Prof. Anirban Sengupta, CSE, Indian Institute of Technology Indore
https://www.anirban-sengupta.com/

AspireHLS
An HLS Tool/Framework for Data Intensive Hardware Applications

It describes our '*AspireHLS*' tool that is an HLS tool/framework that generates RTL datapath (as hypergraph) from input intermediate representation (IR) as DFG (generated from its high-level code such as C).

This document provides detailed tutorial of *AspireHLS* tool which comprises of three segments:

- HLS HLT Tool Package
- DSE Package
- RTL Datapath Generator Package

The HLS HLT Tool package is explained step-by-step as follows:

The HLS HLT Tool package is called by '**HLS_HLT_IR**' folder. The package folder comprises of the following:

a. **HLS_HLT.jar**
b. **Input**
c. **Output**

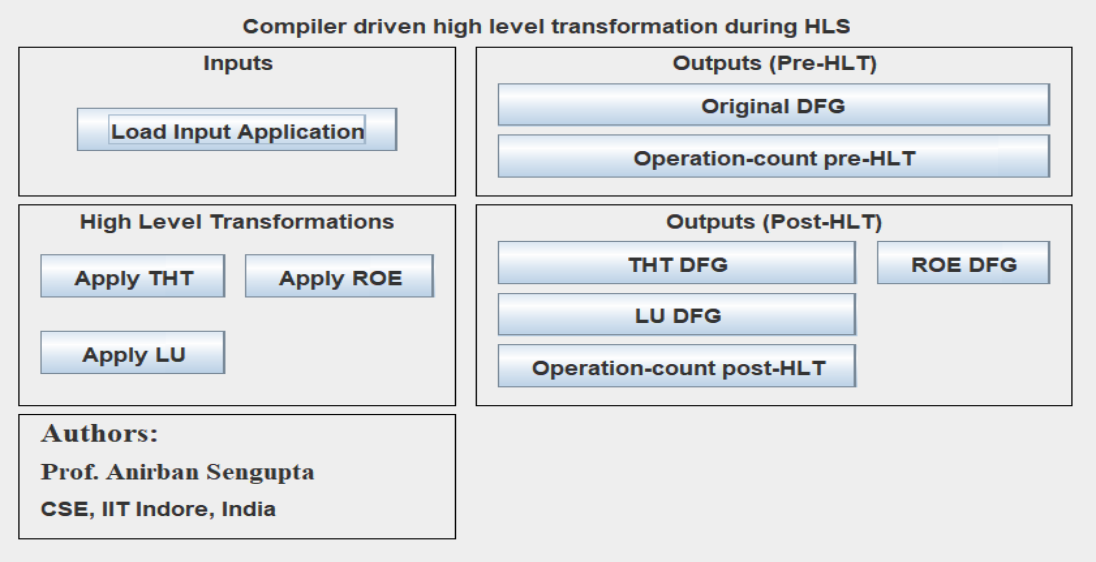
The HLS_HLT.jar file needs the following JDK17 version installed in the machine where the Tool is supposed to run: https://adoptium.net/temurin/releases?&version=17&os=any&arch=any

The 'Input' folder comprises of the raw DFG as intermediate representation (IR) in text file format (generated from parsing high-level code such as C). For demonstration, the 'Input' folder currently comprises of the following data intensive application as HLS benchmarks: *Convolutional layer.txt, DCT_8Point.txt, embossment filter.txt, FIR filter.txt, IIR Butterworth Filter.txt, Sharpening filter.txt*.

