

# DISEÑO Y ESTUDIO TEMPORAL DE UNA UNIDAD ARITMÉTICO LÓGICA

## PRÁCTICA 4



Introducción a los Computadores  
1º Grado Ingeniería Informática

Natalia Ayuso   Luis M. Ramos   Juan Segarra   Víctor Viñals  
nayuso@unizar.es   luisma@unizar.es   jsegarra@unizar.es   victor@unizar.es



Escuela de  
Ingeniería y Arquitectura  
**Universidad Zaragoza**



Departamento de  
Informática e Ingeniería  
de Sistemas  
**Universidad Zaragoza**

# 1 RESUMEN

En esta práctica vamos a construir una Unidad Aritmético-Lógica (ALU) de 16 bits con Logisim. Además del resultado, la ALU calculará también varios indicadores (flags) sobre el resultado: cero, signo, acarreo y desbordamiento.

El diseño será modular. Esto significa construir módulos sencillos, con funciones específicas, y luego replicarlos y conectarlos para conseguir una función más compleja.

Finalmente realizaremos un estudio temporal de todo el diseño, para estimar la frecuencia máxima de cálculo que podría ofrecer la ALU.

Para poder realizar la práctica hay que enseñar al responsable de la clase de laboratorio el Estudio Previo del Apartado 2.2 RESUELTO.

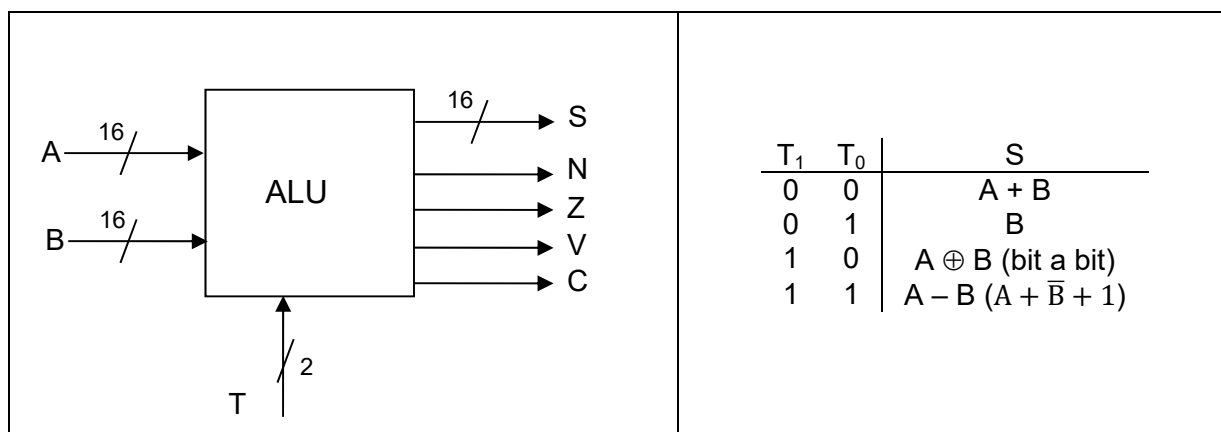
La práctica finaliza cuando los circuitos funcionan correctamente y se ha entregado el resultado de los apartados 2.3 y 3 a través del recurso Moodle correspondiente.

## 2 DISEÑO UNA UNIDAD ARITMÉTICO-LÓGICA (ALU)

### 2.1 ESPECIFICACIÓN DE FUNCIONAMIENTO DE LA ALU

La ALU a diseñar tiene dos vectores de entrada de 16 bits (A y B) y obtiene como resultado otro vector de 16 bits (S). Dependiendo del valor de la entrada de control T, la salida S corresponde a la operación especificada en la tabla.

A = ( $a_{15}, \dots, a_0$ ), B = ( $b_{15}, \dots, b_0$ ), T = ( $t_1, t_0$ ), S = ( $s_{15}, \dots, s_0$ ).



