Diseño e implementación de ALU

Autor: Díaz Gerardo Agustín

Carrera: Ingeniería Electrónica

Materia: Electrónica II

Año: 2025

Resumen

En esta actividad se diseñó e implementó una Unidad Aritmético-Lógica (ALU) de W bits parametrizable, codificada en VHDL.

La ALU permite ejecutar operaciones aritméticas, lógicas y de desplazamiento según una entrada de selección sel_fn.

A partir de esta ALU, se desarrolló una calculadora de 4 bits implementada en la placa edu-ciaa-fpga, que utiliza interruptores para la carga de operandos y displays de ocho segmentos para mostrar los resultados.

El sistema fue validado mediante simulación y banco de pruebas automático, comprobando el correcto funcionamiento de cada operación y la detección de resultados nulos mediante la señal Z.

Introducción

La Unidad Aritmético-Lógica (ALU) constituye el núcleo funcional de cualquier procesador digital.

Su función principal es realizar operaciones aritméticas como suma, resta y operaciones lógicas AND, OR, XOR, desplazamientos y comparaciones entre dos operandos.

En este trabajo, la ALU se diseñó de forma parametrizable, permitiendo variar su ancho de palabra (W) mediante un parámetro genérico, lo que la hace adaptable a distintos sistemas.

El diseño se basa en principios vistos en actividades anteriores: el uso de flip-flops D para el almacenamiento de resultados y la sincronización con un reloj de 12 MHz.

La ALU desarrollada responde a un selector de funciones sel_fn de 4 bits, según la tabla de la consigna, e incluye operaciones tales como:

- (A + B), (A – B) (aritméticas)
- (A << B), (A >> B) (desplazamientos)
- (A < B), (A XOR B), (A OR B), (A AND B) (comparación y lógicas)

Cada operación produce un resultado Y (de W bits) y una bandera Z (bit único) que indica si el resultado fue igual a cero.

En la segunda parte de la actividad, la ALU se integró en una calculadora de 4 bits.

El sistema dispone de dos registros (A y B), cargados por interruptores, y un selector de funciones sel_fn que determina la operación a ejecutar.

El resultado se visualiza en un display de ocho segmentos en formato hexadecimal, y el punto decimal del display refleja el estado de la bandera Z.

Este diseño permite comprender cómo una ALU se puede integrar dentro de un sistema digital mayor.

Desarrollo

El diseño realizado consiste en una Unidad Aritmético-Lógica (ALU) completamente combinacional y parametrizable en el número de bits (W).

Esto significa que puede adaptarse fácilmente para operar con diferentes tamaños de palabra, por ejemplo, 4, 8 o 16 bits modificando un solo valor genérico en el código.

Estructura general

El sistema está compuesto por tres bloques conceptuales principales:

1. **Bloque de entradas:**

Recibe los dos operandos (A y B) y el código de selección de función (sel_fn), el cual define la operación a realizar.

A y B son vectores binarios de W bits, y sel_fn es un código de 4 bits que permite elegir entre distintas operaciones aritméticas y lógicas.

2. **Bloque de operaciones internas (núcleo de la ALU):**

Aquí se interpretan las señales de control y se ejecuta la operación correspondiente.

El diseño distingue entre operaciones con signo (en complemento a 2) y sin signo, permitiendo realizar correctamente comparaciones aritméticas según el tipo de dato.

De este modo, la ALU puede ejecutar operaciones de suma, resta, desplazamientos lógicos a izquierda o derecha, comparaciones con y sin signo, y operaciones lógicas bit a bit como AND, OR, XOR, entre otras.

3. **Bloque de salidas y banderas:**

El resultado de la operación seleccionada se entrega por la salida Y, de W bits para ser mostrada en el display.

Además, se genera una bandera (Z) que vale ‘1’ únicamente cuando el resultado es igual a cero.

- En la Figura 1 se muestra el diseño de ALU que se implementó en VHDL.

