

Grado en Ingeniería Informática

Arquitectura de Computadores
Curso 2017/18

Bloque 3: Procesador SuperDLX



UNIVERSIDAD DE JAÉN

Autor: Manuel Alférez Ruiz

1. Captación, decodificación y finalización

Código:

```
.data
.global v
v: .double 1,2,3,4,5,6,7,8,9,10,2,5,6,4,7,8,9,3,1,4,6,11,4,5,7,9,8,4,6,10

.global i
i: .word 29
.global j
j: .word 29

.text

add r1,r0,v      ;Puntero al menor
add r2,r0,v      ;Puntero para recorrer el vector
add r3,r0,i
lw  r4,0(r3)     ;Contador i=2
add r3,r0,j
lw  r5,0(r3)     ;Contador j=2

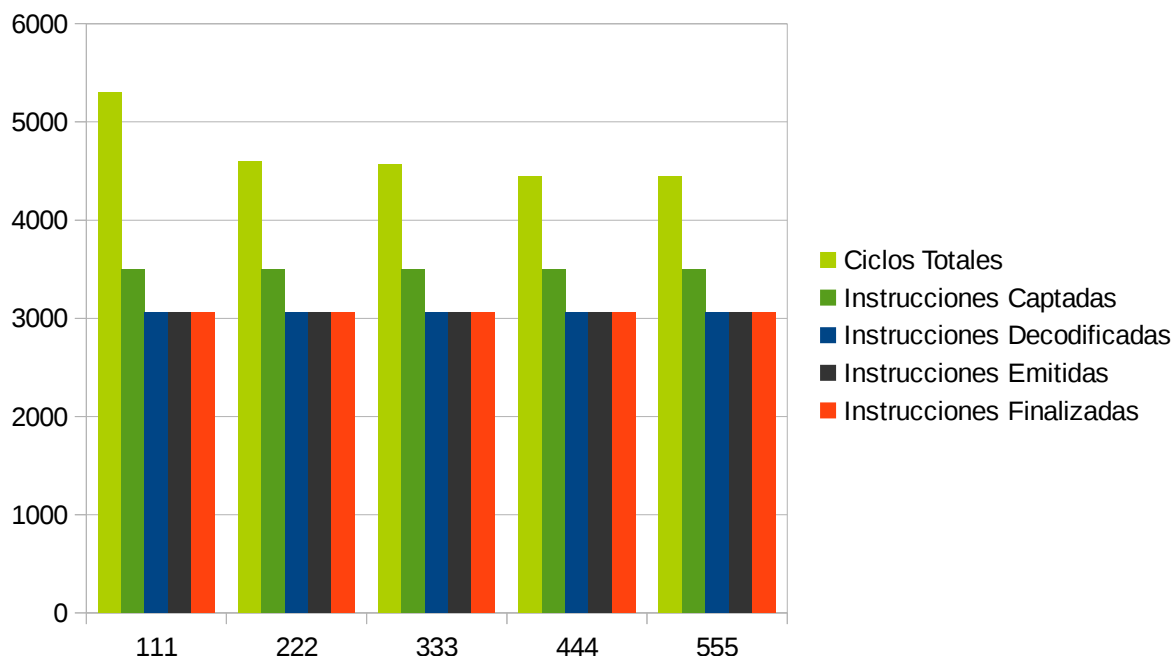
bucle1:
ld  f0,0(r1)     ;Cargo en f0->el valor del puntero r1
addi r2,r1,#8    ;R2-> Apunta al r1 (modificado anteriormente)
beqz r4,fin      ;Comprobamos si es cero
add r6,r5,r0;Para ir decrementando r6

bucle2:
ld  f2,0(r2)     ;Cargamos en f2-> el valor del puntero r2
subi r6,r6,#1    ;Decrementamos j
gtd f0,f2        ;f0>f2-> 1 (hay que cambiar valores)
bfpf no_cambia
sd  0(r1),f2     ;Almacenamos f2->donde apunta r1
sd  0(r2),f0     ;Almacenamos f0->donde apunta r2
ld  f0,0(r1)
no_cambia:
beqz r6,decrementa ;Fin de bucle de j
addi r2,r2,#8    ;Apuntamos al siguiente dato
j   bucle2

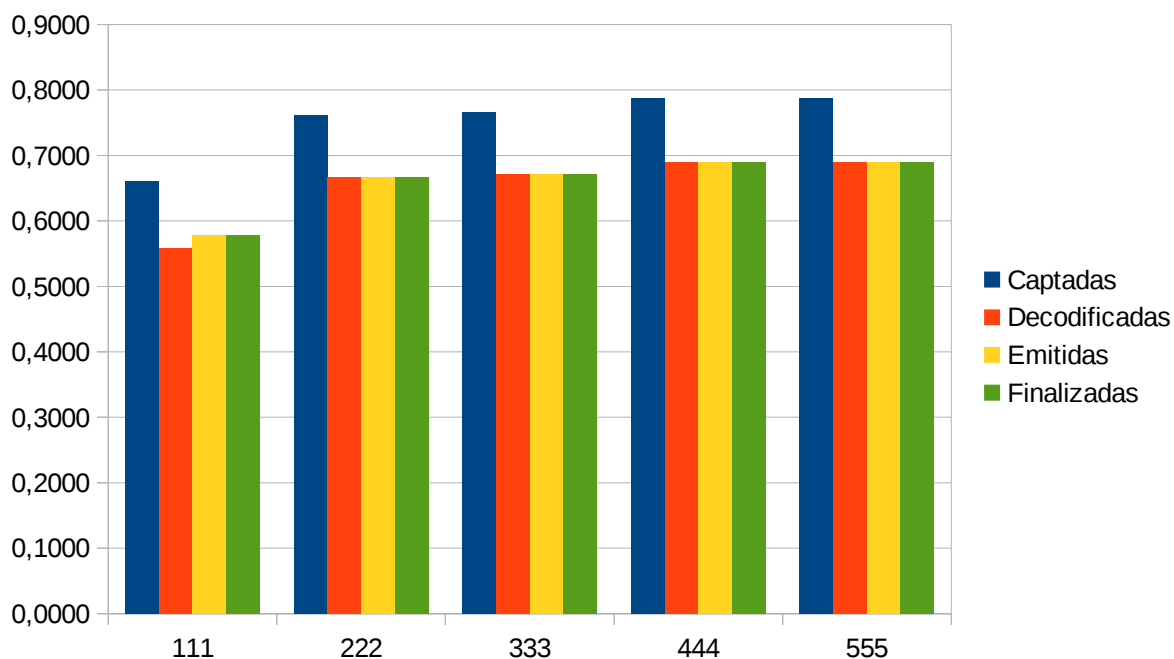
decrementa:
subi r5,r5,#1    ;Decrementamos el contador j en uno
subi r4,r4,#1    ;Decrementamos contador i
addi r1,r1,#8    ;Apunto al siguiente dato r1
j   bucle1

fin: trap #0
```

