



Laboratorio Electrónica Digital II

REPORTE PRACTICA 2 Calculadora de Punto Flotante

Integrantes:

Juan Carlos Maya Sánchez – C.C. 1017131966

Maria José Guerrero Muriel – C.C. 1152457472

Profesor:

Ricardo Andrés Velásquez Vélez

Universidad de Antioquia

Facultada de Ingeniería

Departamento de Ingeniería Electrónica

Medellín, Colombia

2018

La práctica se divide en cuatro partes: Unidad Aritmética (ALU), driver teclado matricial 4x4, sistema de visualización en VGA y finalmente, la calculadora completa

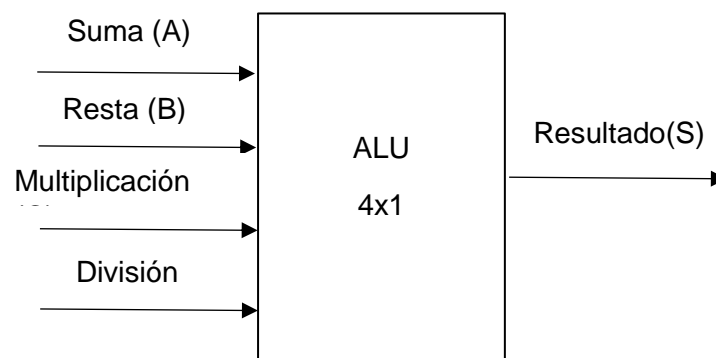
Primera parte: ALU

El formato de punto flotante elegido fue F-28-10 con 17 bits de mantisa y 10 bits de exponente.

Además, se tienen los siguientes casos:

Clase	Exp	Mantisa
Ceros	0	0
Números normalizados	1-511	cualquiera
Infinitos	512	0
NaN(not a number)	512	Diferente de 0

Inicialmente se tiene un mux (4x1) con el cual se seleccionará la operación que se desea realizar



Para realizar cada operación se tiene la siguiente lógica

Suma y Resta:

Para realizar la suma, se ingresan los valores, se dividen entre signo, exponente y mantisa y se realizan los pasos ilustrados en el siguiente diagrama de bloque.

Se verifican si las operaciones de los números dan como resultado alguno de los casos especiales especificados en la tabla al comienzo, luego se verifica la diferencia de exponentes. Si esta diferencia es mayor a los bits de mantisa, el resultado es el de mayor argumento. Posteriormente se igualan exponentes en caso de que difieran, el menor se iguala al número de bits del exponente mayor y se debe desplazar la mantisa del argumento con menor exponente a la derecha. Luego de esto se debe completar, si es necesario, los argumentos con diferentes signos y la mantisa del exponente menor. Después de esto se suman las mantisas y se obtiene de allí la mantisa resultante. Finalmente, se normaliza el resultado desplazando la mantisa de ser necesario y aumentando el exponente si el valor de la mantisa es mayor o igual a 2 y desplazando a la izquierda la mantisa y disminuyendo el exponente si la mantisa es menor a 1.

Para la resta el procedimiento es el mismo, solo que al momento de realizar la xor, la lógica cambia. Cuando el resultado del signo total es 1 en la resta, se realizan los pasos cuando la xor en la suma da 0 y de igual forma para el caso contrario.

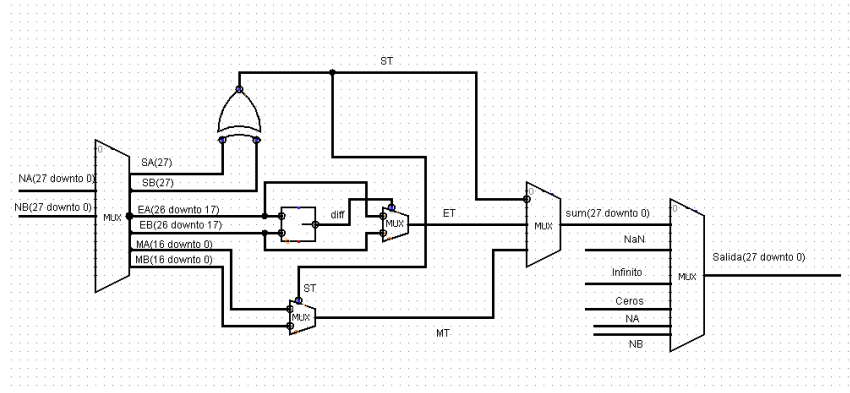


Ilustración 1. diagrama de bloques Suma y Resta

Multiplicación:

En la multiplicación se debe tener en cuenta el número de Bias que para este caso es de 511. Se ingresan los números y se dividen en signo, Exponente y mantisa. El signo indicará si el número resultante es negativo (xor=1) o si es positivo(xor=0). Los exponentes se suman y al resultado se le resta el bias y finalmente las mantisas se multiplican y luego se normaliza (si es necesario). El procedimiento se ve ilustrado en el diagrama de bloques siguiente:

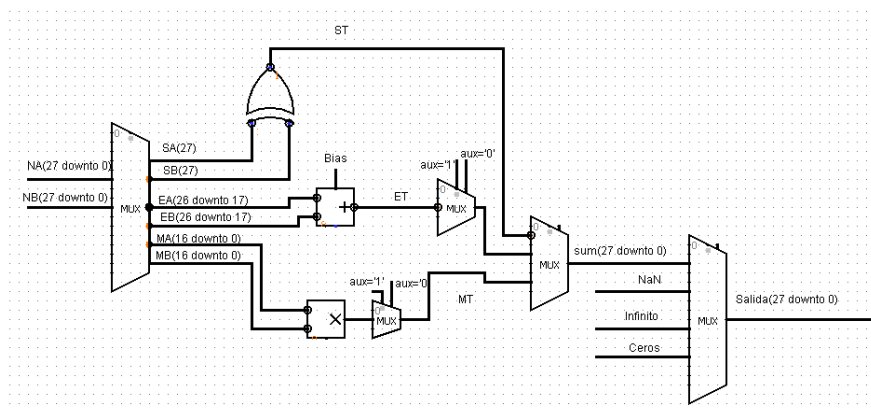


Ilustración 2. Diagrama de bloques de multiplicación

División:

La división es un procedimiento similar al de la multiplicación. El signo define si el resultado es positivo o negativo, para los exponentes, si la mantisa A es mayor o igual a la de B se restan los exponentes y al resultado se le adiciona el bias. Caso contrario se restan exponentes y al resultado se le adiciona Bias-1. Las mantisas se dividen y el resultado se normaliza con corrimiento de bits. El procedimiento se muestra en la ilustración 3.

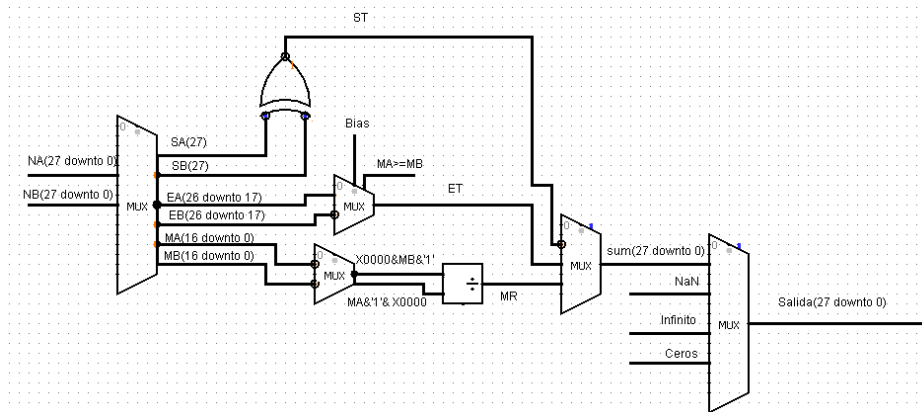


Ilustración 3. Diagrama de bloques División

Segunda parte: Driver teclado matricial

Para el teclado matricial primero se buscó la forma de recibir un dato cuando se digitaba algún valor, posteriormente este valor debía ser almacenado para armar los números con los cuales va a funcionar la calculadora. Para esto, se realizó un pulso de 1 solo flanco de subida y uno de bajada con el fin de que este fuera quien detectara la acción de la tecla y a partir de este pulso poder realizar una serie de acciones que conllevaban al almacenamiento de cada valor. Para comprobar la concordancia de los valores y que solo se recibiera un solo pulso (anti rebote) se usaron los leds integrados en la FPGA y posteriormente la visualización por VGA