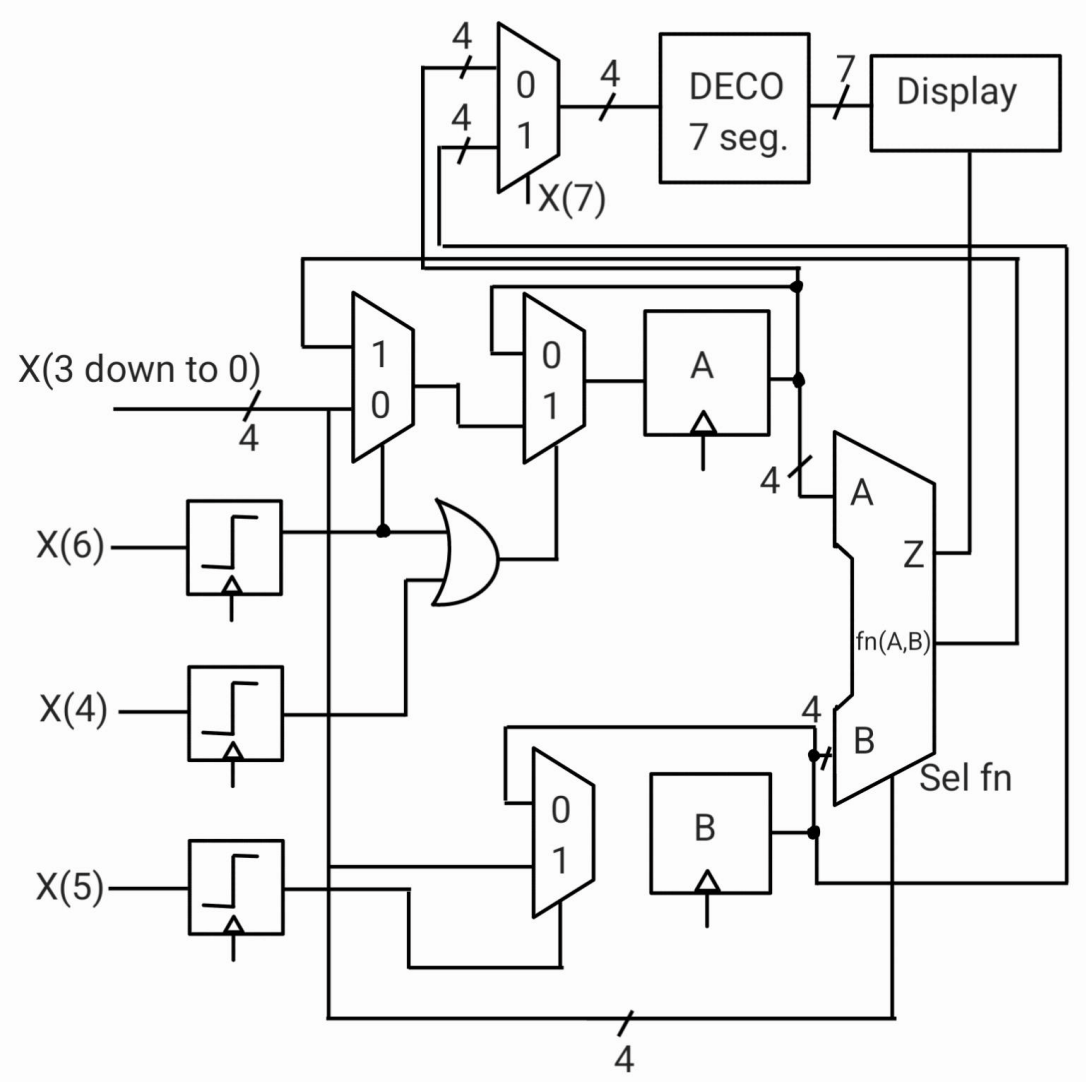

Figura 1: Diagrama de ALU

Funcionamiento general

El funcionamiento de la ALU puede describirse así:

Los operandos A y B se aplican a las entradas.

El selector sel_fn indica qué tipo de operación realizar (por ejemplo, suma, resta, comparación, etc.).