A continuación, partiendo del fichero de configuración, **machinefile**, calculamos los ciclos de ejecución cuando se modifican las características del **procesador superescalar**. En primer lugar, se muestra una gráfica que muestra los *ciclos totales*, las *instrucciones captadas*, *decodificadas*, *emitidas* y *finalizadas*:



Gráfica de las *instrucciones por unidad de ciclo* para cada una de las configuraciones anteriores:

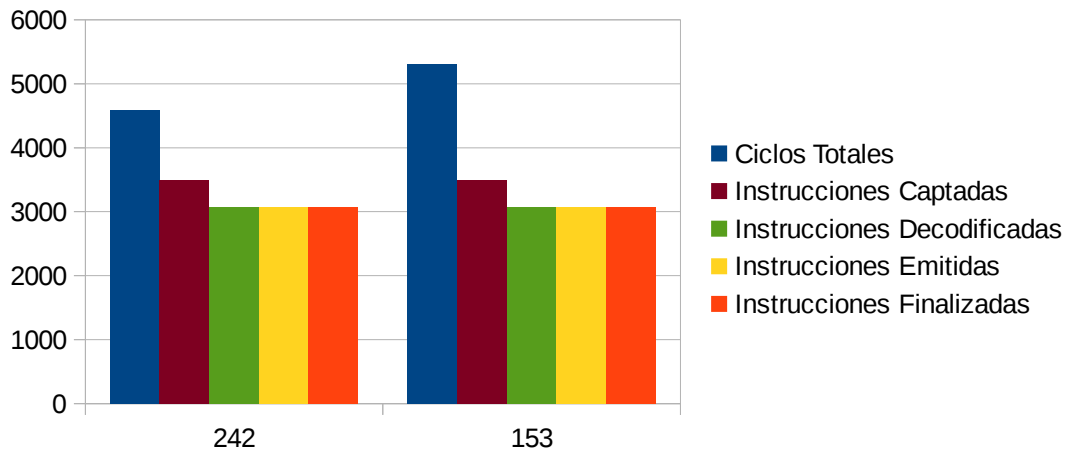


Como se puede observar en las gráficas, se llega a un **punto** en el que a pesar de que se capte, decodifique y se finalice más instrucciones, **no disminuyen los ciclos totales**.

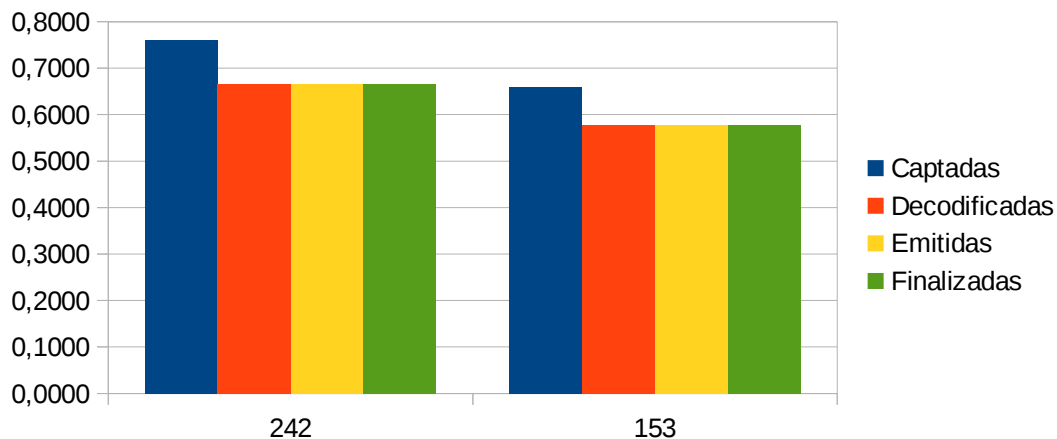
Al igual sucede con las instrucciones captadas, decodificadas, emitidas y finalizadas por ciclo, tampoco se consigue que se puedan incrementar estos valores.

