Arquitectura e Ingeniería de Computadores - 3er Curso. Sol. del Examen 1er parcial 2/11/2021
Responde cada pregunta en una hoja distinta. Tiempo disponible: 2h30m

1. **(**$2{,}5$ **puntos)** Un computador posee un procesador que funciona a 2GHz e integra una unidad de instrucción segmentada capaz de ejecutar 1 instrucción por ciclo. El procesador es capaz de resolver todos los riesgos de datos y estructurales sin necesidad de insertar ciclos de parada e integra una BTB perfecta que predice los saltos con una precisión del 100 %. La ejecución de un programa P en dicho computador tarda 1 hora y 30 minutos, repartiendo dicho tiempo en el uso del procesador y otras operaciones.

   Con el objetivo de acelerar la ejecución de P, se proponen dos mejoras:

   1. La primera consiste en cambiar la unidad de instrucción segmentada del procesador para poder soportar la ejecución de 4 instrucciones por ciclo sin necesidad de insertar ciclos de parada por riesgos estructurales, de datos o control. Sin embargo, este cambio supone reducir la frecuencia del procesador en un 10 %.
   2. La segunda consiste en cambiar el procesador original por uno con 2 núcleos. Cada uno de estos núcleos integrará una unidad de instrucción segmentada idéntica a la del procesador original del computador.

   Contesta a las siguientes preguntas:

   a) Considerando la primera mejora, ¿Cuánto mejoraría este cambio las prestaciones de la unidad de instrucción segmentada con especto a su diseño original? Expresa el resultado en porcentaje.

   b) Si el computador reduce el tiempo de ejecución de P a 50 minutos gracias a la primera de las mejoras, ¿Qué porcentaje del tiempo de ejecución de P pasaba el computador original usando el procesador? Asume en tu respuesta que la ejecución de instrucciones en el procesador no solapa con ninguna otra actividad del computador.

   c) Si finalmente se adoptara la primera mejora, ¿Qué porcentaje del tiempo de ejecución de P pasará el computador usando el procesador?

   d) ¿Cuál sería el mejor tiempo de ejecución que podríamos llegar a obtener para P mejorando únicamente la unidad de instrucción segmentada del procesador del computador?

   e) Si consideramos la segunda de las mejoras, y asumiendo que es posible repartir equitativamente la ejecución de P entre los dos núcleos disponibles, ¿cuál debería ser la frecuencia del procesador para obtener unas prestaciones globales idénticas a las ofrecidas por la primera mejora planteada?

Solución:

a) **(0.5 puntos)** Pasamos de ejecutar 1 a ejecutar 4 instrucciones por ciclo y, al mismo tiempo, la frecuencia del procesador se reduce en un 10 %, con lo que aumenta su periodo. Por tanto:
$S_{pipeline}$ = 4 * 0.9 = 3.6 $\rightarrow$ 260 %

b) **(0.5 puntos)** $S_{global} = \frac{90}{50} = 1{,}8$ y $S_{pipeline} = 3{,}6$
Por otra parte la fraccción del tiempo de ejecución que se nos pide puede calcularse aplicando la ley de Amdahl y despejando F, con lo que obtendremos que $F = \frac{S_{global} * S_{pipeline} - S_{pipeline}}{S_{global} * S_{pipeline} - S_{global}} = 0{,}6154$, con lo que el sistema original empleabla el 61.54 % del tiempo de ejecución de P utilizando el pipeline de su procesador.

c) **(0.5 puntos)** En el sistema que integra el nuevo pipeline, de cada 100 segundos empleados por el antiguo sistema, 38.46 segundos no se verán afectados por la mejora y 61.54 segundos sí, viéndose reducidos a $\frac{61{,}54}{3{,}6} = 17{,}1$ segundos, con lo que el sistema utilizará su nuevo pipeline durante el $\frac{17{,}1}{38{,}4+17{,}1} * 100 = 30{,}81\,\%$ del tiempo de ejecución de P.

