

# Unidad2problemassoluciones.pdf



**user\_3290738**



**Ingenieria de los computadores**



**3º Grado en Ingeniería Informática**



**Escuela Politécnica Superior  
Universidad de Alicante**

Máster

**Online en Ciberseguridad**

Nº1 en España según El Mundo



**Hasta el 46%  
de beca**



Mejor Máster  
según el  
Ranking de  
ELMUNDO

Para ser el mejor hay que aprender  
de los mejores.

**IMEF**

Smart Education

**Deloitte**

**Infórmate**

# Consigue Empleo o Prácticas

Matricúlate en IMF y accede sin coste a nuestro servicio de Desarrollo Profesional con más de 7.000 ofertas de empleo y prácticas al mes.



## IC Unidad 2: Superescalares - Problemas de ejecución fuera de orden

1. Fragmento de código a ejecutar en procesador superescalar:
- Capaz de captar, decodificar y emitir mediante ventana centralizada 3 instrucciones por ciclo. Asumimos ventana centralizada de 3 líneas
  - 2 UF de suma/resta (2 ciclos) y 1 UF de multiplicación (4 ciclos).
  - ROB que permite retirar hasta 2 instrucciones por ciclo.
- (1) subd f2, f2, f1  
(2) addd f4, f2, f3  
(3) subd f5, f2, f3  
(4) multd f6, f2, f5  
(5) subd f1, f2, f5  
(6) subd f7, f4, f6

a) Emisión ordenada (no alineada)

#Instr	Instrucción	IF	ID/ISS	Ventana*1	EX	ROB*2	WB
1	subd f2, f2, f1	1	2	2	3-4	5	6
2	addd f4, f2, f3	1	2-4	2	5-6	7	8
3	subd f5, f2, f3	1	2-4	2	5-6	7	8
4	multd f6, f2, f5	2	3-6	3	7-10	11	12
5	subd f1, f2, f5	2	3-6	5	7-8	9	12
6	subd f7, f4, f6	2	3-10	5	11-12	13	14

b) Emisión desordenada (alineada)

#Instr	Instrucción	IF	ID/ISS	Ventana*1	EX	ROB*2	WB
1	subd f2, f2, f1	1	2	2	3-4	5	6
2	addd f4, f2, f3	1	2-4	2	5-6	7	8
3	subd f5, f2, f3	1	2-4	2	5-6	7	8
4	multd f6, f2, f5	2	3-6	5	7-10	11	12
5	subd f1, f2, f5	2	3-6	5	7-8	9	12
6	subd f7, f4, f6	2	3-10	5	11-12	13	14

c) Emisión desordenada (no alineada)

#Instr	Instrucción	IF	ID/ISS	Ventana*1	EX	ROB*2	WB
1	subd f2, f2, f1	1	2	2	3-4	5	6
2	addd f4, f2, f3	1	2-4	2	5-6	7	8
3	subd f5, f2, f3	1	2-4	2	5-6	7	8
4	multd f6, f2, f5	2	3-6	3	7-10	11	12
5	subd f1, f2, f5	2	3-6	5	7-8	9	12
6	subd f7, f4, f6	2	3-10	5	11-12	13	14

\*1 Ciclo en el que las instrucciones se insertan en la ventana

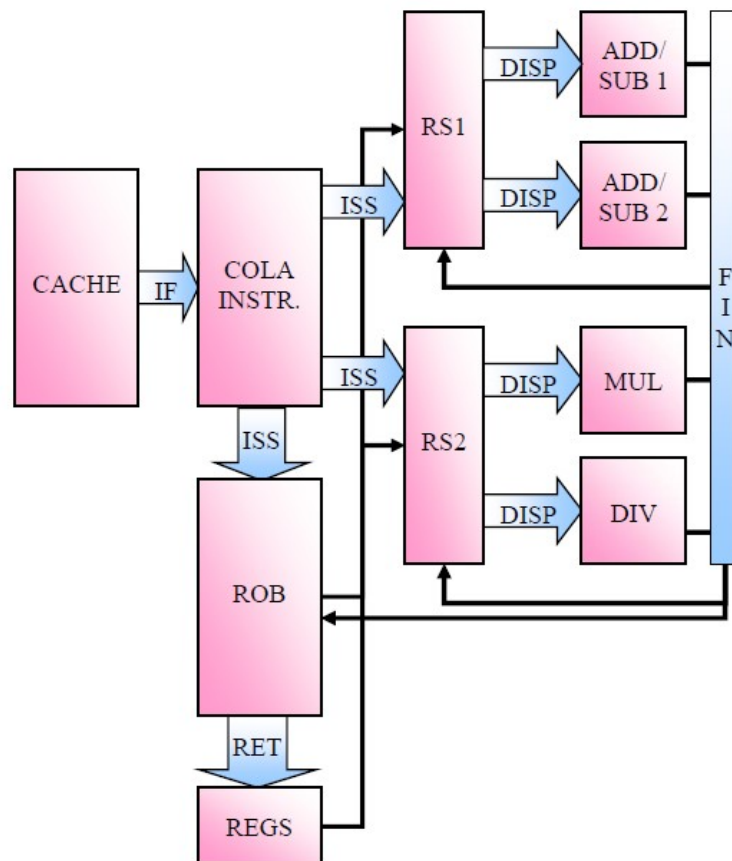
\*2 Ciclo en el que se actualizan los valores de los registros destino en el ROB y en las entradas relacionadas de la ventana de instrucciones