Observamos que a partir de la configuración **(4,4,4)**, no se consigue mejoría alguna; por lo que es inútil seguir aumentando las características del procesador.

A continuación, se efectúan otras modificaciones a las características del procesador:



Gráfica de las *instrucciones por unidad de ciclo* para cada una de las configuraciones anteriores:

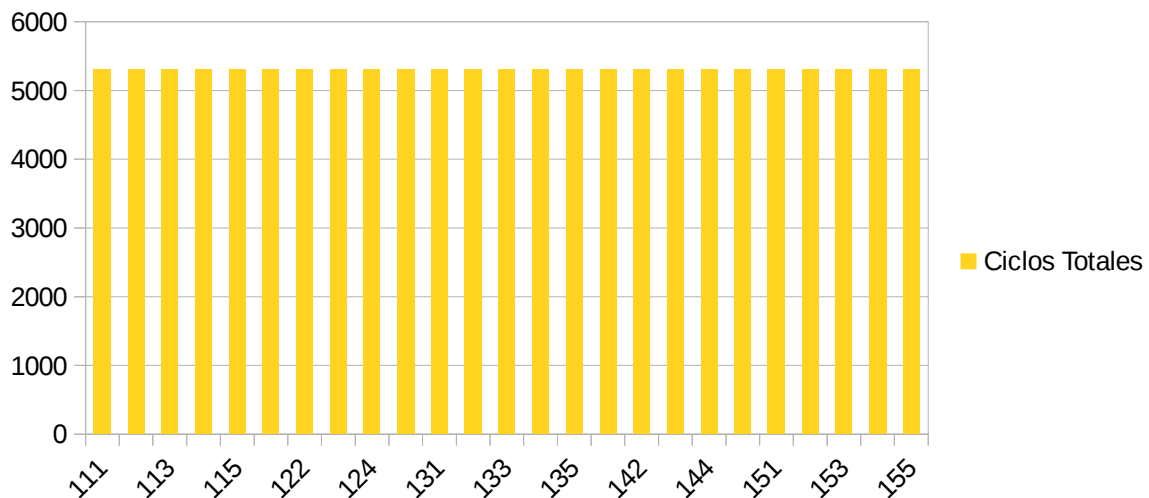


Echando un vistazo a las gráficas se observa que ambas configuraciones son distintas. Vemos una **superioridad** de la configuración **(2,4,2)**, ya que emplea **4594 ciclos**, mientras que la configuración **(1,5,3)** emplea **5300**.

Por lo tanto, la **segunda configuración** está haciendo un **mal aprovechamiento** del procesador ya que puede captar, decodificar, emitir y finalizar menos instrucciones por ciclo que la primera configuración.

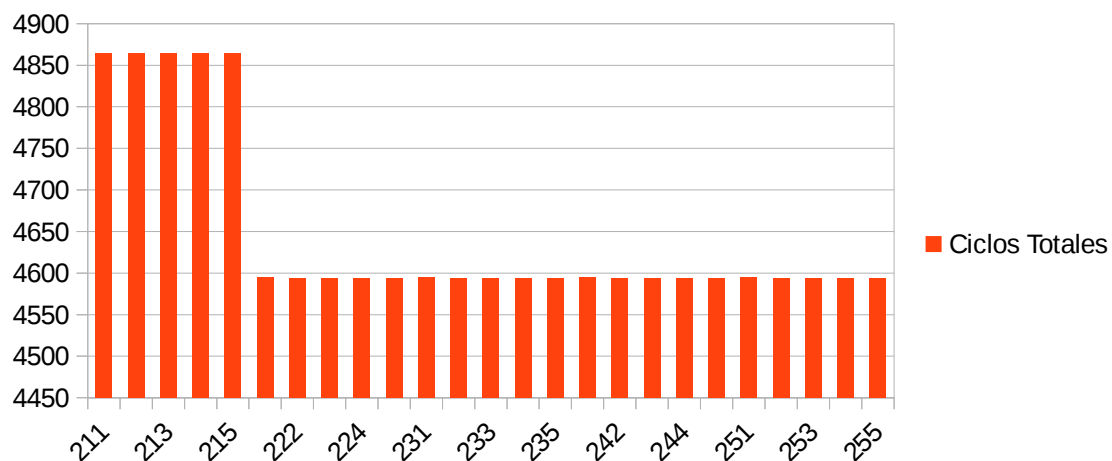
Por lo que si tuviéramos que decantarnos por alguna de las dos, la primera sería la mejor opción.

