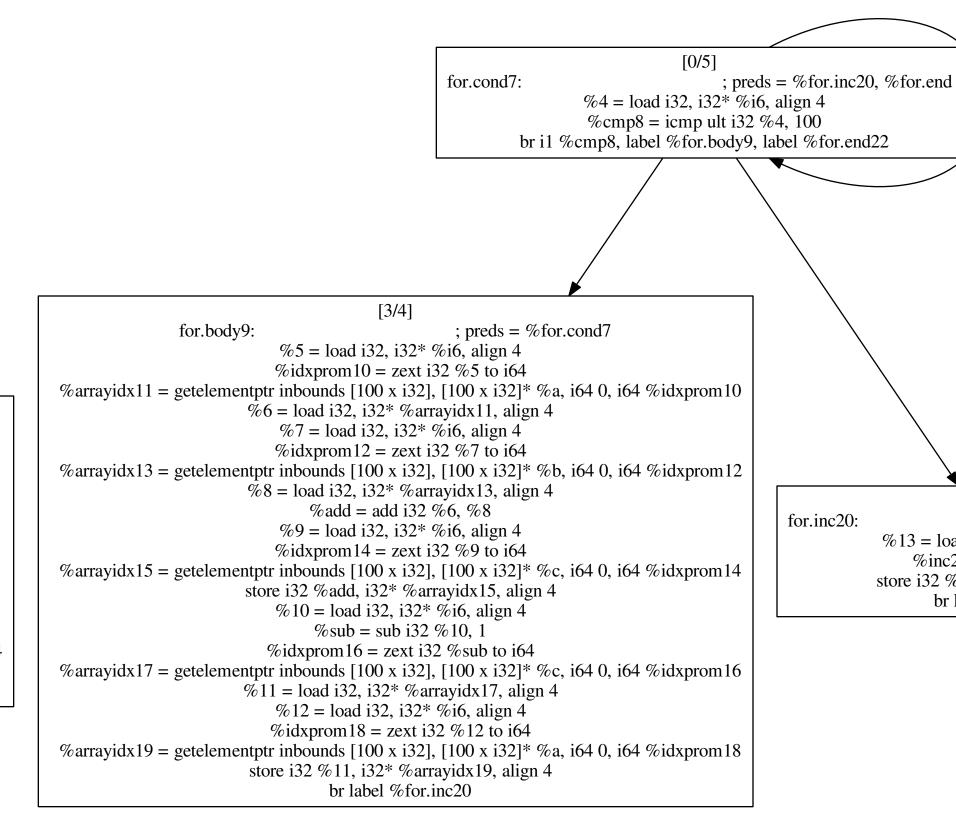
[8/11] [14/15] ; preds = %for.inc, %entry [16/17] for.cond: for.end: preds = % for .cond ; preds = % for.cond7 for.end22: %0 = load i 32, i 32*% i, align 4store i32 1, i32* %i6, align 4 ret i32 0 %cmp = icmp ult i32 %0, 100 br label %for.cond7 br i1 %cmp, label %for.body, label %for.end for.body: ; preds = %for.cond %call1 = call i32 @rand() #2 %rem = urem i32 %call1, 10 %1 = load i32, i32* %i, align 4[6/7] %idxprom = zext i32 %1 to i64 ; preds = % for.bodyfor.inc: %arrayidx = getelementptr inbounds [100 x i32], [100 x i32]* %a, i64 0, i64 %idxprom %3 = load i32, i32* %i, align 4store i32 %rem, i32* %arrayidx, align 4 %inc = add i32 %3, 1 %call2 = call i32 @rand() #2 store i32 %inc, i32* %i, align 4 %rem3 = urem i32 %call2, 10 br label %for.cond %2 = load i 32, i 32*%i, align 4%idxprom4 = zext i32 %2 to i64 %arravidx5 = getelementptr inbounds [100 x i32], [100 x i32]* %b, i64 0, i64 %idxprom4 store i32 %rem3, i32* %arrayidx5, align 4

br label %for.inc



[12/13]
entry:
%retval = alloca i32, align 4
%a = alloca [100 x i32], align 16
%b = alloca [100 x i32], align 16
%c = alloca [100 x i32], align 16
%i = alloca i32, align 4
%i6 = alloca i32, align 4
store i32 0, i32* %retval, align 4
%call = call i64 @time(i64* null) #2
%conv = trunc i64 %call to i32
call void @srand(i32 %conv) #2
store i32 0, i32* %i, align 4
br label %for.cond

; preds = % for.body9

[1/2]

%13 = load i32, i32* %i6, align 4

%inc21 = add i32 %13, 1

store i32 %inc21, i32* %i6, align 4

br label %for.cond7