Chapter 1
Introduction

Contents

Chapter 1 ........................................................................................................................................... 1
1.1 Motivation .................................................................................................................................. 1
1.2 Objectives of the thesis .............................................................................................................. 2
1.3 Organization of the thesis .......................................................................................................... 3

1.1 Motivation

In a microprocessor or a digital signal processor (DSP), data path plays a prominent role since performance metrics like the die-area, speed of operation, power dissipation etc., depend directly on the efficiency of data-path. As is known, core of the data path involves complex computations like addition, subtraction, multiplication and division, etc. Thus, realizing efficient hardware units for these computations, which directly affect the performance of data path, is of prime importance.

The most executed operation in the data path is addition, which requires a binary adder that adds two given numbers. Adders also play a vital role in more complex computations like multiplication, division and decimal operations. Hence, an efficient implementation of binary adder is crucial to an efficient data path. Relatively significant work has been done in proposing and realizing efficient adder circuits for binary addition as described in the next chapter. However, as the technology is scaling down new design issues like fan-out and wiring complexity are appearing in the front-line. These issues are addressed to some extent by new adder architectures known as sparse adders. As operand size increases, sparse adders also suffer from above design issues that are becoming vital as they have direct impact on the performance of an adder. Thus, there is an urgent need to develop alternative sparse adder architectures which can address these design issues.

The next most important block in data-path after adder is the multiplier, which is also very crucial in ASICs and DSPs. High speed multipliers reported in literature use parallel
1.2 Objectives of the thesis

multiplier architectures that employ counters/compressors along with adders as basic building blocks. Counters are multi-input, multi-output combinational logic circuits that determine the number of logic ‘1s’ in their input vectors and generate a binary coded output vector that corresponds to this number. A counter differs from a compressor in that compressors have carry inputs and carry outputs in addition to the “normal” inputs and outputs which counters do not have. As these blocks lie directly within the critical path of a given design, thus dictating the overall circuit performance, there is an urgent need to design and validate new high speed/low power counters and compressors.

Further, some of the recursive arithmetic operations that appear in processors/controllers other than addition and multiplication are increment and decrement operations. The increment and decrement operations count up or down by one step which can be performed by incrementer/decrementer (INC/DEC) block. This block also finds its application in address generation unit in processors and frequency dividers. The architectures of binary INC/DEC block are mainly based on adder/subtractor, counter or carry look-ahead adder.

Finally, despite the widespread use of binary arithmetic, decimal computation remains essential for many applications. Not only is it required whenever numbers are presented for human inspection, but is also often a necessity when fractions are involved. Decimal fractions are pervasive in human endeavors, yet most cannot be represented by binary fractions. Still, the major consideration while implementing Binary Coded Decimal (BCD) arithmetic will be to enhance its speed as much as possible while facilitating even binary applications on the same hardware. There are different architectures that support BCD as well as binary operations on the same hardware. However, when signed computations are required, the existing architectures, use 10’s or 9’s complement to implement subtraction in BCD. This introduces extra latency in the conversion process, which requires an architecture that can reduce/eliminate the correction latency in BCD adders.

1.2 Objectives of the thesis

With the above modifications, the following objectives are proposed to be addressed in this thesis

- Efficient realization of higher operand bit adders (for 32-bit and above).
- Realization of efficient counters/compressors for high speed parallel multiplication.
- Design of high performance stand-alone blocks like incrementer, decremementer etc.
1.3 Organization of the thesis

- Implementation a unified Binary/BCD adder with improved performance.
- Demonstration of efficient arithmetic section of an ALU using the proposed basic units

1.3 Organization of the thesis

As the adder is the basic building block in designing most of the arithmetic circuits, chapter 2 discusses in detail various types of adder architectures and their realization in hardware. A high performance sparse adder architecture (for 32-bit and above) is proposed and studied in detail.

Multiplication finds a wide range of usage in signal processing hardware implementations. In Chapter 3, a special block known as counter/compressor is used in partial product reduction tree in multipliers and analyzed. An efficient counter/compressor block is proposed which makes use of signals and their complements available in CMOS implementation. A generalized n-bit counter is proposed which can be customized to any operand bit size.

The branching and interrupt instructions in any microprocessor need the help of special hardware blocks. In Chapter 4, dedicated hardware blocks like Incrementer/Decrementer, 2’s complementer, priority encoder etc., are analyzed and a multi-functional block is proposed which can perform all the above operations using the same hardware.

The emphasis on error free arithmetic is increasing day by day and decimal arithmetic circuits are slowly taking the center stage. In Chapter 5, efficient decimal arithmetic hardware implementation that can perform both signed and unsigned arithmetic is proposed. The implementation reduces the hardware and thereby propagation delay with the proposed end around carry method of subtraction. The same hardware that implements the decimal arithmetic can be used for binary arithmetic without any degradation in performance.

The usage of the above-proposed arithmetic blocks in a multiplier and floating point adder is studied in Chapter 6 and a segment of a processor core is designed with the proposed arithmetic units in Chapter 7. Finally, the scope for further work is suggested in Chapter 8.