d) **(0.5 puntos)** $S_{max} = \frac{1}{1-F} = 2{,}6$, con lo que dada esa mejora el mejor tiempo de ejecución posible para P será de $T_{ej_{P_{min}}} = \frac{90}{2{,}6} \approx 34{,}62 min.$

e) **(0.5 puntos)** $S_{global} = 1{,}8 \leq \frac{1}{(1-F)+\frac{F}{S_{local}}}$, en nuestro caso $S_{local} = 2 * x$, donde x es la mejora que introducimos en la frecuencia del procesador y $F = 61{,}54\,\%$. Sustituyendo y despejando obtenemos que $x \approx 1{,}8$, con lo que la frecuencia del procesador pasará de 2GHz a, como mínimo, $2 * 1{,}8 = 3{,}6$ GHz.

□

2. **(**2,5 **puntos)** Disponemos de un sistema con un procesador que funciona a 3GHz e implementa una ruta de datos segmentada con las prestaciones de cada tipo de instrucción que se enumeran en la siguiente tabla:

Tipo de instrucción	%	CPI
ALU	52	1
Saltos	24	2
Carga	19	2
Almacenamiento	5	2

El coste del sistema es de 5000€.

Después de hacer un análisis de las aplicaciones que se ejecutan en dicho sistema se obtiene la distribución de tipos de instrucciones mostradas en la tabla anterior. También se comprueba que el 20 % de las instrucciones de tipo ALU se utilizan para realizar una asignación condicional de un valor de memoria a un registro. En concreto, un dato leído de memoria se asigna a un registro si su valor es positivo. En caso contrario el valor no se asigna al registro. A continuación se muestra un ejemplo del código asociado a la operación:

```
        ...
                          ; lectura condicional de memoria
        ld r1, X(r2)      ;   lee valor de memoria en r1 (temporal)
        blt r1, r0, label1 ;   salta si r1 < 0 (negativo)
        dadd r3, r1, r0   ;   asignación a r3 (r1 es positivo)
label1: ...
```

Para mejorar las prestaciones se plantea una optimización que consiste en añadir una nueva instrucción de carga condicional que realice la asignación directamente solo si el valor es positivo. La instrucción:

```
        ld.positive r3, X(r2)    ; r3 = Mem[X+r2] si Mem[X+r2]>0
```

La nueva instrucción (*ld.positive*) tiene un CPI de 3 y en el nuevo diseño la frecuencia del procesador se reduce en un 5 %.

Se pide:

a) Calcula el tiempo de ejecución de un programa de 100 millones de instrucciones ejecutado en el sistema original. Indica el tiempo de ejecución en milisegundos.

b) Calcula el tiempo de ejecución del mismo programa pero esta vez adaptado a la optimización.

c) Calcula cual sería el coste máximo asumible de la optimización desde el punto de vista de la relación coste/prestaciones.

Solución:

a) Calcula el tiempo de ejecución de un programa de 100 millones de instrucciones ejecutado en el sistema original. Indica el tiempo de ejecución en milisegundos.
Para calcular el tiempo de ejecución aplicamos la fórmula: $T_{eje} = I \times CPI \times T$.
Del enunciado deducimos $I = 10^8$ y $T = \frac{1}{F} = \frac{1}{3GHz} = 0{,}33\ ns$.
El CPI lo obtenemos promediando: $CPI = (0{,}52 \times 1) + (0{,}48 \times 2) = 1{,}48$
Por tanto, $T_{eje} = 10^8 \times 1{,}48 \times 0{,}33\ ns = 48,84\ ms$

b) Calcula el tiempo de ejecución del mismo programa pero esta vez adaptado a la optimización.
El porcentaje de casos donde se aplica la mejora es $0{,}52 \times 0{,}2 = 0{,}104$ pues representa el 20 % de las instrucciones aritméticas (ALU).
La nueva distribución de tipos de instrucciones y frecuencias de aparición serán:

Operación	#	CPI
ALU	0.52 - 0.104 = 0.416	1
Saltos	0.24 - 0.104 = 0.136	2
Carga	0.19 - 0.104 = 0.086	2
Nueva instr	0.104	3
Almacenamiento	0.05	2

La suma de frecuencias de aparición de instrucciones es $0{,}792$. Este valor lo denominaremos factor de reducción de instrucciones.
El CPI de la optimización lo obtenemos al promediar y normalizar por el factor de reducción de instrucciones:
$CPI_{opt} = \frac{(0{,}416\times1)+(0{,}136\times2)+(0{,}086\times2)+(0{,}104\times3)+(0{,}05\times2)}{0{,}792} = 1{,}606$
El número de instrucciones que se ejecutarán lo obtenemos a partir del número de instrucciones originales y del factor de reducción de instrucciones:
$I_{opt} = I \times 0{,}792 = 79{,}2 \times 10^6$
Y el tiempo de ciclo del reloj lo obtenemos a partir del original y del factor de incremento:
$T_{opt} = T \times 1{,}05 = 0{,}3496\ ns$
Con todo esto ya podemos calcular el tiempo de ejecución:
$T_{eje\_opt} = 79{,}2 \times 10^6 \times 1{,}606 \times 0{,}3496 \times 10^{-9} = 44{,}46\ ms$

c) Calcula cual sería el coste máximo asumible de la optimización desde el punto de vista de coste/prestaciones.
El incremento en coste (IC) no debe ser mayor que el incremento en la mejora de prestaciones (S). Por tanto, el incremento en coste máximo es el que iguala IC con S:
$IC = S$
donde:
$IC = \frac{5000+x}{5000}$
$S = \frac{T_{eje}}{T_{eje\_opt}} = \frac{48{,}84\ ms}{44{,}46\ ms} = 1,098$
Despejamos x y obtenemos: $x = 490e$

□

3. **(**$2{,}5$ **puntos)** Sea el siguiente programa en ensamblador del MIPS64 que realiza unos determinados cálculos sobre vectores,

```
start:
      dadd r1,$gp,x
      dadd r4,r1,#128
      dadd r2,$gp,y
      dadd r3,$gp,z
      l.d f6,a($gp)
      l.d f5,c($gp)

loop:
      l.d f2,0(r1)
```

```
mul.d f2,f6,f2
add.d f6,f6,f5
l.d f4,0(r2)
add.d f4,f2,f4
s.d f4,0(r3)
dadd r1,r1,#8
dadd r2,r2,#8
dadd r3,r3,#8
seq r5,r4,r1
beqz r5,loop
trap #0
```

el procesador dispone de una unidad de instrucción segmentada con operadores multiciclo. Se ha ejecutado el programa y se ha obtenido el siguiente diagrama de instruciones/tiempo que muestra la evolución de la primera iteración del bucle junto con la primera instrucción de la segunda iteración,

PC	Instruccion	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
loop	l.d f2,0(r1)	IF	ID	EX	ME	WB																		
4124	mul.d f2,f6,f2		IF	id	ID	M1	M2	M3	M4	WB														
4128	add.d f6,f6,f5			if	IF	id	ID	A1	A2	A3	WB													
4132	l.d f4,0(r2)					if	IF	id	ID	EX	ME	WB												
4136	add.d f4,f2,f4							if	IF	id	ID	A1	A2	A3	WB									
4140	s.d f4,0(r3)									if	IF	id	ID	EX	ME									
4144	dadd r1,r1,8											if	IF	ID	EX	ME	WB							
4148	dadd r2,r2,8													IF	ID	EX	ME	WB						
4152	dadd r3,r3,8														IF	ID	EX	ME	WB					
4156	seq r5,r4,r1															IF	ID	EX	ME	WB				
4160	beqz r5,loop																IF	id	ID	EX	ME	WB		
4164	trap #0																	if	~~IF~~					
loop	l.d f2,0(r1)																			IF	ID	EX	ME	WB

sabemos que el procesador hace uso de cortocircuitos y que emplea *predict-not-taken* para resolver los riesgos de control. Responde a las siguientes cuestiones:

a) Identificar todos los cortocircuitos aplicados indicando el ciclo en el que se producen y las etapas involucradas.

