ARQUITECTURA DE COMPUTADORES

Grado en Ingeniería Informática

Práctica – Simulador de una ALU de Coma Flotante

David Martínez Martínez

Índice

1. Introducción

Desarrollo de un simulador en lenguaje C++ de una Unidad Aritmético-Lógica que opere con valores representados en coma flotante según el estándar 754 de IEEE.

2. Consideraciones

1. Es obligatorio utilizar los algoritmos de Aritmética en coma flotante. No se dará por válido cualquier otro algoritmo que se utilice para la resolución de la práctica. Si la práctica no está resuelta con dichos algoritmos tendrá una calificación de 0 y la parte práctica de la asignatura estará suspensa.
2. Crear una clase que proporcione la funcionalidad de una ALU para coma flotante con las siguientes especificaciones:
   - Opera con valores de 32 bits que interpreta como reales de precisión simple según el estándar 754 de IEEE. El resultado tiene las mismas características que los datos.
   - Todos los valores posibles representables con el estándar IEEE 754 son válidos como operandos fuente y como resultado.
   - El código correspondiente a la clase no debe utilizar ningún tipo real de los suministrados por el lenguaje C++, ni tampoco utilizar caracteres para representar bits individuales.
   - Todas las variables utilizadas para implementar la clase deben ser invisibles desde fuera de la propia clase.
   - La clase ALU debe reconocer la operación que se ha de realizar entre los dos operandos a partir de una señal de control que se le envíe desde la interfaz de usuario.
   - La ALU soporta las operaciones producto, suma y división (utilizando el algoritmo de Newton).
   - Solamente se comprobará si hay desbordamientos cuando la operación sea un producto.
   - Especificaciones para el algoritmo de Goldschmidt:
     - Tabla para aproximar el inverso de un número real:

b	b'≈1/b
1.00	1.00
1.25	0.80

- El resto de operaciones (sumas y productos) de dicho algoritmo debe realizarlas la propia ALU.
- El proceso iterativo finaliza cuando la diferencia entre un resultado y el de la iteración anterior sea menor que 10^{-4}.

3. Funciones de la interfaz de usuario:
   - Leer los valores de los dos operandos, que el usuario introducirá desde el teclado como valores reales.
   - Identificar el tipo de operación que se desea realizar.
   - Comunicar a un ejemplar de la clase ALU los valores de los operandos, el tipo de la operación y solicitarle el resultado.
   - Mostrar el valor en formato hexadecimal tanto de los operandos como del resultado calculado.
   - Mostrar el valor decimal del resultado.

3. Formato de entrega

- Esta entrega deberá realizarse a través de agora.unileon.es
- La entrega debe contener:
    - **Código fuente**.
    - Archivo de texto **README** con los pasos a seguir para compilar y ejecutar el programa. Estas instrucciones deben ser independientes del entorno de desarrollo o editor utilizados.
    - Esquema con la jerarquía de clases utilizada.
- Todos estos archivos se entregarán en un único archivo comprimido en formato zip, cuyo formato será:

  Idusuario_practica4.zip

  Donde idusuario es el usuario del correo electrónico:

  idusuario@estudiantes.unileon.es
- **Fecha límite de entrega:** 13 de Mayo de 2019
- **Las defensas de las prácticas se harán el siguiente día que tengamos clase.**

Arquitectura de Computadores

David Martínez Martínez

dmartm@unileon.es