...ción de este código, si el procesador usase un **predictor de salto** ...
...el tiempo de ejecución.

ESTRUCTURA DE COMPUTADORES
Primera evaluación (ejercicios)

APELLIDO: _____ **NOMBRE:** _____

2.0p] El siguiente código se ejecuta en un procesador segmentado de 5 etapas como el MIPS estudiado en clase: IF, ID, EX, MEM y WB. El salto se decide en la etapa ID. El procesador usa la técnica de salto fijo no efectivo, tiene una unidad de detección de riesgos en la etapa ID y una unidad de anticipación en la etapa EX. La ejecución de una operación en la unidad de suma en punto flotante requiere 3 ciclos. Las etapas MEM y WB solo tienen capacidad para albergar una única instrucción.

```
1.        addi $t1, $0, 4
bucle :
2.        lwc1 $f2, 0($a0)
3.        add.s $f4, $f4, $f2
4.        addi $a0, $a0, 4
5.        addi $t1, $t1, -1
6.        swc1 $f4, 0x40($a0)
7.        bne $t1, $0, bucle
8.        swc1 $f3, 0($a0)
```

○ dependencia de flujo de datos
□ dependencia de salida
△ antidependencias
} dependencias del código

▷ dependencia de control

a [0.3p] Identifica **sobre el código anterior** las dependencias entre instrucciones.

b [0.4p] Muestra el diagrama multiciclo para la primera iteración, indicando explícitamente **anticipaciones y bloqueos**. Señala cómo cambiará ese diagrama en la última iteración del bucle.

c [0.2p] Identifica **sobre el diagrama** los riesgos que ocurren y su tipo.

las anticipaciones son siempre hacia la etapa de EX

→ = anticipación

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1 addi	IF	ID	EX	MEM	WB														
2 lwc1	IF	ID	EX	MEM	WB														
3 add.s		IF	ID	(ID)	A1	A2	A3	MEM	WB										
4 addi			IF	(IF)	ID	EX	MEM	WB											
5 addi					IF	ID	(ID)	EX	MEM	WB									
6 swc1						IF	(IF)	ID	EX	MEM	WB								
7 bne							IF	ID	(ID)	EX	MEM	WB							
8 swc1								IF	(IF)	nop									
[$t1 = $t0]																			
[$t1 != $t0]																			
bne									IF	ID	(ID)	EX	MEM	WB					
swc1									[$t1 == $t0] sale del bucle	IF	(IF)	ID	EX	MEM	WB				

riesgo RAW

xq es en punto flotante es como hacer EX en 3veces

pone una A xq es una addi

(no hace falta A xq es entero)

riesgo RAW

riesgo estructural

1 iteración

riesgo de control

por lo que decía yo, no cumple la condición entonces se cancela la operación y se rellena con 0.

