ETH login ID:
(Please print in capital letters)

Full name: ______________________

263-2300: How to Write Fast Numerical Code

ETH/CS, Spring 2012
Midterm Exam
Friday, April 27, 2012

Instructions

- Make sure that your exam is not missing any sheets, then write your full name and login ID on the front.
- The exam has a maximum score of 100 points.
- No books, notes, calculators, laptops, cell phones, or other electronic devices are allowed.

Problem 1 (12 = 4 each)	
Problem 2 (15 = 5 each)	
Problem 3 (18 = 14 + 4)	
Problem 4 (18)	
Problem 5 (16 = 6 + 3 + 4 + 3)	
Problem 6 (21 = 3 + 3 + 15)	
Total (100)	

Problem 1 (12 points)

1. Consider the following program:

```
double A[768*4096];
double B[768*4096];
double C[768];
int i, j;
for (i = 0; i < 768; i++)
  for (j = 0; j < 768; j++)
    C[i] += A[j * 4096] * B[i * 4096];
return;
```

   On a particular Core 2 platform, you know that the working set (the part of A that is reused in every iteration of the i-loop) fits into cache. Even though you measure the runtime with warmed up the cache, the performance is bad.

   (a) What is the most likely reason?

   (b) How would you resolve it?

2. Illustrate with a simple pseudocode example a case where a write-back cache should be preferrable over a write-through cache.

Problem 2 (15 points)

Consider the following loop performing a simple filter on an $n \times n$ array A of single precision floats.

```
float h = 1/2.;
for (i=1; i < n-1; i++)
  for (j=1; j < n-1; j++)
    B[i][j] = h*(A[i][j-1] + A[i][j+1] + A[i+1][j] + A[i-1][j]);
```

We assume a processor that can complete 1 floating point add and 1 floating point mult per cycle, and which has a memory bandwidth for loads of 2 bytes/cycle.

1. Determine a lower bound (in cycles) for the runtime based on the floating point operations performed.

2. Determine a lower bound (in cycles) based on loads (floating point only) and assuming a cold writeback cache with a cache block size of one double.

3. Based on the above, determine an upper bound (not asymptotic) on the operational intensity measured in flops/byte.

Problem 3 (18 points)

We consider double precision matrix-vector multiplication (MVM) in the form $y = Ax$, where x, y are of length n and A is $n \times n$. Assume a cache of size $C \geq 4n$ doubles, and a cache block size of 8 doubles. Further we assume a cold cache.

1. Using the above assumptions, estimate the number of cache misses (as function of n) of the below standard double loop MVM. Ignore possible conflicts between the placement of x, y, A in cache.

   ```
   for (i = 0; i < n; i++)
      for (j = 0; j < n; j++)
         y[i] += A[i][j]*x[j];
   ```

   Give enough detail so we know how you did it.

2. Using the result of the previous task give a bound on the operational intensity in O notation (including very short explanation).

Problem 4 (18 points)

Given is the following Matlab function implementing a divide-and-conquer algorithm. The input T is an $n \times n$ matrix, where $n = 2^d$. The output F is also an $n \times n$ matrix. The auxiliary function `aux(A, B, C)` assumes all matrices A, B, C are $k \times k$ for some k and requires $2k^3$ many floating point operations.

```
function F = func(T)

n = size(T);

% base case: n = 1
if n < 2
  F = T(1,1)^2;
else
  m = n/2;
  u = 1:m;
  v = m+1:n;

  % recurse
  F1 = func(T(u,u));
  F3 = func(T(v,v));

  C  = F1*T(u,v); % matrix multiplication
  F2 = aux(T(u,u), T(v,v), C);

  F = [F1 F2; zeros(n-m,m) F3]; % no ops here
end
```

Compute the exact floating point operations count $C(n)$ for an $n \times n$ input T. **Show the derivation.**

The formula $f_k = ca^k + \sum_{i=0}^{k-1} a^i s_{k-i}$ solving $f_0 = c,\ f_k = af_{k-1} + s_k,\ k \geq 1$ may be helpful.

