ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΜ&ΜΥ

Προηγμένα Θέματα
Αρχιτεκτονικής Υπολογιστών

4η Άσκηση
Ακ. έτος 2011-2012

Γρηγόρης Λύρας Α.Μ.: 03109687

18 Ιουλίου 2012

Εισαγωγή

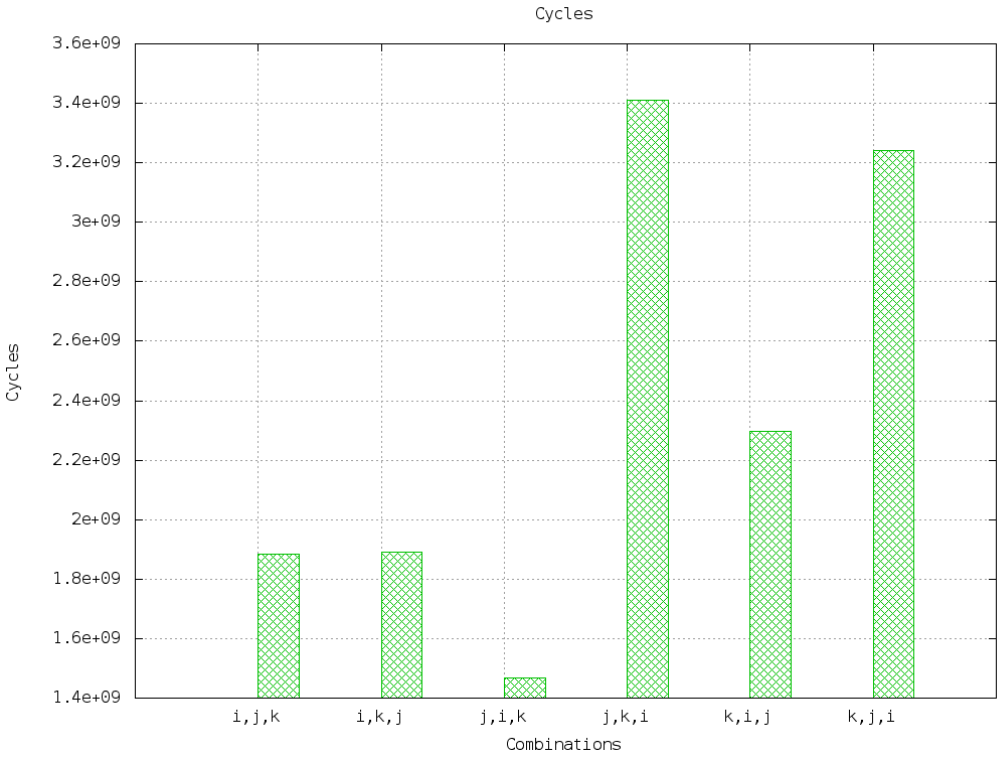
I) Προσομοίωση

```c
/* -.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.
 * File Name : partA.c
 * Creation Date : 16-07-2012
 * Last Modified : Mon 16 Jul 2012 01:46:39 PM EEST
 * Created By : Greg Liras <gregliras@gmail.com>
 _.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-.*/

#include <stdio.h>
#include <stdlib.h>
#define __MAGIC_CASSERT(p) do {                         \
    typedef int __check_magic_argument[(p) ? 1 : -1];   \
} while (0)

#define MAGIC(n) do {                       \
    __MAGIC_CASSERT(!(n));                  \
    __asm__ __volatile__ ("xchg %bx,%bx");  \
} while (0)

#define MAGIC_BREAKPOINT MAGIC(0)



inline int min(int a, int b)
{
    if(a<=b)return a;
    else return b;
}
void init_matrix(float **mat, int n)
{
    unsigned int i,j;
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            mat[i][j] = (float)(i+j);
}
int main(int argc, char **argv)
{
    float **A,**B,**C;
    int i,j,k,N;
    N=atoi(argv[1]);
    A=(float**)malloc(N*sizeof(float*));

    for(i=0; i<N; i++)
        A[i]=(float*)malloc(N*sizeof(float));

    B=(float**)malloc(N*sizeof(float*));

    for(i=0; i<N; i++)
        B[i]=(float*)malloc(N*sizeof(float));

    C=(float**)malloc(N*sizeof(float*));

    for(i=0; i<N; i++)
        C[i]=(float*)malloc(N*sizeof(float));

    fprintf(stderr, "Initializing matrices...\n");
    init_matrix(A, N);
    init_matrix(B, N);
    init_matrix(C, N);
    MAGIC_BREAKPOINT;
    for(i=0; i<N; i++) {
        for(j=0; j<N; j++)
            for(k=0; k<N; k++)
                C[i][j] += A[i][k]*B[k][j];
    }
    MAGIC_BREAKPOINT;
    return 0;
}
```

