

# INFORME PRÁCTICA 1

Alejandro Santorum Varela - alejandro.santorum@estudiante.uam.es

David Cabornero Pascual - david.cabornero@estudiante.uam.es

Prácticas Arquitecturas de Ordenadores - Pareja PM19

Universidad Autónoma de Madrid

07-10-2018

## Contents

1	Introducción	2
2	Ejercicio 1: MIPS unicycle	2
3	Ejercicio 2: MIPS segmentado	2
4	Algunas decisiones de diseño	4
5	Conclusión	4

# 1 Introducción

Este documento recoge un pequeño informe de la práctica 1 de Arquitecturas de ordenadores. En el primer apartado se hablará del microprocesador MIPS uniciclo y, en el segundo apartado, de su versión segmentada.

## 2 Ejercicio 1: MIPS uniciclo

El primer ejercicio consiste en programar en VHDL el procesador MIPS uniciclo. Esto mismo ya se hizo en Estructuras de Computadores el primer curso, con la ligera variación de algunas instrucciones diferentes.

Después de programar esta versión, se ha probado su correcto funcionamiento con ModelSim junto con el programa facilitado por los profesores de la asignatura.

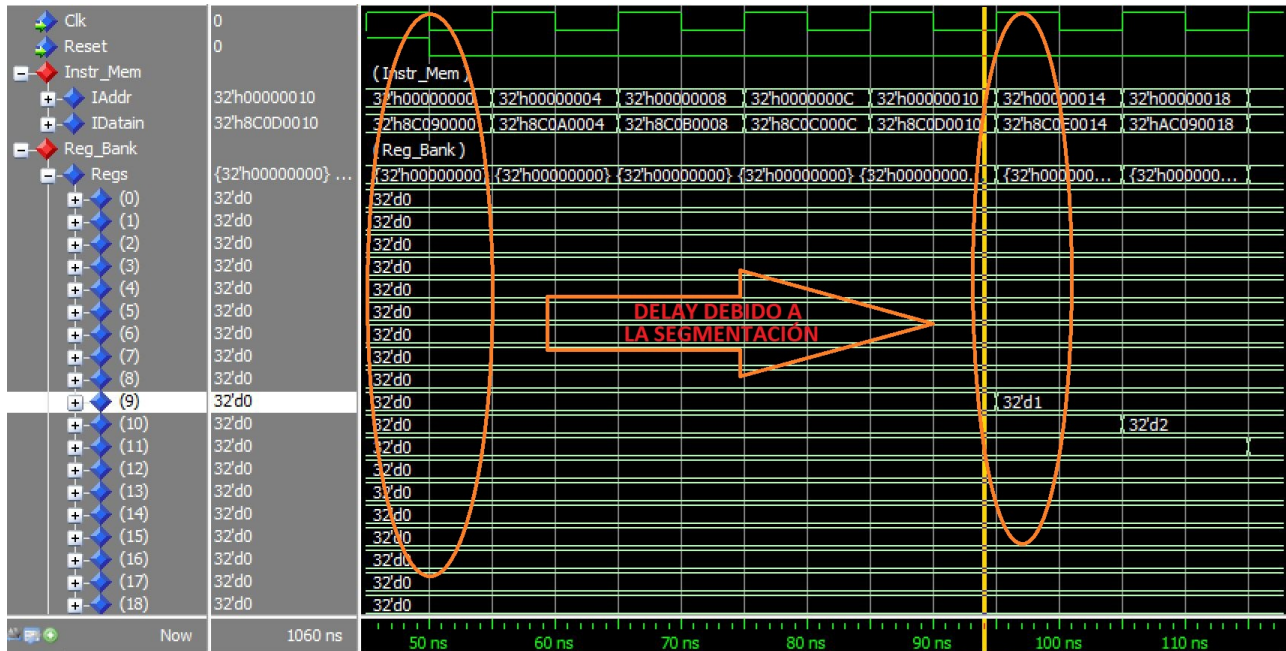
En primer lugar, podemos comprobar que las instrucciones **LW** funcionan como se esperaba en el ciclo, instrucción y registro adecuados (comprobar simulación). A continuación se prueban varias instrucciones de **Tipo R** (R-type), y, por último, instrucciones **LUI** y el **BEQ** que realiza un salto efectivo a la dirección de salto (fíjese que de la instrucción 0x12320004 salta a la 0x02F6022).

