

<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:</b> CE-4302 Arquitectura de Computadores II <b>Profesores:</b> Luis Barboza Artavia Ronald García Fernández <b>Semestre:</b> I 2024	<b>II Examen Parcial</b> <b>Fecha:</b> 27/05/24 <b>Puntos totales:</b> 100 <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, seudocódigo 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 una variante de la topología **MP** conocida como **mesh** en la cual la comunicación directa entre *PEs* se puede dar únicamente entre vecinos (según lo indica las flechas).

En esta configuración mediante *benchmarks* se obtuvieron las siguientes métricas:

- El costo de acceso remoto  $C(n) = 3n + 1$  donde  $n$  es el número de *PEs* que hay que atravesar para comunicar dos *PEs* (incluyendo el destino)
- El Remote Rate Request es de 10%
- CPI base cuando no hay *cache misses* es 0,2 ciclos/instrucción
- La penalización al usar más de un *PE* es aproximada mediante la función:

$P(m, i) = \frac{2^m}{i}$  donde  $m$  es la cantidad de *PEs* con 3 o más vecinos,  $i$  el número de *PEs* en el mesh.

Respecto a este sistema se le solicita:

- a- Determine las rutas posibles para comunicar *PE0* y *PE5* minimizando la cantidad de *PEs* necesarios. (5 puntos)
- b- Si se tiene que *PE1* y *PE4* únicamente están disponibles el 10% del tiempo calcule el nuevo CPI esperado en el caso de comunicación *PE0-PE5*, discuta sus resultados. (10 puntos)
- c- Para el caso que no hay ninguna penalización en la comunicación entre *PEs* cual debe ser la fracción no paralelizable que permite obtener una mejora de 6,25 con todos los *PEs* disponibles ( $N=9$ ). (5 puntos)
- d- Suponiendo que  $N=9$  es únicamente posible durante el 90% del tiempo de ejecución, cual es la fracción de tiempo necesaria en la que se tienen 7 *PEs* activos, tal que permite una mejora de 5,5. Discuta sus resultados. (10 puntos)
- e- Usando la función de penalización como cambian los resultados de *speedup* esperado si se tiene una fracción de tiempo no paralelizable de 15% y todos los *PEs* están disponibles, explique qué sucede si se incrementa el *mesh* a 4x4, que implicaciones tiene desde el punto de vista de SW y HW. (10 puntos)

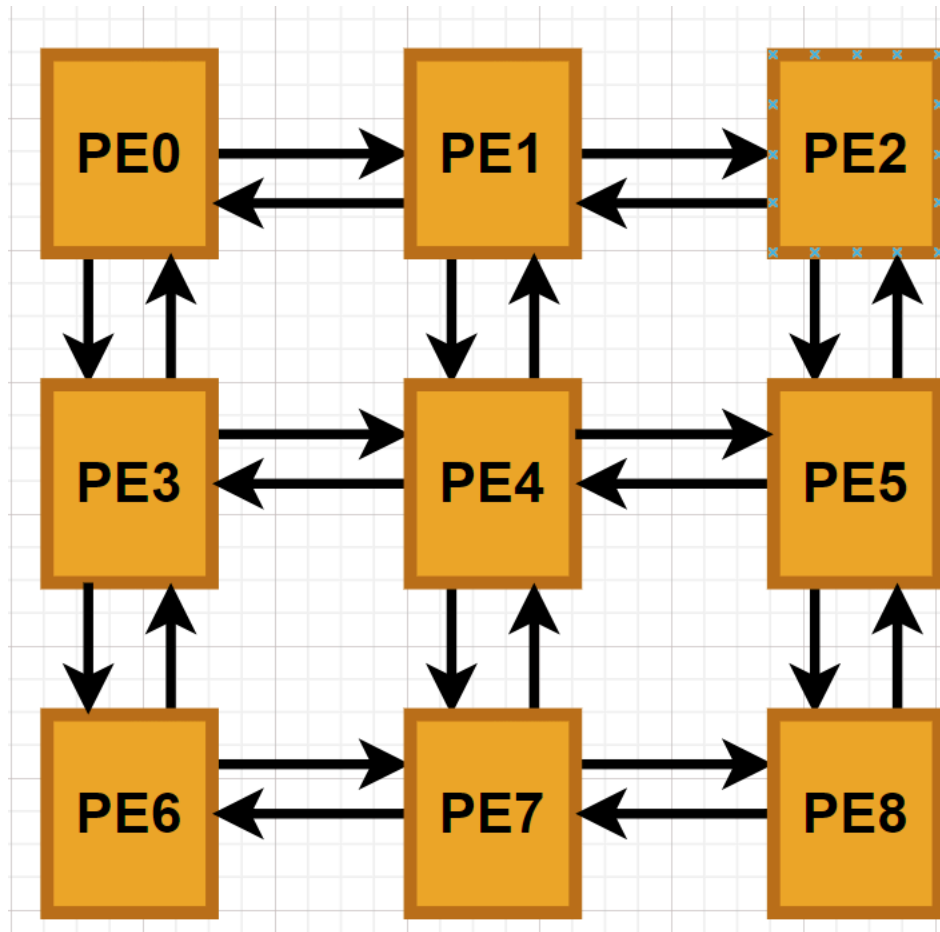


Figura 1. Arreglo de PEs en variante *mesh*.

- 2- Explique en qué consiste la coherencia de cache, y cómo permite mejorar el desempeño para la migración de tareas o cambio de *threads*? (10 puntos)
- 3- Sobre la consistencia de memoria explique por qué es necesario un *memory model* en sistemas *multicore*, y en qué consiste dicho modelo? (10 puntos)

- 4- Una de las formas de aproximar  $\pi$  es mediante el uso de la serie de Leibniz de la siguiente forma:

$$\pi = 4 \sum_{k \geq 0} (-1)^k \frac{1}{2k+1}$$

Respecto a la serie anterior se le solicita:

- a- Implemente un algoritmo “serial” que permita aproximar  $\pi$  con un parámetro  $N > 10000$  (5 puntos)
- b- Implemente un algoritmo que permita aproximar  $\pi$  mediante el uso de SMT en un sistema que soporta por lo menos 4 *HW threads* con un parámetro  $N > 10000$  (15 puntos)
- c- Discuta las diferencias entre los algoritmos implementados en el punto a y b, respecto al uso de memoria caché, protocolos de coherencia, manejo de *threads*, y consistencia de memoria. (10 puntos)
- d- Qué tipo de sistema MP elegiría y por qué para implementar la solución presentada en b, considerando que el sistema realiza más tareas, es decir no solo la aproximación de  $\pi$  (5 puntos)
- e- Respecto a la solución propuesta en el punto d qué papel juega el Sistema Operativo en dicho sistema? (5 puntos)