

<b>Instituto Tecnológico de Costa Rica</b> <b>Escuela de Ingeniería en Computadores</b> <b>Programa de Licenciatura en Ingeniería en Computadores</b> <b>Curso: CE-4302 Arquitectura de Computadores II</b> <b>Profesores:</b> Luis Barboza Artavia Ronald García Fernández <b>Semestre: I 2024</b>	<b>I Examen Parcial</b> <b>Fecha: 18/03/24</b> <b>Puntos totales: 100</b> <b>Puntos obtenidos: _____</b>
--	---

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 las figuras 2 y 3 se muestran los *roof-line models* de implementaciones de diferentes arquitecturas **X** y **Y** respectivamente.

```

1  static const int msize = 2048;
2
3  int main() {
4
5      auto a = new int[msize][msize];
6      auto b = new int[msize][msize];
7      auto c = new int[msize][msize];
8      auto d = new int[msize][msize];
9      auto y = new float[msize][msize];
10
11
12      int i, j, k;
13      for (i = 0; i < msize; i++) {
14          for (j = 0; j < msize; j++) {
15              for (k = 0; k < msize; k++) {
16                  c[i][j] = (a[i][k] * b[k][j]);
17                  d[i][j] = (a[i][k] / b[k][j]);
18                  y[i][j] = 0.1f * static_cast<float>(a[i][k]) - static_cast<float>(b[k][j]);
19              }
20          }
21      }
22      return 0;
23  }

```

Figura 1. Compute Kernel

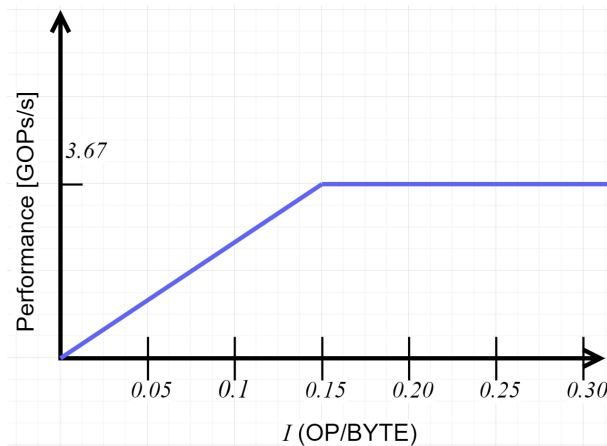


Figura 2. Roof-line model for X

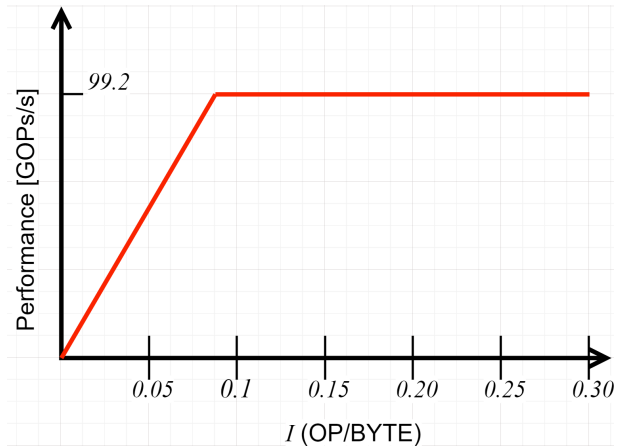


Figura 3. Roof-line model for Y

Respecto a las figuras 1, 2 y 3 se le solicita:

- Se desea definir una intensidad aritmética de enteros (int)  $I_{INT}$  calcule dicho valor. (5 puntos)
- Se desea definir una intensidad ( $I_{TOT}$ ) que combine los tipos de datos float e int bajo este requisito, explique cómo se calcula la nueva métrica, obtenga el valor de  $I_{TOT}$ , realice un análisis comparativo respecto al valor calculado en a. (10 puntos)
- Ubique en los *roofline models* de X y Y los valores calculados en los puntos a y b, e indique en qué zona se encuentran para cada implementación, justifique mediante criterios de diseño cuál de las 2 implementaciones (X ó Y) preferiría emplear para el *kernel*. (10 puntos)
- Indique mediante cálculos cómo es posible obtener una mejora de 25% en  $I_{TOT}$ , justifique que técnicas en concreto permiten el aumento en  $I_{TOT}$  (5 puntos)

- 2- Una operación típica al manipular archivos de texto de gran tamaño es el conteo de ocurrencias de una palabra o un carácter (1 byte, UTF-8). En la figura 4 se muestra como ejemplo para una cadena de 16 caracteres **str**. Se puede realizar la búsqueda del carácter ‘;’ mediante una comparación entre cada elemento de **str** y el valor de ‘;’ (**char\_x**), obteniendo una cadena de salida **out\_str** con 2 valores posibles **0x00** y **0xFF**.

