

Georgios Zacharopoulos

Laura Pozzi

ClrFreqCFGPrinter :

A TOOL FOR FREQUENCY-ANNOTATED CONTROL FLOW GRAPH GENERATION

Università della Svizzera italiana (USI Lugano),
Faculty of Informatics

EuroLLVM 2017

Mar 28, 2017
Saarbrücken,
Germany

PROFILING

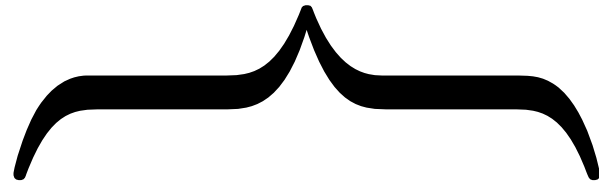
LLVM **profdata**

LLVM **Analysis Pass**

PROFILING

LLVM **profdata**

LLVM **Analysis Pass**



LLVM-PROFDATA TOOL

PROFILING

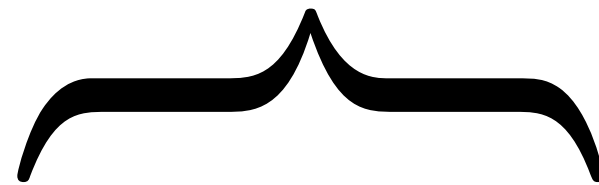
LLVM **profdata**

LLVM **Analysis Pass**

PROFILING

LLVM **profdata**

LLVM **Analysis Pass**



BB FREQUENCY INFO ANALYSIS METHODS

PROFILING

LLVM **profdata**

LLVM **Analysis Pass**

PROFILING

LLVM **profdata**

LLVM **Analysis Pass**



PROFILING

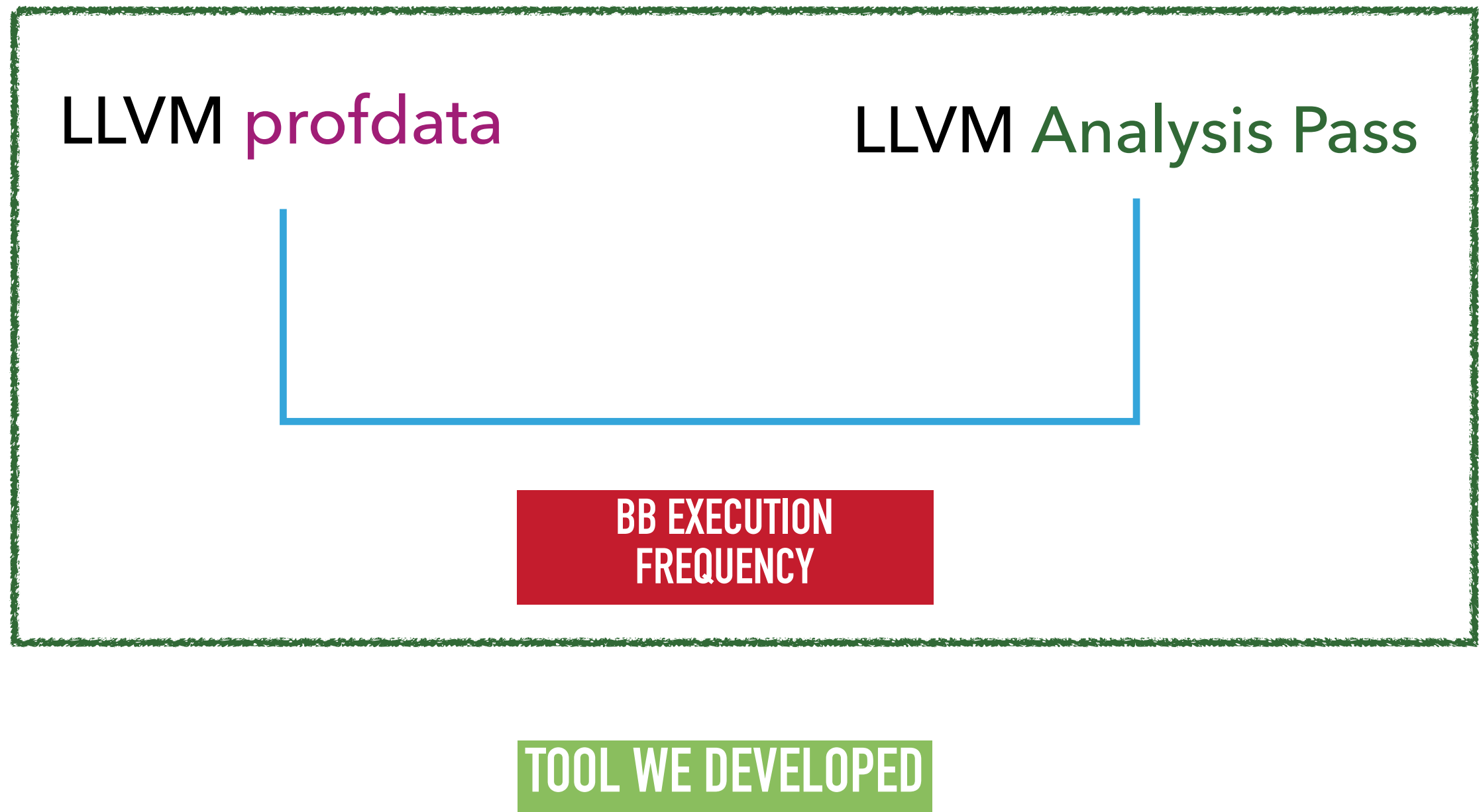
LLVM **profdata**

LLVM **Analysis Pass**

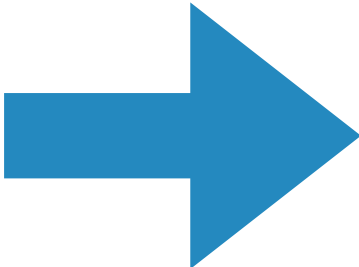
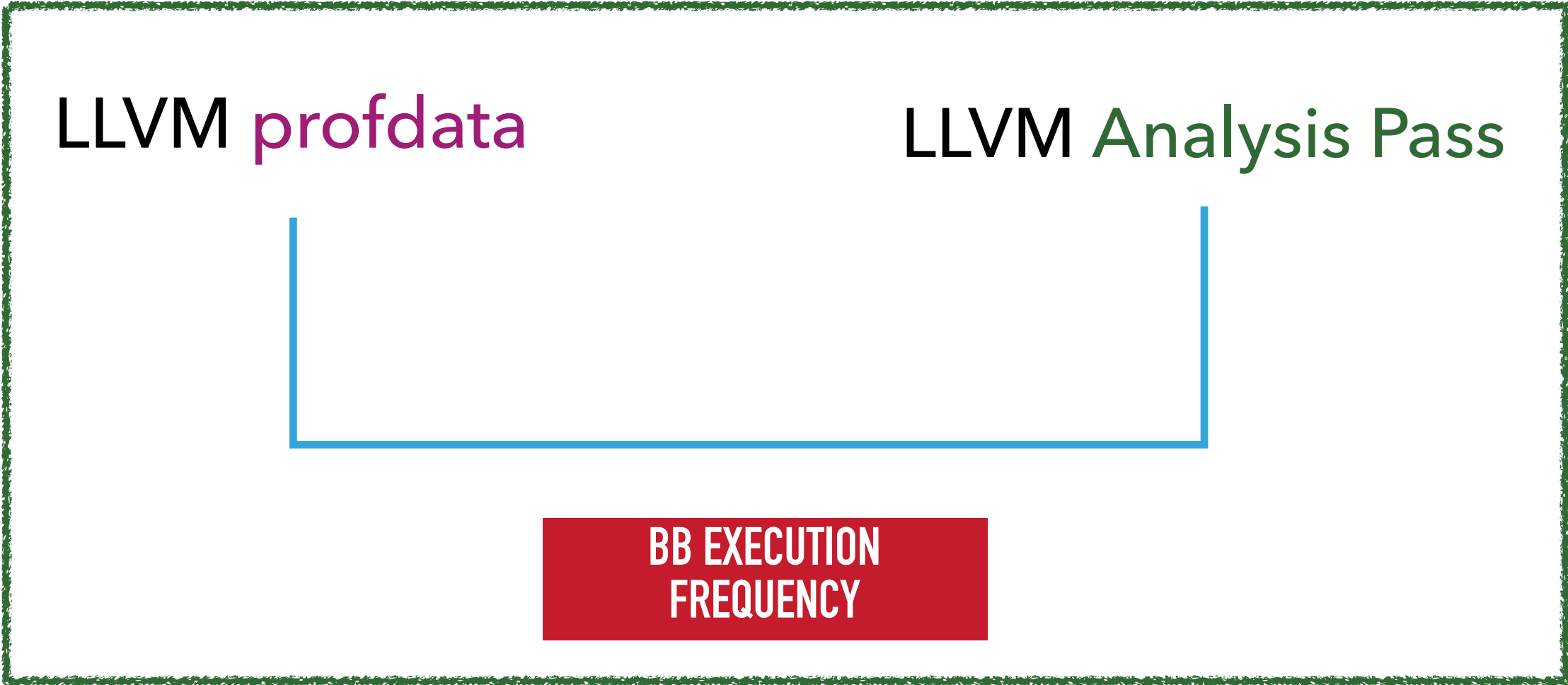


