VLSI PROJECT ABSTRACTS

Network Security & Cryptographic Sciences, Digital Signal Processing, Arithmetic Core and Digital Electronics, Digital Communications and Information theory, Digital Image Processing, Bus Protocols and System on Chip

IEEE VLSI Based projects based on verilog and Xilin in Arithmetic Core

1. High Speed 16-bit Digital Vedic Multiplier using FPGA.

In the present paper our objective is to emphasize the importance of Vedic Mathematics for digital applications. Ancient vedic mathematics not only facilitate the complex mathematical operations but also useful for logical applications. In the present work we are using the concept of Urdhva-tiryakbyham, i.e., vertically and crosswise multiplication and it's implementation for 16-bit multiplication. This technique optimizes the output in term of steps of calculation and therefore reduces the delay of a digital circuit. We implemented these results with the help of front end language— Verilog. Results obtained from simulation and syntheses have been verified on Spartan 3E FPGA using Xilinx ISE Suite are discussed in details. Obtained results have been compared with the most frequently used multipliers in digital circuits which illustrate 38 % reduction in device utilization and 62% reduction in delay.

2. Design and implementation of fast floating point multiplier unit.

Floating point numbers are the quantities that cannot be represented by integers, either because they contain fractional values or because they lie outside the range re presentable within the system's bit width. Multiplication of two floating point numbers is very important for processors. Architecture for a fast floating point multiplier yielding with the single precision IEEE 754- 2008 standard has been used in this project. The floating point representation can preserve the resolution and accuracy compared to fixed point. Pipeline is a technique where multiple instructions are overlapped in execution. Multiple operations performed at the same time by pipeline will increase the instruction throughput. In several high performance computing systems such as digital signal processors, FIR filters, microprocessors, etc multipliers are key components. The most important aim of the design is to make the multiplier quicker by decreasing delay. Decrease of delay can be caused by propagation of carry in the adders having smallest amount power delay constant.

3. FPGA Implementation of Vedic Floating Point Multiplier.

Most of the scientific operation involve floating point computations. It is necessary to implement faster multipliers occupying less area and consuming less power. Multipliers play a critical role in any digital design. Even though various multiplication algorithms have been in use, the performance of Vedic multipliers has not drawn a wider attention. Vedic mathematics involves application of 16 sutras or algorithms. One among these, the Urdhva tiryakbhyam sutra for multiplication has been considered in this work. An IEEE-754 based Vedic multiplier has been developed to carry out both single precision and double precision format floating point operations and its performance has been compared with Booth and Karatsuba based floating point multipliers. Xilinx FPGA has been made use of while implementing these algorithms and a resource utilization and timing performance based comparison has also been made.
Keywords:- Vedic multiplication, FPGA, Floating Point Hardware Testing Cores.

VLSI PROJECT ABSTRACTS

Network Security & Cryptographic Sciences, Digital Signal Processing, Arithmetic Core and Digital Electronics,
Digital Communications and Information theory, Digital Image Processing, Bus Protocols and System on Chip

4. Low-Cost Scan-Chain-Based Technique to Recover Multiple Errors in TMR Systems

In this paper, we present a scan-chain-based multiple error recovery technique for triple modular redundancy (TMR) systems (SMERTMR). The proposed technique reuses scan-chain flip-flops fabricated for testability purposes to detect and correct faulty modules in the presence of single or multiple transient faults. In the proposed technique, the manifested errors are detected at the modules' outputs, while the latent faults are detected by comparing the internal states of the TMR modules. Upon detection of any mismatch, the faulty modules are located and the state of a fault-free module is copied into the faulty modules. In case of detecting a permanent fault, the system is degraded to a master/checker configuration by disregarding the faulty module. FPGA-based fault injection experiments reveal that SMERTMR has the error detection and recovery coverage of 100% and 99.7% in the presence of single and two faulty modules, respectively, while imposing negligible area and performance overheads on the traditional TMR systems.

5. Low-Power and Area-Efficient Carry Select Adder

Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. From the structure of the CSLA, it is clear that there is scope for reducing the area and power consumption in the CSLA. This work uses a simple and efficient gate-level modification to significantly reduce the area and power of the CSLA. Based on this modification 8-, 16-, 32-, and 64-b square-root CSLA (SQRT CSLA) architecture have been developed and compared with the regular SQRT CSLA architecture. The proposed design has reduced area and power as compared with the regular SQRT CSLA with only a slight increase in the delay. This work evaluates the performance of the proposed designs in terms of delay, area, power, and their products by hand with logical effort and through custom design and layout in 0.18- m CMOS process technology. The results analysis shows that the proposed CSLA structure is better than the regular SQRT CSLA.

6. FPGA Design of a Fast 32-bit Floating Point Multiplier Unit

An architecture for a fast 32-bit floating point multiplier compliant with the single precision IEEE 754-2008 standard has been proposed in this paper. This design intends to make the multiplier faster by reducing the delay caused by the propagation of the carry by implementing adders having the least power delay constant. The implementation of the multiplier module has been done in a top down approach. The sub-modules have been written in Verilog HDL and then synthesized and simulated using the Xilinx ISE 12.1 targeted on the Spartan 3E

7. Design & Implementation of Floating point ALU

In this paper, the implementation of DSP modules such as a floating point ALU are presented and designed. The design is based on high performance FPGA "Cyclone TI" and implementation is done after functional and timing simulation. The simulation tool used is ModelSim. The tool for synthesis and implementation is Quartus n. The experimental results shows the functional and timing analysis for all the DSP modules carried out using high performance synthesis software from Altera.

8. A FPGA IEEE-754-2008 DECIMAL64 FLOATING-POINT ADDER/SUBTRACTOR

This paper describes the FPGA implementation of a Decimal Floating Point (DFP) adder/subtractor. The design performs addition and subtraction on 64-bit operands that use the IEEE 754-2008 decimal encoding of DFP numbers and is based on a fully pipelined circuit. The design presents a novel hardware for pre-signal generation stage and an enhanced version of previously published leading zero stage. The design can operate at a frequency of 200 MHZ on a Virtex-5 with a latency of 8 cycles. The presented DFP adder/subtractor supports operations on the decimal64 format and it is easily extendable for the decimal128 format. To our knowledge, this is the first hardware FPGA design for adding and subtracting IEEE 754-2008 using decimal64 encoding.

9. An Efficient Implementation of Floating Point Multiplier

In this paper we describe an efficient implementation of an IEEE 754 single precision floating point multiplier targeted for Xilinx Virtex-5 FPGA. VHDL is used to implement a technology-independent pipelined design. The multiplier implementation handles the overflow and underflow cases. Rounding is not implemented to give more precision when using the multiplier in a Multiply and Accumulate (MAC) unit. With latency of three clock cycles the design achieves 301 MFLOPs. The multiplier was verified against Xilinx floating point multiplier core.