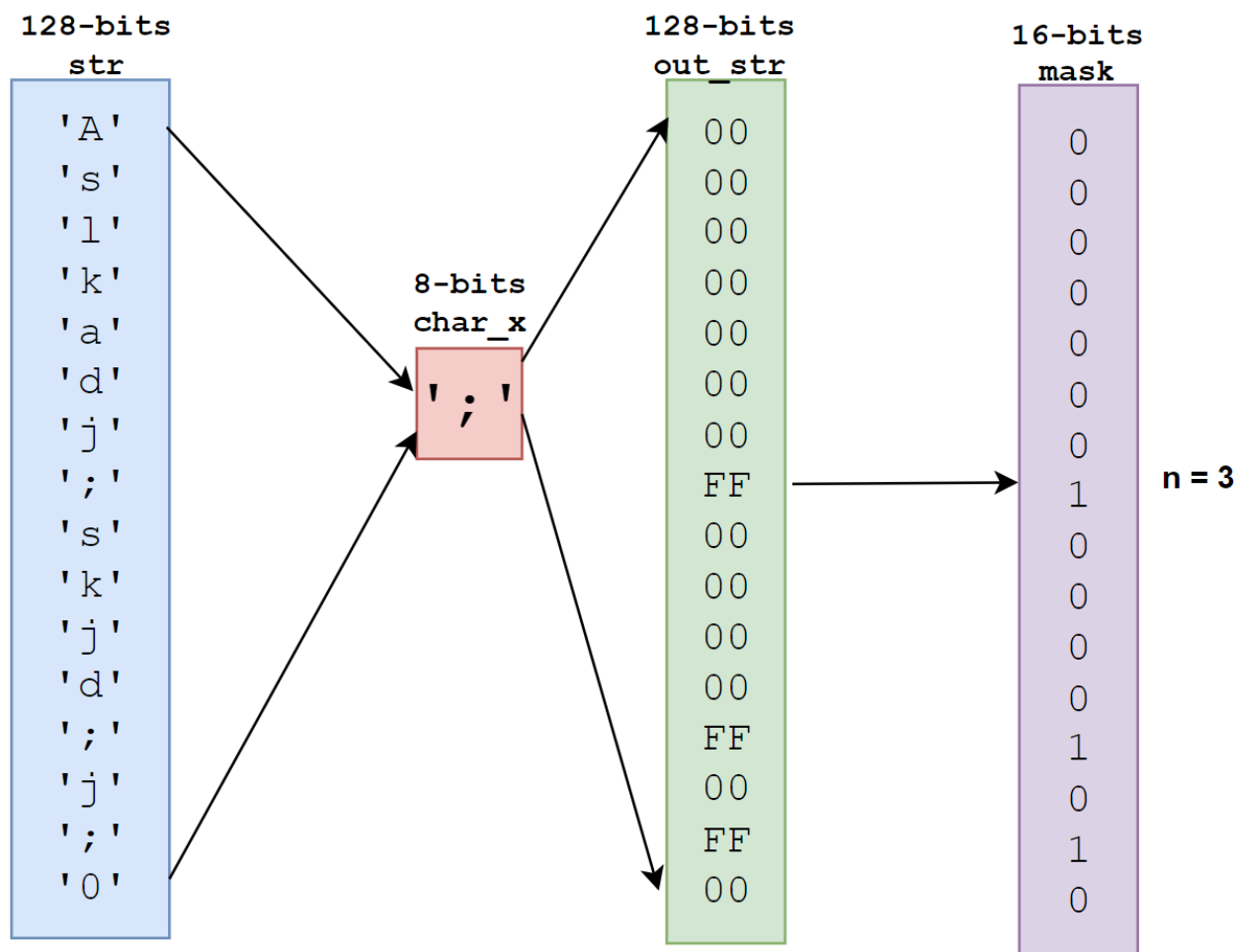


Figura 4. Ejemplo de búsqueda/conteo del carácter ‘;’

De esta forma se puede interpretar el valor **0xFF** como un **1 lógico** y **0x00** como un **0 lógico**, y así reducir **out\_str** a un vector **mask de 16 bits** sobre el cual sólo es necesario contar el número bits con valor ‘b1’ (**n = 3** para este ejemplo).

Respecto a la aplicación anterior y suponiendo que dispone de 2 funciones para obtener el vector mask en una variable (**mask = get\_mask(simd\_vector)**) y contar el número de bits en ‘1’ de una variable (**bits\_set = count\_ones(mask)**), se le solicita lo siguiente:

- a- Implemente el pseudocódigo que realice la operación de manera ‘serializada’ (sin vectorizar, ni usar mask, etc) para obtener la cantidad de ocurrencias de un carácter **char\_x** en una cadena **str** de longitud **M**. (5 puntos)
  - b- Basándose en la figura 4, diseñe un *intrinsic*/instrucción que permite replicar el valor de **char\_x** en un vector *SIMD* de 128 bits **vect\_x** (5 puntos)
  - c- Basándose en la figura 4, diseñe un *intrinsic*/instrucción que permite comparar dos vectores *SIMD* de **128 bits vect\_x** y **vect\_y** (5 puntos)
  - d- Empleando técnicas de vectorización y los *intrinsics* de los puntos **b** y **c** implemente el pseudocódigo que permite encontrar la cantidad de ocurrencias de un carácter **char\_x** en una cadena **str** de longitud **M**, considere todos los posibles casos de longitud de **M** y explique cómo manejarlos de manera “segura”. (25 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)
- 3- En la figura 5 se muestra un extracto de un código basado en RV64V con LANE = 1, clk = 2.5GHz y n = 64

```

1  vfmv    v3, fa5      #Replicate a scalar fa5 into v3
2  vld     v1, 0(a4)    #Vector load v3 from addr
3  vld     v2, 0(a5)    #Vector load v3 from addr
4  vfmadd  v1, v3, v2    #Vector-Vector multiply and add
5  vadd    v30, v1, v0   #Vector-vector add
6  vst     v30, 0(a2)    #Vector store v30 at addr
7  vst     v1, 0(a5)     #Vector store v1 at addr

```

Figura 5. 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- Discuta qué implicaciones tiene el cambiar el LANE > 1 a nivel de HW y SW (5 puntos)