¿Quieres conocer todos los servicios?



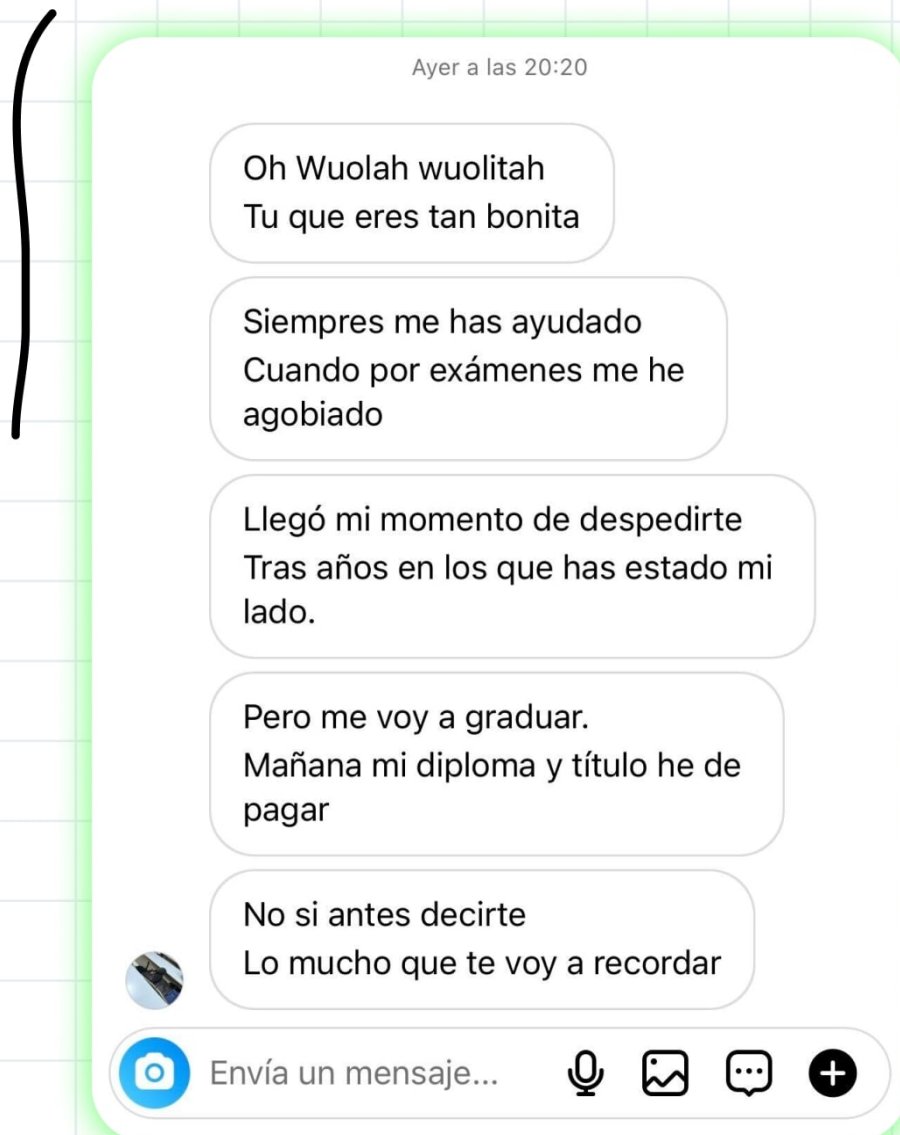
2. En la **cola de instrucciones** encontramos:

- (1) addd f3, f1, f2 ;  $f3 = f1 + f2$
- (2) addd f2, f3, f2 ;  $f2 = f3 + f2$
- (3) multd f4, f3, f2 ;  $f4 = f3 * f2$
- (4) divd f5, f2, f1 ;  $f5 = f2 / f1$
- (5) subd f2, f3, f1 ;  $f2 = f3 - f1$