Finalmente, se procede a buscar un número de instrucciones captadas, decodificadas y finalizadas por ciclo que genere un número de ciclos de ejecución adecuado, **minimizando recursos**:



Para todas las configuraciones posibles con **captación fija a 1**, se observa que es **indiferente** cualquier cambio efectuado de las demás variables (decodificación y finalización), ya que para todas los ciclos totales se mantienen constantes.

Además los ciclos totales se mantienen muy **altos** en comparación con las próximas configuraciones. Por tanto, ninguna de estas configuraciones son optimas.

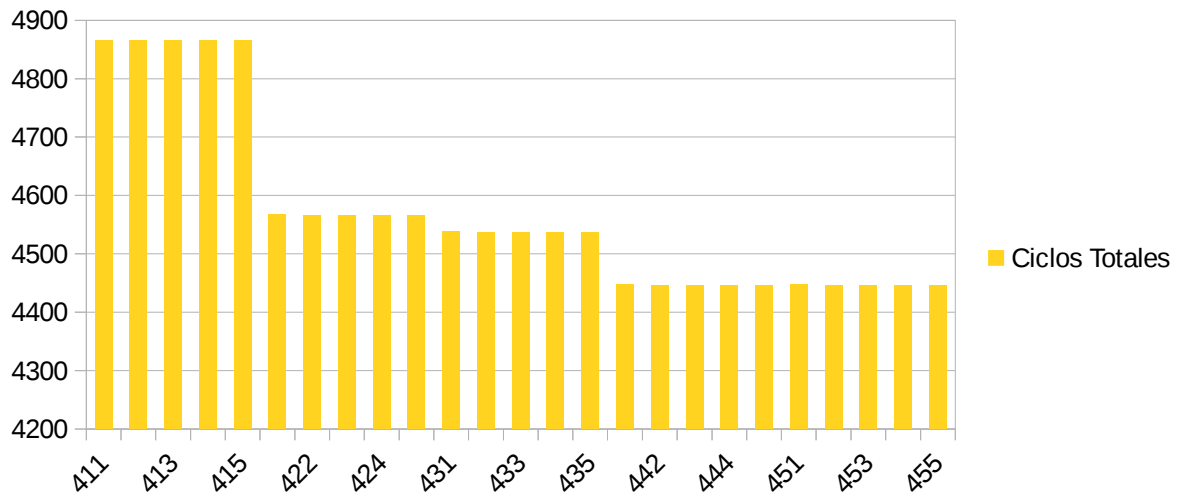


Para todas las configuraciones posibles con **captación fija a 2**, apreciamos una **mejoría** a partir de la configuración **(2,2,1)**, pero a partir de este punto, cualquier modificación del procesador mantiene los **ciclos totales constantes**.

Por tanto, en este caso la configuración **más optima es (2,2,1)**.

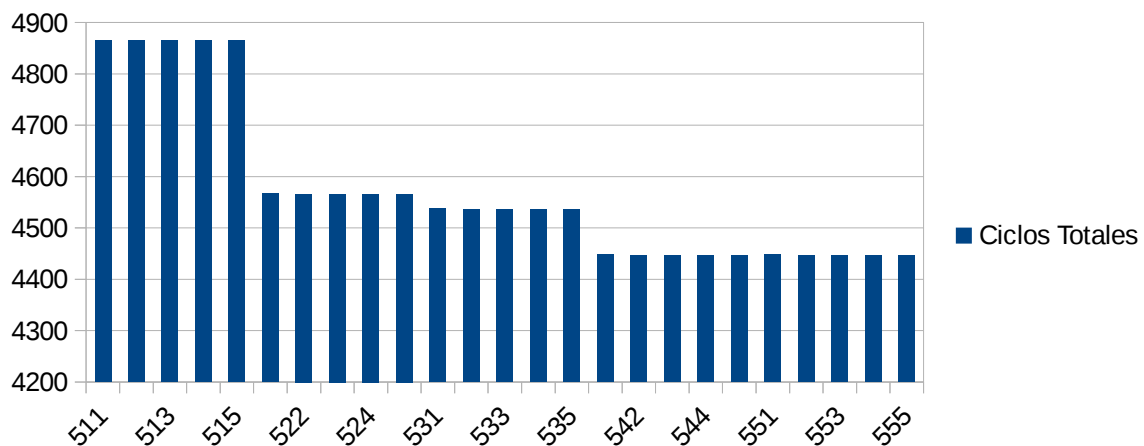
Ahora analizamos los resultados para todas las configuraciones posibles con **captación fija a 3**, y a partir de la configuración **(3,2,1)** ya no existen cambios significativos y los ciclos totales se mantienen en 4567.

Por tanto, la configuración más óptima en este caso es de **(3,2,1)**, aunque en **(3,3,2)** obtenemos **4564 ciclos totales** (3 ciclos menos), por lo que ambas son óptimas.



Para todas las configuraciones posibles con **captación fija a 4**, se aprecia mejorías escalonadas. A partir de la configuración **(4,4,2)** encontramos **4446 ciclos totales** que serían los mínimos obtenidos con todas las configuraciones del procesador.

