

1. Para un hipercubo de dimensión 6, ¿Cuántos nodos tiene el nodo 13 a distancia 2?

Nodo 13 = 001101 al ser 6 dimensiones, trataremos con 6 dígitos

Por lo tanto, queremos calcular todos los nodos que estén a distancia hamming de 2. Para ello:

$$\binom{n}{d} = \frac{n!}{d!(n-d)!}$$

$n = \text{Numero de dimensiones}$
 $d = \text{Distancia buscada}$

$$\binom{6}{2} = \frac{6!}{2!(6-2)!} = 15$$

El nodo 13 del hipercubo tiene 15 nodos a distancia 2.

2. Un multicomputador utiliza una red de comunicación en la que los enlaces son de 1Gb/s. La técnica de comunicación es almacenamiento y reenvío. Mandar un paquete de 32 bytes a una distancia de 6 cuesta 1.56µs. ¿Cuántas veces sería más rápida la comunicación si la técnica de comunicación/conmutación fuera Virtual Cut-Through (VCT)? Se supone tráfico 0, flits de 8 bits y un flit de cabecera.

$$s\&f \rightarrow t_{s\&f} = D(tr + tw) + D*tw[L/W]$$

$$vct \rightarrow t_{vct} = D(tr + tw) + tw[L/W]$$

$$t_{s\&f} = 1.56\mu s \rightarrow 1560 \text{ ns}$$

$$D = 6$$

$$tr ??$$

$$tw ?? \rightarrow (\text{numero bits del flit}) * 1/\text{AnchoBanda}$$

$$1Gb/s \rightarrow 10^9 \text{ bits/s}$$

$$tw (\text{para un flit}) = 8 * (1/10^9) = 8 \text{ ns};$$

De los 32 bytes que enviamos omitimos la cabecera, trabajamos en bytes ya que t_r y t_w es tiempo por flit, y un flit son 8 bits \rightarrow 1 byte.

$$1560 \text{ ns} = 6(tr + 8ns) + 6 * 8 * (31)$$

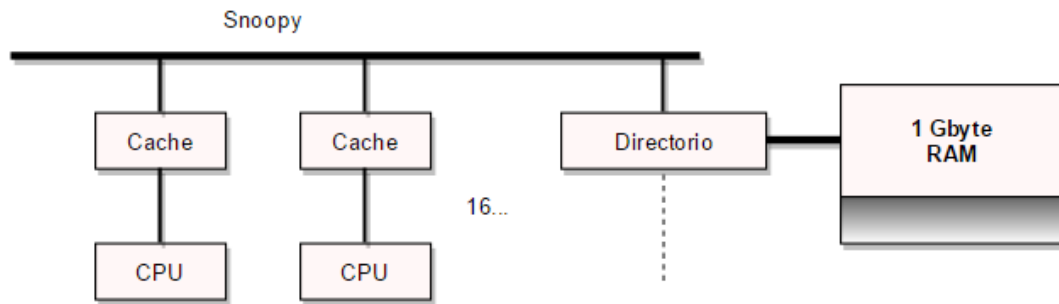
$$tr = 4 \text{ ns}$$

¿Ahora bien cuanto es más rápido?

$$t_{vct} = 6(4 + 8) + 8*(31) = 320 \text{ ns}$$

$$SP = t_{s\&f} / t_{vct} \rightarrow 1560 / 320 \approx 4$$

3. Tengo una arquitectura SMP p+1 centralizado con caches. Disponemos de 32 clusters snoopy (1 por directorio) en los cuales tenemos 16 procesadores por clúster. Los clusters disponen de 1GByte de memoria y 1 Mbyte de caché, siendo las líneas de ésta última de 128 bytes, además necesitaremos 2 bits por cada línea de caché para que ésta pueda ser referenciada. ¿Cuál será la sobrecarga de memoria necesaria para mantener este sistema?



EN CACHE:

- Líneas de caché: $\frac{1 \text{ Mbyte}}{128 \text{ bytes}} = \frac{2^{20}}{2^7} = 2^{13} = 2^{10} + 2^3 = \text{8K líneas}$
- Numero de cachés: $32 * 16 = 512$. Cada procesador tiene su caché.
- $512 \text{ caches} * 8\text{k líneas} * 2 \text{ bits/lineacache} = 2^{23} = \text{1MByte extra por cache.}$

EN DIRECTORIO:

- $P+1 = 32+1$ (un bit por nodo + 1 adicional) = **33 bits por directorio**

Característica del p+1 centralizado, un bit por nodo más uno adicional. Recordamos que estamos en el directorio.

