Unit:2 (Detailed Notes)

Fixed and floating point representation: They are the methods used in arithmetic and logic units to handle numerical data in digital systems.

① Fixed Point Representation: • Numbers are represented with a fixed number of integer and fractional bits.

- Format: For example, in a 16 bit fixed point format with 8 integer bits and 8 fractional bits, the range might be from -128 to 127.9999

- Arithmetic Operations: Fixed-point arithmetic operations involve simple integer arithmetic, followed by scaling to adjust for fractional parts.

★ Pros: • Simple to implement in Hardware.
  • Suitable for applications with known and limited range and precision requirements.

★ Cons: • Limited dynamic range and precision.
- Can lead to loss of precision or overflow/underflow errors in calculations involving very large or very small numbers.

② Floating Point Representation: • In floating point representations numbers are represented as a combination of a sign bit, exponent and mantissa (fraction)

- Format: Typically follows the IEEE 754 standards with different precision operations like Single Precision (32 bit) or double precision (64 bits)

- Arithmetic operations: Floating point arithmetic Involves complex operations to handle normalization, rounding, and exponent manipulation.

- Pros: • Offers a wide dynamic range and precision, suitable for scientific and engineering applications.

- Allows representation of very large and very small numbers with reasonable precision.

- Cons: • More complex to implement in Hardware compared to fixed point arithmetic.

- Requires additional processing time and resources for operations like normalization and rounding

IEEE Standards for floating point representation:

★ Single Precision Format (32 bits)

1) Sign bit : 1 bit (Represents the sign of the number, 0 for positive and 1 for negative).

2) Exponent: 8 bits (Represents the exponent of the floating point number in biased form).

3) Significand (Mantissa): 23 bits (Represents the fractional part of the floating point number.

★ Double Precision Format:

1) Sign bit: 1 bit
2) Exponent: 11 bit
3) Significand (Mantissa) 52 bits

Standard features: 1) Normalized Numbers
2) Exponent Bias
3) Special Values
4) Rounding Modes
5) Arithmetic operations.

Booth's Algorithm: It is a multiplication algorithm that efficiently multiplies two signed binary numbers using a sequence of addition and shifting operations.

Process: a) Represent the 2 numbers in signed binary format

b) Divide one of the numbers into groups of adjacent bits.

c) Apply a series of addition & shifting operations based on the bit pattern of the other number.

Advantages: • Requires fewer arithmetic operations compared to traditional methods, especially for large numbers.

- Suitable for hardware implementation in digital circuits due to regular and repetitive structure.

Disadvantages: • Complexity increases with the size of the numbers being multiplied.

- May require additional hardware resources for implementation in hardware circuits.

Array Multiplier: It is a hardware implementation of multiplication that utilizes an array of adders and shifters to perform the multiplication operation.

Process: a) Represent the 2 numbers to be multiplied in binary format.

b) Organize the bits of one number into rows of a matrix.

c) Organize the bits of the other number into columns of the same matrix.

d) Perform a series of partial products to obtain the final result by adding them.

Advantages:
- Can handle wide input sizes.
- Provides a straightforward implementation in hardware using adders & shifters.

Disadvantages:
- Requires large numbers of adders & shifters.
- Slower for large operands.

Components of ALU Design: 1) Arithmetic Unit: Handles arithmetic operations.

- Consists of circuits for performing binary addition and subtraction using carry propagate or carry-lookahead techniques.
- Multipliers and dividers may also be included in multiplication and division operations.

2) Logic Unit: • AND, OR, NOT, XOR. are performed by Logic Unit.

- Utilizes Logic Gates (AND, OR, NOT) to perform operations on binary data.

3) Control Unit: • Decodes instructions fetched from memory and generates control signals to coordinate the activities of ALU.

- Determines which operation to perform based on the instruction opcode and operands.

4) Registers: • Holds operands, intermediate results, and operation flags during ALU operations.

- Includes GPRs, used for storing data and special purpose registers for storing data and special flags and control operations.

Design Consideration: ① Speed: ALU operations should be executed quickly to minimize overall processing time.

② Precision: ALU should provide accurate results.

③ Flexibility: ALU design should support a wide range of operations and data types.

④ Power Efficiency: Power Consumption should be energy efficient.

Implementation Techniques: 1) Combinational Logic: Basic operations are performed by combinational logic circuits.

2) Sequential Logic: More complex ALU designs include flip flops and state machines.

3) Parallelism: By processing multiple operations simultaneously by using parallel adders, multipliers and pipelined execution.

4) Hardware Accelaration: Hardware Accelarators like floating point units (FPUs) and Vector Processing

Units (VPU's) maybe included to accelarate arithmetic and logical operations.

Modern Trends:

1) Vector Processing: ALUs optimized for Vector operations are used in Scientific Computing and Graphic processing.

2) Customization: ALU designs can be customized for specific applications and architectures.

3) Heterogeneous Computing: ALUs are integrated into heterogeneous computing systems alongside other specialized processing units like GPUs, TPUs and accelerators to leverage their respective strengths for different types of computations.