

Práctica 3: Simulador de un procesador fuera de orden

Los objetivos de esta práctica son: i) entender y modelar un procesador segmentado con varios caminos de ejecución, reorder buffer, renombre de registros y lanzamiento de instrucciones fuera de orden (OOO), ii) conocer en detalle el papel de la ventana de instrucciones y modelarla para nuestro procesador, iii) estudiar el comportamiento y cuantificar las prestaciones del procesador.

1.Descripción del simulador de partida

En esta práctica se estudia y posteriormente se modifica un simulador de un procesador segmentado con tres caminos de ejecución, reorder buffer, renombre de registros y lanzamiento de instrucciones en desorden. El renombre se realiza en la etapa de Decodificación mediante una tabla de renombre. Se modela un banco de registros físicos único, sobre el que se mapean tanto las versiones consolidadas como las no consolidadas. El simulador consta de varios ficheros en C que se encuentran en:

```
pilgor:/export/extra/alumnos/a01/OOO/
```

Debes copiar todos los ficheros puesto que algunos han sufrido modificaciones respecto a la versión de la practica 2.

Los ficheros registros.c rob.c e iw.c contienen las estructuras de datos y las funciones necesarias para trabajar con renombre de registros, reorder buffer y ventana de instrucciones respectivamente. El fichero cpu.c contiene el modelo del procesador que incluye las llamadas a las funciones que implementan los citados componentes.

Se ha mejorado la ejecución paso a paso. Como en las versiones de las prácticas anteriores, el fichero chivato.c contiene el núcleo del visualizador paso a paso. Además, se usan funciones específicas del ROB, de la IW, y del banco de registros para visualizar su contenido. Cada ciclo se muestran por pantalla las instrucciones en la etapa de Búsqueda y Decodificación, las que contiene la IW y las del ROB. Entre las instrucciones de la IW se marcan las preparadas y se destacan en color los registros fuente NO-disponibles. Entre las instrucciones del ROB se distinguen por colores las que esperan ser lanzadas, las que están en ejecución y las ya terminadas. Mediante el comando "pf" se vuelca el estado (disponible o NO) de cada uno de los registros físicos.

2.Trabajo a realizar

2.1. Estudio del modelo

Antes de empezar a modificar el modelo debes entenderlo completamente. Estudia las estructuras de datos y las funciones que implementan el renombre de registros, el Reorder Buffer y la ventana de instrucciones. Analiza el código del procesador en el fichero `cpu.c` hasta entender completamente el modelo. Copia el fichero binario `simseg` desde el directorio `OOO` (tiene activada la opción `STEP`). Ejecuta un ejemplo y analiza la salida paso a paso hasta entender su comportamiento.

2.2. Modelado de la ventana de instrucciones

La función `IWsaca()` del fichero `iw.c` está vacía. Se trata de diseñar y escribir el código de esta función según la siguiente especificación.

La función recibe como parámetro el registro de desplazamiento que modela la ocupación del puerto de escritura del banco de registros, y un booleano que indica si la unidad de ejecución de punto flotante está libre (las otras unidades funcionales nunca provocan riesgo estructural, son totalmente segmentadas). La función deberá seleccionar la instrucción mas vieja de la ventana que tenga sus operandos fuentes preparados, su unidad funcional libre y el puerto de escritura libre cuando lo necesite.

El banco de registros nos ofrece la función `Rdisponible()`, que recibe como parámetro un número de registro físico y responde con un booleano que indica si el registro está disponible o no.

Modelamos la IW de forma que si tenemos `N` instrucciones en la ventana, éstas ocupan las posiciones `0..N-1`, y están ordenadas por edad. La posición cero contiene la instrucción más vieja y la `N-1` contiene la instrucción mas joven.

La función `IWsaca` devuelve una estructura del tipo `IReg`, que contiene los datos de la instrucción seleccionada o la instrucción inula si no hay ninguna que se pueda ejecutar.

2.3. Calibrado de las latencias

La tabla de latencias del fichero `cpu.c` es la misma que usamos en la práctica anterior (`latenciasWR`). Sin embargo, el modelo del segmentado ha cambiado y han aparecido nuevas etapas. Recordemos que la etapa `D` (Decodificación) se ha partido en tres etapas: `D` (Decodificación), `I` (lanzamiento desde IW) y `R` (lectura de banco de registros). Analiza el código del fichero `cpu.c` para descubrir la forma en que se refleja este cambio.

2.4. Actualización de la monitorización

Diseña un conjunto de contadores para monitorizar las razones de parada en la salida de la etapa `D`, en la salida de la IW y en la salida de ROB.

2.5. Experimentación y análisis de resultados

Para el benchmark `matrizc`, medir tiempos de ejecución para los distintos modelos.