```
hh20.
           edx = *(memLd1+(edi + eax*4)/sizeof(int));
                                                                                        Guard
                                                                      If(ecx-eax > 2)
           edx += *(memLd2 + (esi + eax*4)/sizeof(int));
                                                                                      condition
           *(memSt + (ebx+eax*4)/sizeof(int)) = edx:
                                                                     void function1(int ecx, int eax ...) {
           eax = eax+1:
                                                                      loop ind = 0:
                                                                                                               Parallel
           if(eax != ecx)
                                                                                                               threads
                                                                         void function2(int ecx. int eax ...) {
            goto bb20:
                                                                          loop ind = (ecx-eax)/2+1:
                 canoncalize
                                                                          upper bound = ecx-eax;
                                                                          bb20:
                                                                           edx = *(memLd1+(edi + (eax+loop ind)*4)/sizeof(int));
loop_ind = 0:
                                                                           edx += *(memLd2 + (esi + (eax+loop ind)*4)/sizeof(int));
upper bound = ecx-eax:
                                                                           *(memSt + (ebx+(eax+loop ind)*4)/sizeof(int)) = edx:
bb20:
                                                                            loop ind = loop ind+1:
  edx = *(memLd1+(edi + (eax+loop ind)*4)/sizeof(int));
                                                                            if(loop ind != upper bound)
  edx += *(memLd2+ (esi + (eax+loop_ind)*4)/sizeof(int));
                                                                             goto bb20:
  *(memSt + (ebx+(eax+loop ind)*4)/sizeof(int)) = edx:
  loop ind = loop ind+1:
  if(loop ind != upper bound)
   goto bb20:
```