

UNIVERSIDAD NACIONAL DEL CENTRO DE LA PROVINCIA DE BUENOS AIRES

# Consideraciones Adicionales – TPE

---

Introducción a la Arquitectura de Computadoras

**Autores: Ferrero Martino – Herrero Juan Francisco**

**29/06/2022**



## **Consideraciones adicionales**

A continuación se detallarán las diversas consideraciones adicionales al enunciado que se tuvieron en cuenta al realizar el Trabajo Práctico Especial (TPE) de la materia. Estas últimas se ven reflejadas en el código VHDL adjunto en la resolución del trabajo, llevado a cabo en la herramienta “EDA Playground” propuesta por la cátedra.

- 1- Cada etapa posee un proceso regido por el clock y el reset para facilitar la legibilidad y comprensión del código, aunque también esto podría resolverse utilizando un único proceso.
- 2- Se consideró que la escritura de registros es en flanco descendente de reloj.
- 3- Se separaron en distintos diseños los componentes ALU, ALU control, banco de registros, unidad de control y la memoria de instrucciones y de datos.
- 4- Se consideró que todos los procesos son con reset asíncrono con activo en alto y flanco alto de reloj.
- 5- Dadas las siguientes instrucciones del programa proporcionado por la cátedra:

***slt \$s7, \$t1, \$t2***

***slt \$t8, \$s0, \$t2***

***beq \$t1, \$s7, salto1***

Se observa una dependencia y un riesgo RAW generado en torno al registro \$s7, el cual es leído para realizar la comparación del beq y a su vez es escrito por el primer slt. Considerando que hay una sola instrucción de por medio no le damos la cantidad de ciclos necesarios para que el valor leído del registro \$s7 sea el correcto (ya que no se llega a escribir el nuevo valor del mismo antes de que el beq lo lea). Por lo tanto, se decidió agregar una instrucción NOP, particularmente formada por 32 ceros en binario.

Cabe aclarar que la unidad de control decodifica el NOP como una instrucción de tipo R, pero esto no afecta porque siempre se escribirá en el registro 0 (\$zero).

- 6- Dadas las siguientes instrucciones del programa proporcionado por la cátedra:

***add \$t1, \$t1, \$t1***

***salto1: beq \$t1, \$zero, salto2***

Se observa una dependencia y un riesgo RAW generado en torno al registro \$t1, el cual es leído para realizar la comparación del beq y a su vez es escrito por el add. En este caso es necesario agregar dos instrucciones NOP entre medio con el fin de salvar dicho riesgo.

- 7- En cuanto a la unidad de control se consideró que para la instrucción lui se agrega una rama del if para la misma con la señal alu\_op en 11. Por otra parte dado el alu\_op indicado, en la ALU Control se pone el alu\_control\_out en 100. Y dado el valor de dicha

señal, en la ALU se lleva a cabo la respectiva concatenación del inmediato en la parte alta.

8- En este procesador se lleva a cabo un adelantamiento para los saltos a la etapa ID. Para esto se tienen en cuenta las siguientes consideraciones:

- a. Se realiza la comparación de los valores de los registros implicados en la instrucción en la etapa ID (de cualquier modo la lógica del Z en la ALU se sigue manteniendo, más allá de que no se utilice para el cálculo de la condición de salto).
- b. Se realiza el cálculo de la condición de salto (branch and comparación\_beq) en la etapa ID:

***pc\_src <= branch and comparacion\_beq***

- c. Se lleva a cabo el flush del pipeline IF/ID (se ponen todos los registros en 0) cuando haya un flanco alto de reloj y el pc\_src sea 1.