

# **Validación 3 – Arquitectura de Computadores**

**Juan Francisco Abán Fontecha**



**2015**

## **GUION 8. Captación, Decodificación y Finalización**

**Dado un vector de 30 elementos de doble precisión almacenado en memoria, realizar un programa que ordene dicho vector y lo almacene en la misma posición de memoria.**

*Esta parte la realizare con el código propio, a pesar de que dicho código no consigo la ordenación total como ya comente a Macarena, refiriéndome a que me lo dejara algún compañero para hacerlo.*

*////////////////CODIGO////////////////////////*

*.data 1000*

*.double 1,2,3,4,5,6,7,8,9,10*

*.double 11,12,13,14,15,16,17,18,19,20*

*.double 21,22,23,24,25,26,27,28,29,30*

*.text 256*

*add r1,r0,5*

*Bucle:*

*ld f2,0(r6);Cargo un minimo*

*add r4,r0,0*

*sub r3,r1,1*

*Bucleminimo:*

*ld f4,0(r4)*

*sub r3,r3,1*

*ged f2,f4*

*bfpt min*

*retorno:*

*add r4,r4,8;Sumamos desplazamiento*

*bnez r3,Bucleminimo; condicion de salida*

*sd 0(r10),f2;almacena el minimo*

*add r10,r10,8*

*;FINBUCLEMINIMO*

*sub r1,r1,1*

*add r6,r6,8*

*seq r9,r1,1*

*beqz r9,Bucle*

*sd 0(r10),f4*

*trap #0*

*min:        add f2,f4,f0*

*j retorno*

//////////////////FIN CODIGO//////////////////

- **Partiendo del fichero de configuración, machinefile, calcule los ciclos de ejecución cuando se modifican las siguientes características del procesador superescalar:**  
Sin modificar las características vemos lo siguiente:

General Information			
Number of Cycles:	134		
Instructions Fetched:	118		
Instructions Decoded:	108	91,525	% of total Fetched
Instructions Issued:	108	91,525	% of total Fetched
Integers:	59	50	% of total Issued
Floating Points:	49	41,525	% of total Issued
Instructions Committed:	108	91,525	% of total Fetched
Integers:	59	50	% of total Committed
Floating Points:	49	41,525	% of total Committed
Writes to Registers:	79	73,148	% of total Committed
Useless Writes:	0	0	% of total Writes
<b>Per Cycle Rates:</b> Fetch: 0,8805 Instructions / Cycle Decode: 0,8059 Instructions / Cycle Issue: 0,8059 Instructions / Cycle Commit: 0,8059 Instructions / Cycle Loads Blocked by Stores: 0 0 % of Total Loads Number of branches: 24 Taken: 19 79,166 % Untaken: 5 20,833 %			
Fetch Stalls: 101 75,373 % of Total Cycle Count Decode Stalls: 73 54,477 % of Total Cycle Count Fetch stalls due to full buffers: 0 0 % of Total Stalls			

- Captación, Decodificación y Finalización (1,1,1)

Number of Cycles:	170
Instructions Fetched:	118
Instructions Decoded:	108 91,525 % of total Fetched

- Captación, Decodificación y Finalización (2,2,2)

Number of Cycles:	134
Instructions Fetched:	118
Instructions Decoded:	108 91,525 % of total Fetched

- Captación, Decodificación y Finalización (3,3,3)

Number of Cycles:	129		
Instructions Fetched:	118		
Instructions Decoded:	108	91,525	% of total Fetched