b) En el caso de las instrucciones de salto condicional, ¿en qué etapa se calcula la condición y se modifica el PC en caso de que el salto sea efectivo? Según esto, ¿cuál será su latencia (penalización) de salto? Justifica tu respuesta.

c) En el diagrama se observa que se han insertado 6 ciclos de parada. Indicar el motivo (tipo de riesgo y pequeña descripción) por el que se han insertado en los ciclos siguientes:

- Ciclo 3:
- Ciclo 5:
- Ciclo 7:
- Ciclo 17:

d) Calcula el CPI y el tiempo de ejecución en ciclos para un número N de iteraciones grande, asumiendo que todas las iteraciones son iguales.

e) Empleando para la resolución de los riesgos de control *ciclos de parada* y suponiendo que el cálculo de la condición y la modificación del PC se realizan en la etapa EX, rehacer el diagrama desde el ciclo 15 (búsqueda de la instrucción $seq\ r5, r4, r1$) hasta la finalización de la primera instrucción de la segunda iteración.

Solución:

Responde a las siguientes cuestiones:

a) Identificar todos los cortocircuitos aplicados indicando el ciclo en el que se producen y las etapas involucradas.

- ciclo 5 (WB-M1)
- ciclo 11 (WB-A1)
- ciclo 14 (WB-ME)
- ciclo 18 (ME-ID)

b) En el caso de las instrucciones de salto condicional, ¿en qué etapa se calcula la condición y se modifica el PC en caso de que el salto sea efectivo? Según esto, ¿cuál será su latencia de salto? Justifica tu respuesta.

Observando el diagrama vemos que sólo se anula una instrucción en la $beqz$ al comprobar que el salto es efectivo, esto quiere decir que el cálculo de la condición y la modificación del PC tienen que hacerse en su etapa ID. Esto implica una latencia de salto de 1.

c) En el diagrama se observa que se han insertado 6 ciclos de parada. Indicar el motivo (tipo de riesgo y pequeña descripción) por el que se han insertado en los ciclos siguientes:

- Ciclo 3: Riesgo de datos. Para que se pueda realizar posteriormente el cortocircuito a $mul.d$
- Ciclo 5: Riesgo estructural. Escritura simultánea en banco registros flotantes.
- Ciclo 7: Riesgo estructural. Escritura simultánea en banco registros flotantes.
- Ciclo 17: Riesgo de datos. Para que se pueda realizar posteriormente el cortocircuito a la etapa ID de $beqz$.

d) Estimar el CPI y el tiempo de ejecución en ciclos para un número N de iteraciones grande, asumiendo que todas las iteraciones son iguales.

La iteración empieza en el ciclo 1 con el lanzamiento de $l.d\ f2, 0(r1)$ y acaba en el 18, un ciclo antes de que se inicie la $l.d\ f2, 0(r1)$ de la segunda iteración.

$$CPI = \frac{N \times n^o ciclos\ iteracion}{N \times n^o instrucciones\ iteracion} = \frac{N \times (18-1+1)}{N \times 11} = 1{,}64$$

$$T_e = I \times CPI\ ciclos = 11 \cdot N \times 1{,}64 = 18{,}04 \cdot N\ ciclos$$

e) Empleando para la resolución de los riesgos de control *delay slot* y suponiendo que el cálculo de la condición y la modificación del PC se realizan en la etapa EX. Rehacer el diagrama desde el ciclo 15 (búsqueda de la instrucción $seq\ r5, r4, r1$) hasta la finalización de la primera instrucción de la segunda iteración.