**BB EXECUTION
FREQUENCY**

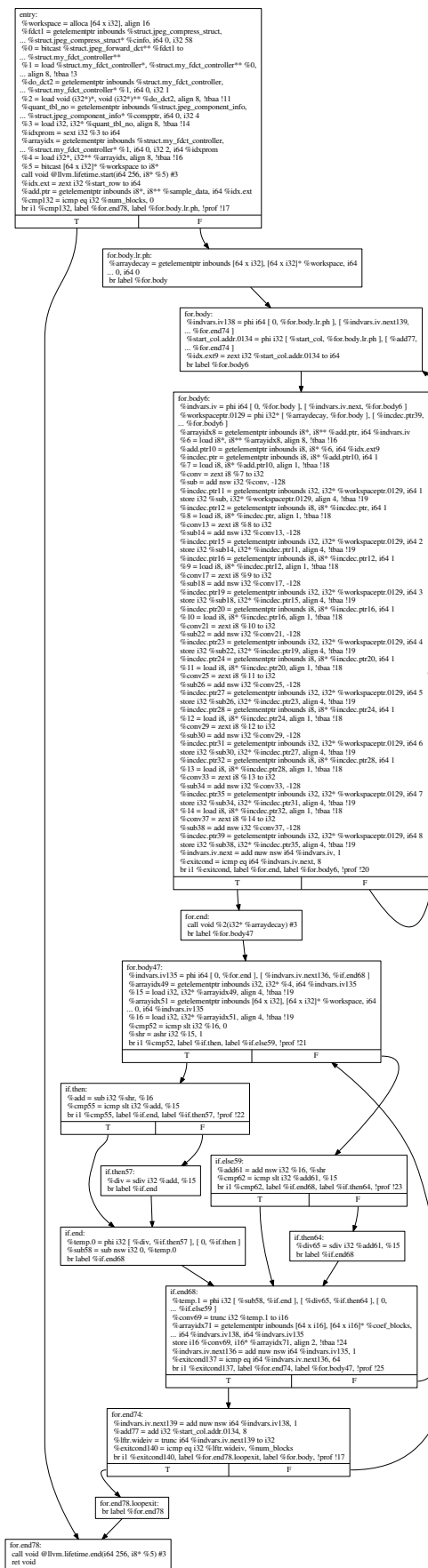
PROFILING



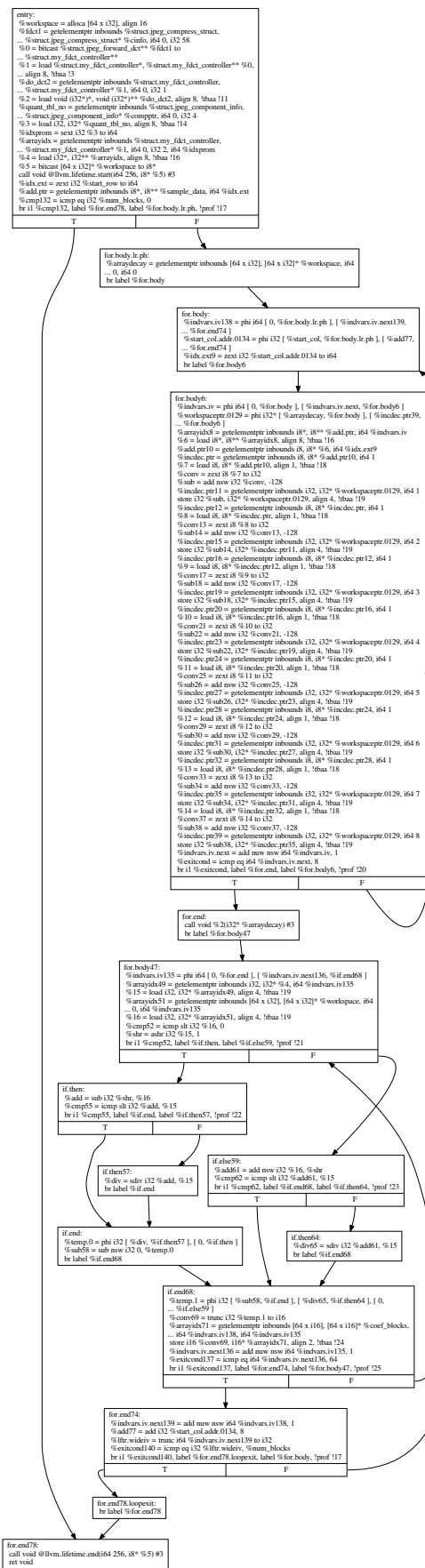
PROFILING



COLOUR DISTRIBUTION
IN THE CFG

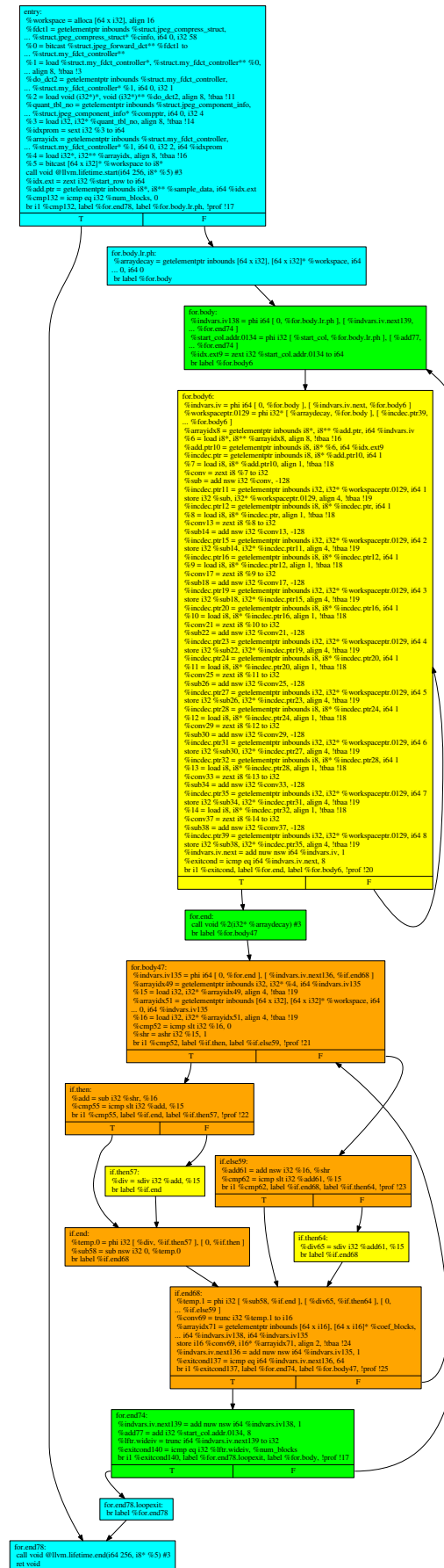


CFG for 'forwrd_DCT' function

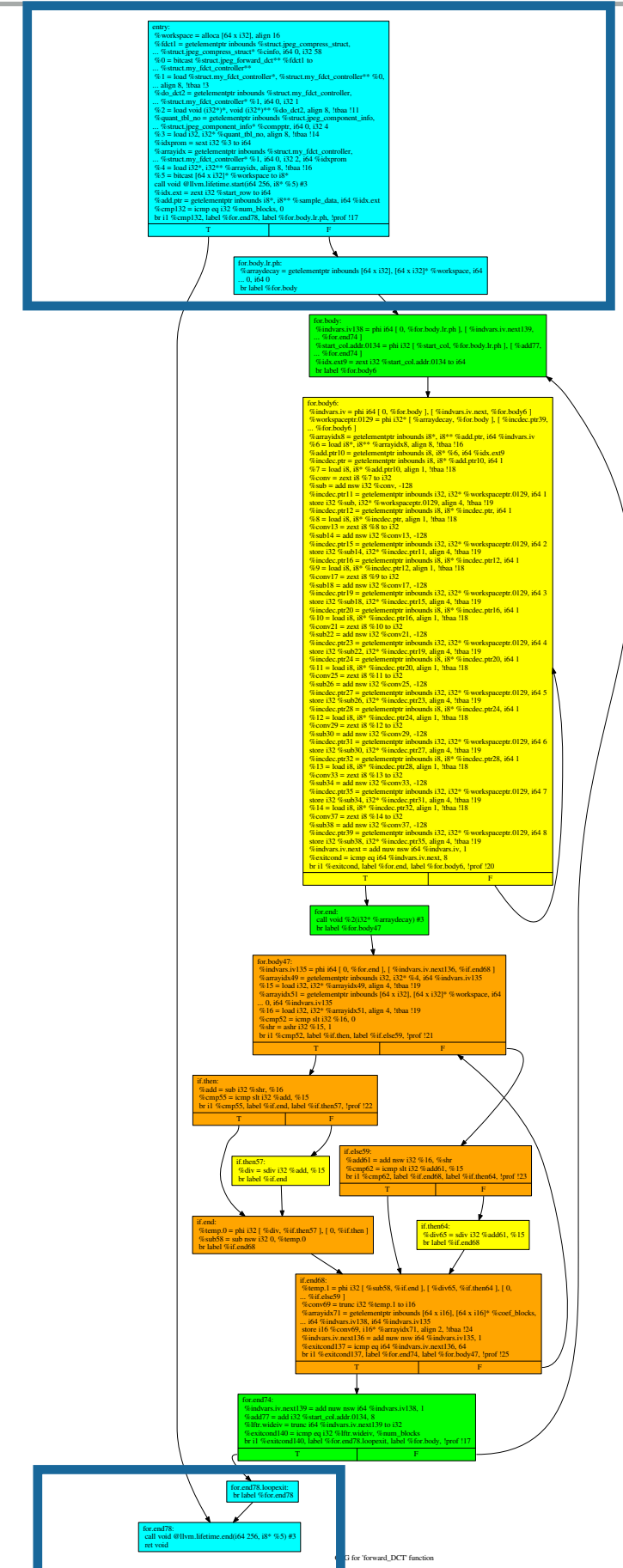




Coldest Regions of an application

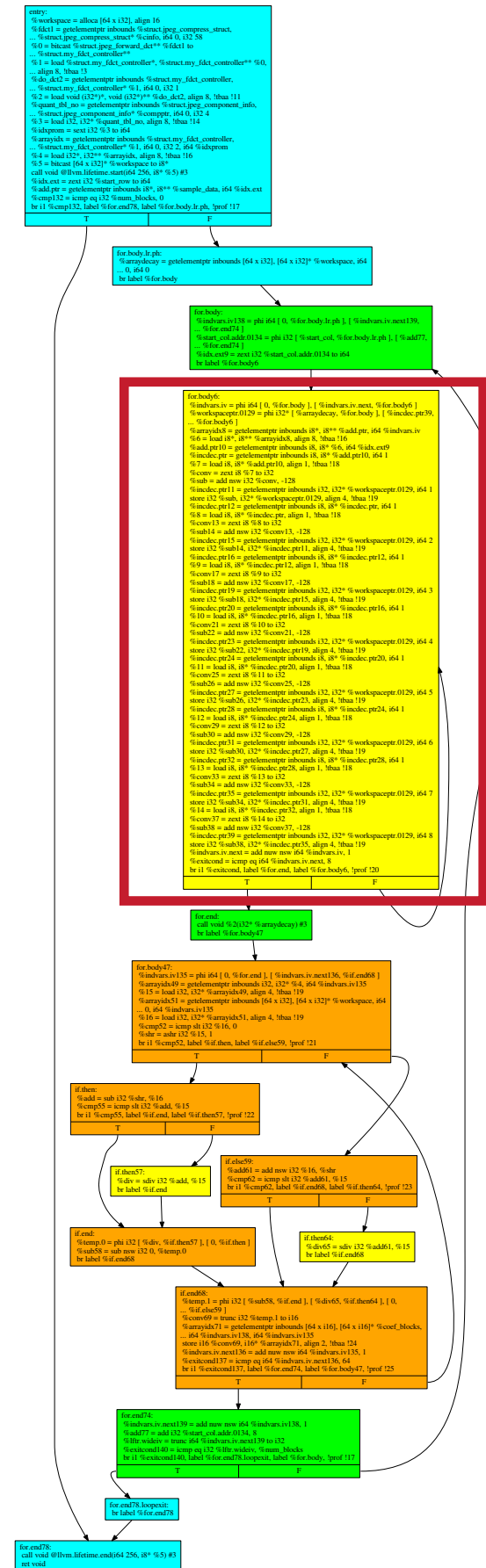


Coldest Regions of an application



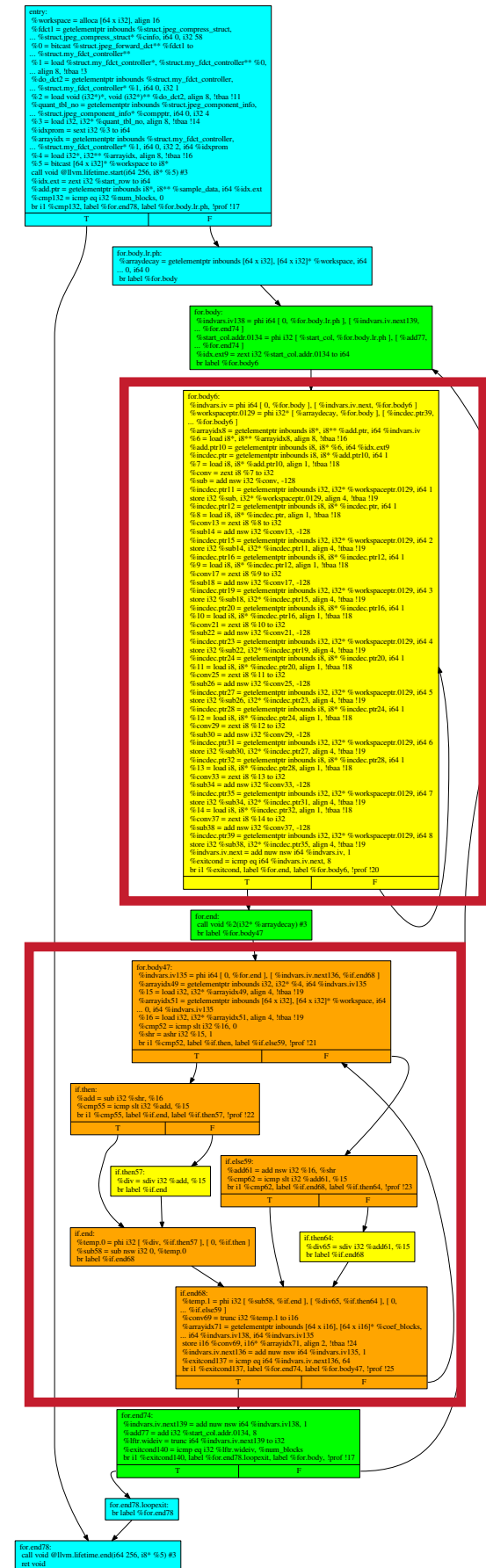
Coldest Regions of an application

Hottest Regions of an application



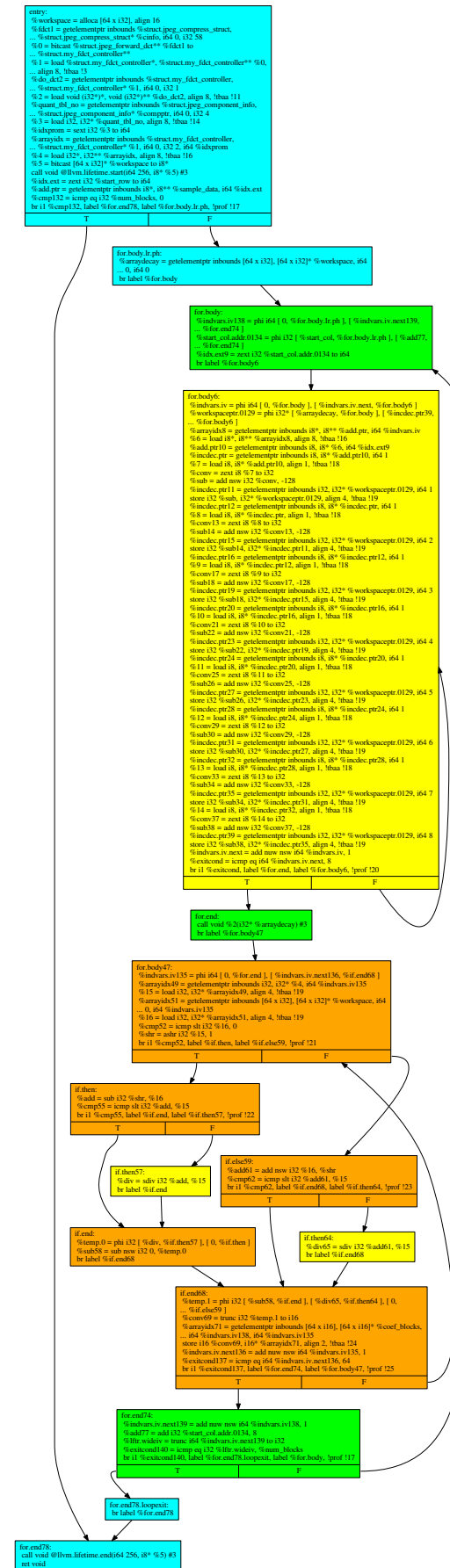
Coldest Regions of an application

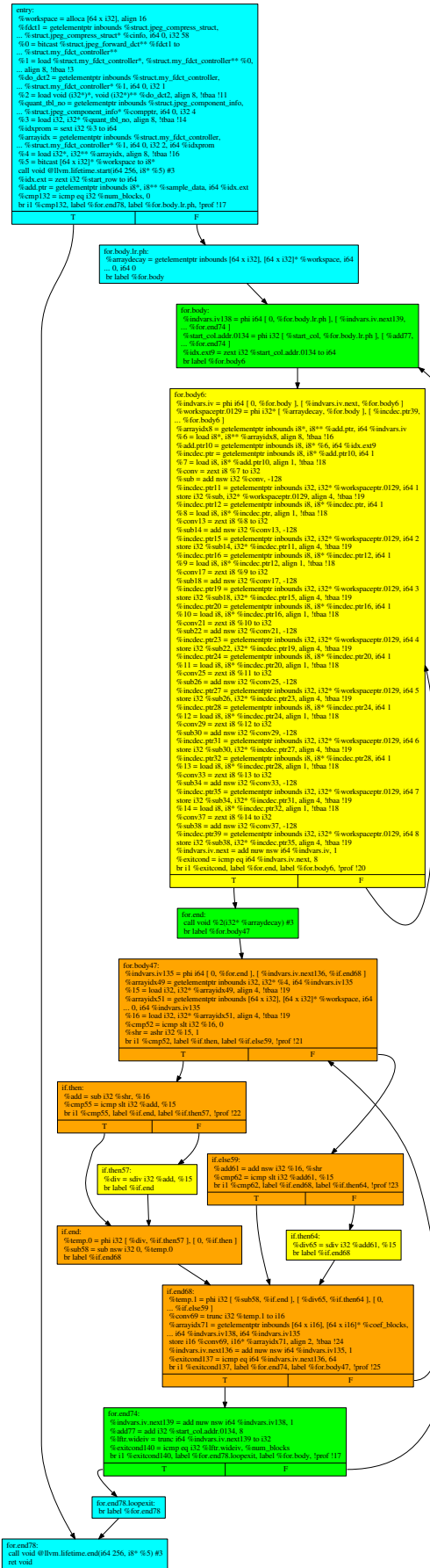
Hottest Regions of an application



Coldest Regions of an application

Hottest Regions of an application





```

entry:
