```c
void sample_16(__m512 dl, __m512 ul, __m512 pl, __m512 cl,
               __m512 dr, __m512 ur, __m512 pr, __m512 cr,
               __m512 pm, __m512 um,
               __m512 *d, __m512 *u, __m512 *p)
{
    __m512 c, ums, pms, sh, st, s, uc;
    __mmask16 cond_um, cond_pm, cond_sh, cond_st, cond_s, cond_sh_st;

    // d/u/p/c/ums
    cond_um = _mm512_cmp_ps_mask(um, z, _MM_CMPINT_LT);
    *d = _mm512_mask_blend_ps(cond_um, dl, dr);
    *u = _mm512_mask_blend_ps(cond_um, ul, ur);
    *p = _mm512_mask_blend_ps(cond_um, pl, pr);
    c = _mm512_mask_blend_ps(cond_um, cl, cr);
    ums = um;
    *u = _mm512_mask_sub_ps(*u, cond_um, z, *u);
    ums = _mm512_mask_sub_ps(ums, cond_um, z, ums);

    // Calculate main values.
    pms = DIV(pm, *p);
    sh = SUB(*u, c);
    st = FNMADD(POW(pms, g1), c, ums);
    s = FNMADD(c, SQRT(FMADD(g2, pms, g1)), *u);

    // Conditions.
    cond_pm = _mm512_cmp_ps_mask(pm, *p, _MM_CMPINT_LE);
    cond_sh = _mm512_mask_cmp_ps_mask(cond_pm, sh, z, _MM_CMPINT_LT);
    cond_st = _mm512_mask_cmp_ps_mask(cond_sh, st, z, _MM_CMPINT_LT);
    cond_s = _mm512_mask_cmp_ps_mask(~cond_pm, s, z, _MM_CMPINT_LT);

    // Store.
    *d = _mm512_mask_mov_ps(*d, cond_st,
                            MUL(*d, POW(pms, SET1(1.0 / GAMA))));
    *d = _mm512_mask_mov_ps(*d, cond_s MUL(*d, DIV(ADD(pms, g6),
                                       FMADD(pms, g6, one))));
    *u = _mm512_mask_mov_ps(*u, cond_st | cond_s, ums);
    *p = _mm512_mask_mov_ps(*p, cond_st | cond_s, pm);

    // Low prob - ignore it.
    cond_sh_st = cond_sh & ~cond_st;
    if (cond_sh_st != 0x0)
    {
        *u = _mm512_mask_mov_ps(*u, cond_sh_st, MUL(g5, FMADD(g7, *u, c)));
        uc = DIV(*u, c);
        *d = _mm512_mask_mov_ps(*d, cond_sh_st, MUL(*d, POW(uc, g4)));
        *p = _mm512_mask_mov_ps(*p, cond_sh_st, MUL(*p, POW(uc, g3)));
    }

    // Final store.
    *u = _mm512_mask_sub_ps(*u, cond_um, z, *u);
}
```