Arquitectura e Ingeniería de Computadores - 3er Curso. Sol. del Examen 1er parcial 5/11/2018

Responde cada pregunta en una hoja distinta. Tiempo disponible: 2h30m

1. (2,5 **puntos)** Sea un sistema formado por un procesador K compuesto por 8 núcleos a 3Ghz, un hardware digitalizador de audio y un programa P destinado al reconocimiento de palabras clave en conversaciones telefónicas. Dicho sistema digitaliza las grabaciones de audio que se le suministran, aplicando a continuación una transformada rápida de Fourier (FFT) para descomponer los sonidos en frecuencias y alimentar así a una red neuronal recurrente (RNN) que identifica las palabras activando las alarmas correspondientes. El sistema tiene un coste de 4000€.

   Diariamente llegan en promedio 1360 horas de grabaciones, pero sólo pueden ser procesadas por el sistema con la configuración actual 800 horas en ese mismo periodo de tiempo.

   Tras hacer un *profiling* durante 24 horas se observa que el sistema ha empleado un 10 % del tiempo en digitalizar los audios suministrados, un 35 % del tiempo en calcular la FFT, un 45 % en procesar los datos en la red neuronal y el tiempo restante ha sido empleado en almacenar los resultados y mostrar las alarmas.

   Se proponen diversas opciones de mejora con la intención de incrementar la productividad como mínimo a las 1360h/día necesarias para poder llevar los análisis al día.

   Las opciones propuestas son la siguientes:

   I Reemplazar el digitalizador actual por uno un 250 % más rápido, con un coste de 500€ y reentrenar la red neuronal para que pueda funcionar con los datos originales evitando así la necesidad de calcular la FFT. El coste de reentrenar la red es de 3000€.

   II Instalar una GPU Tesla P4 que acelera la ejecución de la red neuronal un 1200 %, con un coste de 2500€.

   III Adquirir una nueva unidad del sistema actual y repartir la carga de audio equitativamente entre ambos.

   Se pide:

   a) ¿Qué mínimo porcentaje de mejora se requiere para alcanzar la productividad deseada de 1360 h/día?

   b) Calcular la mejora obtenida en los tres casos propuestos.

   c) Según el criterio de coste/prestaciones. ¿Cuál de las propuestas que alcanzan la productividad deseada es la más interesante?

   Solución:

   a) ¿Qué mínimo porcentaje de mejora se requiere para alcanzar la productividad deseada de 1360 h/día?
   $S = \frac{1360}{800} = 1{,}7 \rightarrow 70\,\%$

   b) Calcular la mejora obtenida en los tres casos propuestos.
   - Dig+FFT $\rightarrow S = \frac{1}{\frac{0{,}10}{3{,}5}+\frac{0{,}35}{\infty}+0{,}45+0{,}10} = 1{,}73, \$ = \frac{3500+4000}{4000} = 1{,}88$
   - RNN $\rightarrow S = \frac{1}{0{,}10+0{,}35+\frac{0{,}45}{13}+0{,}10} = 1{,}71, \$ = \frac{2500+4000}{4000} = 1{,}63$
   - Replica $\rightarrow$ S=2, $=2

   c) Según el criterio de coste/prestaciones. ¿Cuál de las propuestas que alcanzan la productividad deseada es la más interesante?
   La CNN, obtiene una mejora del 71 %, suficiente, con un incremento de coste del 63 %.

□

2. (2,5 **puntos)** En un computador similar al MIPS se ejecuta un benchmark representativo de la carga del sistema obteniendo las siguientes estadísticas para las distintas categorías de instrucciones:

Operación	%	CPI
ALU	45	1.2
Load	22	1
Store	13	1
Saltos	20	1.8

Se ha detectado que los ciclos de parada de los saltos son debidos a riesgos de datos con las instrucciones previas. Para evitar esta situación se plantea introducir una nueva instrucción de salto con acceso a memoria, *salto-mem*, con el siguiente formato: `beqz (reg), etiq`. Las instrucciones *salto-mem* tendrían CPI igual a 2 y permitirían sustituir a una instrucción de tipo *load* y una instrucción de *salto* tal como se muestra en el ejemplo a continuación.

```
código original       |     nuevo código
----------------------|--------------------------
ld r1,0(r10)          |     beqz (r10), destino
beqz r1, destino      |
```

Esta sustitución se podría llevar a cabo en el 50 % de los saltos, quedando el resto de los saltos con un CPI de 1,3. Sin embargo, dado que la nueva instrucción no tiene desplazamiento para el calculo de la dirección, en el 20 % de las sustituciones sería necesario añadir una nueva instrucción de tipo ALU que mantendría su CPI original ($CPI_{\text{ALU}} = 1{,}2$).