Por tanto, **la configuración más óptima es (4,4,2).**



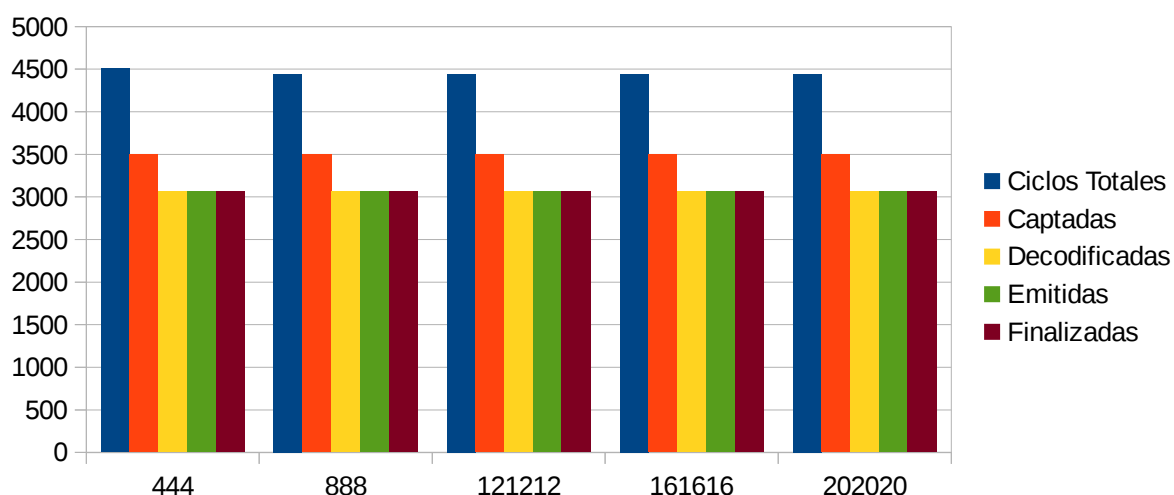
Para todas las configuraciones posibles con **captación fija a 5**, observamos que se mantiene los mismo resultados que la gráfica anterior. Por tanto, estamos usando más recursos y estamos obteniendo los mismos resultados.

Dichas configuraciones **no son las óptimas** por lo dicho anteriormente.

La configuración de 4 instrucciones captadas, 4 instrucciones emitidas y 2 finalizadas es la más adecuada ya que optimiza los recursos.

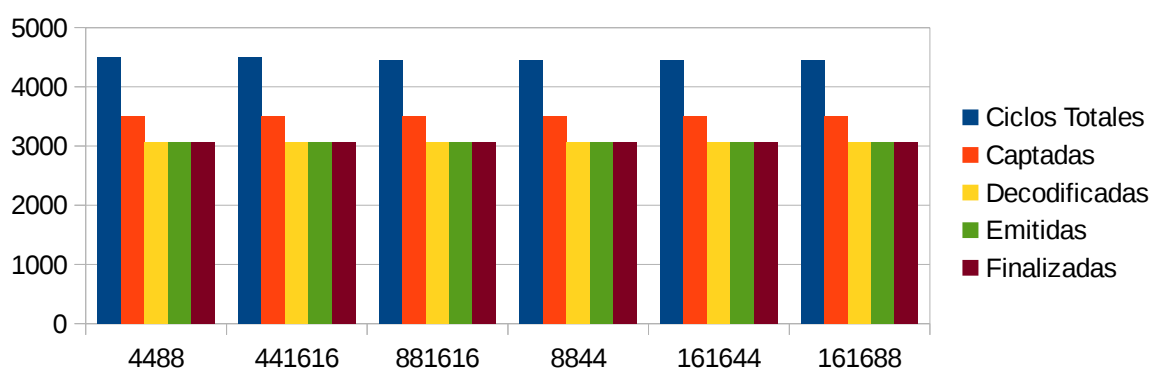
2. Cola de instrucciones (CI), ventana de instrucción (VI) y buffer de reorden (ROB)

Partiendo de la configuración (5,5,5), vamos a ver cómo varían los ciclos cuando modificamos las distintas estructuras:



Observando los datos obtenidos observamos que a partir de un **tamaño** de cola de instrucción, de ventana de instrucción y de buffer de reorden de **8**, no se obtienen mejora del procesador.

