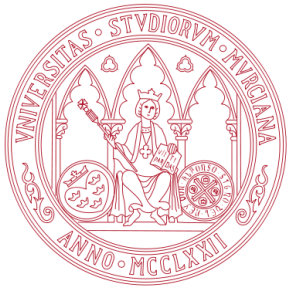

Universidad de Murcia

Facultad de Informática

TÍTULO DE GRADO EN
INGENIERÍA INFORMÁTICA

Ampliación de Estructura de Computadores

Tema 3: Segmentación avanzada

Boletín de ejercicios

CURSO 2024 / 2025

Departamento de Ingeniería y Tecnología de Computadores

Área de Arquitectura y Tecnología de Computadores

Ejercicios Básicos

1. Tenemos el siguiente código:

```
Loop:   BEQZ R1,Cero  ; S1
        SUBI R1,R1,#1
        BEQZ R1,Sigue ; S2
Cero:   ADDI R1,R1,#1
Sigue:  SUBI R2,R2,#1
        BNEZ R2,Loop  ; S3
```

   a) Indica cómo se comportaría un predictor de saltos con correlación que utiliza contadores saturados de un bit con un histórico de un salto (1,1), como el visto en clase, con este código si inicialmente R1=0 y R2=100. Utiliza una tabla como la que se muestra para detallar las primeras 5 iteraciones. ¿Qué ocurre con el salto S1?.

r1	r2	Salto	Último salto	Predicción	Comportamiento	Nueva Predicción
0	100	S1	NT	**NT**/NT	T	**T**/NT
		S2	...	...	...	...

   OJO: la tabla que se muestra es un ejemplo del formato, no tiene por qué coincidir con el comportamiento real de los saltos.

   b) Cambiamos el predictor anterior por otro de un bit pero con dos bits de correlación donde el bit menos significativo (derecha) contiene el salto más reciente. Mostrar en una tabla como la anterior el comportamiento ahora. ¿Qué ocurre ahora con el salto S1?

2. Tenemos un procesador con el siguiente pipeline: $F\ D\ X_1\ M_1\ X_2\ M_2\ W$ donde se calcula el destino del salto en la etapa X_1 y la condición en la etapa X_2. Tenemos un 20 % de saltos, con un 60 % de los mismos tomados.

   a) Ponemos un BTB en la etapa F. La probabilidad de acertar en el buffer es del 80 % y la de acertar la predicción es del 90 %. Se pide calcular la mejora de rendimiento obtenida con respecto a utilizar la estrategia de predecir estáticamente el salto como no tomado.

   b) En lugar de la opción anterior, empleamos un predictor en la etapa X_1 y anulamos las instrucciones entradas erróneamente cuando la predicción es tomado. La tasa de acierto es del 95 %. ¿Cuál es el rendimiento ahora frente a la predicción estática del salto como no tomado?

   c) Como última opción decidimos unir las dos anteriores. De esta forma el BTB ofrece una predicción inicial que puede coincidir o no con la del predictor. En caso de conflicto, el predictor se considera más fiable que el BTB. Compara el rendimiento con respecto a las dos opciones anteriores.

3. Suponer un nuevo procesador que emplea una estructura de *pipeline* similar a la estudiada en clase para DLX pero con 6 etapas (*F D X1 M X2 W*). La etapa *X2* permite añadir al ISA del procesador instrucciones de salto condicional del tipo *BEQZM/BNEZM Cte(Registro), Destino*, las cuales se comportan como *BEQZ/BNEZ* salvo que el operando sobre el que se va a hacer la comparación no está en un registro sino en la posición de memoria indicada en la instrucción. Sabiendo que los saltos de tipo *BEQZ/BNEZ* se resuelven (dirección destino y condición) en la etapa *D* y los de tipo *BEQZM/BNEZM* resuelven la dirección destino en la etapa *D* y la condición en la *X2*, que para el primer tipo de saltos se emplea una predicción estática del salto como no tomado mientras que los del segundo tipo se predicen estáticamente como tomados, y suponiendo que los saltos del primer tipo son el triple de frecuentes que los del segundo tipo y la siguiente distribución de instrucciones:

