Application Specific Asynchronous Microengines for Efficient High-level Control

Hans Jacobson, Ganesh Gopalakrishnan

Technical Report: UUCS-97-007
Department of Computer Science
University of Utah, Salt Lake City, U.S.A.
E-mail: hans@cs.utah.edu, ganesh@cs.utah.edu

Abstract—Despite the growing interest in asynchronous circuits, programmable asynchronous controllers based on the idea of microprogramming have not been actively pursued. Since programmable control is widely used in many commercial ASICs to allow late correction of design errors, to easily upgrade product families, to meet the time to market, and even effect run-time modifications to control in adaptive systems, we consider it crucial that self-timed techniques support efficient programmable control. This is especially true given that asynchronous (self-timed) circuits are well suited for realizing reactive and control-intensive designs.

We offer a practical solution to programmable asynchronous control in the form of application-specific microprogrammed asynchronous controllers (or microengines). The features of our solution include a modular and easily extensible datapath structure, support for two main styles of handshaking (namely two-phase and four-phase), and many efficiency measures based on exploiting concurrency between operations and employing efficient circuit structures. Our results demonstrate that the proposed microengine can yield high performance—in fact performance close to that offered by automated high-level synthesis tools—targeting custom hard-wired burstmode machines.

I. INTRODUCTION

Sequencing of activities in most VLSI digital circuits is achieved by means of a global clock. Supporting global clocking often comes at very high engineering costs, especially given the trend towards deep submicron VLSI. A well-designed clocking system must, among other things, ensure that the clock cycle time not wasted by the sub-modules. Ensuring this situation involves considerable engineering effort, given the ever-increasing wire-to-transistor delay ratios. This becomes more of a problem in circuits that are reactive and control-intensive in nature. Such circuits receive data values from the external world at unpredictable moments and have to perform efficiently a piece of computation for each data value received, where the computations and control decisions may take a data dependent amount of time. Clocking power is also an increasingly important issue, given the packaging and cooling issues that highly dissipative circuits involve. Though advanced clocking techniques in this area such as distributed clocking methods [1], [2] and/or gated clocking [3] offer a solution to these problems, these techniques are not ready yet for widespread incorporation into general application specific integrated circuit (ASIC) design in a manner that is cost-effective and meets the time to market.

Asynchronous (self-timed) circuits are quite natural for realizing circuits of a reactive and control-intensive nature. Encouraging results are being obtained by many groups in designing self-timed circuits in this domain, for example in communications components used in multiprocessors [4], hardware to network portable electronic devices [5], and digital signal processing algorithms used in audio-electronics hardware [6]. Despite the growing interest in asynchronous circuits, programmable asynchronous controllers based on the idea of microprogramming have not been actively pursued. Since programmable control is widely used in many commercial ASICs to allow late correction of design errors, to easily upgrade product families, to meet the time to market, and even effect run-time modifications to control in adaptive systems, we consider it crucial that self-timed techniques support efficient programmable control. This is especially true given that asynchronous (self-timed) circuits are well suited for realizing reactive and control-intensive designs. For example, supporting families of component types, such as bus adapter chips, is greatly facilitated by programmability. Other examples of systems realized using programmable control (but not using asynchronous control) are the S3MP processor [7] which uses a microprogram engine, and the FLASH processor [8] which uses a processor-core. Programmable asynchronous circuits have also recently shown advantages in embedded and DSP applications [9], [10].

Many of these programmable approaches are very general purpose in their organization to accommodate both pre- and post-fabrication changes of a broad nature. For example, processor cores can be easily re-programmed, and general-purpose microprogram sequencers can be easily equipped with modified microcode. We demonstrate in this work that application specific microprogrammed structures can be easily designed for many classes of circuits, perform at least an order of magnitude better than general-purpose solutions based on processor cores, and even approach the performance of hard-wired control in many cases. The method proposed in this report combines the advantages of programmability and self-timing in an application-specific manner. More specifically, the main contribution of this report is the design and experimental evaluation of a general