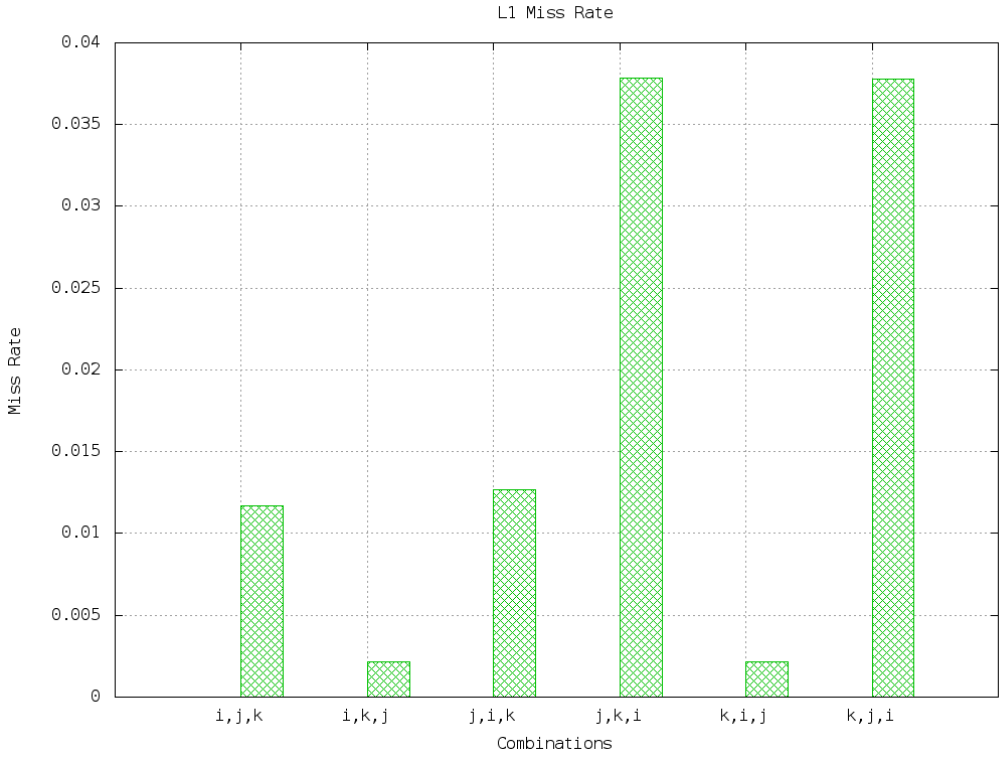
II) Ιεραρχία μνήμης και μοντέλο απόδοσης

