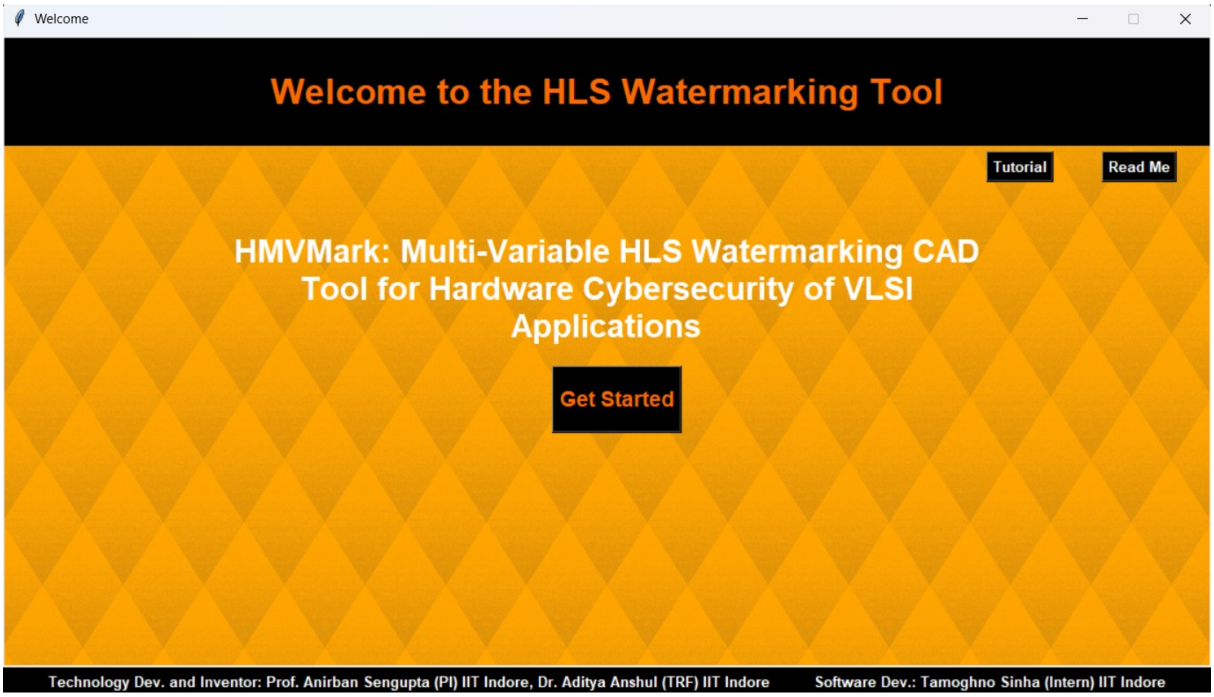
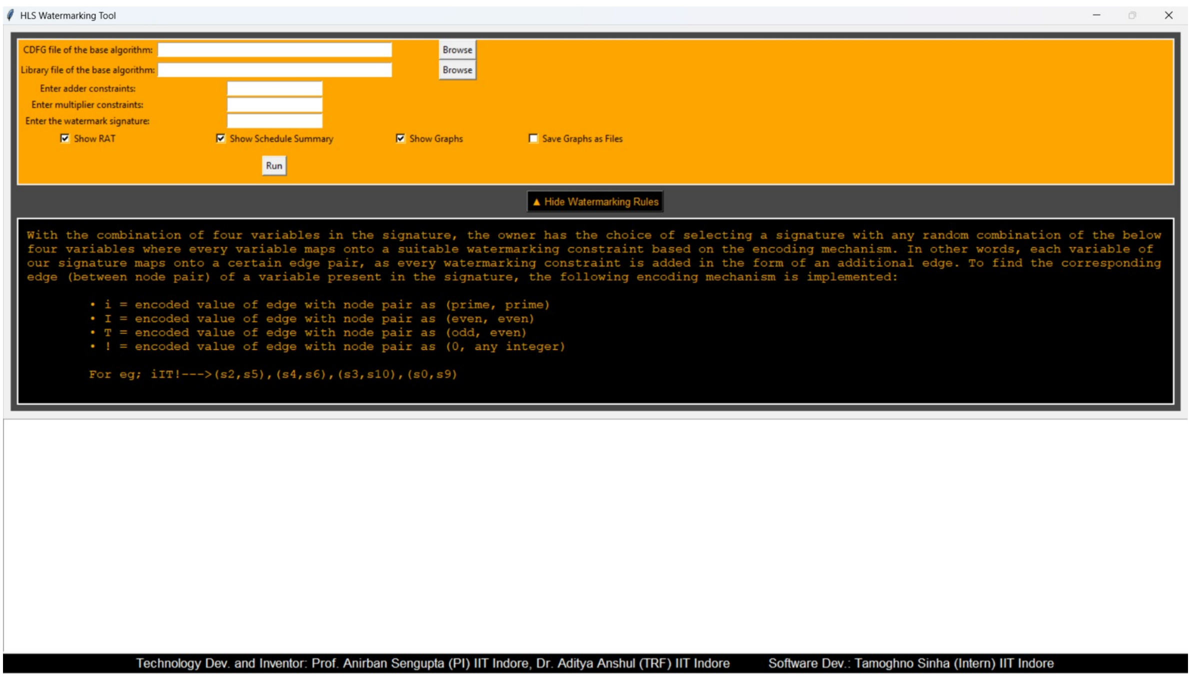
HMVMark Tool Walkthrough

