STARS: A System for Tuning and Actively Reconfiguring SoC Links

Gregory Diamos and Sudhakar Yalamanchili
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta GA, USA
Email: { gtg250v@mail.gatech.edu, sudha@ece.gatech.edu }

Jose Duato
Department of Computer Engineering
Universidad Politecnica de Valencia
Valencia, Spain
Email: { jduato@gap.upv.es }

Introduction

Modern design regimes are based on worst-case assumptions – a regime that incurs prohibitive costs in performance and yield in the presence of Process, Voltage and Temperature (PVT) variations that are projected to occur at technology nodes of 65nm and below. To combat this problem we propose STARS – a system for actively tuning links to operate at better than worst case margins for timing and voltage. Prior work in this area has focused on developing systems for detecting and recovering from errors. Worm et al [1] have proposed combining bus encoding schemes with Dynamic Voltage Scaling (DVS) to develop robust and efficient links. Similar microarchitecture systems like Razor [3] double sample the signals produced by pipeline stages to determine if aggressive clocking has caused a timing error. These systems allow for robust, better than worst case operation, but must include extra recovery and correction logic to tolerate operation with nonzero error rates.

The STARS System

This abstract presents an approach for dynamically tuning voltage and clock frequency on System on Chip (SoC) synchronous interconnection links to meet given latency and power constraints that is resistant to variable link propagation time and PVT variations. A three latch system with dynamically configurable latch-to-latch delay is used to determine the minimum and maximum signal propagation time across a link driven at a fixed voltage level. A Voltage Controlled Oscillator (VCO) and control logic are then used to adjust the global link clock to match the current signal propagation time. The implementation can be optimized for links with relatively constant propagation times and are tuned continuously, or optimized for links with propagation times that change in response to environmental conditions.

Due to the varying physical dimensions of SoC cores different links will have different physical dimensions and therefore different delay characteristics. A subset of links, called *critical links*, will have the longest propagation delays and determine the maximum clock frequency of the system in the wiring limited regime of future SoC design. Our tuning system dynamically measures the propagation time across these critical links and adjusts the global clock frequency to meet this timing requirement. This model can be expanded to links that require multiple clock cycles to transfer a signal as well as systems with multiple clock domains where critical links in each clock domain are separately tuned.

Our tuning system is founded on techniques developed for datapaths and consists of three main components shown in Figure 1 below: a three latch system with variable latch-to-latch delays, error generation and control logic, and a VCO and voltage regulator. The three latch system is an adaptation of a technique developed at the University of Washington [2] for tuning pipeline stages in multiprocessors and is used to determine the propagation time across the critical link. The error generation logic consists of XOR gates that determine mismatches between the three latches. These error signals are evaluated by the state machine control logic to adjust the delay between latches and control the external VCO and voltage regulator. The figure only illustrates a

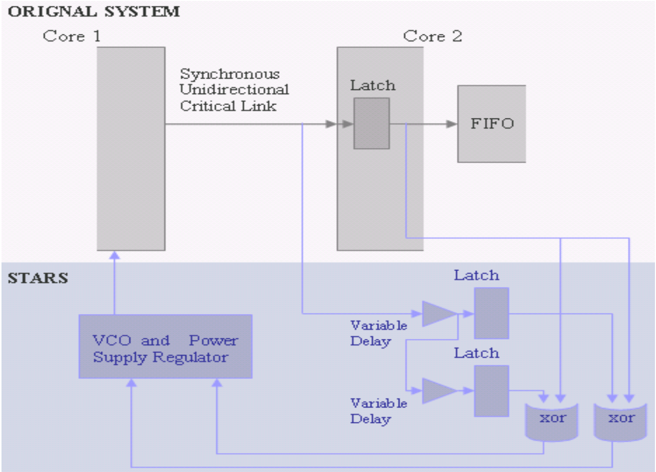

Figure 1. High Level Overview of the STARS System

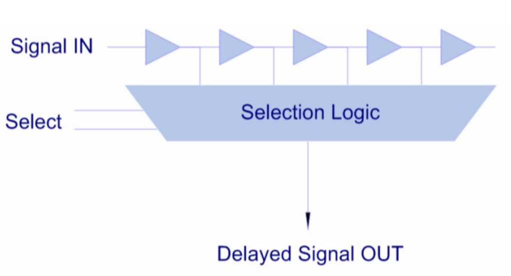

Figure 2. Variable Delay Element

single bit plane or a multi-bit implementation. The system operates under the assumption that the critical link propagation time will have a bounded minimum and maximum value over a relatively long interval of time. The signal generated by the critical link is immediately latched into the first latch at each clock edge and passed into the next logic stage. However, it passes through a variable delay before reaching the second latch and it passes through another variable delay before reaching the third latch. The variable delays shown in Figure 2 above are created by driving the signal from the critical link across a cascade of buffers and selecting the output of a specific buffer.

The system begins with a conservative clock rate and determines the maximum propagation delay of the critical link by increasing the delay to the third latch until the value recorded by the third latch does not match the first latch. In order to avoid incorrect matches between the first and third latch that occur when the signal on the critical link remains constant for consecutive clock cycles, comparisons between latch 1 and 3 are only made on cycles that latch 1 changes. Also, to avoid meta-stable values [3] being saved into the third latch and fluctuations in the propagation time of the critical link, the first and third latch must disagree for a statistically significant number of signal transitions. The minimum propagation time is found in a similar fashion. The delay to the second latch is set to one unit delay less than the third latch. Its delay is then decreased until the first and second latches agree on the same value. Finally, if the delay between the first and second latch is above a threshold value, the clock rate is increased and the process is repeated.

Once the system has been tuned, it will continuously attempt to adjust the delay to the second latch. If the delay between the first and second latch ever drops below a threshold value, the clock speed will be decreased. This allows the system to react to fluctuations in critical link propagation time caused by relatively short term effects such as power supply fluctuations, temperature changes, and cross-talk noise as well as long term effects like electromigration. We are also considering using a fourth latch that is operated on a delayed clock [3] to catch changes in propagation time that occur too fast for the three latch system to react to. Because the fourth latch would operate on a delayed clock, it could only detect errors after they have already occurred and use a recovery protocol such as the one described in [3] for datapaths.

Our approach expands upon previous tuning systems [2] [3] by adding an intelligent control system that uses variable delay elements to tune the clock frequency for a critical link. It is also superior to other schemes including encoding in that it does not require any redesign of existing SoC architectures: the first latch is assumed to be part of the original critical link and the rest of the tuning system does not produce any signals that are consumed by the next logic stage in the SoC. The concept of dynamically tuning critical links has several potential consequences. It hardens systems against manufacturing errors and PVT variations by finding the best operating clock frequency for individual chips. It can adapt to changing environmental conditions like noise and temperature and long term effects like electromigration. It has all of the benefits of a system using DVS because it automatically reacts to changes in supply voltage, i.e., the voltage could be lowered to reduce the chip power consumption and the clock frequency would scale in order to meet the new timing requirements. Finally, it has the potential to increase yield over that achieved by designing to worst case margins.

Present and Future Work

Though some pieces of the STARS architecture, specifically the variable delays and DCO, have strict timing and placement requirements that would greatly benefit from a custom VLSI implementation, we chose to implement STARS first on an FPGA to determine the validity of the tuning algorithm and see if it would be worthwhile to implement a more complete system using a custom VLSI design flow. The complete STARS architecture was simulated using a timing aware VHDL simulator. The tuning algorithm was implemented using a finite state machine, the DCO was implemented as an adjustable clock divider, and the error detection logic was implemented using primitive logic gates. From the simulation, we determined that the tuning algorithm was able to converge to an optimal clock frequency for a given path as long as the resolution in the clock divider was less than or equal to the maximum delay through a variable delay. For paths with delays that changed over time, STARS was able to converge to an optimal clock frequency for a given path as long as the change in delay per cycle was less than the minimum delay between the first and second latch.

