# PROYECTO DE ORGANIZACIÓN DE COMPUTADORES 2018-2S: PRIMER PARCIAL

Tema: Convertidor de números de punto flotante a binario en formato IEEE 754 de simple y doble precisión.

**Descripción:**

El proyecto consiste en modelar un convertidor de números de punto flotante (erróneamente llamados decimales) a números binarios. Utilizando el estándar IEEE 754, que es utilizado en muchas y diferentes arquitecturas.

El objetivo de este trabajo es mostrar cómo guarda un computador internamente los números de punto flotante en la memoria, y a su vez, conocer cómo se almacenan en el Registro.

Adjuntos a este archivo se encuentran dos archivos con el código fuente del programa, uno en lenguaje C y otro en lenguaje ensamblador con estándar MIPS.

**Metodología:**

La elaboración de proyecto consistió en tres fases: la investigación sobre la relación entre la memoria y los números flotantes, la búsqueda de trabajos previos y finalmente la implementación del programa en ambos lenguajes.

Los números de punto flotante se guardan en memoria con el formato IEEE 754 (Koppelman, D.), por lo cual, una conversión como tal no es necesaria. La lógica del programa consiste entonces en manipular estos registros para presentar la información al usuario de manera natural. En este sentido, el programa en C difiere un poco al programa en ensamblador, en cuanto a sus mecanismos.

Para presentar el número en binario, el programa en C utiliza estructuras de tipo *union* para alocar espacio contiguo en memoria, de manera que el numero ingresado pueda ser presentado por partes debido a la facilidad de localización de estas. Se crearon dos estructuras llamadas *floatieee* y *doubleieee*, las cuales guardan el número ingresado por el usuario especificando cuántos bits les corresponden a las partes signo, exponente y fracción. Luego de obtener la entrada del usuario dos veces, esta se guarda en ambas estructuras y entonces el programa utiliza la función *aBinario* (Tushar, B.)para imprimir el número en binario por partes. Para la compilación del programa se utilizó el compilador gcc de MinGW para Windows.

En el programa escrito en ensamblador, el proceso consiste en guardar el número de punto flotante tanto en el Registro normal, como en el registro del Coprocesador1, el cual es capaz de almacenar valores de 64 bits, a diferencia del registro que es de 32 bits. Se utilizaron varias llamadas al sistema (*syscall*) para operaciones de entrada y salida (Brock, T.).

**Funcionamiento del programa:**

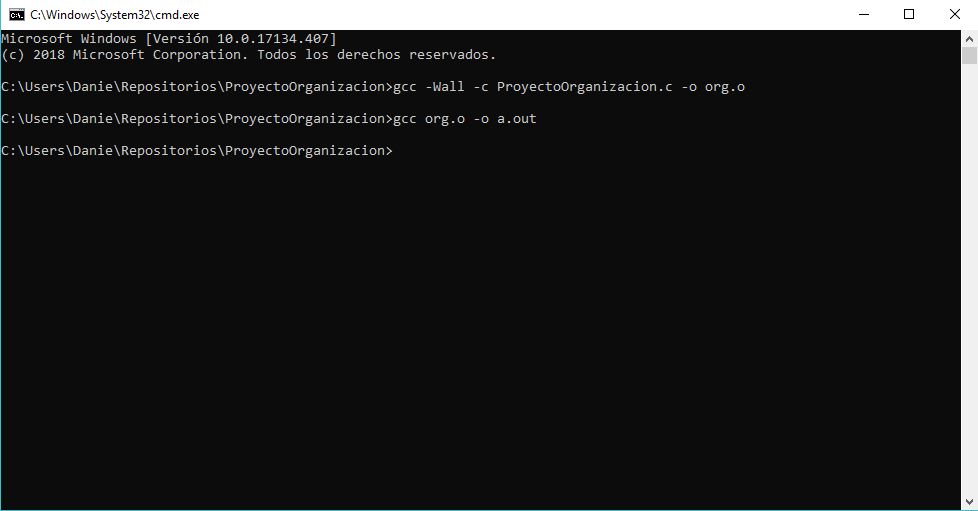
1.El programa solicita un número de parte del usuario, una vez hecho esto

2.Muestra en pantalla su representación en el formato IEEE-754 simple

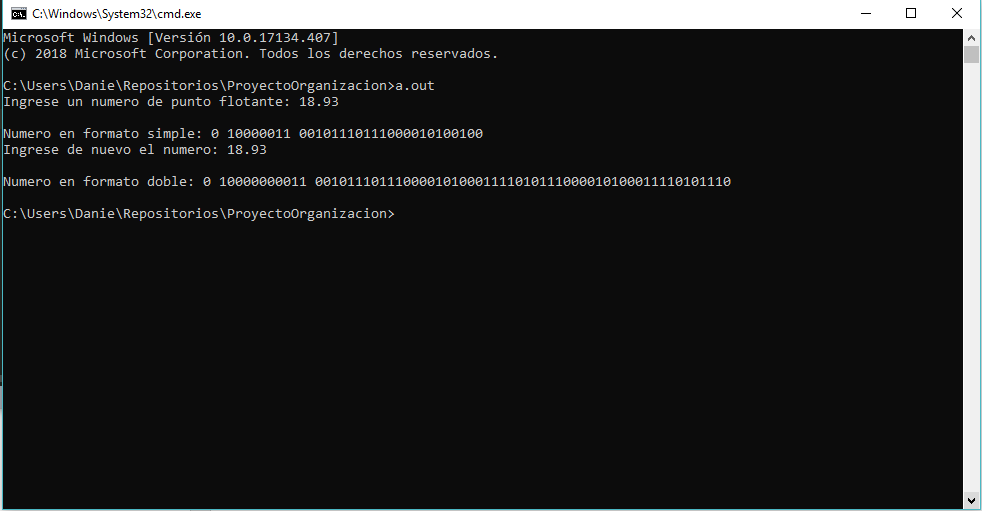
3.Muestra en pantalla su representación en el formato IEEE-754 doble.

