INFORME DE LABORATORIO 1

Autores: *María Del Mar Arbeláez Sandoval, Julián Mauricio Sánchez Ceballos*

Laboratorio de Electrónica Digital 2
Departamento de Ingeniería Electrónica y de Telecomunicaciones
Universidad de Antioquia

Resumen

Esta práctica consiste en el diseño e implementación de un sistema electrónico digital para la suma de números de punto flotante, esto, implementando el formato de precisión simple del estándar IEEE-754 que consiste en una representación de 32 bits, 1 de signo, 8 de exponente y 23 de fracción, esta implementación se realiza haciendo uso del lenguaje de descripción de hardware System Verilog y herramientas de desarrollo para FPGAs. El diseño de la práctica se realiza mediante la creación de dos unidades, datapath y control: datapath es la unidad encargada de obtener los datos, procesarlos, y mostrarlos, teniendo algunos módulos extra como el modulo sumador, por otra parte, la unidad de control consiste en una simple máquina de estados que le indica al datapath que tarea realizar y cuando realizarla.

Palabras clave: System Verilog, FPGA, Suma, punto flotante, IEE-754.

Introducción

En esta práctica de laboratorio, se pide implementar un sistema electrónico digital para la suma de números decimales, empleando el formato de precisión simple, el cual consta de 32 bits. El hardware del circuito es descrito mediante dos bloques principales, control y datapath. El datapath será la unidad encargada de manipular los datos de entrada, realizar la suma y llevar el resultado a displays de 7 segmentos de manera apropiada, por otro lado. La unidad de control coordinará el funcionamiento del datapath. Cada una de las unidades tiene otras sub-unidades encargadas de funciones específicas, como la suma, el despliegue del resultado en displays, la decodificación de los resultados, etc. La descripción del sumador y los otros módulos necesarios se realiza mediante el lenguaje de descripción System Verilog.

Diseño

El diseño de este sumador parte de entender gráficamente lo requerido, para eso los estudiantes se apoyan del siguiente diagrama, el cual muestra los periféricos de cada unidad y las unidades que determinan el funcionamiento del sistema:

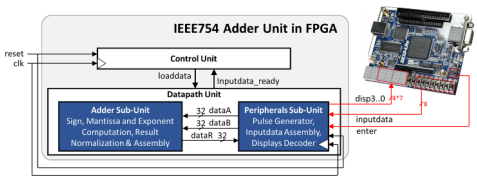

Figura **0-1**: Diagrama de bloques para el sumador del estándar IEEE-754.

Es posible observar las dos unidades principales de las cuales ya se había comentado previamente (datapath y control) y que determina las entradas y salidas del sistema.

Control Unit

La unidad de control es una maquina de estados encargada de coordinar el funcionamiento del datapath, esto lo realiza mediante una máquina de estados que tiene dos estados, como se verá en el código estos dos estados son: s_loaddata y s_sum.

Después de una señal de reset, el estado s_loaddata le indicará al datapath que se deben cargar los valores de A y B, mediante un contador datapath le indicará a la unidad de control que ya se ingresó los valores y procederá al estado s_sum que le indica a datapath que haga una suma.

Datapath Unit

La unidad de datapath se encargará de tomar los datos A y B desde los switchs del sistema FPGA, después realizará la suma y finalmente mostrará el resultado en los displays de 7 segmentos, para estas funciones se definen dos bloques principales: Adder y peripherals

Adder Sub-Unit:

Es un modulo combinacional que se encarga de realizarla suma entre A y B y generar el resultado R en el formato de precisión simple del estándar IEEE-754. En este modulo también se exponen los casos especiales:

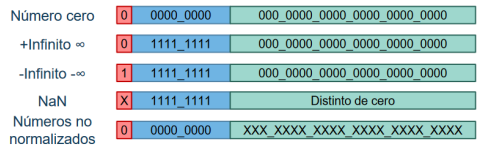

Figura **0-2**: Casos especiales para el formato IEEE-754.

La suma de estos casos especiales son tratados diferentes en el modulo de suma, se comparan las situaciones para saber, en caso de tener alguno de estos casos poder dar una solución correcta. El modulo de suma de dos números normales funciona siguiendo los pasos expuestos en clase para la suma de dos números en formato IEEE-754:

- Extraer los exponentes y fracciones de A y B
- Agregar un 1 a la parte mas significativa de las fracciones de A y B para formar las mantisas.
- Restar ambos exponentes y usar la diferencia para desplazar la mantisa del valor más pequeño.
- Sumar las mantisas, dependiendo del signo, la suma será un numero negativo o no.
- Normalizar la mantisa resultante y ajustar el exponente, si es necesario.
- Ensamblar el exponente y fracción para tomar el número en punto flotante resultante.