%workspace = alloca [64 x i32], align 16
%idx1 = getelementptr inbounds %struct.jpeg_compress_struct,
... %struct.jpeg_compress_struct* %cinfo, #0, i32 58
%0 = bitcast %struct.jpeg_forwrd_dct** %idx1 to
... %struct.my_fct_controller**
%1 = load %struct.my_fct_controller*, %struct.my_fct_controller** %0,
align 8, !tbaa 13
%idx2 = getelementptr inbounds %struct.my_fct_controller,
... %struct.my_fct_controller* %1, #0, i32 1
%2 = load void (i32*)*, void (i32*)* %idx2, align 8, !tbaa 111
%quant_8d_no = getelementptr inbounds %struct.jpeg_component_info,
... %struct.jpeg_component_info* %cinfo, #0, i32 4
%3 = load i32, i32* %quant_8d_no, align 8, !tbaa 114
%idxprom = sext i32 %3 to i64
%arrayidx = getelementptr inbounds %struct.my_fct_controller,
... %struct.my_fct_controller* %1, #0, i32 2, #0 %idxprom
%4 = load i32*, i32** %arrayidx, align 8, !tbaa 116
%5 = bitcast [64 x i32]* %workspace to i8*
call void @llvm.lifetime.start@64 256, i8* %5 #3
%idxext = sext i32 %start_row to i64
%add_ptr = getelementptr inbounds i8*, i8** %sample_data, #0 %idxext
%cmp132 = icmp eq i32 %num_blocks, 0
br i1 %cmp132, label %for_end78, label %for_body_1r_ph, !prof 117

```

```

for_body_1r_ph:
%arraydecay = getelementptr inbounds [64 x i32], [64 x i32]* %workspace, #0,
... 0, #0
br label %for_body

```

```

for_body:
%indvars.iv18 = phi i64 [ 0, %for_body ], [ %indvars.iv.next139,
... %for_end74 ]
%start_row_addr_0134 = phi i32 [ %start_row, %for_body ], [ %add77,
... %for_end74 ]
%idxext = sext i32 %start_row_addr_0134 to i64
br label %for_body6