Instrucción	ID/ISS	DISP/EX	FIN/ROB	RET/WB	Comentario
addd f3, f1, f2	1	2-3	4	5	$f1=10, f2=5 \rightarrow f3=15$
addd f2, f3, f2	1	4-5	6	7	RAW (1) $\rightarrow f2=20$
multd f4, f3, f2	2	6-10	11	12	RAW(1)(2) $\rightarrow f4=300$
divd f5, f2, f1	2	6-45	46	47	RAW(2) $\rightarrow f5=2$
subd f2, f3, f1	3	4-5	6	47	RAW(1) $\rightarrow f2=5$

**Que no te escriban poemas de amor  
cuando terminen la carrera ▶▶▶▶▶▶▶▶**  
(a nosotros por suerte nos pasa) 😊



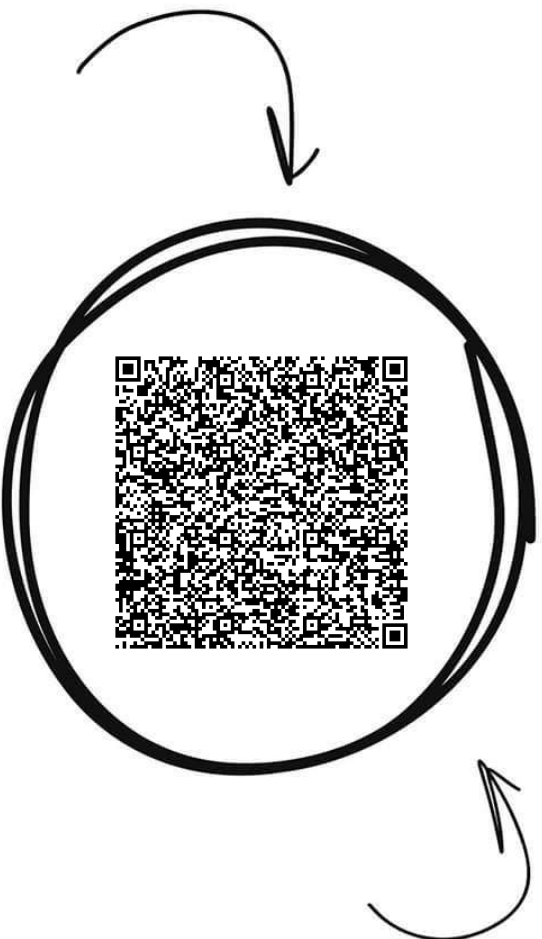
**WUOLAH**



# Ingeniería de los computadores



**Comparte estos flyers en tu clase y consigue más dinero y recompensas**



**Banco de apuntes de la**

**WUOLAH**

- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes

- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR



## IC Unidad 2: Superescalares - Problemas de saltos

- Indicar la penalización efectiva (porcentaje de casos en que falla la predicción) que se produce si se utiliza

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
S	S	N	N	N	S	S	N	S	N	S	N	S	S	S	S	S	N

- Predicción fija (siempre *Not Taken*)

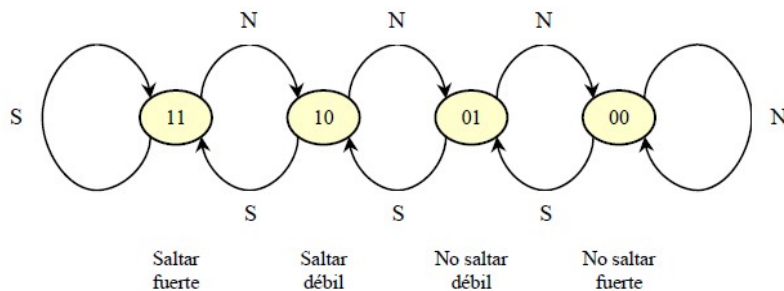
Números de fallos: 11  $P = 11/18 = 0'61 = 61\%$  penalización.

- Predicción estática (si salto atrás *Taken*, si hacia adelante *Not Taken*)

Número de No Tomados = 7 Número de fallos = 7

$P = 7/18 = 0.39 = 39\%$  penalización

- Predicción dinámica con 2 bits (inicialmente en 11)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Bits predicción	11	11	10	01	00	01	10	01	10	01	10	01	10	11	11	11	11	10
Predicción Salto	S	S	S	S	N	N	N	S	N	S	N	S	N	S	S	S	S	S
Resultado Salto	S	S	N	N	N	S	S	N	S	N	S	N	S	S	S	S	S	N
Penalización	0	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	1

$P = 11/18 = 0'61 = 61\%$  penalización

- Predicción dinámica con 3 bits (inicialmente en 111)

Cada entrada guarda las 3 últimas ejecuciones del salto (0 No tomado, 1 Tomado). Se predice según el bit mayoritario. La actualización de los bits de estado se realiza en modo





# Encuentra los mejores lugares para una escapada con... (con amigos, con tu crush, con el casi algo...)

Ven al sur



FIFO, desplazando de bit de mayor peso a bit de menor peso, insertando un 0 si el último salto no ha sido efectivo o un 1 si lo ha sido. Queda la siguiente tabla:

Estado	Salto
000	N
001	N
010	N
011	S
100	N
101	S
110	S
111	S

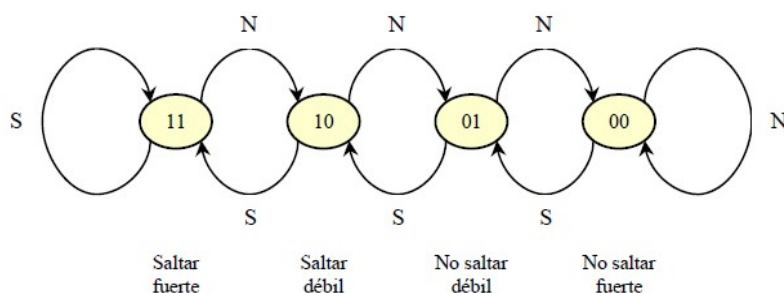
	INICIO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Bits predicción	111	111	111	011	001	000	100	110	011	101	010	101	010	101	110	111	111	111	011
Predicción Salto		S	S	S	S	N	N	N	S	S	S	N	S	N	S	S	S	S	S
Resultado Salto		S	S	N	N	N	S	S	N	S	N	S	N	S	S	S	S	S	N
Penalización		0	0	1	1	0	1	1	1	0	1	1	1	1	0	0	0	0	1

$P = 10/18 = 0'56 = 56\%$  penalización



2. El programa del enunciado calcula el valor absoluto de los elementos del vector  $X$ . Va procesándolos de uno en uno y les cambia el signo si no son positivos. Consta de dos saltos, el primero de ellos, notado como S1 en el código de más abajo, sirve para detectar si los números son positivos y el segundo, notado como S2, se emplea para volver a ejecutar el código mientras queden elementos por procesar en el vector  $X$ .

```
(1) lw r1, N
(2) add r2, r0, r0
(3) bucle: lw r3, X(r2)
(4) sgt r4, r3, r0
(5) bnz r4, mayor S1
(6) sub r3, r0, r3
(7) mayor: sw X(r2), r3
(8) add r2, r2, #4
(9) sub r1, r1, #1
(10) bnz r1, bucle S2
```



Para cada salto, su predictor dinámico se inicializará según el predictor estático del computador, que predice como tomados los saltos hacia atrás. Por tanto, el predictor dinámico para S1 se iniciará al valor 00 (no saltar fuerte), ya que es un salto hacia delante, y el predictor dinámico para S2 se iniciará al valor 11 (saltar fuerte) porque salta hacia atrás.

Una vez inicializados los predictores dinámicos, la penalización que introduzca cada uno de ellos dependerá del comportamiento de cada salto. En el caso de S1, su comportamiento está determinado por el valor de cada elemento del vector  $X$ . S1 Saltará siempre que  $X(i)$  sea mayor que cero, y según el enunciado, el vector  $X$  es de la forma  $X = \{1, 0, 1, 0, 1, 0, \dots\}$ , por lo que S1 saltará una vez sí y otra no hasta que se termine de procesar el vector  $X$ . Por tanto, el comportamiento del predictor dinámico para S1 viene explicado según la siguiente tabla:

Iteración	1	2	3	4	6	7	8	...
Valor de $X$	1	0	1	0	1	0	1	...
Estado actual	00	01	00	01	00	01	00	...
Predicción	N	N	N	N	N	N	N	...
Ejecución	S	N	S	N	S	N	S	...
Penalización	<b>P</b>		<b>P</b>		<b>P</b>		<b>P</b>	...
Estado siguiente	01	00	01	00	01	00	01	...



Por lo que la penalización que introduce S1 es de:

$$Ps1 = 5 \times (N/2) \text{ ciclos}$$

El salto S2 saltará tantas veces como elementos tenga el vector X, es decir N veces, ya que el programa se dedica a procesar dicho vector. Teniendo esto en cuenta, el comportamiento del predictor dinámico para S2 viene explicado según la siguiente tabla:

Iteración	1	2	3	4	...	$N-2$	$N-1$	$N$
Estado actual	11	11	11	11	...	11	11	
Predicción	S	S	S	S	...	S	S	S
Ejecución	S	S	S	S	...	S	S	N
Penalización								<b>P</b>
Estado siguiente	11	11	11	11	...	11	11	10

El comportamiento de este salto es muy fácil de predecir, ya que salta siempre excepto en la última iteración del bucle, por lo que introduce una penalización de:

$$Ps2 = 5 \text{ ciclos}$$

Por tanto, teniendo en cuenta los dos predictores, la penalización total es de:

$$P_{total} = 5 \times (1 + N/2) \text{ ciclos}$$