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

AspireHLS
An HLS Tool/Framework for Data Intensive Hardware Applications

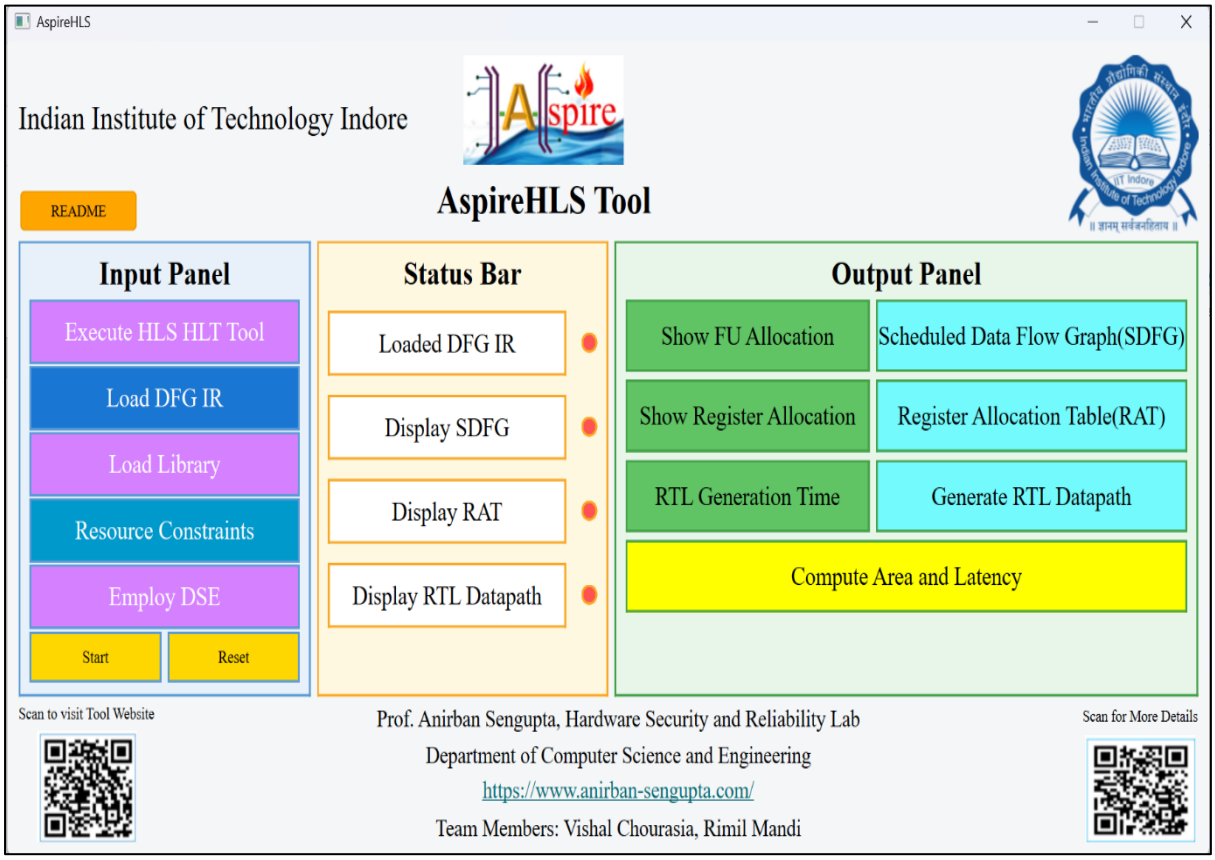
It describes our '*AspireHLS*' tool that is an HLS tool/framework that generates RTL datapath (as hypergraph) from input intermediate representation (IR) as DFG (generated from its high-level code such as C).

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

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

The screenshot below provides the GUI of the '**AspireHLS Tool'.** The interface comprises of Input Panel, Status Panel and Output Panel. The designer needs to invoke/call the *HLS HLT Tool* button, to execute the HLT Tool that is responsible for converting a raw DFG intermediate representation (IR) into processed DFG IR. The execution opens another GUI for performing this task.

The working/execution of the HLS HLT Tool package is explained in the next page.



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

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

The '**AspireHLS Tool'** folder comprises of the following:

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

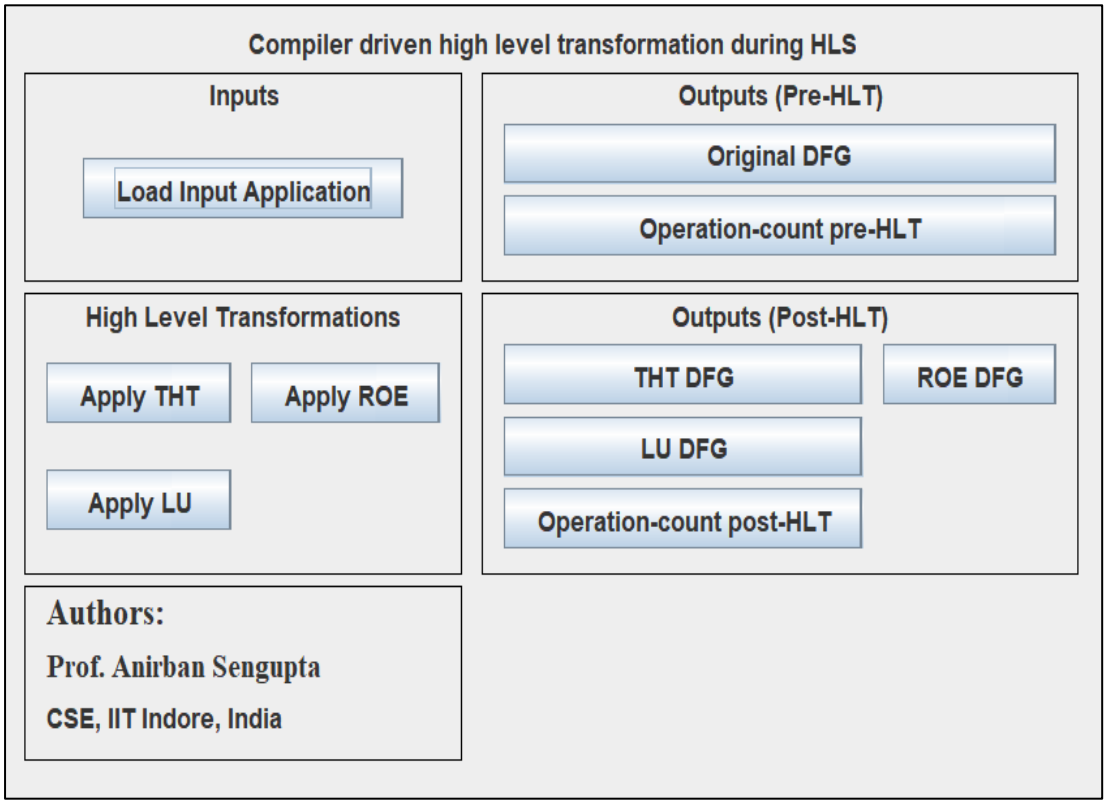
The HLS_HLT.jar file needs the following JDK17 version installed in the machine where the Tool is supposed to run: https://adoptium.net/temurin/releases?&version=17&os=any&arch=any

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*.

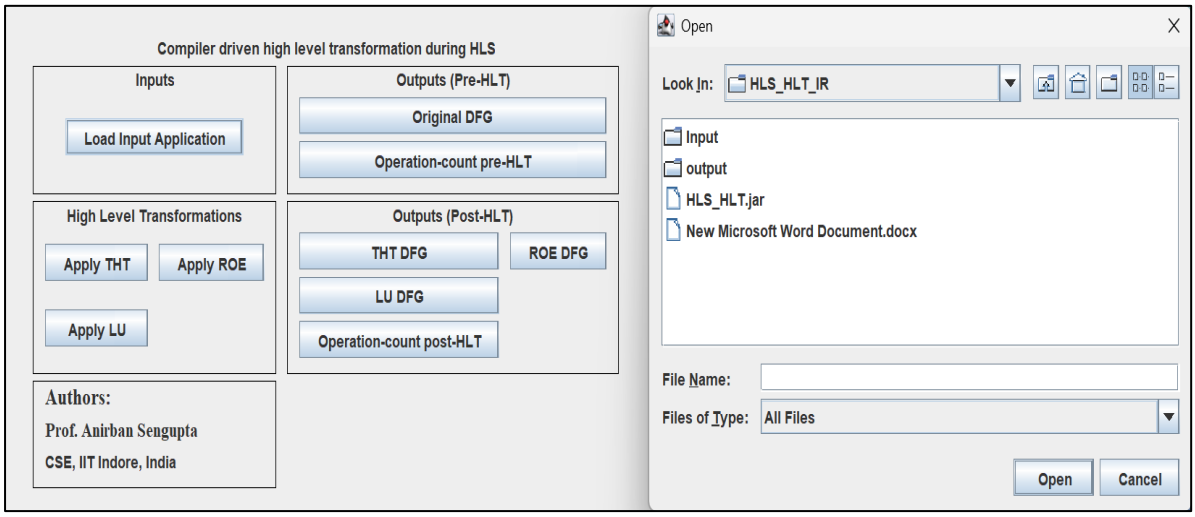
The 'Output' folder will contain the processed DFG as intermediate representation in text file format after undergoing compiler driven high level transformations (HLTs) such as loop unrolling (LU), tree height transformation (THT) and redundant operation elimination (ROE).