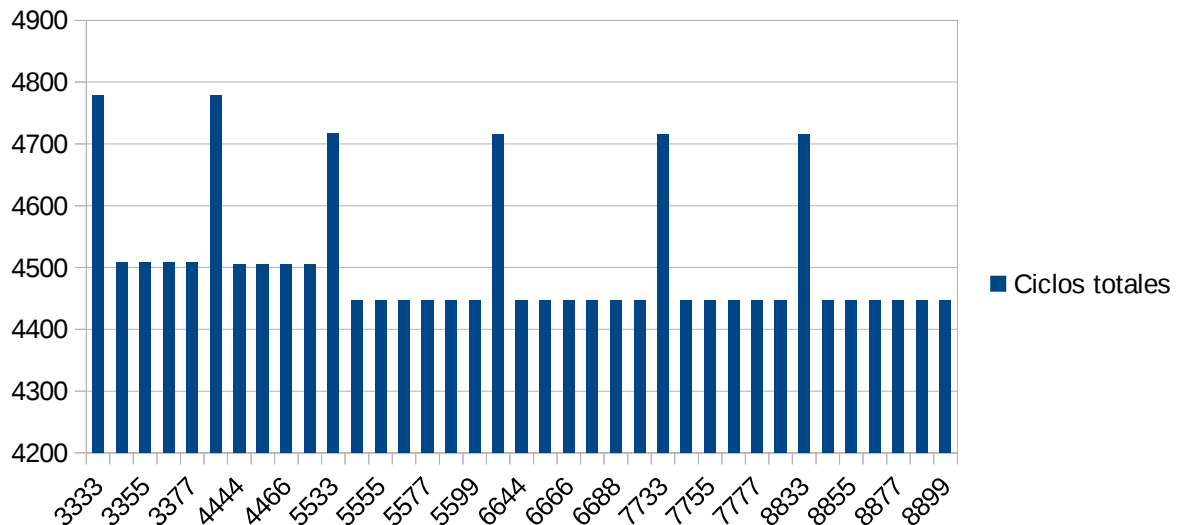
Por otro lado, procedemos a calcular los ciclos de ejecuciones para **otras configuraciones** de las características del procesador superescalar. En este caso se mantiene constantes la captación, decodificación y finalización (5,5,5) y también la cola de instrucciones (16).



Observando estas configuraciones, vemos que teniendo una **ventana de instrucción** para enteros de 8 y una de 4 para flotantes, y un **ROB** de 8 para enteros y 4 para flotantes obtenemos una configuración más óptima de entre todas las que aparecen en la gráfica, ya que obtenemos **4446 ciclos de ejecución**.

Si aumentamos más recursos lo único que se ha conseguido es **desperdiciarlos**, ya que no se obtiene mejora.

Finalmente, se procede a buscar un número de instrucciones captadas, decodificadas y finalizadas por ciclo que genere un número de ciclos de ejecución adecuado, **minimizando recursos**, y junto con un tamaño de la cola de instrucción, ventana de instrucciones y buffer de reorden:



En este análisis de las distintas configuraciones hemos mantenido la cola de instrucciones fija a **16**, y un total de instrucciones captadas, decodificadas y finalizadas iguales a la configuración más óptima encontrada en el apartado 1, **(4,4,2)**.

Por tanto, que las **dos configuraciones más óptimas son**:

Tamaño del ROB: 5 para enteros y 4 para flotantes.

Tamaño de la ventana de instrucciones: 5 para enteros y 4 para flotantes.

Ciclos totales: 4447

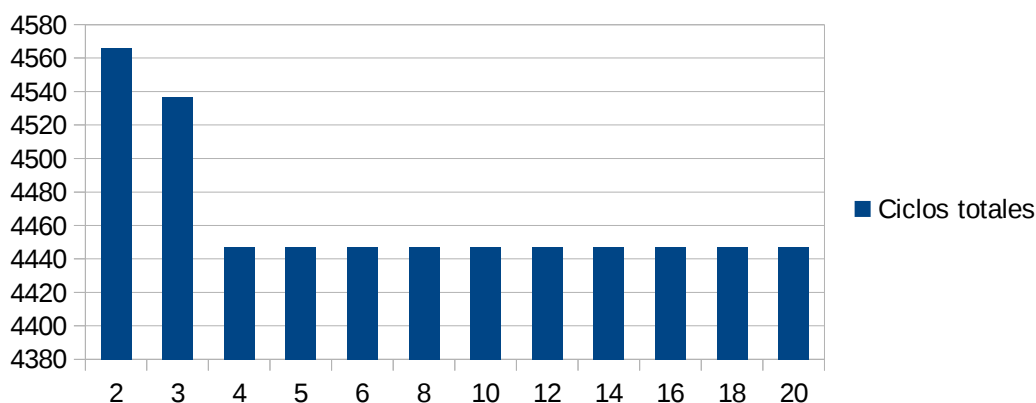
Tamaño del ROB: 6 para enteros y 4 para flotantes.

Tamaño de la ventana de instrucciones: 6 para enteros y 4 para flotantes.

Ciclos totales: 4446

Obstando por la primera configuración ya que aunque obtengamos un ciclo menos hace un uso menor de los recursos del procesador superescalar, y tarda el mínimo de ciclos en ejecutarse.

A continuación se procede a modificar el **tamaño de la cola de instrucciones** (anteriormente mantenida constante), y manteniendo las demás configuraciones constantes con los valores que hemos calculado como los más **óptimos**:



Observamos en la gráfica que el ***tamaño más óptimo de la cola de instrucciones es 4***, ya que a obtenemos ***4447 ciclos*** y a partir de ahí lo único que conseguimos es mantener ese número de ciclos.

Finalmente, se muestra la configuración más óptima:

Instrucciones captadas por ciclo: 4

Instrucciones decodificadas por ciclo: 4

Instrucciones finalizadas por ciclo: 2

Tamaño del ROB: 5 para enteros y 4 para flotantes.

Tamaño de la ventana de instrucciones: 5 para enteros y 4 para flotantes.

