Práctica 2:

PROCESADORES SUPERESCALARES con SuperDLX

5º curso Ing. Informática
Arquitectura de Computadores I
José Antonio Guerrero Avilés
75485683M
cany@correo.ugr.es

(1) Evalúe y justifique los cambios en los tiempos de ejecución cuando se modifican las características del procesador superescalar:
- Número de instrucciones que captan, decodifican o finalizan ('commit') por ciclo de reloj.

- Voy a variar el número de instrucciones que se captan (IF) y voy a dejar igual el resto de opciones que vienen por defecto (decode = 2, commit = 4).

Captan (Fetch)	pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
2	68	67	21	664	1284	174
4	68	67	21	664	1284	174
8	68	67	21	664	1284	174

Como se puede observar, variar el número de instrucciones que se captan, no varía el número de ciclos que tarda en ejecutarse un programa.

- Voy a variar el número de instrucciones que se decodifican (ID) y voy a dejar igual el resto de opciones que vienen por defecto (fetch = 4, commit = 4).

Decodifican (Decode)	pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
2	68	67	21	664	1284	174
4	43	43	21	494	755	174
8	43	43	21	494	745	174

Como se puede observar, variar el número de instrucciones que se decodifican, varía notablemente el número de ciclos que tarda en ejecutarse un programa.

- Voy a variar el número de instrucciones que finalizan (commit) y voy a dejar igual el resto de opciones que vienen por defecto (fetch = 4, decode = 2).

Finalizan (Commit)	pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
2	68	68	22	665	1284	181
4	68	67	21	664	1260	174
8	68	67	21	664	1260	173

Como se puede observar, variar el número de instrucciones que finalizan, no varía a penas el número de ciclos que tarda en ejecutarse un programa.

- Tamaño de los buffers de reorden, ventanas de instrucciones, y cola de instrucciones.

- Vuelvo a la configuración inicial y cambio el tamaño de los buffer de reordenamiento.

Tamaño Buffer	pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
Enteros = 10 Float = 10	68	67	23	664	1284	180
Enteros = 20 Float = 20	68	67	21	664	1284	174
Enteros = 30 Float = 30	68	67	21	664	1284	174

Según los datos obtenidos, apenas afecta a nuestros programas cambiar el tamaño de los buffer de reordenamiento.
Lo que no he probado aún es disminuir el tamaño del buffer de reordenamiento, así que voy a hacerlo;

Tamaño Buffer	pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
Enteros = 5 Float = 5	92	82	31	904	1586	243

En este caso, si que varía significativamente el número de ciclos que tardan en ejecutarse los programas, pero, en este caso, a peor. Es decir, hay un momento (a partir del tamaño 10 para ambos buffer) que a penas varía el número de ciclos, sin embargo, si cogemos un tamaño pequeño para ambos, si que se ve afectado el número de ciclos y el programa tarda más.

- Vuelvo a la configuración inicial y cambio el tamaño de la ventana de instrucciones.

Tamaño Ventana	pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
Enteros = 10 Float = 10	68	67	23	664	1284	174
Enteros = 20 Float = 20	68	67	21	664	1284	174
Enteros = 30 Float = 30	68	67	21	664	1284	174

Según los datos obtenidos, no afecta a nuestros programas cambiar el tamaño de las ventanas de instrucciones
Lo que no he probado aún es disminuir el tamaño, así que voy a hacerlo;

Tamaño Ventana	pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
Enteros = 5 Float = 5	68	67	21	664	1284	174

En este caso, tampoco varía el número de ciclos que tardan los programas en ejecutarse.

- Predicción/No Predicción de Saltos y número de bits de predicción.

Predicción de saltos	pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
Activada	68	67	23	638	1257	174
Desactivada	68	67	23	664	1284	174

En este caso, no afecta que activemos o desactivemos la predicción de saltos.

Con la predicción de saltos activada, voy a ver lo que ocurre si varío los bits de predicción.

Bits de predicción	pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
1	68	67	23	638	1257	174
2	68	67	23	638	1257	174
3	68	67	23	638	1257	174
4	68	67	23	641	1260	174

Como se puede ver, tampoco afecta que cambiemos el número de bits de predicción.

Esto es debido al poco número de saltos que tenemos en nuestros programas.

