Appendix

Appendix A

Fixed point binary number assumes a **pre-determined number of bits** before and after the point.

Floating Point Binary Numbers

Floating point binary allows very large numbers to be represented.

In **scientific notation** $m \times 10^n$, m is known as the **mantissa** and n the **exponent**.

- The leftmost bit of both the mantissa and the exponent are sign bits.
- Since both numbers are represented using **two's complement**.
- The binary point is to the right of the sign bit.

Normalisation

Normalisation is the process of moving the binary point of a floating point number to provide the **maximum level of precision** for a given number of bits.

This is achieved by ensuring that the first digit after the binary point is a significant digit. In normalised floating point form,

- A positive number has a sign bit of 0, and the next bit is always 1.
- A negative number has a sign bit of 1, and the next bit is always 0.

The size of the mantissa determine the **precision** of the number, the size of the exponent determine the **range** of the numbers that can be held.

Rounding errors are unavoidable and result in a loss of accuracy.

- The **absolute error** is calculated as the difference between the number to be represented, and the actual binary number that is the closest possible approximation in the given number of bits.
- The **relative error** is the absolute error divided by the number.

Advantages and Disadvantages

- Floating point allows a far **greater range** of numbers using the same number of bits - very large numbers and very small fractional numbers can be represented.

  The larger the mantissa, the greater the precision, the larger the exponent, the greater the range.

- In fixed point binary, the range and precision of the numbers that can be represented **depends on the position of the binary point**.

  Fixed point binary is a **simpler system** and is faster to process.

Underflow occurs when a number is too small to be represented in the allotted number of bits. **Overflow** occurs when the result of a calculation is too large to be held in the number of bits allotted.

Appendix B

- A **half-adders** take an input of two bits and give a two-bit output as the correct result of an **addition of the two inputs**.
- A **full-adder** combines two half-adders to **add three bits together**.

Multiple full adders can be connected together, n full adders can be connected together to create an adder capable of adding a binary number of n bits.

D-type Flip-flops

- A **flip-flop** is an elemental **sequential logic circuit** that can store one bit and flit between two states.

  It has two inputs - a **control input** D and a **clock signal**.

- A **clock** is a type of sequential logic circuit that changes state at regular time intervals.

  Clocks are needed to synchronise the change of state of flip-flop circuits.

- A **D-type flip-flop** is a **positive edge-triggered flip-flop**, it can only change the output value from 1 to 0 or vice versa when the clock is at a rising edge.

  When the clock is not at a positive edge, the input value is held and does not change.

The flip-flop circuit is important because it can be **used as a memory cell** to store the state of a bit.

A flip-flop comprises of several NAND gates and is effectively 1-bit memory. **Register memories** and **static RAM** are created using D-type flip-flops.