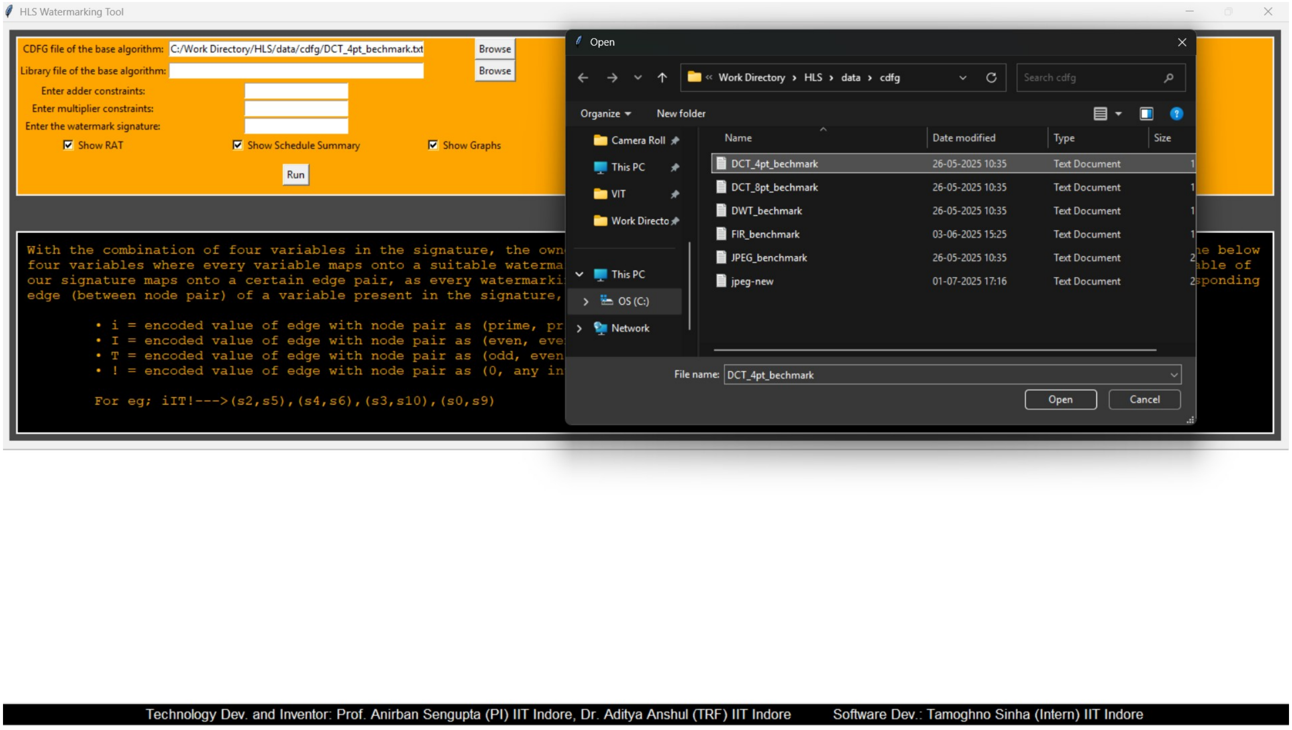
<u>Step 1</u>: Upon running the executable file (*'HMVMark.exe'*), a display window (Welcome panel) appears as shown below. The welcome panel displays the tool name and contains three buttons: (a) Tutorial, (b) Read Me and (c) Get Started. The 'Read Me' button directs to a PDF file containing the necessary details regarding the tool operations.