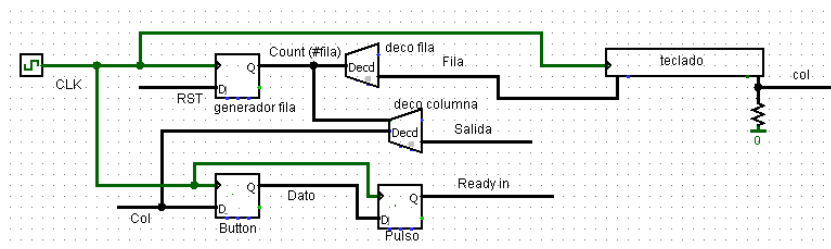


Ilustración 4. diagrama de bloques ingreso teclado

Tercera parte: sistema de visualización en VGA

Para el sistema de visualización se utilizó un módulo VGA brindado por el docente de laboratorio que posteriormente fue modificado para las condiciones que se necesitaban: visualización de dos números con parte entera de 8 dígitos y parte decimal de 4, la operación que se desea realizar y finalmente el resultado que queríamos obtener.

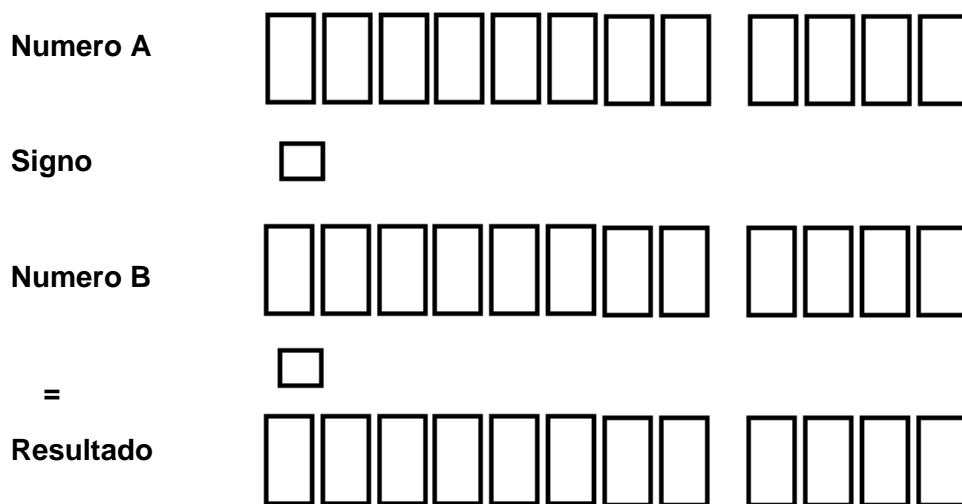


Ilustración 5. sistema de visualización

Para esto se instanciaron displays de 7 segmentos.

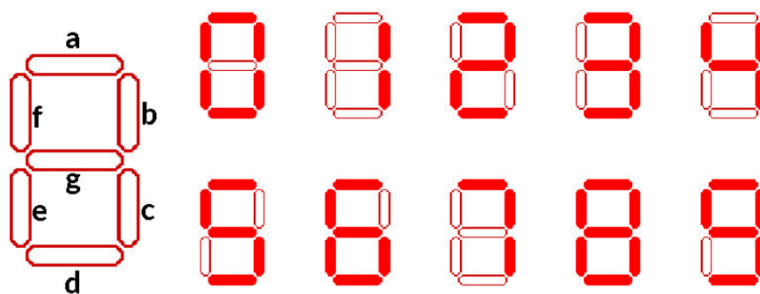


Ilustración 6. números en display 7 segmentos

Cuarta parte: Calculadora completa.

