VLSI – IEEE 2011

www.slitetechnologies.com

info@slitetechnologies.com

support@slitetechnologies.com

slitetechnologies@gmail.com

Mobile: 9944389727

The Effect of Multi-Bit Correlation on the Design of Field-Programmable Gate Array Routing Resources

Abstract:

As the logic capacity of field-programmable gate arrays (FPGAs) increases, they are being increasingly used to implement large arithmetic-intensive applications. Large arithmetic intensive applications often contain a large proportion of datapath circuits. Since datapath circuits are designed to process multiple-bit-wide data, FPGAs implementing these circuits often have to transport a large amount of multiple-bit-wide signals from one computing element (such as a logic block, a DSP block, or a multi-bit addressable memory cell) to another. In this work, we investigate the area efficiency of FPGA routing resources for transporting multiple-bit-wide signals. It is shown that, for datapath circuits, the switch patterns used by the conventional routing architecture, which uniformly distribute routing switches across the routing tracks, are inefficient for connecting the computing elements to their tracks. The more efficient multi-bit aware patterns, which contain a densely populated single-bit region and a sparsely populated multi-bit region, can be effectively used to reduce the routing area of FPGAs for implementing arithmeticintensive applications by 6%–10%. It is also shown that the further sharing of configuration memory among the switches within the multi-bit aware patterns does not significantly increase their area efficiency since datapath circuits typically contain a mixture of multi-bit and single-bit signals—while configuration memory sharing can substantially increase the area efficiency of routing resources for transporting multi-bit signals, it also significantly reduces their ability for transporting single-bit signals. More importantly, configuration memory sharing can significantly reduce the effectiveness of the enhanced multi-bit aware patterns—patterns that incorporate both multi-bit aware and single-bit

oriented switches within a single region in order to increase its ability for transporting both single-bit and multi-bit signals.

FPGA Implementation of an Adaptive Filter Robust to Impulsive Noise: Two Approaches

Abstract—

Adaptive filters are used in a wide range of applications such as echo cancellation, noise cancellation, system identification, and prediction. Its hardware implementation becomes essential in many cases where real-time execution is needed. However, impulsive noise affects the proper operation of the filter and the adaptation process. This noise is one of the most damaging types of signal distortion, not always considered when implementing algorithms, particularly in specific hardware platforms. Fieldprogrammable gate arrays (FPGAs) are used widely for real-time applications where timing requirements are strict. Nowadays, two main design processes can be followed for embedded system design, namely, a hardware description language (e.g., VHDL) and a high-level synthesis design tool. This paper proposes the FPGA implementation of an adaptive algorithm that is robust to impulsive noise using these two approaches. Final comparison results are provided in order to test accuracy, performance, and logic occupation.

A High-Resolution Time-to-Digital Converter on FPGA Using Dynamic Reconfiguration

Abstract—

A high-resolution high-precision time-to-digital converter (TDC) architecture is presented for implementation on field-programmable gate arrays (FPGAs) supporting dynamic reconfiguration The proposed architecture relies on multiple parallel high-resolution delay lines implemented by the programmable interconnection points within the routing switch fabric. These delay lines feature a 1-ps resolution over a range of 3 ns. A calibration process is proposed to take process-voltage-temperature variations, as well as clock skew, into account. A TDC with a 50-ps resolution and precision as high as 35 ps has been implemented on a Virtex-II Pro FPGA. Results show that the proposed architecture and calibration process can be used to achieve resolutions as fine as 10 ps.

High Throughput DA-Based DCT With High Accuracy Error-Compensated Adder Tree

Abstract—

In this brief, by operating the shifting and addition in parallel, an error-compensated adder-tree (ECAT) is proposed to deal with the truncation errors and to achieve low-error and high-throughput discrete cosine transform (DCT) design. Instead of the 12 bits used in previous works, 9-bit distributed arithmetic-precision is chosen for this work so as to meet peak-signal-to-noise-ratio (PSNR) requirements. Thus, an area-efficient DCT core is implemented to achieve 1 Gpels/s throughput rate with gate counts of 22.2 K for the PSNR requirements outlined in the previous works.

Time Multiplexed VLSI Architecture for Real-Time Barrel Distortion Correction in Video-Endoscopic Images

Abstract—

A low-cost VLSI implementation of real-time correction of barrel distortion for video-endoscopic images is presented in this paper. The correcting mathematical model is based on least-squares estimation. To decrease the computing complexity, we use an odd-order polynomial to approximate the back-mapping expansion polynomial. By algebraic transformation, the approximated polynomial becomes a monomial form which can be solved by Hornor's algorithm. With the iterative characteristic of Hornor's algorithm, the hardware cost and memory requirement can be conserved by time multiplexed design. In addition, a simplified architecture of the linear interpolation is used to reduce more computing resource and silicon area. The VLSI architecture of this work contains 13.9-K gates by using a 0.18-µm CMOS process. Compared with some existing distortion correction techniques, this work reduces at least 69 % hardware cost and 75 % memory requirement.

