entry: %put\_bits1.i = getelementptr inbounds %struct.phuff\_entropy\_encoder, ... %struct.phuff\_entropy\_encoder\* %entropy, i64 0, i32 5 %gather\_statistics.i = getelementptr inbounds %struct.phuff\_entropy\_encoder, ... %struct.phuff\_entropy\_encoder\* %entropy, i64 0, i32 1 %0 = load i32, i32\* %gather\_statistics.i, align 8, !tbaa !3 %tobool.i = icmp eq i32 %0, 0 br i1 %tobool.i, label %if.end7.i, label %entry.emit\_bits.exit\_crit\_edge, .. !prof !11 F T if.end7.i: %1 = load i32, i32\* %put\_bits1.i, align 8, !tbaa !12 %add.i = add i32 %1, 7 %sub8.i = sub i32 17, %1 %sh\_prom9.i = zext i32 %sub8.i to i64 %shl10.i = shl i64 127, %sh\_prom9.i %put\_buffer11.i = getelementptr inbounds %struct.phuff\_entropy\_encoder, ... %struct.phuff\_entropy\_encoder\* %entropy, i64 0, i32 4 %2 = load i64, i64\* %put\_buffer11.i, align 8, !tbaa !13 %or.i = or i64 %shl10.i, %2 %cmp1267.i = icmp sgt i32 %add.i, 7 br i1 %cmp1267.i, label %while.body.lr.ph.i, label %while.end.i, !prof !14 while.body.lr.ph.i: %next\_output\_byte.i = getelementptr inbounds %struct.phuff\_entropy\_encoder, ... %struct.phuff\_entropy\_encoder\* %entropy, i64 0, i32 2 %free\_in\_buffer.i = getelementptr inbounds %struct.phuff\_entropy\_encoder, entry.emit\_bits.exit\_crit\_edge: %.pre = getelementptr inbounds %struct.phuff\_entropy\_encoder, ... %struct.phuff\_entropy\_encoder\* %entropy, i64 0, i32 3 %cinfo.i5 = getelementptr inbounds %struct.phuff\_entropy\_encoder, .. %struct.phuff\_entropy\_encoder\* %entropy, i64 0, i32 4 ... %struct.phuff\_entropy\_encoder\* %entropy, i64 0, i32 6 br label %emit\_bits.exit  $%3 = bitcast i8** %next_output_byte.i to <2 x i64>*$ %4 = bitcast i8\*\* %next\_output\_byte.i to <2 x i64>\* br label %while.body.i while.body.i: %put\_buffer.069.i = phi i64 [ %or.i, %while.body.lr.ph.i ], [ %shl33.i, ... %if.end32.i ] %put\_bits.068.i = phi i32 [ %add.i, %while.body.lr.ph.i ], [ %sub34.i, .. %if.end32.i ] %shr66.i = lshr i64 %put\_buffer.069.i, 16 %conv16.i = trunc i64 %shr66.i to i8  $\%5 = \text{load i8*}, \text{i8**} \% \text{next\_output\_byte.i}, \text{align 8, !tbaa !15}$ %incdec.ptr.i = getelementptr inbounds i8, i8\* %5, i64 1 store i8\* %incdec.ptr.i, i8\*\* %next\_output\_byte.i, align 8, !tbaa !15 store i8 %conv16.i, i8\* %5, align 1, !tbaa !16 %6 = load i64, i64\* %free\_in\_buffer.i, align 8, !tbaa !17 % dec.i = add i64 %6, -1store i64 %dec.i, i64\* %free\_in\_buffer.i, align 8, !tbaa !17 %cmp17.i = icmp eq i64 %dec.i, 0 br i1 %cmp17.i, label %if.then19.i, label %if.end20.i, !prof!18 if.end20.i: %conv15.i = and i64 %shr66.i. 255 %cmp21.i = icmp eq i64 %conv15.i, 255 br i1 %cmp21.i, label %if.then23.i, label %if.end32.i, !prof!31 if.end32.i: %shl33.i = shl i64 %put\_buffer.069.i, 8 %sub34.i = add nsw i32 %put\_bits.068.i, -8 %cmp12.i = icmp sgt i32 %sub34.i, 7 br i1 %cmp12.i, label %while.body.i, label %while.end.loopexit.i, !prof!14 while.end.loopexit.i: %shl33.i.lcssa = phi i64 [ %shl33.i, %if.end32.i ] %27 = and i32 %add.i, 7 br label %while.end.i while.end.i: %put\_buffer.0.lcssa.i = phi i64 [ %or.i, %if.end7.i ], [ %shl33.i.lcssa, ... %while.end.loopexit.i ] %put\_bits.0.lcssa.i = phi i32 [ %add.i, %if.end7.i ], [ %27, ... %while.end.loopexit.i ] store i64 %put\_buffer.0.lcssa.i, i64\* %put\_buffer11.i, align 8, !tbaa !13 store i32 %put bits.0.lcssa.i, i32\* %put bits1.i, align 8, !tbaa !12 br label %emit bits.exit emit bits.exit: %put\_buffer.pre-phi = phi i64\* [ %.pre, %entry.emit\_bits.exit\_crit\_edge ], [ ... %put buffer11.i, %while.end.i

store i64 0, i64\* %put\_buffer.pre-phi, align 8, !tbaa !13

store i32 0, i32\* %put\_bits1.i, align 8, !tbaa !12