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

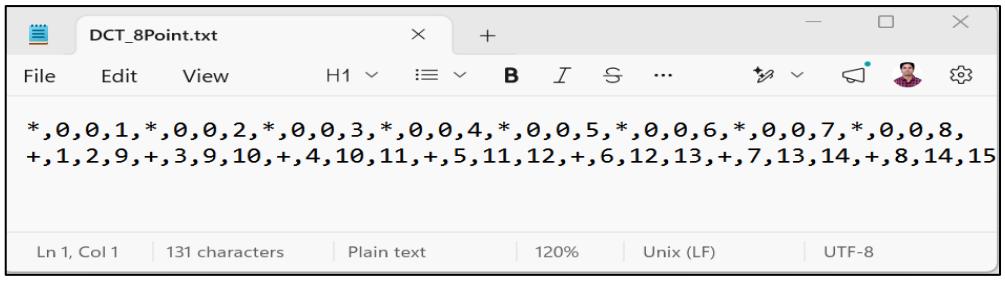
Step 1: Execute **HLS_HLT.jar.** Post execution, the GUI of the Tool opens:



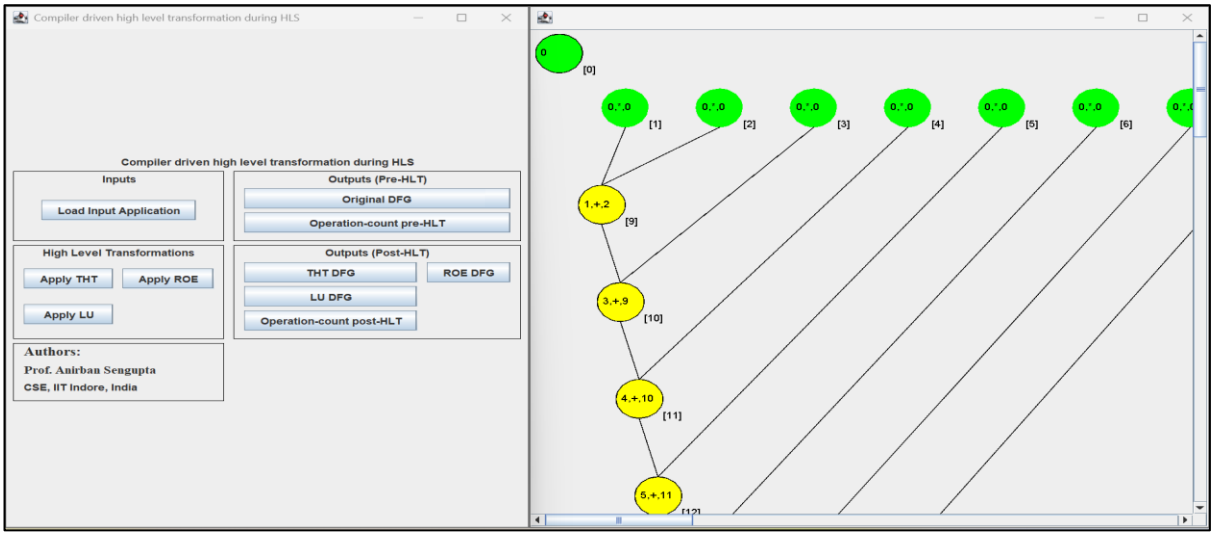
Step 2: Click on '***Load Input Application***' to load the respective HLS application in the form of raw DFG as intermediate representation in text file format. Browse to the '**HLS_HLT_IR'** folder and select the desired application inside the '**Input**' folder.



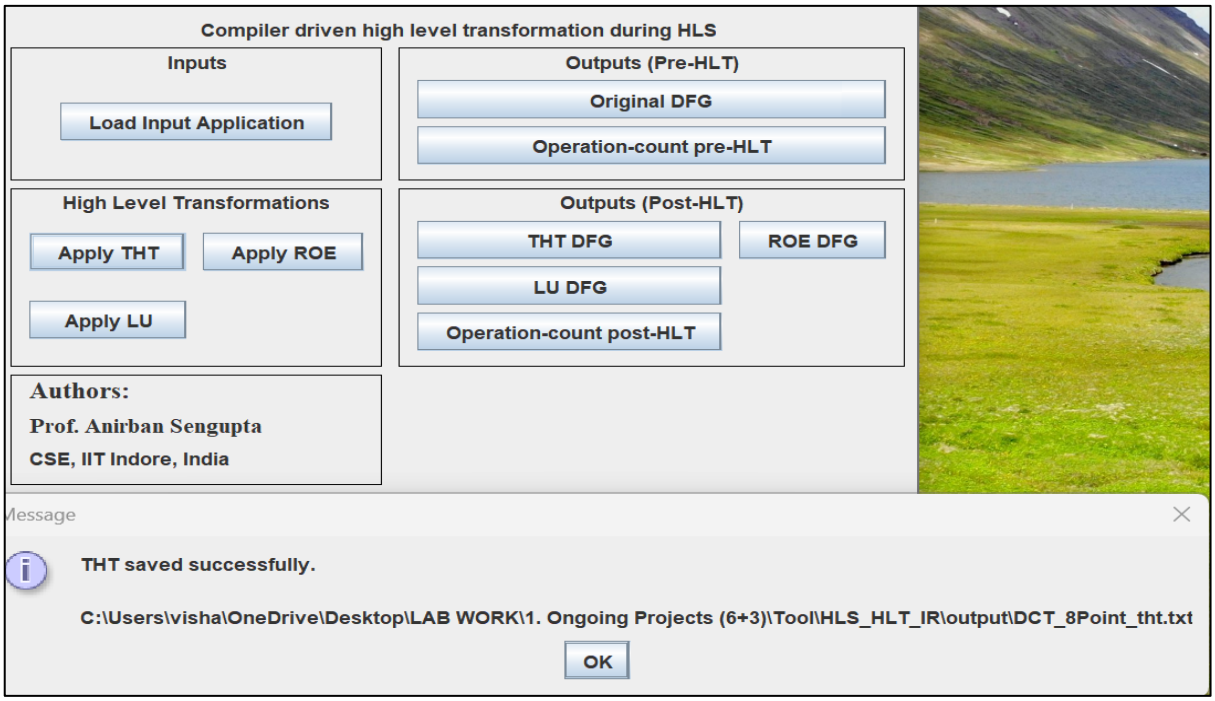
Step 3: On loading desired application, the corresponding raw DFG can be viewed by clicking on the '***Original DFG***' button. For demonstration we have loaded '***DCT_8Point.txt***'. The text file format of its IR is shown below:



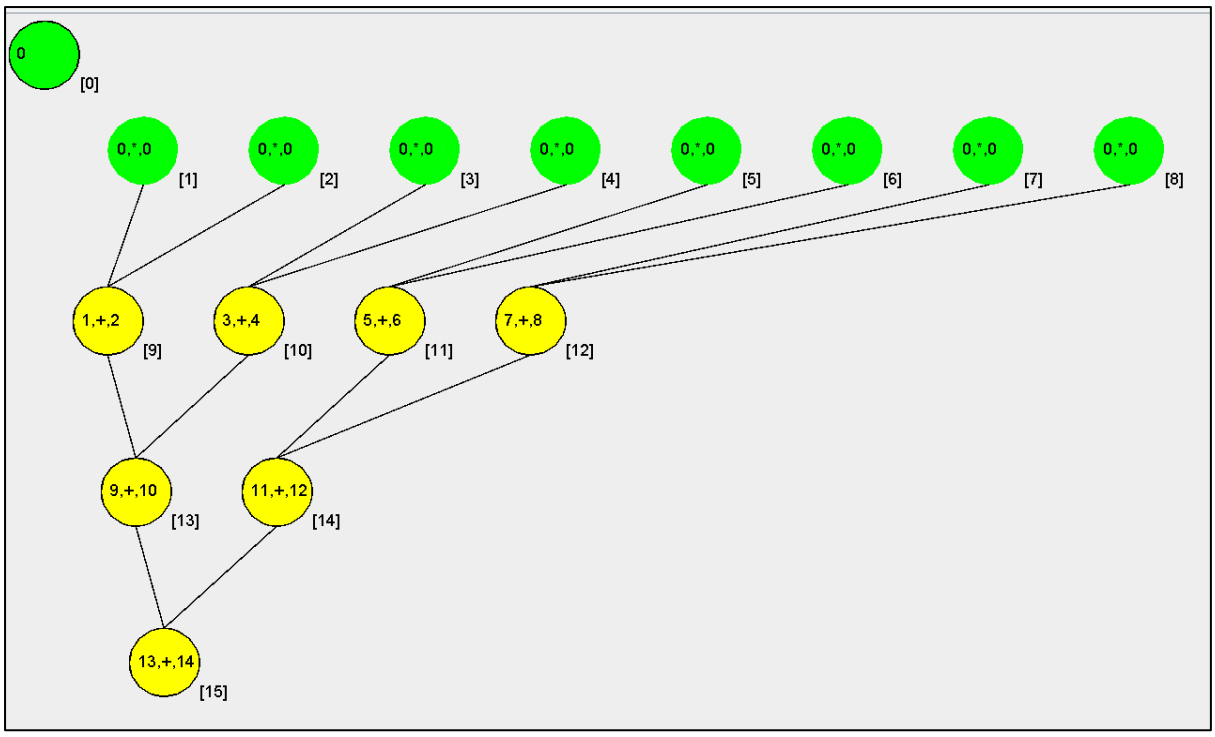
The corresponding DFG is generated as shown below. Here, each node represents an operation (addition, multiplication etc.). The independent nodes are indicated with (0,0) as inputs, while the dependent nodes are indicated by their respective input node IDs. For example, opn. 9 accepts outputs from node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



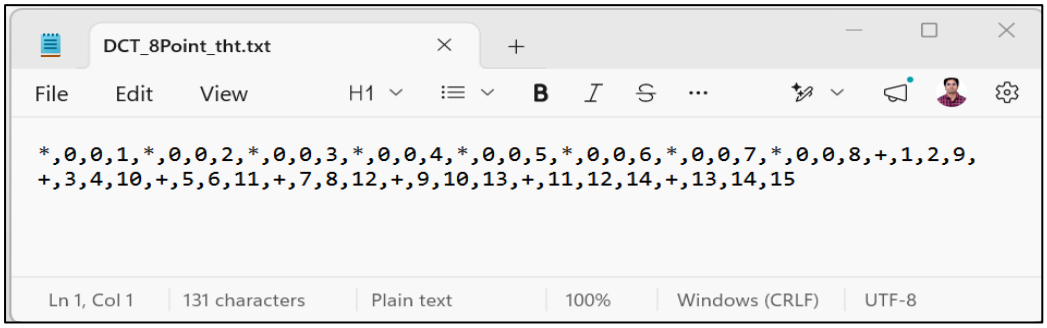
Step 4: Next click on '***Apply THT***' button to apply tree height transformation HLT on the raw DFG IR. On successful transformation the Tool generates a pop-up message stating the folder where the processed DFG IR text file format is generated and saved. This is typically in the ***'Output'*** folder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



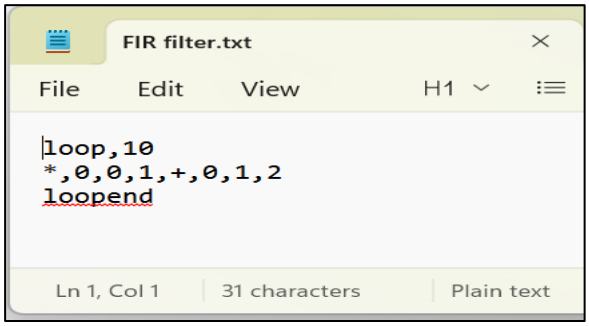
Step 5: The THT processed DFG IR can be viewed by clicking on the '***THT DFG***' under 'Outputs (post-HLT)'. For example, the THT processed DFG IR generated for '***DCT_8Point.txt***' is shown in the screenshot below:



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.

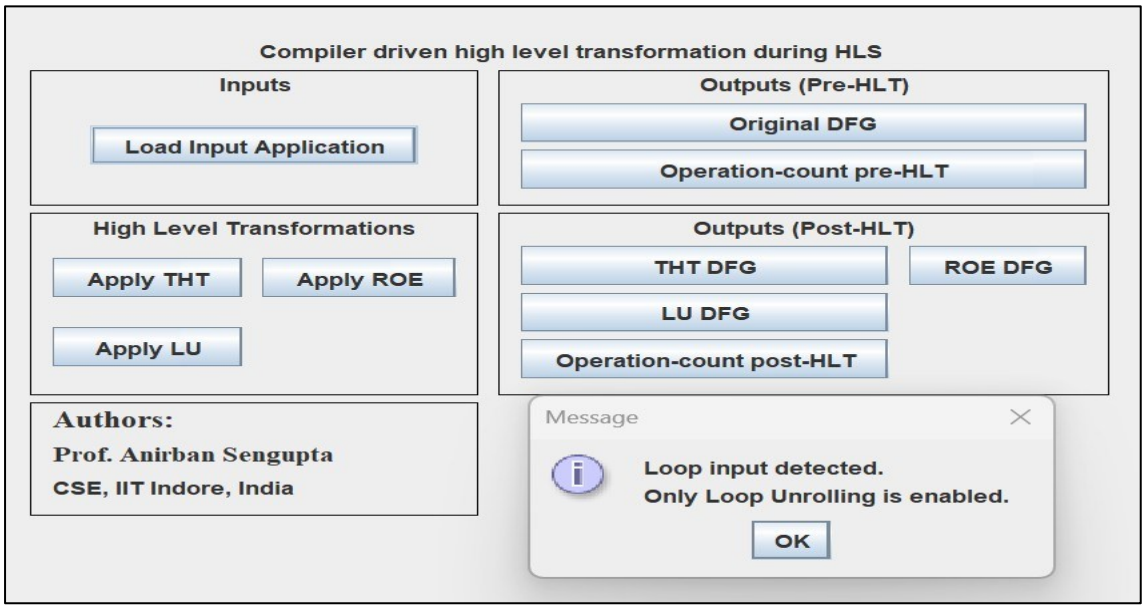


Step 6: The user/designer needs to close the **HLS_HLT.jar** interface and relaunch again before loading a new application. For example, for demonstration, we close the jar interface before relaunching the interface again for loading new application called ***'FIR filter.txt'***, using the steps as explained 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:

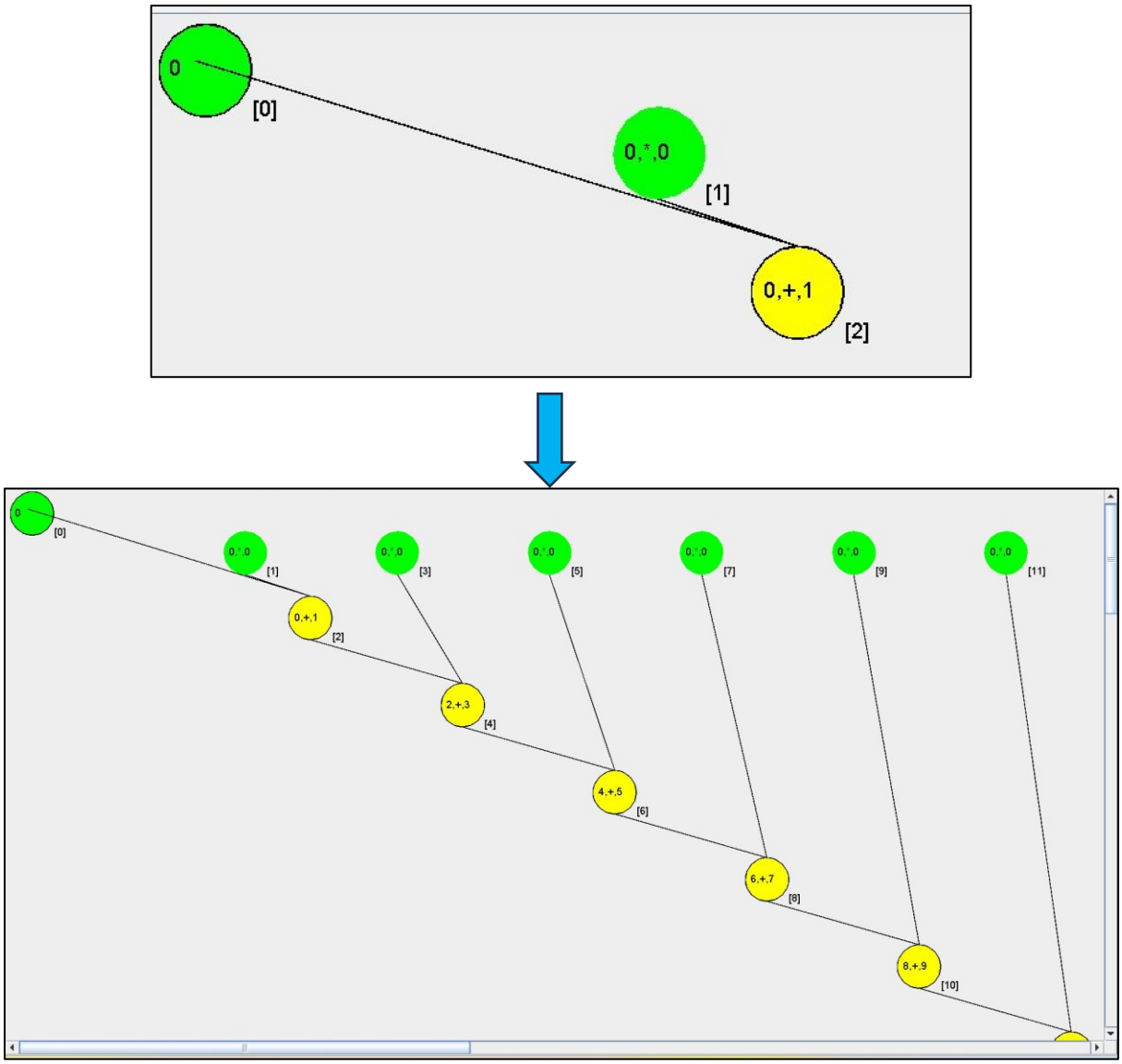


As seen above the ***'FIR filter.txt'*** contains *loop* keyword along with loop unrolling factor (UF) value. In this case the loop UF value is specified as '*10*'. This UF value can be changed as per the designer's choice. The loop UF indicates the number of times the loop body is duplicated. In the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