(2) Determine, a partir de los resultados obtenidos en el apartado anterior, la configuración de menor costo que permita obtener los mejores tiempos de ejecución en cada programa. Para ello, debe demostrar que el aumento de las capacidades del procesador, no repercute en una reducción de ciclos o que la relación entre el aumento del coste y la reducción de tiempos no se justifica.

Suponiendo que la siguiente tabla muestra la mejor configuración general, vamos a comprobar el ahorro en ciclos que supone aplicar esta configuración:

Fetch	8
Decode	8
Commit	8
Tam. Buffer	Enteros = 20 Float = 20
Tam. Ventana	Enteros = 20 Float = 20
Predicción de Saltos	Activada

Aplicándola obtenemos el siguiente número de ciclos para cada programa:

Pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
43	43	21	494	745	174

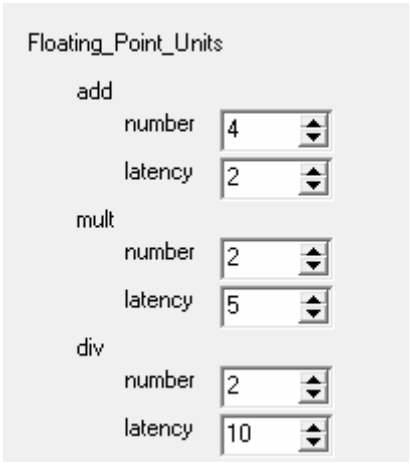
Si bien es cierto que obtenemos los mejores resultados respecto a los que hemos visto en el apartado anterior, es necesario hacer una compensación entre el coste y el número de ciclos, así que lo que voy a hacer es ver la configuración idónea de cada programa por separado, ya que usar la configuración anterior para todos, aún dándonos una mejora en el número de ciclos, tiene un coste muy alto.

Con los datos que hemos obtenido en el apartado anterior, podemos configurar la siguiente tabla con la configuración más idónea para cada uno de los programas por separado que hemos ido probando:

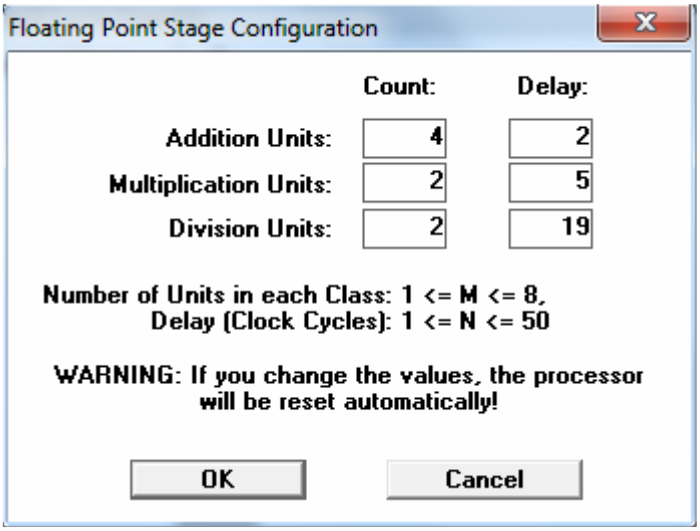
Configuración	Pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
Fetch	4	4	4	4	4	4
Decode	4	4	4	4	8	4
Commit	4	4	4	4	4	8
Tam. Buffer	Enteros = 10 Float = 10	Enteros = 10 Float = 10	Enteros = 20 Float = 20	Enteros = 10 Float = 10	Enteros = 10 Float = 10	Enteros = 20 Float = 20
Tam. Ventana	Enteros = 5 Float = 5	Enteros = 5 Float = 5	Enteros = 5 Float = 5	Enteros = 5 Float = 5	Enteros = 5 Float = 5	Enteros = 5 Float = 5
Predicción de Saltos	Desactivada	Desactivada	Desactivada	Activada	Activada	Activada
Bits de Predicción	1	1	1	1	1	1
Ciclos	43	43	21	494	745	174

(3) Compare los resultados obtenidos para el procesador superescalar y los que se obtuvieron para el procesador escalar con el simulador winDLX. Para esta comparación utilice el mismo número de unidades funcionales con los mismos retardos en los dos procesadores.

