KØBENHAVNS UNIVERSITET

PMPH - Assignment 4

Mikkel Willén

13. oktober 2022

Indhold

Task 1

Task 1.a and 1.b

MSI

Processor access	Operation	# cycles	Traffic
R_1/x	Read Request	40 cycles	6 + 32 bytes
W_1/x	Bus Upgrade	10 cycles	10 bytes
W_1/x	Write Hit	1 cycle	0 bytes
R_2/x	Read Request	40 cycles	6 + 32 bytes
W_2/x	Bus Upgrade	10 cycles	10 bytes
W_2/x	Write Hit	1 cycle	0 bytes
R_3/x	Read Request	40 cycles	6 + 32 bytes
W_3/x	Bus Upgrade	10 cycles	10 bytes
$W_3$1/x	Write Hit	1 cycle	0 bytes
R_4/x	Read Request	40 cycles	6 + 32 bytes
W_4/x	Bus Upgrade	10 cycles	10 bytes
W_4/x	Write Hit	1 cycle	0 bytes
Total		204 cycles	192 bytes

MESI

Processor access	Operation	# cycles	Traffic
R_1/x	Read Exlusive Request	40 cycles	6 + 32 bytes
W_1/x	Write Hit	1 cycles	0 bytes
W_1/x	Write Hit	1 cycle	0 bytes
R_2/x	Read Request	40 cycles	6 + 32 bytes
W_2/x	Bus Upgrade	10 cycles	10 bytes
W_2/x	Write Hit	1 cycle	0 bytes
R_3/x	Read Request	40 cycles	6 + 32 bytes
W_3/x	Bus Upgrade	10 cycles	10 bytes
$W_3$1/x	Write Hit	1 cycle	0 bytes
R_4/x	Read Request	40 cycles	6 + 32 bytes
W_4/x	Bus Upgrade	10 cycles	10 bytes
W_4/x	Write Hit	1 cycle	0 bytes
Total		193 cycles	182 bytes

Task 2

Task 2.a

Time	Miss type
1	Cold miss
2	Cold miss
3	Cold miss
4	Hit (invalidates block)
5	Cold miss
6	False sharing miss
7	Hit (invalidates block)
8	Replacement miss
9	True sharing miss

Task 2.b

We can ignore the miss on time 6, since the value it wants to read, hasn't been overwritten, though the entire block has been invalidated at time 4.

Task 3

Task 3.a

This is the same for MSI and DASH, the following table shows the steps and calculations:

Home = Local - Memory copy is clean

Operation	# cycles	Traffic
Read miss	1 cycle	0 bytes
Directory lookup	50 cycles	0 bytes
total	51 cycles	0 bytes

Task 3.b

This is the same for MSI and DASH, the following table shows the steps and calculations:

Home = Local - Memory copy is dirty

Operation	# cycles	Traffic
Read miss	1 cycle	0 bytes
Directory lookup	50 cycles	0 bytes
Remote read	20 cycles	6 bytes
Cache lookup	50 cycles	0 bytes
Flush	100 cycles	6 + 32 bytes
Install	50 cycles	0 bytes
total	271 cycles	44 bytes

Task 3.c

This is the same for MSI and DASH, the following table shows the steps and calculations:

Home ≠ Local - Memory copy is clean

Operation	# cycles	Traffic
Read miss	1 cycle	0 bytes
Bus read	20 cycles	6 bytes
Directory lookup	50 cycles	0 bytes
Flush	100 cycles	6 + 32 bytes
Install	50 cycles	0 bytes
total	221 cycles	44 bytes

Task 3.d

This is the same for MSI and DASH, the following table shows the steps and calculations:

Home ≠ Local, Home = Remote - Memory copy is dirty

Operation	# cycles	Traffic
Read miss	1 cycle	0 bytes
Bus read	20 cycles	6 bytes
Directory lookup	50 cycles	0 bytes
Flush	100 cycles	6 + 32 bytes
Install	50 cycles	0 bytes
total	221 cycles	44 bytes

Task 3.e

This is not the same for MSI and DASH. The following table shows the steps and calculations for MSI:

Home $\neq$ Local, Home $\neq$ Remote - Memory copy is Dirty

Operation	# cycles	Traffic
Read miss	1 cycle	0 bytes
Bus read	20 cycles	6 bytes
Directory lookup	50 cycles	0 bytes
Remote read	20 cycles	6 bytes
Cache lookup	50 cycles	0 bytes
Flush to home	100 cycles	6 + 32 bytes
Update home	50 cycles	0 bytes
Flush to local	100 cycles	6 + 32 bytes
Install local	50 cycles	0 bytes
total	441 cycles	88 bytes

and the following table shows the steps and calculations for DASH:

Home $\neq$ Local, Home $\neq$ Remote - Memory copy is Dirty

Operation	# cycles	Traffic
Read miss	1 cycle	0 bytes
Bus read	20 cycles	6 bytes
Directory lookup	50 cycles	0 bytes
Remote read	20 cycles	6 bytes
Cache lookup	50 cycles	0 bytes
Flush to home and local	100 cycles	6 + 32 bytes ·2
Update home and install local	50 cycles	0 bytes
total	291 cycles	88 bytes

Task 4

Task 4.a

For an n-by-n tori the network diameter is n, and the network diameter for this tori is therefore 16.

Task 4.b

The bisection width for an n-by-n tori is 2n, and we have:

$$\begin{aligned}\text{bisection bandwidth} &= \text{bisection width} \cdot \text{link bandwidth} \\ &= 32 \cdot 100 \text{ Mbits/s} \\ &= 3.2 \text{ Gbits/s}\end{aligned}$$

Task 4.c

The total bandwidth is given by:

$$\begin{aligned}\text{total bandwidth} &= \text{number of links} \cdot \text{link bandwidth} \\ &= 2n^2 \cdot 100 \text{ Mbits/s}\end{aligned}$$

and we have the following:

$$\begin{aligned}\text{bandwith per node} &= \frac{\text{total bandwidth}}{\text{number of nodes}}\\ &= \frac{512 \cdot 100 \text{ Mbits/s}}{256}\\ &= 200 \text{ Mbits/s}\end{aligned}$$

Task 5

Task 5.a

We start by calculating the dimensions of the tori and the hypercube with the following formulas:

$$\text{size of n-by-n tori} = n^2$$
$$\text{size of hypercube} = 2^k$$

Size (N)	4	16	64	256
dimensions of n-by-n tori (n)	2	4	8	16
dimensions of hypercube (k)	2	4	6	8

Now we can compute the bisections width with the following formulas:

$$\text{bisection width of n-by-n tori} = 2n$$
$$\text{bisection width of hypercube} = 2^{k-1}$$

Size (N)	4	16	64	256
bisection width of n-by-n tori	4	8	16	32
bisection width of hypercube	2	8	32	128

As we can see from the calculations in the table, the hypercube has higher bisections width than the torus for $N \geq 64$.

Task 5.b

Size (N)	4	16	64	256
network diameter of n-by-n tori	2	4	8	16
network diameter of hypercube	2	4	6	8

Size (N)	4	16	64	256
switch degree of n-by-n tori	4	4	4	4
switch degree of hypercube	2	4	6	8

Network diameter and switch degree is both smaller for the hypercube at $N = 64$.

Task 5.c

When we get to networks of bigger sizes, the hypercube has higher bisection width, which means fewer potentiel bottlenecks, and lower network diameter, which means smaller worst case routing distance between two nodes. The switch degree is higher for the hypercube, which gives it a higher approximate cost.