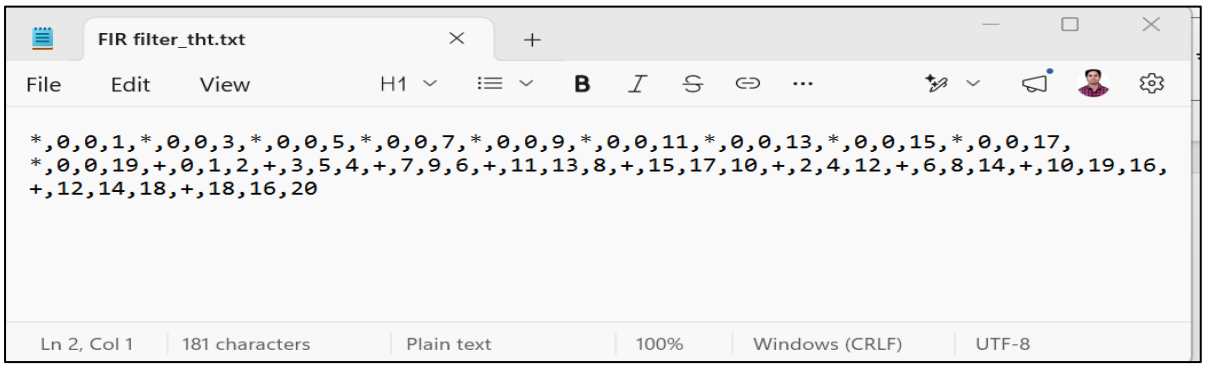
Step 7: On loading the ***'FIR filter.txt'***, the loop unrolling feature as HLT is detected and enabled by the Tool as shown below:

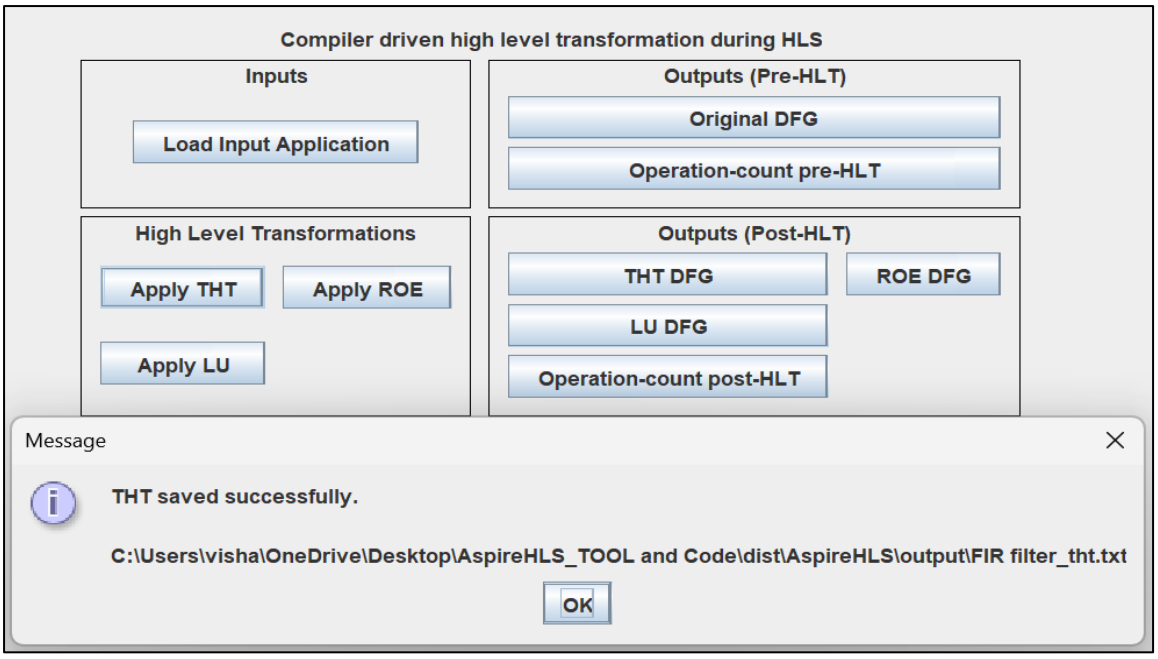


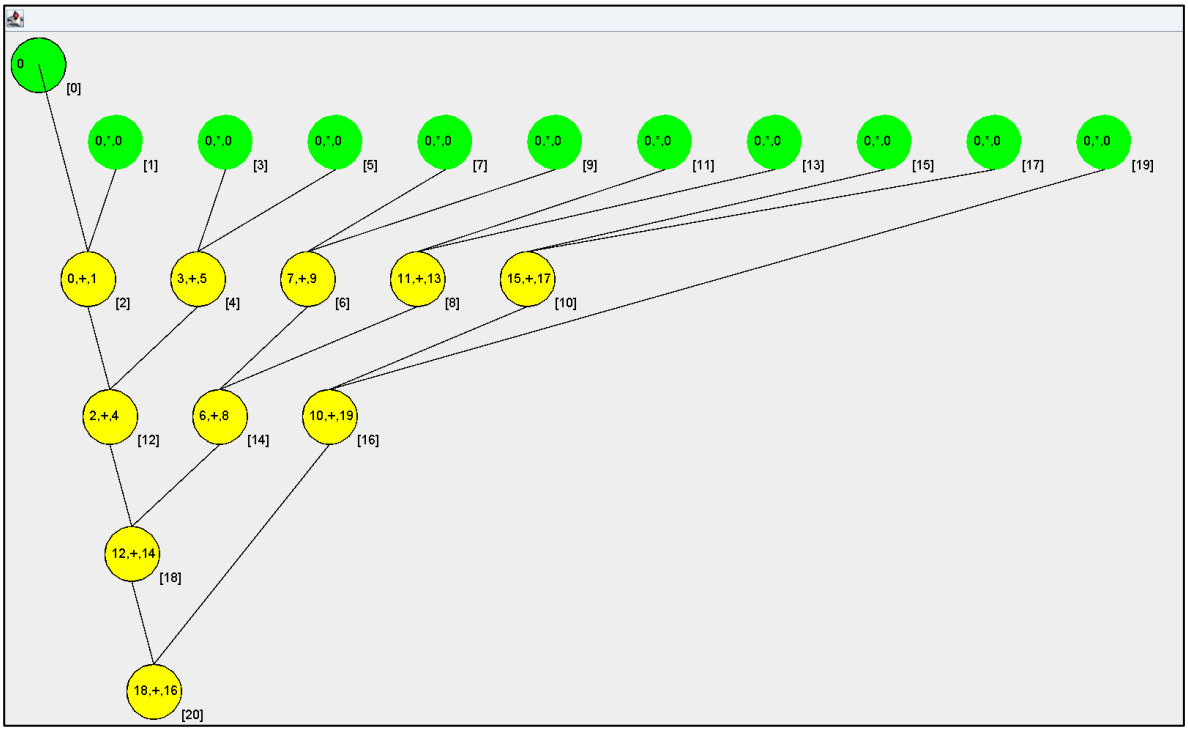
Step 8: On clicking the '***LU DFG***' button under '***Outputs (Post-HLT)***', the LU transformed processed DFG IR of '***FIR filter.txt***' is generated, as shown below:



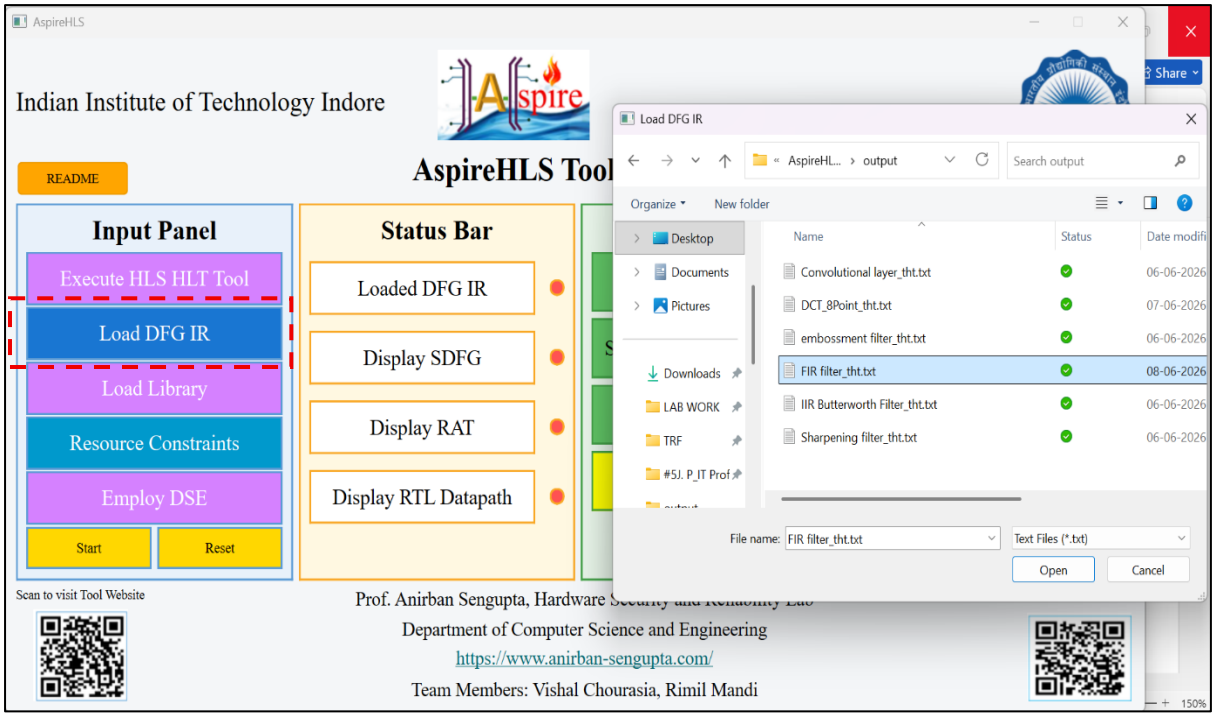
Step 9: The LU transformed processed DFG IR of '**FIR filter.txt**' can be further processed through THT based HLT technique by clicking on '**Apply THT**' button, resulting into THT transformed process DFG as '**FIR filter_tht.txt**' generated and saved in 'Output' folder. The respective THT processed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



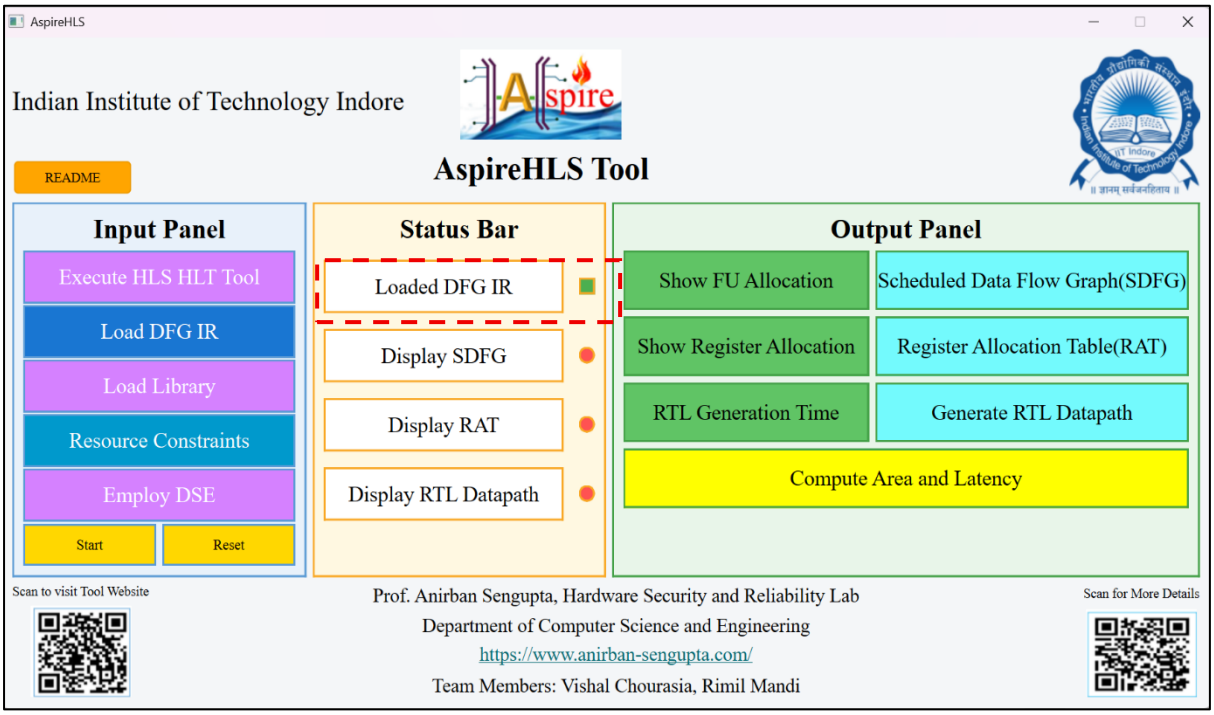

Compiler driven high level transformation during HLS
Inputs
Load Input Application
Outputs (Pre-HLT)
Original DFG
Operation-count pre-HLT
High Level Transformations
Apply THT
Apply ROE
Apply LU
Outputs (Post-HLT)
THT DFG
ROE DFG
LU DFG
Operation-count post-HLT
Message
THT saved successfully.
C:\Users\visha\OneDrive\Desktop\AspireHLS_TOOL and Code\dist\AspireHLS\output\FIR filter_tht.txt
OK


0
[0]
0,*,0
[1]
0,*,0
[3]
0,*,0
[5]
0,*,0
[7]
0,*,0
[9]
0,*,0
[11]
0,*,0
[13]
0,*,0
[15]
0,*,0
[17]
0,*,0
[19]
0,+,1
[2]
3,+,5
[4]
7,+,9
[6]
11,+,13
[8]
15,+,17
[10]
2,+,4
[12]
6,+,8
[14]
10,+,19
[16]
12,+,14
[18]
18,+,16
[20]

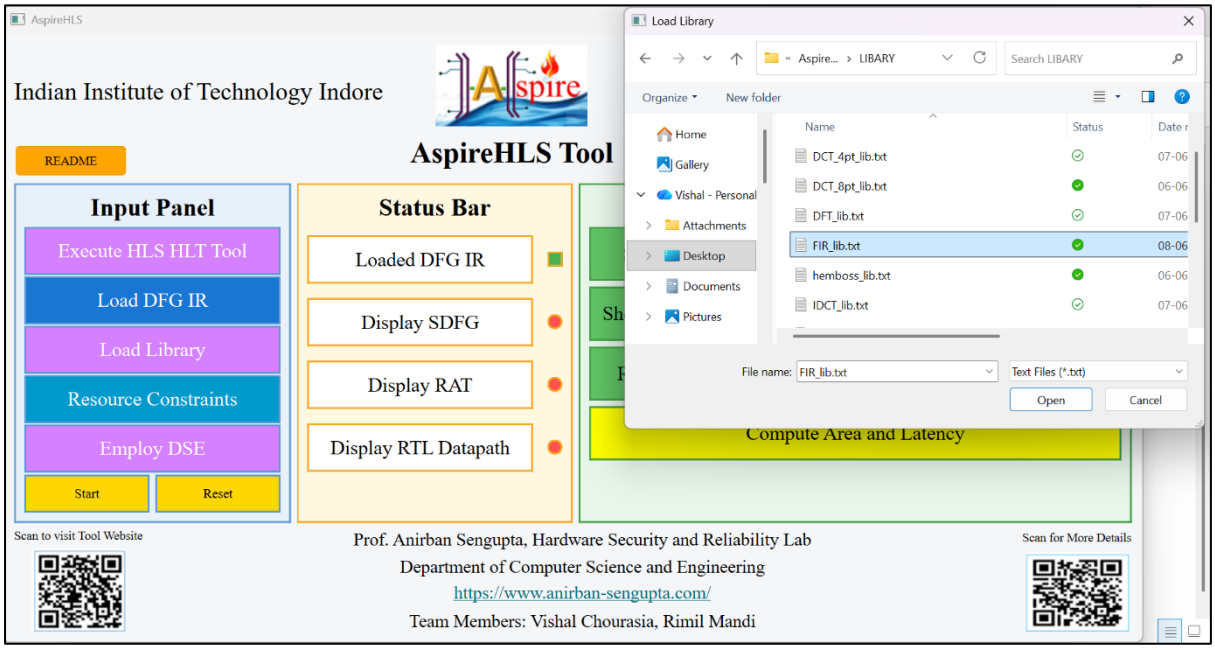
Step 10: we explain subsequent steps of the Tool with the help of '**FIR filter_tht.txt**' as processed DFG IR obtained in Step 9. This generated processed DFG IR file is available in the 'output' folder. The designer needs to load this processed DFG IR file as shown below:



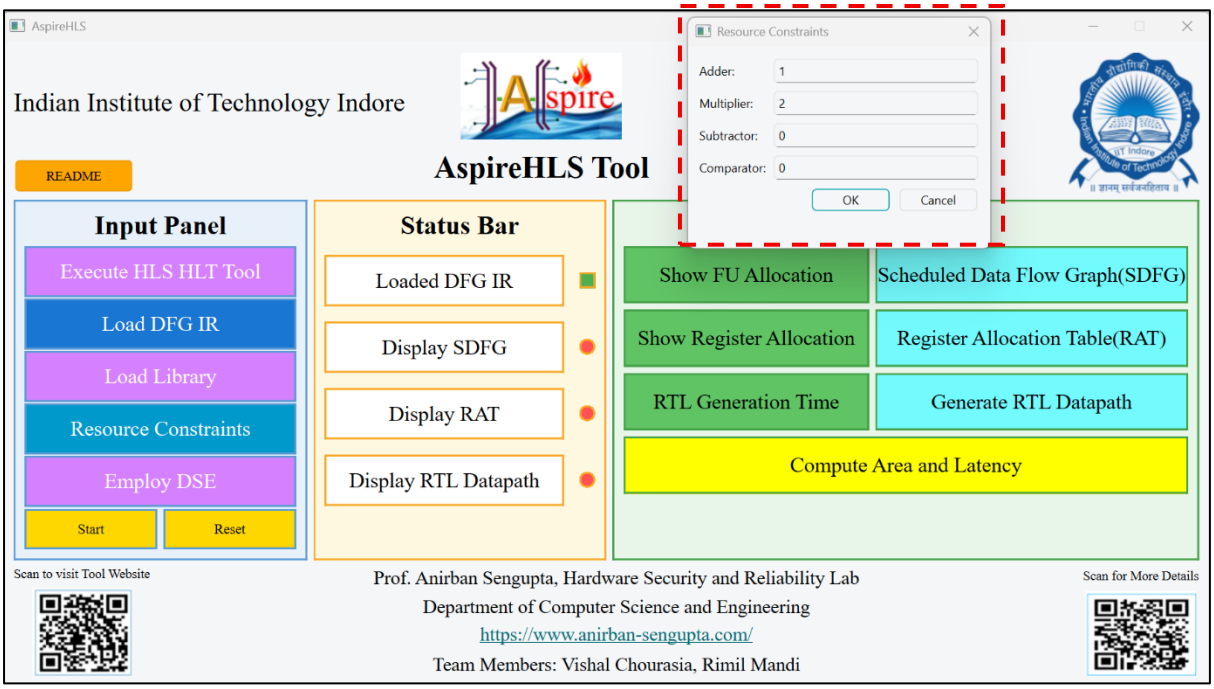
The AspireHLS Tool on loading this processed DFG IR file changes status from red to green (as evident from Status bar) shown below:

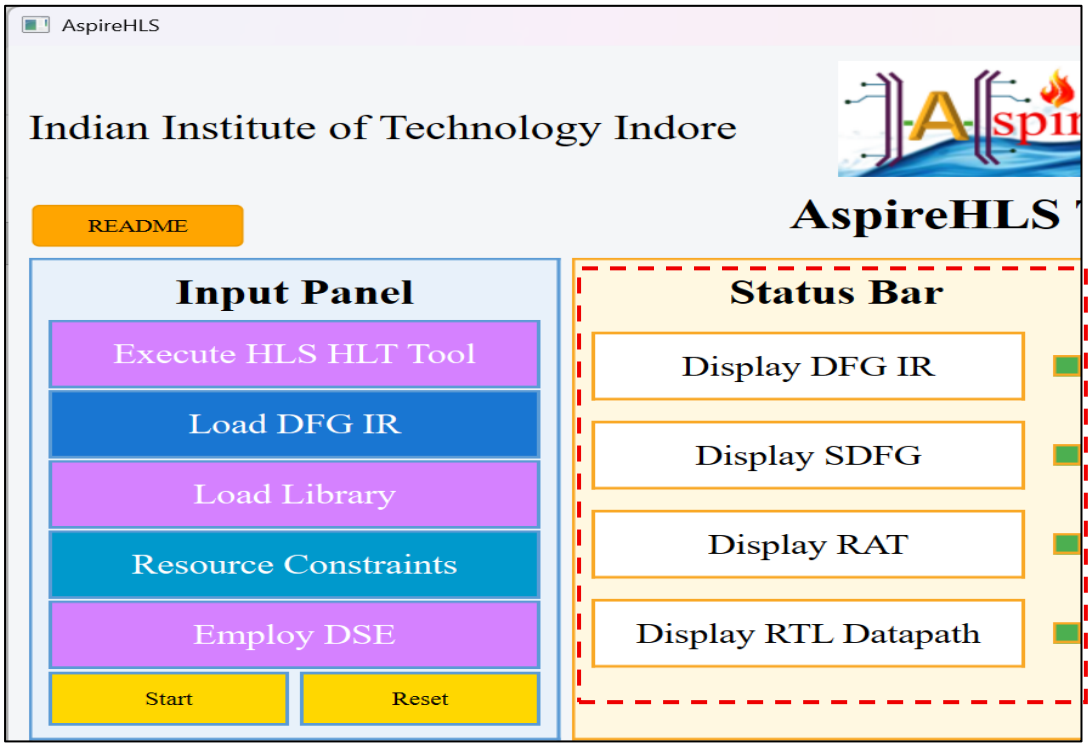


Step 11: Next step is to load the library file corresponding to the application. For example, as shown in the screenshot below, the library file corresponding to FIR filter is loaded.

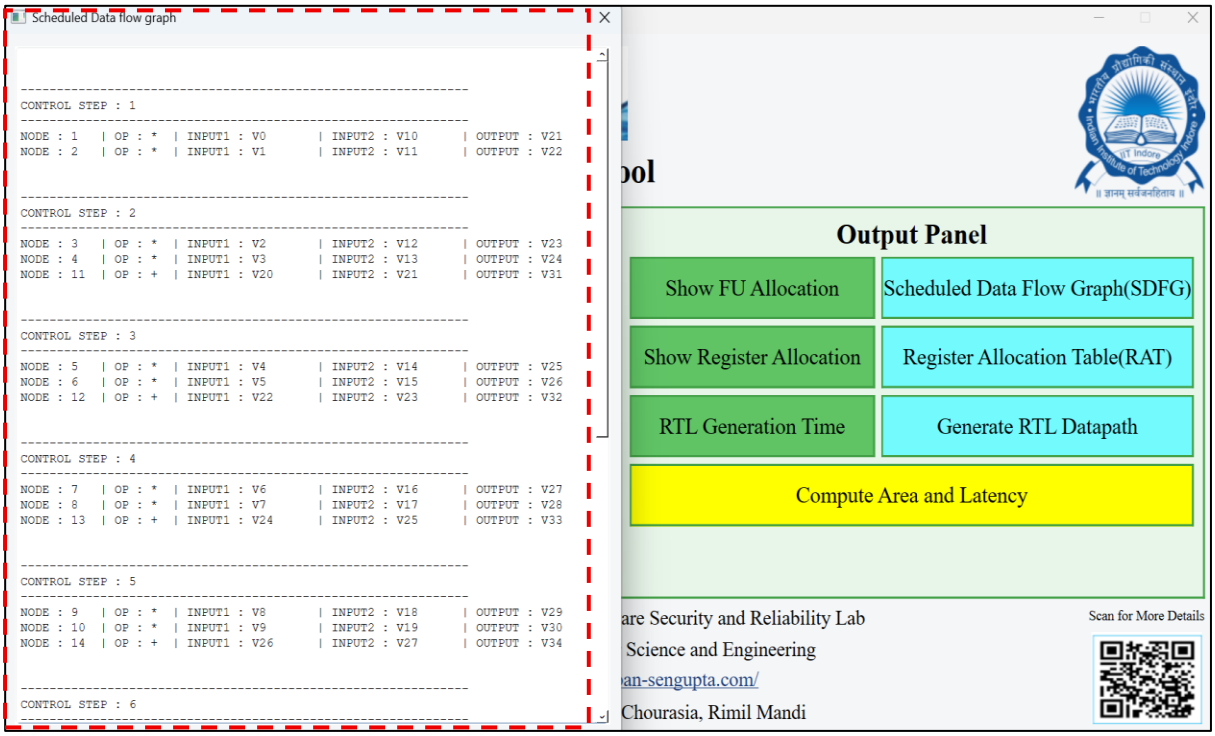


Step 12: Next step is to specify the resource constraints by clicking on the '*Resource Constraints*' button. Alternatively, the designer can employ design space exploration (DSE) to explore the optimal resource configuration based on area-delay trade-off. This can be performed by clicking on the '*Employ DSE*' button. Details about the step is provided in Step 17 onward of this tutorial.
In this step we continue explaining the subsequent process after clicking '*Resource Constraints*' button. On clicking, a pop-up window opens up which provides the option to specify the resource constraints as shown below. The designer (user) next needs to click on the '*Start'* button to process/activate the remaining Tool functions. As soon as the 'Start' button is pressed, the 'Status bar' of the tool turns green as shown below.

