```
loop ind0 = 0;
                    Memory Level
                                                                 loop ind1 = 0:
                      Parallelism
                                                                 upper bound0 = ecx-ebx:
                                                                 upper bound1 = ecx-eax;
 loop ind0 = 0;
                                                                 bb10:
 loop ind1 = 0;
                                                                  loop ind1 = 0:
 upper bound0 = ecx-ebx;
                                                                  bb20:
 upper bound1 = ecx-eax;
                                                                   edx = *(mem + (edi + (eax + loop ind 1)*4)/sizeof(int)):
 bb10:
                                                                   *(mem + (ebx+(eax+loop ind1)*4)/sizeof(int)) = edx;
  loop ind1 = 0:
                                                                   loop ind1 = loop ind1+1;
  hh20:
                                                                   if(loop ind1 != upper bound1)
   edx = *(memLd+(edi + (eax+loop ind1)*4)/sizeof(int));
                                                                    goto bb20;
   *(memSt + (ebx+(eax+loop ind1)*4)/sizeof(int)) = edx;
                                                                  loop_ind0 = loop_ind0+1;
   loop ind1 = loop ind1+1;
                                                                  if(loop ind0 != upper bound0)
   if(loop ind1 != upper bound1)
                                                                   goto bb10
    goto bb20:
                                                                 bb end:
  loop ind0 = loop ind0+1:
                                                                                       Thread Level Parallelism
  if(loop ind0 != upper bound0)
                                    Vector to test: (*,*)
                                                                                          at the outer level
   goto bb10
 bb end:
                                                               loop ind0 0 = 0;
                   Add
                                                                loop ind0 1 = (ecx-ebx)/2;
                 barrier
                                                                upper bound0 0 = (ecx-ebx)/2;
                                                                upper bound0 1 = ecx-ebx:
                                                                loop ind1 = 0;
loop ind0 = 0;
                                                                upper bound1 = ecx-eax;
loop ind1 = 0;
                                                                bb10:
upper bound0 = ecx-ebx;
                                                                 loop ind1 = 0;
upper bound1 = ecx-eax;
                                                                 bb20:
bb10:
                                                                  edx = *(mem + (edi + (eax + loop ind 1)*4)/sizeof(int));
 loop ind1 = 0;
                                                                  *(mem + (ebx+(eax+loop ind1)*4)/sizeof(int)) = edx;
 bb20:
                                                                  loop ind1 = loop ind1+1;
  edx = *(memLd+(edi + (eax+loop_ind1)*4)/sizeof(int));
                                                                  if(loop ind1 != upper bound1)
  *(memSt + (ebx+(eax+loop ind1)*4)/sizeof(int)) = edx;
                                                                   goto bb20;
  loop ind1 = loop ind1+1;
                                                                 loop ind0 0 = loop ind0 0+1;
  if(loop ind1!= upper bound1)
                                                                 if(loop_ind0 != upper_bound0_0)
   goto bb20;
                                                                 goto bb10;
 barrier();
                                                                if(upper bound0 1 <= upper bound0 0)
 loop ind0 = loop ind0+1;
                                                                 goto bb end;
 if(loop ind0 != upper bound0)
                                 Vector to test: (=,*)
                                                                bb 10 dup:
 goto bb10
                                                                 loop ind1 = 0;
bb end:
                                                                 bb20:
                                                                  edx = *(mem+(edi + (eax+loop ind1)*4)/sizeof(int));
                                                                  *(mem + (ebx+(eax+loop ind1)*4)/sizeof(int)) = edx;
                                                                  loop_ind1 = loop_ind1+1;
                                                                  if(loop_ind1 != upper_bound1)
                                                                   goto bb20;
                                                                 loop ind0 1 = loop ind0 1+1;
                                                                 if(loop ind0 1!= upper bound0 1)
                                                                 goto bb10 dup;
                                                                                                     Vector to test: (<,*)
                                                               bb end:
```