El funcionamiento de la calculadora completa sigue la lógica ilustrada en el siguiente diagrama de bloques. Todos estos procesos explicados anteriormente.

Adicionalmente, se realizó el procedimiento para la conversión de datos de decimal a punto flotante para la realización de los cálculos en la ALU y se convirtió de punto flotante a decimal para mostrar el resultado de la operación en pantalla.

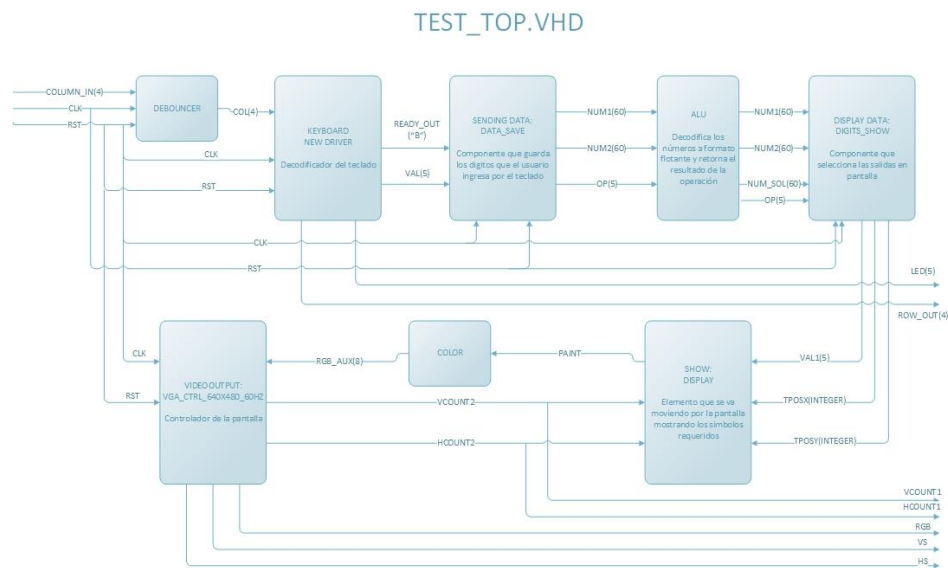


Ilustración 7. diagrama calculadora completa

Se realizó la simulación de la ALU para los números NA = 10.5 y NB = 5.75

Como resultado se obtuvo y se comprobó el buen funcionamiento de está arrojando los siguientes resultados:

NA+NB = 16.25 en decimal

F-28-10 = 0|1000000011|0000010000000000000000000000 en el formato que fue elegido.

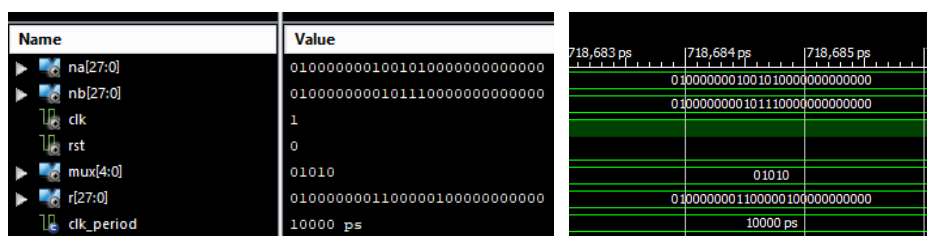


Ilustración 8. Resultado simulación suma

NA-NB=4,75 en decimal

F-28-10 = 0|1000000001|001100000000000000000000000000 en el formato que fue elegido.