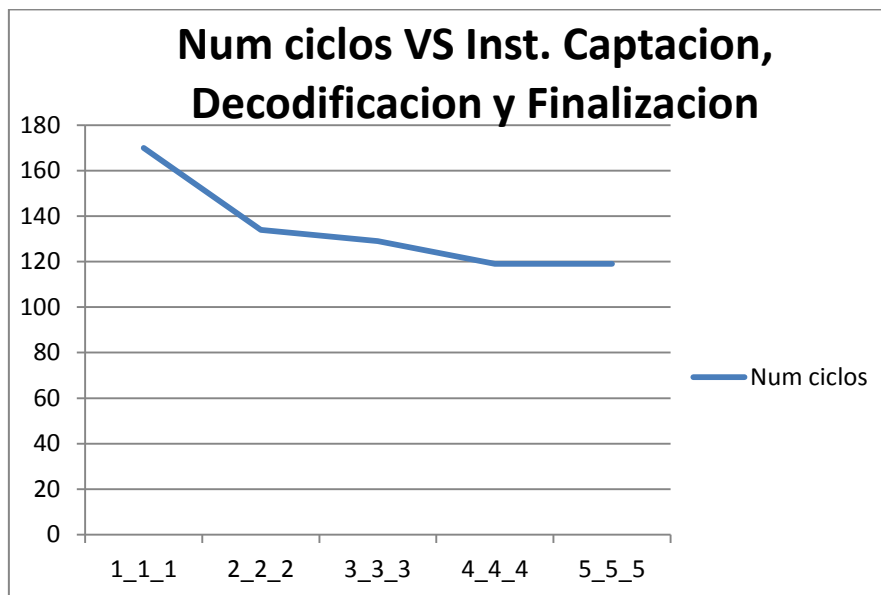
- Captación, Decodificación y Finalización (4,4,4)

Number of Cycles:	119		
Instructions Fetched:	118		
Instructions Decoded:	108	91,525	% of total Fetched

- Captación, Decodificación y Finalización (5,5,5)

Number of Cycles:	119		
Instructions Fetched:	118		
Instructions Decoded:	108	91,525	% of total Fetched

- **Elabore una gráfica donde se ilustre el comportamiento en ciclos del procesador superescalar y comente los resultados.**



Esta grafica como vemos tiene un gran descenso en ciclos ejecución cuando pasa de tener 1 ciclo para Captación, Decodificación y Finalización a 2 por cada uno, pero en el resto vemos como poco a poco se va estabilizando, hasta llegar al momento de que es igual cuando tengamos 4 y 5 ciclos respectivamente.

- **Calcule los ciclos de ejecución con las siguientes características del procesador superescalar, comentando los resultados alcanzados:**
  - **Captación, Decodificación y Finalización (2,4,2)**

Number of Cycles:	<input type="text" value="134"/>		
Instructions Fetched:	<input type="text" value="118"/>		
Instructions Decoded:	<input type="text" value="108"/>	<input type="text" value="91,525"/>	% of total Fetched

- **Captación, Decodificación y Finalización (1,5,3)**

Number of Cycles:	<input type="text" value="170"/>		
Instructions Fetched:	<input type="text" value="118"/>		
Instructions Decoded:	<input type="text" value="108"/>	<input type="text" value="91,525"/>	% of total Fetched

Como vemos en los resultados de ambos, se emplea menos ciclos en ejecución en el primero, esto es debido a que la captación es mayor, y al captar una sola instrucción se realiza un cuello de botella, que no permite decodificar bien el resto de instrucciones sino que conforme las va captando.

- **Busca una combinación de ciclos de Captación, Decodificación y Finalización que genere un numero de ciclos en ejecución adecuado, buscando optimizar recursos.**

Comparando la mejor combinación optimizando recursos es 4,3,2 de forma que obtenemos los siguientes resultados:

Number of Cycles:	<input type="text" value="119"/>		
Instructions Fetched:	<input type="text" value="118"/>		
Instructions Decoded:	<input type="text" value="108"/>	<input type="text" value="91,525"/>	% of total Fetched

## **GUION 9. Cola de instrucción (CI), Ventana de Instrucción (VI) y Buffer de Reorden(ROB).**

Partiendo del guion anterior.

- Calcule los ciclos de ejecución cuando se modifican las siguientes características del procesador superescalar:

- Captación, Decodificación y Finalización (5,5,5) “Siempre”

- CI,VI y ROB (4,4,4)

Number of Cycles:	123		
Instructions Fetched:	118		
Instructions Decoded:	108	91,525	% of total Fetched

- CI,VI y ROB (8,8,8)

Number of Cycles:	119		
Instructions Fetched:	118		
Instructions Decoded:	108	91,525	% of total Fetched

- CI,VI y ROB (12,12,12)

Number of Cycles:	119		
Instructions Fetched:	118		
Instructions Decoded:	108	91,525	% of total Fetched

- CI,VI y ROB (16,16,16)

Number of Cycles:	119		
Instructions Fetched:	118		
Instructions Decoded:	108	91,525	% of total Fetched

