EI1061

PRÁCTICA 1

PROCESADOR SEGMENTADO ESCALAR CON PLANIFICACIÓN DINÁMICA

ENUNCIADO

El objetivo de esta práctica es implementar un simulador de un procesador segmentado, escalar con planificación dinámica.

Características del procesador

- Cauce segmentado. Procesa una instrucción en cada una de sus etapas: ISS, EX, WB.
- Ejecución de instrucciones fuera de orden (planificación dinámica).
- El procesador dispone de estructuras de almacenamiento para la planificación dinámica: estaciones de reserva (ER). Tiene una estación de reserva por tipo de instrucción y unidad funcional. El tamaño de estas estructuras de almacenamiento es ilimitado.
- Características de las unidades funcionales (UF) donde se realizan las operaciones:
    - No segmentadas.
    - El procesador tiene las siguientes unidades funcionales no segmentadas:
        - 1 UF de Carga/almacenamiento. Tiempo de operación 2 ciclos.
        - 1 UF de suma/resta de FP. Tiempo de operación 3 ciclos.
        - 1 UF de multiplicación de FP. Tiempo de operación 5 ciclos.
    - Cada UF dispone de una ER.
    - La etapa EX puede tener en ejecución tantas instrucciones cómo unidades funcionales disponga, aunque sólo envía a ejecutar una instrucción por ciclo.

- Juego de instrucciones: El juego de instrucciones está definido en la siguiente tabla:

  ld rt, inm(rs) (rt) = (MEM(inm+(rs)))

  sd rt, inm(rs) (MEM(inm+(rs))) = (rt)

  fadd rd, rs, rt (rd) = (rs) +(rt)

  fsub rd, rs, rt (rd) = (rs) - (rt)

  fmult rd, rs, rt rd = (rs) * (rt)

  donde rs y rt son registros que contienen los operandos fuente de las instrucciones. Excepto la instrucción ld, (carga desde memoria) donde rt será el registro en el que se almacenará el contenido leído de memoria.

Descripción de las etapas de ejecución

Etapa ISS

- Inserta la instrucción en la ER correspondiente:
    - Busca una línea libre en la ER de la UF donde se ejecutará la instrucción.
    - Marca línea como ocupada y actualiza el resto de los campos
    - Si operandos disponibles (valor válido) en los registros, se cargan en ER, sino se pone la etiqueta de la línea ER que proporcionará ese resultado. Según la instrucción debe buscar un número distinto de operandos:
        - ALU: dos operandos
        - ld un operando y dato inmediato (está en la instrucción)
        - sd: dos operandos y dato inmediato (está en la instrucción)
- Actualiza el contenido registro destino poniendo a no válido y apuntando a la etiqueta de la línea de ER donde está almacenada la instrucción que genera ese resultado.

Etapa EX:

- Revisa todas las unidades funcionales que están en ejecución realizando una operación e incrementa un ciclo.
- Si alguna UF ha llegado al último ciclo de la operación, genera el resultado y lo marca como válido. No se libera hasta que se ejecute etapa WB.
- Si alguna UF no está en funcionamiento, se revisa en la ER de la UF para comprobar si hay alguna instrucción con los operandos disponibles para que pueda ser enviada a realizar la operación. Se es así se inicializa la UF correspondiente inicializando todos sus campos e iniciando operación (contador de ciclos a 1).
- Funcionalidad de cada UF:
    - MEM: Ejecuta operaciones de acceso a memoria.
        - Tiene que calcular la dirección de acceso a memoria Inm+(rs).
        - Según operación:
            - sd: Calcula la dirección de memoria donde almacenar el dato. Realiza una escritura en memoria del dato rt en la dirección de memoria calculada.
            - ld: Calcula la dirección de memoria. Deben transcurrir los ciclos de operación para generar el dato leído de memoria. EN el último ciclo lee ese dato de memoria y será el resultado generado que marcará como válido.
    - ALU: Ejecuta operaciones aritméticas de suma y multiplicación.
        - Cualquier unidad funcional de este tipo genera un resultado después de transcurridos los ciclos de operación. Cuando finaliza la operación marcará genera el resultado y lo marca como válido.

Etapa WB:

- Revisa todas las unidades funcionales y si alguna tiene marcada el resultado como válido y se ha validado en un ciclo anterior al actual:
    - En una instrucción sd no hace nada.
    - Se actualizan todas las líneas de las ER que estén pendientes de ese resultado con ese valor y se marcan operandos como válidos.
    - Se libera la UF desde la que se ha actualizado el resultado. En ese mismo ciclo se puede iniciar una nueva operación en ella.
    - Se libera la línea de la ER desde la que se ha enviado la instrucción.
    - Escribe el resultado en el registro destino que tenga marcado su contenido como no válido y apunte a la línea de TAG_ER cuya operación acaba de finalizar en la UF

Ejemplos de simulación Y PRUEBAS DE VERIFICACIÓN

A continuación, se muestran unos ejemplos sencillos de cómo se deben desarrollar los ciclos de ejecución según las características de funcionamiento enunciadas.

Ld f2,0(x1)	ISS	EX1	EX2	WB (Escribe f2 y Estación de reserva (f2))				
fadd f4,f2,f5		ISS	(RAW f2)	(RAW f2)	EX1	EX2	EX3	WB

fadd f2,f3,f4	ISS	EX1	EX2	EX3	WB (libera UF)			
fadd f4,f5,f6		ISS	R. estructural	R. estructural	EX1	EX2	EX3	WB

fadd f2,f3,f4	ISS	EX1	EX2	EX3	WB				
fadd f5,f2,f5		ISS	RAW (f2)	RAW f2	RAW (f2)	EX1	EX2	EX3	WB

ld f2,0(x1)	ISS	EX1	EX2	WB (unidad funcional libre)		
sd f3,0(x1)		ISS	R. estructural	EX1	EX2	WB (no hace nada)

sd f2,0(x1)	ISS	EX1	EX2	WB (UF libre y dato escrito en memoria)		
Ld f3, 0(x2)		ISS	STALL (RAW y estructural)	EX1	EX2	WB

fadd f2,f3,f4	ISS	EX1	EX2	EX3	WB				
fadd f4,f2,f6		ISS	RAW f2	RAW F2	RAW F2	EX1	EX2	EX3	WB