PREGUNTAS TIPO TEST

El buffer de reorden se usa para eliminar dependencias WAW:
RESPUESTA: VERDADERO

En un VLIW, una instrucción decodificada que no disponga de unidad para su ejecución está ocupando una entrada de una estación de reserva.
RESPUESTA: FALSO

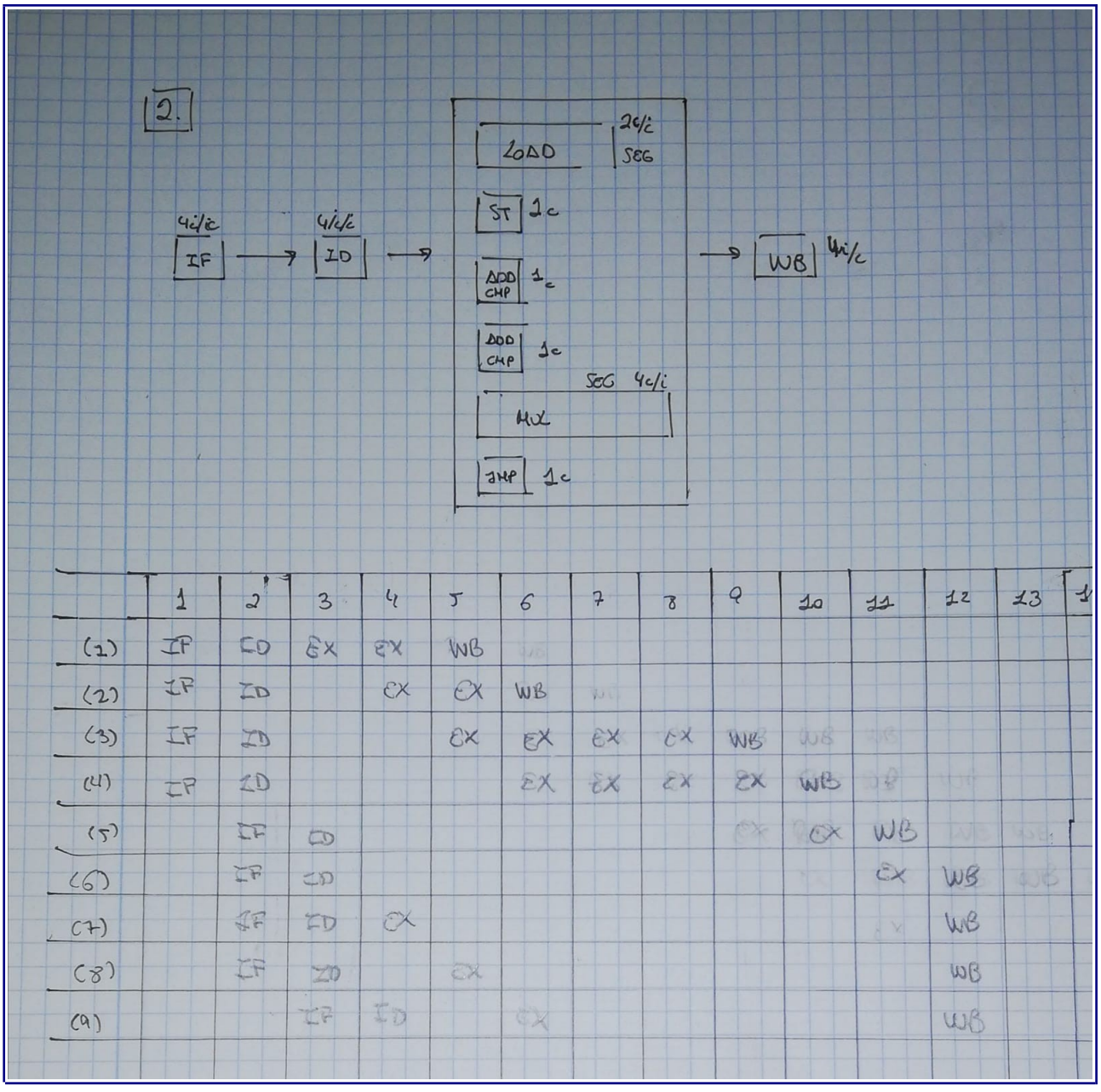
Hasta que una instrucción decodificada no disponga de los operandos para su ejecución permanecerá en una ventana de instrucciones de un VLIW.
RESPUESTA: FALSO