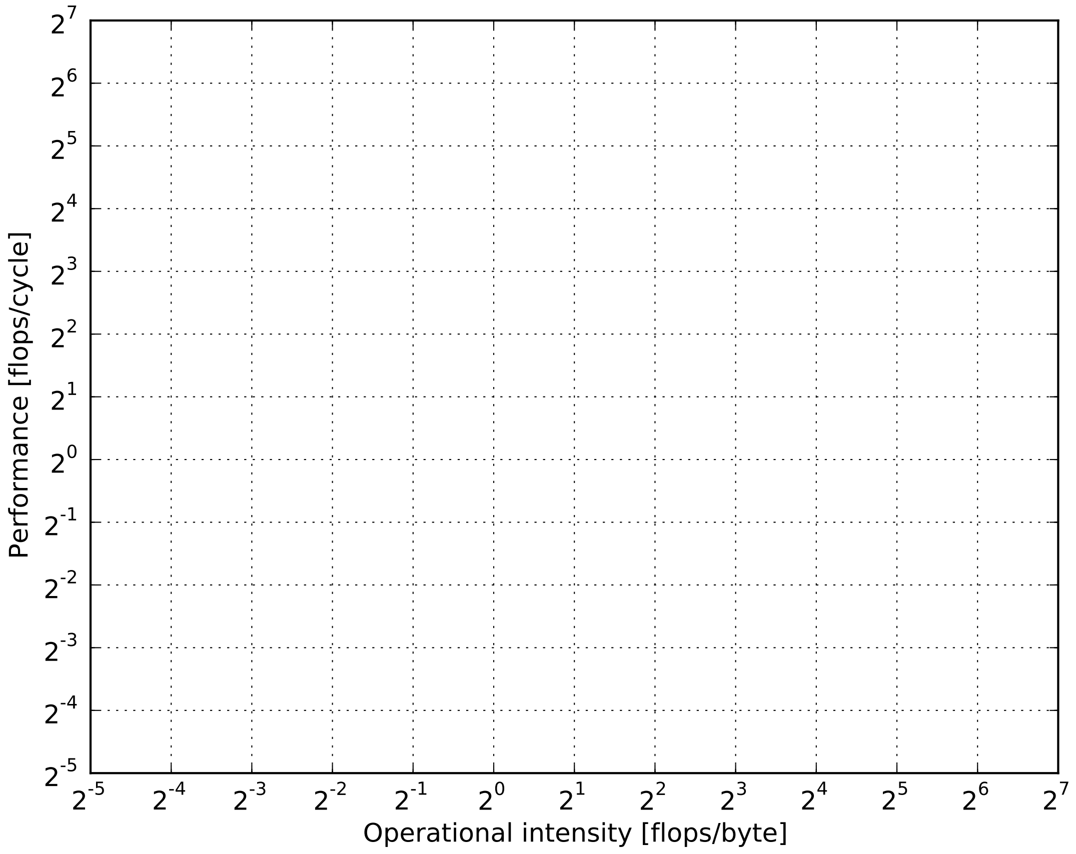
Problem 5 (16 points)

Assume you are using a system with the following features:

- A processor with a peak performance of 2 Gflop/s (double precision), and a CPU frequency of 1 GHz.
- The interconnection between CPU and main memory has a maximal bandwidth of 2 GB/s.

Answer the following questions:

1. Draw the roofline plot for this system. The units for x-axis and y-axis are performance in flops/byte and operational intensity in flops/cycle, both in log scale. The plot will contain two lines determining upper bounds on the achievable performance.



2. Assume the CPU frequency is increased to 2 GHz. How does this affect the two lines in 1.? (No need to redraw, just say.)

3. Consider the following code:

```
double v[64];
double sum = 0.;
double mul = 1.;

  for(i = 0; i < 64; i++) {
    sum += 3*v[i];
    mul *= 2*(v[i]^2) + 5;
  }
```

Assuming a cold cache:

(a) Compute the operational intensity assuming you performed scalar replacement on this code.

(b) Based on the result: is the computation compute- or memory-bound (platform is still the one from the previous page) and why?

Problem 6 (21 points)

We consider the following program.

```
double A[4][4], B[4][4]

void transpose()
{
  int i, j;
  for (i = 0; i < 4; i++)
    for (j = 0; j < 4; j++)
      B[j][i] = A[i][j]
  return;
}
```

We assume a direct-mapped writeback cache of size 128 bytes and a cache block size of 16 bytes (2 doubles). We assume a cold cache. In the questions below only consider accesses to the arrays A and B. A is cache aligned, i.e., $A[0][0]$ would begin the first cache block. $B[0][0]$ is in memory directly after $A[3][3]$.
Answer the following

1. Where is spatial locality?

2. Where is temporal locality?

3. Determine the hit-miss sequence (something like HMMHHM...) for both A and B. It helps to draw the cache after each iteration of the i-loop and derive the sequence iteration by iteration. Show enough detail so we know how you did it.