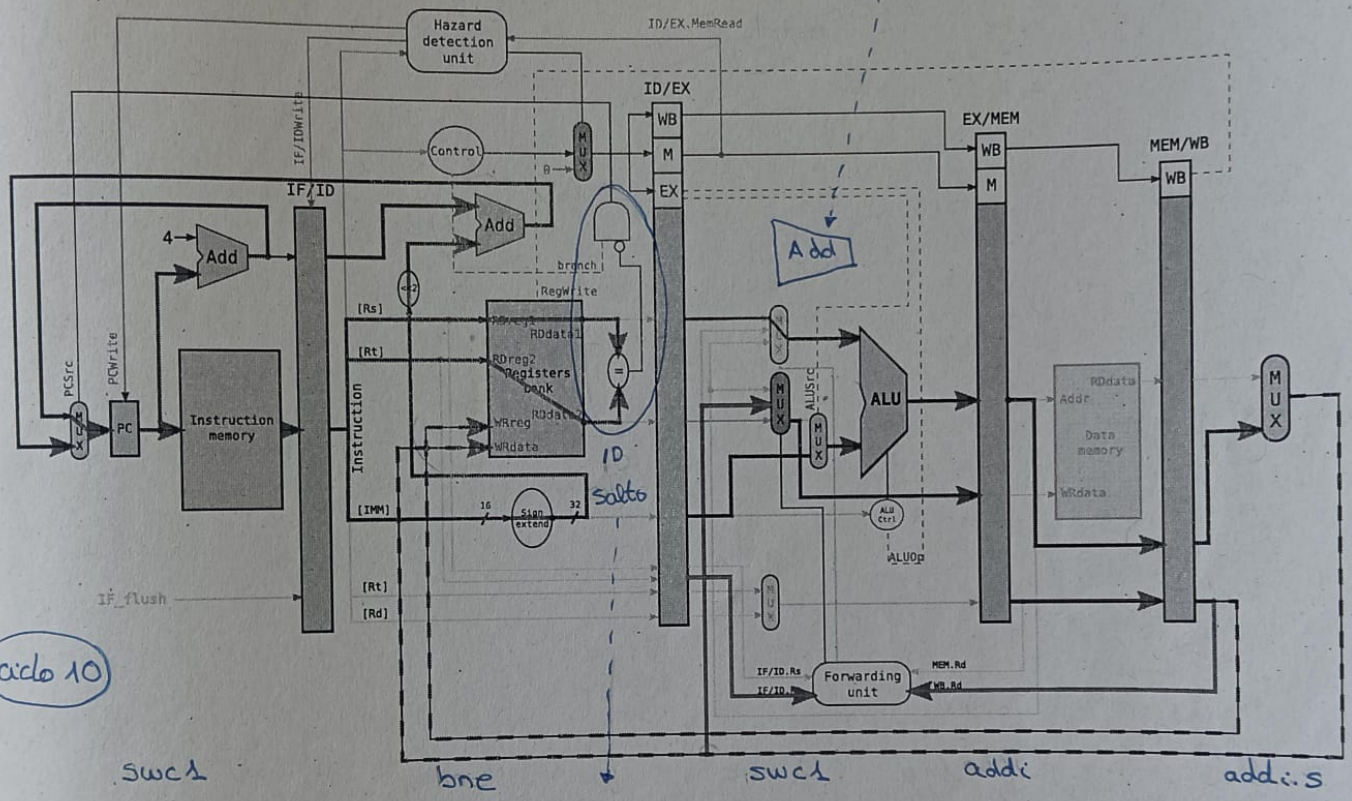
última iteración

d [0.3p] En la ejecución de este código, si el procesador usase un **predictor de salto dinámico** indica razonadamente cómo afectaría al tiempo de ejecución.

e [0.3p] Si en lugar de salto fijo no efectivo el procesador usase **salto retardado**, indica razonadamente qué instrucción o instrucciones son las más adecuadas para ocupar el hueco de retardo.

f [0.5p] El siguiente diagrama muestra las líneas de datos y señales de control significativas en el procesador en un instante concreto de la ejecución del programa anterior. Identifica razonadamente **qué instrucción se encuentra en cada etapa**.

(si el add está en EX se hace ahí el salto)



ciclo 10

swc1 bne swc1 addi addi.s

hay que fijarse rg al estar eso se hace un salto y por eso es el ciclo 10.

○ dependencia de flujo de datos
□ dependencia de salida
△ antidependencia

Ejercicio 1

```
tag: 1 add.s $f0, $f2, $f4
     2 lwc1 $f0, 0($a0)
     3 add.s $f1, $f0, $f4
     4 addi $a0, $a0, 4
     5 addi $t0, $t0, -1
     6 swc1 $f1, 0($a0)
     7 bne $t0, $0, tag
     8 swc1 $f1, 4($a0)
```