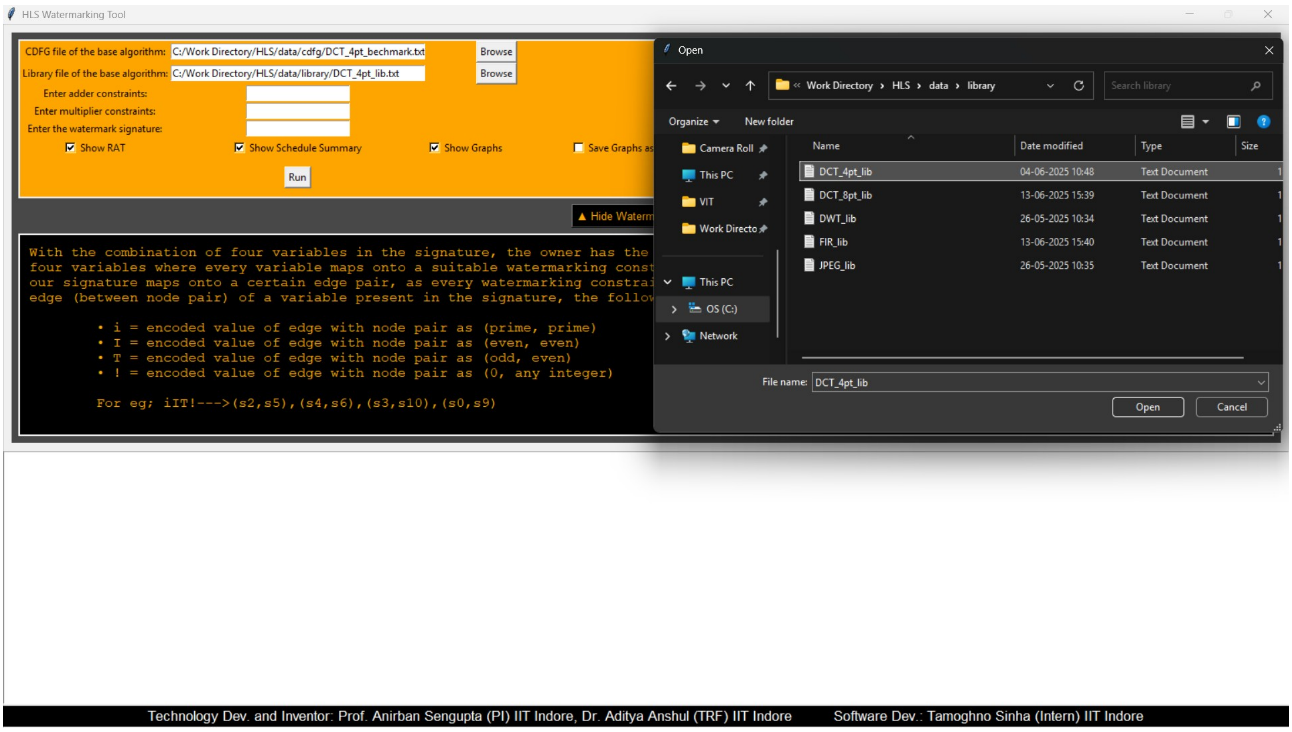
<u>Step 2</u>: The 'Get Started' button leads to the next step i.e., opening the main window of the tool. The main window of the tool comprises of two panels: (a) input panel & (b) output panel. Initially, all the relevant checkboxes are selected by default, and the user is supposed to change it only if necessary.