Tipo de instrucción	Porcentaje
ALU	40 %
Cargas	20 %
Almacenamientos	10 %
Saltos condicionales	30 % (80 % tomados, 20 % no tomados)

a) Explicar brevemente la unidad o unidades funcionales que habría que añadir para implementar la etapa *X2*, mostrando el trabajo que se realiza en cada etapa del pipeline para la ejecución una instrucción del tipo *BEQZM/BNEZM*.

b) Calcular el CPI de los saltos condicionales.

c) Como opción alternativa añadimos un BTB. La probabilidad de acertar en el buffer es del 80 % y la de acertar la predicción es del 90 %. ¿Cuál es el CPI se obtiene ahora para los saltos condicionales?

d) Finalmente, el equipo de diseño se plantea la posibilidad de usar la segmentación en 5 etapas estudiada en clase para el procesador DLX, utilizando la estrategia de predecir estáticamente los saltos condicionales como no tomados. Para ello se considera la eliminación de las instrucciones del tipo *BEQZM/BNEZM* del ISA del procesador y su tratamiento como pseudoinstrucciones, que serán traducidas por el compilador en las instrucciones máquina correspondientes. ¿Cuál sería ahora el CPI de los saltos?

e) De entre las cuatro alternativas anteriores (b, c y d), ¿cuál daría lugar a una versión más rápida del procesador? Supón que el tiempo de ciclo no se ve afectado por ninguno de los cambios introducidos.

4. Suponiendo que el valor inicial del R3 es R2+80, y usando el siguiente código:

```
Loop: LD    F0,0(R2)
      LD    F2,8(R2)
      MULTD F0,F0,F2
      ADDI  R2,R2,#8
      ADDD  F4,F0,F4
      SUB   R4,R3,R2
      BNEZ  R4,Loop
```

Para ejecutarlo en un procesador DLX de punto flotante en el que:

- se puede hacer una lectura y una escritura del banco de registros en el mismo ciclo de reloj,
- los saltos se manejan deteniendo el cauce y no se resuelven hasta el final de la etapa ID,
- las unidades en punto flotante están segmentadas y la suma tarda 2 ciclos y la multiplicación 3,
- solo las intrucciones enteras pasan por la etapa MEM,
- y la etapa WB es distinta para las instrucciones enteras y de coma flotante.

a) Muestra la temporización de la secuencia de instrucciones considerando que todas las referencias a memoria aciertan en caché para un cauce **sin** adelantamiento hardware. ¿Cuántos ciclos tardará en ejecutarse el código?

b) Muestra la temporización de la secuencia de instrucciones considerando que todas las referencias a memoria aciertan en caché para un cauce **con** adelantamiento hardware. ¿Cuántos ciclos tardará en ejecutarse el código? Señala los adelantamientos que se produzcan.

c) Vamos a suponer que poner adelantamiento hardware incrementa la duración del ciclo de reloj un 10 %. ¿Cuál sería la mejora de implementar adelantamiento hardware?