La configuración de WinSuperDLX es la siguiente:



Así que tenemos que configurar WinDLX de la siguiente manera para poder comparar correctamente los resultados:

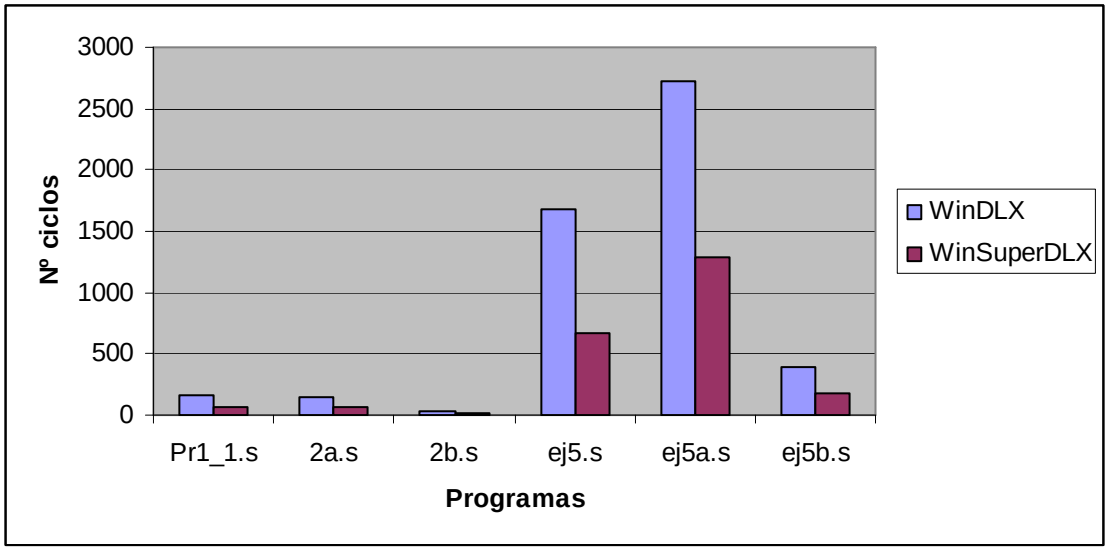


Con estas configuraciones obtenemos la siguiente tabla comparativa:

Ciclos	Pr1_1.s	2a.s	2b.s	ej5.s	ej5a.s	ej5b.s
WinDLX	168	139	38	1676	2726	395
WinSuperDLX	68	67	21	664	1284	174

Como podemos observar el procesador superescalar es en todos los casos mejor que el procesador escalar.

Se puede ver mucho mejor en la siguiente representación gráfica:



Esto es así, debido a que el procesador superescalar (SuperWinDLX) puede decodificar, captar y finalizar más de una instrucción por ciclo, mientras que en el procesador escalar (WinDLX) solo una instrucción por ciclo.

(4) Para los programas inic1.s, inic2.s, inic3.s, inic4.s, e inic5.s obtenga los valores de las ganancias de velocidad del procesador superescalar con respecto al procesador sin segmentar de referencia ssrDLX y compárelas con las obtenidas para el procesador segmentado con WinDLX. ¿Que conclusiones se pueden extraer? (NOTA: Utilice una configuración de unidades funcionales en WinDLX similar a la del procesador superescalar. Considere tamaños suficientemente grandes para las colas, ventanas de instrucciones, y ROBs, de forma que no supongan una limitación en la velocidad de ejecución de instrucciones).

Para poder realizar este estudio, voy a ver primero cuanto tarda en ejecutarse cada programa en cada uno de los procesadores:

- SuperWinDLX

Como dice la nota, con un tamaño para las colas, ventanas de instrucciones y ROB suficientemente grande para que no suponga una limitación en la velocidad de ejecución.

Programa	Nº Ciclos
inic1_x.s	69
Inic2_x.s	69
Inic3_x.s	69
Inic4_x.s	69
Inic5_x.s	69

- WinDLX

Con la misma configuración de las unidades funcionales que en SuperWinDLX. (La misma que en el ejercicio 3).