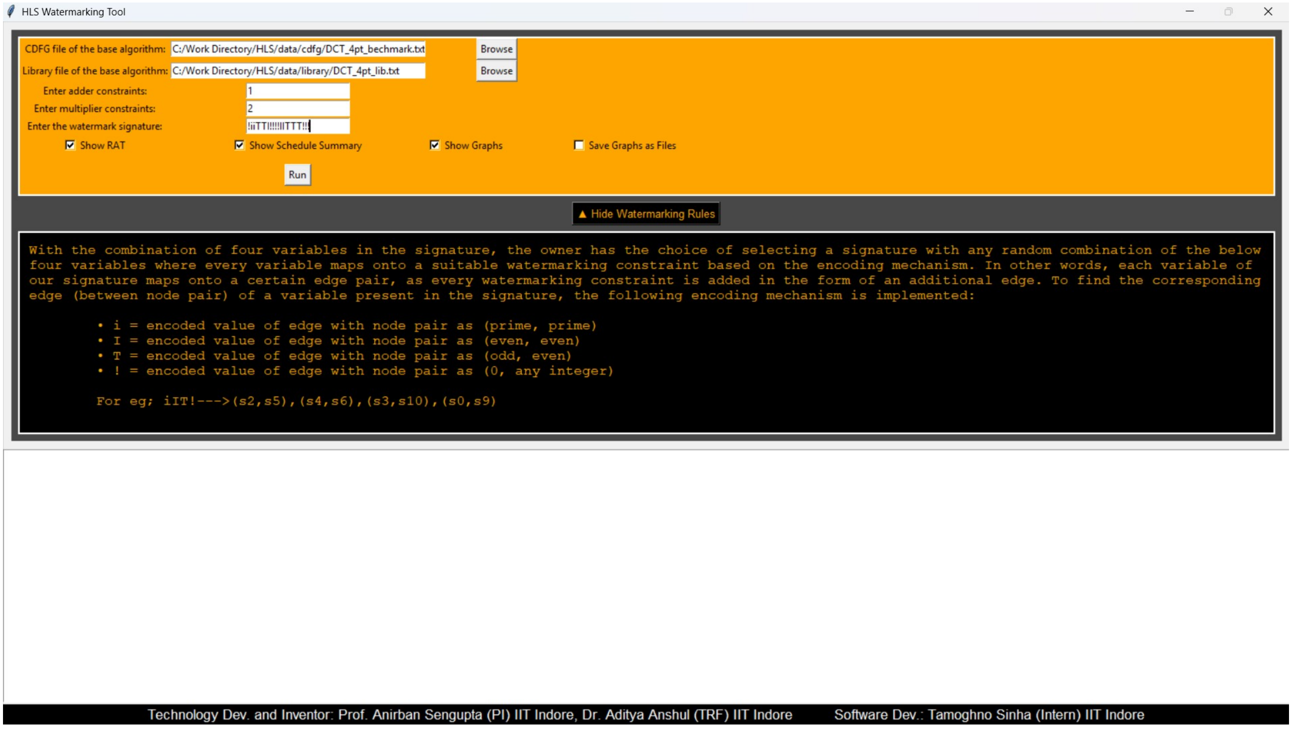
<u>Step 3</u>: In the input panel, click on the ‘Browse’ button on the right side of the CDFG file input field to select the data intensive application’s CDFG text file and load it into the tool.
E.g.; DCT_4pt_benchmark has been selected as shown below.