Ejercicios Adicionales

5. Tenemos un procesador con el siguiente pipeline F_1 F_2 D X_1 M_1 M_2 X_2 W que permite ejecutar saltos condicionales de la forma *BGE R1,100(R2), Label* calculando la dirección de destino en la etapa D y la condición en la etapa X_2. En la etapa D también tenemos un predictor de saltos con un 95 % de eficacia. Tras hacer simulaciones hemos comprobado que el 60 % de los saltos son tomados y el 40 % no tomados. En cada caso el pipeline predice el comportamiento de los saltos anulando, si es necesario, las instrucciones que entraran incorrectamente. Queremos ejecutar unos benchmarks que tienen un 20 % de saltos condicionales, de los cuales la cuarta parte son del tipo anterior y el resto únicamente usan registros. Para los saltos que sólo usan registros, se predicen estáticamente como no tomados y se conoce tanto la condición como la dirección de destino del salto en la etapa D. Por otro lado tenemos un segundo procesador con el pipeline visto en clase F D X M W y un BTB en el que encontramos el salto con un 95 % de probabilidad y acertamos en la predicción un 90 % de las veces. Sin embargo, este procesador no tiene la instrucción de salto anterior, así que el compilador genera el mismo código que en el caso anterior pero sustituyendo cada salto con referencia a memoria por dos instrucciones: *LW R3,100(R2) / BGE R1,R3,Label*. Se pide:

   a) Calcular el CPI del primer procesador con el programa dado.

   b) Calcular el CPI del segundo procesador con el programa dado.

   c) Si sabemos que el primer procesador funciona a una frecuencia de 1 GHz, ¿a qué frecuencia tenemos que hacer trabajar el segundo para que tenga el mismo rendimiento?

6. Tenemos un procesador con el siguiente pipeline: F D X_1 X_2 M_1 M_2 W. Donde se calcula la dirección de destino de los saltos en la etapa D y la condición en X_2. Queremos comparar su rendimiento con varias opciones de diseño para el tratamiento de los saltos, sabiendo que tenemos una frecuencia de saltos condicionales del 20 %, de los cuales el 60 % se toman.

   i) El salto se predice como no tomado.

   ii) Usamos un predictor dinámico en la etapa de decodificación con una tasa de acierto del 97 %.

   iii) Un buffer de destino de saltos (BTB) como el visto en clase para DLX con su mismo funcionamiento, con un 80 % de tasa de acierto en buffer y un 90 % de tasa de acierto en la predicción.

   Dado lo anterior, se pide:

   a) Calcula el CPI y compara el rendimiento de las configuraciones i) y ii).

   b) Calcula el CPI y compara el rendimiento de las configuraciones ii) y iii).

   c) Estamos pensando en agrupar las etapas X_1 y X_2 en el mismo ciclo, pero entonces la duración del ciclo aumentaría un 30 %. ¿Nos convendría esta modificación si usamos la configuración i) en estas condiciones, con respecto a la antigua ii)?

7. Supongamos un procesador con una frecuencia de reloj 3.6 GHz y un pipeline estructurado en las siguientes etapas: *F D1 D2 R X1 X2 M1 M2 WB*. La búsqueda de la instrucción se realiza durante la etapa *F*, su decodificación durante las etapas *D1* y *D2*, la lectura de los operandos durante la etapa *R*, la ejecución de las instrucciones en las etapas *X1* y *X2*, los accesos a memoria en las etapas *M1* y *M2*, y la escritura de resultados en la etapa *WB*. Además, en la etapa *X1* se

realiza el cálculo de la dirección destino de los saltos, y en la etapa *X2* se calcula la condición del salto. Se sabe que el 40 % de las instrucciones ejecutadas por el procesador son de tipo ALU, el 25 % son cargas, el 10 % almacenamientos y el 25 % saltos. De las instrucciones de salto se ha observado que el 80 % son saltos condicionales, de los cuales sólo el 40 % son tomados. Por otra parte también se ha evaluado la influencia de los riesgos que puedan aparecer. En concreto, se ha observado que el 30 % de las operaciones ALU y el 25 % de las cargas presentan riesgo de datos que implica 1 ciclo de detención, y para el 20 % de los almacenamientos esos riesgos suponen 2 ciclos de parada. Sin embargo, el pipeline se ha diseñado de forma que no existen riesgos estructurales. Respecto a los riesgos de control, se intenta paliar su efecto negativo usando un BTB en la etapa F. La probabilidad de que se encuentre una entrada en el BTB es del 90 % y la de acertar es del 80 %. Además, para los saltos condicionales, en caso de que no se encuentre en el BTB se predice como tomado, mientras que para los saltos incondicionales, caso de no encontrar una entrada en el BTB se detiene el cauce. En caso de que el BTB falle, se pierde un ciclo extra debido a su actualización. Se pide calcular el tiempo medio por instrucción en nseg.