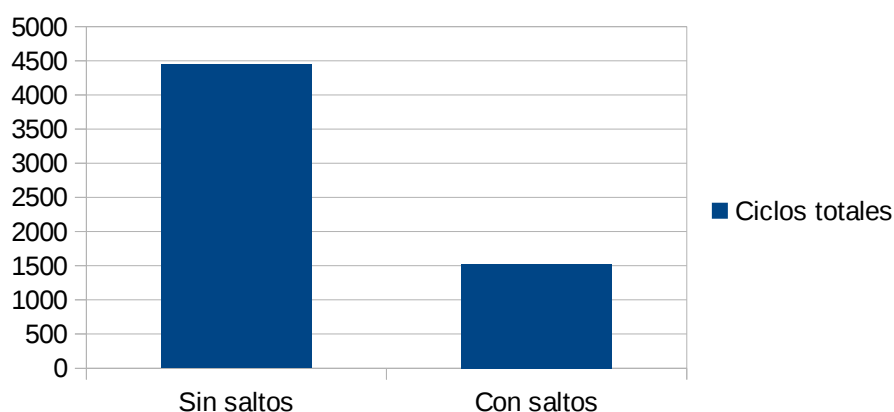
Tamaño de la cola de instrucciones: 4

Ciclos totales: 4447

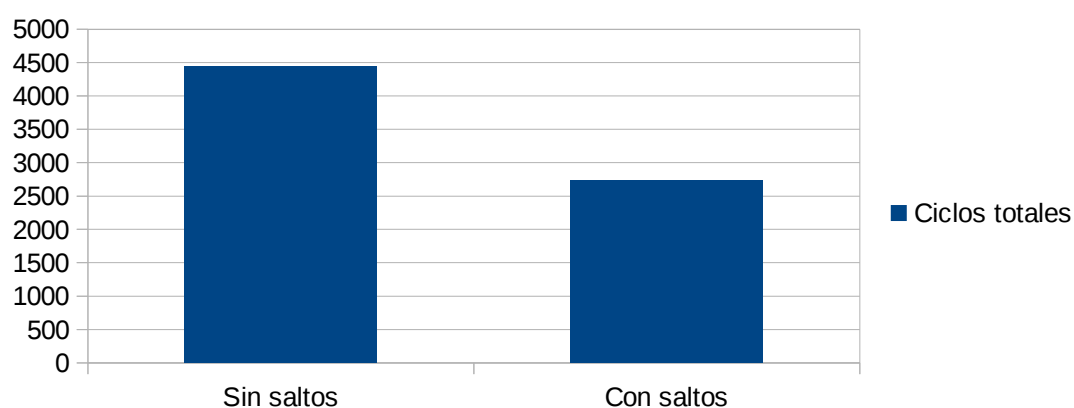
3. Predicción de saltos y coste

A continuación vamos a analizar que resultados se obtiene si modificamos la opción de predicción de saltos:

Para el **apartado 1**:



Para el **apartado 2**:



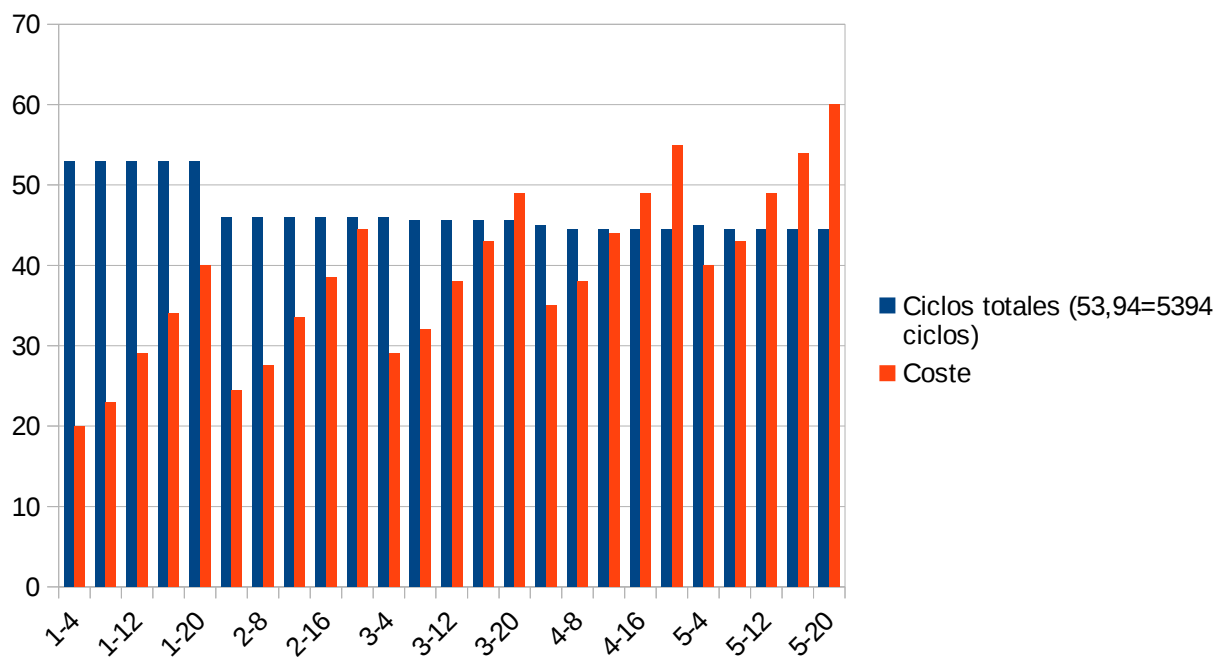
Observamos un decremento en ambas configuraciones activando el predictor de saltos.

