

# Laboratorio de Estructura de Computadores

## Curso 2016-2017

---

### Práctica 3:

### Unidad Aritmético Lógica y Banco de Registros

---

La entrega estará compuesta de un único fichero que incluya:

- Todos los “.vhd” solicitados.
- Un fichero llamado “autores.txt” que contenga los nombres completos e identificadores de la pareja.

Todos estos ficheros deben entregarse en un único fichero comprimido.

Los ejercicios deberán subirse a la actividad correspondiente de Moodle, identificada por el número de práctica (ej. La práctica 2 se entrega en la tarea «Entrega Práctica 2»). Sólo es necesario que suba los ficheros un miembro de la pareja. En caso de que ambos miembros de la pareja suban un fichero a Moodle, se corregirá sólo uno.

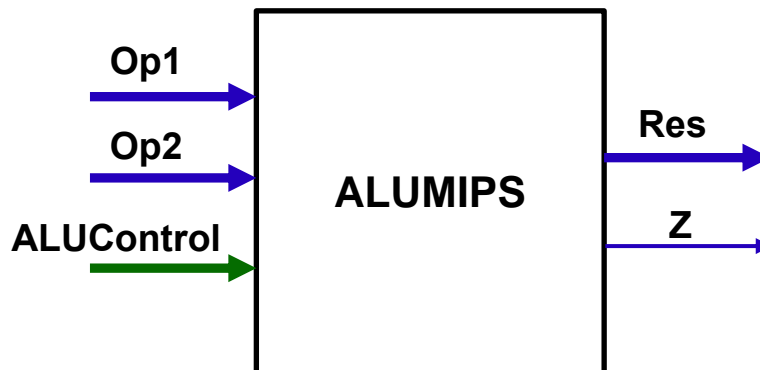
Todos los ficheros deberán contener en la cabecera los nombres de los autores y el identificador de la pareja. Asimismo, se recuerda que los ficheros deberán estar correctamente comentados. Dado que no se entrega una memoria adicional, los comentarios de los ficheros deben ser claros y concisos. No atenerse a estas normas implicará una penalización en la nota.

**El límite de entrega de esta práctica será el día anterior al comienzo de la siguiente práctica. Las parejas que no hayan entregado en plazo y forma serán calificadas con un cero en esta práctica.**

---

#### Ejercicio 1. Diseño de la ALU (8 puntos).

Diseñar una unidad aritmético-lógica combinacional (entidad *AluMIPS*) con dos entradas de datos de 32 bits (señales *OP1* y *OP2*), una entrada de selección de operación de 3 bits (señal *ALUControl*), una salida de datos de 32 bits (*Res*) y una salida de estado de un bit (*Z*). Esta señal de estado corresponde a la bandera de *Z*, la cual se debe activar cuando el resultado sea igual a cero.



Los valores de entrada de selección codifican las siguientes operaciones:

ALUControl[2:0]	Operación
<b>010</b>	OP1 + OP2
<b>110</b>	OP1 - OP2
<b>000</b>	OP1 and OP2
<b>001</b>	OP1 or OP2
<b>111</b>	OP1 slt OP2
<b>101</b>	OP1 nor OP2

La instrucción *slt* pone a 0x1 el resultado cuando el operando 1 es menor que el operando 2. En caso contrario, el resultado será 0x0. Esta instrucción tiene en cuenta el signo de los operandos de entrada que considera en complemento a 2 (p. ej. un número negativo es menor que uno positivo).

La ALU que se entregue deberá implementar exclusivamente las operaciones que se especifican en el enunciado de esta práctica.

Para verificar el diseño correcto de la ALU, se proporciona un *testbench* (“*ALUMIPSTb.vhd*”) que comprueba el correcto funcionamiento de la ALU. El *testbench* indicará si la ALU no realiza correctamente alguna operación.

## Objetivo

Diseñar y verificar mediante un testbench el funcionamiento de una ALU. Esta ALU será la que se utilice en la implementación del micro completo.

## Entrega

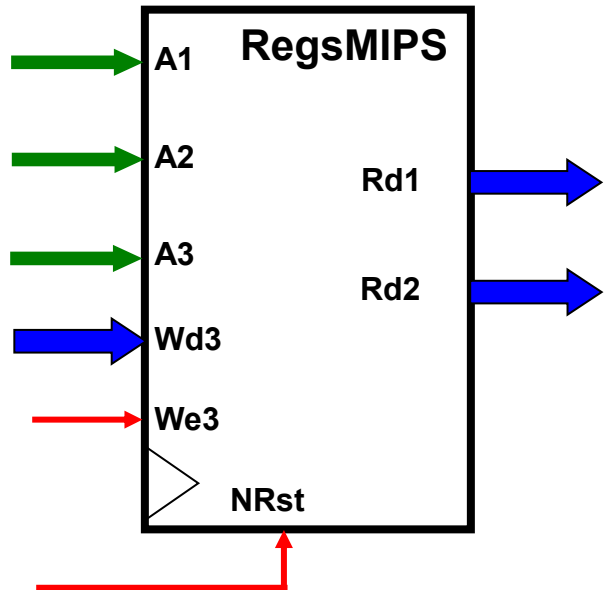
Se debe [entregar el fichero “ALUMIPS.vhd”](#). Este fichero debe seguir las normas de entrega de la asignatura.

## Ejercicio 2. Diseño del banco de registros (2 puntos).

Diseñar en VHDL el banco de registros de propósito general (entidad "RegsMIPS") del microprocesador MIPS, cuya interfaz se muestra en la figura adjunta. Se proporciona el banco de pruebas completo (fichero "RegsMIPSTb.vhd").

Este banco consta de 32 registros de 32 bits, que están identificados del 0 al 31. El registro 0 siempre tiene el valor de 0.

El banco permite la escritura síncrona de un único registro (registro destino). La dirección de este registro viene dada por la señal de 5 bits *A3*, que puede tomar valores entre 0 y 31. El valor que tomará el registro se indica en la entrada de 32 bits *Wd3*. La escritura se hace efectiva cuando se produce un flanco ascendente del reloj y está activada la señal de habilitación *We3*. El banco de registros incluye una señal de *NRst* activa a nivel bajo que inicializará todos los registros a valor 0 cuando se active.



Además, este banco permite la lectura asíncrona simultánea de dos registros independientes. Las direcciones de estos registros se indican respectivamente en las señales de entrada de 5 bits: *A1* y *A2*. Los valores de los registros seleccionados son copiados a las salidas *Rd1* y *Rd2* respectivamente.

### Objetivo

Diseñar y verificar mediante el testbench proporcionado ("*RegsMIPSTb.vhd*") el funcionamiento del banco de registros. Este banco de registros será el que se utilice en la implementación del micro completo.

### Entrega

Se debe entregar el fichero "[RegsMIPS.vhd](#)", que contendrá el Banco de Registros. Este fichero debe seguir las normas de entrega de la asignatura.