Programa	Nº Ciclos
inic1_x.s	230
Inic2_x.s	215
Inic3_x.s	158
Inic4_x.s	140
Inic5_x.s	146

- ssrDLX

El tiempo de una instrucción en el procesador sin segmentar se puede calcular partiendo de las etapas por las que pasa cada instrucción y sabiendo el número de ciclos de cada etapa.
Analizando los tipos de instrucción que tenemos en el programa y por las etapas que pasa, hacemos una estimación de los ciclos que consume cada operación en el procesador sin segmentar.

A continuación se muestra la tabla, que indica todas las operaciones y las etapas por la que pasa cada instrucción:

Instrucción	IF	ID	EX	MEM	WB	TOTAL
ld	1	0.8	1	1	0.8	**4.6**
sd	1	0.8	1	1	-	**3.8**
add	1	0.8	1	-	0.8	**3.6**
multd	1	0.8	5	-	0.8	**7.6**
sub	1	0.8	1	-	0.8	**3.6**
bnez	1	0.8		-	-	**1.8**
trap	1	0.8	-	-	-	**1.8**
J	1	0.8	-	-	-	**1.8**

Ahora vamos a ver qué instrucciones hay en cada programa y cuanto dura, aproximadamente, cada uno de los programas haciendo una estimación del número de ciclos mediante el número de instrucciones que tiene cada programa, su tipo y lo que tarda cada una de ellas en la tabla anterior.

- **Inic1_x.s**

Inicio:
LD = 1*4,6 = 4,6
ADD = 2*3,6 = 7,2

Bucle:
LD = 4*4,6 = 18,4
SD = 4*3,8 = 15,2
Multd = 4*7,6 = 30,4
Sub = 2*3,6 = 7,2
Bnez = 1*1,8 = 1,8

Fin:
Trap = 1*1,8 = 1,8

Total ciclos = 11,8+438+1,8=451,6

- **Inic2_x.s**

Inicio:
LD = 1*4,6 = 4,6
ADD = 2*3,6 = 7,2

Bucle:
LD = 8*4,6 = 36,8
SD = 8*3,8 = 30,4
Multd = 8*7,6 = 60,8
Sub = 2*3,6 = 7,2
Bnez = 1*1,8 = 1,8

Fin:
Trap = 1*1,8=1,8

Total ciclos = 11,8+389,4+1,8 = 403

- **Inic3_x.s**

Inicio:
LD = 1*4,6 = 4,6
ADD = 2*3,6 = 7,2

Bucle:
LD = 4*4,6 = 18,4
SD = 4*3,8 = 15,2
Multd = 4*7,6 =30,4
Sub = 2*3,6 =7,2
Bnez = 1*1,8 = 1,8

Fin:
Trap = 1*1,8=1,8

Total ciclos = 11,8+438+1,8 = 451,6

- **Inic4_x.s**

Inicio:
LD = 1*4,6 = 4,6
ADD = 2*3,6 = 7,2

Bucle:
LD = 4*4,6 = 18,4
SD = 4*3,8 = 15,2
Multd = 4*7,6 = 30,4
Sub = 2*3,6 =7,2
Bnez = 1*1,8 =1,8