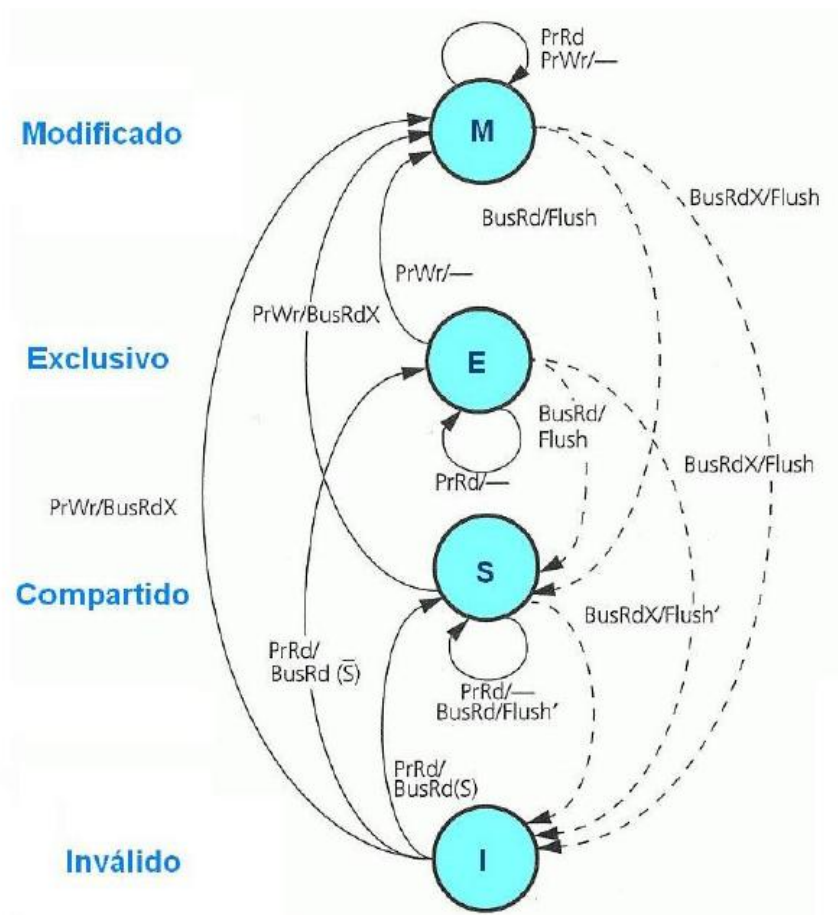
- Bloques de RAM: $\frac{1 \text{ Gbyte}}{128 \text{ bytes}} = \text{8 MBloques por directorio.}$
- $8 \text{ MB/directorio} * 33 \text{ bits/directorio} * 32 \text{ directorios} = \text{8448 Mbits}$

4. Disponemos de un multiprocesador con 4 nodos, cada nodo tiene caché, el protocolo snoopy (o de sondeo) elegido es el Illinois (NOTA: Llamado también MESI, pero tened cuidado y aprenderos los dos nombres). Las caches pueden alojar un máximo de 2 palabras, siendo el tamaño de bloque de 1 palabra y la política de reemplazo aleatoria. Dadas las siguientes referencias a memoria, indique cómo evoluciona el estado de las cachés y de la memoria principal.

```
rd1 @1; wr1 (@1, 22);
wr3 (@2, -11); rd2 @4
```

Cache 1			Cache 2			Cache 3			Cache 4			MEMORIA PRINCIPAL			
@1	5	E	@1	-3	I	@2	9	S	@2	9	S	@1	5	@2	9
@2	9	S	@5	2	E	@3	6	M	@4	10	M	@3	10	@4	10
												@5	2	@6	9

Lo primero, dibujamos el diagrama MESI



1. Acceso 1. rd1 @1

Se accede a la línea 1 de la caché 1. ¿Se produce fallo de caché?

Cache 1			Cache 2			Cache 3			Cache 4			MEMORIA PRINCIPAL			
@1	5	E	@1	-3	I	@2	9	S	@2	9	S	@1	5	@2	9
@2	9	S	@5	2	E	@3	6	M	@4	10	M	@3	10	@4	10
												@5	2	@6	9

→ Acierto en lectura

Pr1Rd/- → P1 lee, no se inyecta nada en el bus

Pasa de estado E a estado E.

2. Acceso 2. Wr1 (@1, 22)

Se accede a la línea 1 de la caché 1. ¿Se produce fallo de caché? Se escribe en la línea 1 de la caché 1 el valor 22. (El @ indica la línea, el numero detrás de la instrucción el número de caché)

Cache 1			Cache 2			Cache 3			Cache 4			MEMORIA PRINCIPAL			
@1	22	M	@1	-3	I	@2	9	S	@2	9	S	@1	22	@2	9
@2	9	S	@5	2	E	@3	6	M	@4	10	M	@3	10	@4	10
												@5	2	@6	9

Al actualizar la caché 1, se actualiza también la memoria principal.

3. Acceso 3. Wr3 (@2, -11)

Se accede a la línea 1 de la caché 3 y se escribe un -11.

Cache 1			Cache 2			Cache 3			Cache 4			MEMORIA PRINCIPAL			
@1	22	M	@1	-3	I	@2	-11	M	@2	9	I	@1	22	@2	-11
@2	9	I	@5	2	E	@3	6	M	@4	10	M	@3	10	@4	10
												@5	2	@6	9

4. Acceso 4. Rd2 @4

Se accede a la línea 4 de la caché 2 y se lee el contenido de la misma. Como la línea 4 en la caché 2 no existe, tenemos que referencia a la política de reemplazo expuesta en el enunciado, en este caso, aleatoria.

Cache 1			Cache 2			Cache 3			Cache 4			MEMORIA PRINCIPAL			
@1	22	M	@4	10	S	@2	-11	M	@2	9	I	@1	22	@2	-11
@2	9	I	@5	2	E	@3	6	M	@4	10	S	@3	10	@4	10
												@5	2	@6	9