- CI,VI y ROB (20,20,20)

Number of Cycles:	119		
Instructions Fetched:	118		
Instructions Decoded:	108	91,525	% of total Fetched

- **Calcule los ciclos de ejecución cuando se modifican las siguientes características del procesador superescalar:**
  - **Captación, Decodificación y Finalización (5,5,5) “Siempre”**
  - **CI: 16 “Siempre”**
    - **VI y ROB para enteros (4,4) VI y ROB para flotantes (8,8)**

Number of Cycles:	<input type="text" value="123"/>	
Instructions Fetched:	<input type="text" value="118"/>	
Instructions Decoded:	<input type="text" value="108"/>	<input type="text" value="91,525"/> % of total Fetched

- **VI y ROB para enteros (4,4) VI y ROB para flotantes (16,16)**

Number of Cycles:	<input type="text" value="123"/>	
Instructions Fetched:	<input type="text" value="118"/>	
Instructions Decoded:	<input type="text" value="108"/>	<input type="text" value="91,525"/> % of total Fetched

- **VI y ROB para enteros (8,8) VI y ROB para flotantes (16,16)**

Number of Cycles:	<input type="text" value="119"/>	
Instructions Fetched:	<input type="text" value="118"/>	
Instructions Decoded:	<input type="text" value="108"/>	<input type="text" value="91,525"/> % of total Fetched

- **VI y ROB para enteros (8,8) VI y ROB para flotantes (4,4)**

Number of Cycles:	<input type="text" value="119"/>	
Instructions Fetched:	<input type="text" value="118"/>	
Instructions Decoded:	<input type="text" value="108"/>	<input type="text" value="91,525"/> % of total Fetched

- **VI y ROB para enteros (16,16) VI y ROB para flotantes (4,4)**

Number of Cycles:	<input type="text" value="119"/>	
Instructions Fetched:	<input type="text" value="118"/>	
Instructions Decoded:	<input type="text" value="108"/>	<input type="text" value="91,525"/> % of total Fetched

- **VI y ROB para enteros (16,16) VI y ROB para flotantes (8,8)**

Number of Cycles:	<input type="text" value="119"/>	
Instructions Fetched:	<input type="text" value="118"/>	
Instructions Decoded:	<input type="text" value="108"/>	<input type="text" value="91,525"/> % of total Fetched

- **Busque una combinación de numero de instrucciones captadas, decodificadas y finalizadas por ciclo junto con el tamaño de la cola de instrucción, ventana de instrucción(enteros y flotantes) y buffer de reorden (enteros y flotantes) que genere un numero de ciclos de ejecución adecuado, buscando optimizar los recursos.**

La combinación de numero de instrucciones captadas, decodificadas y finalizadas es de (4,3,2) por ciclo, junto con el tamaño de cola de instrucción 4, ventana de instrucción de enteros 8 y de flotantes 4, y buffer de reorden de enteros de 8 y de flotantes 4, consiguiendo así el siguiente resultado:

Statistics							
General Information		Renaming Information		Instruction Process		Occupancy Rate	
Number of Cycles:	119						
Instructions Fetched:	118						
Instructions Decoded:	108	91,525	% of total Fetched				
Instructions Issued:	108	91,525	% of total Fetched				
Integers:	59	50	% of total Issued				
Floating Points:	49	41,525	% of total Issued				
Instructions Committed:	108	91,525	% of total Fetched				
Integers:	59	50	% of total Committed				
Floating Points:	49	41,525	% of total Committed				
Writes to Registers:	79	73,148	% of total Committed				
Useless Writes:	0	0	% of total Writes				
				Per Cycle Rates:			
				Fetch:	0,9915	Instructions / Cycle	
				Decode:	0,9075	Instructions / Cycle	
				Issue:	0,9075	Instructions / Cycle	
				Commit:	0,9075	Instructions / Cycle	
				Loads Blocked by Stores:			
				0		%	
				0		% of Total Loads	
				Number of branches:			
				24			
				Taken:	19	79,166	%
				Untaken:	5	20,833	%
				Fetch Stalls:			
				85		71,428	% of Total Cycle Count
				Decode Stalls:			
				76		63,865	% of Total Cycle Count
				Fetch stalls due to full buffers:			
				0		0	% of Total Stalls



