CPSC 3300

Homework 5

Reagan Leonard

Due 11:59 PM Wednesday, April 15

1. Consider an array declared in C as "double a[100];". How many 64-byte cache lines are required to hold the complete array? [8pts]

total # lines in cache = cache size / line size

= (100\*8) / 64 bytes

= 100 / 8

= 12.5, rounded up is **13 64-byte cache lines**

1. Consider the byte address 0x002468ac. What is the value modulo 64? (That is, what is the offset of this address within a 64-byte block?) [8pts]

0x002468ac in binary is 0000 0000 0010 0100 0110 1000 1010 1100

64 = 2^6 so we right shift by 6 bits

0000 0000 0000 0000 1001 0001 1010 0010 and (10 1100) is what’s leftover

So we have 0010 1100 which in hex is 0x0000002c

So 0x002468ac % 64 = **0x0000002c**

1. Consider the byte address 0x002468ac. What is the value shifted to the right by 6 bits? (That is, what is the block address corresponding to this byte address when using 64-byte blocks?) [8pts]

As worked out above, 0x002468ac in binary is 0000 0000 0010 0100 0110 1000 1010 1100

Shifted right 6 bits, this is 0000 0000 0000 0000 1001 0001 1010 0010

In hex, this is **0x000091a2**

1. Consider matrix transpose written in C. Which array is exhibiting spatial locality: array "a", "b", or both? (Note that NROWS and NCOLS could each be relatively large compared to the size of the cache.) [8pts]

for(i=0;i<NROWS;i++){

for(j=0;j<NCOLS;j++){

b[i][j] = a[j][i];

}

}

In the inner loop, b[0][1], b[0][2] … b[0][n] will be set to a[1][0], a[2][0] , … a[n][0] **so b is exhibiting spatial locality but not a** because b[0][1] and b[0][2] are likely to be next to each other in contiguous memory.

1. Consider a 4 GB byte-addressable main memory (32-bit address) with a level-1 data cache that is eight-way set-associative, 32 KB in size, with 64-byte block size. [24pts (8pts each)]
2. How many total blocks are there in cache?

# of blocks = cache size / block size = 32 KB / 64 B

32 KB = 32,768 B to be precise

So 32 KB / 64 B = (32768/64) = **512 total blocks**

1. How many sets are there?

There will be **8 sets** of 64 blocks a piece

1. Show how the main memory address is partitioned into fields for the cache access and give the bit lengths of these fields.

A close up of a piece of paper

Description automatically generated

1. Consider a direct-mapped data cache design in which a 32-bit address is divided into these three fields: 18-bit tag, 10-bit index, and 4-bit offset. [24pts (6pts each)]
2. How large is a line in number of bytes?

2^offset bits = 2^4 = **16 bytes**

1. How many lines are in the cache?

2^index bits = 2^10 = **1024 lines**

1. How large is the cache in number of bytes?

1024 lines at 16 bytes each = 1024\*16 = **16384 bytes**

1. For the following segment of code written in C, where "sum" and the array "a" are typed as 4-byte integers, what is the miss rate?

(Assume the variable "sum" and the loop index "i" are register-allocated by the compiler within the body of the loop and thus do not cause data cache accesses within the loop.)

for(i=0;i<4096;i++){

sum = sum + a[i];

}

Block size = 16 bytes, so we have 4 4-byte ints per block

So when accessing 4 integers in a contiguous memory location, the first access will be a

a miss and the 3 subsequent accesses will be hits. Since 4096 integers will be accessed

and this pattern of 1 miss followed by 3 hits will repeat throughout, 1 out of every 4

accesses will be a miss (1024/4096). So the miss rate will be **25%.**

1. Assume a 256-byte main memory and a four-line cache with four bytes Per line. The cache is initially empty. For the byte address reference stream (reads) given below circle which of the references are hits for the different cache placement schemes. Also, show the final contents of the cache. (The byte addresses are in decimal.) [20pts (10pts each)]

a) direct-mapped

0, 16, 8, 1, 10, 30, 18, 29, 2, 25

A close up of a piece of paper

Description automatically generated

2

hit

hit



b) fully-associative with first-in-first-out replacement

0, 16, 8, 1, 10, 30, 18, 29, 2, 25

1 is a hit

A close up of a piece of paper

Description automatically generated

5

hit

hit

hit

hit