En MIPS se procede la transformación usando los registros de punto flotante simple y doble, primero se recibe el valor asumiendo que es un valor de punto flotante simple y debido a la representación de los valores en los registros en MIPS, el número al ser guardado en un registro tiene el formato IEEE-754 simple, por lo que se procede a convertir el valor de formato simple a doble con la operación cvt.d.s y se guardan en los registros del coprocesador 1, seguido de esto, se recorren uno a uno cada bit dentro de un registro y por medio de una máscara de bits se guarda el valor en un registro de valores enteros, y se guarda en un buffer, el cual muestra los resultados en bloques de 32 bits

*Programa en C:*

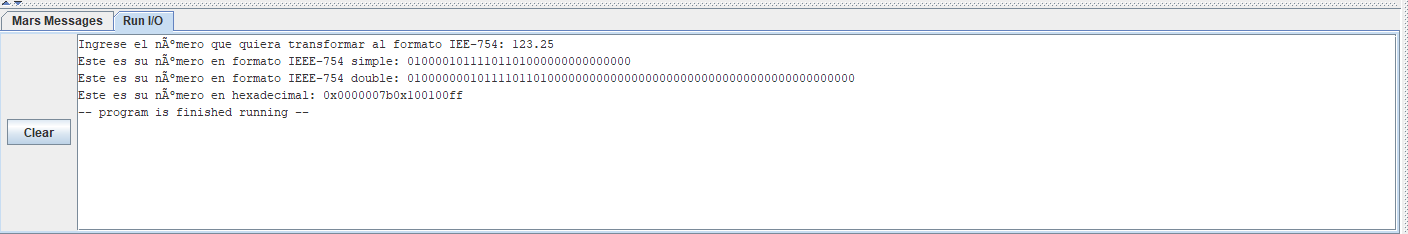


*Imagen 1: Compilación del programa en consola con el compilador gcc de MinGW*

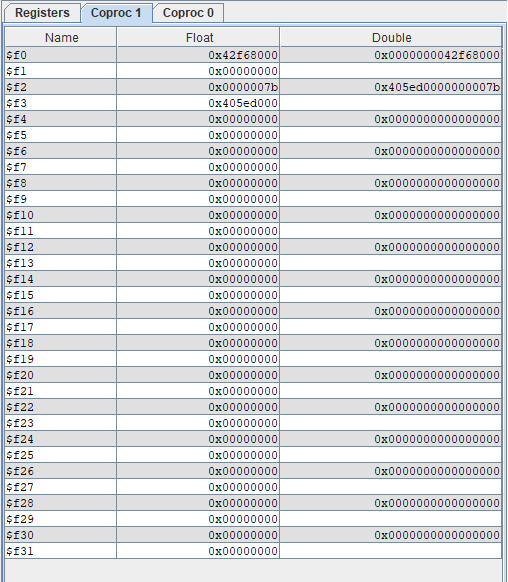
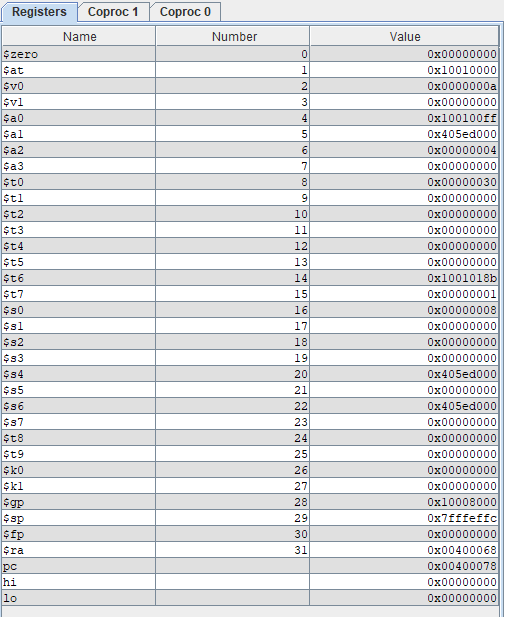


*Imagen 2: Funcionamiento del programa en C, luego de haberlo compilado.*

*Programa en MIPS:*



*Imagen 3: Salida del programa en MIPS ingresando el número 123.25*



*Imagen 4: Registro y Coprocesador 1 para el ejemplo dado en la imagen 3*

**Conclusión:**

Aunque el diseño presenta desventajas en cuando a interacción usuario-sistema (como la solicitud de número dos veces o el formato de presentación de la información), es capaz de mostrar la conversión correcta del número de punto flotante en binario IEEE 754.

**Referencias:**

* Tushar, B. (2013). *C Program to find the floating point IEEE 754 representation* [Artículo]. Disponible: <http://itsitrc.blogspot.com/2013/09/c-program-to-find-floating-point-ieee.html>
* Koppelman, D. (2014). *MIPS Floating Point. Division of Electrical and Computer Engineering*, Louisiana State University, Baton Rouge, LA, Estados Unidos. Disponible: <http://www.ece.lsu.edu/ee4720/2014/lfp.s.html>
* Amell [Quasar Distant]. (25 dic. 2014). *MIPS Tutorial 7 Printing a Double* [Video]. Disponible: <https://www.youtube.com/watch?v=HpxHhCM_gP0>
* Brock, T. (2007). *SYSCALL functions available in MARS*. Missouri State University, Missouri, Estados Unidos. Disponible: <http://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html>