## **GUION 10. Predicción de saltos y Coste.**

- **Partiendo del guión anterior**
  - **Repetir guiones 8 y 9. Activando la opción de predicción de saltos y comente los resultados.**

Repitiendo todos los ejercicios anteriores podemos observar como la predicción de salto nos ahorra muchos ciclos, vamos a ver el cambio brusco que pega, por ejemplo con la configuración ideal que habíamos cogido como gran optimización en el ultimo ejercicio del guion 9:

Number of Cycles:	78	
Instructions Fetched:	141	
Instructions Decoded:	111	78,723 % of total Fetched

Como vemos el numero de ciclos en ejecución pasa de 119 a 78 lo que es una gran optimización gracias a la predicción de saltos.

- **A partir de los resultados obtenidos, busque una configuración que permita obtener el mayor rendimiento teniendo en cuenta los siguientes costes y estando limitado al gasto de 35€**
  - **Predicción de saltos: 15€**
  - **Cada línea de captación, decodificación y finalización:**
    - 1 inst/ciclo: 0€
    - 2 inst/ciclo: 3€
    - 3 inst/ciclo: 4€
    - 4 inst/ciclo: 5€
    - 5 inst/ciclo: 6€
  - **Tamaño de CI,VI(enteros-flotantes) y ROB (enteros-flotantes)**
    - 4 líneas: 0€
    - 8 líneas: 3€
    - 12 líneas: 4€
    - 16 líneas: 5€
    - 20 líneas: 6€

Cogiendo la configuración ideal anterior obtenemos:

Number of Cycles:	78	
Instructions Fetched:	141	
Instructions Decoded:	111	78,723 % of total Fetched

- Captación, Decodificación y Finalización (4,3,2):  $5+4+3=12€$
- CI: 4: 0€
- VI(enteros y flotantes): 8,4: 3€

## Arquitectura de Computadores

- ROB(enteros y flotantes): 8,4: 3€
- Predicción de salto: 15€

Este presupuesto nos haría un total de 33€, nos pasaríamos del presupuesto, por lo que no nos sería de utilidad.

Si cambiamos algunas propiedades nos quedaría así:

Number of Cycles:	<input type="text" value="82"/>	
Instructions Fetched:	<input type="text" value="140"/>	
Instructions Decoded:	<input type="text" value="110"/>	<input type="text" value="78,571"/> % of total Fetched

- Captación, Decodificación y Finalización (4,3,2): 5+4+3=12€
- CI: 4: 0€
- VI(enteros y flotantes): 4,4: 0€
- ROB(enteros y flotantes): 4,4: 0€
- predicción de salto: 15€

Este presupuesto nos haría un total de 27€, no nos pasaríamos del presupuesto, por lo que nos sería de utilidad dado al rendimiento que podemos obtener.

Si cambiamos algunas propiedades nos quedaría así:

Number of Cycles:	<input type="text" value="78"/>	
Instructions Fetched:	<input type="text" value="141"/>	
Instructions Decoded:	<input type="text" value="111"/>	<input type="text" value="78,723"/> % of total Fetched

- Captación, Decodificación y Finalización (4,3,2): 5+4+3=12€
- CI: 4: 0€
- VI(enteros y flotantes): 4,4: 0€
- ROB(enteros y flotantes): 8,4: 3€
- predicción de salto: 15€

Este presupuesto nos haría un total de 30€, no nos pasaríamos del presupuesto, por lo que nos sería de utilidad dado al rendimiento que podemos obtener esta configuración sería la ideal, de cara al rendimiento obtenido y al precio del mismo.

## GUION 8. Captación, Decodificación y Finalización

Dado un vector de 30 elementos de doble precisión almacenado en memoria, realizar un programa que ordene dicho vector y lo almacene en la misma posición de memoria.

Esta parte la realizare con el código obtenido en la página web <http://apuntes.ujacraft.es/>, dicho código no lo he revisado simplemente copiado de cara hacer esta practica, ya que no me interesa copiar código sino saber realizarlo por mi mismo.