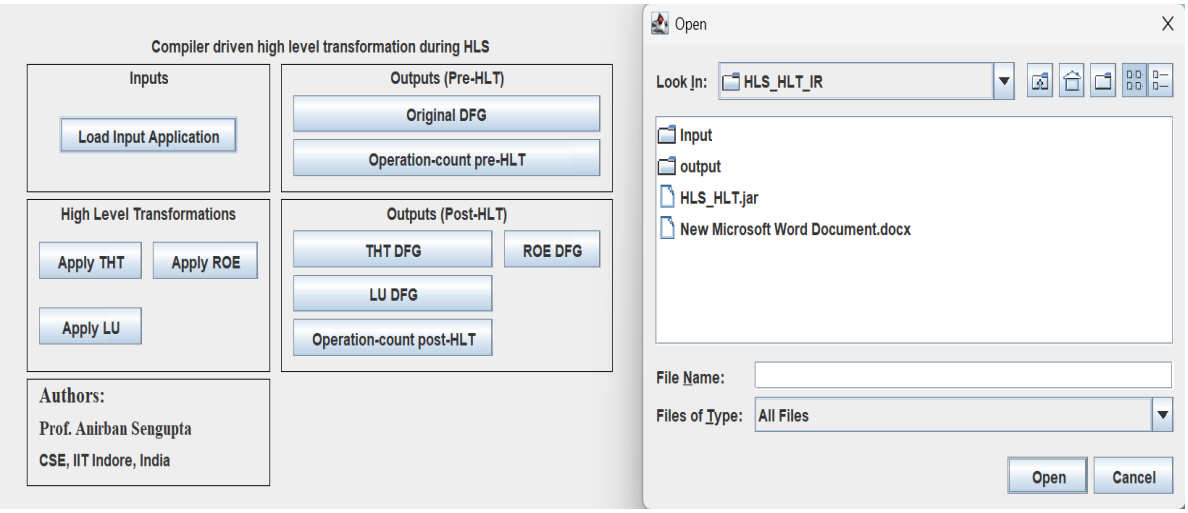
The 'Output' folder will contain the processed DFG as intermediate representation in text file format after undergoing compiler driven high level transformations (HLTs) such as loop unrolling (LU), tree height transformation (THT) and redundant operation elimination (ROE).

Now we explain the step-by-step process of executing **HLS_HLT.jar** file.

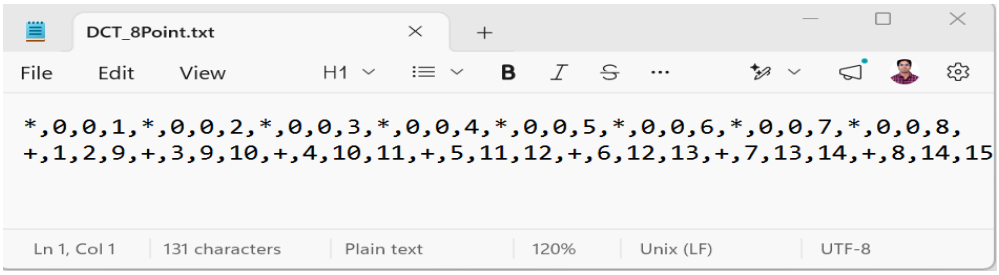
Step 1: Execute **HLS_HLT.jar.** Post execution, the GUI of the Tool opens:



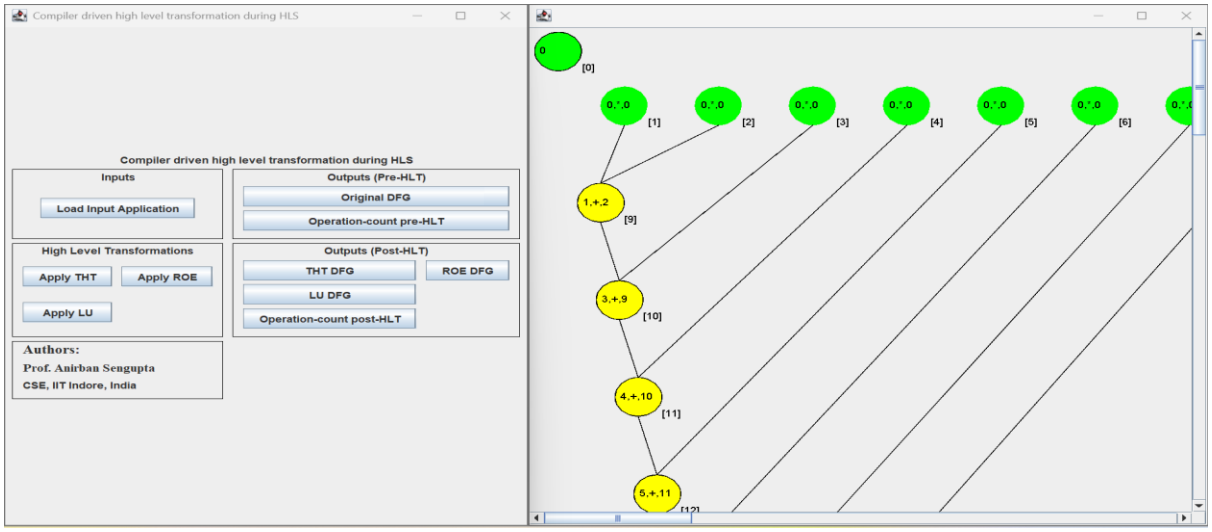
Step 2: Click on '***Load Input Application***' to load the respective HLS application in the form of raw DFG as intermediate representation in text file format. Browse to the '**HLS_HLT_IR'** folder and select the desired application inside the '**Input**' folder.



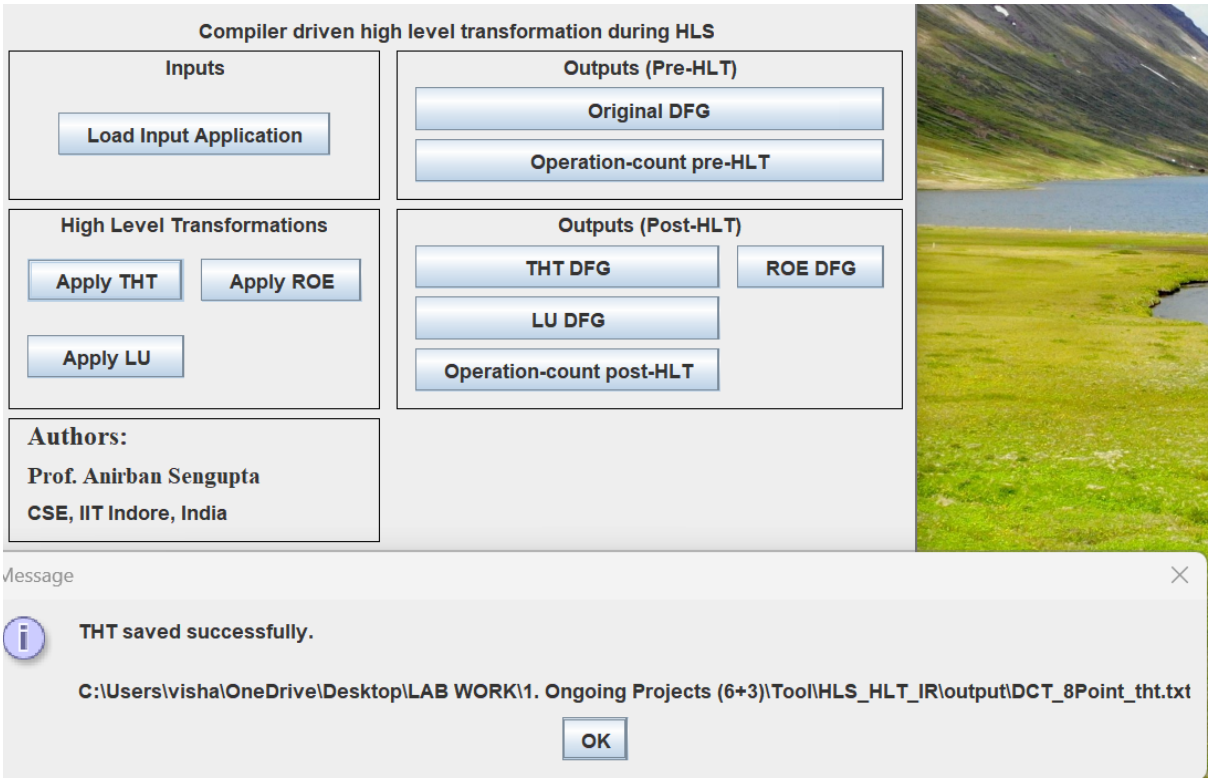
Step 3: On loading desired application, the corresponding raw DFG can be viewed by clicking on the '***Original DFG***' button. For demonstration we have loaded '***DCT_8Point.txt***'. The text file format of its IR is shown below:



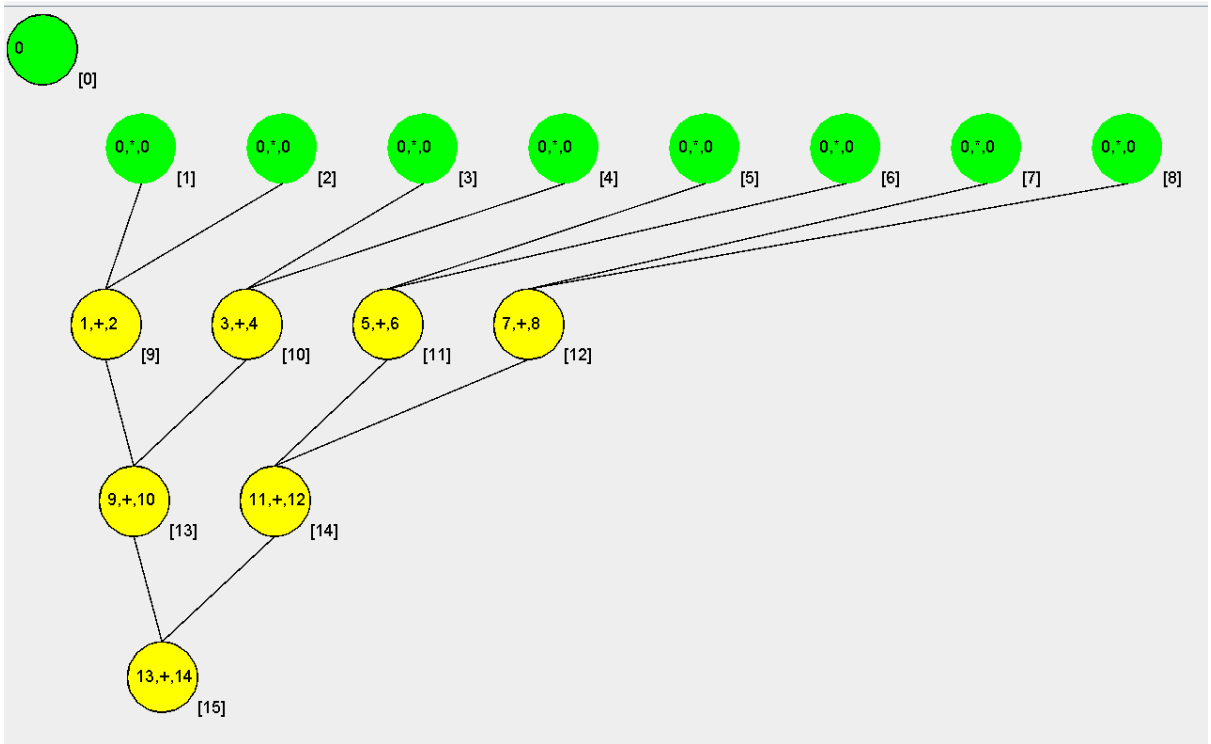
The corresponding DFG is generated as shown below. Here, each node represents an operation (addition, multiplication etc.). The independent nodes are indicated with (0,0) as inputs, while the dependent nodes are indicated by their respective input node IDs. For example, opn. 9 accepts outputs from node 1 and node 2 respectively. Here, green nodes are independent ones and yellow nodes are dependent ones.



