

Nombre: _____

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
Lw R10, 0(R20)																								
Lw R11, 0(R20)																								
subu R10, R10, R11																								
addiu R20, R20, #4																								
sw R10, 200(R20)																								
subu R5, R23, R20																								
bnez R5, LOOP																								
lw R10, 0(R20)																								

- b) (0,4 puntos) Obtén el número total de ciclos necesarios para completar 2 iteraciones del bucle y calcula el CPI medio obtenido en completar esta ejecución. Muestra cómo has obtenido los valores.
- c) (0,5 puntos) Supongamos ahora que se implementa la técnica de *forwarding* en la máquina anterior. Con esta nueva condición rellenar el diagrama de temporización siguiente:

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
Lw R10, 0(R20)																								
Lw R11, 0(R20)																								
subu R10, R10, R11																								
addiu R20, R20, #4																								
sw R10, 200(R20)																								
subu R5, R23, R20																								
bnez R5, LOOP																								
lw R10, 0(R20)																								

- d) (0,4 puntos) Obtén el número total de ciclos necesarios para completar 2 iteraciones del bucle. ¿Cuál es ahora el CPI medio obtenido? ¿Cuál es la ganancia obtenida con esta alternativa?
- e) (0,4 puntos) Supongamos ahora que el procesador implementa saltos retardados (*delay slot*). ¿Podrías reorganizar el código para evitar cualquier parada, aprovechar el *delay slot* y conseguir mayor ganancia? Razona la respuesta. Muestra en cualquier caso como rellenarías el *delay slot*.
- f) (0,3 puntos) ¿Cuál sería en este caso el número de ciclos que tardaría en ejecutarse 2 iteraciones del bucle? ¿Cuál sería la ganancia conseguida para este código respecto a la versión original si la hubiere?
- g) (0,5 puntos) Se está estudiando dos posibilidades para mejorar la segmentación en los saltos. La primera posibilidad es utilizar saltos retardados, en este caso en una ejecución típica del 25% de las instrucciones de salto se consigue rellenar el 50% de los *delay slot* con instrucciones útiles. Calcula la ganancia que se consigue al implementar esta técnica. La segunda posibilidad es suponer que los saltos son no efectivos. En este caso en una ejecución típica el 25% de los saltos en un bucle son no efectivos y el 80% de las instrucciones son saltos dentro de un bucle. Calcula la ganancia que se consigue al implementar esta técnica. Por simplicidad suponer que las instrucciones que no son de salto su CPI es 1. ¿Qué optimización es la mejor opción?
5. (1,5 puntos) La arquitectura Zen de AMD ha mejorado en muchos aspectos la anterior Bulldozer de la misma marca. Se ha dejado atrás el diseño *write-thru* (escritura directa) en favor del esquema *write-back* (postescritura). En concreto, el Ryzen 7 1800X de AMD tiene una caché de datos de nivel 1 (L1D) de 32KB y 8 vías.
- a) (0,5 puntos) Indica las ventajas de un diseño *write-back* como el de la arquitectura Zen frente a un diseño *write-thru* como el de la arquitectura Bulldozer.
- El punto débil de la arquitectura del Ryzen 7 1800X parece estar en la Caché de nivel 3 (L3). Esta memoria presenta unos retardos muy elevados que son incluso mayores a los de arquitecturas anteriores de la misma marca. La L3 de esta arquitectura tiene una capacidad de 16 MB y tiene una asociatividad de 16 vías.
- b) (1 punto) Suponiendo un tamaño de bloque de 4 KB y un espacio de direccionamiento de la memoria principal de 32 bits se pide para la caché L3 del Ryzen 7 1800X:
- (0,2 puntos) Número de bits de la dirección que especifica una palabra (byte) dentro de un bloque
 - (0,2 puntos) Número de líneas de la caché
 - (0,2 puntos) Número de conjuntos de la caché
 - (0,4 puntos) Indica en la siguiente tabla, para cada dirección de memoria que aparece, la etiqueta asociada del bloque y el conjunto en el que se ubicaría en la caché

Dirección (Hexadecimal)	Etiqueta de bloque (Hexadecimal)	Nº Conjunto (Hexadecimal)
FDAC230C		
8FF37124		