Suponiendo que un programa ejecuta n instrucciones en el computador original, responde a las siguientes preguntas razonando y justificando todas tus respuestas:

a) Calcula el CPI promedio del computador original y su tiempo de ejecución en ciclos en función del número de instrucciones n.

b) Calcula el nuevo número de instrucciones ejecutadas al incorporar las instrucciones *salto-mem* y realizar todas las sustituciones posibles.

c) Calcula la nueva distribución de instrucciones al incorporar las instrucciones *salto-mem*. Mantened las proporciones de cada tipo de instrucciones en formato de fracción.

d) Calcula el nuevo CPI promedio del computador tras incorporar las instrucciones *salto-mem*.

e) Indica el nuevo tiempo de ejecución en ciclos en función del numero de instrucciones originales n y la aceleración obtenida con respecto al computador original.

Solución:

a) Calcula el CPI promedio del computador original.

$$\begin{aligned} CPI_{\text{org}} &= \overbrace{0{,}45 \times 1{,}2}^{\text{alu}} + \overbrace{0{,}22 \times 1}^{\text{load}} + \overbrace{0{,}13 \times 1}^{\text{store}} + \overbrace{0{,}20 \times 1{,}8}^{\text{salto}} \\ &= 0{,}54 + 0{,}22 + 0{,}13 + 0{,}36 = 1{,}25 \end{aligned}$$

$$T_{ej\ org} = I \times CPI \times T = n \times 1{,}25 = 1{,}25\, n \text{ ciclos}$$

b) Calcula el nuevo número de instrucciones ejecutadas al incorporar las instrucciones *salto-mem* y realizar todas las sustituciones posibles.
En el 50 % de los saltos (20 % de las ejecutadas), se sustituye un *salto* y un *load* por una nueva instrucción de *salto-mem*. Por tanto, el número de instrucciones disminuye en 1 en cada sustitución. Sin embargo en el 20 % de dichas sustituciones se añade una nueva *alu*. Por lo tanto:

$$I_a = I \times (1 - 0{,}5 \times 0{,}20 \times 1 + 0{,}5 \times 0{,}20 \times 0{,}20 \times 1) = (1 - 0,08)\ I = 0,92\ I$$

c) Calcula la nueva distribución de instrucciones al incorporar las instrucciones *salto-mem*. Mantened las proporciones de cada tipo de instrucciones en formato de fracción.
Las instrucciones ALU aumentan, ya que en el 20 % de las sustituciones se añade una nueva *alu*. Por otra parte, en cada sustitución se elimina una instrucción *Load* seguida de un Saltos, cambiándola por la nueva instrucción de *salto-mem*:

Operación	#	%	CPI
ALU	0,45+0,2 × 0,5 × 0,2 × 1 = 0,47	$\frac{0,47}{0,92} = 51$	1.2
Load	0,22-0,2 × 0,5 × 1 = 0,12	$\frac{0,12}{0,92} = 13$	1
Store	0,13	$\frac{0,13}{0,92} = 14$	1
Saltos	0,20-0,2 × 0,5 × 1 = 0,10	$\frac{0,10}{0,92} = 11$	1.3
salto-mem	0,2 × 0,5 × 1 = 0,10	$\frac{0,10}{0,92} = 11$	2
Total	0,92	100	

d) Calcula el nuevo CPI promedio del computador tras incorporar las instrucciones *salto-mem*.

$$CPI = \frac{\overbrace{0,47 \times 1,2}^{\text{alu}} + \overbrace{0,12 \times 1}^{\text{load}} + \overbrace{0,13 \times 1}^{\text{store}} + \overbrace{0,10 \times 1,3}^{\text{salto}} + \overbrace{0,10 \times 2}^{\text{salto-mem}}}{0,92} = 1,245$$

Puesto que ya tenemos calculado el % de aparición de cada instrucción, otra sería:

$$CPI = \overbrace{0,51 \times 1,2}^{\text{alu}} + \overbrace{0,13 \times 1}^{\text{load}} + \overbrace{0,14 \times 1}^{\text{store}} + \overbrace{0,11 \times 1,3}^{\text{salto}} + \overbrace{0,11 \times 2}^{\text{salto-mem}} = 1,245$$

e) Indica el nuevo tiempo de ejecución en ciclos en función del numero de instrucciones originales n y la aceleración obtenida con respecto al computador original.

$$T_{ej\ a} = I \times CPI \times T = 0,92 \times n \times 1,245 = 1,145\ n \text{ ciclos}$$

$$S = \frac{1,25 \cdot n}{1,145 \cdot n} = 1,09$$

El nuevo procesador es 1,09 veces más rápido que el original (un 9 % más rápido).