8. Dado el procesador estudiado en clase con un pipeline de 5 etapas (*F D X M W*), que no implementa adelantamiento (*forwarding*) y en el que los saltos se resuelven en la cuarta etapa, y suponiendo que el 45 % de las instrucciones ejecutadas son de tipo ALU, el 20 % son cargas, el 10 % almacenamientos y el 25 % saltos condicionales. Se pide:

   a) Calcular el CPI total, sabiendo que el 15 % de las instrucciones de carga y el 10 % de las instrucciones de las de ALU van seguidas inmediatamente de una instrucción ALU que usa el dato leído de memoria y calculado por la ALU respectivamente, y que el 80 % de los saltos condicionales son tomados y que se predicen estáticamente como no tomados.

   b) Dado que se implementa la técnica de adelantamiento y que se introduce en la etapa *D* un predictor de saltos con una tasa de aciertos del 90 % (los saltos siguen siendo resueltos en la cuarta etapa y las instrucciones buscadas de forma errónea son anuladas), ¿qué CPI se obtiene ahora? (Considera que la dirección destino del salto se calcula en la etapa D.)

   c) Añadimos ahora un BTB en la etapa *F*. La probabilidad de acertar en el buffer es del 85 % y la de acertar la predicción es del 70 %. El BTB ofrece una predicción inicial que puede coincidir o no con la del predictor. En caso de conflicto, el predictor se considera más fiable que el BTB. ¿Qué CPI se obtiene ahora?

   d) Se propone cambiar el diseño del pipeline por el siguiente *F D X1 M1 X2 M2 W*. Se observa que al ejecutar la misma distribución de instrucciones las de ALU tienen un CPI final de 1.4 ciclos, las cargas 2.2 ciclos, los almacenamientos 1.4 ciclos y los saltos condicionales 1.6 ciclos. Sin embargo, esta nueva versión del procesador incluye la instrucción `DBEZ 0(R1),Etiqueta` (decrementa y si es distinto de cero, salta), que no aparecía en la versión del pipeline de 5 etapas y que ahora podemos utilizar en lugar del código que había hasta este momento: `LD R2,0(R1) / SUBI R2,R2,1 / BEQZ R2,Label`. Al rehacer el programa para que utilice esta instrucción vemos que aparece en la versión final con una frecuencia del 5 % y un CPI de 1.8 ciclos. Suponiendo que el resto de las instrucciones no han variado, ¿cuál es el CPI total ahora?

9. Para el DLX de enteros habitual y el siguiente fragmento de código:

```
Loop: LW    R1,0(R2)
      ADDI  R1,R1,#1
      SW    0(R2),R1
      ADDI  R2,R2,#4
      SUB   R4,R3,R2
      BNEZ  R4,Loop
```

Suponer que el valor inicial de R3 es R2+396, y que todos los accesos a memoria cache son aciertos:

a) Muestra la temporización de esta secuencia de instrucciones para el cauce DLX sin ningún adelantamiento hardware, pero suponiendo que se puede hacer una lectura y una escritura de registros en el mismo ciclo de reloj. Suponer que el salto se maneja parando el cauce y que no se resuelve hasta el final de la etapa MEM. Si todas las referencias a memoria aciertan en cache, ¿cuántos ciclos tardará en ejecutarse este bucle?

b) Muestra la temporización de esta secuencia de instrucciones para el cauce DLX con adelantamiento hardware. Supón que el salto se maneja prediciéndolo como no-tomado y que no se resuelve hasta el final de la etapa MEM. Si todas las referencias a memoria aciertan en cache, ¿cuántos ciclos tardará en ejecutarse este bucle?

c) Se ha considerado que poner adelantamiento hardware y predicción de saltos incrementa la duración del ciclo de reloj un 90 %. Suponiendo que este programa es significativo como representación de los programas que se van a usar, evalúa si son interesantes cada una de las mejoras anteriores y cuál sería la ganancia con cada una de las propuestas.

