

Apellidos: _____

Nombre: _____

1) Para un procesador MIPS monociclo las latencias asociadas a las distintas etapas del ciclo de instrucción son las que se muestran en la siguiente tabla:

IF	ID	EX	MEM	WB
200ps	150ps	125ps	250ps	125ps

(a) ¿Cuál sería el tiempo de ciclo de reloj de una implementación monociclo del procesador?

(b) ¿Cuál sería el tiempo de ciclo de reloj si se segmenta el cauce del procesador en esas 5 etapas?

En la implementación segmentada del procesador, no se ha implementado anticipación en el banco de registros (adelantamiento). Además, el HW del procesador se ha modificado de tal manera que la terminación (= carga del PC) tanto del salto condicional como del incondicional se realiza en la etapa ID.

(c) Para el siguiente código, que se ha numerado para vuestra conveniencia, marca las dependencias que dan lugar a riesgos en el cauce segmentado descrito con anterioridad, indicando de qué tipo son y el registro causante.

Dependencias que dan lugar a riesgos:
<pre>1. add \$6, \$0, \$0 2. add \$10, \$0, \$0 3. et: lw \$12, 48(\$10) 4. lw \$13, 20(\$10) 5. sub \$14, \$13, \$12 6. add \$6, \$6, \$14 7. addi \$10, \$10, 4 8. bne \$10, \$15, et 9. sw \$8, 8(\$0)</pre>

- (d) Reescribe el código, primero introduciendo el menor número de NOPs para evitar los riesgos. Posteriormente, **si es posible**, reordena el código para reducir al mínimo el número de NOPs y que la ejecución del código siga siendo correcta. Especifica para cada código el tiempo de ejecución suponiendo que se realiza un número N_{iter} de iteraciones (el código original se ejecutaría en la implementación monociclo, mientras que los otros dos códigos se ejecutarían en la implementación segmentada).

Código original (monociclo)	Código con NOPs	Código Reordenado
<pre> add \$6, \$0, \$0 add \$10, \$0, \$0 et: lw \$12, 48(\$10) lw \$13, 20(\$10) sub \$14, \$13, \$12 add \$6, \$6, \$14 addi \$10, \$10, 4 bne \$10, \$15, et sw \$8, 8(\$0) </pre>		
Tiempo monociclo:	Tiempo ej.:	Tiempo ej.:

2) Dado un conjunto de 4 instrucciones para un procesador de tamaño de palabra de 8 bits, dos registros **R0** y **R1** y una memoria de 64 bytes:

Instrucción	Acción	Formato de instrucción							
ADI rd, rs, cte	$rd \leftarrow rs + \text{ExtSig}(\text{cte})$	Opcode		rd	rs	cte			
		0	1						
ADM rd, rs	$rd \leftarrow rd + M(rs[5:0])$	Opcode		rd	rs				
		1	0			x	x	x	x
STM rd, rs	$M(rs[5:0]) \leftarrow rd$	Opcode		rd	rs				
		1	1			x	x	x	x
BBQ dir	If $R1 < R0$, $PC \leftarrow \text{dir}$	Opcode		dir					
		0	0						

donde cte es un entero representado en complemento a 2.

- Diseña una unidad de datos para poder ejecutar esas cuatro instrucciones, en base a los elementos hardware proporcionados. Además puedes utilizar los elementos hardware adicionales que creas oportunos (registros, multiplexores, sumadores, comparadores, etc). Indica claramente los puntos de control necesarios.
- Construye la tabla de verdad donde se indique claramente qué señales de control deben activarse (1) y cuales no (0) para cada instrucción.

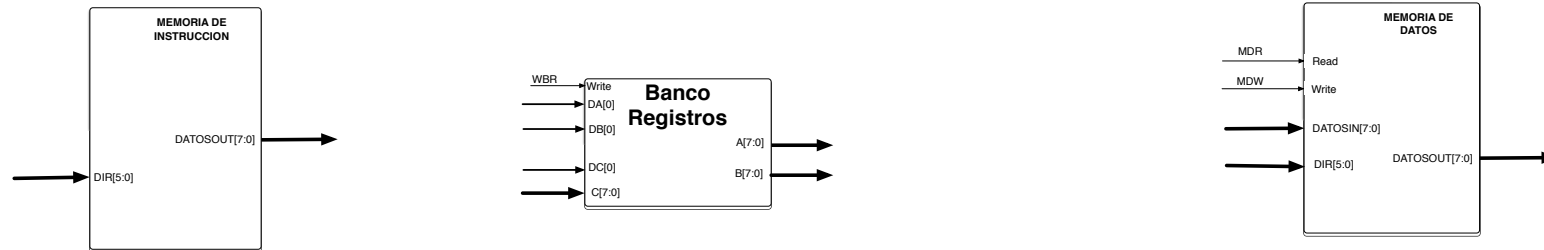
Tabla de control:

	Señales de control													
Instrucción														
ADI														
ADM														
STM														
BBQ														

Si inicialmente el contenido de los registros es 0 (incluido el contador de programa, PC):

- Describir la evolución del contenido de los registros y de la memoria (traza de ejecución) si se ejecutan 6 instrucciones.

Memoria		Trazo de ejecución			
Pos	Contenido	PC	Instrucción (Ej. ADI R1, R0, 3)	Registro o posición de memoria a escribir	Valor a escribir
0	52 _{hex}	0			
1	65 _{hex}				
2	d0 _{hex}				
3	90 _{hex}				
4	02 _{hex}				
5	59 _{hex}				
6	--				
7	--				
...	...				



Unidad de Datos