Raising FPGA Logic Density Through Synthesis-Inspired Architecture

Abstract—

We leverage properties of the logic synthesis netlist to define both a new field-programmable gate-array (FPGA) logic element (function generator) architecture and an associated technology mapping algorithm that together provide improved logic density. We demonstrate that an "extended" logic element with slightly modified K-input lookup tables (LUTs) achieves much of the benefit of an architecture with K+1-input LUTs, while consuming silicon area close to a -LUT (a -LUT requires half the area of a K+1-LUT). We introduce the notion of "non-inverting paths" in a circuit's AND-inverter graph (AIG) and show their utility in mapping into the proposed logic element architectures. We propose a general family of logic element architectures, and present results showing that they offer a variety of area/performance tradeoffs. One of our key results demonstrates that while circuits mapped to a traditional 5-LUT architecture need 15% more LUTs and have 14% more depth than a 6-LUT architecture, our extended 5-LUT architecture requires only 7% more LUTs and 5% more depth than 6-LUTs, on average. Nearly all of the depth reduction associated with moving from K-input to K+1-input LUTs can be achieved with considerably less area using extended K-LUTs. We further show that 6-LUT optimal mapping depths can be achieved with a small fraction of the LUTs in hardware being 6-LUTs and the remainder being extended 5-LUTs, suggesting that a heterogeneous logic block architecture may prove to be advantageous.

High-Performance and Compact Architecture for Regular Expression Matching on FPGA

Abstract—

We present the design, implementation and evaluation of a high-performance architecture for regular expression matching (REM) on field-programmable gate array (FPGA). Each regular expression (regex) is first parsed into a concise token list representation, then compiled to a modular nondeterministic finite automaton (RE-NFA) using a modified version of the McNaughton-Yamada algorithm.

The RE-NFA can be mapped directly onto a compact register-transistor level (RTL) circuit. A number of optimizations are applied to improve the circuit performance: (1) spatial stacking is used to construct an REM circuit processing $m \geq 1$ input characters per clock cycle; (2) single-character constrained repetitions are matched efficiently by parallel shift-register lookup tables; (3) complex character classes are matched by a BRAM based classifier shared across regexes; (4) a multi-pipeline architecture is used to organize a large number of RE-NFAs into priority groups to limit the I/O size of the circuit. We implemented 2,630 unique PCRE regexes from Snort rules (February 2010) in the proposed REM architecture. Based on the place-and-route results from Xilinx ISE 11.1 targeting Virtex5 LX-220 FPGAs, the proposed REM architecture achieved up to 11Gbps conurrent throughput for various regex sets and up to 2.67x the throughput efficiency of other state-of-the-art designs.

Parallel Interleavers Through Optimized Memory Address Remapping

Abstract—

This work presents mathematical models and collision- free exchange rules for a parallel interleaver, using which it develops an optimized memory address remapping (OPMM) scheme that enables a classic interleaver to be exchanged for a parallel interleaver readily and efficiently. Both analytic and experimental results demonstrate that the rate of annealing achieved using the OPMM approach is much faster than that achieved using the traditional memory address remapping (MM) method.

Improving FPGA Performance for Carry-Save Arithmetic

Abstract—

The selective use of carry-save arithmetic, where appropriate, can accelerate a variety of arithmetic-dominated circuits. Carry-save arithmetic occurs naturally in a variety of DSP applications, and further opportunities to exploit it can be exposed through systematic data flow transformations that can be applied by a hardware compiler. Field-programmable gate arrays (FPGAs), however, are not particularly well suited to carry-save arithmetic. To address this concern, we introduce the "field programmable counter array" (FPCA), an accelerator for carry-save arithmetic intended for integration into an FPGA as an alternative to DSP blocks. In addition to multiplication and multiply accumulation, the FPCA can accelerate more general carry-save operations, such as multi-input addition (e.g., add integers) and multipliers that have been fused with other adders. Our experiments show that the FPCA accelerates a wider variety of applications than DSP blocks

and improves performance, area utilization, and energy consumption compared with soft FPGA logic.

The ARPA-MT Embedded SMT Processor and Its RTOS Hardware Accelerator

Abstract—

The high-level modeling and parameterization capabilities of current hardware description languages, as well as the huge integration capacity and flexibility provided by modern fieldprogrammable gate arrays (FPGAs), open the way to designing processors tuned to given applications and favoring specific properties. This paper presents the Advanced Real-time Processor Architecture (ARPA)—MultiThreaded processor—a customizable, synthesizable, and time-predictable processor model optimized for multitasking real-time embedded systems, which efficiently explores modern FPGA technology. A fundamental processor component is the *ARPA operating system* (OS) *coprocessor* designed for hardware implementation of the basic real-time OS management functions, such as timing, task scheduling, synchronization and switching, efficient interrupt handling, and verification of the timing constraints. The hardware implementation of these functions allows executing them faster and more predictably, reducing the OS overhead, and improving its determinism. The performance evaluation has shown reductions of one to two orders of magnitude in the execution time of some functions of a real-time executive, in comparison with an analogous software implementation.