

## ABSTRACT

The advancement in Very Large Scale Integration (VLSI) technology has allowed the integration of more and more functionalities onto a single chip. Precisely estimating power for complex digital VLSI circuits at an earlier stage can save the complicated and expensive redesign task which would occur, if power constraints in the specifications are violated. Portable systems like notebook computers, laptops, mobiles and Personal Digital Assistants (PDA) demand reduced power consumption to enhance the battery lifetime. There is a steady growth in the operating frequency of Microprocessors ( $\mu$ Ps) and Digital Signal Processors (DSPs) in day-today life. This causes increased power dissipation which in turn can create thermal hot spots. This may lead to reduced circuit reliability and life expectancy. Hence there is a strong need to reduce power consumption when designing complex microelectronic digital circuits and systems. Major power dissipation blocks for  $\mu$ P and DSP are adders and multipliers residing in the Arithmetic and Logic Unit (ALU).

This thesis mainly focuses on power estimation of Benchmark circuits and power-delay optimization of arithmetic circuits. Binary addition is the most important primitive block in an ALU. Hence it is necessary to design high performance low power adders. First work, in this thesis, involves the design of a high performance adder cell for multiply and accumulate units. A new 14-Transistor (14-T) adder has been proposed. The proposed adder has

less power delay product compared to many of the existing ones. The Proposed 14-T adder is tested for cascading capabilities by implementing Array, Carry-Save and Dadda multipliers. The performance of the Proposed 14-T adder is also tested by using it in DSP filters. Based on the performance metrics such as power, delay, power-delay product and output voltage signal swing, it is observed that the Proposed 14-T is best suited for DSP applications.

Second work in this thesis focuses on design of high speed multipliers. Multiplier forms the prime operation in a DSP for executing dedicated algorithms such as convolution, correlation and filtering. Two algorithms have been proposed for implementing multipliers. The first algorithm based on decomposition technique is tested on Carry-Save, Wallace and Dadda multipliers. It is observed that multipliers implemented using decomposition technique outperform their un-decomposed equivalent one's in terms of power-delay product. It is observed that the best method for decomposition of a  $N \times N$  multiplication is using four  $\frac{N}{2} \times \frac{N}{2}$  partitioned blocks. The second algorithm based on bypassing technique is tested on Wallace and Dadda multipliers. This algorithm offers minimal power-delay product when the operand has more number of zeros than ones.

Third work in this thesis involves the design of a new Parallel Prefix Adder (PPA) architecture. PPA is best suited for wider word lengths. A modified PPA architecture has been proposed for 8-bit, 16-bit, 32-bit and 64-bit word lengths. Proposed architectures employ four different

computational cells to achieve least power-delay product amongst the existing PPA architectures. The performance of the proposed architectures is studied using three schemes namely Scheme I, Scheme II and Scheme III. It is observed that Scheme I provides least delay and least power-delay product for all word lengths amongst three schemes. Scheme III has least power for all word lengths amongst the three schemes.

The fourth work involves estimation of power in International Symposium on Circuits and Systems 1989 (ISCAS'89) Benchmark circuits using Back Propagation Neural Network (BPNN) and Radial Basis Function Neural Network (RBFNN). BPNN is trained with the data set containing information regarding the count of the number of inputs, outputs, inverters, gate count and D Flip-flops. Target for each vector is the Monte Carlo (MC) power values for that circuit. BPNN is trained using eleven different training algorithms namely Traingd, Traingda, Trainrp, Traingdx, Traingdm, Traincgf, Traincgp, Traincgb, Trainscg, Trainoss and Trainbfg. It is found that scaled conjugate gradient (trainscg) training function is best suited for power estimation. RBFNN is trained using the same data set. It is observed that RBFNN outperforms BPNN.