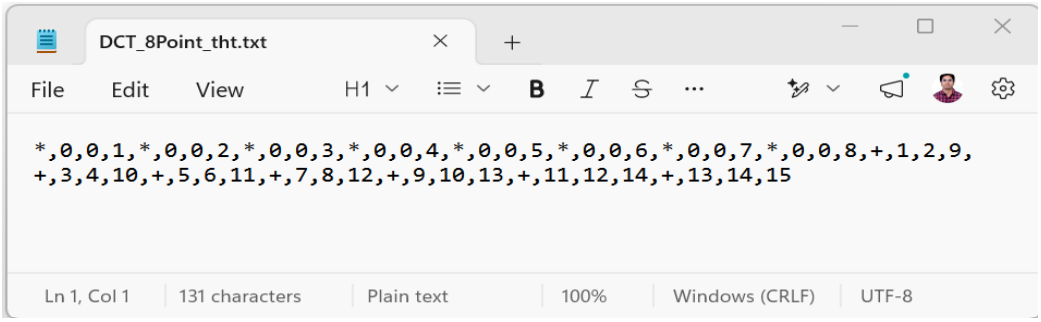
Step 4: Next click on '***Apply THT***' button to apply tree height transformation HLT on the raw DFG IR. On successful transformation the Tool generates a pop-up message stating the folder where the processed DFG IR text file format is generated and saved. This is typically in the ***'Output'*** folder of the Tool package. Please see the screenshot below:



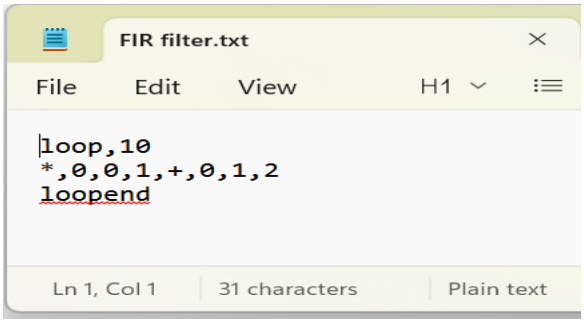
Step 5: The THT processed DFG IR can be viewed by clicking on the '***THT DFG***' under 'Outputs (post-HLT)'. For example, the THT processed DFG IR generated for '***DCT_8Point.txt***' is shown in the screenshot below:



The respective processed DFG IR as text file format of '***DCT_8Point.txt***', generated and saved in the *'Output'* folder as ***'DCT_8Point_tht.txt'*** is shown below. As evident, the output IR text file format is different from input IR text file format.