□

3. **(**2,5 **puntos)** Se dispone de un procesador MIPS con los siguientes operadores multiciclo:

   - Sumador/Restador segmentado lineal. Lat= 4, IR= 1.
   - Multiplicador segmentado lineal. Lat= 5, IR= 1.

   Los riesgos estructurales y de datos se detectan en la fase ID, insertando tantos ciclos de parada como sean necesarios y utilizando cortocircuitos siempre que sea posible. Los riesgos de control se resuelven mediante predict-not-taken, teniendo en cuenta que la condición y el destino del salto se calculan en la etapa ID del salto y el PC se actualiza con la dirección correcta al final de la etapa EX.

   El código ejecutado en dicho procesador es:

```
        dadd r1, r0, x        ; inicio del vector x
        dadd r2, r0, y        ; inicio del vector y
        daddi r4, r1, #512    ; el vector x tiene 64 elementos
        l.d f0, a(r0)         ; a
        l.d f2, b(r0)         ; b
loop:
        l.d f4, 0(r1)         ; x
        l.d f6, 0(r2)         ; y
        mul.d f4, f4, f0      ; ax
        mul.d f6, f6, f2      ; by
        add.d f6, f4, f6      ; ax+by
        s.d f6, 0(r1)         ; x=ax+by
        daddi r1, r1, #8
        daddi r2, r2, #8
        dsub r5, r4, r1
```

```
bnez r5, loop
nop
nop
nop
trap 0.
```

Responda a las siguiente cuestiones utilizando la notación: `IF` fase de búsqueda, `ID` fase de decodificación, `EX` fase de ejecución monociclo, `A1,A2,A3,A4` fases de ejecución del sumador/restador, `M1,M2,M3,M4,M5` fases de ejecución del multiplicador, `ME` fase de acceso a memoria y `WB` fase de escritura en registros. Las instrucciones multiciclo no realizan la fase `ME`.

a) Complete el diagrama instrucciones-tiempo de la primera iteración del bucle incluyendo hasta la primera instrucción de la segunda iteración. Tenga en cuenta que al ser la primera iteración deberán incluirse en el diagrama las instrucciones de inicialización que hay antes de la etiqueta `loop`.

b) Calcule el CPI medio de una iteración del bucle.

c) Si en lugar de tener predict-not-taken tuviéramos salto retardado, ¿podría utilizarse el mismo código proporcionado, o tendría que modificarse obligatoriamente para que la ejecución fuera correcta? Si es necesario modificar el código, indique las lineas que cambiarían.

Solución:

a) Complete el diagrama instrucciones-tiempo de la primera iteración del bucle incluyendo hasta la primera instrucción de la segunda iteración. Tenga en cuenta que al ser la primera iteración deberá incluirse en el diagrama las instrucciones de inicialización que hay antes de la etiqueta `loop`.

```
                  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
      dadd r1,r0,x  IF ID EX ME WB
      dadd r2,r0,y     IF ID EX ME WB
      daddi r4,r1,#512    IF ID EX ME WB
      l.d f0,a(r0)           IF ID EX ME WB
      l.d f2,b(r0)              IF ID EX ME WB
loop: l.d f4,0(r1)                 IF ID EX ME WB
      l.d f6,0(r2)                 |  IF ID EX ME WB
      mul.d f4,f4,f0               |     IF ID M1 M2 M3 M4 M5 WB
      mul.d f6,f6,f2               |        IF ID M1 M2 M3 M4 M5 WB
      add.d f6,f6,f4               |           IF id id id id ID A1 A2 A3 A4 WB
      s.d f6,0(r1)                 |              if if if if IF id id ID EX ME WB
      daddi r1,r1,#8               |                                  if if IF ID EX ME WB
      daddi r2,r2,#8               |                                     IF ID EX ME WB
      dsub r5,r4,r1                |                                        IF ID EX ME WB
      bnez r5,loop                 |                                           IF id ID EX ME WB
      nop                          |                                              if IF ID x
      nop                          |                                                 IF x
loop: l.d f2,0(r1)                 |                                                  | IF ID ..
                                   |-------------------- 19 CICLOS ----------------------|
```

b) Calcule el CPI medio de una iteración del bucle.
$CPI = \frac{19\ ciclos}{10\ instrucciones} = 1{,}9$

c) Si en lugar de tener predict-not-taken tuviéramos salto retardado, ¿podría utilizarse el mismo código proporcionado, o tendría que modificarse obligatoriamente para que la ejecución fuera correcta? Si es necesario modificar el código, indique las lineas que cambiarían.
No seria necesario modificar nada. El código proporcionado podría ejecutarse perfectamente en el procesador con salto retardado, ya que el delay-slot estaría compuesto por dos instrucciones nop lo cual no afectaría a la correcta ejecución del programa.

