```
%13 = \text{shl } i64 \%9, 5
                                                                                 %14 = \text{shl } i64 \%10, 3
                                                                                 %cmp740.i = icmp sgt i32 \%7, 0
                                                                                 %15 = sext i32 %6 to i64
                                                                                 %wide.trip.count.i = zext i32 %7 to i64
                                                                                 %16 = add nsw i64 %wide.trip.count.i, -1
                                                                                 %xtraiter.i = and i64 %wide.trip.count.i, 3
                                                                                 %17 = icmp ult i64 %16, 3
                                                                                 %unroll iter.i = sub nuw nsw i64 %wide.trip.count.i, %xtraiter.i
                                                                                 %lcmp.mod.i = icmp eq i64 %xtraiter.i, 0
                                                                                 br label %pregion for entry.pregion for init.i
                                                                                 pregion for entry.pregion for init.i:
                                                                                 %_local_id_y.0 = phi i64 [ 0, %12 ], [ %57, %pregion_for_end.i ] %18 = add nuw nsw i64 %_local_id_y.0, %14
                                                                                  \%conv2.i = trunc i64 \%18 to i32
                                                                                  %cmp.i = icmp slt i32 %conv2.i, %5
                                                                                  %mul.i = mul nsw i32 %conv2.i, %6
                                                                                  %mul9.i = mul nsw i32 %conv2.i, %7
                                                                                  %19 = sext i32 %mul9.i to i64
                                                                                 br label %pregion for entry.entry.i
                                                         pregion for entry.entry.i:
                                                          %20 = add nuw nsw i64 %_local_id_x.0, %13
                                                          %conv.i = trunc i64 %20 to i32
                                                          %cmp4.i = icmp slt i32 %conv.i, %6
                                                          %or.cond.i = and i1 %cmp.i, %cmp4.i
                                                          br i1 %or.cond.i, label %if.then.i, label %if.end.i
                                                                                                               F
                             if.then.i:
                              %add.i = add nsw i32 %mul.i, %conv.i
                              %idxprom.i = sext i32 %add.i to i64
                              %arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i %21 = load float, float* %arrayidx.i, align 4, !tbaa !12
                              %mul6.i = fmul float %21, %4
                              store float %mul6.i, float* %arrayidx.i, align 4, !tbaa !12,
                             ...!llvm.access.group!16
                             br i1 %cmp740.i, label %for.body.lr.ph.i, label %if.end.i
                                                                                    F
for.body.lr.ph.i:
%sext.i = shl i64 %20, 32
%22 = ashr exact i64 %sext.i, 32
br i1 %17, label %if.end.loopexit.unr-lcssa.i, label %for.body.i.preheader
                 Τ
                                                        F
                                              for.body.i.preheader:
                                              br label %for.body.i
                  for.bodv.i:
                   %niter.nsub.3.i7 = phi i64 [ %niter.nsub.3.i, %for.body.i ], [
                   ... %unroll iter.i, %for.body.i.preheader ]
                   %indvars.īv.next.3.i4 = phi i64 [ %indvars.iv.next.3.i, %for.body.i ], [ 0,
                   ... %for.body.i.preheader ]
                   \%23 = \text{phi float} [\%47, \% \text{for.body.i}], [\% \text{mul6.i}, \% \text{for.body.i.preheader}]
                   %24 = add nsw i64 %indvars.iv.next.3.i4, %19
                   %arrayidx12.i = getelementptr inbounds float, float* %0, i64 %24
                   %25 = load float, float* %arrayidx12.i, align 4, !tbaa !12
                   %mul13.i = fmul float %25, %3
                   %26 = mul nsw i64 %indvars.iv.next.3.i4, %15
                   %27 = add nsw i64 %26, %22
                   %arrayidx17.i = getelementptr inbounds float, float* %1, i64 %27
                   %28 = load float, float* %arrayidx17.i, align 4, !tbaa !12
                   %29 = tail call float @llvm.fmuladd.f32(float %mul13.i, float %28, float
                   ... %23) #2
                   store float %29, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                   ... !16
                   %indvars.iv.next.i = or i64 %indvars.iv.next.3.i4, 1
                   %30 = add nsw i64 %indvars.iv.next.i, %19
                   %arrayidx12.1.i = getelementptr inbounds float, float* %0, i64 %30
                   %31 = load float, float* %arrayidx12.1.i, align 4, !tbaa !12
                   %mul13.1.i = fmul float %31, %3
                   %32 = mul nsw i64 %indvars.iv.next.i, %15
                   %33 = add nsw i64 %32, %22
                   %arrayidx17.1.i = getelementptr inbounds float, float* %1, i64 %33
                   %34 = load float, float* %arrayidx17.1.i, align 4, !tbaa !12
                   %35 = tail call float @llvm.fmuladd.f32(float %mul13.1.i, float %34, float
                   ... %29) #2