Τεχνικές Βελτιστοποίησης

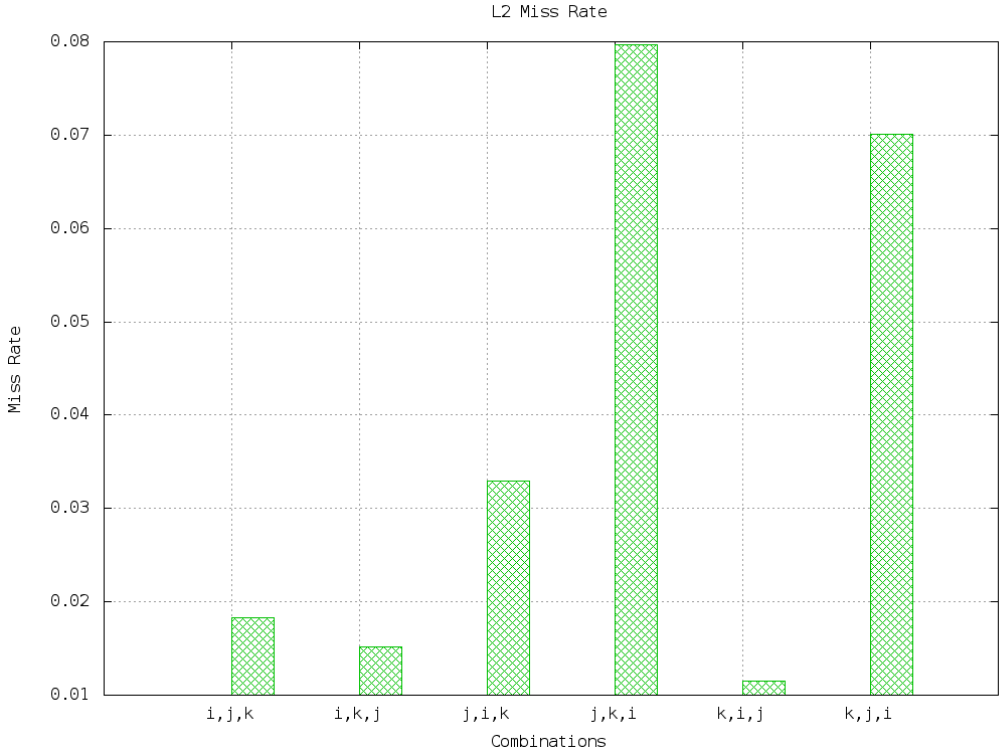
III) Loop Interchange



Σχήμα 1: Cycles



Σχήμα 2: L1 Miss Rate

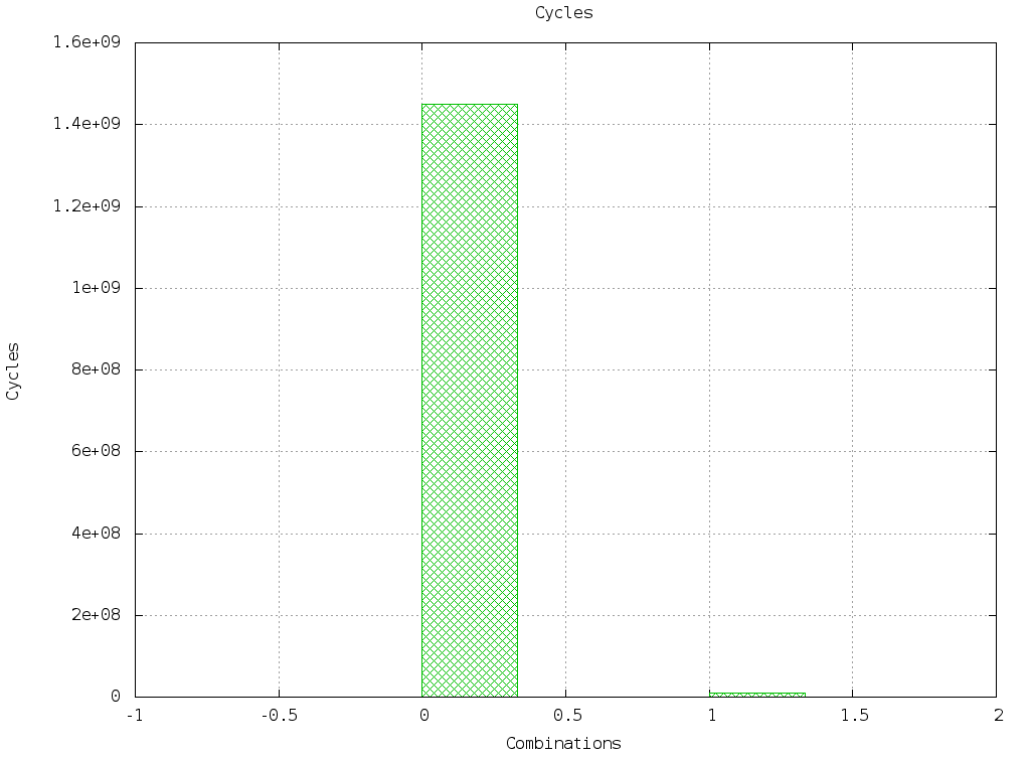