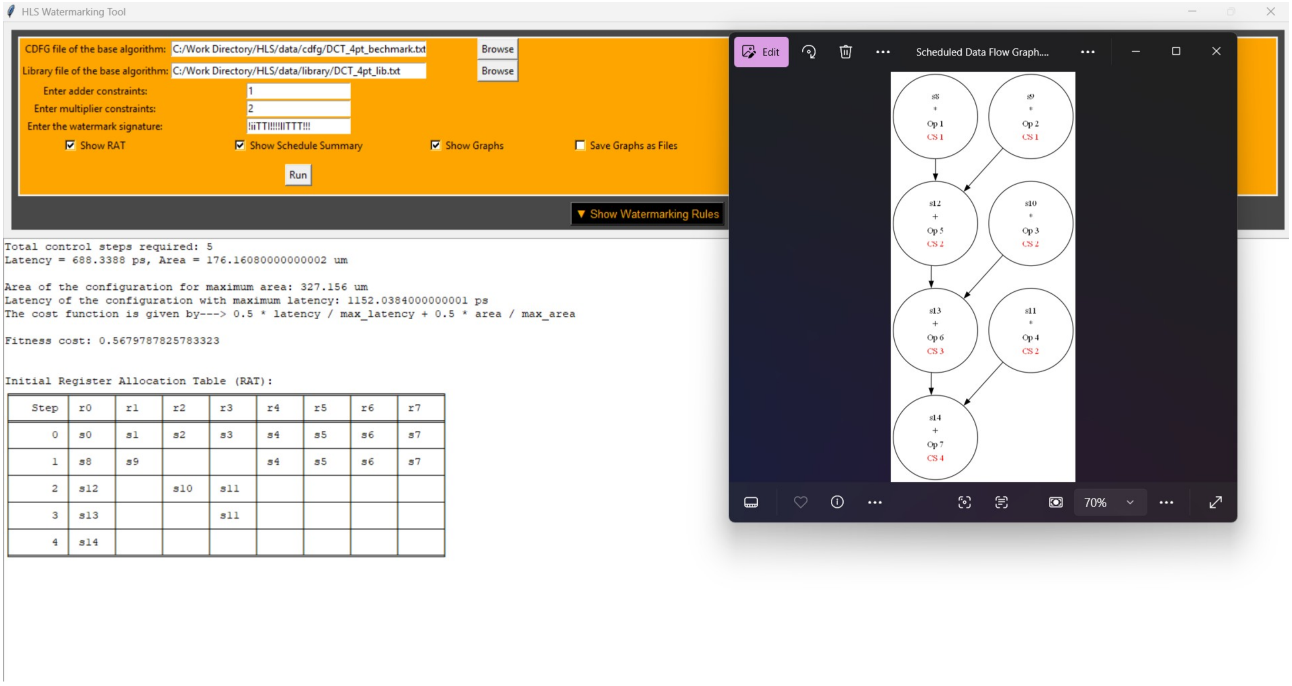
<u>Step 4</u>: Next, click on the ‘Browse’ button to the right of library file input field to select the respective module library text file and provide it as an input to the application.
E.g.; DCT_4pt_lib, the module library for the above CDFG file has been selected as shown below.