////////////////CODIGO////////////////

```
.data 1000

.global numbers

numbers: .word 11

        .word 8,10,13,2,5,3,4,1,6,10

        .word 11,20,3,8,19,12,8,10,19,20

        .word 21,1,3,45,67,5,65,67,65,32

.global tam

tam:     .word 40

        .text 256

        add r15,r0,tam                ; r15=[tam]

        lw r15,(r15)                  ; r15=tam;

        addi r2,r0,#4                 ; r2=i=4

        addi r1,r0,numbers            ; r1=[numbers]

BUCLE1:  add r6,r1,r2                  ; r6=[numbers[i]]

        lw r4,(r6)                    ; r4=numbers[i]

        subi r3,r2,4                  ; r3=a=i-1

BUCLE2:  sge r12,r3,0                  ; a>=0

        ;BEQZ r12,BUCLE1CONT

        ;ADD r6,r1,r3                  ; [r6=numbers[a]]

        ;LW r10,(r6)                  ; r10=numbers[a]

        ;SGT r12,r10,r4                ; numbers[a]>numbers[i]

        ;beqz r12,BUCLE1CONT
```

## Arquitectura de Computadores

```

;ADDI r7,r3,#4                                ; r7=a+1

;ADD r5,r1,r3                                ; r5=[numbers[a]]

;ADD r7,r1,r7                                ; r7=[numbers[a+1]]

;LW r6,(r5)                                    ; r6=[numbers[a]]

;SW (r7),r6                                    ; r7=[numbers[a]]

;SUBI r3,r3,#4                                ; a=a-1

;SGE r12,r3,r0                                ; comprobamos si a =0

;bnez r12,BUCLE2

BUCLE1CONT:  addi r7,r3,#4                    ; a=a+1

            add r7,r1,r7                    ; r7=[numbers[a+1]]

            sw (r7),r4                    ; numbers[a+1]=index(numbers[i])

            addi r2,r2,#4                    ; i++

            slt r12,r2,r15                ; si i es menor que 40 salta a bucle, sino fin

            bnez r12,BUCLE1 ;

FIN:        trap #0 ;
  
```

//////////////////FIN CODIGO //////////////////////

**Partiendo del fichero de configuración, machinefile, calcule los ciclos de ejecución cuando se modifican las siguientes características del procesador superescalar:**

Sin modificar las características vemos lo siguiente:

General Information	Renaming Information	Instruction Process	Occupancy Rate
Number of Cycles: 114			
Instructions Fetched: 94			
Instructions Decoded: 94	100 % of total Fetched		
Instructions Issued: 94	100 % of total Fetched		
Integers: 94	100 % of total Issued		
Floating Points: 0	0 % of total Issued		
Instructions Committed: 94	100 % of total Fetched		
Integers: 94	100 % of total Committed		
Floating Points: 0	0 % of total Committed		
Writes to Registers: 76	80,851 % of total Committed		
Useless Writes: 0	0 % of total Writes		
			Per Cycle Rates:
			Fetch: 0,8245 Instructions / Cycle
			Decode: 0,8245 Instructions / Cycle
			Issue: 0,8245 Instructions / Cycle
			Commit: 0,8245 Instructions / Cycle
			Loads Blocked by Stores: 0
			0 % of Total Loads
			Number of branches: 9
			Taken: 8 88,888 %
			Untaken: 1 11,111 %
			Fetch Stalls: 20 17,543 % of Total Cycle Count
			Decode Stalls: 20 17,543 % of Total Cycle Count
			Fetch stalls due to full buffers: 0 0 % of Total Stalls

- Captación, Decodificación y Finalización (1,1,1)

Number of Cycles:	115		
Instructions Fetched:	94		
Instructions Decoded:	94	100	% of total Fetched

- Captación, Decodificación y Finalización (2,2,2)

Number of Cycles:	77		
Instructions Fetched:	94		
Instructions Decoded:	94	100	% of total Fetched

- Captación, Decodificación y Finalización (3,3,3)

Number of Cycles:	66		
Instructions Fetched:	94		
Instructions Decoded:	94	100	% of total Fetched

