11329524

DCIT207

The ninth edition of William Stallings' "Computer Architecture and Organisation" includes a chapter on computer arithmetic. The concepts and methods underlying executing arithmetic operations in computer systems are examined in this chapter.

Number systems used in computers, such as binary, decimal, and hexadecimal, are introduced at the beginning of the chapter. It describes how binary numbers are expressed and goes over several ways to convert between different number systems. Additionally covered are the applications of sign bits, magnitude representation, and two's complement representation for signed numbers.

The chapter then explores binary arithmetic. Addition, subtraction, multiplication, and division are all covered. Each operation's methods and algorithms are thoroughly described. Concepts like carry propagation and borrow propagation are involved in addition and subtraction.

Techniques including ripple carry adders, carry-lookahead adders, and carry-save adders are covered in this chapter. It draws attention to the compromises made between performance and hardware complexity in various adder designs.

The chapter examines many multiplication algorithms, including Wallace tree multiplication and Booth's method. It illustrates how these techniques may lower the amount of necessary hardware components and increase performance. The design and implementation of multiplier circuits, including ideas like partial products and carry-save adders, are also covered in this chapter.

The chapter discusses division notions such SRT division, non-restoring division, and restoring division. It talks about the hardware needed for division operations and describes procedures in these algorithms.

One of the most important topics discussed in this chapter is floating-point arithmetic, which is necessary for handling real numbers and scientific computations. It explains how to express floating-point integers with an exponent, a sign bit, and a mantissa. The chapter covers addition, subtraction, multiplication, and division on floating-point integers as well as the IEEE 754 standard for floating-point representation. Special instances like zero, infinity, and NaN (Not a Number) are also covered.

This chapter delves deeper into the architecture and functionality of arithmetic logic units (ALUs) found in central processing units (CPUs) of computers. Adders, subtractors, and multipliers are among the ALU components that are covered. Explained are several ALU architectures, including as single-cycle and multi-cycle designs. The chapter explores the control logic needed to operate an ALU and emphasises the function of control signals in carrying out arithmetic instructions.

Many subjects pertaining to computer arithmetic are covered in this chapter. This involves the idea of fixed-point arithmetic, which is used to problems requiring a low degree of accuracy. The representation and manipulation of fixed-point numbers are explained in this chapter, along with the difficulties posed by overflow and underflow.

In addition, the chapter discusses carry lookahead adders, parallel adders, and the use of arithmetic operations in high-level programming languages. It investigates the trade-offs between performance and hardware complexity in various arithmetic systems.

The chapter also discusses problems with precision and accuracy in computer arithmetic. It clarifies the restrictions brought about by limited precision as well as the incidence of rounding mistakes. Additionally covered are the idea of significant numbers and how mathematical operations affect accuracy.

The chapter explores complex subjects like modular arithmetic, the Chinese Remainder Theorem, and hardware implementation of division algorithms. For effective modular arithmetic, it encompasses ideas like Barrett reduction and Montgomery multiplication.

The chapter also covers how computer arithmetic affects system performance as a whole. It draws attention to the significance of effective arithmetic operations in a range of applications and offers suggestions for performance optimisation techniques. In order to improve computing efficiency, it looks at methods like pipelining, parallel processing, and the usage of specialised arithmetic units, such graphics processing units.

The difficulties and factors to be taken into account while creating arithmetic units for various processor types, including high-performance computing architectures, embedded systems, and general-purpose CPUs, are also covered in this chapter.

In conclusion, "Computer Architecture and Organization"'s Chapter 10 offers a thorough review of computer arithmetic. It includes binary arithmetic, floating-point arithmetic, number representation, and designing arithmetic logic units.

The chapter examines the methods, procedures, and features of arithmetic operations in computer systems from an implementation perspective. Fixed-point arithmetic, parallel adders, accuracy, precision, sophisticated algorithms, modular arithmetic, and system performance concerns are among the other subjects it covers. The concepts and methods behind carrying out arithmetic operations in computer systems are covered in this chapter.