



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;
}
```