

Módulo 4: Paralelismo Interno. Software para Ejecución de Instrucciones en Computador Segmentado.

PRÁCTICA 4. Programación para DLX.

TEORÍA:

TEMA 4: PARALELISMO INTERNO EN LOS SISTEMAS COMPUTADORES.

El objetivo de esta práctica es el estudio experimental de un procesador segmentado para ilustrar la mejora de prestaciones que se puede conseguir con un cauce segmentado, analizando las dificultades y las limitaciones existentes para alcanzar dicha mejora. Para realizar el trabajo experimental correspondiente a esta práctica se utilizará el simulador WinDLX.

El simulador WinDLX ha sido desarrollado en la Universidad de Viena (ver Referencia Bibliográfica), y permite simular el procesador segmentado DLX en el sistema Windows de Microsoft. Posee un interfaz bastante amigable y permite analizar los problemas que plantea la ejecución, en el cauce segmentado del procesador, de un programa escrito en el lenguaje ensamblador DLX. Para ello, el simulador representa las dependencias que existen entre las instrucciones, y ofrece información acerca del número de ciclos que ha necesitado la ejecución del procesador, los ciclos que se han perdido por dependencias (de datos, de control, o estructurales) entre instrucciones, etc.

4.1 Realización de la Práctica

Elegir uno de entre los ejercicios propuestos y realizar los siguientes pasos:

- Escribir el código asociado al ejercicio en código ensamblador DLX.
- Indicar el número de ciclos de reloj empleados.
- Analizar detalladamente el diagrama de ciclos de reloj: las dependencias entre instrucciones (diferenciando entre tener activada la opción de *bypass* o no tenerla activada), y los riesgos de interrupción del *pipeline*.
- Aplicar técnicas de optimización de código para eliminar bloqueos.
- Emplear instrucciones multiciclo y analizar la información presentada en el diagrama de ciclos de reloj.

4.2 Evaluación

Al finalizar la práctica el alumno debe entender y dominar los siguientes conceptos:

- Computador DLX: registros, secuencia de fases, programación en ensamblador.
- Análisis de resultados presentados por el Simulador WinDLX.
- Técnicas de optimización de código.

4.3 Referencias bibliográficas

- Grünbacher, H.: "WinDLX V1.2. A DLX-Simulator for MS-Windows". Vienna University of Technology. Enero, 1992.

4.4 Ejercicios Propuestos

Ejercicio 1

Para $i = 0$ hasta $i = 9$, con $i = i + 1$

$$B = \sum A[i]$$

Ejercicio 2

Para $i = 0$ hasta $i = 9$, con $i = i + 1$

$$A[i] = B * A[i], \text{ siendo } B = 2$$

Ejercicio 3

Para $i = 0$ hasta $i = 9$, con $i = i + 1$

$$C[i] = A[i] - B[i]$$

Ejercicio 4

Para $i = 0$ hasta $i = 9$, con $i = i + 1$

$$B = A[9] - A[8] - A[7] - \dots - A[0]$$

Ejercicio 5

Para $i = 0$ hasta $i = 9$, con $i = i + 1$

Para $j = 9$ hasta $j = 0$, con $j = j - 1$

$$C[i] = A[i] + B[j]$$

Ejercicio 6

Para $i = 0$ hasta $i = 9$, con $i = i + 1$

$$A[0] = 0, A[1] = 1, A[2] = A[1] + A[0], \dots, A[i] = A[i-1] + A[i-2]$$

Ejercicio 7

Sean A y B dos valores cualesquiera $B \geq 10$, obtener C de la forma siguiente:

$$C = A + A + \dots + A; \text{ hacer la suma } B \text{ veces}$$

Por ejemplo, si $A = 2$ y $B = 10$, entonces $C = 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2$;

Comprobar el resultado

Ejercicio 8

Para $i = 0$ hasta $i = 4$, con $i = i + 1$

$$C = C + A[i] + B[i]; \text{ con } C \text{ inicializada a } 0$$

Ejercicio 9

Para $i = 0$ hasta $i = 9$, con $i = i + 1$

Sea A una cadena de 10 elementos alguno de ellos con valor 0

B = número de ceros que contiene la cadena A

Ejercicio 10

Para $i = 0$ hasta $i = 9$, con $i = i + 1$ y para $j = 9$ hasta $j = 0$, con $j = j - 1$

$B[j] = A[i]$