```

```

for_body6:
%indvars.iv = phi i64 [ 0, %for_body ], [ %indvars.iv.next, %for_body6 ]
%workspaceptr_0129 = phi i32* [ %arraydecay, %for_body ], [ %indec.ptr39,
... %for_body ]
%arrayidx8 = getelementptr inbounds i8*, i8** %add_ptr, #0 %indvars.iv
%4 = load i8*, i8** %arrayidx8, align 8, !tbaa 116
%add_ptr10 = getelementptr inbounds i8*, i8** %4, #0 %idxext9
%indec.ptr = getelementptr inbounds i8*, i8** %add_ptr10, #0, i32 1
%7 = load i8, i8* %add_ptr10, align 1, !tbaa 118
%conv = sext i8 %7 to i32
%sub = add nsw i32 %conv, -128
%indec.ptr11 = getelementptr inbounds i32, i32* %workspaceptr_0129, #0, i32 1
store i32 %sub, i32* %workspaceptr_0129, align 4, !tbaa 119
%indec.ptr12 = getelementptr inbounds i8, i8* %indec.ptr, #0, i32 1
%8 = load i8, i8* %indec.ptr, align 1, !tbaa 118
%sub14 = add nsw i32 %conv, -128
%indec.ptr15 = getelementptr inbounds i32, i32* %workspaceptr_0129, #0, i32 2
store i32 %sub14, i32* %indec.ptr15, align 4, !tbaa 119
%indec.ptr16 = getelementptr inbounds i8, i8* %indec.ptr12, #0, i32 1
%9 = load i8, i8* %indec.ptr12, align 1, !tbaa 118
%conv17 = sext i8 %9 to i32
%sub18 = add nsw i32 %conv17, -128
%indec.ptr19 = getelementptr inbounds i32, i32* %workspaceptr_0129, #0, i32 3
store i32 %sub18, i32* %indec.ptr19, align 4, !tbaa 119
%indec.ptr20 = getelementptr inbounds i8, i8* %indec.ptr16, #0, i32 1
%10 = load i8, i8* %indec.ptr16, align 1, !tbaa 118
%conv21 = sext i8 %10 to i32
%sub22 = add nsw i32 %conv21, -128
%indec.ptr23 = getelementptr inbounds i32, i32* %workspaceptr_0129, #0, i32 4
store i32 %sub22, i32* %indec.ptr23, align 4, !tbaa 119
%indec.ptr24 = getelementptr inbounds i8, i8* %indec.ptr20, #0, i32 1
%11 = load i8, i8* %indec.ptr20, align 1, !tbaa 118
%conv25 = sext i8 %11 to i32
%sub26 = add nsw i32 %conv25, -128
%indec.ptr27 = getelementptr inbounds i32, i32* %workspaceptr_0129, #0, i32 5
store i32 %sub26, i32* %indec.ptr27, align 4, !tbaa 119
%indec.ptr28 = getelementptr inbounds i8, i8* %indec.ptr24, #0, i32 1
%12 = load i8, i8* %indec.ptr24, align 1, !tbaa 118
%conv29 = sext i8 %12 to i32
%sub30 = add nsw i32 %conv29, -128
%indec.ptr31 = getelementptr inbounds i32, i32* %workspaceptr_0129, #0, i32 6
store i32 %sub30, i32* %indec.ptr31, align 4, !tbaa 119
%indec.ptr32 = getelementptr inbounds i8, i8* %indec.ptr28, #0, i32 1
%13 = load i8, i8* %indec.ptr28, align 1, !tbaa 118
%conv33 = sext i8 %13 to i32
%sub34 = add nsw i32 %conv33, -128
%indec.ptr35 = getelementptr inbounds i32, i32* %workspaceptr_0129, #0, i32 7
store i32 %sub34, i32* %indec.ptr35, align 4, !tbaa 119
%indec.ptr39 = getelementptr inbounds i32, i32* %workspaceptr_0129, #0, i32 8
store i32 %sub34, i32* %indec.ptr39, align 4, !tbaa 119
%indvars.iv.next = add nsw i64 %indvars.iv, 1
%idxcond = icmp eq i64 %indvars.iv.next, 8
br i1 %idxcond, label %for_end, label %for_body6, !prof 120

```

```

for_end:
call void @20(i32* %arraydecay) #3
br label %for_body47

```

```

for_body47:
%indvars.iv15 = phi i64 [ 0, %for_end ], [ %indvars.iv.next136, %if_end68 ]
%arrayidx9 = getelementptr inbounds i32, i32* %4, #0 %indvars.iv15
%15 = load i32, i32* %arrayidx9, align 4, !tbaa 119
%arrayidx1 = getelementptr inbounds [64 x i32], [64 x i32]* %workspace, #0,
... 0, #0 %indvars.iv15
%16 = load i32, i32* %arrayidx1, align 4, !tbaa 119
%cmp52 = icmp slt i32 %16, 0
%shr = ashr i32 %15, 1
br i1 %cmp52, label %if_then, label %if_else59, !prof 121

```

```

if_then:
%add = add i32 %shr, %16
%cmp55 = icmp slt i32 %add, %15
br i1 %cmp55, label %if_end, label %if_then57, !prof 122

```

```

if_then57:
%shr = ashr i32 %add, %15
br label %if_end

```

```

if_else59:
%addb1 = add nsw i32 %16, %shr
%cmp62 = icmp slt i32 %addb1, %15
br i1 %cmp62, label %if_endb8, label %if_thenb4, !prof 123

```

```

if_endb8:
%div5 = ashr i32 %addb1, %15
br label %if_endb8

```

```

if_endb4:
%temp.1 = phi i32 [ %sub58, %if_end ], [ %div5, %if_thenb4 ], [ 0,
... %if_else69 ]
%conv69 = trunc i32 %temp.1 to i16
%arrayidx71 = getelementptr inbounds [64 x i16], [64 x i16]* %conv_blocks,
... #0 %indvars.iv18, #0 %indvars.iv139
store i16 %conv69, i16* %arrayidx71, align 2, !tbaa 124
%indvars.iv.next136 = add nsw i64 %indvars.iv135, 1
%idxcond137 = icmp eq i64 %indvars.iv.next136, 64
br i1 %idxcond137, label %for_end74, label %for_body47, !prof 125

```

```

for_end74:
%indvars.iv.next139 = add nsw i64 %indvars.iv136, 1
%add77 = add i32 %start_row_addr_0134, 1
%idxext = sext i32 %start_row to i64
%add_ptr = getelementptr inbounds i8*, i8** %sample_data, #0 %idxext
%conv13 = icmp eq i32 %num_blocks, %num_blocks
br i1 %conv13, label %for_end78, label %for_body, !prof 117

```

```

for_end78:
call void @llvm.lifetime.end@64 256, i8* %5 #3
ret void

```

CFG for 'forward_dct' function

```

entry:
%mem1 = getelementptr inbounds %struct.jpeg_common_struct,
... %struct.jpeg_common_struct* %cinfo, #0, i32 1
%0 = bitcast %struct.jpeg_memory_mgr** %mem1 to %struct.my_memory_mgr**
%1 = load %struct.my_memory_mgr*, %struct.my_memory_mgr** %0, align 8, !tbaa
... 13
%conv = sext i32 %samplesperrow to i64
%div = udiv i64 999999976, %conv
%cmp = icmp eq i64 %div, 0
br i1 %cmp, label %if_then, label %if_end, !prof 19

```

```

if_then:
%ext = getelementptr inbounds %struct.jpeg_common_struct,
... %struct.jpeg_common_struct* %cinfo, #0, i32 0
%2 = load %struct.jpeg_error_mgr*, %struct.jpeg_error_mgr** %err, align 8,
... !tbaa 110
%msg_code = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %2, #0, i32 3
store i32 %2, i32* %msg_code, align 8, !tbaa 111
%3 = bitcast %struct.jpeg_error_mgr* %2 to void
(%struct.jpeg_common_struct)**
%4 = load void (%struct.jpeg_common_struct)*, void
... (%struct.jpeg_common_struct)** %3, align 8, !tbaa 114
call void @44(%struct.jpeg_common_struct* nonnull %cinfo) #5
br label %if_end

```

```

if_end:
%conv5 = sext i32 %numrows to i64
%cmp6 = icmp ult i64 %div, %conv5
%conv9 = trunc i64 %div to i32
%rowssperchunk.0 = select i1 %cmp6, i32 %conv9, i32 %numrows
%last_rowssperchunk = getelementptr inbounds %struct.my_memory_mgr,
... %struct.my_memory_mgr* %1, #0, i32 6
store i32 %rowssperchunk.0, i32* %last_rowssperchunk, align 8, !tbaa 115
%mul12 = shl nsw nsw i64 %conv5, 3
%call = tail call i8* @alloc_small(%struct.jpeg_common_struct* nonnull
... %cinfo, i32 %pool_id, i64 %mul12)
%5 = bitcast i8** %call to i8**
%cmp1361 = icmp eq i32 %numrows, 0
br i1 %cmp1361, label %while_end, label %while_body_preheader, !prof 118

```

```

while_body_preheader:
br label %while_body

```

```

while_body:
%currow.063 = phi i32 [ %currow.1.lesss, %while_cond.loopexit ], [ 0,
... %while_body_preheader ]
%rowssperchunk.162 = phi i32 [ %rowssperchunk.1.sub, %while_cond.loopexit ], [
... %rowssperchunk.0, %while_body_preheader ]
%sub = sub i32 %numrows, %currow.063
%cmp15 = icmp ult i32 %rowssperchunk.162, %sub
%rowssperchunk.1.sub = select i1 %cmp15, i32 %rowssperchunk.162, i32 %sub
%conv18 = sext i32 %rowssperchunk.1.sub to i64
%mul20 = mul nsw i64 %conv18, %conv
%call22 = tail call i8* @alloc_large(%struct.jpeg_common_struct* %cinfo, i32
... %pool_id, i64 %mul20)
%cmp2357 = icmp eq i32 %rowssperchunk.1.sub, 0
br i1 %cmp2357, label %while_cond.loopexit, label %for_body_preheader, !prof
... 119