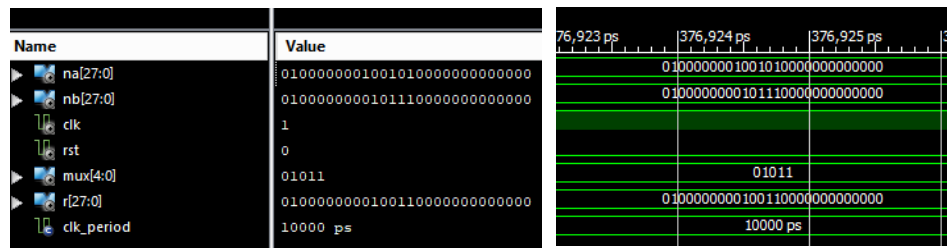


Ilustración 9. resultado simulación resta

NA*NB = 60.375 en decimal

F-28-10 = 0|1000000100|111000110000000000000000000000 en el formato que fue elegido

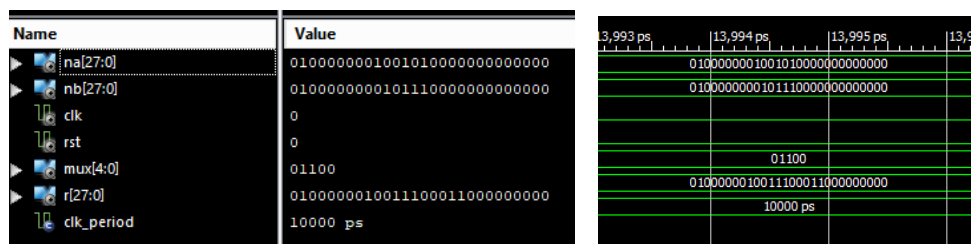


Ilustración 10. Simulación multiplicación

NA / NB = 1.82609 (aproximado) en decimal

F-28-10 = 0|1000000100|111000110000000000000000000000 en el formato que fue elegido

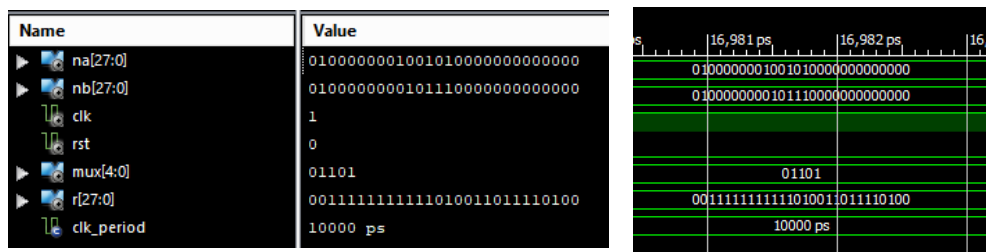


Ilustración 11. simulación división

Análisis

A nivel de implementación se usó un porcentaje bajo de las componentes de la FPGA. 3% en registros, 37 flip-flops, 12% de Slices, 10% de LUTs y 8% de BUFGMUXs.

Gracias a la implementación de solo tres displays (generando varios segmentos en las tres filas) para la visualización, se logró ahorrar un gran porcentaje de FPGA permitiendo así, implementar todo lo que la calculadora requería. Además, se ahorró tiempo en la implementación, pues el generar muchos displays y dependiendo del procesador en el que

se está trabajando (casi siempre el de los computadores del laboratorio) requiere de varios minutos para cargar en la Nexys 2.

Esta práctica permite aclarar toda la teoría acerca de las operaciones en punto flotante en un formato asignado, además, permite continuar con la ampliación de conocimientos en el lenguaje VHDL y todo lo que conlleva trabajar con un reloj de por medio y tener que brindar sincronía a todos los módulos trabajos.

La simulación de la ALU permite ver de manera previa el funcionamiento de esta, pues para el funcionamiento real requiere de codificadores y decodificadores de Decimal a binario teniendo en cuenta también, que al tener una parte entera y otra fracción hay que saber asignar el peso de cada número para poder tener una correcta interpretación de él.

En esta práctica se entrega la ALU, el driver de el teclado, Visualización por VGA y la conversión de decimal a binario y a punto flotante (ingreso de valores con sus respectivos pesos). No se alcanzó a implementar la calculadora completa.

Bibliografía

- Di Lella Daniel. Revista mercado electrónico *“Matemática punto flotante”*
- Minchola Guardia Carlos Eduardo. *“Diseño e implementación de operaciones aritméticas en punto flotante según el estándar IEEE 754-2008 ”*
https://repositorio.uam.es/bitstream/handle/10486/665066/Minchola_Guardia_CarlosEduardo_tfm.pdf?sequence=1&isAllowed=y