Internamente, la ALU interpreta ese código y ejecuta la operación correspondiente.

El resultado de esa operación aparece en la salida Y.

La señal Z se activa si ese resultado es igual a cero.

Por ejemplo:

Si sel_fn = "0000", la ALU realiza la suma A + B.

Si sel_fn = "0001", ejecuta la resta A − B.

Si sel_fn = "0110", compara A y B como valores sin signo y entrega 1 si A < B, o 0 en caso contrario.

Si sel_fn = "1101", aplica la operación lógica OR bit a bit entre A y B.

Esto convierte a la ALU en un módulo versátil, capaz de cubrir las funciones más comunes de un procesador digital, actuando como el núcleo de cálculo de un sistema mayor.

Comparaciones con y sin signo

Otro aspecto importante es que el diseño puede comparar números considerando el signo (complemento a 2) o sin considerarlo.

Esto es esencial porque en hardware digital, los números binarios pueden interpretarse de las dos maneras:

Como valores positivos sin signo (naturales), o como valores enteros con signo (positivos y negativos).

De esta forma, la ALU puede decidir correctamente si un número "es menor" que otro en función del tipo de dato, reproduciendo el comportamiento de comparadores aritméticos reales.

Operaciones lógicas y aritméticas

La ALU implementa las operaciones básicas utilizadas en casi todos los microprocesadores:

Aritméticas: Suma (A + B) y resta (A - B).

Desplazamientos: Lógicos a la izquierda o derecha, según el valor de B.

Comparaciones: Menor que (A < B) con y sin signo.

Lógicas: AND, OR, XOR bit a bit.

Bandera de resultado nulo

La salida Z funciona como un indicador de resultado cero.

Su utilidad es permitir que otras partes del sistema detecten si el resultado de la operación fue igual a 0.

Por ejemplo, si A y B son iguales y se realiza la resta A - B, entonces Z se activa.

En una aplicación práctica, esta bandera puede encender un LED o activar el punto decimal de un display para señalar que el resultado es cero.

Integración

La ALU se integró más tarde en un sistema de calculadora digital, donde sus entradas se controlan mediante interruptores físicos (switches) y su resultado se muestra en un display de siete segmentos.

Resultados

Para comprobar el funcionamiento de la ALU, se utilizó un testbench automatizado que lee un archivo con diferentes casos de prueba (alu_tb_datos.txt).

Cada línea del archivo contiene los valores de entrada (A, B, y sel_fn) y los resultados esperados (Y y Z).

El programa aplica estos valores a la ALU, espera unos nanosegundos y luego compara automáticamente las salidas reales con las esperadas.

Durante la simulación, el testbench indicó si las operaciones fueron correctas: sumas, restas, desplazamientos, comparaciones y operaciones lógicas (AND, OR, XOR).

La bandera Z también funcionó, indicando solo cuando el resultado fue cero.

Al final, se mostró un mensaje con el número de pruebas procesadas y ninguna diferencia detectada entre los resultados simulados y los esperados.

Además de la simulación, el código VHDL de la ALU fue cargado y probado en la placa FPGA EDU-CIAA.

Mediante los interruptores se ingresaron los operandos y el selector de operación, y el resultado se visualizó en el display de ocho segmentos.

Las operaciones aritméticas y lógicas funcionaron correctamente en el hardware, confirmando que el diseño sintetizado conserva el mismo comportamiento observado en la simulación.

Conclusiones

La implementación de una Unidad Aritmético-Lógica (ALU) permite concentrar en un solo bloque todas las operaciones básicas que un sistema digital necesita para procesar información.

Su principal ventaja es que simplifica el diseño de circuitos complejos, ya que puede reutilizarse como parte central de procesadores, calculadoras o controladores lógicos.

Durante las pruebas tanto en simulación como en la placa FPGA EDU-CIAA, la ALU funcionó correctamente y entregó los resultados esperados, confirmando la validez del diseño y su comportamiento.

Referencias

- Harris, S. L., & Harris, D. (2019). Digital design and computer architecture: RISC-V edition. Morgan Kaufmann.