Homework-4 (Cache miss experiments)
CO21BTECH11004

Single Cycle Processor Used

Program1: -

- In program1, there are two nested while loops and register x20 is loaded with 0(x12), and x12 is incremented by 8 in each iteration.
- When there is a cache miss, blocks are bought from the memory.
- In the while loop, this will happen: 1 cache miss, then (no of blocks - 1) cache hits. Thus, Program1 has a spatial locality.
- One hit-and-miss before the while loop.
-

Program2: -

- In program2, same as program1, two nested while loop. In the first while loop, x12 is equated to x3 + x10*8, and in the second while loop, x20 is loaded with the value 0(x12), and x12 is incremented by 64.

Program3: -

- In program3, same as program2 but two load are there in x20: - 0(x12) and 1024(x12).

Que1: -

- Lines
  - Program1: -
    - If we are increasing lines, the number of indexes is increasing, so there is no effect on the hit rate as 1 miss and 3 hits will be the same for any number of lines.

Program1 with L1(8,8) with lines variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
1.0	0.0	2.0	0.7424	49.0	17.0	66.0
2.0	0.0	2.0	0.7424	49.0	17.0	66.0
3.0	0.0	2.0	0.7424	49.0	17.0	66.0

Program1 with L1(16,16) with lines variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
1.0	0.0	2.0	0.7481	193.0	65.0	258.0
2.0	0.0	2.0	0.7481	193.0	65.0	258.0
3.0	0.0	2.0	0.7481	193.0	65.0	258.0

- Program2: -
  - X12 is increment by 64(2^6) , 8 times, so 8 different indexes each time if (line=3), so almost all cache miss for line=1,2,3.
  - As block = 2, the index is from bit 5-7 and 64 increments, so increment in the 6th bit.

Program2 with L1(8,8) with lines variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
1.0	0.0	2.0	0.03030	2.0	64.0	66.0
2.0	0.0	2.0	0.03030	2.0	64.0	66.0
3.0	0.0	2.0	0.04545	3.0	63.0	66.0

Program2 with L1(16,16) with lines variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
1.0	0.0	2.0	0.007752	2.0	256.0	258.0
2.0	0.0	2.0	0.007752	2.0	256.0	258.0
3.0	0.0	2.0	0.007752	2.0	256.0	258.0

- Blocks
  - Program1: -
    - If blocks are increasing, then the cache hit rate will increase as in program 1 has spatial locality.
    - 1 miss, so (no of blocks-1) hits.

Program1 with L1(8,8) with blocks variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
3.0	0.0	3.0	0.8636	57.0	9.0	66.0
3.0	0.0	4.0	0.9242	61.0	5.0	66.0
3.0	0.0	5.0	0.9545	63.0	3.0	66.0

Program1 with L1(16,16) with blocks variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
3.0	0.0	3.0	0.8721	225.0	33.0	258.0
3.0	0.0	4.0	0.9341	241.0	17.0	258.0
3.0	0.0	5.0	0.9651	249.0	9.0	258.0

- Program2: -
  - For index=3 and block =3, there is a cache miss for the first iteration of the while loop, then x12 = x3 + x10*8 (x10=1), so now all cache hit for all index but in the next block, and so on.
  - As blocks increase starting from 3 with index =3, misses are reduced by a factor of 2. See the table below.
  - The above two points were for L1(8,8). For L1(16,16) we need to start from index=4 and block =4 to see a similar effect.

Program2 with L1(8,8) with blocks variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
3.0	0.0	3.0	0.8636	57.0	9.0	66.0
3.0	0.0	4.0	0.9242	61.0	5.0	66.0
3.0	0.0	5.0	0.9545	63.0	3.0	66.0

Program2 with L1(16,16) with lines variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
1.0	0.0	2.0	0.007752	2.0	256.0	258.0
2.0	0.0	2.0	0.007752	2.0	256.0	258.0
3.0	0.0	2.0	0.007752	2.0	256.0	258.0

- Ways
  - Program1: -
    - As the number of blocks per tag remains the same, the hit ratio will remain the same.

Program1 with L1(8,8) with ways variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
3.0	0.0	2.0	0.7424	49.0	17.0	66.0
3.0	1.0	2.0	0.7424	49.0	17.0	66.0

Program1 with L1(16,16) with ways variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
3.0	0.0	2.0	0.7481	193.0	65.0	258.0
3.0	1.0	2.0	0.7481	193.0	65.0	258.0

  - Program2: -
    - Basically, providing n block in the set index, so misses decrease for 1-way L1(8,8)
    - For L1(16,16) misses will decrease if we increase ways to 3. Here no change as ways =0,1.