Σχήμα 3: L3 Miss Rate

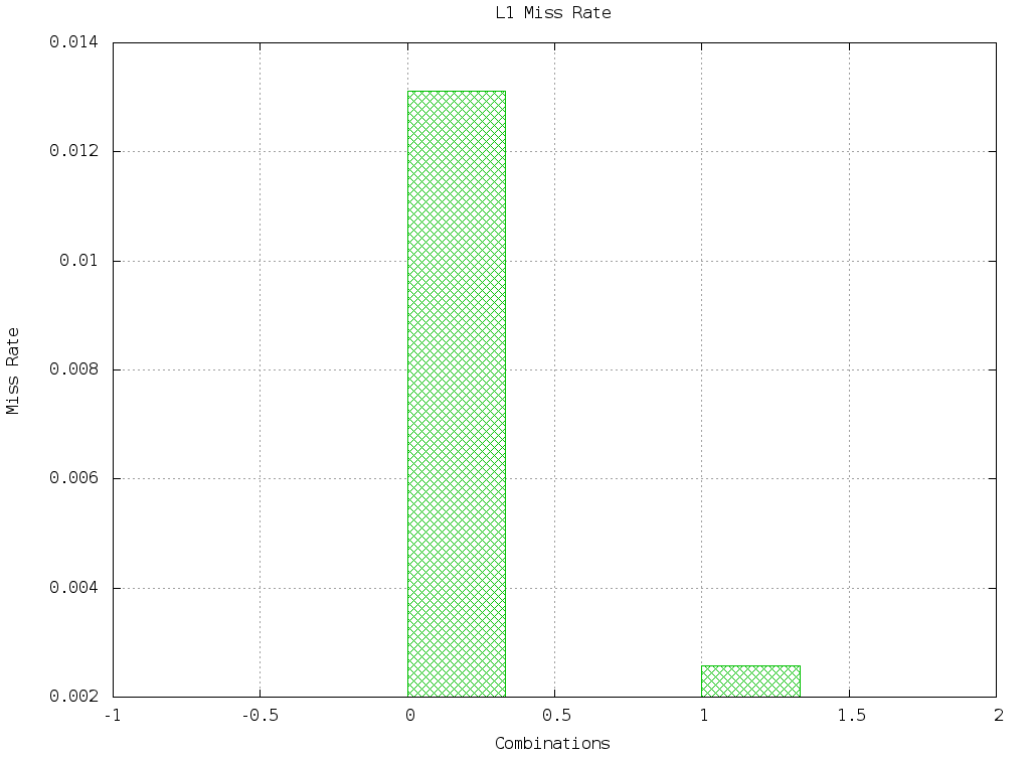
IV) Cache Blocking

```
1 /* -.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.
2  * File Name : partA.c
```