```

```

for_body_preheader:
br label %for_body

```

```

for_body:
%workspace.060 = phi i8* [ %add_ptr, %for_body ], [ %call22,
... %for_body_preheader ]
%i.059 = phi i32 [ %dec, %for_body ], [ %rowssperchunk.1.sub,
... %for_body_preheader ]
%currow.158 = phi i32 [ %inc, %for_body ], [ %currow.063,
... %for_body_preheader ]
%inc = add i32 %currow.158, 1
%idxprom = sext i32 %currow.158 to i64
%arrayidx = getelementptr inbounds i8*, i8** %5, #0 %idxprom
store i8* %workspace.060, i8** %arrayidx, align 8, !tbaa 120
%add_ptr = getelementptr inbounds i8, i8* %workspace.060, #0 %conv
%dec = add i32 %i.059, -1
%cmp23 = icmp eq i32 %dec, 0
br i1 %cmp23, label %while_cond.loopexit.loopexit, label %for_body, !prof 119

```

```

while_cond.loopexit.loopexit:
%inc.lesss = phi i32 [ %inc, %for_body ], [ %while_cond.loopexit ]
br label %while_cond.loopexit

```

```

while_cond.loopexit:
%currow.1.lesss = phi i32 [ %currow.063, %while_body ], [ %inc.lesss,
... %while_cond.loopexit.loopexit ]
%cmp13 = icmp ult i32 %currow.1.lesss, %numrows
br i1 %cmp13, label %while_body, label %while_end.loopexit, !prof 118

```

```

while_end.loopexit:
br label %while_end

```

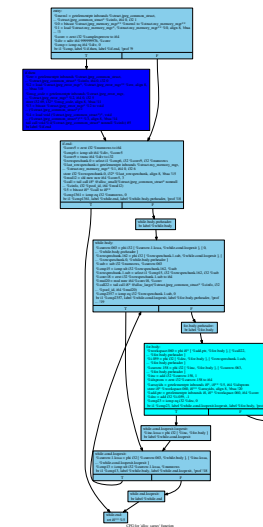
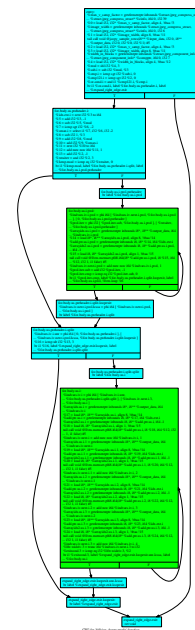
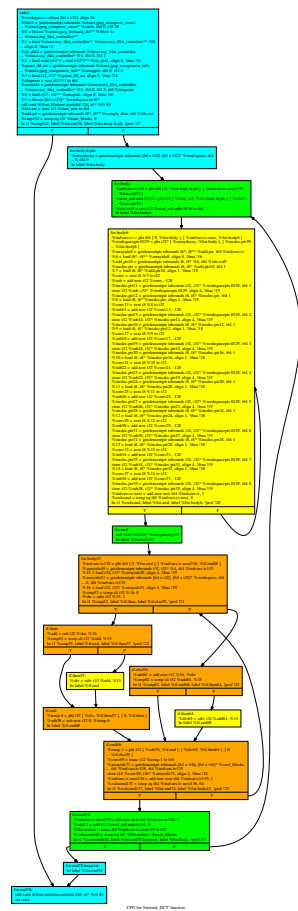
```

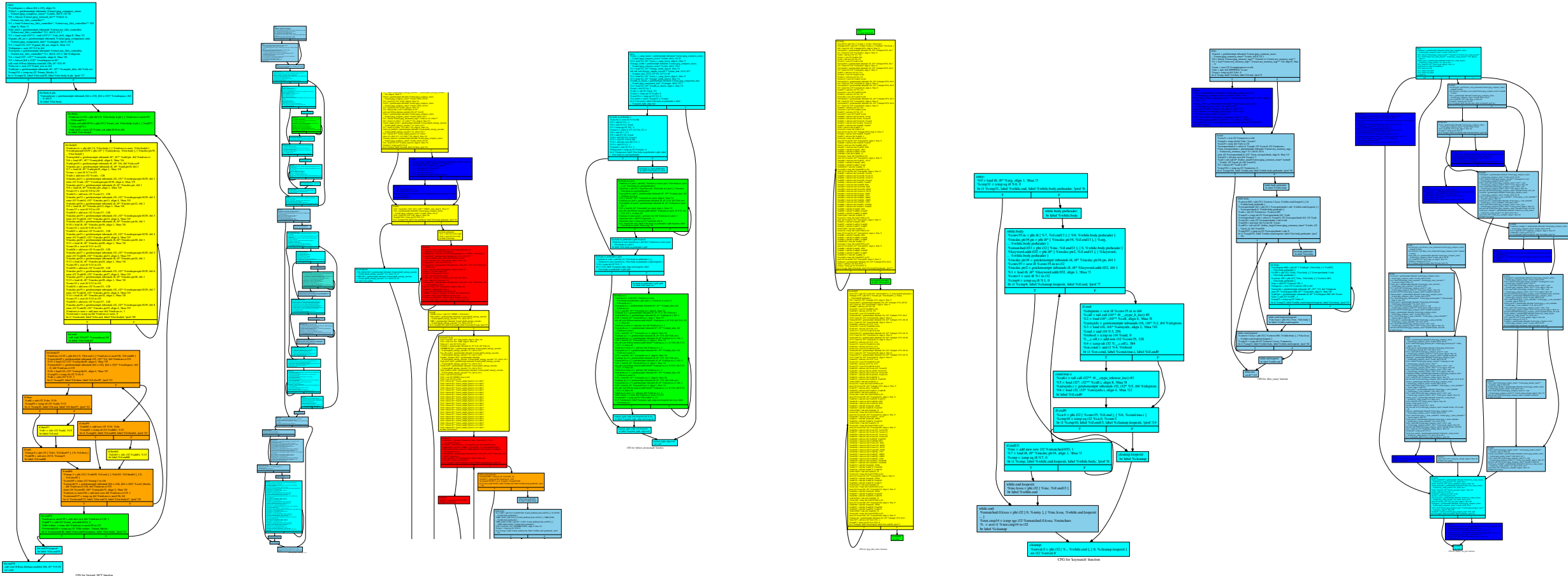
while_end:
ret i8** %5

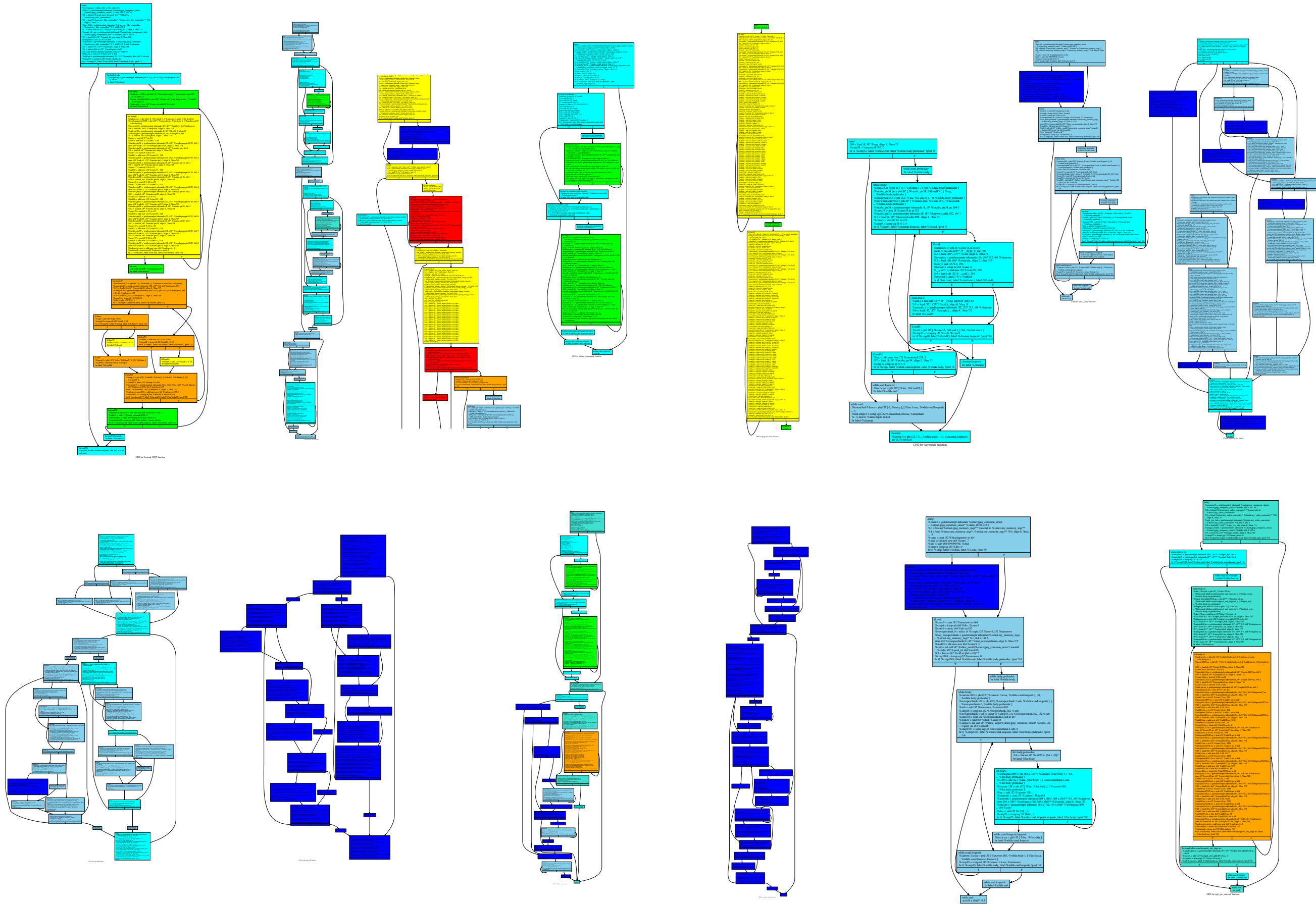
```