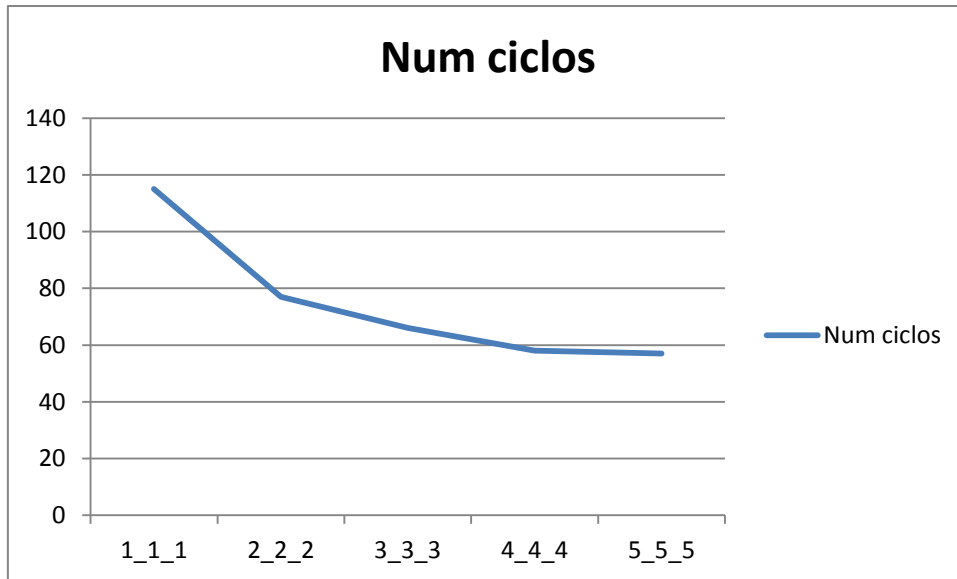
- Captación, Decodificación y Finalización (4,4,4)

Number of Cycles:	58		
Instructions Fetched:	94		
Instructions Decoded:	94	100	% of total Fetched

- Captación, Decodificación y Finalización (5,5,5)

Number of Cycles:	57		
Instructions Fetched:	94		
Instructions Decoded:	94	100	% of total Fetched

- **Elabore una gráfica donde se ilustre el comportamiento en ciclos del procesador superescalar y comente los resultados.**



Esta grafica como vemos tiene un gran descenso en ciclos ejecución cuando pasa de tener 1 ciclo para Captación, Decodificación y Finalización a 2 por cada uno, pero en el resto vemos como se estabiliza, poco a poco, conforme van aumentando el numero de instrucciones captadas, decodificadas y finalizadas por ciclo.

- **Calcule los ciclos de ejecución con las siguientes características del procesador superescalador, comentando los resultados alcanzados:**
  - **Captación, Decodificación y Finalización (2,4,2)**

Number of Cycles:	<input type="text" value="77"/>		
Instructions Fetched:	<input type="text" value="94"/>		
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/>	% of total Fetched

- **Captación, Decodificación y Finalización (1,5,3)**

Number of Cycles:	<input type="text" value="114"/>		
Instructions Fetched:	<input type="text" value="94"/>		
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/>	% of total Fetched

Como vemos en los resultados de ambos, se emplea menos ciclos en ejecución en el primero, esto es debido a que la captación es mayor, y al captar una sola instrucción se realiza un cuello de botella, que no permite decodificar bien el resto de instrucciones sino que conforme las va captando.

- **Busca una combinación de ciclos de Captación, Decodificación y Finalización que genere un numero de ciclos en ejecución adecuado, buscando optimizar recursos.**

Comparando la mejor combinación optimizando recursos es 4,3,3 de forma que obtenemos los siguientes resultados:

Number of Cycles:	<input type="text" value="66"/>		
Instructions Fetched:	<input type="text" value="94"/>		
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/>	% of total Fetched

## **GUION 9. Cola de instrucción (CI), Ventana de Instrucción (VI) y Buffer de Reorden(ROB).**

Partiendo del guion anterior.

- Calcule los ciclos de ejecución cuando se modifican las siguientes características del procesador superescalar:

- Captación, Decodificación y Finalización (5,5,5) “Siempre”

- CI,VI y ROB (4,4,4)

Number of Cycles:	<input type="text" value="105"/>		
Instructions Fetched:	<input type="text" value="94"/>		
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/>	% of total Fetched

- CI,VI y ROB (8,8,8)

Number of Cycles:	<input type="text" value="76"/>		
Instructions Fetched:	<input type="text" value="94"/>		
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/>	% of total Fetched

- CI,VI y ROB (12,12,12)

Number of Cycles:	<input type="text" value="58"/>		
Instructions Fetched:	<input type="text" value="94"/>		
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/>	% of total Fetched