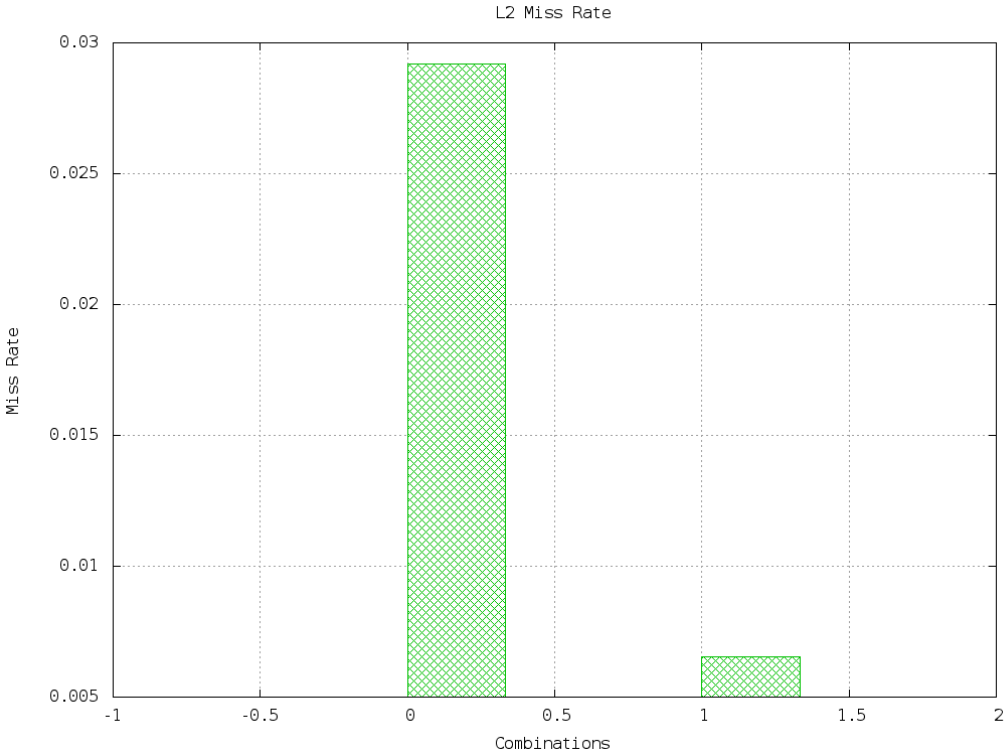
```
 * Creation Date : 16-07-2012
 * Last Modified : Tue 17 Jul 2012 06:07:55 PM EEST
 * Created By : Greg Liras <gregliras@gmail.com>
 _.-._._._._._._._._._._._._._._._._._._._._.*/

#include <stdio.h>
#include <stdlib.h>
#define __MAGIC_CASSERT(p) do {                         \
    typedef int __check_magic_argument[(p) ? 1 : -1];   \
} while (0)

#define MAGIC(n) do {                       \
    __MAGIC_CASSERT(!(n));                  \
    __asm__ __volatile__ ("xchg %bx,%bx");  \
} while (0)

#define MAGIC_BREAKPOINT MAGIC(0)



inline int min(int a, int b)
{
    if(a<=b)return a;
    else return b;
}
void init_matrix(float **mat, int n)
{
    unsigned int i,j;
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            mat[i][j] = (float)(i+j);
}
int main(int argc, char **argv)
{
    float **A,**B,**C;
    int i,j,k,N;
    int start,stop;
    N=atoi(argv[1]);
    A=(float**)malloc(N*sizeof(float*));

    for(i=0; i<N; i++)
        A[i]=(float*)malloc(N*sizeof(float));

    B=(float**)malloc(N*sizeof(float*));

    for(i=0; i<N; i++)
        B[i]=(float*)malloc(N*sizeof(float));

    C=(float**)malloc(N*sizeof(float*));

    for(i=0; i<N; i++)
        C[i]=(float*)malloc(N*sizeof(float));

    fprintf(stderr, "Initializing matrices...\n");
    init_matrix(A, N);
    init_matrix(B, N);
    init_matrix(C, N);
    MAGIC_BREAKPOINT;
    for(start=0; start<N; start+=16) {
        stop = start + 16;
        stop = stop <= N ? stop : N;
        for(j=start; j<stop; j++)
            for(i=start; i<stop; i++)
                for(k=start; k<stop; k++)
                    C[i][j] += A[i][k]*B[k][j];
    }
    MAGIC_BREAKPOINT;
    return 0;
}
```



Σχήμα 4: Cycles



Σχήμα 5: L1 Miss Rate



Σχήμα 6: L3 Miss Rate