Además, sobre el resultado se calculan los siguientes indicadores:

- flag de signo: indica el signo del resultado
- flag de cero: indica que el resultado es cero
- flag de desbordamiento de enteros:

$$N = s_{15}$$

$$Z = 1 \leftrightarrow S = 0$$

$$V = \overline{a_{15}} \cdot \overline{b'_{15}} \cdot s_{15} + a_{15} \cdot b'_{15} \cdot \overline{s_{15}}$$

Indica resultado incorrecto (no representable)

$$\text{Nota: } b'_{15} = \begin{cases} b_{15} & \text{si } \underline{T} = 00 \\ \overline{b_{15}} & \text{si } \underline{T} = 11 \end{cases}$$

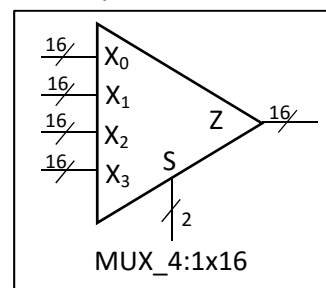
- flag de acarreo: último acarreo del sumador

$$C = c_{16}$$


## 2.2 TRABAJO PREVIO: DISEÑO MODULAR

Antes de llegar al laboratorio diseña sobre papel los siguientes componentes:

- **MUX\_4:1**: multiplexor de 4 entradas. Diseñalo usando puertas AND, OR y NOT.
- **MUX\_4:1x16**: multiplexor de 4 entradas de 16 bits. Usa 16 MUX\_4:1. Es muy importante que tengas claro este bloque combinacional. Dispone de 4 entradas de 16 bits y una salida de 16 bits. Con dos bits de selección se indica cuál de las cuatro entradas de 16 bits se selecciona en la salida de 16 bits.
- **ALU**: unidad aritmético-lógica. Usa el MUX\_4:1x16, un sumador de naturales de 16 bits, puertas XOR y otras puertas para el cálculo de los *flags*.



## 2.3 TRABAJO EN EL LABORATORIO: CONSTRUCCIÓN Y SIMULACIÓN CON LOGISIM

- En una hoja nueva de Logisim añade un componente nuevo (en el menú de componentes *click* derecho en  *Sin Título* y selecciona *Añadir circuito*. Nómbralo MUX4:1. Añade cuatro entradas de un bit x0, x1, x2 y x3. Añade dos entradas de un bit para la selección s1 y s0. Añade una salida de un bit z. Dibuja tu diseño y comprueba su funcionamiento.
- Vuelve a la hoja principal (doble *click* en *main*) y añade otro componente. Nómbralo MUX\_4:1x16. Coloca 16 MUX\_4:1. Coloca las entradas (X0, X1, X2, X3, s1 y s0) y una salida Z. Define X0, X1, X2, X3 y Z como entradas/salidas de 16 bits (en *tabla de propiedades / Bits de Datos*). Utiliza el componente *Separador* (carpeta de componentes *Wiring*) para separar los bits de los buses de entrada. Configura los *Separadores* con *Fanout* 16 y *Bits De Entrada* 16. Conecta los bits obtenidos a los MUX\_4:1. Utiliza otro *Separador* para juntar los bits que salen de los MUX\_4:1 y llevar el bus a la salida Z. Comprueba su funcionamiento.
- Vuelve a la hoja principal y coloca un MUX\_4:1x16. Añade un sumador de naturales (componente *Aritmética / Sumador*) y puertas XOR de 16 *Bits de Datos*. Finalmente añade otras puertas necesarias para el cálculo de los *flags*.
- Comprueba que la ALU funciona correctamente. Puedes utilizar el componente *Ver* para visualizar datos de los buses. Ten en cuenta que se puede cambiar la forma en la que el componente *Ver* interpreta el dato (*tabla de propiedades / Radix*).

## 3 ESTUDIO TEMPORAL

- Realiza un análisis temporal de cada uno de los módulos diseñados en el apartado 2.2, teniendo en cuenta que los retardos de las puertas utilizadas son:  $d_{NOT} = 5ps$ ;  $d_{OR2-4} = 20ps$ ;  $d_{AND2-4} = 20ps$ ;  $d_{NOR16} = 55ps$ ;  $d_{XOR2} = 30ps$  ( $1ps = 10^{-12} s$ ). Representa los retardos de cada módulo con la notación gráfica vista en clase.
- Dibuja el camino crítico de la ALU y calcula su retardo ( $d_{max}$ ). Asume que el sumador está implementado con *Full Adders* de 5 puertas (2 XOR, 2 AND, 1 OR) y propagación de acarreo.
- Calcula la frecuencia máxima de cálculo expresada en número de Giga operaciones/segundo ( $1 G = 10^9$ ).
- Finalmente entrega tu diseño Logisim y responde a las preguntas en el recurso Moodle "Entrega de la práctica 4".