

Ampliación de Arquitecturas de Computadoras

Examen de febrero (7/2/03) Teoría

Durante el examen no se permiten apuntes ni nada encima de la mesa. El tiempo para la realización de esta parte es de 1h y 10 minutos.

1. Supongamos que un procesador vectorial tiene una memoria entrelazada de orden bajo con 3 bits para la selección del módulo. Suponiendo que el tiempo de acceso a un módulo de memoria es de 100 ns., contesta a las siguientes preguntas: ¿Cuántos ciclos de reloj son necesarios para acceder a 10 elementos consecutivos de memoria, sabiendo que el periodo del reloj son 50 ns? ¿Y si el periodo del reloj fueran 200 ns? Razona y demuestra cada respuesta.
2. Suponiendo que se tiene un procesador matricial con N elementos de proceso, explica cómo se pueden multiplicar dos matrices a y b de dimensión N. Explica cómo se distribuyen los elementos de las matrices en las memorias de los elementos de proceso y donde se encontraría el resultado. Da el orden del coste temporal de este algoritmo y compáralo con el de un escalar que sería $O(N^3)$
3. Explica los diferentes retrasos que se producen cuando se envía un paquete a través de una red con un control de flujo cualquiera. Como ejemplo concreto explica además el tiempo que tarda un paquete en ir de un origen a un destino suponiendo un control de flujo de tipo lombriz. Explica también la fórmula obtenida del retraso para la conmutación de lombriz y haz un esquema nodo/tiempo suponiendo tres nodos en el camino.
4. Dado el siguiente algoritmo de encaminamiento completamente adaptativo, justificar si está libre de bloqueos mortales o no. Se debe razonar la respuesta con todo detalle, explicando en profundidad por qué está libre de bloqueos o por qué no lo está. Este algoritmo se aplica a una red malla 2D, con conmutación de paquetes, los canales x_a , x_b , y_a , y_b son los canales virtuales en las dimensiones X e Y (un signo '+' o '-' añadido al canal indica el sentido en cada dimensión).

```
Inputs: Xdest,Ydest,Xact,Yact; // Coordenadas nodo destino y actual
Outputs: Canal; // Canal a tomar
```

Procedimiento:

```
Xoff=Xdest-Xact; // Coordenadas relativas
Yoff=Ydest-Yact; // o desplazamiento (offset)
if Xoff<0 then
{ if Yoff<0 then Canal=(Xa-,Xb-,Ya-); // Uno de estos a elegir.
  if Yoff>0 then Canal=(Xa-,Xb-,Ya+);
  if Yoff==0 then Canal=(Xa-,Xb-);
}
if Xoff>0 then
{ if Yoff<0 then Canal=(Xa+,Xb+,Ya-,Yb-);
  if Yoff>0 then Canal=(Xa+,Xb+,Ya+,Yb+);
  if Yoff==0 then Canal=(Xa+,Xb+);
}
if Xoff==0 then
{ if Yoff<0 then Canal=(Ya-,Yb-);
  if Yoff>0 then Canal=(Ya+,Yb+);
  if Yoff==0 then Canal=Interno; // Ya ha llegado
}
```