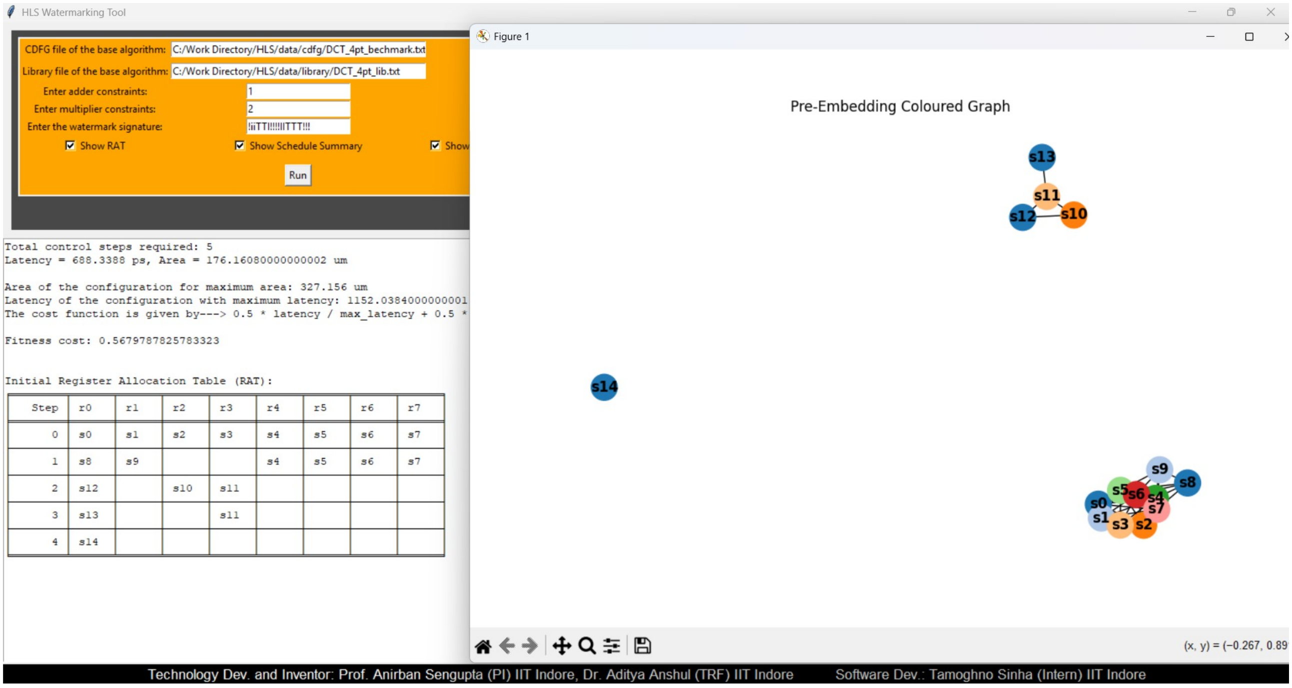


Step 5: Post uploading the CDFG application file and the library file, the user has to enter the adder and multiplier constraints based on which the design (*i.e. SDFG*) is to be scheduled. The user should be aware that providing resource values which exceed the maximum limit set in the module library will throw an error message and notify the user about the set limits. Next, the 4 variable watermark signature is to be provided in its respective input field (*the watermarking rules are mentioned in the panel below*).
E.g.; For DCT_4pt we have set the resource constraints to (1,2), i.e. 1 adder and 2 multipliers.



Step 6: Now, by entering the resources within the specified limit, the user can generate the corresponding scheduled design. After clicking on the 'Run' button, the output panel will calculate the design area and latency corresponding to the input and will generate the cost function, initial register allocation table (RAT). The obtained scheduled data flow graph (SDFG) and pre-watermarking coloured interval graph (CIG) will be displayed on the screen. The user can zoom in to observe the node bindings and corresponding edges more precisely.





<u>**Step 7**</u>**:** After closing the output pre-watermarking CIG, the output panel will embed the watermark signature as per the user's input and display the edge pairs (*The user can match these values with the watermarking rules which are shown below the input panel*), post watermark embedding RAT, modified CIG and the RTL Information table corresponding to the watermark signature's characters. The edge pairs are further sub-divided into pre-existing edges and unique post-watermarking edges for better understanding. The user may note that the red edges present in the modified CIG represent the unique post-watermarking edges.

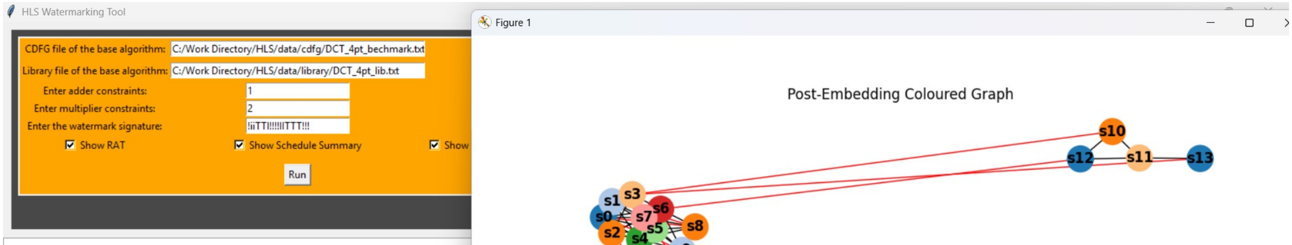

```
(s11,s13)
(s0,s3)
(s0,s6)
(s0,s2)
(s0,s5)
(s1,s6)

New node bindings(after watermark embedding):
(s6,s12)
(s2,s14)
(s5,s14)
(s0,s9)
(s3,s10)
(s1,s8)
(s3,s13)
(s4,s14)
(s0,s8)

Post embedding Register Allocation table(RAT):
```