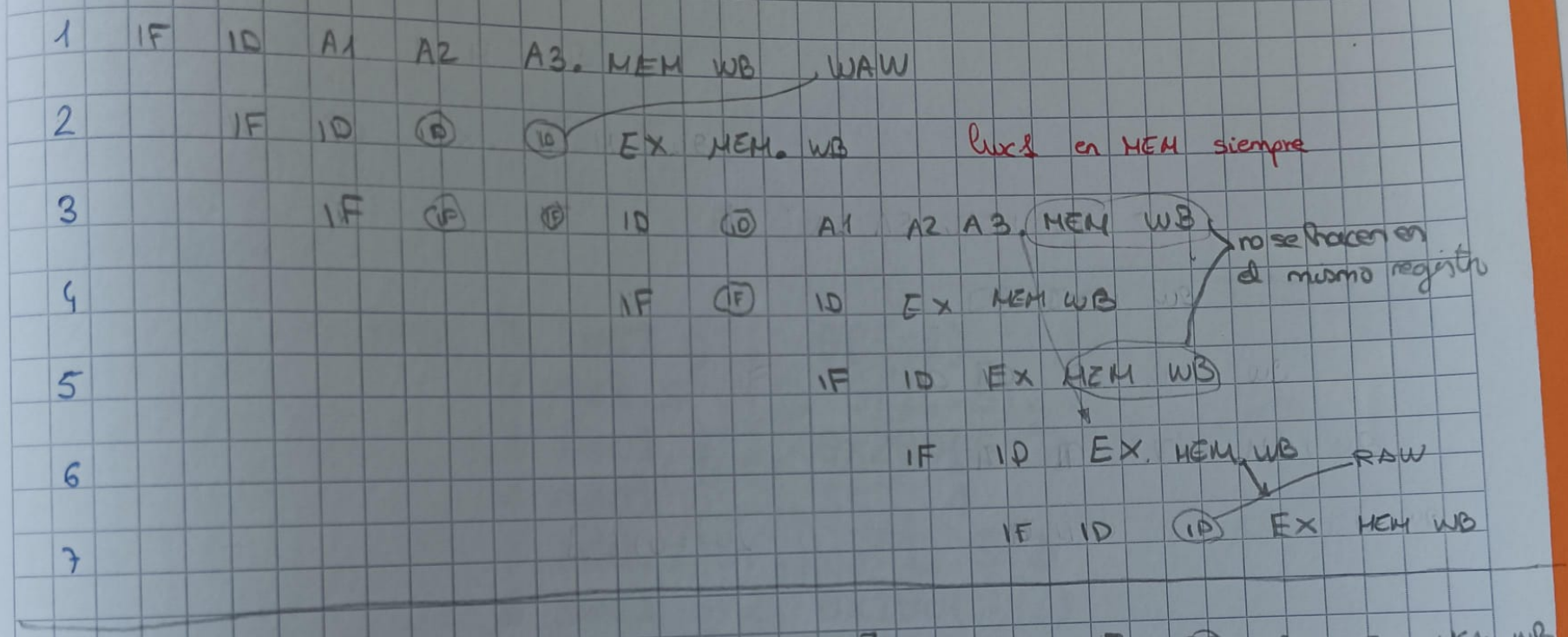
Este código se ejecuta en un procesador con unidad de detección de riesgos en ID, unidad de anticipación en EX.
El salto se decide en la etapa ID e implementa salto fijo no efectivo.
La unidad de suma en punto flotante está segmentada y tiene una latencia de 3 ciclos.
En este procesador pueden coincidir dos instrucciones en la etapa MEM si no acceden ambas a memoria, y en la etapa WB si no escriben el mismo banco de registros (es el comportamiento del SIMULA).

1. Dibujar el diagrama multiciclo para una iteración del código indicando explícitamente las anticipaciones y los riesgos en el procesador.
2. Si el procesador utilizase salto retardado, ¿qué instrucciones podemos colocar en el hueco de retardo?

This code runs in a processor with a hazard detection unit in the ID stage, and a forwarding unit in the EX stage.
Branches are decided in the ID stage and implements fixed non-taken branch technique.
The floating point adder is pipelined and has a latency of 3 clock cycles.
In this processor, MEM stage can hold several instructions as long as at most one of them accesses the data memory, and WB as well as long as they do not write the same registers bank.

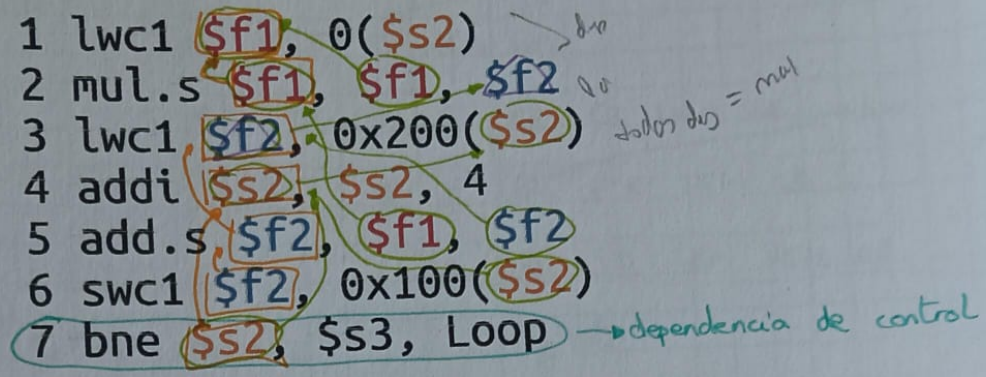
1. *Draw the multi-cycle diagram for one iteration of the loop. Mark forwardings and stalls in the processor.*
2. *If this processor used delayed branch, which instructions could we move into the delay slot?*