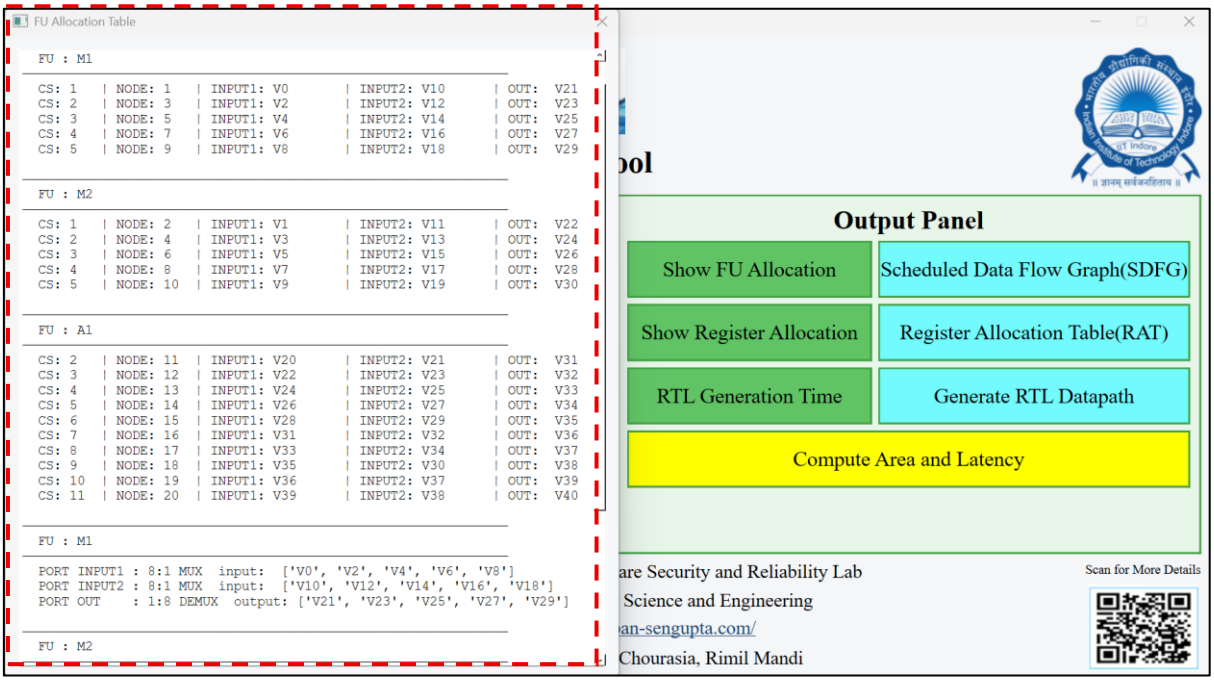




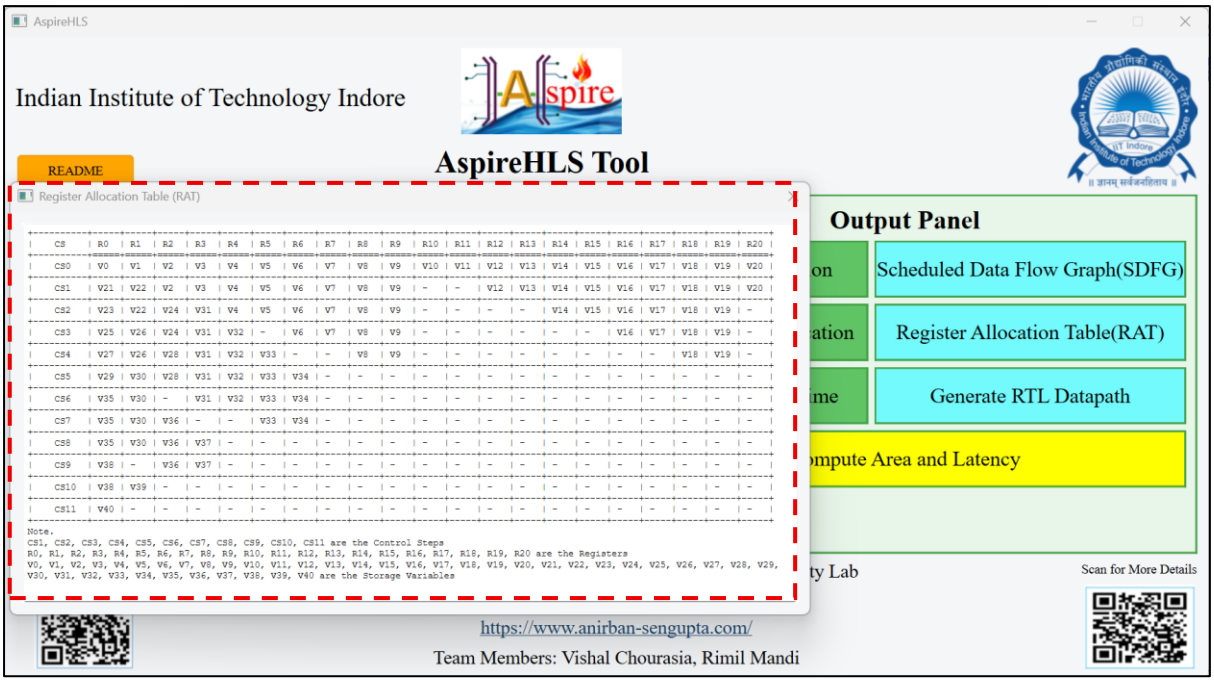
Step 13: The designer can view the scheduled DFG of the application by clicking on the '*Scheduled Data Flow Graph (SDFG)*' button present in the 'Output' panel, as shown below.



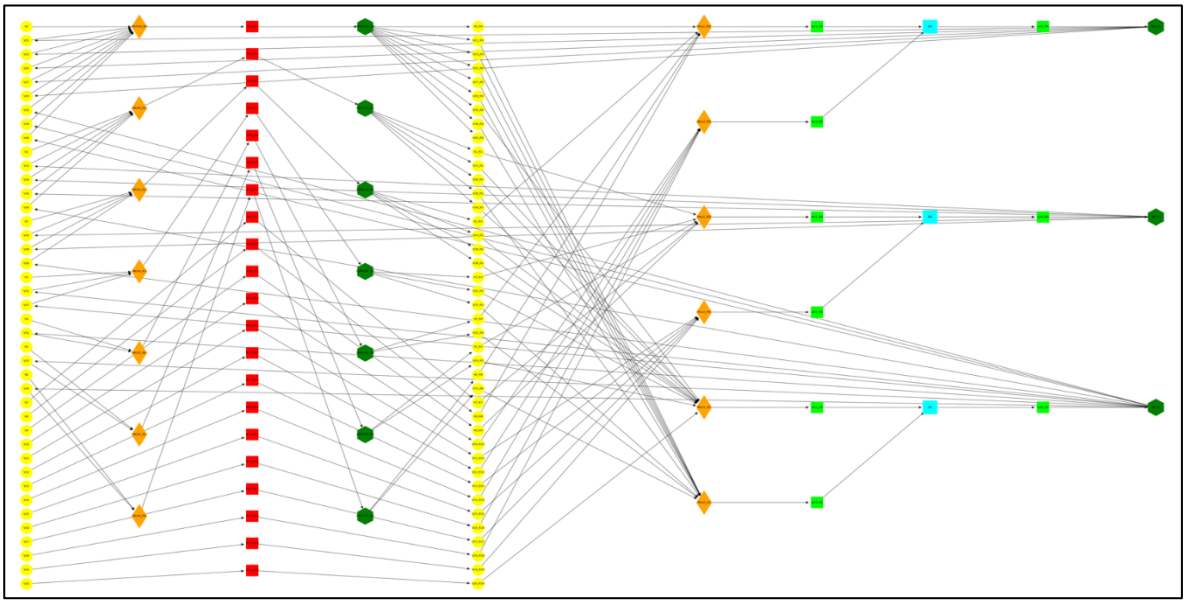
Step 14: The designer can view the FU allocation details by clicking on the '*Show FU Allocation*' button. For example, the FU allocation details of the FIR filter is shown below.