Nombre: _____

Arquitectura de los Computadores. Primera convocatoria 2017

1. (2 puntos) La empresa CVApps (Computer Vision Applications) está diseñando un sistema de guiado de vehículos mediante metodologías basadas en "deep learning", cuyos requerimientos de rendimiento vienen impuestos por las altas velocidades de los vehículos. Se ha hecho un estudio de las partes del código del sistema de guiado que son paralelizables y ejecutables en GPUs. El código paralelizable se ha ejecutado en modelos de GPU con diferentes prestaciones:

	No Paralelizable CPU	Paralelizable
CPU	20s	50s
GTX 1080 Ti	20s	20s
Radeon Pro Duo	20s	4s
GTX TITAN X	20s	1,67s
GTX 1060, 3 GB	20s	0,33 s

- a) (1 punto) Calcula la aceleración global y la aceleración mejorada respecto a la opción CPU para cada una de las GPUs
- b) (1 punto) El equipo de diseño hardware ha comprobado que es capaz de mejorar la "GTX 1080 Ti" reduciendo a la mitad los tiempos de GPU. Se desea saber si el equipo de desarrollo software puede incrementar el porcentaje de paralelización (fracción mejorada) para mejorar el rendimiento manteniendo la versión actual de la "GTX 1080 Ti". ¿Qué incremento en el porcentaje de paralelización se necesitará para obtener la misma ganancia de rendimiento que el equipo hardware?
2. (1,5 puntos) La misma empresa CVApps (Computer Vision Applications) está diseñando un procesador especializado para un sistema de visualización realista, donde es necesario un alto rendimiento en aplicaciones de generación de gráficos. En este procesador y ejecutando estas aplicaciones la mezcla de instrucciones y CPIs son:

Instrucción	Frecuencia	CPI
ALU	40%	1
LOAD	11%	2
STORE	29%	2
SALTO	15%	1
IMP	5%	4

La máquina, debe realizar siempre instrucciones STORE para almacenar los datos que utiliza la instrucción IMP para imprimir en pantalla. La empresa está pensando en realizar una modificación para que IMP cargue directamente los datos a imprimir en pantalla, sin necesidad de realizar antes una STORE. Supongamos que este repertorio extendido de instrucciones incrementa en 1 el número de ciclos de reloj para la instrucción IMP, pero sin afectar a la duración del ciclo de reloj.

- a) Calcula la aceleración de la versión supuestamente mejorada respecto a la anterior
3. (2 puntos) La compañía IBM quiere diseñar un nuevo procesador para los robots Pepper de Aldebaran. La compañía parte de cero en el diseño del procesador y desea comenzar por el diseño del repertorio de instrucciones sabiendo que: el coste total del procesador debe ser bajo, los programas que se ejecuten en el procesador se compilarán con el compilador GCC y tienen pensado una organización segmentada para el computador. A partir del estudio inicial de sus ingenieros, ya conocemos ciertas restricciones:
- Las instrucciones que generará el compilador para los programas de Pepper serán aritméticos/lógicas, transferencias de datos, control, sistema, de punto flotante y multimedia (para audio y video). En total dispondrá de 110 instrucciones diferentes. De las cuales, 60 son subtipos de aritméticas y lógicas
 - La memoria es de 32 bits para el bus de datos y de direcciones
 - El robot trabajará con enteros de 32 bits y puntos flotantes de doble precisión (64 bits)
- Indica cuál o cuáles serían los formatos de instrucción de manera justificada que debería diseñar IBM para el robot Pepper. El formato debería indicar claramente los campos y el tamaño de cada uno. La elección del campo y el tamaño del mismo tienen que estar justificados. Es decir, ¿por qué decides incluir un campo de determinado tipo y de tamaño n bits? En otro caso no se puntuará el ejercicio.

4. (3 puntos) Considera que el segmento de código que aparece a continuación se ejecuta sobre una máquina segmentada MIPS estándar de 5 etapas. Supongamos que el valor inicial del registro R23 es mucho mayor que el valor inicial del registro R20 y que todas las referencias de memoria se encuentran en las cachés y TLBs.

```

LOOP: lw R10, 0(R20)
      lw R11, 100(R20)
      subu R10, R10, R11
      addiu R20, R20, 4
      sw R10, 200(R20)
      subu R5, R23, R20
      bnez R5, LOOP
    
```

- a) (0,5 puntos) Dibuja el diagrama de temporización para la ejecución de la secuencia de instrucciones que se muestran en la tabla. Considera que el salto se resuelve en la etapa ID y que además la técnica del adelantamiento o forwarding no está implementada en esta máquina.