	...	15	16	17	18	19	20	21	22	23	24
...	...										
seq r5,r4,r1	...	IF	ID	EX	ME	WB					
beqz r5, loop	...		IF	ID	EX	ME	WB				
trap #0	...			*if*	*if*						
l.d f2,0(r1)	...					IF	ID	EX	ME	WB	
	...										

□

4. **(**2,5 **puntos)** El siguiente bucle en ensamblador del MIPS64 contabiliza en el registro R5 el número de elementos del vector A que valen cero. La instrucción `bnez r2, final` realiza el salto si el elemento leído es distinto de cero.

```
 loop: ld r2, A(r1)
       daddi r1, r1, 8
       bnez r2, final
       daddi r5, r5, 1
final: dsub r4, r3, r1
       beqz r4,loop
```

Se pide:

a) Si los elementos del vector A son $\vec{A} = (0,1,1,1,0,1,1,1,0,1)$ y el procesador implementa un predictor de 1 bit inicialmente a cero, indica la precisión del predictor únicamente para la instrucción `bnez r2, final` que salta cuando el elemento es distinto de cero.

Deben rellenarse todos los campos de la tabla de la hoja de respuestas, indicando para cada ejecución de la instrucción bnez el valor del predictor, la predicción realizada (0 no salta, 1 salta), el resultado de la condición de salto (0 no salta, 1 salta), y si hay acierto o fallo (0 fallo, 1 acierto). A modo de ejemplo, se ofrece el resultado de la primera ejecución del salto `bnez r2, final` en la hoja de respuestas.

b) Obtén la tabla para la instrucción `bnez r2, final` para un predictor de 2 bits (contador) con saturación inicialmente a cero. El valor del estado debe indicarse en binario.

c) Suponga ahora que el predictor anterior de 2 bits se implementa como predictor de una BTB accedida en IF, y que el procesador realiza en ID el cálculo de la dirección, condición y escritura del PC, ¿cuál hubiese sido la penalización de salto acumulada (suma de todas las penalizaciones) cuantificada en ciclos en el supuesto caso que se hubiese fallado en 7 de 10 ejecuciones de la instrucción `bnez r2, final` ? Razone la respuesta.

d) ¿Cuál hubiese sido la penalización en ciclos para la instrucción `bnez r2, final` si el calculo de la dirección y condición hubiese sido en ID pero la escritura del PC se hubiese realizado en la etapa EX? Razone la respuesta.

Solución:

a) Predictor de 1 bit

Ejecución bnez	1	2	3	4	5	6	7	8	9	10
Valor estado (contador) en IF	0	0	1	1	1	0	1	1	1	0
Predicción	0	0	1	1	1	0	1	1	1	0
Condición	0	1	1	1	0	1	1	1	0	1
Acierto	1	0	1	1	0	0	1	1	0	0

$Precision = 5/10 = 0{,}5$, es decir, del $50\,\%$.

b) Predictor de 2 bits con saturación.

Ejecución bnez	1	2	3	4	5	6	7	8	9	10
Valor estado (contador) en IF	00	00	01	10	11	10	11	11	11	10
Predicción	0	0	0	1	1	1	1	1	1	1
Condición	0	1	1	1	0	1	1	1	0	1
Acierto	1	0	0	1	0	1	1	1	0	1

$Precision = 6/10 = 0{,}6$, es decir, del $60\,\%$.

c) Penalización acumulada: 7 fallos x 1 ciclo/fallo = 7 ciclos. La penalización es de 1 ciclo por fallo porque el PC se escribe en ID.

d) Penalización acumulada: 7 fallos x 2 ciclo/fallo = 14 ciclos. La penalización es de 2 ciclos por fallo porque el PC se escribe en EX.

□