Fin:
Trap = 1*1,8 =1,8

Total ciclos = 11,8+438+1,8 = 451,6

- **Inic5_x.s**

Inicio:
LD = 5*4,6 = 23
ADD = 2*3,6 = 7,2
jj = 1*1,8 = 1,8

Bucle 1:
LD = 4*4,6 = 18,4
SD = 4*3,8 = 15,2
sub = 1*3,6=3,6

Bucle 2:
Multd = 4*7,6 = 30,4
Sub = 1*3,6 = 3,6
Bnez = 1*1,8 = 1,8

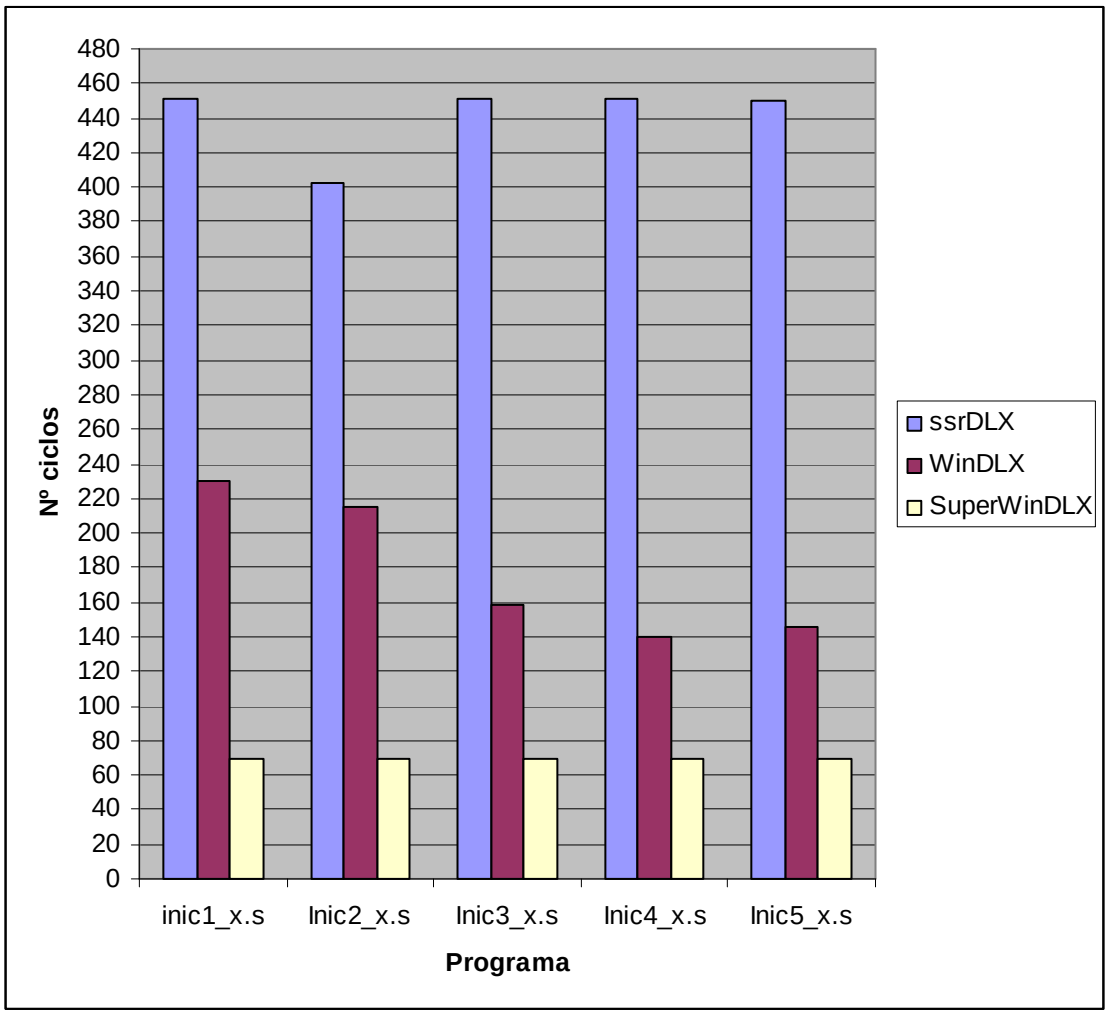
Fin:
Sd = 4*3,8 = 15,2
Trap = 1*1,8 = 1,8

Total Ciclos = 32+186+214,8+17 = 449,8

Esta es la tabla del número de ciclos que tarda cada programa en ejecutarse en cada uno de los procesadores:

Programa	ssrDLX	WinDLX	SuperWinDLX
inic1_x.s	451,6	230	69
Inic2_x.s	403	215	69
Inic3_x.s	451,6	158	69
Inic4_x.s	451,6	140	69
Inic5_x.s	449,8	146	69

En el siguiente gráfico se puede apreciar cómo mejora el tiempo de ejecución de cada programa respecto a cada procesador:



Ahora, con los datos obtenidos, vamos a ver, para cada programa, la ganancia de un procesador respecto a otro.

