

# Spot the Difference with **LLVM-FLOW** :

an open-source interactive visualization tool  
for comparing IR CFGs

Jinmyoung Lee



# 1. Control Flow Graph (CFG)

# Control Flow Graph (CFG)

A **Control Flow Graph** is a directed graph that represents the program's control flow, where nodes represent basic blocks, and edges represent the flow of control between those blocks.

Representing LLVM IR as a CFG allows for efficient analysis and optimization of code.

The LLVM optimizer has analysis pass (`-dot-cfg`, `-view-cfg`) that visualizes the CFG.

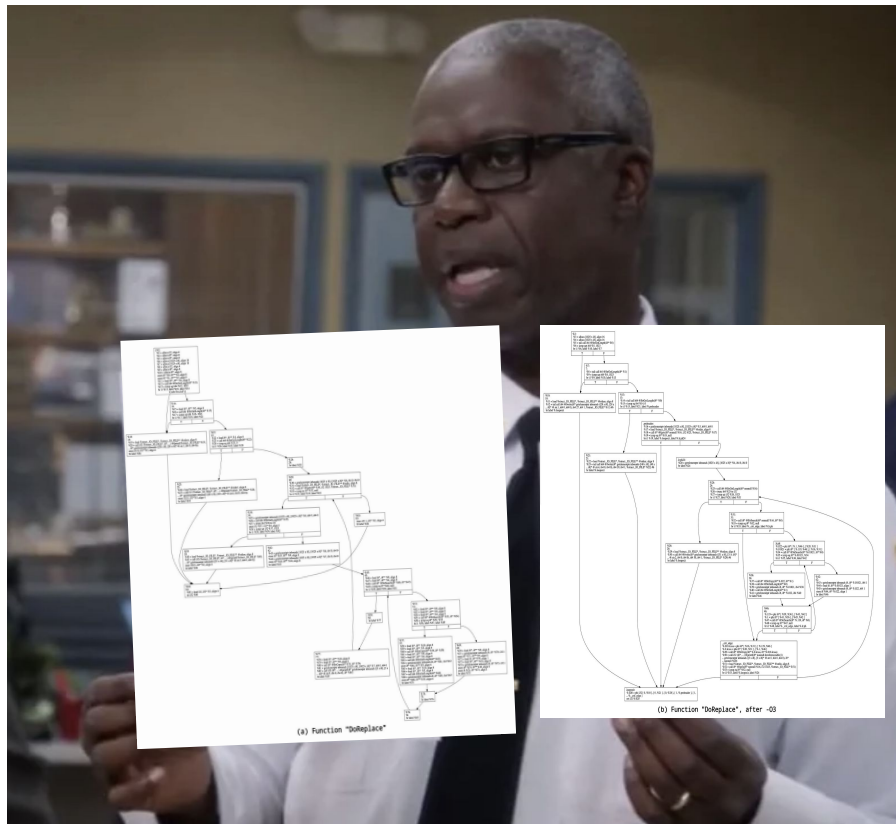


<https://llvm.org/docs/Passes.html>, <https://releases.llvm.org/11.0.0/tools/flang/docs/ControlFlowGraph.html>

However,

Here are two pictures.  
One is CFG before optimization,  
the other is after.

Can you tell what has changed?

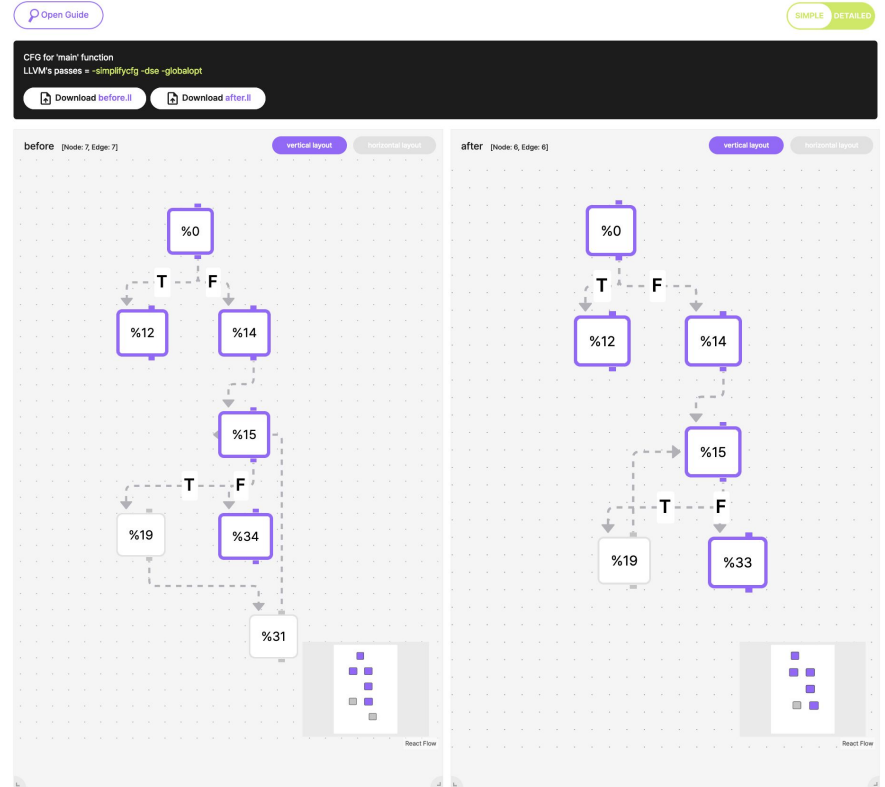


from 'Brooklyn Nine-Nine'

## **2. LLVM-FLOW**

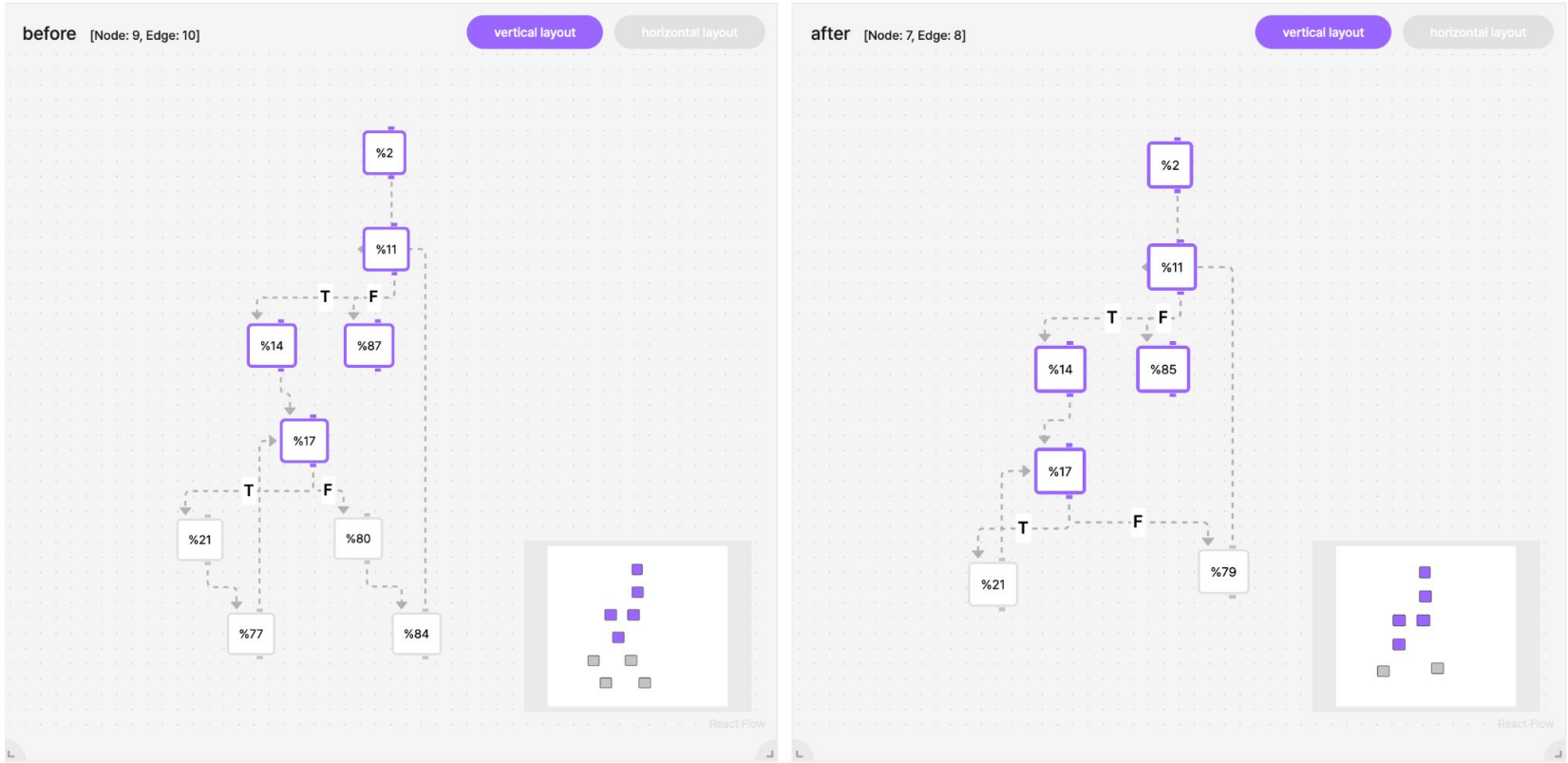
# What is LLVM-FLOW ?