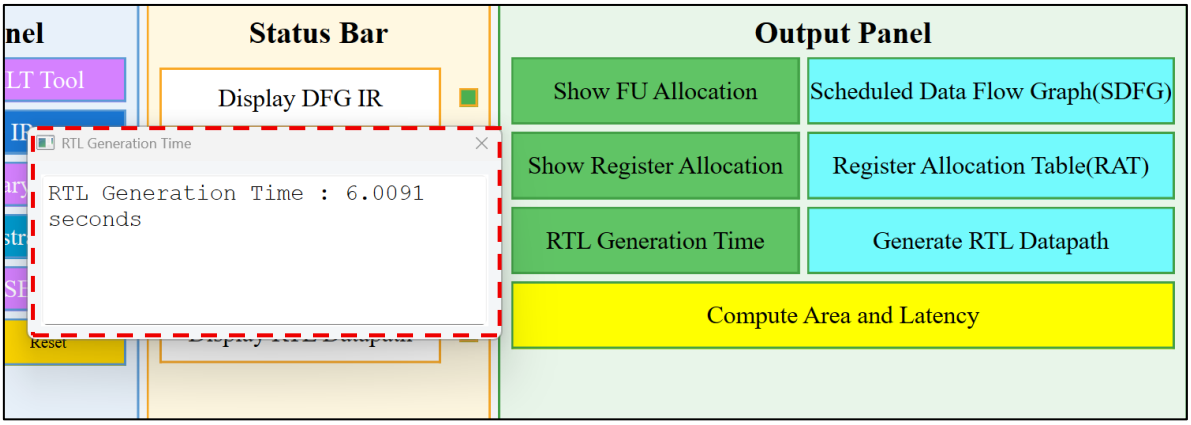


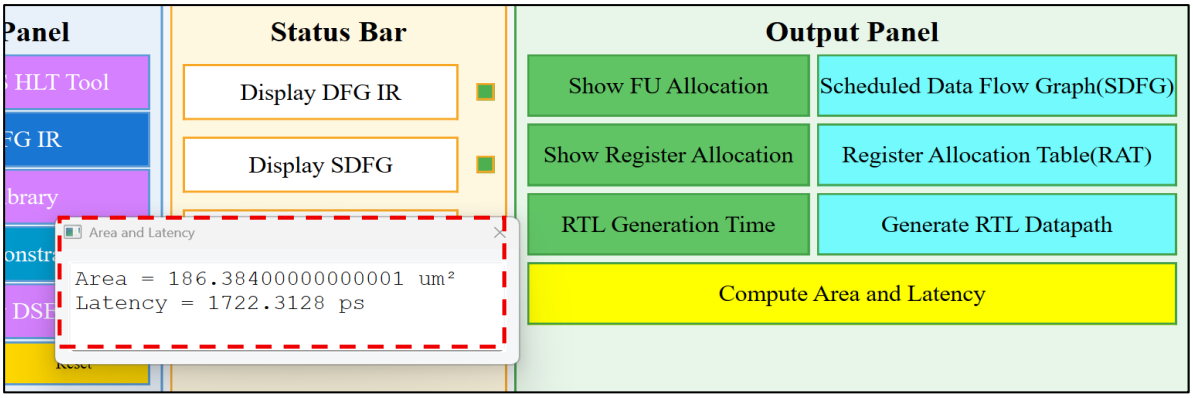
Step 15: The designer can view the register allocation information by clicking on the 'Register Allocation Table (RAT)' button. For example, the RAT of scheduled FIR filter is shown below.



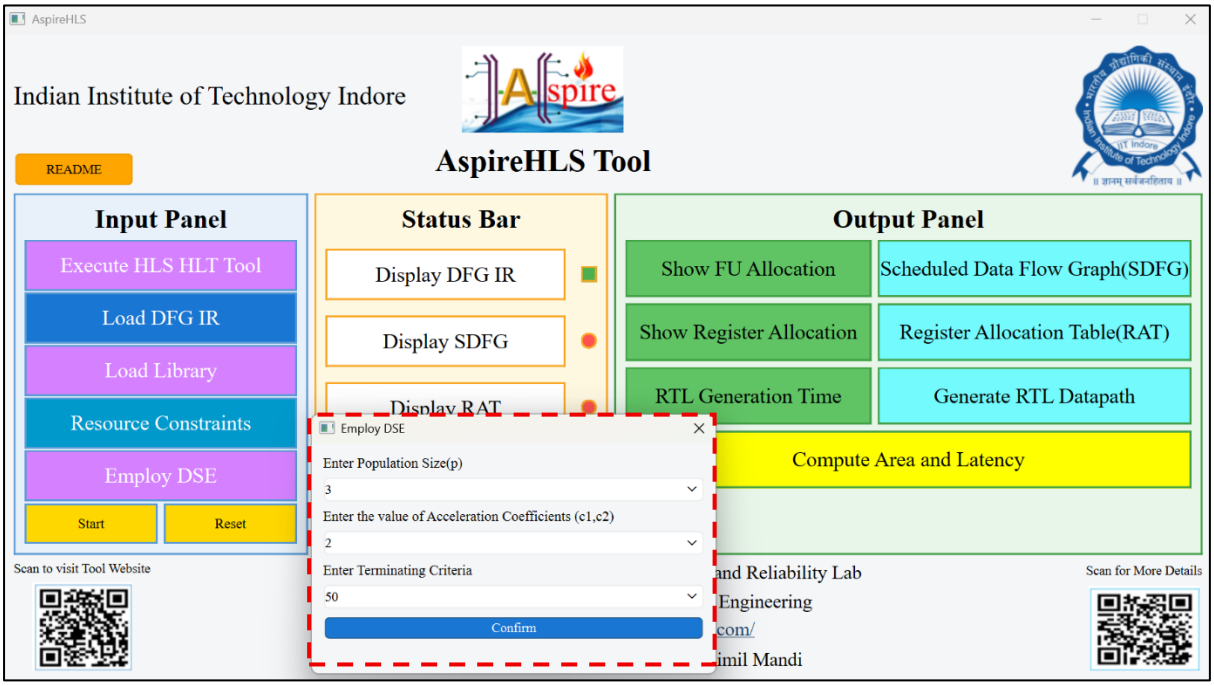
Step 16: The RTL datapath corresponding to the scheduled DFG and RAT can be generated and viewed by clicking on the '*Generated RTL Datapath*' button. For example, the generated RTL datapath corresponding to FIR filter (UF=10) is shown below. The respective runtime to generate the RTL datapath can be viewed by clicking on the '*RTL Generation Time*' button, as shown below. The area and latency of RTL datapath can also be viewed by clicking on the 'Compute Area and Latency' button, as shown below.

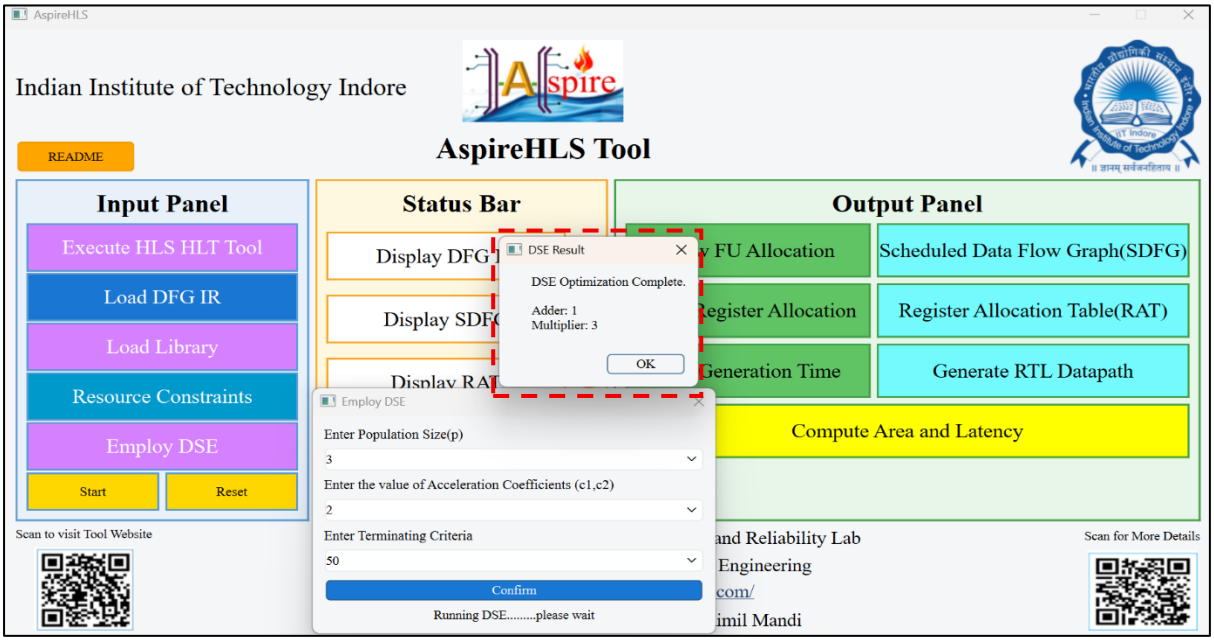






Step 17: The designer on clicking the 'Employ DSE' button, executes the particle swarm optimization (PSO)-DSE system for performing optimization of resource configuration (architecture) based on area-delay trade-off. This opens up a pop-up window with drop-down menu, as shown below, with the following options: (a) Enter population size (p) (b) Enter the value of Acceleration Coefficients (c1,c2) (c) Enter Terminating Criteria. The default value of each are automatically pre-selected. However, designer can choose the values from the drop-down menu.





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

Key Features and Summary of the AspireHLS Tool