```
%10:
                                                %mul.i.i = shl i64 %7, 8
                                                %mul2.i = mul nsw i32 %5, %3
                                                %add3.i = add nsw i32 %mul2.i, %3
                                                %idxprom4.i = sext i32 %add3.i to i64
                                                %arrayidx5.i = getelementptr inbounds float, float* %1, i64 %idxprom4.i
                                                br label %pregion for entry.entry.i
                                                     pregion for entry.entry.i:
                                                      % local id x.0 = phi i64 [0, %10], [%16, %if.end.r exit.i.1]
                                                      %add1.i.i = add nuw nsw i64 % local id x.0, %mul.i.i
                                                      %conv.i = trunc i64 %add1.i.i to i32
                                                      %cmp.i = icmp slt i32 %conv.i, %4
                                                      br i1 %cmp.i, label %if.then.i, label %if.end.r exit.i
                 if.then.i:
                 %mul.i = mul nsw i32 %conv.i, %5
                 %add.i = add nsw i32 %mul.i, %3
                 %idxprom.i = sext i32 %add.i to i64
                 %arrayidx.i = getelementptr inbounds float, float* %0, i64 %idxprom.i
                 %11 = load float, float* %arrayidx.i, align 4, !tbaa !12
                 %12 = load float, float* %arrayidx5.i, align 4, !tbaa !12
                 %div.i = fdiv float %11, %12, !fpmath !16
                 %arrayidx9.i = getelementptr inbounds float, float* %2, i64 %idxprom.i
                 store float %div.i, float* %arrayidx9.i, align 4, !tbaa !12,
                 ...!llvm.access.group!17
                 br label %if.end.r exit.i
                              if.end.r exit.i:
                              %13 = \text{ or } i64 \% \text{ local id } x.0, 1
                              %add1.i.i.1 = add nuw nsw i64 %13, %mul.i.i
                              %conv.i.1 = trunc i64 %add1.i.i.1 to i32
                              %cmp.i.1 = icmp slt i32 %conv.i.1, %4
                              br i1 %cmp.i.1, label %if.then.i.1, label %if.end.r exit.i.1
%mul.i.1 = mul nsw i32 %conv.i.1, %5
%add.i.1 = add nsw i32 %mul.i.1, %3
%idxprom.i.1 = sext i32 %add.i.1 to i64
%arrayidx.i.1 = getelementptr inbounds float, float* %0, i64 %idxprom.i.1
%14 = load float, float* %arrayidx.i.1, align 4, !tbaa !12
%15 = load float, float* %arrayidx5.i, align 4, !tbaa !12
%div.i.1 = fdiv float %14, %15, !fpmath !16
%arravidx9.i.1 = getelementptr inbounds float, float* %2, i64 %idxprom.i.1
store float %div.i.1, float* %arrayidx9.i.1, align 4, !tbaa !12,
...!llvm.access.group!17
br label %if.end.r exit.i.1
                                               if.end.r exit.i.1:
                                                %16 = add nuw nsw i64 \% local id x.0, 2
                                                %exitcond.not.1 = icmp eq i64 %16, 256
                                                br i1 %exitcond.not.1, label %gramschmidt kernel2.exit, label
                                               ... %pregion for entry.entry.i, !llvm.loop !19
                                                                                               F
                                                 gramschmidt kernel2.exit:
                                                 ret void
```

if.then.i.1: