```
void starpu_16(__m512 dl, __m512 ul, __m512 pl, __m512 cl,
               __m512 dr, __m512 ur, __m512 pr, __m512 cr,
               __m512 *p, __m512 *u)
{
    __m512 two, tolpre, tolpre2, udiff, pold, fl, fld, fr, frd, change;
    __mmask16 cond_break, cond_neg, m;
    const int nriter = 20;
    int iter = 1;

    two = SET1(2.0);
    tolpre = SET1(1.0e-6);
    tolpre2 = SET1(5.0e-7);
    udiff = SUB(ur, ul);

    guessp_16(dl, ul, pl, cl, dr, ur, pr, cr, &pold);

    // Start with full mask.
    m = 0xFFFF;

    for (; (iter <= nriter) && (m != 0x0); iter++)
    {
        prefun_16(&fl, &fld, pold, dl, pl, cl, m);
        prefun_16(&fr, &frd, pold, dr, pr, cr, m);
        *p = _mm512_mask_sub_ps(*p, m, pold,
                                _mm512_mask_div_ps(z, m,
                                                   ADD(ADD(fl, fr), udiff),
                                                   ADD(fld, frd)));
        change = ABS(_mm512_mask_div_ps(z, m, SUB(*p, pold),
                                        ADD(*p, pold)));
        cond_break = _mm512_mask_cmp_ps_mask(m, change,
                                             tolpre2, _MM_CMPINT_LE);
        m &= ~cond_break;
        cond_neg = _mm512_mask_cmp_ps_mask(m, *p, z, _MM_CMPINT_LT);
        *p = _mm512_mask_mov_ps(*p, cond_neg, tolpre);
        pold = _mm512_mask_mov_ps(pold, m, *p);
    }

    // Check for divergence.
    if (iter > nriter)
    {
        cout << "divergence in Newton-Raphson iteration" << endl;
        exit(1);
    }

    *u = MUL(SET1(0.5), ADD(ADD(ul, ur), SUB(fr, fl)));
}
```