Peripherals Sub-Unit:

Encargada de cargar los datos A y B y mostrar los resultados, dado que la FPGA no tiene una pantalla con 32 displays de 7 segmentos los operandos se obtienen como entradas de 4 grupos de 8 bits en hexadecimal, este funcionamiento se mostrará más adelante en la simulación. Esta unidad también está encargada de mostrar el resultado R, lo hará en el mismo modo en que se carga la información, en 4 grupos de 8 bits que cambiaran pulsando el botón de enter.

Todos los módulos y submódulos previamente descritos anteriormente se unen en un archivo .sv llamado top, que funciona como acople de todos los módulos, y es aquí donde se diseña el test bench para la simulación.

Simulación

La simulación se diseña de modo que los datos se ingresen en el mismo orden de como se pide en la practica, en 4 grupos de 8 bits. En primer lugar se inician las variables internas de la simulación y se instancia el modulo TOP que contiene la unión del datapath y de la unidad control:

```
logic clk = 0;
logic reset = 1'b0;
logic enter = 1'b0;
logic [7:0] inputdata;
logic [6:0] disp3, disp2, disp1, disp0;
localparam delay = 100ps;

// Instantiate the top module
top tb(clk, ~reset, ~enter, inputdata, disp3, disp2, disp1, disp0);
```

Figura **0-3**: Instancia del modulo TOP y definición de variables.

La forma de ingresar los datos en la simulación se describe de la siguiente manera:

```
inputdata = 8'hC1;
enter = 1'b1;
#((delay)*2)
enter = 1'b0;
#((delay)*2)

//Dato A2
inputdata = 8'h90;
enter = 1'b1;
#((delay)*2)
enter = 1'b0;
#((delay)*2)

//Dato A1
inputdata = 8'h00;
enter = 1'b1;
#((delay)*2)
enter = 1'b0;
#((delay)*2)

//Dato A0
inputdata = 8'h00;
enter = 1'b1;
#((delay)*2)
enter = 1'b0;
#((delay)*2)
```

Figura **0-4**: Descripción de la entrada A en la simulación.

Ingresar el dato B se realiza de manera análoga a como se muestra:

```
//Dato B3
inputdata = 8'h41;
enter = 1'b1;
#((delay)*2)
enter = 1'b0;
#((delay)*2)

//Dato B2
inputdata = 8'h1.8;
enter = 1'b1;
#((delay)*2)
enter = 1'b0;
#((delay)*2)

//Dato B1
inputdata = 8'h00;
enter = 1'b1;
#((delay)*2)
enter = 1'b0;
#((delay)*2)

//Dato B0
inputdata = 8'h00;
enter = 1'b1;
#((delay)*2)
enter = 1'b0;
#((delay)*2)
```

Figura **0-5**: Descripción de la entrada B en la simulación.

En el resultado de la simulación se puede ver la actualización de cada dato a medida de que se ingresan los grupos de 8 bits y se presiona el botón de enter. Los números ingresados son: A = 18(0xC1900000) y B = 9.5(0x41180000), la cual tiene como resultado R = 8.5(0xC1080000) y se muestra a continuación:

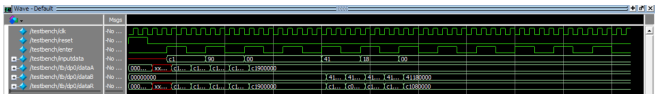

Figura **0-6**: Resultados de la simulación.

Conclusiones

- Dentro de la implementación del sumador de números flotantes hay unos casos donde es importante tener en cuenta procedimientos extras o casos especiales donde el resultado ya es conocido, ejemplo de ello puede ser una suma de un NaN con un numero cualquiera, lo cual será un NaN. Dentro de los problemas durante la implementación se pudo observar que no tener en cuanta la normalización de los números al formato IEEE-754 ocasiona un mal funcionamiento del modulo sumador, arrojando resultados erróneos, la forma de resolver este problema es mediante una confirmación de normalización del numero ingresado y de una posterior normalización en caso de que se haya ingresado un número no normalizado.

- En la implementación fue posible observar una desventaja en el uso de este formato para la representación de números decimales en punto flotante, este problema tiene que ver con la precisión que se pierde en casos como el representar números que estén cerca del limite mínimo y máximo, pero también se puede presentar esta perdida de precisión cuando hay que normalizar la mantisa y ajustar el exponente, en algunos casos ensayados con la tarjeta DE10-Lite se pudo observar este problema, el cual también se puede describir como una suma de un bit en el bit menos significativo de la mantisa. Esto bajo la interpretación de los integrantes del grupo no significa un resultado incorrecto en el sumador, pero si una pérdida de precisión en los resultados.