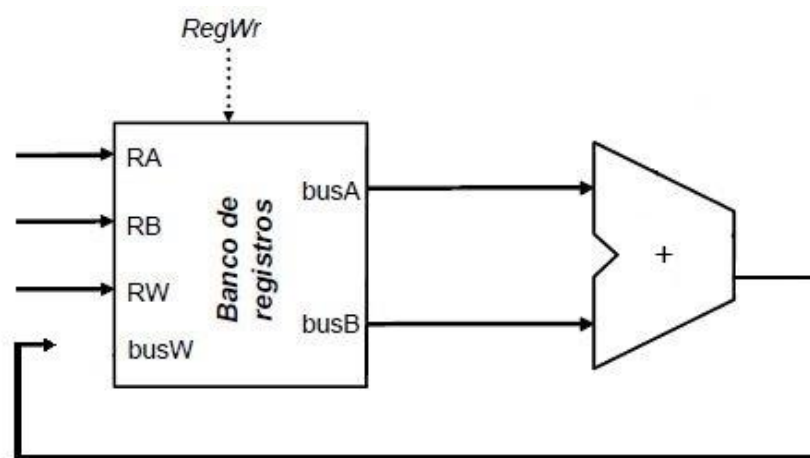


# DISEÑO DE UN PROCESADOR CON BANCO DE REGISTROS

## Práctica 1 (2ª sesión)



Arquitectura y Organización de Computadores 2  
2º Grado Ingeniería Informática

Luis M. Ramos  
luisma@unizar.es

Alejandro Valero  
alvabre@unizar.es

José Luis Briz  
briz@unizar.es

Javier Resano  
jresano@unizar.es



Escuela de  
Ingeniería y Arquitectura  
**Universidad Zaragoza**



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

# 1 RESUMEN

---

En esta segunda sesión vamos a diseñar un procesador muy sencillo que trabaje con el banco de registros y un sumador. Lo programaremos para que realice una serie de escrituras y sumas. Una vez diseñado realizaremos un análisis temporal para averiguar la frecuencia máxima a la que puede funcionar.

**Al entrar al laboratorio** coloca el **trabajo previo del apartado 2.1** encima de la mesa, de forma visible. Además, debes disponer del **banco de registros** diseñado en la sesión anterior **finalizado y funcionando correctamente**. Antes de empezar a trabajar con Logisim conéctate a Moodle desde uno de los equipos del laboratorio y realiza el **control de asistencia a la sesión**.

La práctica finaliza cuando los procesadores funcionan correctamente y han sido entregados a través de **Moodle**. También hay que entregar los **resultados del apartado 3**.

## 2 DISEÑO DE UN PROCESADOR CON BANCO DE REGISTROS

---

### 2.1 TRABAJO PREVIO: DISEÑO EN PAPEL

- a) Dibuja sobre papel los siguientes componentes: un banco de registros **BR32**, una ROM, un contador, un sumador de 32 bits y un multiplexor. Interconecta los componentes, teniendo en cuenta que:
- En cada dirección de la ROM se escribirá una instrucción.
  - El contador se utilizará para direccionar la ROM.
  - Con el sumador se calculará la suma de los dos registros leídos del **BR32**.
  - El multiplexor seleccionará el dato adecuado a escribir en el **BR32**.
  - El procesador debe soportar las siguientes instrucciones:

<code>mov K,rd</code>	<code>; SignExt(K) → BR(rd)</code>	<code>K cte de 16 bits</code>
<code>add ra,rb,rd</code>	<code>; BR(ra) + BR(rb) → BR(rd)</code>	

- b) Define el formato de instrucción y la codificación de las instrucciones.
- c) Diseña la unidad de control del procesador, que calcule el valor de las señales de control (**RegWr** y bit selección MUX) a partir del código de operación de la instrucción.
- d) Utilizando las instrucciones anteriores escribe un programa ensamblador (NIA.asm) que guarde los 6 dígitos de tu NIA en los registros r0 - r5, y calcule la suma de todos los dígitos y la guarde en r6.

### 2.2 TRABAJO EN EL LABORATORIO: DISEÑO CON LOGISIM

- a) Abre en Logisim el circuito de la práctica anterior. Asegúrate que no usas ningún componente *Reloj* (la señal de reloj debe estar conectada a una entrada normal). Añade un circuito nuevo (*Proyecto / Añadir circuito*). Nómbralo **PROC** y márcalo como circuito principal (*Proyecto / Seleccionar Como Circuito Principal*). Guárdalo en un fichero nuevo (NIA-mono.circ).
- b) Coloca los componentes del apartado 2.1 e interconéctalos según tu diseño. Añade un componente Reloj (*Wiring/Reloj*) y conéctalo al contador y al **BR32**.
- c) Traduce tu programa del apartado 2.1.d a hexadecimal y programa la ROM.
- d) Ejecuta ciclo a ciclo hasta que funcione correctamente.

- e) Finalmente entrega tu circuito a través del recurso Moodle “Entrega NIA-mono”. Antes de entregar comprueba que has marcado PROC como circuito principal y que usas un único componente Reloj.

### 2.2.1 Diseño multiciclo

- a) Guarda el diseño anterior en un fichero nuevo (NIA-multi.circ).
- b) Modifica el diseño del procesador conectando dos registros a la entrada del sumador (**RA** y **RB**) y otro a la salida (**RC**).
- c) Dibuja el autómata de la unidad de control para el procesador multiciclo. Impleméntala usando un registro (estado) y una o dos ROM (función de transición y función de salida).
- d) Ejecuta ciclo a ciclo hasta que funcione correctamente.
- e) Finalmente entrega tu circuito a través del recurso Moodle “Entrega NIA-multi”. Antes de entregar comprueba que has marcado PROC como circuito principal y que usas un único componente Reloj. Debes entregar también el autómata de la unidad de control NIA-multi.fsm (dibujado con la herramienta Qfsm disponible en Moodle).

## 3 ANÁLISIS TEMPORAL

---

- a) Considera los siguientes retardos:  $d_{NOT} = 5 \text{ ps}$ ;  $d_{OR2-4} = 20 \text{ ps}$ ;  $d_{AND2-4} = 20 \text{ ps}$ ;  $d_{REG} = 50 \text{ ps}$ ;  $t_{setup_{REG}} = 30 \text{ ps}$ ;  $d_{ROM} = 80 \text{ ps}$ ;  $d_{CONT} = 60 \text{ ps}$ ;  $d_{SUM} = 260 \text{ ps}$ .
- b) Identifica el camino crítico del procesador monociclo y su retardo ( $d_{max_{mono}}$ ). ¿Cuál es su frecuencia máxima (GHz) de funcionamiento ( $f_{max_{mono}}$ )?
- c) Identifica el camino crítico del procesador multiciclo y su retardo ( $d_{max_{multi}}$ ). ¿Cuál es su frecuencia máxima de funcionamiento ( $f_{max_{multi}}$ )?
- d) ¿Cuánto tiempo tarda el procesador monociclo en ejecutar tu programa ( $t_{ex_{mono}}$ )?
- e) ¿Cuánto tarda el procesador multiciclo ( $t_{ex_{multi}}$ )?
- f) ¿Cuál es la relación entre los tiempos de ambos procesadores ( $t_{ex_{multi}} / t_{ex_{mono}}$ )?
- g) Cuando tengas todos los cálculos realizados introdúcelos a través del recurso Moodle “Análisis temporal monociclo vs multiciclo”.