- CI,VI y ROB (16,16,16)

Number of Cycles:	<input type="text" value="57"/>		
Instructions Fetched:	<input type="text" value="94"/>		
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/>	% of total Fetched

- CI,VI y ROB (20,20,20)

Number of Cycles:	<input type="text" value="57"/>		
Instructions Fetched:	<input type="text" value="94"/>		
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/>	% of total Fetched



- **Calcule los ciclos de ejecución cuando se modifican las siguientes características del procesador superescalar:**
  - **Captación, Decodificación y Finalización (5,5,5) “Siempre”**
  - **CI: 16 “Siempre”**
    - **VI y ROB para enteros (4,4) VI y ROB para flotantes (8,8)**

Number of Cycles:	<input type="text" value="105"/>	
Instructions Fetched:	<input type="text" value="94"/>	
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/> % of total Fetched

- **VI y ROB para enteros (4,4) VI y ROB para flotantes (16,16)**

Number of Cycles:	<input type="text" value="105"/>	
Instructions Fetched:	<input type="text" value="94"/>	
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/> % of total Fetched

- **VI y ROB para enteros (8,8) VI y ROB para flotantes (16,16)**

Number of Cycles:	<input type="text" value="76"/>	
Instructions Fetched:	<input type="text" value="94"/>	
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/> % of total Fetched

- **VI y ROB para enteros (8,8) VI y ROB para flotantes (4,4)**

Number of Cycles:	<input type="text" value="76"/>	
Instructions Fetched:	<input type="text" value="94"/>	
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/> % of total Fetched

- **VI y ROB para enteros (16,16) VI y ROB para flotantes (4,4)**

Number of Cycles:	<input type="text" value="57"/>	
Instructions Fetched:	<input type="text" value="94"/>	
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/> % of total Fetched

- **VI y ROB para enteros (16,16) VI y ROB para flotantes (8,8)**

Number of Cycles:	<input type="text" value="57"/>	
Instructions Fetched:	<input type="text" value="94"/>	
Instructions Decoded:	<input type="text" value="94"/>	<input type="text" value="100"/> % of total Fetched

- **Busque una combinación de numero de instrucciones captadas, decodificadas y finalizadas por ciclo junto con el tamaño de la cola de instrucción, ventana de instrucción(enteros y flotantes) y buffer de reorden (enteros y flotantes) que genere un numero de ciclos de ejecución adecuado, buscando optimizar los recursos.**

La combinación de numero de instrucciones captadas, decodificadas y finalizadas es de (4,3,4) por ciclo, junto con el tamaño de cola de instrucción 4, ventana de instrucción de enteros 8 y de flotantes 4, y buffer de reorden de enteros de 8 y de flotantes 4, consiguiendo así el siguiente resultado:

General Information	Renaming Information	Instruction Process	Occupancy Rate
<p>Number of Cycles: <input type="text" value="76"/></p>			
<p>Instructions Fetched: <input type="text" value="94"/></p>			
<p>Instructions Decoded: <input type="text" value="94"/> <input type="text" value="100"/> % of total Fetched</p>			
<p>Instructions Issued: <input type="text" value="94"/> <input type="text" value="100"/> % of total Fetched</p>			
<p>Integers: <input type="text" value="94"/> <input type="text" value="100"/> % of total Issued</p>			
<p>Floating Points: <input type="text" value="0"/> <input type="text" value="0"/> % of total Issued</p>			
<p>Instructions Committed: <input type="text" value="94"/> <input type="text" value="100"/> % of total Fetched</p>			
<p>Integers: <input type="text" value="94"/> <input type="text" value="100"/> % of total Committed</p>			
<p>Floating Points: <input type="text" value="0"/> <input type="text" value="0"/> % of total Committed</p>			
<p>Writes to Registers: <input type="text" value="76"/> <input type="text" value="80,851"/> % of total Committed</p>			
<p>Useless Writes: <input type="text" value="0"/> <input type="text" value="0"/> % of total Writes</p>			
<p>Per Cycle Rates:</p>			
<p>Fetch: <input type="text" value="1,2368"/> Instructions / Cycle</p>			
<p>Decode: <input type="text" value="1,2368"/> Instructions / Cycle</p>			
<p>Issue: <input type="text" value="1,2368"/> Instructions / Cycle</p>			
<p>Commit: <input type="text" value="1,2368"/> Instructions / Cycle</p>			
<p>Loads Blocked by Stores: <input type="text" value="0"/> <input type="text" value="0"/> % of Total Loads</p>			
<p>Number of branches: <input type="text" value="9"/></p>			
<p>Taken: <input type="text" value="8"/> <input type="text" value="88,888"/> %</p>			
<p>Untaken: <input type="text" value="1"/> <input type="text" value="11,111"/> %</p>			
<p>Fetch Stalls: <input type="text" value="46"/> <input type="text" value="60,526"/> % of Total Cycle Count</p>			
<p>Decode Stalls: <input type="text" value="29"/> <input type="text" value="38,157"/> % of Total Cycle Count</p>			
<p>Fetch stalls due to full buffers: <input type="text" value="1"/> <input type="text" value="2,1739"/> % of Total Stalls</p>			
<p><input type="button" value="OK"/></p>			