EJERCICIO TABLA

2 V/F

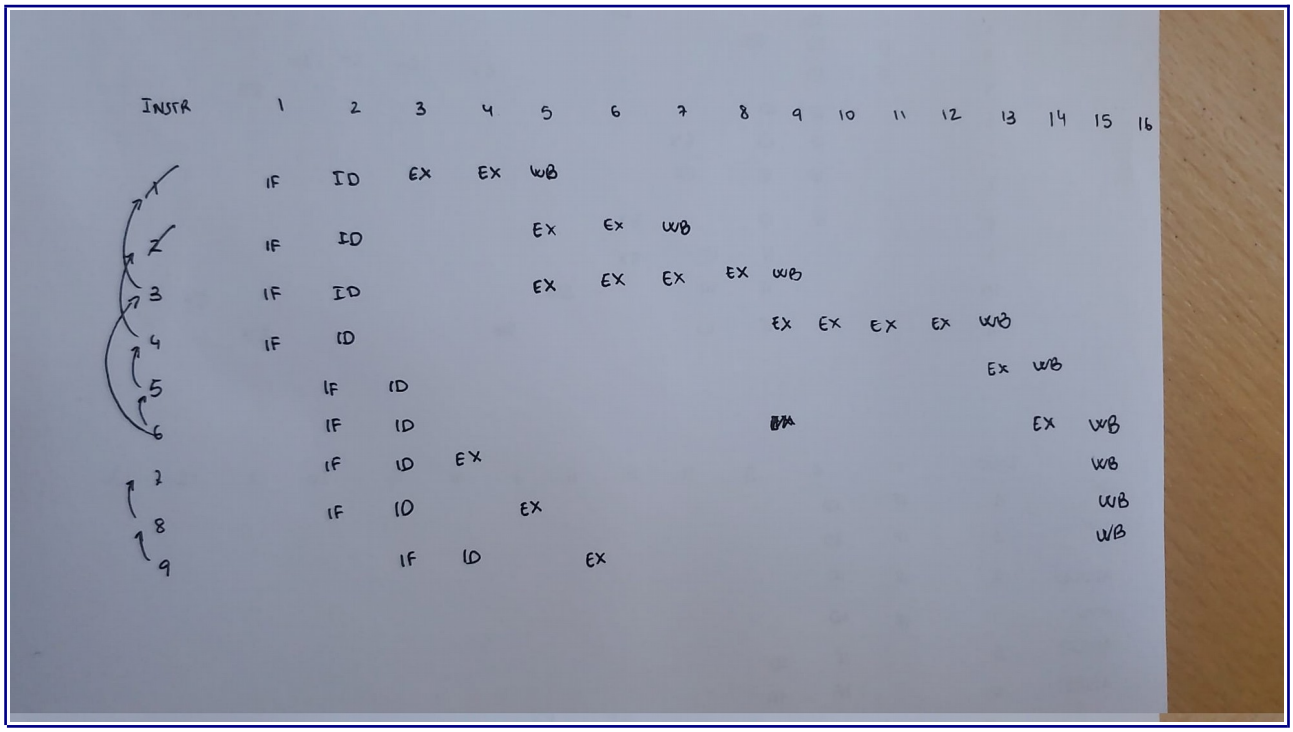
(A) El cauce de un superescalar tiene las siguientes etapas: IF (1 ciclo=1 c para cada instrucción) capaz de procesar 4 instrucciones por ciclo (i/c), ID (1 c) capaz de procesar 4 i/c, EX (de 1c a 4 c de latencia dependiendo de la unidad) y WB (1 c) capaz de retirar del buffer de reorden (ROB) 4 i/c. Unidades: 1 para carga de memoria segmentada en dos etapas de 1 c cada una, 1 de almacenamiento en memoria (1 c), 2 unidades para $addq$ y $cmpl$ (1 c), 1 para $addsd$ (1 c), una para $mulsd$ segmentada en 4 etapas de 1 c cada una, y 1 para saltos (1 c).
Tenga en cuenta que no hay límite en las entradas del ROB y de la ventana de instrucciones centralizada y que se pueden emitir un máximo de 4 i/c. ¿El siguiente código tarda en procesarse 14 ciclos en el cauce descrito?

```
.L6:
 (1) movsd  (%r12,%rax,8), %xmm2  ; xmm2=M[r12+rax*8]
 (2) movsd  0(%rbp,%rax,8), %xmm4 ; xmm4=M[rbp+rax*8]
 (3) mulsd  %xmm1, %xmm2          ; xmm2=xmm2*xmm1
 (4) mulsd  %xmm3, %xmm4          ; xmm4=xmm4*xmm3
 (5) addsd  %xmm4, %xmm2          ; xmm2=xmm2+xmm4
 (6) movsd  %xmm2, 0(%r13,%rax,8) ; M[r13+rax*8]=xmm2
 (7) addq   $1, %rax                ; rax=rax+1
 (8) cmpl   %eax, %ebx            ; eax-ebx
 (9) jg     .L6                   ; Salto si eax-ebx>0
```



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
(1)	IF	ID	EX	EX	WB									
(2)	IF	ID		EX	EX	WB								
(3)	IF	ID			EX	EX	EX	EX	WB	WB				
(4)	IF	ID				EX	EX	EX	EX	WB				
(5)		IF	ID						EX	EX	WB			
(6)		IF	ID								EX	WB		
(7)		IF	ID	EX								WB		
(8)		IF	ID		EX							WB		
(9)			IF	ID		EX						WB		

¿Cuántos ciclos tardaría si ninguna de las unidades funcionales estuviera segmentada?



Predicción de saltos:

→| Código de multiplicación de matrices:

```
for (int i=0; i<10; i++) {          → (1)
    for(int j=0; j<10; j++) {       → (2)
        for (int k=0; k<10; k++) {  → (3)
            ...
        }
    }
}
```

→| Calculemos el número de fallos totales con:

→| Predicción estática: Se toma el salto si el desplazamiento es negativo (hacia atrás)

→| Siempre saltamos hacia atrás, por tanto, nuestra predicción siempre será tomar el salto:

→| Bucle iterado por k:

→| Saltamos 9 veces. No saltamos 1 vez (k=10)

→| Fallamos 1 vez en (3).

→| Bucle iterado por j:

→| ~~Ejecutamos el bucle 10 veces también~~. Saltamos 9, fallamos 1 en (2).

→| Ejecutamos, para cada j, el bucle k, (10 veces)

→| 1 fallo por cada ejecución del bucle k:

→| En total: 10 fallos. (de 3)

→| Bucle iterado por i:

→| Saltamos 9 veces. No saltamos 1. (en 1)

→| Ejecutamos el bucle iterado por j 10 veces.

→| Ejecutamos el bucle iterado por k 100 veces

→| 10 veces por cada j.

→| Fallos en total:

→| 100 fallos del bucle (3)

→| 10 fallos del bucle (2)

→| 1 fallo del bucle (1)

→ Fallos = 100 + 10 + 1 = 111 fallos.

→| Penalización: 111 · (factor)