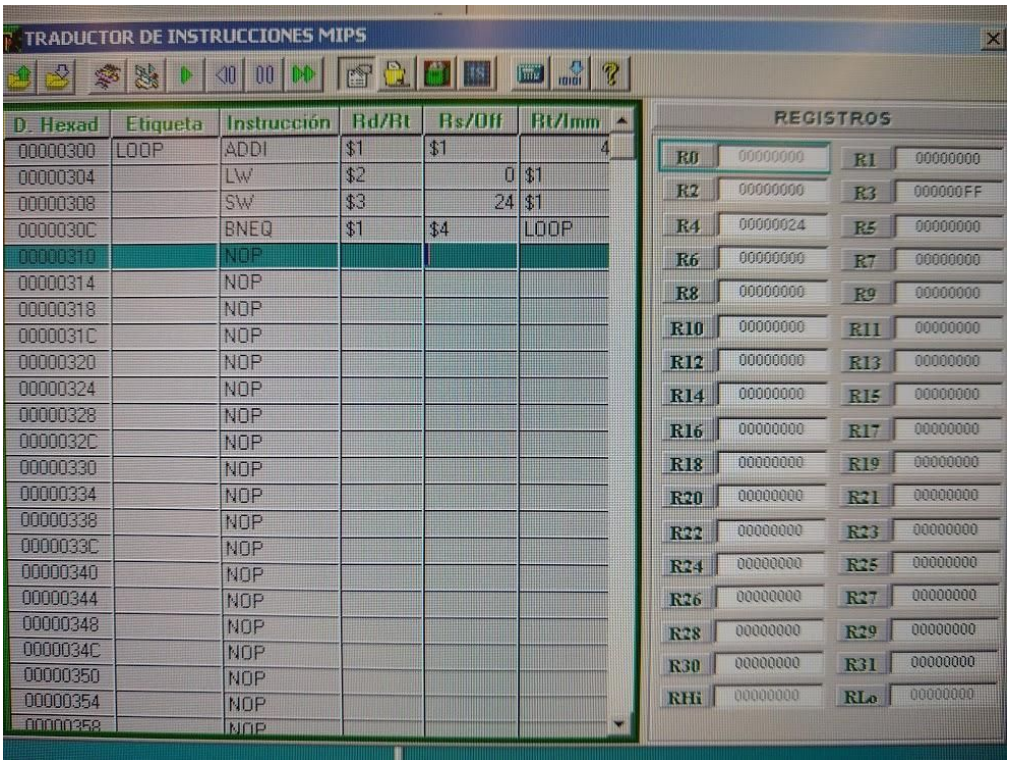
SICAM

Práctica 4 de Arquitectura de Computadores

José Márquez Doblas GG2

El programa realizado en la práctica es un programa sencillo que consiste en un bucle que hace diversos accesos a memoria mediante las instrucciones LW y SW (Load Word y Save Word respectivamente).

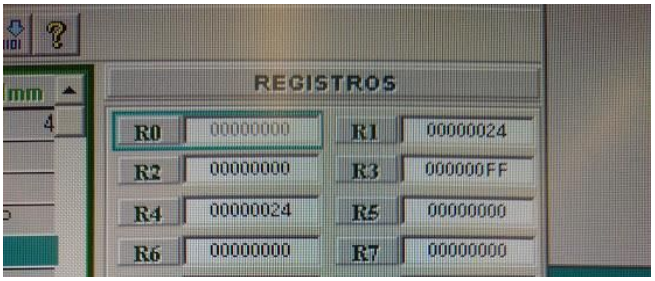


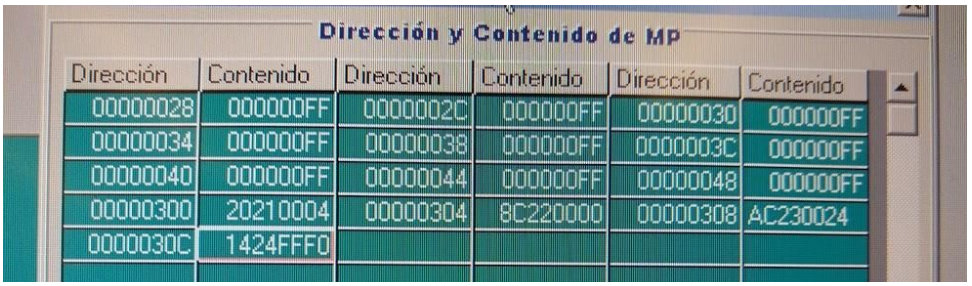
El funcionamiento del programa es el siguiente:

ADDI va sumando 4 al contenido del registro $1 y guarda el resultado en ese mismo registro, LW carga en el registro $2 la palabra cuya dirección es el contenido del registro $1, de igual forma SW guarda el registro $3, inicializado a FF por nosotros, en la palabra direccionada por el contenido del registro $1 más 24. Finalmente BNEQ bifurcará a ADDI si el registro $1 es diferente al registro $4, inicializado a 24 por nosotros, terminando el programa cuando estos sean iguales, es decir, cuando el contenido del registro $1 sea 24.

El contenido del registro $2 no debería variar de 0 puesto que la primera dirección que modifica SW es la 28 y la última que carga LW es 24. Por lo tanto si nos quedase FF en el registro $2 sabríamos que algo ha fallado en el bucle.

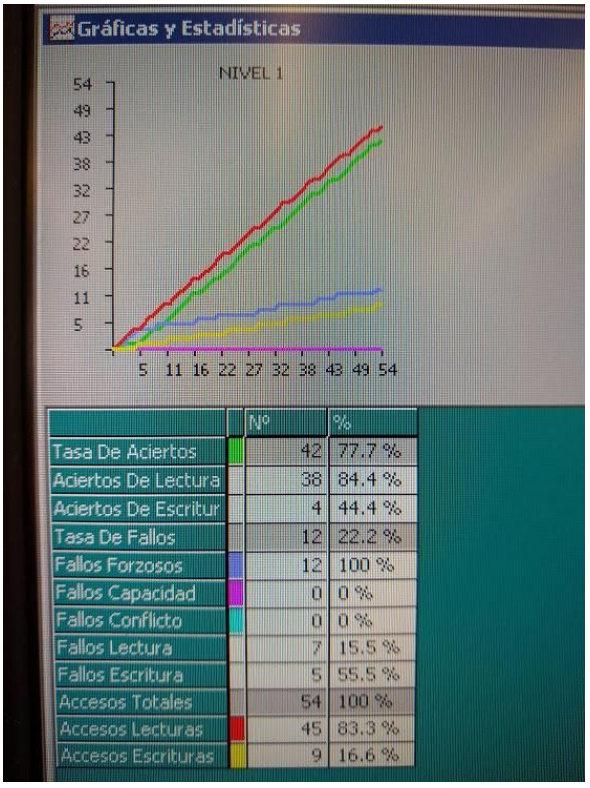
Tras realizar la simulación obtenemos los siguientes estados de los registros y la memoria principal:

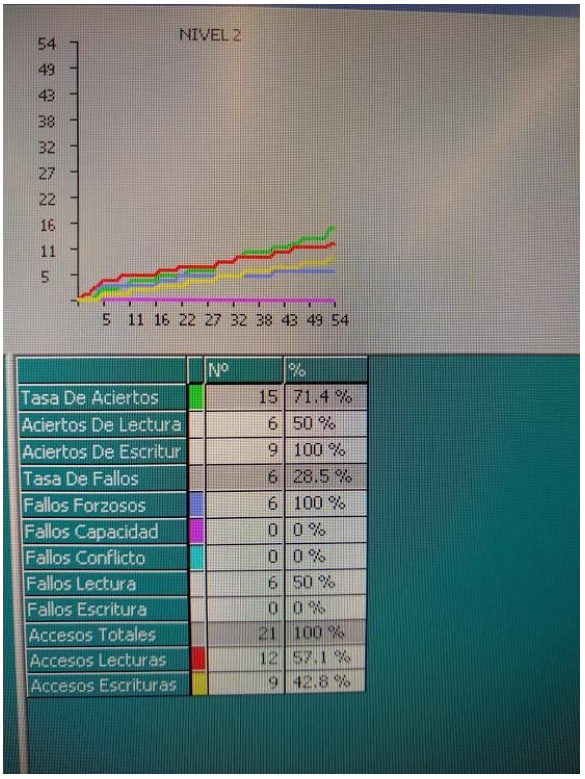




Tal y como he descrito antes, el contenido del registro $2 quedará a 0 al finalizar el proceso.

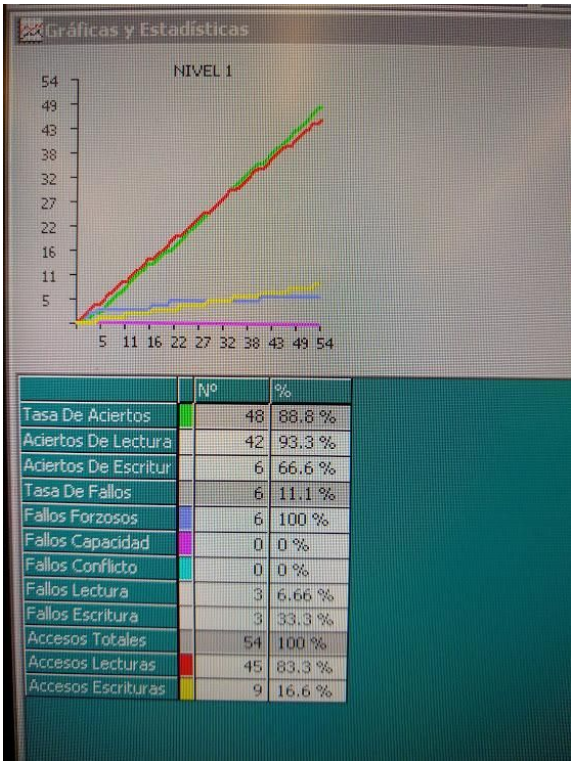
Compilamos y ejecutamos el programa de simulación con su configuración por defecto, obteniendo los siguientes resultados:





Podemos concluir de los resultados obtenidos que tanto la tasa de aciertos de L1 como la de L2 no son remarcables con un 77,7% y un 71,4% respectivamente. Si podemos analizar la tasa de aciertos en lectura y escritura por separado. En L2 vemos que la tasa de aciertos en lectura es del 50% mientras que en L1 es de 84,4%, esto nos dice que L1 es más eficiente en lectura que L2. Sin embargo, L2 tiene una tasa de aciertos en escritura del 100%, es decir, que todas las veces que ha querido escribir lo ha hecho, por el otro lado L1, con una tasa de aciertos en escritura del 44,4%, denota su baja eficiencia en escritura frente a L2, mucho más que su comparación en lectura.

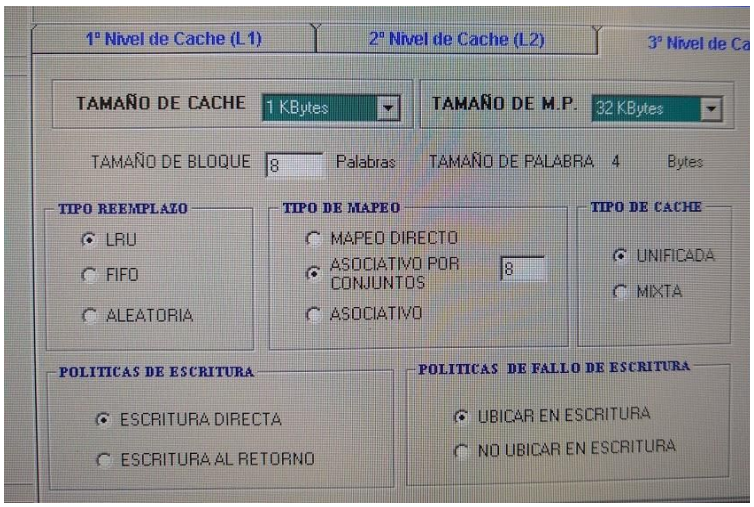
Si cambiamos la configuración de memoria caché a un solo nivel y cambiamos su tamaño de bloque a 4 palabras, obtenemos los siguientes resultados:

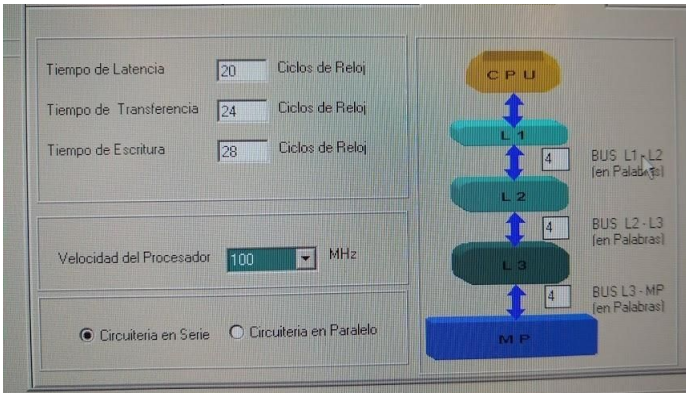


	Nº	%
Tasa De Aciertos	48	88.8 %
Aciertos De Lectura	42	93.3 %
Aciertos De Escritur	6	66.6 %
Tasa De Fallos	6	11.1 %
Fallos Forzosos	6	100 %
Fallos Capacidad	0	0 %
Fallos Conflicto	0	0 %
Fallos Lectura	3	6.66 %
Fallos Escritura	3	33.3 %
Accesos Totales	54	100 %
Accesos Lecturas	45	83.3 %
Accesos Escrituras	9	16.6 %

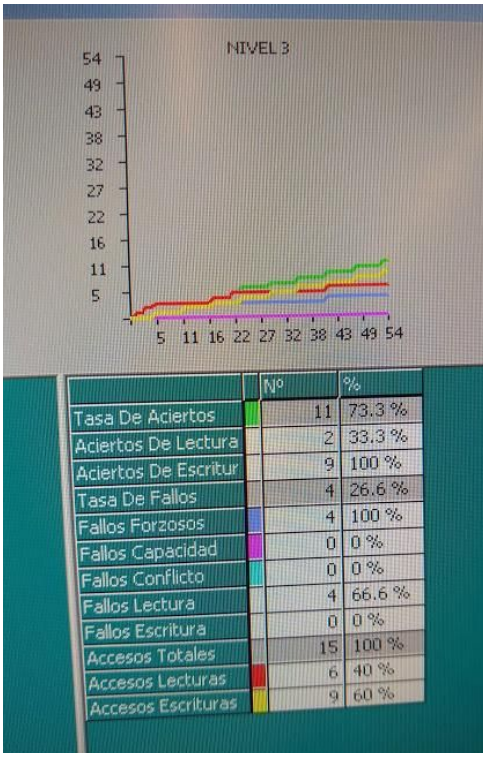
Dado que nuestro problema modifica direcciones de memoria próximas las unas a las otras era fácil concluir que iban a salir estos resultados. Al aumentar el tamaño de bloque, cada vez que copia un bloque se trae más palabras de memoria haciendo que la tasa de aciertos sea mayor, aunque sigue sin ser remarcable, aumentando además tanto la tasa de aciertos en lectura como en escritura, a pesar de que esta última sigue siendo bastante baja.

Por último cambiaremos nuestra configuración de memoria a un memoria caché de 3 niveles dejando la configuración de la primera simulación a L1 y L2 y teniendo L3 con la siguiente:





Y obtenemos los mismos resultados en L1 y L2 que en el primer ejemplo y los siguientes resultados en L3:



Podemos concluir de estos resultados que L3 es todavía menos eficiente que L2 en lectura pero mantiene su tasa de aciertos en escritura del 100%, sin embargo, tiene una tasa de errores menor ya que el número de accesos totales es menor también.