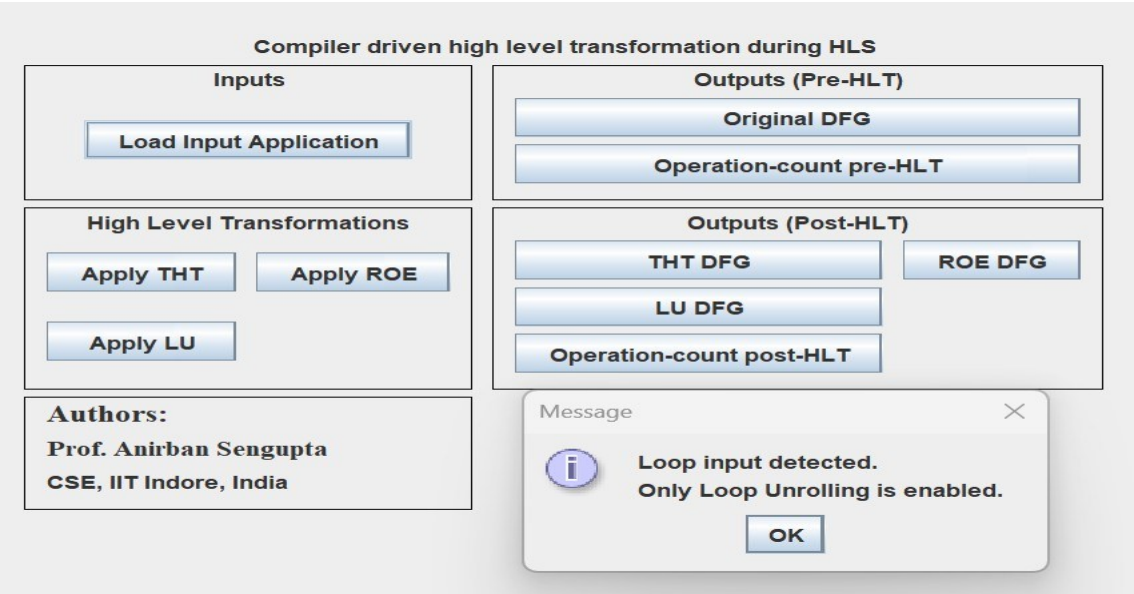


Step 6: The user/designer needs to close the **HLS_HLT.jar** interface and relaunch again before loading a new application. For example, for demonstration, we close the jar interface before relaunching the interface again for loading new application called ***'FIR filter.txt'***, using the steps as explained in steps 1 and 2. ***'FIR filter.txt'*** contains loop, therefore, the input DFG IR text file format includes *loop* keyword inside it as shown below:

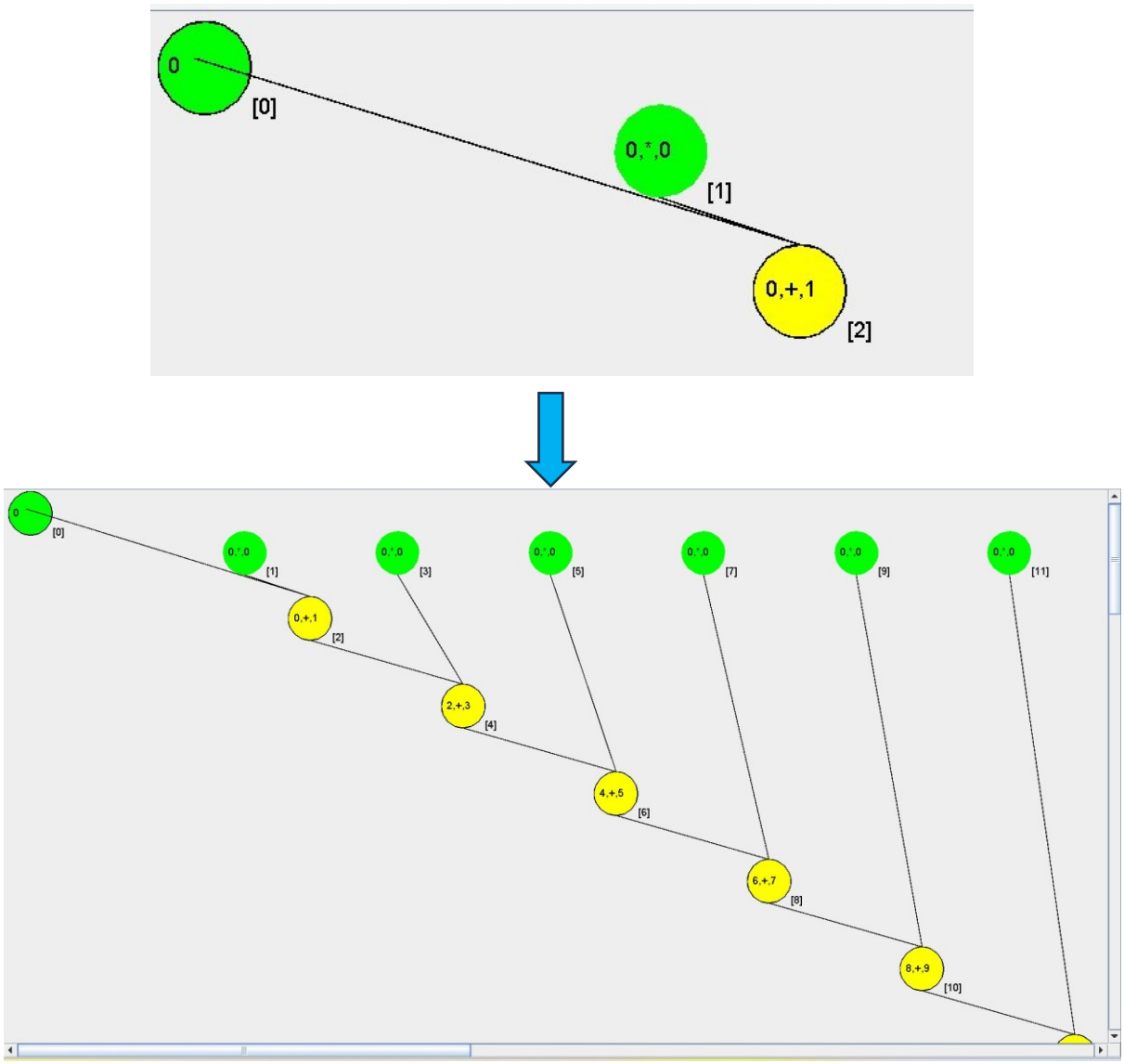


As seen above the ***'FIR filter.txt'*** contains *loop* keyword along with loop unrolling factor (UF) value. In this case the loop UF value is specified as '*10*'. This UF value can be changed as per the designer's choice. The loop UF indicates the number of times the loop body is duplicated. In the text file format, the loop body information is specified within the *loop* and *loopend* keywords.

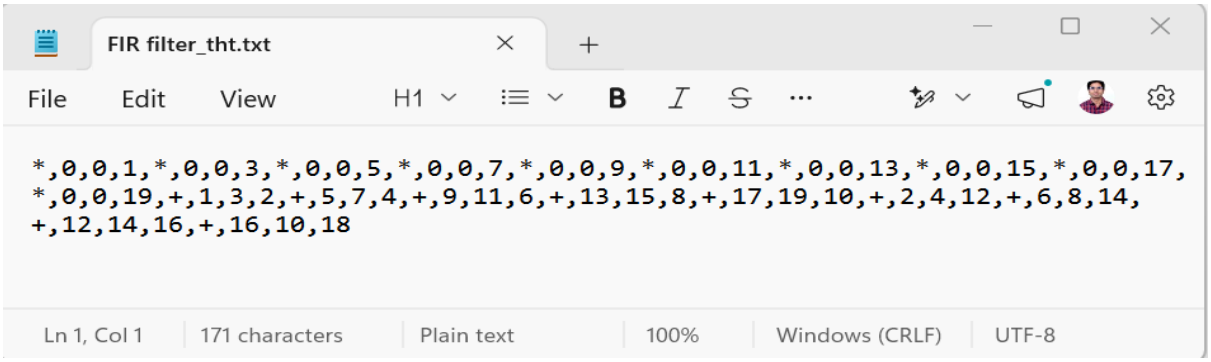
Step 7: On loading the ***'FIR filter.txt'***, the loop unrolling feature as HLT is detected and enabled by the Tool as shown below:



Step 8: On clicking the '***LU DFG***' button under '***Outputs (Post-HLT)***', the LU transformed processed DFG IR of '***FIR filter.txt***' is generated, as shown below:



Step 9: The LU transformed processed DFG IR of '**FIR filter.txt**' can be further processed through THT based HLT technique by clicking on '**Apply THT**' button, resulting into THT transformed process DFG as '**FIR filter_tht.txt**' generated and saved in 'Output' folder. The respective THT processed DFG IR can be viewed by clicking on '**THT DFG**' button. The '**FIR filter_tht.txt**' and THT processed DFG IR are shown below:



Prof. Anirban Sengupta, CSE, Indian Institute of Technology Indore
https://www.anirban-sengupta.com/