- ✓ visualization tool for comparing IR CFGs
- ✓ interactive web-based interface
- ✓ open-source



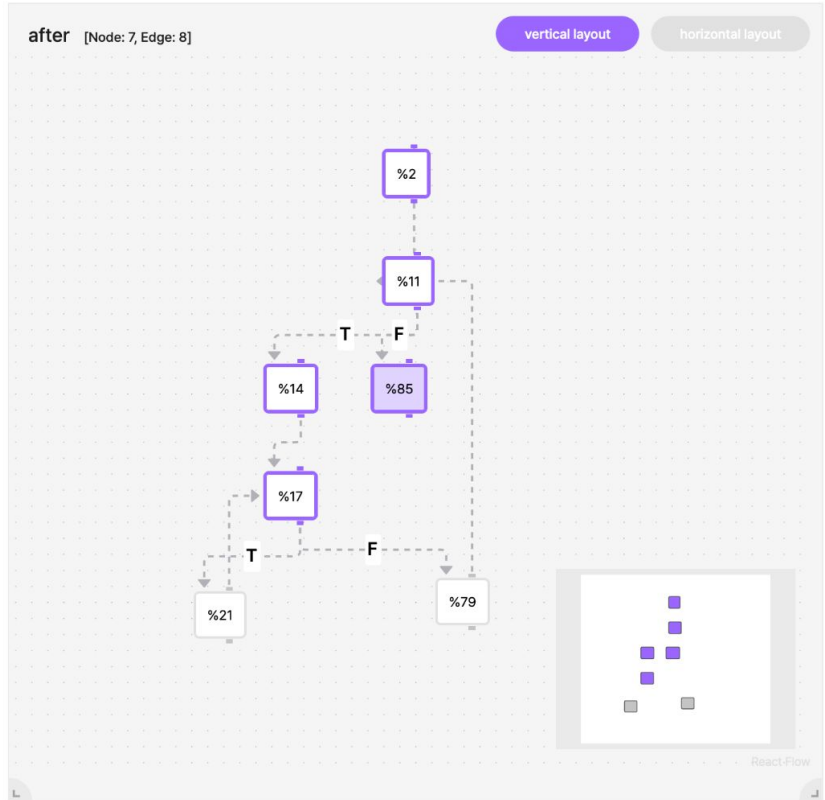
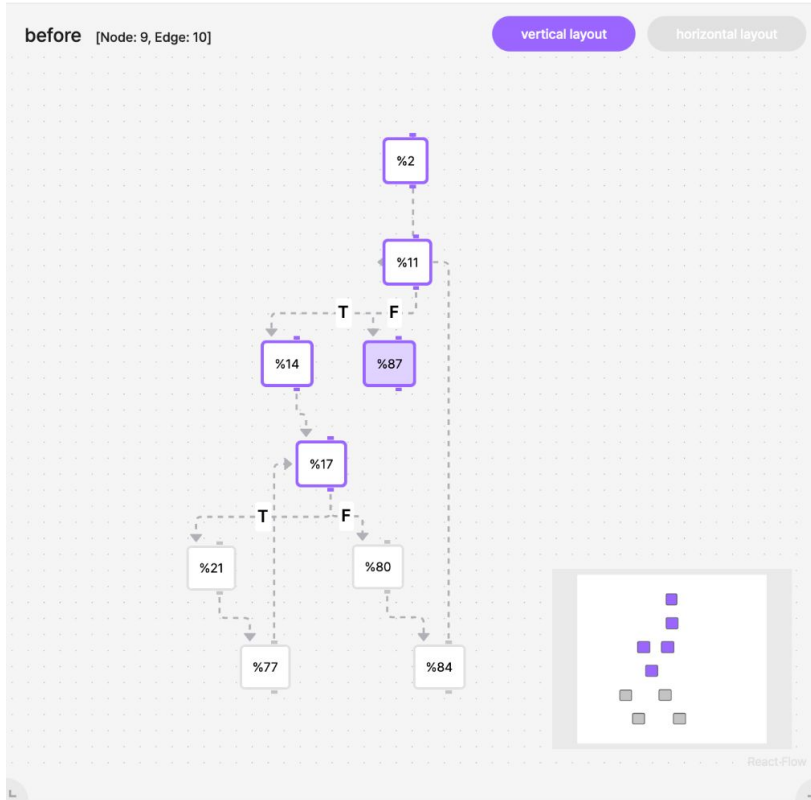