10. Se tiene un procesador segmentado en 5 etapas (*IF*: búsqueda; *ID*: decod. y lectura de registros; *EX*: ALU, unidad detectora de ceros y cálculo de la dirección destino; *M*: acceso a memoria y *WB*: escritura de registro). Incorpora el repertorio de instrucciones del DLX y posee cache de instrucciones y de datos. La frecuencia de reloj es de 200 MHz. Las instrucciones de salto utilizan una estrategia $predict - not - taken$. Usa adelantamiento hardware.

    Los programas ejecutan por término medio un 18 % de saltos, 33 % de cargas/almacenamientos y un 49 % de instrucciones aritméticas. Las cargas son el doble de frecuentes que los almacenamientos. Un 55 % de las instrucciones de carga van seguidas de otra instrucción que consume el dato procedente de la memoria, debiendo insertar los ciclos de parada necesarios. El 70 % de las instrucciones de salto no se toman.

    Con el objeto de mejorar las prestaciones, el equipo de diseño plantea realizar las siguientes modificaciones (ambas a la vez):

    - Ubicar la cache de datos fuera de la pastilla del procesador. El tiempo de acceso a la cache de datos es ahora de 3 ciclos de reloj. La etapa *M* del ciclo de instrucción se convierte en 3 etapas: M_1, M_2 y M_3, quedando la segmentación en 7 etapas.
    - Aumentar la frecuencia de reloj, al simplificar el diseño del procesador.

    Se pide calcular:

    a) Tiempo de ejecución de la máquina original.

    b) CPI de la máquina modificada.

    c) La frecuencia de reloj que debería alcanzarse, como mínimo, para que sea interesante incorporar las modificaciones descritas.

11. Suponiendo que el valor inicial del R4 es R2+792, y usando el siguiente código:

```
Loop: LD    F0,0(R2)
      LD    F4,0(R3)
      MULTD F0,F0,F4
      ADDD  F2,F0,F2
      ADDI  R2,R2,#8
      ADDI  R3,R3,#8
      SUB   R5,R4,R2
      BNEZ  R5,Loop
```

a) Muestra la temporización de esta secuencia de instrucciones para el cauce DLX punto flotante sin adelantamiento hardware, pero suponiendo que se puede hacer una lectura y una escritura del banco de registros en el mismo ciclo de reloj. Todas las intrucciones pasan por la etapa MEM, incluso las de punto flotante, y la etapa WB es común para las instrucciones enteras y de coma flotante. Vamos a suponer que el salto se maneja deteniendo el cauce y que no se resuelve hasta el final de la etapa MEM. Si todas las referencias a memoria aciertan en cache, ¿cuántos ciclos tardará en ejecutarse este bucle?.

b) Muestra la temporización de esta secuencia de instrucciones para el cauce DLX punto flotante con adelantamiento hardware. Todas las intrucciones pasan por la etapa MEM, incluso las de punto flotante, y la etapa WB es común para las instrucciones enteras y de coma flotante. Vamos a suponer que el salto se maneja prediciéndolo como no-tomado, pero que no se resuelve hasta el final de la etapa MEM. Si todas las referencias a memoria aciertan en cache, ¿cuántos ciclos tardará en ejecutarse este bucle?. Señala los adelantamientos que se produzcan.

c) Vamos a suponer que poner adelantamiento hardware y/o predicción de saltos incrementa la duración del ciclo de reloj un 50 %. Suponiendo que este programa es significativo como representante de los programas que se van a usar, evalúa si son interesantes cada una de las mejoras anteriores y cuál sería la ganancia con cada una de las propuestas.