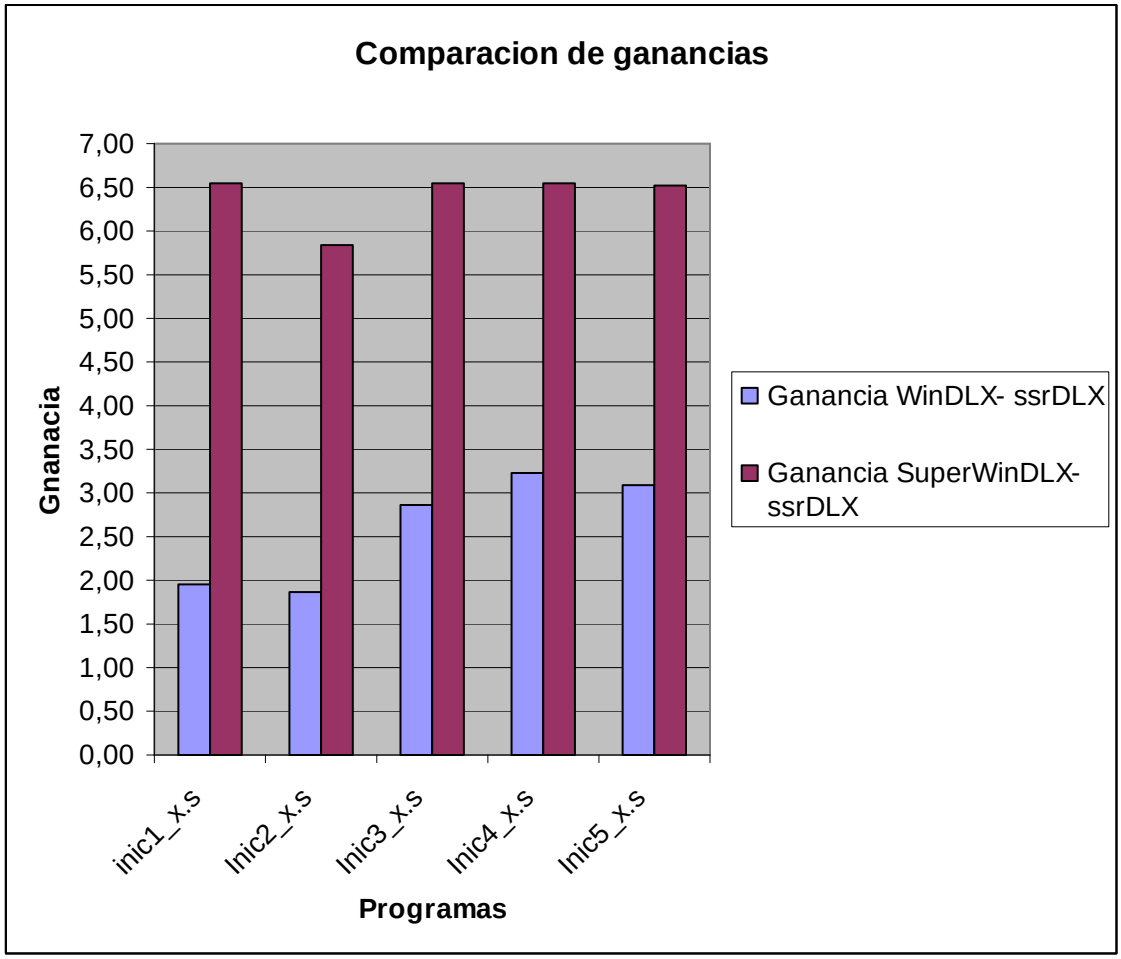
- **WinDLX- ssrDLX**

Programa	Nº Ciclos ssrDLX	Nº Ciclos winDLX	Ganancia WinDLX- ssrDLX
inic1_x.s	451,6	230	1.963
Inic2_x.s	403	215	1.874
Inic3_x.s	451,6	158	2.858
Inic4_x.s	451,6	140	3.225
Inic5_x.s	449,8	146	3.080

- **SuperWinDLX- ssrDLX**

Programa	Nº Ciclos ssrDLX	Nº Ciclos SuperwinDLX	Ganancia WinDLX- ssrDLX
inic1_x.s	451,6	69	6.544
Inic2_x.s	403	69	5.840
Inic3_x.s	451,6	69	6.544
Inic4_x.s	451,6	69	6.544
Inic5_x.s	449,8	69	6.518

En la siguiente gráfica podemos apreciar la mejora del procesador SuperWinDLX sobre el procesador WinDLX, comparándolos con el procesador ssrDLX:



Esto se debe al menor número de ciclos que tarda cada programa en ejecutarse en el procesador superescalar (SuperWinDLX) ya que puede captar, decodificar, ejecutar y finalizar varias instrucciones a la vez.