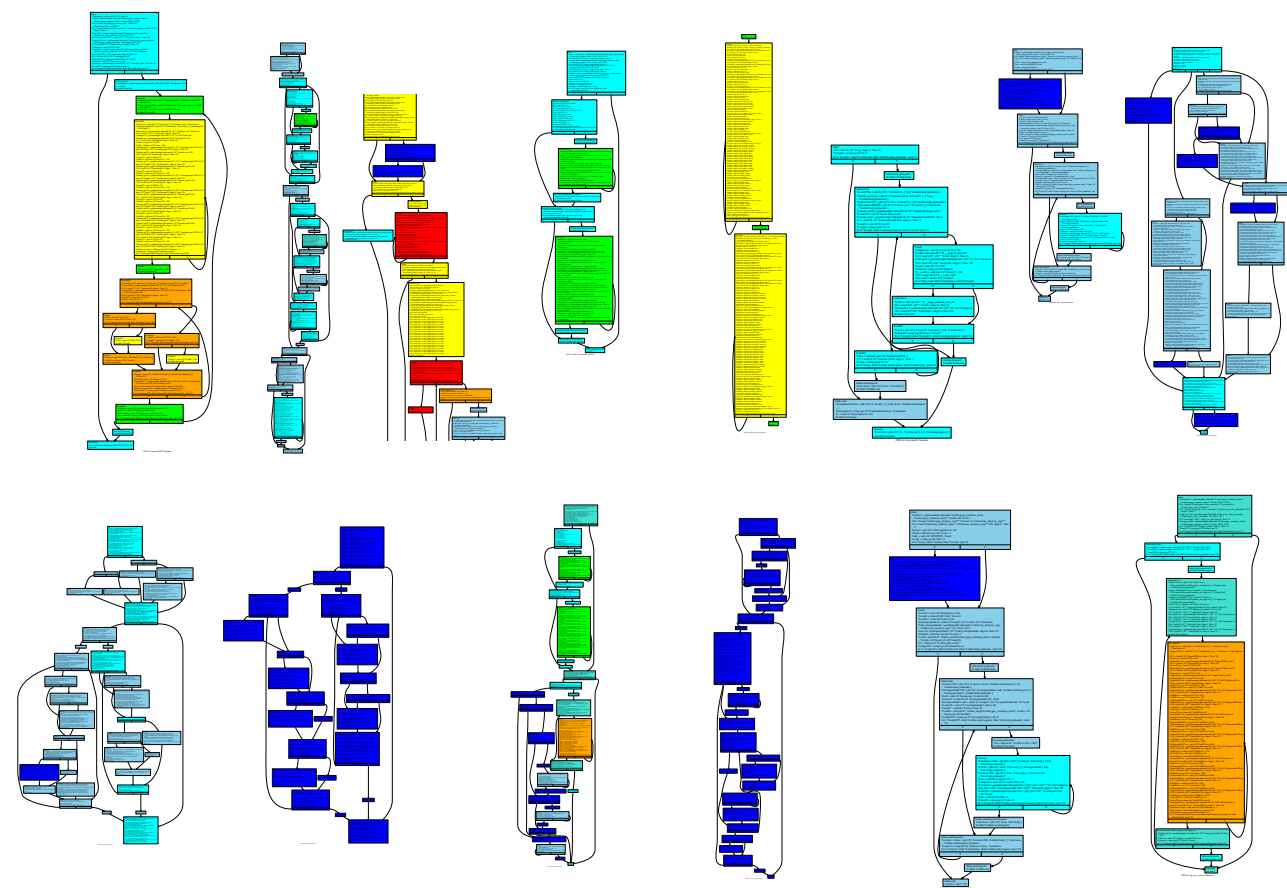
CFG for 'alloc_sarray' function

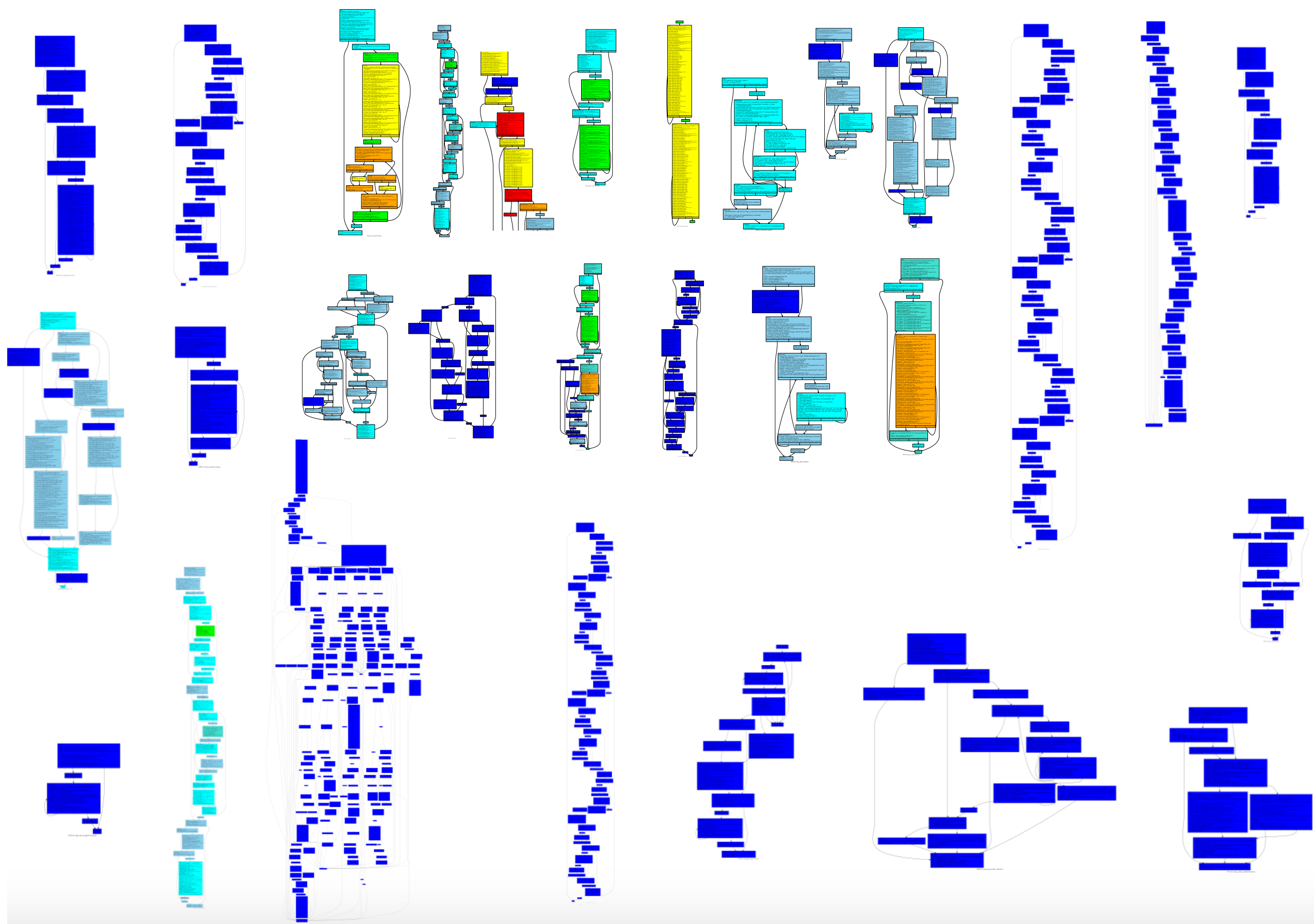








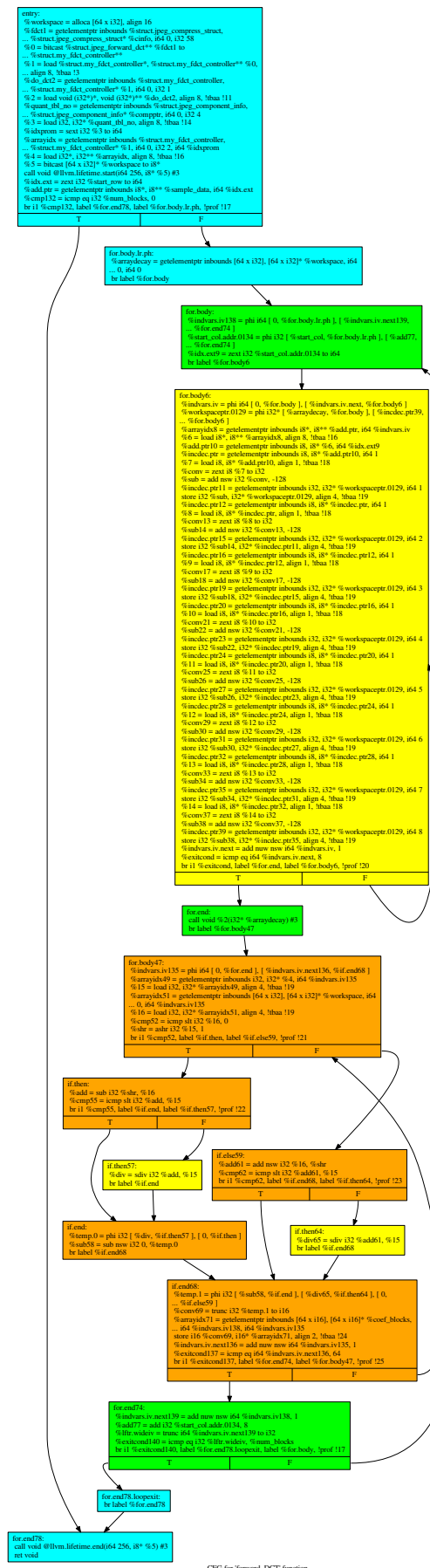




Coldest Regions of an application

INSTANT INTUITION ON EXECUTION FREQUENCY

Hottest Regions of an application



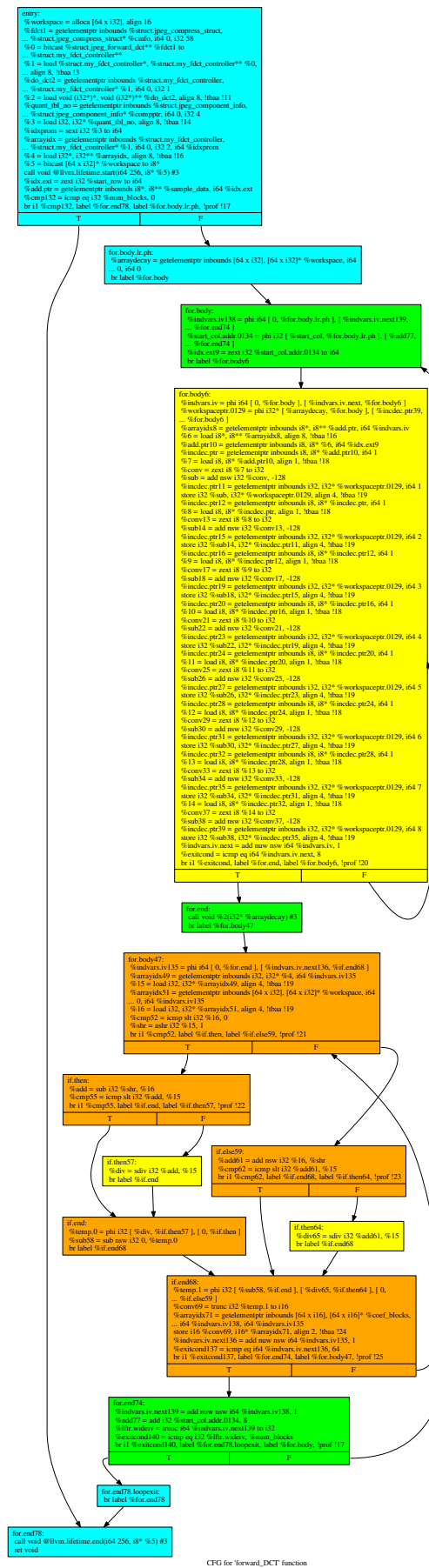
CFG for 'forward_DCT' function

Coldest Regions of an application

INSTANT INTUITION ON EXECUTION FREQUENCY

