

entry:  
%post1 = getelementptr inbounds %struct.jpeg\_decompress\_struct,  
... %struct.jpeg\_decompress\_struct\* %cinfo, i64 0, i32 76  
%0 = bitcast %struct.jpeg\_d\_post\_controller\*\* %post1 to  
... %struct.my\_post\_controller\*\*  
%1 = load %struct.my\_post\_controller\*, %struct.my\_post\_controller\*\* %0,  
... align 8, !tbaa !2  
%next\_row = getelementptr inbounds %struct.my\_post\_controller,  
... %struct.my\_post\_controller\* %1, i64 0, i32 5  
%2 = load i32, i32\* %next\_row, align 8, !tbaa !10  
%cmp = icmp eq i32 %2, 0  
br i1 %cmp, label %if.then, label %entry.if.end\_crit\_edge

T	F
---	---

if.then:  
%mem = getelementptr inbounds %struct.jpeg\_decompress\_struct,  
... %struct.jpeg\_decompress\_struct\* %cinfo, i64 0, i32 1  
%3 = load %struct.jpeg\_memory\_mgr\*, %struct.jpeg\_memory\_mgr\*\* %mem, align 8,  
... !tbaa !14  
%access\_virt\_sarray = getelementptr inbounds %struct.jpeg\_memory\_mgr,  
... %struct.jpeg\_memory\_mgr\* %3, i64 0, i32 7  
%4 = load i8\*\* (%struct.jpeg\_common\_struct\*, %struct.jvirt\_sarray\_control\*,  
... i32, i32, i32)\*, i8\*\* (%struct.jpeg\_common\_struct\*,  
... %struct.jvirt\_sarray\_control\*, i32, i32, i32)\*\* %access\_virt\_sarray, align 8,  
... !tbaa !15  
%5 = bitcast %struct.jpeg\_decompress\_struct\* %cinfo to  
... %struct.jpeg\_common\_struct\*  
%whole\_image = getelementptr inbounds %struct.my\_post\_controller,  
... %struct.my\_post\_controller\* %1, i64 0, i32 1  
%6 = load %struct.jvirt\_sarray\_control\*, %struct.jvirt\_sarray\_control\*\*  
... %whole\_image, align 8, !tbaa !18  
%starting\_row = getelementptr inbounds %struct.my\_post\_controller,  
... %struct.my\_post\_controller\* %1, i64 0, i32 4  
%7 = load i32, i32\* %starting\_row, align 4, !tbaa !19  
%strip\_height = getelementptr inbounds %struct.my\_post\_controller,  
... %struct.my\_post\_controller\* %1, i64 0, i32 3  
%8 = load i32, i32\* %strip\_height, align 8, !tbaa !20  
%call = tail call i8\*\* %4(%struct.jpeg\_common\_struct\* %5,  
... %struct.jvirt\_sarray\_control\* %6, i32 %7, i32 %8, i32 1) #3  
%buffer = getelementptr inbounds %struct.my\_post\_controller,  
... %struct.my\_post\_controller\* %1, i64 0, i32 2  
store i8\*\* %call, i8\*\*\* %buffer, align 8, !tbaa !13  
%.pre = load i32, i32\* %next\_row, align 8, !tbaa !10  
br label %if.end

entry.if.end\_crit\_edge:  
%buffer4.phi.trans.insert = getelementptr inbounds  
... %struct.my\_post\_controller, %struct.my\_post\_controller\* %1, i64 0, i32 2  
%.pre50 = load i8\*\*, i8\*\*\* %buffer4.phi.trans.insert, align 8, !tbaa !13  
%.pre52 = getelementptr inbounds %struct.my\_post\_controller,  
... %struct.my\_post\_controller\* %1, i64 0, i32 3  
br label %if.end