## **GUION 10. Predicción de saltos y Coste.**

- **Partiendo del guión anterior**
  - **Repetir guiones 8 y 9. Activando la opción de predicción de saltos y comentar los resultados.**

Repetiendo todos los ejercicios anteriores podemos observar como la predicción de salto nos ahorra muchos ciclos, vamos a ver el cambio brusco que pega, por ejemplo con la configuración ideal que habíamos cogido como gran optimización en el ultimo ejercicio del guion 9:

Number of Cycles:	52		
Instructions Fetched:	100		
Instructions Decoded:	96	96	% of total Fetched

Como vemos el número de ciclos en ejecución pasa de 76 a 52 lo que es una gran optimización gracias a la predicción de saltos.

- **A partir de los resultados obtenidos, busque una configuración que permita obtener el mayor rendimiento teniendo en cuenta los siguientes costes y estando limitado al gasto de 35€**
  - **Predicción de saltos: 15€**
  - **Cada línea de captación, decodificación y finalización:**
    - 1 inst/ciclo: 0€
    - 2 inst/ciclo: 3€
    - 3 inst/ciclo: 4€
    - 4 inst/ciclo: 5€
    - 5 inst/ciclo: 6€
  - **Tamaño de CI, VI (enteros-flotantes) y ROB (enteros-flotantes)**
    - 4 líneas: 0€
    - 8 líneas: 3€
    - 12 líneas: 4€
    - 16 líneas: 5€
    - 20 líneas: 6€

Cogiendo la configuración ideal anterior obtenemos:

Number of Cycles:	52		
Instructions Fetched:	100		
Instructions Decoded:	96	96	% of total Fetched

- Captación, Decodificación y Finalización (4,3,4):  $5+4+5=14€$
- CI: 4: 0€
- VI (enteros y flotantes): 8,4: 3€
- ROB (enteros y flotantes): 8,4: 3€

- Predicción de salto: 15€

Este presupuesto nos haría un total de 36€, nos pasaríamos del presupuesto, por lo que no nos sería de utilidad.

Si cambiamos algunas propiedades nos quedaría así:

Number of Cycles:	53		
Instructions Fetched:	101		
Instructions Decoded:	97	96,039	% of total Fetched

- Captación, Decodificación y Finalización (4,3,2): 5+4+3=12€
- CI: 4: 0€
- VI(enteros y flotantes): 8,4: 3€
- ROB(enteros y flotantes): 8,4: 3€
- predicción de salto: 15€

Este presupuesto nos haría un total de 33€, nos pasaríamos del presupuesto, por lo que no nos sería de utilidad dado al rendimiento que podemos obtener, a pesar de que solo es 1€.

Si cambiamos algunas propiedades nos quedaría así:

Number of Cycles:	53		
Instructions Fetched:	101		
Instructions Decoded:	97	96,039	% of total Fetched

- Captación, Decodificación y Finalización (4,3,2): 5+4+3=12€
- CI: 4: 0€
- VI(enteros y flotantes): 4,4: 0€
- ROB(enteros y flotantes): 8,4: 3€
- predicción de salto: 15€

Este presupuesto nos haría un total de 30€, no nos pasaríamos del presupuesto, por lo que nos sería de utilidad dado al rendimiento que podemos obtener esta configuración sería la ideal, de cara al rendimiento obtenido y al precio del mismo.