Todo esto se puede observar en la simulación los los ficheros VHDL entregados.

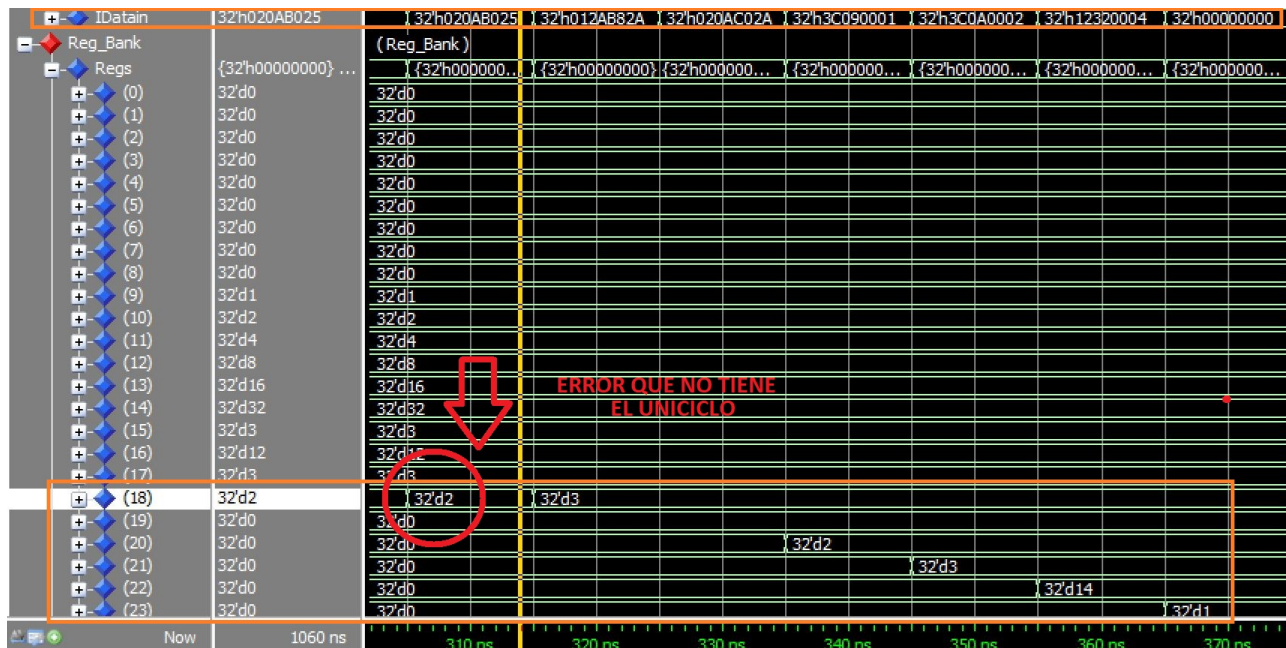
## 3 Ejercicio 2: MIPS segmentado

La versión segmentada del MIPS permite que su funcionamiento sea más eficiente, ya que pueden varias instrucciones pueden estarse ejecutando al mismo tiempo, siempre y cuando esten usando elementos del procesador independientes. Esto se consigue dividiendo el procesador en fases/areas de funcionamiento común limitadas por registros, que irán transportando los datos y bits de control a lo largo del procesador.