if.end:  
%strip\_height6.pre-phi = phi i32\* [ %.pre52, %entry.if.end\_crit\_edge ], [  
... %strip\_height, %if.then ]  
%buffer4.pre-phi = phi i8\*\*\* [ %buffer4.phi.trans.insert,  
... %entry.if.end\_crit\_edge ], [ %buffer, %if.then ]  
%9 = phi i8\*\* [ %.pre50, %entry.if.end\_crit\_edge ], [ %call, %if.then ]  
%10 = phi i32 [ %2, %entry.if.end\_crit\_edge ], [ %.pre, %if.then ]  
%upsample = getelementptr inbounds %struct.jpeg\_decompress\_struct,  
... %struct.jpeg\_decompress\_struct\* %cinfo, i64 0, i32 81  
%11 = load %struct.jpeg\_upsampler\*, %struct.jpeg\_upsampler\*\* %upsample,  
... align 8, !tbaa !21  
%upsample3 = getelementptr inbounds %struct.jpeg\_upsampler,  
... %struct.jpeg\_upsampler\* %11, i64 0, i32 1  
%12 = load void (%struct.jpeg\_decompress\_struct\*, i8\*\*\*, i32\*, i32, i8\*\*,  
... i32\*, i32)\*, void (%struct.jpeg\_decompress\_struct\*, i8\*\*\*, i32\*, i32, i8\*\*,  
... i32\*, i32)\*\* %upsample3, align 8, !tbaa !22  
%13 = load i32, i32\* %strip\_height6.pre-phi, align 8, !tbaa !20  
tail call void %12(%struct.jpeg\_decompress\_struct\* nonnull %cinfo, i8\*\*\*  
... %input\_buf, i32\* %in\_row\_group\_ctr, i32 %in\_row\_groups\_avail, i8\*\* %9, i32\*  
... nonnull %next\_row, i32 %13) #3  
%14 = load i32, i32\* %next\_row, align 8, !tbaa !10  
%cmp8 = icmp ugt i32 %14, %10  
br i1 %cmp8, label %if.then9, label %if.end12

T	F
---	---

if.then9:  
%sub = sub i32 %14, %10  
%cquantize = getelementptr inbounds %struct.jpeg\_decompress\_struct,  
... %struct.jpeg\_decompress\_struct\* %cinfo, i64 0, i32 83  
%15 = load %struct.jpeg\_color\_quantizer\*, %struct.jpeg\_color\_quantizer\*\*  
... %cquantize, align 8, !tbaa !24  
%color\_quantize = getelementptr inbounds %struct.jpeg\_color\_quantizer,  
... %struct.jpeg\_color\_quantizer\* %15, i64 0, i32 1  
%16 = load void (%struct.jpeg\_decompress\_struct\*, i8\*\*, i8\*\*, i32)\*, void  
... (%struct.jpeg\_decompress\_struct\*, i8\*\*, i8\*\*, i32)\*\* %color\_quantize, align  
... 8, !tbaa !25  
%17 = load i8\*\*, i8\*\*\* %buffer4.pre-phi, align 8, !tbaa !13  
%idx.ext = zext i32 %10 to i64  
%add.ptr = getelementptr inbounds i8\*, i8\*\* %17, i64 %idx.ext  
tail call void %16(%struct.jpeg\_decompress\_struct\* nonnull %cinfo, i8\*\*  
... %add.ptr, i8\*\* null, i32 %sub) #3  
%18 = load i32, i32\* %out\_row\_ctr, align 4, !tbaa !27  
%add = add i32 %18, %sub  
store i32 %add, i32\* %out\_row\_ctr, align 4, !tbaa !27  
%.pre51 = load i32, i32\* %next\_row, align 8, !tbaa !10  
br label %if.end12

if.end12:  
%19 = phi i32 [ %.pre51, %if.then9 ], [ %14, %if.end ]  
%20 = bitcast i32\* %strip\_height6.pre-phi to i64\*  
%21 = load i64, i64\* %20, align 8  
%22 = trunc i64 %21 to i32  
%cmp15 = icmp ult i32 %19, %22  
br i1 %cmp15, label %if.end21, label %if.then16

T	F
---	---

if.then16:  
%23 = lshr i64 %21, 32  
%24 = trunc i64 %23 to i32  
%starting\_row18 = getelementptr inbounds %struct.my\_post\_controller,  
... %struct.my\_post\_controller\* %1, i64 0, i32 4  
%add19 = add i32 %24, %22  
store i32 %add19, i32\* %starting\_row18, align 4, !tbaa !19  
store i32 0, i32\* %next\_row, align 8, !tbaa !10  
br label %if.end21

if.end21:  
ret void

CFG for 'post\_process\_prepass' function