□

4. **(**2,5 **puntos)**

Durante la ejecución de una aplicación en un procesador que dispone de un predictor dinámico de saltos BTB con 1 bit para la predicción se observa la siguiente distribución de transiciones entre estados cada vez que se ejecuta un salto cualquiera:

Transición	Estado Origen		Estado Destino	Frecuencia (%)
1	—	→	No Salta	1
2	—	→	Salta	5
3	No Salta	→	No Salta	13
4	No Salta	→	Salta	11
5	Salta	→	No Salta	2
6	Salta	→	Salta	68

Así, por ejemplo, cuando se ejecuta un salto cualquiera, un 1 % de las veces el salto no se encuentra en la BTB y pasa a almacenarse con el estado "No Salta" (transición numerada como 1 en la tabla). Por otra parte, un 11 % de las veces el salto se encuentra en la BTB con el estado "No Salta" y al ejecutarse pasa al estado "Salta" (transición 4), mientras que la gran mayoría de veces (un 68 %), el salto se encuentra como "Salta" y se mantiene en el mismo estado al ejecutarse (transición 6).

Responda a las siguientes cuestiones:

a) Identifique las transiciones de la tabla que impliquen fallo del predictor. Razone la respuesta. NOTA: Se considera que cuando el salto no se encuentra en la BTB la predicción es "No Salta" y que existe un fallo siempre que se cancelen instrucciones.

b) Asumiendo que el CPI de las instrucciones de salto en ausencia de fallo es 1 y que n es la penalización por fallo del predictor, obtenga la expresión que calcula el CPI medio de las instrucciones de salto.

c) Suponga que el procesador está segmentado en las cinco etapas habituales (IF, ID, EX, ME, WB), la condición y el destino del salto se calculan en EX, y, en caso de salto efectivo, el PC se actualiza en ME. Calcule el CPI medio de las instrucciones de salto.

d) La BTB del procesador original se sustituye por una BTB con 2 bits para la predicción que pueden representar los estados *Strongly Not Taken (SNT), Weakly Not Taken (WNT), Weakly Taken (WT), y Strongly Taken (ST)*. En la ejecución de la misma aplicación, se observa la siguiente distribución.

Transición	Estado Origen		Estado Destino	Frecuencia (%)
1	—	→	SNT	1
2	—	→	ST	4
3	SNT	→	SNT	16
4	SNT	→	WNT	6
5	WNT	→	SNT	6
6	WNT	→	ST	2
7	ST	→	ST	59
8	ST	→	WT	1
9	WT	→	ST	4
10	WT	→	SNT	1

Teniendo esto en cuenta, y asumiendo que el CPI de las instrucciones de salto en ausencia de fallo es 1 y que n es la penalización (en ciclos de reloj) por fallo del predictor, obtenga la nueva expresión para calcular el CPI medio de las instrucciones de salto.

Solución:

a) Identifique las transiciones de la tabla que impliquen fallo del predictor. Razone la respuesta.

En el caso del predictor de un bit, si una predicción con estado "Salta" o "No Salta" se actualiza con el estado contrario significa que la predicción era incorrecta y que por tanto ha implicado un fallo del predictor y cancelación de instrucciones (transiciones 4 y 5).
Por otro lado, cuando un salto no se encuentra en la BTB, se predice que no hay salto. Si el salto efectivamente salta, la predicción era incorrecta, lo que implica fallo del predictor, cancelación de instrucciones y que el salto se almacene en la BTB con el estado "Salta" (transición 2).

En resumen, las transiciones que implican un fallo del predictor son la 2, 4, y 5.

b) Asumiendo que el CPI de las instrucciones de salto en ausencia de fallo es 1 y que n es la penalización (en ciclos de reloj) por fallo del predictor, obtenga la expresión que calcula el CPI medio de las instrucciones de salto.

$$\overline{CPI} = 1 + (0{,}05 + 0{,}11 + 0{,}02) \times n = 1 + 0{,}18 \times n$$

c) Suponga que el procesador está segmetando en las cinco etapas habituales (IF, ID, EX, ME, WB), la condición y el destino del salto se calculan en EX, y, en caso de salto efectivo, el PC se actualiza en ME. Calcule el CPI medio de las instrucciones de salto.

En ese caso se cancelan 3 instrucciones cuando el predictor falla, luego $n = 3$, y por tanto:

$$\overline{CPI} = 1 + (0{,}05 + 0{,}11 + 0{,}02) \times 3 = 1{,}54$$

d) Para la BTB con 2 bits para la predicción, obtenga la nueva expresión para calcular el CPI medio de las instrucciones de salto.

$$\overline{CPI} = 1 + (0{,}04 + 0{,}06 + 0{,}02 + 0{,}01 + 0{,}01) \times n = 1 + 0{,}14 \times n$$

□