actel FPGA implementation. Our results are illustrated on a pipelined factorial circuit and a pipelined integer square-root circuit.

Keywords: Asynchronous/Self-timed Systems, High Level Synthesis, Program Transformations

[1]Supported in part by NSF Awards MIP 8902558 and 9215878

[2]The work reported here was part of this author's PhD dissertation work, and was supported by a University of Utah Graduate Fellowship