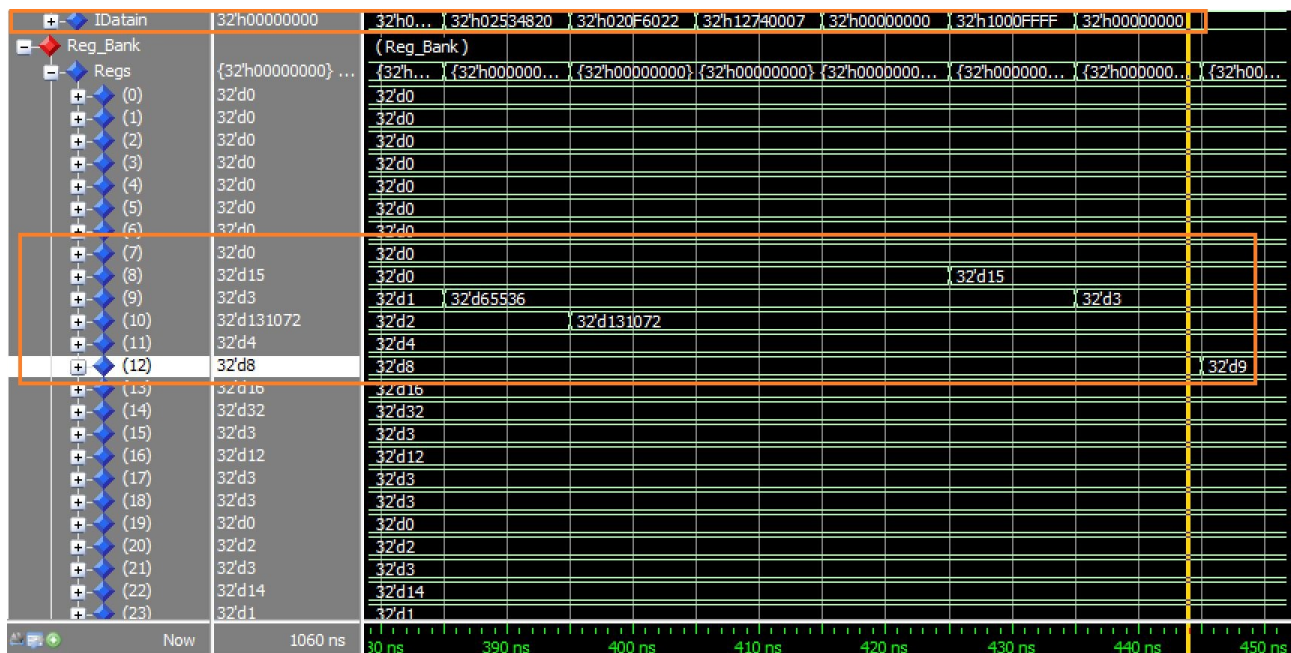
En la primera imagen podemos ya percibir que existe un retraso o *delay* entre la captura de la instrucción y su resultado, que en este caso son instrucciones **LW**, por lo que no se escribe en el banco de registros el dato hasta transcurridos 5 ciclos de reloj.



Ahora es el turno de comprobar el correcto funcionamiento de las instrucciones **Tipo R**. Todo sigue tal y como se especifica en el programa proporcionado, hasta el error que se suponía que tenía que aparecer en la versión segmentada, ya que aún no están incluidos los controles de riesgos:



Funcionamiento de las instrucciones **LUI** y la instrucción **BEQ** que realiza un salto efectivo, pero no hasta pasados tres ciclos de reloj, por lo que se ejecutan dos instrucciones **ADD** que no se deberían haber ejecutado.



Por último, se puede ver en la simulación que efectivamente esta finaliza con un bucle infinito entre la instrucción 0x1000FFFF y varias NOP's.

## 4 Algunas decisiones de diseño

Las decisiones de diseño más importantes que hemos tenido que realizar han sido, en primer lugar, aumentar en 1 bit la señal ALUOp que enviaba información entre la unidad de control y el controlador de la ALU, ya que con dos bits no se podían especificar todos los tipos de instrucciones necesarios.

Por otro lado, la instrucción "especial" **NOP** se ha procesado como si se tratase de una instrucción **add \$zero \$zero \$zero**, lo cual desarrolla la misma funcionalidad que se especificaba en la documentación de la práctica.

Por último, ni en el libro *Estructura y diseño de computadores: La interfaz* de Hennessy y Patterson; ni en las diapositivas de teoría, venía recogido el MIPS segmentado con la ruta de datos de la instrucción **JUMP**. Creemos que la mejor opción es relizar el salto ya en el segundo ciclo de procesamiento, después de los registros IF/ID, ya que aquí ya sabemos todo lo necesario para realizar el salto.

## 5 Conclusión

En esta práctica hemos retomado la programación en VHDL volviendo a programar el MIPS unicity ya diseñado en la asignatura de Estructuras de Computadores, y ahora hemos ido un paso más adelante con el MIPS segmentado. No obstante, este procesador es muy mejorable,

que será el objetivo de la segunda práctica.

Este informe no era obligatorio, pero hemos decidido entregarlo para ayudar en todo lo posible y para mostrar como hemos enfrentado esta práctica.