

Diseño e implementación de una ALU capaz de resolver operaciones en punto flotante (IEEE-754, 32 bits y 16 bits) en FPGA Basys3

1. Resumen

Deben diseñar, implementar y verificar una Unidad Aritmético-Lógica (ALU) con soporte para números en punto flotante conforme al estándar **IEEE-754 (single precision, 32 bits y half precision, 16 bits)**. El proyecto exige la ejecución correcta de operaciones aritméticas básicas y el manejo de los casos especiales definidos por el estándar (NaN, $\pm\text{Inf}$, denormales, distintos modos de redondeo y excepciones).

2. Objetivos generales

- Diseñar una ALU en **HDL (Verilog o VHDL)** que implemente operaciones aritméticas en punto flotante de 32 y 16 bits.
- Verificar la funcionalidad mediante testbenches y vectores de prueba.
- Implementar el diseño en la placa **Basys3** y demostrar su funcionamiento.

3. Objetivos específicos

1. Implementar las siguientes operaciones en punto flotante (IEEE-754, single y half):
 - Suma (fadd)
 - Resta (fsub)
 - Multiplicación (fmul)
 - División (fdiv)
2. Manejar modos de redondeo: *round to nearest even* (obligatorio).
3. Detectar y reportar excepciones/flags: overflow, underflow, divide-by-zero, invalid operation, inexact.
4. Soporte correcto para NaN, $\pm\text{Inf}$, ceros con signo y números denormales.

4. Interfaz y especificación funcional

- Entradas principales:
 - op_a[31:0], op_b[31:0] — operandos (IEEE-754, single o half)
 - op_code[2:0] — selecciona la operación (ADD, SUB, MUL, DIV)
 - mode_fp — bit que selecciona el formato: 0 = half (16 bits), 1 = single (32 bits)
 - clk, rst — señales de control
 - round_mode — modo de redondeo (mínimo: nearest)

- start — activa el inicio de la operación
- Salidas principales:
 - result[31:0] — resultado en formato IEEE-754 (si mode_fp = 0, el resultado válido está en los 16 bits menos significativos)
 - valid_out — señal que indica que el resultado ya está disponible y estable. Permite sincronizar el uso de la salida en sistemas secuenciales o pipelined.
 - flags[4:0]

5. Verificación

- Deben incluir testbenches que cubran:
 - Valores aleatorios comparados con referencia.
 - Casos límite explícitos: NaN, $\pm\text{Inf}$, ± 0 , denormales, overflow/underflow.
 - Propiedades básicas como conmutatividad en suma y multiplicación.
- Reporte con estadísticas: número de vectores probados, casos fallidos con traza.

6. Entregables

1. **Código fuente HDL** (estructurado y comentado).
2. **Testbenches** con los casos de prueba descritos.
3. **XDC y archivos de proyecto** para la Basys3.
4. **Informe técnico (PDF)** (min. 10 páginas + anexos): diseño, diagramas, metodología de verificación, resultados en simulación y pruebas en hardware.
5. **Video presentación** (8–12 min) explicación de su diseño, mostrando la simulación y la ejecución en la Basys3.

7. Criterios de evaluación

INFORME + CODIGO (12 pts)

DEADLINE Domingo 12 de Octubre 23:00

Presentar reporte en pdf

Adjuntar una carpeta (.zip) con el proyecto completo, asegúrese de mandar el proyecto completo, no habrán presentaciones posteriores al deadline.

REPORTE (12 pts):

Preliminar: Cual es la lógica de su implementación?

Diseño: Justifique sus desiciones de diseño.

Diagrama del diseño del ALU 2 pts

Control del ALU 2 pts

Cambios en el codigo, comentada 2 pts

Resultados: Comente sus resultados, son los esperados?

Simulation waveform – varios casos 2 pts

(clk, reset, PC, operandos, control, resultado)

Comentarios 2 pts

Implementacion: Como implementó sus resultados?

Implementacion FPGA 2 pts

VIDEO PRESENTACIÓN MAX 12 MIN (8 pts)

DEADLINE Domingo 12 de Octubre 23:00

Presentación orden (1 pts)

FLOATING 32 bits 4 MIN (2 pts)

DISEÑO

CONTROL

SIMULATION

FLOATING 16 bits 4 MIN (2 pts)

DISEÑO

CONTROL

SIMULATION

BASYS IMPLEMENTATION 4 MIN (3 pts)

COMO SE IMPLEMENTÓ?

DEMOSTRACIÓN CON LA BASYS3