

```

1.  #include <x86intrin.h>
2.  void dgemm (int n, double* A, double* B, double* C)
3.  {
4.      for (int i = 0; i < n; i+=8)
5.          for (int j = 0; j < n; ++j)
6.              {
7.                  __m512d c0 = _mm512_load_pd(C+i+j*n); // c0 = C[i][j]
8.                  for( int k = 0; k < n; k++ )
9.                      { // c0 += A[i][k]*B[k][j]
10.                         __m512d bb = _mm512_broadcastsd_pd(_mm_load_sd(B+j*n+k));
11.                         c0 = _mm512_fmadd_pd(_mm512_load_pd(A+n*k+i), bb, c0);
12.                      }
13.                  _mm512_store_pd(C+i+j*n, c0); // C[i][j] = c0
14.              }
15. }

```

FIGURE 3.21 Optimized version of DGEMM using C intrinsics to generate the AVX512 subword-parallel instructions for the x86. Figure 3.22 shows the assembly language produced by the compiler for the inner loop.