

Arquitectura de Computadores (Grado II)

Práctica de *pipeline* de instrucciones.

Nombre del fichero: *ILps_mat1_mat2_Sx_Fy.pdf* (números de matrícula, sesión y grupo)

Guión del trabajo a realizar por parte del alumno

1. Analizar las dependencias existentes en el programa original.
2. Configurar el simulador **sin adelantamientos** y con ciclos de parada como estrategia de salto y ejecutar el programa original.
 - Justificar los resultados obtenidos, en especial el número de ciclos de parada. Calcular el porcentaje de ciclos de parada debidos a los distintos tipos de dependencias y el CPI.
3. Comprobar el efecto que produce el **adelantamiento** en los resultados de la ejecución del programa. Para ello, configure el simulador con adelantamiento, manteniendo como estrategia de salto los ciclos de parada.
 - Justificar los resultados obtenidos, en especial el número de ciclos de parada y el porcentaje de ciclos de parada debidos a los distintos tipos de dependencias.
 - Calcular el CPI, así como la Ganancia (*speedup*) obtenida con respecto a la configuración anterior
4. **Reordenar** el código para mejorar las prestaciones obtenidas con la configuración del punto 3, y calcular el CPI, así como la ganancia obtenida.
5. **Partiendo del código obtenido en el punto 4** analizar algunas de las estrategias de salto de que dispone el simulador, manteniendo adelantamiento.
 - 5.1. **Con 1 delay slot.**
 - Comprobar si hay que modificar el programa para que éste funcione correctamente, y si es así realizar dicha modificación.
 - Ejecutar el programa resultante y justificar resultados obtenidos: número de instrucciones ejecutadas y ciclos de parada debidos a los distintos tipos de dependencias, y calcular el CPI
 - Reordenar el código para mejorar prestaciones, ejecutar el nuevo programa (con la misma configuración), comparar resultados y calcular la ganancia obtenida.
 - 5.2. **Con Predicción de No salto.**
 - Comprobar si hay que modificar el programa para que éste funcione correctamente, y si es así realizar dicha modificación. Ejecutar el programa resultante, justificar resultados obtenidos y calcular el CPI.
 - 5.3. **Con Predicción dinámica de 1 y 2 bits.**
 - Comprobar si hay que modificar el programa para que éste funcione correctamente, y si es así realizar dicha modificación.
 - Ejecutar el programa resultante, justificar resultados obtenidos y calcular el CPI
 - 5.4. **Comparación de los resultados** obtenidos con las estrategias de salto utilizadas y comentarios personales.