Guerrilla Session 5: Caches

3) "*Cache*, money. Dollar bills, y'all." (24 min, 15 pts)

Suppose we have a standard 32-bit byte-addressed MIPS machine, a single direct-mapped 32KiB cache, a write-through policy, and a 16B block size.

a) Give the T:I:O breakup. 17:11:4 16*8(data)+17(tag)+1(valid) = 146

b) How many bits are there per row on the cache? _____

Use the C code below and the description of the cache above to answer the questions that follow it. Suppose that the only memory accesses are accesses and stores to arrays and that all memory accesses in the code are valid. Assume `A` starts on a block boundary (byte 0 of `A` in byte 0 of block).

```
#define NUM_INTS 32
#define OFFSET   8192    // 8192 = 2^13

int rand(int x, int y); // returns a random integer in the range [x, y)

int main(){
      int A[NUM_INTS + OFFSET];              // Assume A starts on a block boundary

      // START LOOP 1
      for ( int count = 0 ; count < NUM_INTS ; count += 1 ) { // count by 1s
            A[count]          = count;       // ACCESS #1
            A[count + OFFSET] = count+count; // ACCESS #2
      }
      // END LOOP 1

      // START LOOP 2
      for ( int count = 0 ; count < NUM_INTS ; count += 4 ) { // count by 4s now
            for ( int r = 0 ; r < 4 ; r++ ) {                 // …but do it 4 times
                  printf("%d", A[rand(count, count+4)]);
            }
      }
      // END LOOP 2
}
```

c) Hit rate for Loop 1? 0% What types of misses are there? compulsory, conflict

d) Hit rate for Loop 2? 75% What types of misses are there? conflict

Questions (e), (f), and (g) below are three independent variations on the original code & settings.

e) If the cache were 2-way set associative, what would be the hit rate for Loop 2? 100%
(assume the standard LRU replacement policy)

f) If instead we removed the line labeled `ACCESS #2`, what would be the hit rate for Loop 2? 100%

g) Instead, what's the smallest we could shrink `OFFSET` to maximize our Loop 2 hit rate? 32
(assume we still need to maintain the same functionality)

Question 3: ***Our band is called 1023MiB... We haven't had any gigs yet.*** (24 min, 15 pts)

We have a standard 32-bit byte-addressed MIPS machine with a 1KiB direct-mapped write-back cache and 16B block size.

a) How many bits are used for the Tag? ____22____ …Index? ____6____ …Offset? ____4____

Consider the following C code, and answer the questions below. `d` and `s` are pointers to 8-bit unsigned integer arrays, of the same size (a multiple of the cache size) that are aligned on 16-byte boundaries. The arrays contain only one `0x00`, in the last byte. `d` and `s` are not necessarily distinct.

```
void our_strcpy(uint8_t *d, uint8_t *s) {
    char c;
    do {
         c = *s;
        *d = c;
        s++; d++;
    } while (c);
}
```

b) What is the *lowest* possible cache hit rate for `our_strcpy`? _____0_____

c) What *types* of misses are there? Compulsory and conflict

d) What is the *smallest possible value* of `(d - s)` that would get this hit rate? _____1 KiB_____

e) What is the *highest* possible cache hit rate for `our_strcpy`? ____31/32_____

f) What is one possible value of `(d - s)` where we would get this hit rate? _____0_____

2 misses per block * 2^6 blocks/cache * 2^13 caches = 2^20 misses

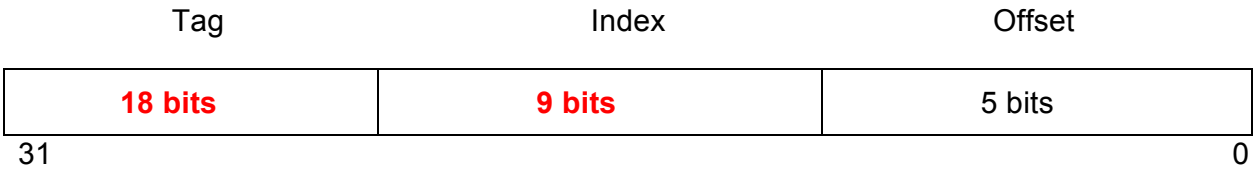
g) If we ran `our_strcpy` with a 4-way set-associative LRU cache, and the size of both `d` and `s` is 8MiB, what is the *most # of misses* possible? _____1 Mebi_____

Show all your work here!

SID: _____

Q4: Cache Operations (6 points)

a) Consider a 32-bit physical memory space and a 32 KiB 2-way associative cache with LRU replacement. You are told the cache uses 5 bits for the offset field. Write in the number of bits in the tag and index fields in the figure below.



b) Assume the same cache as in part a).

```
int ARRAY_SIZE = 64 * 1024;
int arr[ARRAY_SIZE]; // *arr is aligned to a cache block

/* loop 1 */ for (int i = 0; i < ARRAY_SIZE; i += 8) arr[i] = i;
/* loop 2 */ for (int i = ARRAY_SIZE - 8; i >= 0; i -= 8) arr[i+1] = arr[i];
```

1. What is the hit rate of loop 1? What types of misses (of the 3 Cs), if any, occur as a result of loop 1?
   0% hit rate, Compulsory Misses