                   store float %35, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                   ... !16
                   %indvars.iv.next.1.i = or i64 %indvars.iv.next.3.i4, 2
                   %36 = add nsw i64 %indvars.iv.next.1.i, %19
                   %arrayidx12.2.i = getelementptr inbounds float, float* %0, i64 %36
                   %37 = load float, float* %arrayidx12.2.i, align 4, !tbaa !12
                   %mul13.2.i = fmul float %37, %3
                   %38 = mul nsw i64 %indvars.iv.next.1.i, %15
                   %39 = add nsw i64 %38, %22
                   %arrayidx17.2.i = getelementptr inbounds float, float* %1, i64 %39
                   %40 = load float, float* %arrayidx17.2.i, align 4, !tbaa !12
                   %41 = tail call float @llvm.fmuladd.f32(float %mul13.2.i, float %40, float
                   ... %35) #2
                   store float %41, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                   ... !16
                   %indvars.iv.next.2.i = or i64 %indvars.iv.next.3.i4, 3
                   %42 = add nsw i64 %indvars.iv.next.2.i, %19
                   %arrayidx12.3.i = getelementptr inbounds float, float* %0, i64 %42
                   %43 = load float, float* %arrayidx12.3.i, align 4, !tbaa !12
                   %mul13.3.i = fmul float %43, %3
                   %44 = mul nsw i64 %indvars.iv.next.2.i, %15
                   %45 = add nsw i64 %44, %22
                   %arrayidx17.3.i = getelementptr inbounds float, float* %1, i64 %45
                   %46 = load float, float* %arrayidx17.3.i, align 4, !tbaa !12
                   %47 = tail call float @llvm.fmuladd.f32(float %mul13.3.i, float %46, float
                   ... %41) #2
                   store float %47, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                   ... !16
                   %indvars.iv.next.3.i = add nuw nsw i64 %indvars.iv.next.3.i4, 4
                   %niter.nsub.3.i = add i64 %niter.nsub.3.i7, -4
                   %niter.ncmp.3.i = icmp eq i64 %niter.nsub.3.i, 0
                   br i1 %niter.ncmp.3.i, label %if.end.loopexit.unr-lcssa.i.loopexit, label
                   ... %for.body.i
         if.end.loopexit.unr-lcssa.i.loopexit:
          %.lcssa = phi float [ %47, %for.body.i ]
          %indvars.iv.next.3.i.lcssa = phi i64 [ %indvars.iv.next.3.i, %for.body.i ]
          br label %if.end.loopexit.unr-lcssa.i
if.end.loopexit.unr-lcssa.i:
%48 = phi float [ %mul6.i, %for.body.lr.ph.i ], [ %.lcssa,
... %if.end.loopexit.unr-lcssa.i.loopexit ]
\%49 = \text{phi } i64 \text{ [ 0, \%for.body.lr.ph.i ], [ \%indvars.iv.next.3.i.lcssa, ]}
... %if.end.loopexit.unr-lcssa.i.loopexit]
br i1 %lcmp.mod.i, label %if.end.i, label %for.body.epil.i.preheader
                                         for.body.epil.i.preheader:
                                          br label %for.body.epil.i
                 for.body.epil.i:
                  %epil.iter.sub.i13 = phi i64 [ %epil.iter.sub.i, %for.body.epil.i ], [
                  ... %xtraiter.i, %for.body.epil.i.preheader ]
                  %indvars.iv.next.epil.i11 = phi i64 [ %indvars.iv.next.epil.i,
                  ... %for.body.epil.i ], [ %49, %for.body.epil.i.preheader ]
                  %50 = phi float [ %56, %for.body.epil.i ], [ %48, %for.body.epil.i.preheader
                  %51 = add nsw i64 %indvars.iv.next.epil.i11, %19
                  %arrayidx12.epil.i = getelementptr inbounds float, float* %0, i64 %51 %52 = load float, float* %arrayidx12.epil.i, align 4, !tbaa !12
                  %mul13.epil.i = fmul float %52, %3
                  %53 = mul nsw i64 %indvars.iv.next.epil.i11, %15
                  %54 = add nsw i64 %53, %22
                  %arrayidx17.epil.i = getelementptr inbounds float, float* %1, i64 %54
                  %55 = load float, float* %arrayidx17.epil.i, align 4, !tbaa !12
                  %56 = tail call float @llvm.fmuladd.f32(float %mul13.epil.i, float %55,
                  .. float %50) #2
                  store float %56, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                  %indvars.iv.next.epil.i = add nuw nsw i64 %indvars.iv.next.epil.i11, 1
                  %epil.iter.sub.i = add nsw i64 %epil.iter.sub.i13, -1
                  %epil.iter.cmp.i = icmp eq i64 %epil.iter.sub.i, 0
                  br i1 %epil.iter.cmp.i, label %if.end.i.loopexit, label %for.body.epil.i,
                  ... !llvm.loop !19
                                                                           F
                                                     if.end.i.loopexit:
                                                      br label %if.end.i
                                                  if.end.i:
                                                  %58 = add nuw nsw i64 \% local id x.0, 1
                                                  %exitcond = icmp eq i64 \%58, 3\overline{2}
                                                  br i1 %exitcond, label %pregion for end.i, label %pregion for entry.entry.i,
                                                  ...!llvm.loop!23
                                                                                                             F
                                                                    Τ
                                                                  pregion for end.i:
                                                                   \%57 = add nuw nsw i64 % local id y.0, 1
                                                                   \%exitcond14 = icmp eq i6\overline{4} %57, 8
                                                                   br i1 %exitcond14, label %gemm.exit, label
                                                                  ... %pregion for entry.pregion for init.i, !llvm.loop !21
                                                                                                            F
```

gemm.exit: ret void

CFG for 'pocl kernel gemm' function

%12: