

Instituto Tecnológico de Costa Rica Escuela de Ingeniería en Computadores Programa de Licenciatura en Ingeniería en Computadores Curso: CE-4302 Arquitectura de Computadores II Profesores: Luis Barboza Artavia Ronald García Fernández Semestre: II 2024	I Examen Parcial Fecha: 09/09/24 Puntos totales: 100 Puntos obtenidos: _____
---	---

Nombre: _____ Carné: _____

Grupo: _____

Instrucciones Generales

- 1- Trabaje individualmente.
- 2- Utilice cuaderno de examen u hojas blancas numeradas para resolver la prueba.
- 3- Escriba de manera legible y ordenada.
- 4- Sea lo más detallado posible en sus respuestas (cuando se le pide) no deje nada abierto a interpretaciones.
- 5- Utilice bolígrafo para resolver la prueba. **No** se aceptarán reclamos sobre respuestas con lápiz
- 6- El fraude se castiga según estipula el reglamento de enseñanza-aprendizaje del TEC.
- 7- El tiempo para resolver la prueba es de 2 horas.
- 8- No se permite el uso de celulares o algún otro tipo de dispositivo móvil.
- 9- Todo código, programa, pseudocódigo debe poseer comentarios. En caso contrario se asignará un puntaje igual a cero.

I- Parte Única. Desarrollo [100 puntos]

Resuelva cada uno de los siguientes problemas, recuerde indicar todos los pasos que lo llevaron a la solución, no es permitido el uso de materiales de apoyo más que la página acordada en clase la cual tiene que ser entregada junto al resto del examen.

- 1- En la figura 1 se muestra un *compute kernel*, en la figuras 2 *roof-line model* de implementación una arquitectura X

```

1- #define N 10000
2- double X[N] = {...}; //random data
3- double Y[N] = {...}; //random data
4- double Z[N] = {...}; //random data
5- double a    = 5.0;

6- void foo() {
7-     for (int i = 0; i < N; ++i){
8-         Y[i] += a*X[i];
9-         Z[i] *= X[i];
10-        const float x = (float)i; //cast from int to float
11-        X[i] = Y[i];
12-    }
13-}

```

Figura 1. *Compute Kernel*

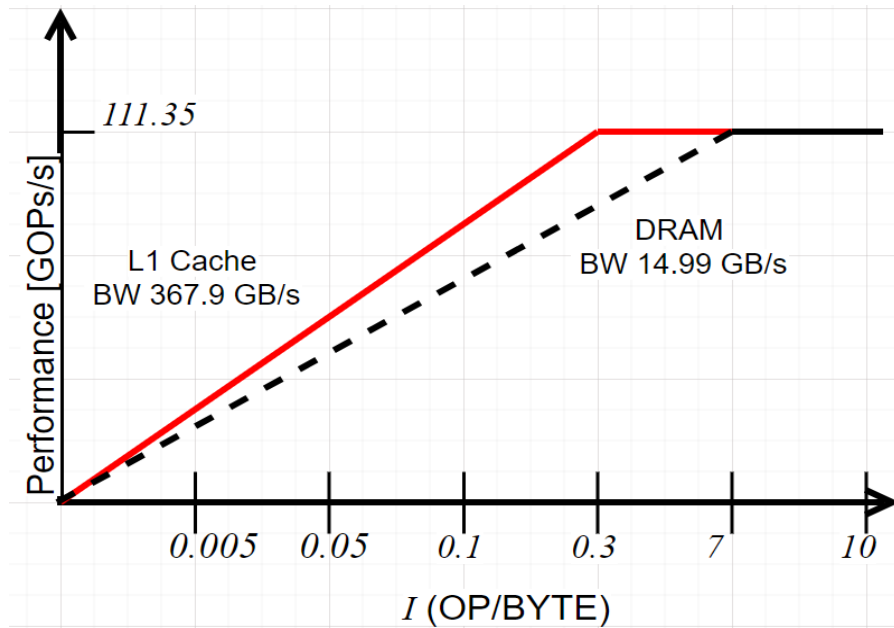


Figura 2. Roof-line model *X*

Respecto a las figuras 1 y 2 se le solicita:

- Calcule la intensidad aritmética (IA) del *kernel* incluya todos los pasos que le llevaron a su respuesta. (10 puntos)
- Identifique la zonas y parámetros en la figura 2 (5 puntos)
- En la figura 2 se muestran 2 techos para *bandwidth L1 Cache* y *DRAM* explique qué significan cada uno de los techos *L1 Cache* o *DRAM* (5 puntos)
- Ubique el valor IA del punto 'a' en el *roofline model* e indique en que zonas encuentra el *kernel* para cada uno de los techos mostrados, y explique bajo que circunstancias el *kernel* puede operar en uno u otro techo. (5 puntos)

- En la figura 3 se muestra un extracto de un código basado en RV64V con LANE = 1, clk = 1GHz y n = 64

```

1- vle64.v    v3, 0(a5) #load vector v3
2- vle64.v    v1, 0(a4) #load vector v1
3- vfmaccc.vv v1, v3, v4 #FP vector-vector multiply-accumulate (v1)
4- vse64.v    v1, 0(a4) #store vector v1
5- addi       a4, a4, 16 #add integer
6- vle64.v    v2, 0(a3) #load vector v2
7- vfmul.vv   v2, v2, v3 #FP vector-vector multiply
8- vse64.v    v2, 0(a3) #store vector v2
9- vse64.v    v1, 0(a5) #store vector v1

```

Figura 3. Extracto de código basado en RV64V

Respecto al fragmento de la figura 5 se le pide:

- a- Identifique los **convoys** encontrados. (5 puntos)
 - b- Calcule el T_e . (5 puntos)
 - c- Calcule el **Total_FLOPS** (5 puntos)
 - d- Respecto al código anterior cómo cambian los parámetros de *start-up time* al cambiar el tamaño de **n**? ¿Qué consideraciones de SW son necesarias para RV64V para cambiar **n**? ¿Qué implica en términos de HW un aumento de **n**? (10 puntos)
- 3- En criptografía, un cifrado simple puede ser utilizando la función lógica XOR. Así, una cadena de texto se puede cifrar aplicando este operador sobre cada carácter usando una clave. A continuación, se muestra un ejemplo de cómo se relaciona el texto con la clave en caso de que fuera 'ab'. Para aplicar el cifrado, se aplicaría carácter por carácter la operación xor.

Texto	Hola
Llave	abab

De acuerdo con esta información, se solicita lo siguiente:

- a- Implemente el pseudocódigo que realice la operación de manera 'serializada' (sin vectorizar) para obtener un texto cifrado de longitud **M bytes** y una llave de 2 bytes y otra llave de 4 bytes. (5 puntos)
- b- Utilizando un diagrama, diseñe cómo se comporta el algoritmo de cifrado de forma vectorizada. Para esto, parta de que el texto se cargará en vectores de 128 bits y las llaves pueden ser de 2 bytes o 4 bytes. (13 puntos)
- c- Basándose en el diagrama del punto b, diseñe los intrinsics/instrucciones necesarias que permitan realizar el cifrado de un texto utilizando vectores de 128 bits. (10 puntos)
- d- Empleando técnicas de vectorización y los intrinsics del punto c implemente el pseudocódigo que permite cifrar un texto de longitud M mediante llaves de 2 o 4 bytes. Considere todos los posibles casos de longitud de M y explique cómo manejarlos de manera "segura". (12 puntos)
- e- Realice un análisis comparativo empleando la teoría vista en clase entre los algoritmos de los puntos a y d incluyendo aspectos de HW y SW en su respuesta. (10 puntos)