Program2 with L1(8,8) with ways variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
3.0	0.0	2.0	0.04545	3.0	63.0	66.0
3.0	1.0	2.0	0.74240	49.0	17.0	66.0

Program2 with L1(16,16) with ways variation

lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
3.0	0.0	2.0	0.007752	2.0	256.0	258.0
3.0	1.0	2.0	0.007752	2.0	256.0	258.0

Que2: -

Program1: -

- In withAllocate means, if the write misses, the cache pulls the block from memory and writes again.
- In noAllocate, if write hits cache is updated, else memory is updated, so more is missed in this scheme in the following tables.

Program1 with L1(8,8)

	lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses	writeBacks
writeThrough_withAllocate	5.0	0.0	2.0	0.74240	49.0	17.0	66.0	64.0
writeThrough_noAllocate	5.0	0.0	2.0	0.04545	3.0	63.0	66.0	64.0
writeBack_withAllocate	5.0	0.0	2.0	0.74240	49.0	17.0	66.0	0.0
writeBack_noAllocate	5.0	0.0	2.0	0.04545	3.0	63.0	66.0	62.0

Program1 with L1(16,16)

	lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses	writeBacks
writeThrough_withAllocate	5.0	0.0	2.0	0.74810	193.0	65.0	258.0	256.0
writeThrough_noAllocate	5.0	0.0	2.0	0.01163	3.0	255.0	258.0	256.0
writeBack_withAllocate	5.0	0.0	2.0	0.74810	193.0	65.0	258.0	0.0
writeBack_noAllocate	5.0	0.0	2.0	0.01163	3.0	255.0	258.0	254.0

Program2: -

- For L1(8,8), the logic of problem 1 works, but for L1(16, 16) hitRate remains the same.
- There will be an increase in hit rate with increased lines, blocks/ways for the withAllocate scheme.

Program2 with L1(8,8)

	lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses	writeBacks
writeThrough_withAllocate	5.0	0.0	2.0	0.74240	49.0	17.0	66.0	64.0
writeThrough_noAllocate	5.0	0.0	2.0	0.04545	3.0	63.0	66.0	64.0
writeBack_withAllocate	5.0	0.0	2.0	0.74740	49.0	17.0	66.0	0.0
writeBack_noAllocate	5.0	0.0	2.0	0.04545	3.0	63.0	66.0	62.0

Program2 with L1(16,16)

	lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses	writeBacks
writeThrough_withAllocate	5.0	0.0	2.0	0.01163	3.0	255.0	258.0	256.0
writeThrough_noAllocate	5.0	0.0	2.0	0.01163	3.0	255.0	258.0	256.0
writeBack_withAllocate	5.0	0.0	2.0	0.01163	3.0	255.0	258.0	223.0
writeBack_noAllocate	5.0	0.0	2.0	0.01163	3.0	255.0	258.0	254.0

Que3: - Program3: -

- In direct-mapped (32-entry 4-word), as both 0(x12) and 1024(x12) have the same index but different tags, almost misses would be there.
- In fully associative, the first time 0(x12) and 1024(x12), there is a miss, and for the next 3 blocks, there is a mit, so the hit rate increases.
- In a 2-way set, the associative hit rate increases, same thing happens here also, the first time 0(x12) and 1024(x12), there is a miss, and for the next 3 blocks, there is a mit, so the hit rate increases.

Program3 with L1(8,8)

	lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
32-entry 4-word direct-mapped	5.0	0.0	2.0	0.01538	2.0	128.0	130.0
32-entry 4-word 2-way set associative	4.0	1.0	2.0	0.73850	96.0	34.0	130.0
32-entry 4-word fully associative	0.0	5.0	2.0	0.73850	96.0	34.0	130.0

Program3 with L1(16,16)

	lines	ways	blocks	hitRate	nHits	nMisses	totalAccesses
32-entry 4-word direct-mapped	5.0	0.0	2.0	0.06809	35.0	479.0	514.0
32-entry 4-word 2-way set associative	4.0	1.0	2.0	0.74710	384.0	130.0	514.0
32-entry 4-word fully associative	0.0	5.0	2.0	0.74710	384.0	130.0	514.0