After the tuning algorithm and error detection mechanisms were validated via simulation, a structural hardware description of STARS was implemented on a Xilinx Spartan 3e XC3S500E FPGA. Because of limitations in the synthesis tool and FPGA architecture, the variable delays were described using shift registers and the DCO was implemented as a digital clock divider. Two reference systems, a chain of 3-bit multipliers and a naive Fast Fourier Transform (FFT), were tuned using the FPGA version of STARS. For both designs, STARS was able to increase the maximum clock frequency by more than 70% compared to that predicted by static timing analysis with less than a 10% increase in area and less than 3% increase in normalized power consumption. Figures 3 and

4 show a comparison of the a signal propagated across a critical link timed using static timing (Figure 3) and STARS (Figure 4) for the FFT system. For this particular system, STARS was able to increase the operating frequency by 2.065x with no errors for over 24 hours.

Ongoing work will consist of the adaptation of our tuning system to existing protocols for synchronous communication links and quantifying its performance relative to worst-case based solutions. A project has already begun that will focus on an ASIC implementation that will hopefully be able to avoid many of the restrictions imposed by an FPGA design. Separate designs will be analyze the ability of STARS to dynamically tune high speed synchronous interconnects as well as datapaths on a modern VLSI process. This evaluation will cover a range of supply voltage levels and include custom VLSI and standard cell designs. In particular we will continue to focus on a coupled prototyping and simulation based approach to evaluate real world examples of PVT variations that will drive the microarchitecture solutions we have developed.

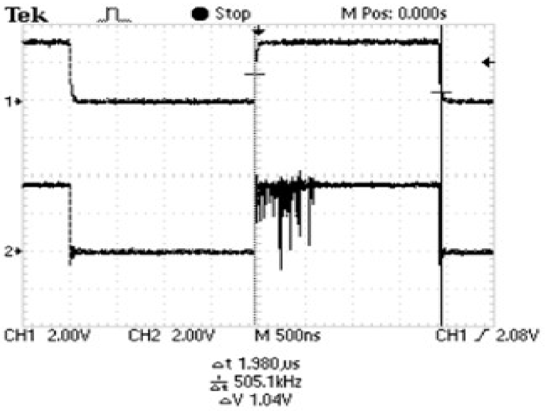

Figure 3*:* Statically Tuned Critical path (FFT)

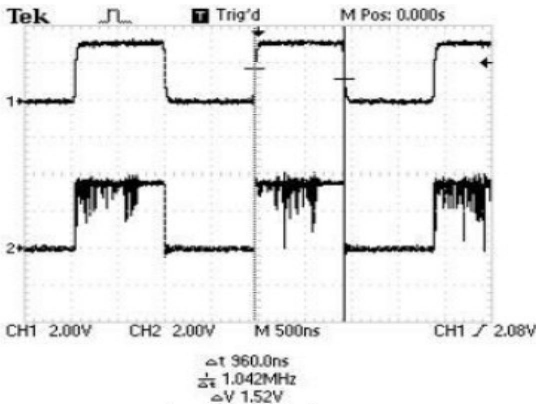

Figure 4: Tuned Critical path (FFT)

References

[1] F. Worm, P. Ienne, P. Thiran and G. De Micheli, “On-Chip Self-Calibrating Communication Robust to Electrical Parameter Variation,” IEEE Design and Test, Vol. 21, No. 6, pp. 524-535, Nov/Dec 2004.

[2] T. Kehl, “Hardware Self-Tuning and Circuit Performance Monitoring,” 1993 Int’l Conference on Computer Design (ICCD-93), October 1993.

[3] Dan Ernst, Nam Sung Kim, Shidhartha Das, Sanjay Pant, Rajeev Rao, Toan Pham, Conrad Ziesler, David Blaauw, Todd Austin, Krisztian Flautner, and Trevor Mudge, "Razor: a low-power pipeline based on circuit-level timing speculation,” Proc. 36th IEEE/ACM Symposium on Microarchitecture. (MICRO 36), 2003, pp. 7-18.