que debo hacer para saber cual es la mejor opción


1 IF ID A1 A2 A3. MEM WB
WAW
2 IF ID (D) (ID) EX MEM. WB
lw $1 en MEM siempre
3 IF (IF) (D) ID (ID) A1 A2 A3. MEM WB
no se hacen en el mismo registro
4 IF (IF) ID EX MEM WB
5 IF ID EX MEM WB
6 IF ID EX. MEM. WB
RAW
7 IF ID (ID) EX MEM WB
8
[$t0 == 0]
IF (IF) ID EX MEM WB
[$t0 != 0]
IF (IF)
salta a 1.

Ejercicio 2

```
Loop: 1 lwc1 $f1, 0($s2)
      2 mul.s $f1, $f1, $f2
      3 lwc1 $f2, 0x200($s2)
      4 addi $s2, $s2, 4
      5 add.s $f2, $f1, $f2
      6 swc1 $f2, 0x100($s2)
      7 bne $s2, $s3, Loop
```

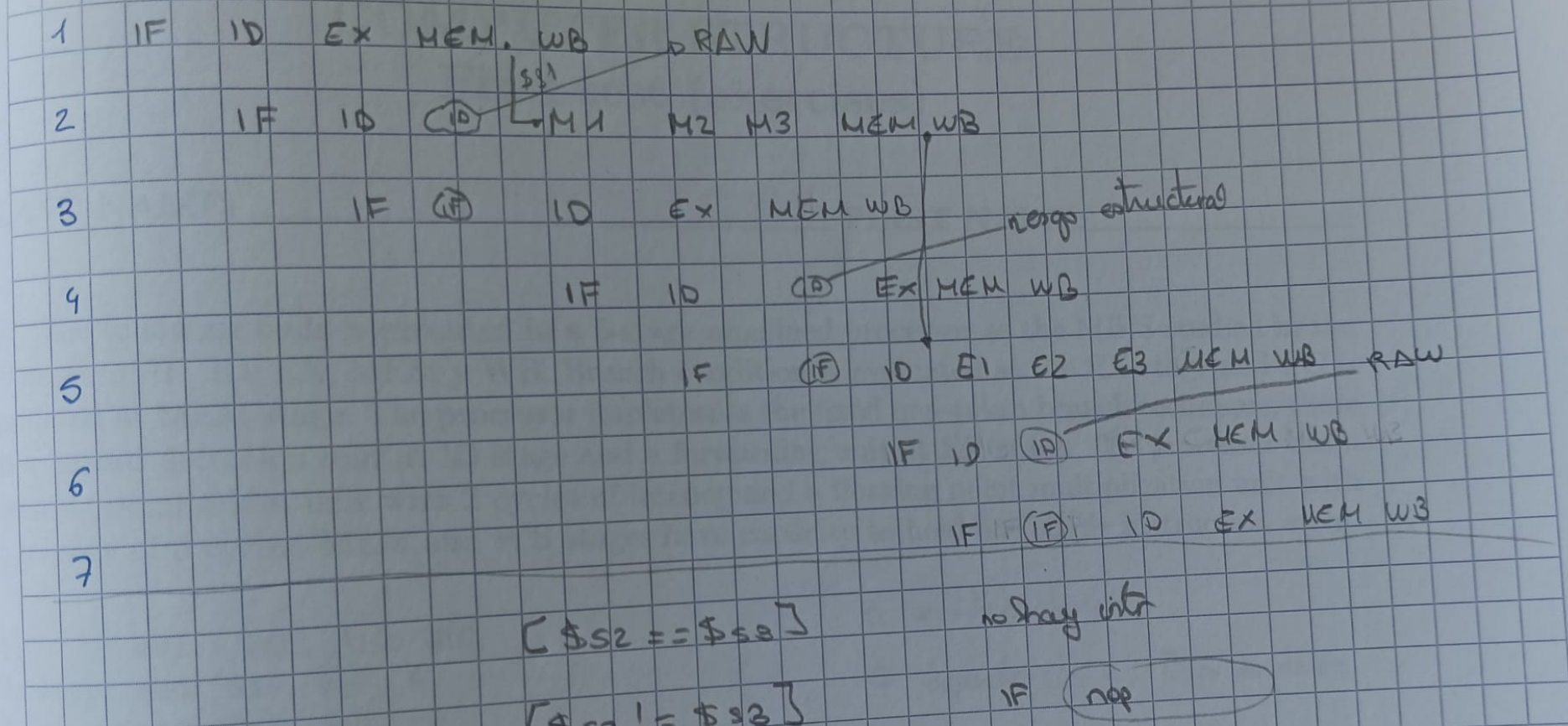


Este código se ejecuta en un procesador con unidad de detección de riesgos en ID, unidad de anticipación en EX.
El salto se decide en la etapa ID e implementa salto fijo no efectivo.
Las unidades de suma y multiplicación en punto flotante están segmentadas y tienen una latencia de 2 y 3 ciclos respectivamente.
En este procesador las etapas MEM y WB sólo pueden albergar una instrucción.

1. Dibujar el diagrama multiciclo para una iteración del código indicando explícitamente las anticipaciones y los riesgos en el procesador.
2. Si el procesador utilizase salto retardado, ¿qué instrucciones podemos colocar en el hueco de retardo?

This code runs in a processor with a hazard detection unit in the ID stage, and a forwarding unit in EX stage.
Branches are decided in the ID stage and implements fixed non-taken branch technique.
The floating point adder and multiplication units are pipelined and have a latency of 2 and 3 clock cycles, respectively.
In this processor, MEM and WB stages can only hold one instruction each.

1. *Draw the multi-cycle diagram for one iteration of the loop. Mark forwardings and stalls in the processor.*
2. *If this processor used delayed branch, which instructions could we move into the delay slot?*


1 IF ID EX MEM WB
RAW
$s1