2. What is the hit rate of loop 2? What types of misses (of the 3 Cs), if any, occur as a result of loop 2?
   9/16 hit rate, Capacity Misses

Q5: AMAT (4 points)

Suppose you have the following system that consists of an:
L1$ with a local hit rate of 80% and a hit time of 2 cycles
L2$ with a global miss rate of 8% and a hit time of 15 cycles
DRAM accesses take 50 cycles

i. What is the AMAT of the L1 cache? _____

2 + 0.2 * (15 + 0.4 * 50) = 9

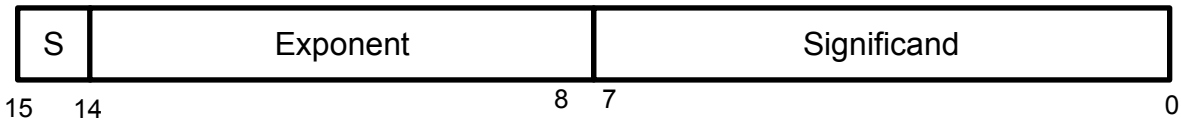
ii. Suppose we want to improve our AMAT, making sure that it is no greater than 6 cycles, by improving our L2$'s hit rate. What is the minimum possible local hit rate for L2$ that allows us to meet our AMAT requirement?

90%

SID: _____

Q6: Floating Point (6 points)

We want to develop a new half-precision floating-point standard for 16-bit machines. The basic structure is as follows:



Here are the design choices:

- 1 bit for sign
- 7 bits for a signed exponent in 2's complement
- 8 bits for the significand
- Everything else follows the IEEE standard 754 for floating point, except in 16 bits.

In this new standard:

a) Convert the decimal number -10.625 to floating point. Write your answer in hexadecimal.

-1010.101
sign: 1
exponent: 3 -> 0x03
significand: 01010100 -> 0x54

0x8354

b) What is the smallest even number that is not representable? **2^10+2**

c) What is the smallest positive denormalized number? **2^-70 or 0x4001**

SID: _____

M2-3) Cache Performance/AMAT (19 pts)

Each subproblem in this question can be done without the previous subproblems.
We have caches with the following configurations:
Cache-size: 1KiB, Block-size: 128B, Memory-size: 4 GiB, Write-policy: Write back and write-allocate

```
int rand(x,y);  // defined elsewhere, returns a random integer in the range [x,y)
#define repcount 4
int A[1024*1024]; // block aligned
int B[8*1024]; // block aligned
...
// ← Start (assume cache is cold)
for(int rep = 0; rep < repcount; rep++)
     for (int i = 0; i < 1024; i++)
          A[8*i] = B[8*i] + A[256*i + 8*i + rand(0, 32)] + A[8*i];
// ← Stop
```

For the following scenarios calculate the hit-rate and specify which types of misses are encountered in the code between the lines marked start and stop. Assume all variables stay in registers and memory reads happen left to right in each line.

a) i) Direct Mapped: # Tag/Index/Offset bits: [_22_____ : _3_____ : _7_____]

ii) What is the best-case hit rate? Also, state the types of misses.

- best case: A and B aren't within the same index at the start
- 32 ints / block
- 4 ints / block are accessed with the A[8*i] and B[8*i] accesses
- A[256*i + 8*i + rand(0,32)] always accesses a new block which shares same index as A[8*i] (since 256*i will make the addresses always differ by a multiple of the cache size)
- rand() doesn't affect anything
- B[8*i] hits 3/4 of the time
- A[256*i + 8*i + rand(0, 32)] always maps to same set as A[8*i] so neither read will ever hit
- A[8*i] write always hits (directly after read)
- 1/4*(1) + 1/4*(3/4) + 0*(1/4)+ 0*(1/4) = 7/16
- We encounter compulsory, capacity, and conflict misses.

iii) What is the worst-case hit rate? Also, state the types of misses.

- worst case: A and B are within the same index at the start
- Accesses keep replacing blocks except the write (since it directly follows the read to A[8*i]
- All misses except write
- Hit rate: 1/4*1 + 1/4*0 + 1/4*0 + 1/4*0 = 1/4
- We encounter compulsory, capacity, and conflict misses.

b) i) 2-way Set Assoc: # Tag/Index/Offset bits: [_23_____ : _2_____ : _7_____]

ii) What is the worst-case hit-rate? Assume we use LRU replacement. Also, state the types of misses.

- worst case: A and B are within the same index at start
- Consider what cache looks like at start:
- Load B[8*i], then load A[256*i + 8i+ rand(0, 32)], A[8*i] then kicks out B[8*i], the write on A[8*i] then hits, the load of B[8*i] then replaces A[256*i + 8i+ rand(0, 32)], A[256*i + 8i+ rand(0, 32)] then replaces A[8*i], then the cycle repeats
- Thus the only hits are on the write
- Hit rate: 1/4*1 + 1/4*0 + 1/4*0 + 1/4*0 = 1/4
- We encounter compulsory, capacity, and conflict misses.

c) Calculate the AMAT in cycles if the L1 local hit rate is 60%, with a hit time of 1 cycle, the L2 global hit rate is 20% with a hit time of 10 cycles and the main memory has a hit time of 200 cycles.

1 + 0.4*(10 + (1 - .2/.4) * 200) = 45