دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

درس آز سیستم‌عامل

استاد حمید بیگی

---

پروژه پایانی — علی‌رضا دیزجی، امیرحسن فتحی، مصطفی قدیمی

---

## سؤال ۱. پیاده‌سازی ضرب ماتریسی ۴ در ۴ به کمک SSE

در این پروژه سعی کرده‌ایم تا ضرب ماتریسی را به کمک دستورالعمل‌های SSE انجام داده و آن را با ضرب ماتریسی معمولی مقایسه کرده و نتایج و مراحل پیاده‌سازی را گزارش کنیم.

- ضرب معمولی ماتریسی

```
void matmult_ref(Mat44 &out, const Mat44 &A, const Mat44 &B)
{
    Mat44 t; // write to temp
    for (int i=0; i < 4; i++) {
        for (int j=0; j < 4; j++) {
            t.m[i][j] = 0;
            t.m[i][j] += A.m[i][0]*B.m[0][j]
            t.m[i][j] += A.m[i][1]*B.m[1][j]
            t.m[i][j] += A.m[i][2]*B.m[2][j]
            t.m[i][j] += A.m[i][3]*B.m[3][j];
        }
    }
    out = t;
}
```

- به دست آوردن ترکیب خطی به کمک دستورات SSE

  در این قسمت در تلاش هستیم تا درایه‌های ماتریس اول را در ردیف‌های ماتریس دوم ضرب کنیم.

```
static inline __m128 lincomb_SSE(const __m128 &a, const Mat44 &B)
{
    __m128 result;
    result = _mm_mul_ps(_mm_shuffle_ps(a, a, 0x00), B.row[0]);
    result = _mm_add_ps(result, _mm_mul_ps(_mm_shuffle_ps(a, a, 0x55), B.row[1]));
    result = _mm_add_ps(result, _mm_mul_ps(_mm_shuffle_ps(a, a, 0xaa), B.row[2]));
    result = _mm_add_ps(result, _mm_mul_ps(_mm_shuffle_ps(a, a, 0xff), B.row[3]));
    return result;
}
```

- انجام ضرب به کمک SSE

```
void matmult_SSE(Mat44 &out, const Mat44 &A, const Mat44 &B)
{
    __m128 out0x = lincomb_SSE(A.row[0], B);
    __m128 out1x = lincomb_SSE(A.row[1], B);
    __m128 out2x = lincomb_SSE(A.row[2], B);
```

```
    __m128 out3x = lincomb_SSE(A.row[3], B);

    out.row[0] = out0x;
    out.row[1] = out1x;
    out.row[2] = out2x;
    out.row[3] = out3x;
}
```

- تست و آزمون کارکرد
  - اجرای ضرب معمولی ماتریسی

```
static void run_ref(Mat44 *out, const Mat44 *A, const Mat44 *B, int count)
{
    for (int i=0; i < count; i++)
    {
        int j = i & the_mask;
        matmult_ref(out[j], A[j], B[j]);
    }
}
```

  - اجرای ضرب به کمک SSE

```
static void run_SSE(Mat44 *out, const Mat44 *A, const Mat44 *B, int count)
{
    for (int i=0; i < count; i++)
    {
        int j = i & the_mask;
        matmult_SSE(out[j], A[j], B[j]);
    }
}
```

- اجرای کد

```
int main(int argc, char **argv)
{
    static const struct {
        const char *name;
        void (*matmult)(Mat44 &out, const Mat44 &A, const Mat44 &B);
    } variants[] = {
        { "ref",      matmult_ref },
        { "SSE",      matmult_SSE },
    };
    static const int nvars = (int) (sizeof(variants) / sizeof(*variants));

    srand(1234);

    // correctness tests
    for (int i=0; i < 1000000; i++)
    {
        Mat44 A, B, out, ref_out;
        randmat(A);
        randmat(B);
        matmult_ref(ref_out, A, B);
```

```
    for (int j=0; j < nvars; j++)
    {
        variants[j].matmult(out, A, B);
        if (memcmp(&out, &ref_out, sizeof(out)) != 0)
        {
            fprintf(stderr, "%s fails test\n", variants[j].name);
            exit(1);
        }
    }
}

printf("all ok.\n");

// performance tests

static const struct {
    const char *name;
    void (*run)(Mat44 *out, const Mat44 *A, const Mat44 *B, int count);
} perf_variants[] = {
    { "ref",      run_ref },
    { "SSE",      run_SSE },
};
static const int nperfvars = (int) (sizeof(perf_variants) / sizeof(*perf_variants));

/*
test done on my laptop with following results:
ref: 00.59 cycles
SSE: 52.20 cycles
*/

Mat44 Aperf, Bperf, out;
randmat(Aperf);
randmat(Bperf);

for (int i=0; i < nvars; i++)
{
    static const int nruns = 4096;
    static const int muls_per_run = 4096;
    unsigned long long best_time = ~0ull;

    for (int run=0; run < nruns; run++)
    {
        unsigned long long time = __rdtsc();
        perf_variants[i].run(&out, &Aperf, &Bperf, muls_per_run);
        time = __rdtsc() - time;
        if (time < best_time)
            best_time = time;
    }

    double cycles_per_run = (double) best_time / (double) muls_per_run;
    printf("%12s: 2f.% cycles\n", perf_variants[i].name, cycles_per_run);
}
```

```
    return 0;
}
```