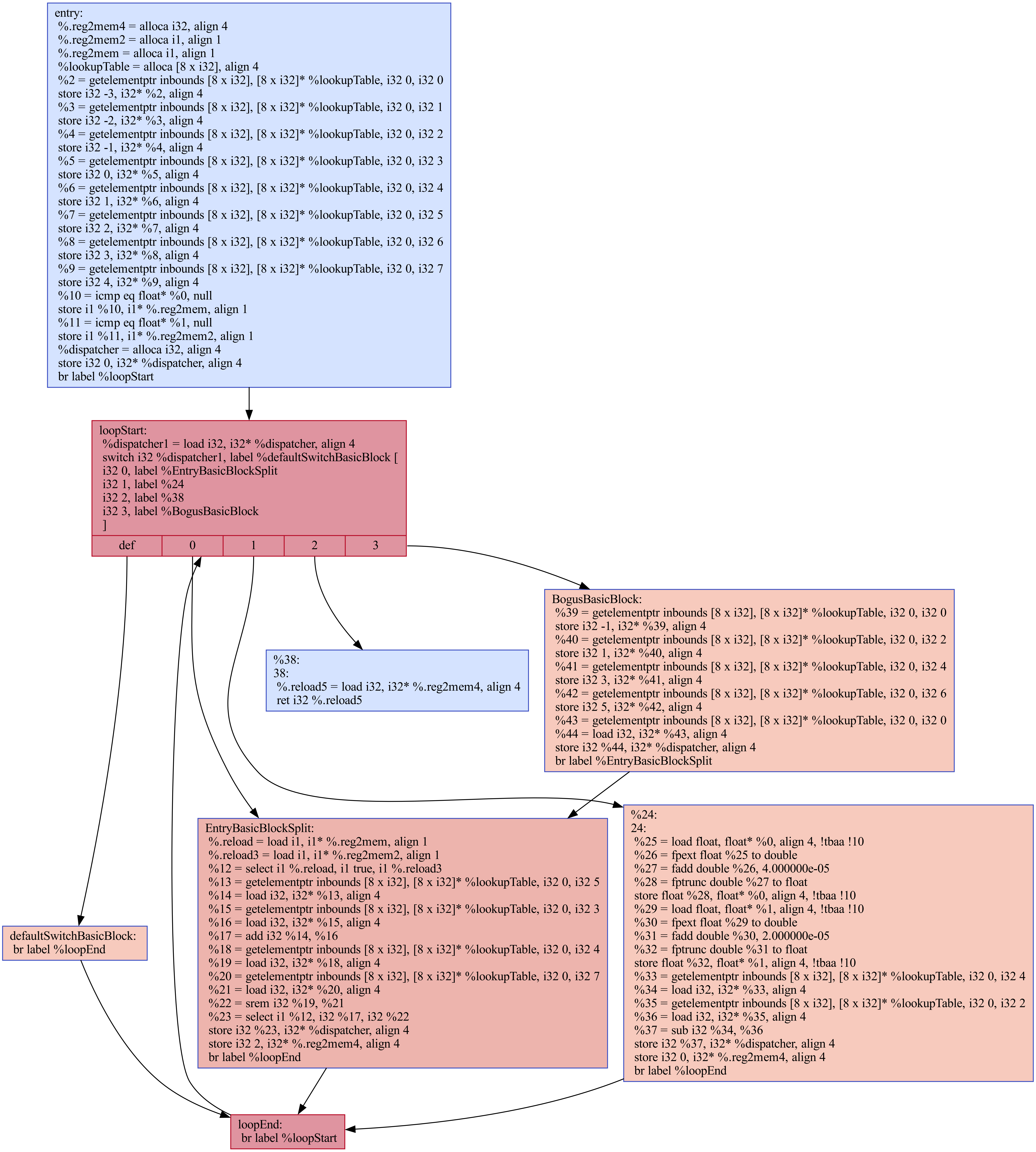

```
entry:
 %.reg2mem4 = alloca i32, align 4
 %.reg2mem2 = alloca i1, align 1
 %.reg2mem = alloca i1, align 1
 %lookupTable = alloca [8 x i32], align 4
 %2 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 0
 store i32 -3, i32* %2, align 4
 %3 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 1
 store i32 -2, i32* %3, align 4
 %4 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 2
 store i32 -1, i32* %4, align 4
 %5 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 3
 store i32 0, i32* %5, align 4
 %6 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 4
 store i32 1, i32* %6, align 4
 %7 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 5
 store i32 2, i32* %7, align 4
 %8 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 6
 store i32 3, i32* %8, align 4
 %9 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 7
 store i32 4, i32* %9, align 4
 %10 = icmp eq float* %0, null
 store i1 %10, i1* %.reg2mem, align 1
 %11 = icmp eq float* %1, null
 store i1 %11, i1* %.reg2mem2, align 1
 %dispatcher = alloca i32, align 4
 store i32 0, i32* %dispatcher, align 4
 br label %loopStart
```

```
loopStart:
 %dispatcher1 = load i32, i32* %dispatcher, align 4
 switch i32 %dispatcher1, label %defaultSwitchBasicBlock [
 i32 0, label %EntryBasicBlockSplit
 i32 1, label %24
 i32 2, label %38
 i32 3, label %BogusBasicBlock
 ]
```

```
BogusBasicBlock:
 %39 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 0
 store i32 -1, i32* %39, align 4
 %40 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 2
 store i32 1, i32* %40, align 4
 %41 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 4
 store i32 3, i32* %41, align 4
 %42 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 6
 store i32 5, i32* %42, align 4
 %43 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 0
 %44 = load i32, i32* %43, align 4
 store i32 %44, i32* %dispatcher, align 4
 br label %EntryBasicBlockSplit
```

```
%38:
38:
 %.reload5 = load i32, i32* %.reg2mem4, align 4
 ret i32 %.reload5
```

```
EntryBasicBlockSplit:
 %.reload = load i1, i1* %.reg2mem, align 1
 %.reload3 = load i1, i1* %.reg2mem2, align 1
 %12 = select i1 %.reload, i1 true, i1 %.reload3
 %13 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 5
 %14 = load i32, i32* %13, align 4
 %15 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 3
 %16 = load i32, i32* %15, align 4
 %17 = add i32 %14, %16
 %18 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 4
 %19 = load i32, i32* %18, align 4
 %20 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 7
 %21 = load i32, i32* %20, align 4
 %22 = srem i32 %19, %21
 %23 = select i1 %12, i32 %17, i32 %22
 store i32 %23, i32* %dispatcher, align 4
 store i32 2, i32* %.reg2mem4, align 4
 br label %loopEnd
```

```
%24:
24:
 %25 = load float, float* %0, align 4, !tbaa !10
 %26 = fpext float %25 to double
 %27 = fadd double %26, 4.000000e-05
 %28 = fptrunc double %27 to float
 store float %28, float* %0, align 4, !tbaa !10
 %29 = load float, float* %1, align 4, !tbaa !10
 %30 = fpext float %29 to double
 %31 = fadd double %30, 2.000000e-05
 %32 = fptrunc double %31 to float
 store float %32, float* %1, align 4, !tbaa !10
 %33 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 4
 %34 = load i32, i32* %33, align 4
 %35 = getelementptr inbounds [8 x i32], [8 x i32]* %lookupTable, i32 0, i32 2
 %36 = load i32, i32* %35, align 4
 %37 = sub i32 %34, %36
 store i32 %37, i32* %dispatcher, align 4
 store i32 0, i32* %.reg2mem4, align 4
 br label %loopEnd
```

```
defaultSwitchBasicBlock:
 br label %loopEnd
```

```
loopEnd:
 br label %loopStart
```

CFG for 'update' function