

## Arquitectura de los Computadores. Primera convocatoria 2015

1. **(2 puntos)** La extensión del repertorio de un microprocesador incorporando instrucciones SSE y el correspondiente hardware de procesamiento SSE permite acelerar los tiempos de cálculo, en lo que a tareas multimedia en punto flotante (TMPF) se refiere, en un factor de 6. Utilizando como benchmark para análisis del rendimiento en punto flotante el programa alvinn de SPEC, que realiza tanto tarea multimedia en punto flotante (TMPF) como no multimedia (TNMPF), se observó que el tiempo de ejecución del programa era de 15 segundos si se compilaba utilizando SSE y de 24 segundos si se compilaba sin utilizar SSE.
  - a) Calcula la aceleración global y la fracción mejorada. **(0,5 puntos)**
  - b) Calcula el tiempo de ejecución que el programa compilado sin SSE consumen en realizar tareas multimedia en punto flotante (TMPF). **(0,5 puntos)**
  - c) Calcula el tiempo de ejecución que el programa compilado con SSE consumen en realizar tareas multimedia en punto flotante (TMPF). **(0,5 puntos)**
  - d) Calcula el tiempo de ejecución que el programa consumen en realizar tarea no multimedia en punto flotante (TNMPF). **(0,5 puntos)**
2. **(1,5 puntos)** En el diseño de una CPU se está considerando la conveniencia de remplazar las instrucciones de salto condicional por una versión que incluya la comparación en el salto. En la alternativa de diseño de la CPU A se han utilizado códigos de condición que son actualizados por la instrucción de comparación CMP. En la alternativa de la CPU B la arquitectura es la misma salvando que las instrucciones de salto realizan la comparación SLTCMP, eliminando las correspondientes instrucciones de comparación y reduciendo el RI. La incorporación de la instrucción SLTCMP incrementa el clk en la CPU B en un 5%. El CPI y frecuencia por tipo de instrucción se detalla en la tabla para la CPU A.

Tipo de instrucción	CPI CPUA	%	CPI CPUB	%
Load Store	2	35%	2	¿?%
ALU	1	30%	1	¿?%
CMP	1	20%	1	¿?%
SLT (Salto)	2	15%	No existe	
SLTCMP (Compara y salta)	No existe		2	¿?%

- a) Calcula la aceleración entre las dos opciones de diseño. **(1 punto)**
  - b) Calcula el % de incremento del ciclo de reloj a partir del cual el cambio deja de ser rentable. **(0,5 puntos)**
3. **(2,5 puntos)** En el proceso de diseño de un computador se ha optado por una filosofía RISC. Indica que decisiones tomarías, valorando las diferentes alternativas, en las siguientes componentes de la arquitectura:
  - a) Número de operandos que se pueden direccionar en memoria en instrucciones ALU **(0,5 puntos)**
  - b) Modos de direccionamiento de operandos **(0,5 puntos)**
  - c) Tipos de instrucciones en el repertorio **(0,5 puntos)**
  - d) Codificación de los modos de direccionamiento. Pon un ejemplo del formato de codificación elegido. **(1 punto)**
4. **(3 puntos)** Preguntas sobre segmentación
  - a) Considerad un procesador no segmentado con una ruta de datos de 5 etapas de ejecución que funciona a 2GHz, en el que las operaciones ALU y salto requieren cuatro ciclos de reloj y las de memoria cinco. Suponer que las frecuencias relativas para estas operaciones son 45%, 25% y 30% respectivamente. Se quiere segmentar la máquina con 5 etapas y debido al sesgo del reloj y a los registros de segmentación se alarga el periodo de reloj en un 10%. El procesador utiliza una cache unificada para datos e instrucciones con un único puerto para el acceso a la memoria lo que provoca un riesgo estructural entre las etapas IF y MEM y por consiguiente una parada de 1 ciclo de reloj. Suponed que el CPI ideal del procesador segmentado ignorando el riesgo estructural es 1. ¿Cuál es la ganancia que se puede conseguir con la segmentación? **(0.4 puntos)**

Nombre: \_\_\_\_\_

- b) Suponed ahora que además se consideran las detenciones por dependencia de datos de 1 ciclo de reloj que representan el 20% de todas las instrucciones ejecutadas y las detenciones por riesgo de control de 2 ciclos y que representan el 5% de todas las instrucciones ejecutadas, ¿Cuál es ahora el nuevo CPI?. ¿En qué porcentaje se reduce la ganancia al considerar esta nueva situación? **(0,4 puntos)**

Suponed que el siguiente código MIPS se ejecuta en la máquina segmentada:

```
addi $3, $0, 100
add $4, $0, $0
Loop: lw $5, 0($1)
      add $4, $4, $5
      lw $6, 0($2)
      sub $4, $4, $6
      addi $1, $1, 4
      addi $2, $2, 4
      addi $3, $3, -1
      bne $3, $0, Loop
```

- c) Muestra el diagrama de temporización para una iteración del bucle suponiendo que no hay forwarding. Completa para ello la siguiente tabla y muestra todos los ciclos de paradas. Suponed que los saltos paran la segmentación solo durante un ciclo de reloj. **(1 punto)**

Instrucción	Ciclos de reloj																										
	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	
addi \$3, \$0, 100																											
add \$4, \$0, \$0																											
lw \$5, 0(\$1)																											
add \$4, \$4, \$5																											
lw \$6, 0(\$2)																											
sub \$4, \$4, \$6																											
addi \$1, \$1, 4																											
addi \$2, \$2, 4																											
addi \$3, \$3, -1																											
bne \$3, \$0, Loop																											
lw \$5, 0(\$1)																											
add \$4, \$4, \$5																											

- d) De acuerdo con el diagrama de temporización del apartado anterior, calcula el número de ciclos de reloj y el CPI medio para ejecutar el código anterior con todas las iteraciones del bucle. **(0,4 puntos)**
- e) Reordena las instrucciones del código anterior y rellena el delay slot del salto que evitar el máximo posible de las paradas. Escribe el código resultante. **(0,4 puntos)**
- f) Calcula el número de ciclos y CPI medio para ejecutar todas las iteraciones del bucle en el nuevo código. ¿Cuál es la ganancia obtenida? **(0,4 puntos)**
5. **(1 punto)** Suponer que a un sistema se le introduce una memoria caché. La adición de esta caché conlleva una penalización de fallos de 100 ciclos de reloj. Considerar que todas las instrucciones normalmente emplean 1.0 ciclos de reloj (ignorando las detenciones de memoria), que la frecuencia de fallos es del 4% y que hay una media de 1.5 referencias a memoria por instrucción.

$$T_{CPU} = NI * \left( CPI_{ejec} + \frac{NM}{NI} * FF * PF \right) * T_{reloj}$$

NM: Número de accesos a memoria  
 FF: Frecuencia de fallos  
 PF: Penalización del fallo

- a) Comenta y razona si existe alguna ventaja al incorporar caché en un sistema con memoria convencional **(0,5 puntos)**
- b) ¿Cuál es el impacto en el rendimiento cuando se incluye el comportamiento de esta caché respecto a una caché ideal (sin fallos)? **(0,25 puntos)**
- c) ¿Cuál es el impacto en el rendimiento cuando se incluye el comportamiento de esta caché respecto al sistema sin caché? **(0,25 puntos)**