# Example

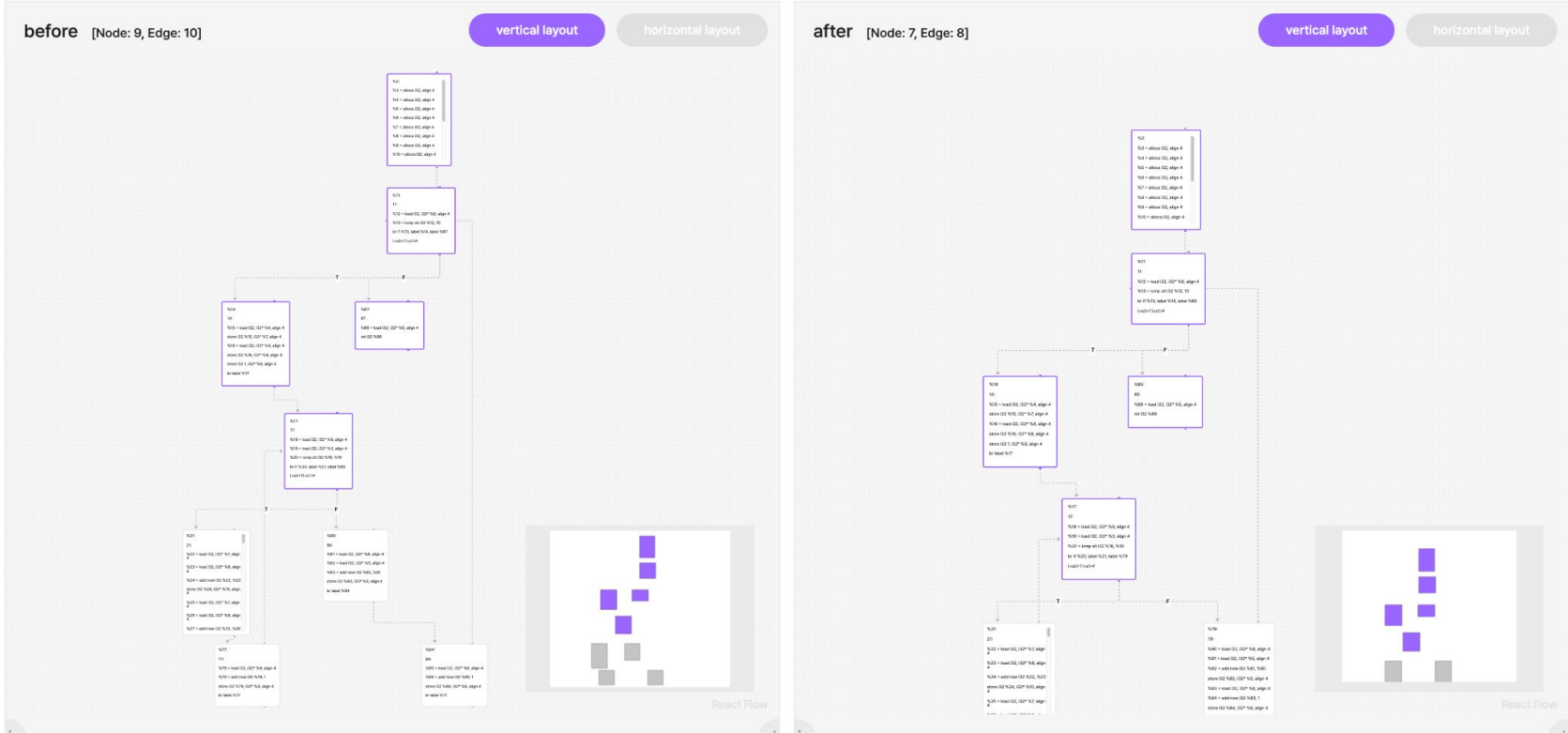




# Example



# Example



# Example

Open Guide

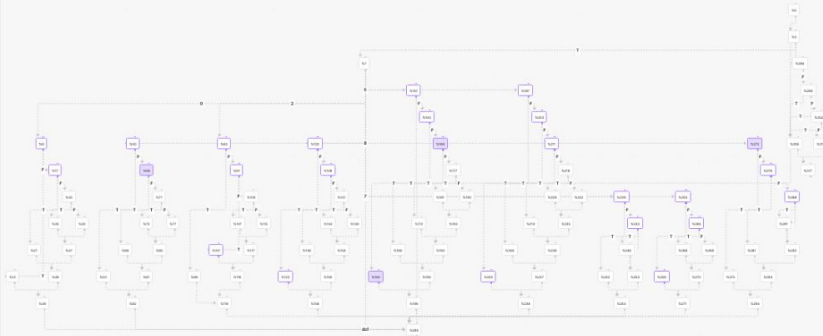
SIMPLE ORIGIN

CFO for 'main' function  
LLVM's passes = -simplifycfg -roa -dse -globalopt -instcombine

initial [Node: 98, Edge: 137]

vertical layout

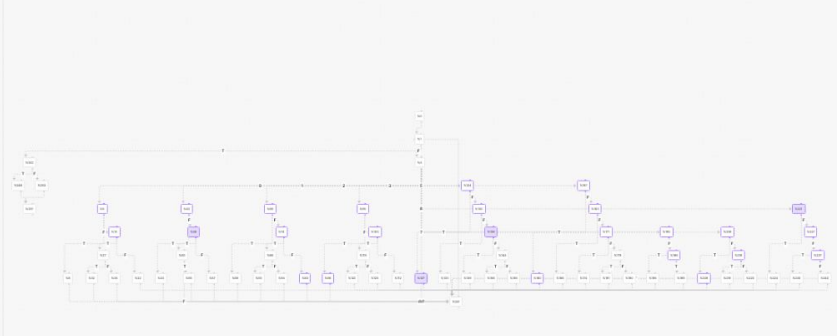
horizontal layout



optimized [Node: 68, Edge: 105]

vertical layout

horizontal layout



---

# How to get started ?

**Get started with :**

- 1. visiting the LLVM-FLOW website (<https://llvmflow.kc-ml2.com/>),**
- 2. running it directly on local environment using Docker.**

The screenshot shows a web browser at the URL `llvmflow.kc-ml2.com`. The page features a navigation bar with links for `Tutorial`, `Board`, `Docs`, `Github`, and a highlighted `Start` button. The main content area has the heading **LLVM-FLOW** and the sub-heading **Visualize the LLVM CFG interactively.** Below this is a button labeled `✓ Tutorial`. A large preview window displays a comparison of two Control Flow Graphs (CFGs). The left side, labeled `Initial (node: 7, edge: 7)`, shows a complex graph with nodes `%0`, `%10`, `%14`, `%18`, `%20`, `%24`, and `%28`. The right side, labeled `optimized (node: 6, edge: 6)`, shows a simplified graph with nodes `%0`, `%12`, `%14`, `%18`, `%19`, and `%23`. The preview window includes a search bar for `Open Guide`, a `Simple` button, and a `LLVM Passes` dropdown menu.