Step	r0	r1	r2	r3	r4	r5	r6	r7
0	s0	s1	s2	s3	s4	s5	s6	s7
1		s9	s8		s4	s5	s6	s7
2	s12		s10	s11				
3	s13			s11				
4	s14							

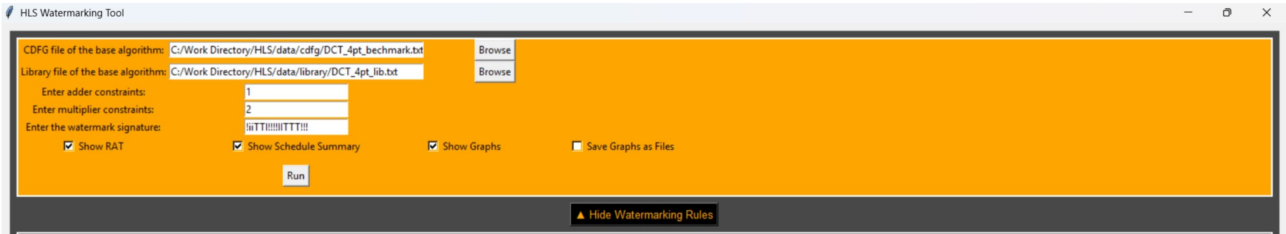
Technology Dev. and Inventor: Prof. Anirban Sengupta (PI) IIT Indore, Dr. Aditya Anshul (TRF) IIT Indore Software Dev.: Tamoghno Sinha (Intern) IIT Indore




```
Storage variable pairs as watermark constraints:
!--->(s0,s8)
i--->(s11,s13)
i--->(s3,s13)
T--->(s5,s8)
T--->(s1,s6)
I--->(s6,s12)
!--->(s0,s9)
!--->(s0,s5)
!--->(s0,s3)
!--->(s0,s2)
I--->(s2,s14)
I--->(s4,s14)
T--->(s3,s10)
T--->(s5,s14)
T--->(s1,s8)
!--->(s0,s6)
!--->(s0,s4)
!--->(s0,s1)


Pre-existing node bindings:
(s0,s1)
(s0,s4)
(s5,s8)
(s11,s13)
(s0,s3)
(s0,s6)
(s0,s2)
(s0,s5)
```

Technology Dev. and Inventor: Prof. Anirban Sengupta (PI) IIT Indore, Dr. Aditya Anshul (TRF) IIT Indore Software Dev.: Tamoghno Sinha (Intern) IIT Indore



With the combination of four variables in the signature, the owner has the choice of selecting a signature with any random combination of the below four variables where every variable maps onto a suitable watermarking constraint based on the encoding mechanism. In other words, each variable of our signature maps onto a certain edge pair, as every watermarking constraint is added in the form of an additional edge. To find the corresponding edge (between node pair) of a variable present in the signature, the following encoding mechanism is implemented:

- i = encoded value of edge with node pair as (prime, prime)
- I = encoded value of edge with node pair as (even, even)
- T = encoded value of edge with node pair as (odd, even)
- ! = encoded value of edge with node pair as (0, any integer)

For eg; iIT!--->(s2,s5),(s4,s6),(s3,s10),(s0,s9)

3	s13			s11				
4	s14							

```
RTL Information
Total registers: 8
Total storage variables: 15
Total adders: 1
Total multipliers: 2
Total latches: 9
Mux--->
4:1_mux: 4
2:1_mux: 10
Demux--->
1:4_demux: 3
1:2_demux: 8
```

Technology Dev. and Inventor: Prof. Anirban Sengupta (PI) IIT Indore, Dr. Aditya Anshul (TRF) IIT Indore Software Dev.: Tamoghno Sinha (Intern) IIT Indore

<u>Conclusion</u>: The user has now obtained the post watermarked RAT, along with the entire RTL Information history. The timer at the bottom records the execution time of the tool and can be used to measure its performance. By clicking on the 'Reset' button, the user may clear the output terminal panel or may proceed to provide a new set of application and library files as input for watermark embedding. Once the user is satisfied, the tool can be closed by simply clicking the cross (x) button at the top-right corner of the interface.