2 IF ID ID M1 M2 M3 MEM WB
3 IF IF ID EX MEM WB
riesgo estructural
4 IF ID ID EX MEM WB
5 IF IF ID E1 E2 E3 MEM WB
RAW
6 IF ID ID EX MEM WB
7 IF IF ID EX MEM WB
[$s2 == $s3]
no hay intr
[$s2 != $s3]
IF nop

COMPUTER STRUCTURE
First test (exercises)

LAST NAME: _____ **FIRST NAME:** _____

[2.0p] The following code is executed in a 5-stage pipelined processor as the MIPS studied in the classroom: IF, ID, EX, MEM y WB. Branch condition is evaluated at the EX stage and PC is updated at MEM stage. The processor implements the fixed non-taken branch technique, there is a hazard detection unit at ID stage and a forwarding unit at EX stage. The processor has a floating point adder unit with 2 cycles of latency and a floating point multiplication unit with a latency of 5 cycles. MEM and WB stages have capacity to hold one single instruction each.

```
1.        mul.s $f0, $f0, $f0
2. loop:  addi $a0, $a0, 4
3.        addi $a1, $a0, -4
4.        lwc1 $f0, 0($a0)
5.        add.s $f2, $f0, $f4
6.        add $s1, $s1, $t5
7.        swc1 $f2, 0($s1)
8.        bne $a1, $a2, loop
9.        sw $s1, 0($a2)
```

△ antidependencia

○ dependencia de flujo de datos

□ dependencia de salida

→ dependencia de control

a [0.3p] Identify the dependencies among instructions **over the source code.**

b [0.4p] Show the multicycle diagram for the first loop iteration, and indicate **forwardings and stalls** explicitely. Mark how the diagram would change in the last loop iteration.

c [0.2p] Identify **over the diagram** the hazards and their type.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	IF	ID	M1	M2	M3	M4	M5	MEM	EX										
2		IF	ID	EX	MEM	WB													
3			IF	ID	EX	MEM	WB												
4				IF	ID			EX	MEM	WB									
5																			
6																			
7																			
8																			
9																			

atraco 2 xq sino queda el dato de arriba