En el primer apartado vemos como los ciclos bajan de **4446 a 1520**.

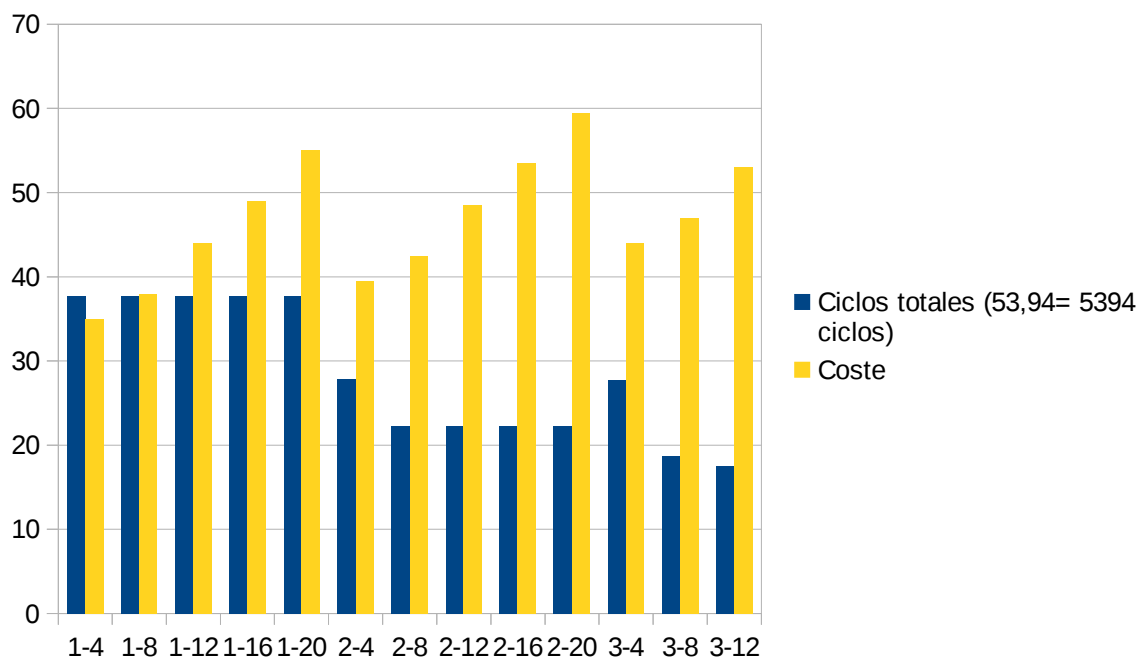
En el segundo apartado bajan de **4447 a 2730**.

Finalmente, se busca una configuración que permita un **mayor rendimiento** ajustándose a los **costes** asignados:

En primer lugar, hacemos un balance de los ciclos empleados para **cada una de las líneas** de captación, decodificación y finalización, y también para los distintos **tamaños de las estructuras**. La predicción de saltos no se tiene en cuenta:



En segundo lugar, se hace el mismo balance que antes pero ahora con la **predicción de saltos activos**:



En la primera gráfica podemos observar que los **ciclos de ejecución son superiores** a los de la segunda. Pero también hay que ver que los **precios de la segunda son más elevados** (incluso a partir de la configuración de 4 instrucciones por ciclo se escapa del presupuesto).

Ahora toca elegir entre si usar predicción de saltos o no. Primero vamos a ver cual es la mejor opción si **no** se elige predicción de saltos y después la mejor opción **si** se elige que la arquitectura tenga predicción de saltos:

En la configuración ***sin predicción de saltos***, la mejor opción es la de 4 instrucciones por ciclo (captadas, decodificadas y finalizadas), y un tamaño del ROB, CI y VI de 8. Obteniendo un gasto por valor de **38 euros** y un total de ciclos de **4446**.

Por otro lado, si se decide usar ***predicción de saltos***, la mejor opción es la de 3 instrucciones por ciclo (captadas, decodificadas y finalizadas), y un tamaño del ROB, CI y VI de 8. Obteniendo un gasto por valor de **47 euros** y un total de ciclos de **1871**.

Por lo tanto, ***la mejor opción es elegir la predicción de saltos***, ya que se realiza un mejor aprovechamiento de la arquitectura y del presupuesto. También hay que decir, que el tiempo que tarda en ejecutarse es ***notablemente inferior*** en dicha configuración.

Resumen de la configuración:

Coste base: 20 euros.

Predicción de saltos: 15 euros.

3 instrucciones por ciclo: 9 euros.

Tamaño de las estructuras de 8 líneas: 3 euros.

Precio total: 47 euros.

Ciclos en ejecutarse: 1871.