

GUÍA UGK PARA LA ARITMÉTICA DE COMA FIJA.

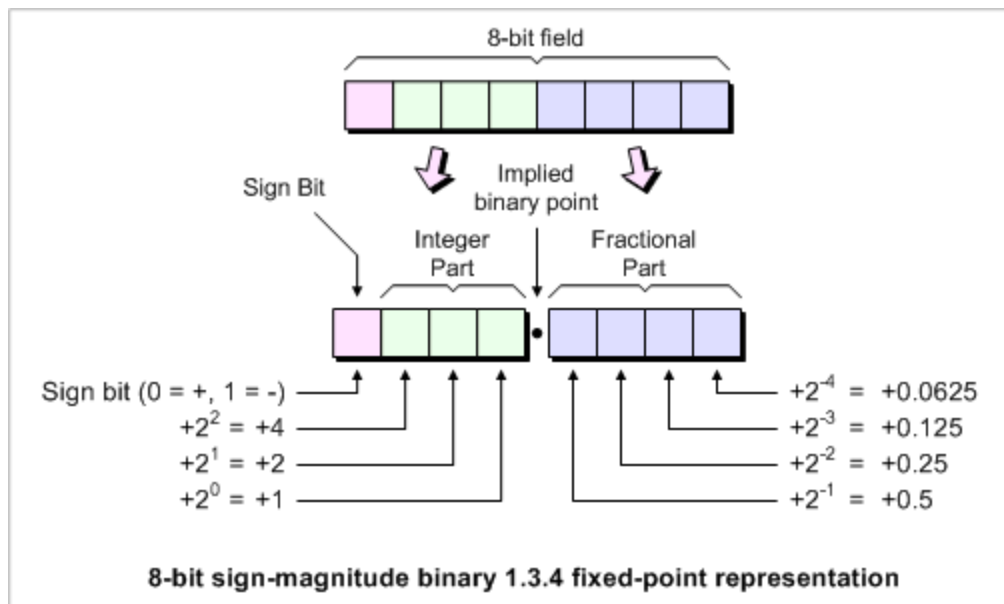
TODO LO QUE SIEMPRE QUISISTE SABER SOBRE BITS,
MANTISAS Y PORQUÉ EL SANDBOX NO COMPILABA

Fantonio & Slovak

Última Actualización de la guía :25/02/2017

1. ¿Qué es la coma fija?

La coma fija es una manera de representar números (notación científica) que se caracteriza por usar un número FIJO de bits para la parte entera y otro número también fijo para la parte decimal. Por este motivo su rango de representación está limitado al número de bits destinados a cada parte. También hay que tener en cuenta que se suele reservar un bit para el signo de los números.



2. ¿Por qué se usa la coma fija? Ventajas e inconvenientes

Para según que aplicaciones puede ser muy útil el uso de la coma fija. Nos referimos a aplicaciones y ámbitos de uso en las que sabemos que no nos saldremos nunca del rango de representación. Para entenderlo explicamos antes una serie de ventajas e inconvenientes:

Inconvenientes

- Problemas de *overflow* que podemos experimentar al tratar representar un número que necesita más bits de los que tenemos.
- Problemas de precisión. Hay operaciones cuyos decimales pueden ser muy largos, al tener un número de bits limitado para la parte decimal hay que poner ciertas limitaciones que suelen traducirse en pérdida de precisión.

Ventajas

- Saber que en todo momento se tiene un número de bits fijo para cada parte permite operar de forma más rápida para según que operaciones. A veces se puede operar por separado con cada una de las partes y después unir el resultado de cierta manera.

Por estos motivos y dependiendo del ámbito puede ser mucho mejor usar la aritmética de coma fija que la de coma flotante. Podemos obtener una mejora en el tiempo de ejecución a cambio de perder un poco de precisión (en muchas ocasiones no es perceptible) y de limitar nuestro rango de representación.

3. Coma fija en el UGK

En el UGK hay dos soluciones que implementan y validan las clases de coma fija:

- **UGKFPA:** Es la solución de Visual Studio donde se encuentran las diferentes implementaciones de coma fija. Existen principalmente cuatro implementaciones generales para diferentes usos (FP0, FP1, FP2, FP3) cada una con un número diferente de bits para cada parte.

	FP0	FP1	FP2	FP3
Bits de signo	1	1	1	1
Bits parte entera	0	7	15	23
Bits parte decimal	31	24	16	8

- **SandBoxFPA:** Es una caja de arena en la que se prueban las implementaciones de coma fija. Se implementó la carga de pruebas desde archivos. **Se recomienda no asumir que el código del Sandbox está bien.** Se han implementado pruebas para medir la coma fija de 32 y 64 bits con respecto a la coma flotante. Teniendo en cuenta precisión y tiempo a lo largo de diferentes operaciones (+, -, *, /, ^, log, sin, cos)

4. Estabilidad del código y trabajo futuro.

Como se ha mencionado previamente los experimentos realizados en la fecha en la que se escribe esta guía no han sido muy esperanzadores. Hay problemas en la implementación de la coma fija y no es precisamente cómoda de usar. En ocasiones se necesita pre-proceso para poder operar con ciertos números. Y hay un fallo de representación que se hace especialmente notable en FPo. Cuando se le pasa un decimal muy pequeño e.g: "0.00000005" creemos que lo interpreta como .0.5 por los resultados obtenidos.

Posible trabajo futuro:

- Arreglar algunos fallos coma fija del UGK.
- Controlar mejor posibles excepciones para evitar problemas de overflow.
- Repetir las pruebas que hemos hecho aprovechando la carga de archivos que hemos implementado.
- Comprobar y testear las clases auxiliares de coma fija.