

Taller de Microprogramación

Organización del Computador 1

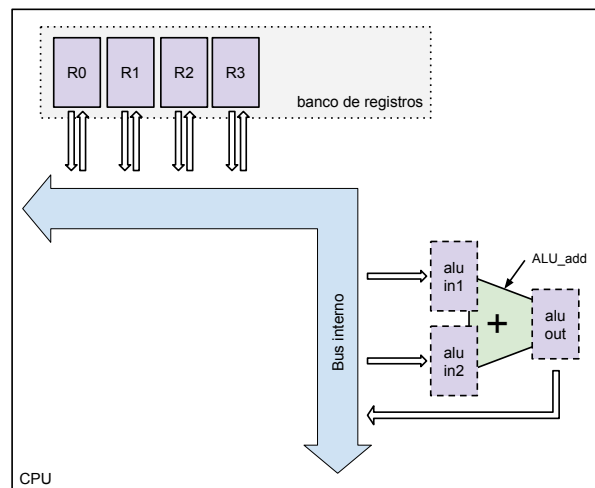
Primer Cuatrimestre 2014

El primer prototipo de la máquina ORGA1, llamado *Orga0*, poseía 4 registros de propósito general de 8 *bits*: R_0 a R_3 . Algunas de las instrucciones que soportaba este prototipo eran:

Instrucción	Descripción
MOV R_i , R_j	Copia el contenido del registro R_j al registro R_i
ADD R_i , R_j	Suma el contenido del registro R_i al contenido del registro R_j y lo guarda en R_i

Parte de la microarquitectura del CPU de *Orga0* ha sido implementado y puede simularse su funcionamiento en Logisim¹.

El camino de datos (*datapath*) *parcial* de la máquina es el siguiente:



1. Las microinstrucciones

La unidad de control de la CPU es microprogramable. Posee una memoria interna que almacena las microinstrucciones (codificadas en 8 *bits*).

Las instrucciones descritas en la arquitectura son implementadas en la microarquitectura con distintas secuencias de microinstrucciones.

En la memoria de la unidad de control se almacenan los microprogramas correspondientes a la implementación de cada instrucción.

1.1. Formato de las microinstrucciones

Las microinstrucciones tienen 8 *bits*, interpretados de la siguiente manera:

Asignaciones de registros

2 bits	3 bits	3 bits
11	registro destino	registro fuente

¹<https://launchpad.net/ubuntu/+source/logisim>

Ejemplo. $R_2 := R_1$

D1 ; En el siguiente ciclo, R_2 tendrá el mismo contenido que R_1 .

Microinstrucciones aritméticas

Para realizar operaciones con la ALU se cuenta con las microinstrucción *ALU_add*

2 bits	3 bits	3 bits
10	000	000

Ejemplo.

80 ; En el siguiente ciclo, el registro *ALU_out* contendrá el resultado de sumar los contenidos de *ALU_in1* y *ALU_in2*.

2. Simulación en Logisim

2.1. El contenido del microprograma

Como ya mencionamos, la unidad de control cuenta con una memoria interna donde almacena el microprograma que ejecutará. Esta memoria interna es cargada desde cualquier archivo de texto, cuya primera línea debe ser

v2.0 raw

En la siguiente línea debe aparecer el contenido de la memoria de microprograma separando cada valor por un espacio. Por ejemplo, con el siguiente programa se ejecutan las dos microinstrucciones explicadas anteriormente, una después de la otra.

v2.0 raw
D1 80

Consigna

- 1) Escribir un microprograma para cada una de las siguientes instrucciones:
 - a) **MOV** R_2, R_3
 - b) **ADD** R_0, R_1
- 2) Sólo agregando conexiones entre los componentes que aparecen en la microarquitectura dotar al CPU de funcionalidad suficiente para poder escribir un microprograma para cada una de las siguientes instrucciones, y escribir cada microprograma:
 - a) **SWAP** R_0, R_3 que intercambia los valores de ambos registros
 - b) **CLEARALL** que pone en cero todos los registros
- 3) Completar el diagrama del *datapath* para que soporte el ciclo de instrucción de la máquina **Orga0**.

Se debe entregar un archivo con cada microprograma escrito y el archivo **.circ** con la modificación realizada. Además, una hoja con la modificación de la arquitectura.

Mandar los archivos por mail a **orga1-doc@dc.uba.ar** con asunto **Taller de Microprogramación**, en el cuerpo del mensaje deben figurar el nombre de los integrantes del grupo y sus números de LU.