# Demo

The screenshot shows a web browser window with the URL `livmflow.kc-ml2.com/uploadCPP`. The page features the LLVM FLOW logo in the top left and navigation links for Tutorial, Board, Docs, Github, and Start in the top right. Below the navigation, there are three buttons: "with .c file", "with .cpp file" (which is highlighted), and "with .ll file".

The main form contains the following fields and elements:

- User Name:** A text input field.
- .cpp File Upload:** A file selection area with the text "파일 선택" and "선택된 파일 없음".
- LLVM's passes:** A text input field containing the following code:

```
clang 10, lvm 10  
opt beforeg.ll -S -o afterg.ll
```
- submit:** A button to submit the form.
- LLVM's Analysis Passes:** A dropdown menu.
- LLVM's Transform Passes:** A dropdown menu.
- [reference](#): A link to a reference document.

# Demo

The screenshot shows a web browser window with the URL `livmflow.kc-ml2.com/uploadCPP`. The page features the LLVM FLOW logo and a navigation menu with links for Tutorial, Board, Docs, Github, and a Start button. Below the navigation, there are three buttons: `with .c file`, `with .cpp file` (which is highlighted), and `with .ll file`. The form contains the following fields:

- User Name:** `jinyoung`
- .cpp File Upload:** `파일 선택` (File Selection) and `simple.cpp`
- LLVM's passes:** `-simplifycfg -dse -globalopt`

A code block displays the compilation command:

```
clang 10, llvm 10  
opt beforeg.ll -S -simplifycfg -dse -globalopt -o afterg.ll
```

Below the code block is a `submit` button. At the bottom of the form, there are two dropdown menus: `LLVM's Analysis Passes` and `LLVM's Transform Passes`, along with a `reference` link.

# Demo

The screenshot shows the LLVM Flow web application interface. At the top, there is a navigation bar with links for "Tutorial", "Board", "Docs", "Github", and "Start". Below the navigation bar, there is a search bar labeled "Open Guide" and a toggle switch for "SIMPLE" and "DETAILED". The main content area displays the CFG for the function `_Z9is_sortedPii`. The LLVM passes used are `-simplifycfg -dse -globalopt`. There are two buttons: "Download before.ll" and "Download after.ll".

The interface is split into two panels: "before" and "after".

- before** (Node: 8, Edge: 9): Shows a complex CFG with 8 nodes and 9 edges. The nodes are arranged in a non-linear, somewhat circular pattern. A "vertical layout" button is active.
- after** (Node: 7, Edge: 8): Shows a simplified CFG with 7 nodes and 8 edges. The nodes are arranged in a more linear, vertical sequence. A "vertical layout" button is also active.

At the bottom of each panel, there is a small icon representing the layout type: a vertical stack of nodes for the "before" panel and a horizontal stack of nodes for the "after" panel.



# Demo

The screenshot shows a web browser window with the URL `llvmflow.kc-ml2.com/board`. The page features the LLVM FLOW logo and navigation links for Tutorial, Board, Docs, Github, and a Start button. A search bar is present with the text 'User Name' and the input 'jinmyoung'. A 'Search' button is located below the input. The search results are displayed in a table with the following columns: date, user name, file, LLVM's passes, and show graph.

| date             | user name | file       | LLVM's passes                | show graph            |
|------------------|-----------|------------|------------------------------|-----------------------|
| 2023.04.24/08:33 | jinmyoung | simple.cpp | -simplifycfg -dse -globalopt | <a href="#">start</a> |
| 2023.04.24/08:32 | jinmyoung | simple.cpp | -simplifycfg -dse -globalopt | <a href="#">start</a> |

Below the table, there is a pagination control showing '< 1 >'.



# **3. Conclusion**

---

## Conclusion

**LLVM-FLOW is an open-source tool that enables easy comparison of changes in the CFG.**

**With your feedback and participation, we hope LLVM-FLOW grows into a project that contributes to the LLVM community :)**

---

## QR code

[llvmflow.kc-ml2.com](https://llvmflow.kc-ml2.com)