SYNTHESISE THE HOTTEST REGIONS IN HW

Hottest Regions of an application



CFG for 'forward_DCT' function

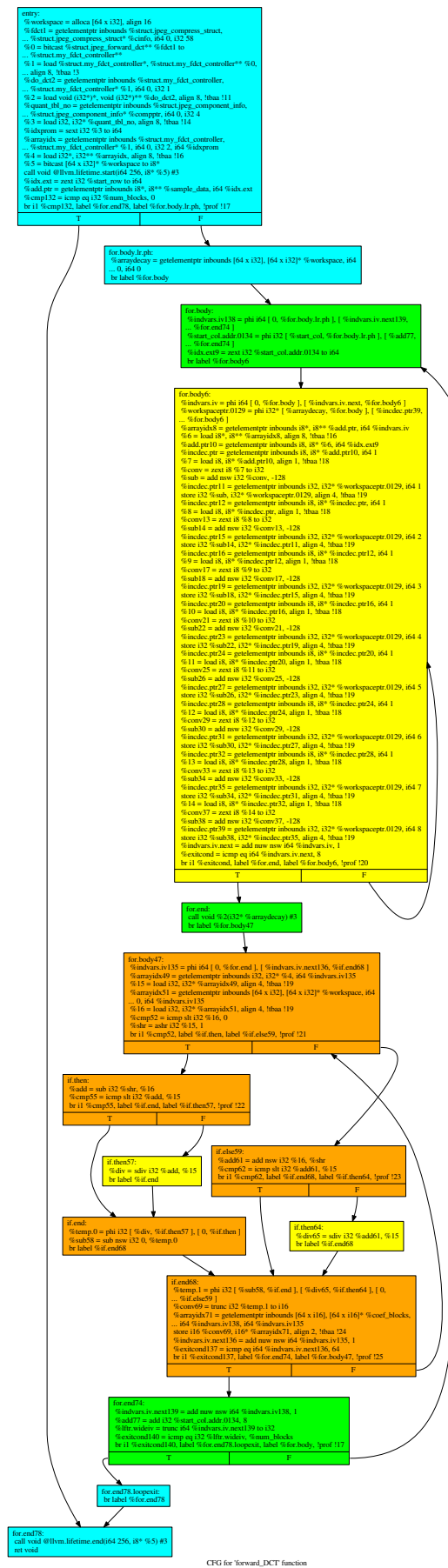
Coldest Regions of an application

INSTANT INTUITION ON EXECUTION FREQUENCY

SYNTHESISE THE HOTTEST REGIONS IN HW

TARGET SW OPTIMIZATION TO HOTTEST REGIONS

Hottest Regions of an application



CFG for 'forward_dct' function

THANK YOU FOR YOUR ATTENTION!