SOFTDEEP: HLS based IP Protection Tool (for Secure Optimal Fault Tolerant Designs with Embedded Encrypted Protein Signature)

By:

Anirban Sengupta (Professor & PI, CSE, IIT Indore)
Rahul Chaurasia (TRF Post-Doc, IIT Indore)
Contact: asengupt@iiti.ac.in

Available publicly for download/installation @
https://cadforassurance.org/tools/ip-ic-protection/softdeep/

IEEE Council on Electronic Design Automation (CEDA) and IEEE Hardware Security and Trust Technical Committee (HSTTC)

- **Introduction ::** Hardware Intellectual Property (IP) cores form key components of system-on-chips (SoC) designs used in wide range of consumer electronics (CE) systems, mission critical systems, autonomous vehicles and medical devices etc. [1].

- It is crucial to assure the detection and/or tolerance of the reusable hardware IP cores against transient faults, as such designs may lead to incorrect output due to vulnerabilities emanating from single event upsets (SEU) [2].
- In the modern IC design cycle, multiple offshore entities (design houses or foundries) are involved to cut down the overall design cost, design complexity and time-to-market. This has posed the serious hardware threats of IP piracy or counterfeiting [3]-[5].
- For high computational and data intensive applications, it becomes crucial to design their dedicated hardware IPs.
- Therefore, our developed tool can be useful for IP designers and SoC design houses.

- **Problem Statement ::** To develop an high-level synthesis (HLS) based IP Protection tool that can be used for generating Secure Optimal Fault Tolerant Hardware IP Designs for data intensive applications with seamless detective control against IP piracy.

- **Novelties ::** A novel molecular biometric based hardware security technique based on protein molecule sequence (where an IP vendor selected protein sequence comprising of 20 unique amino acid combinations is used for signature generation) is presented for the first time to secure IPs.

- Cutting edge innovative technology for designing robust hardware IP cores using facial biometric based encrypted protein molecular biometrics. Thus, the proposed technique incorporates two classes of biometrics to ensure highly robust and unique authentication.
- A novel secure HLS based design flow of unique encrypted protein molecular biometric signature generation and covertly implantation into the design to secure hardware coprocessors is presented.
- Breakthrough technology that can be used to design secure and optimal fault tolerant data intensive hardware coprocessors for DSP, multimedia and machine learning applications.

- Development of standalone HLS CAD tool/software for scientific community.
- Prototyping of optimal and secure k-Cycle transient fault tolerant datapath processor design for digital signal processing (DSP) cores (DCT core, FIR core and IIR core), and multimedia cores (JPEG-codec) in Intel Quartus Tool/ Intel HLS compiler tool.

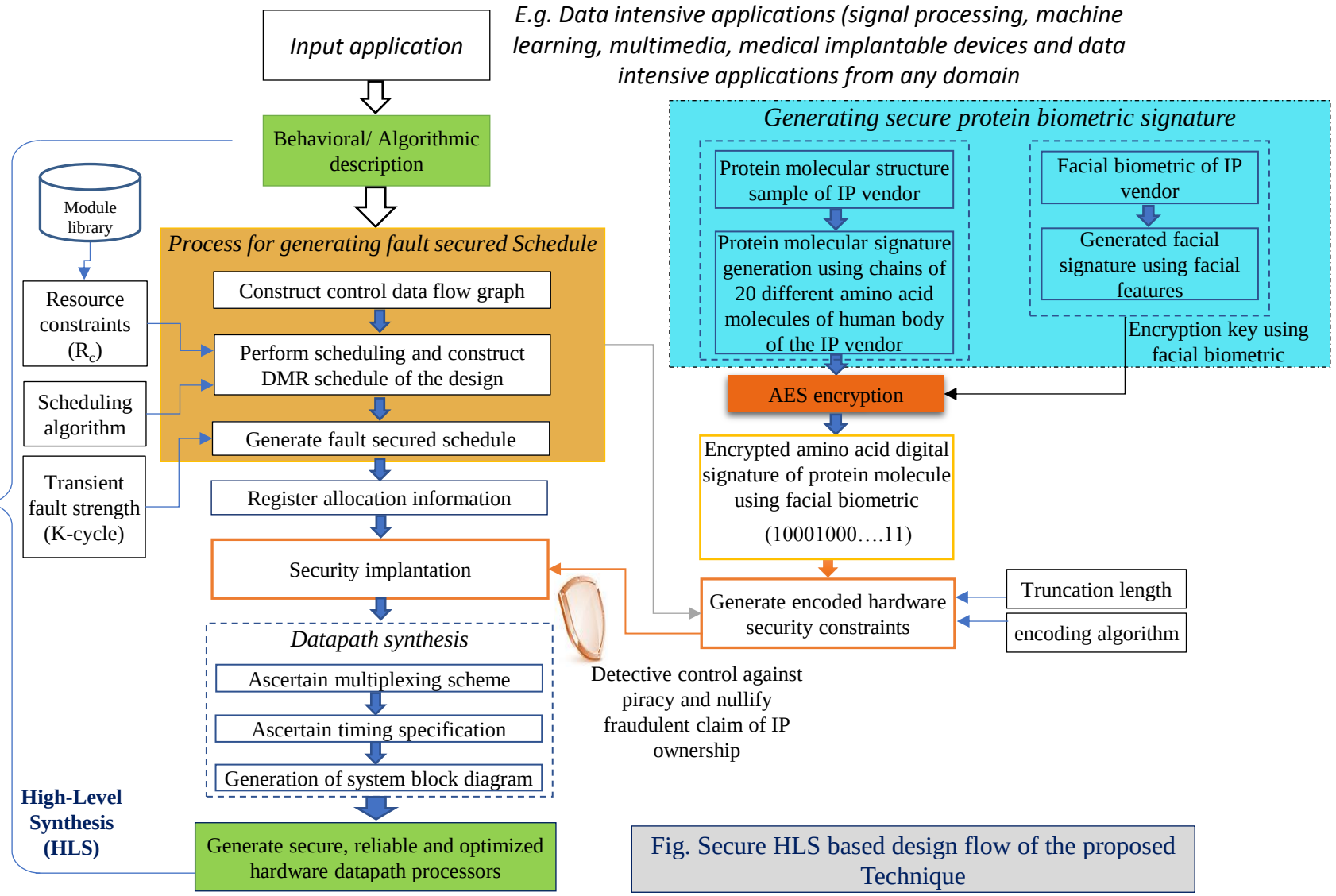
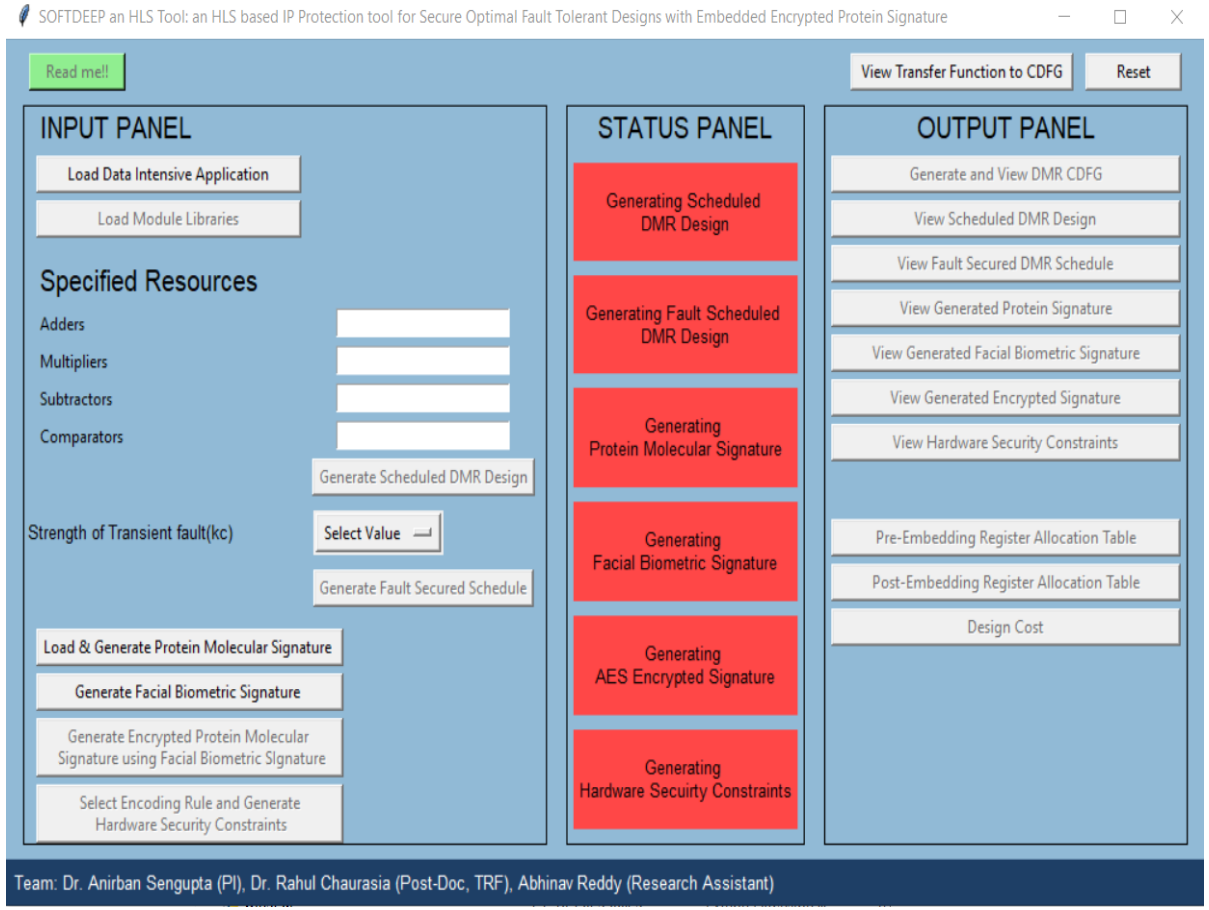
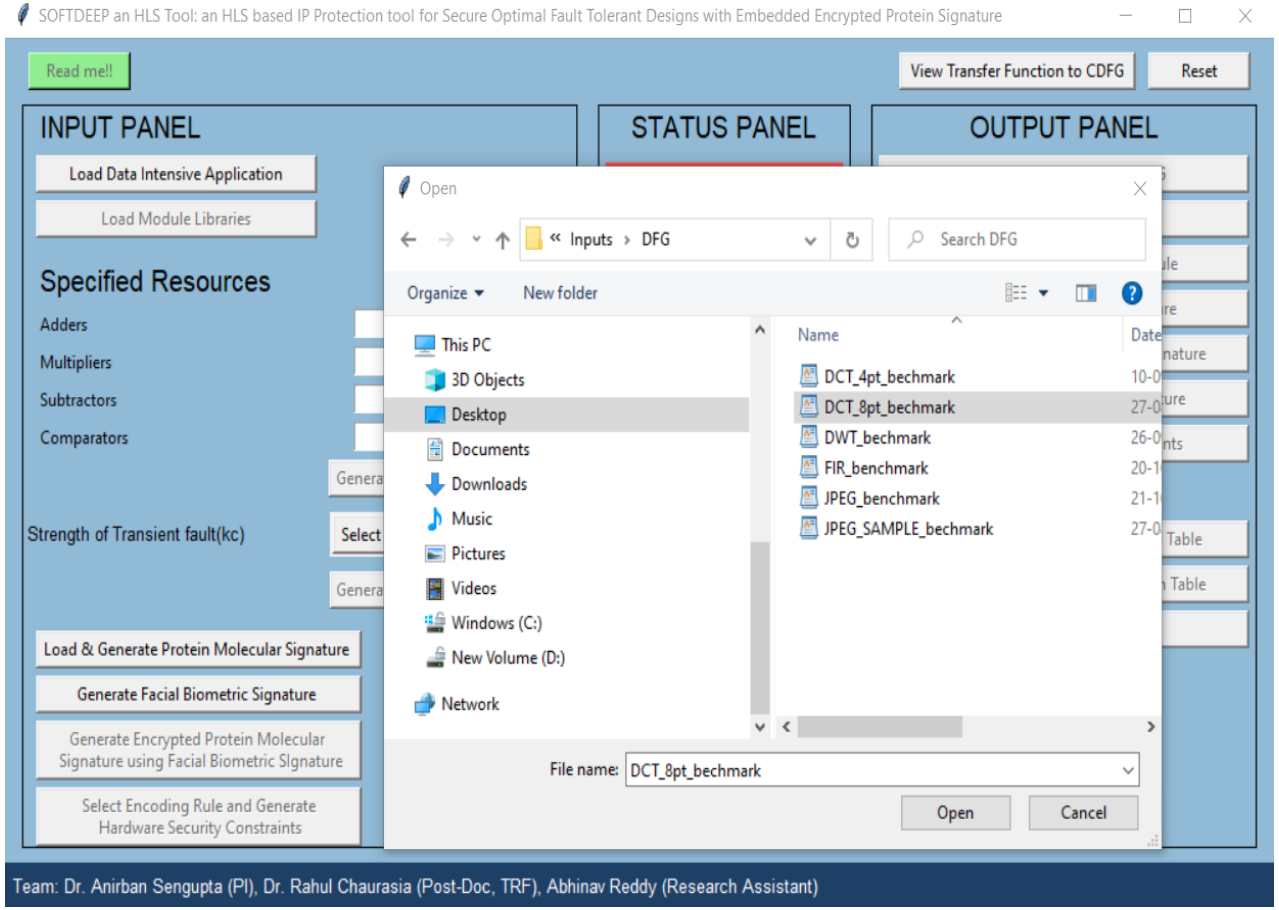


Fig. Secure HLS based design flow of the proposed Technique

STEP-1: To activate the tool, firstly, user needs to open the 'Readme' file (provides the summary and background of the tool to the user), this lets the Tab 'Readme!!' turns GREEN and also the Tab 'Load Data Intensive Application' gets enabled. Further, two other independent module Tabs 'Load & Generate Protein Molecular Signature' and 'Generate Facial Biometric Signature' also gets enabled as shown below. Now, user/IP vendor can access the 'INPUT PANEL' of the tool.

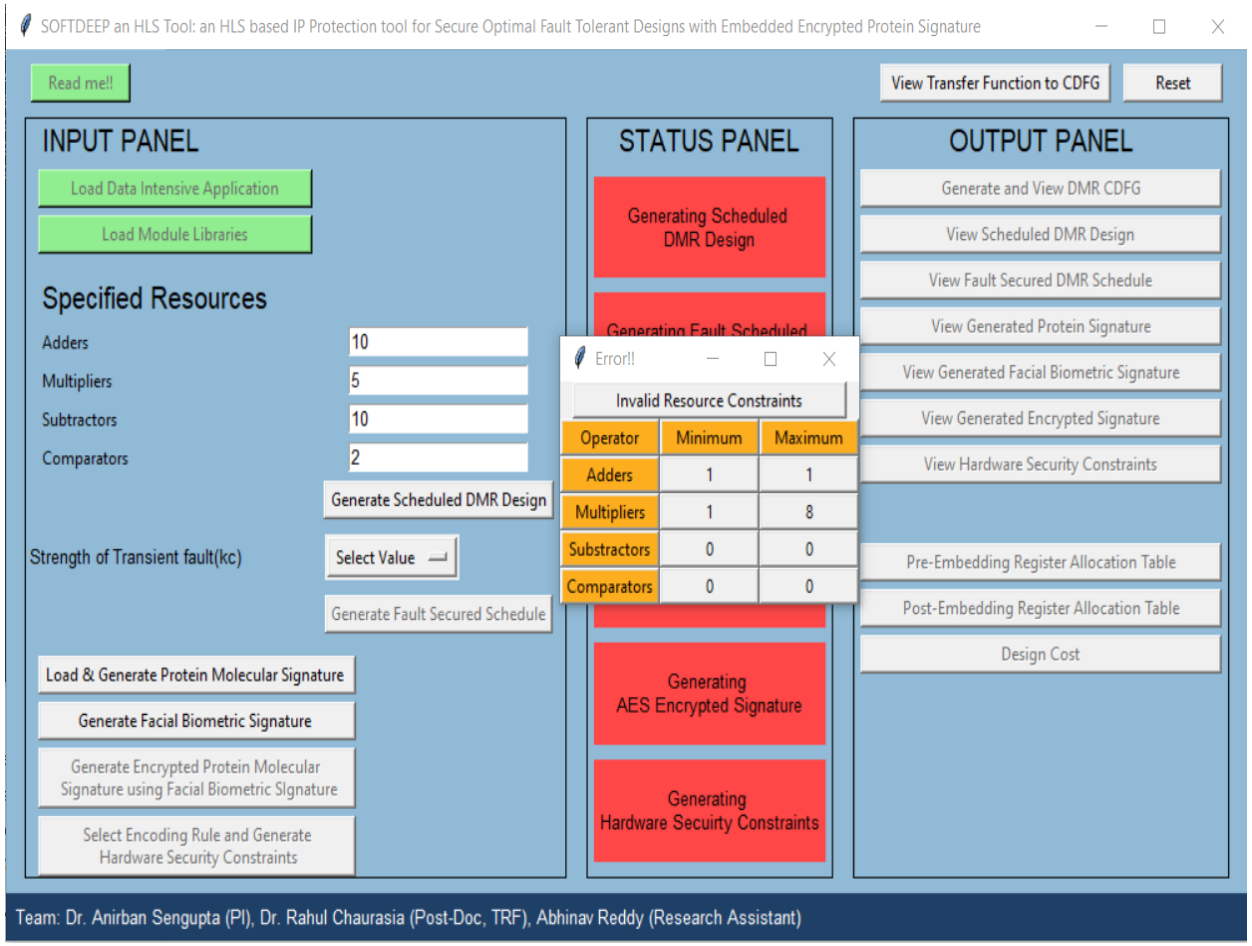
- Here user is asked to load data intensive application (sample application for which secure and optimal K-cycle (Kc) fault detectable design with piracy detective control is to be generated. *Note: For example, DCT_8pt_benchmak has been selected here*).

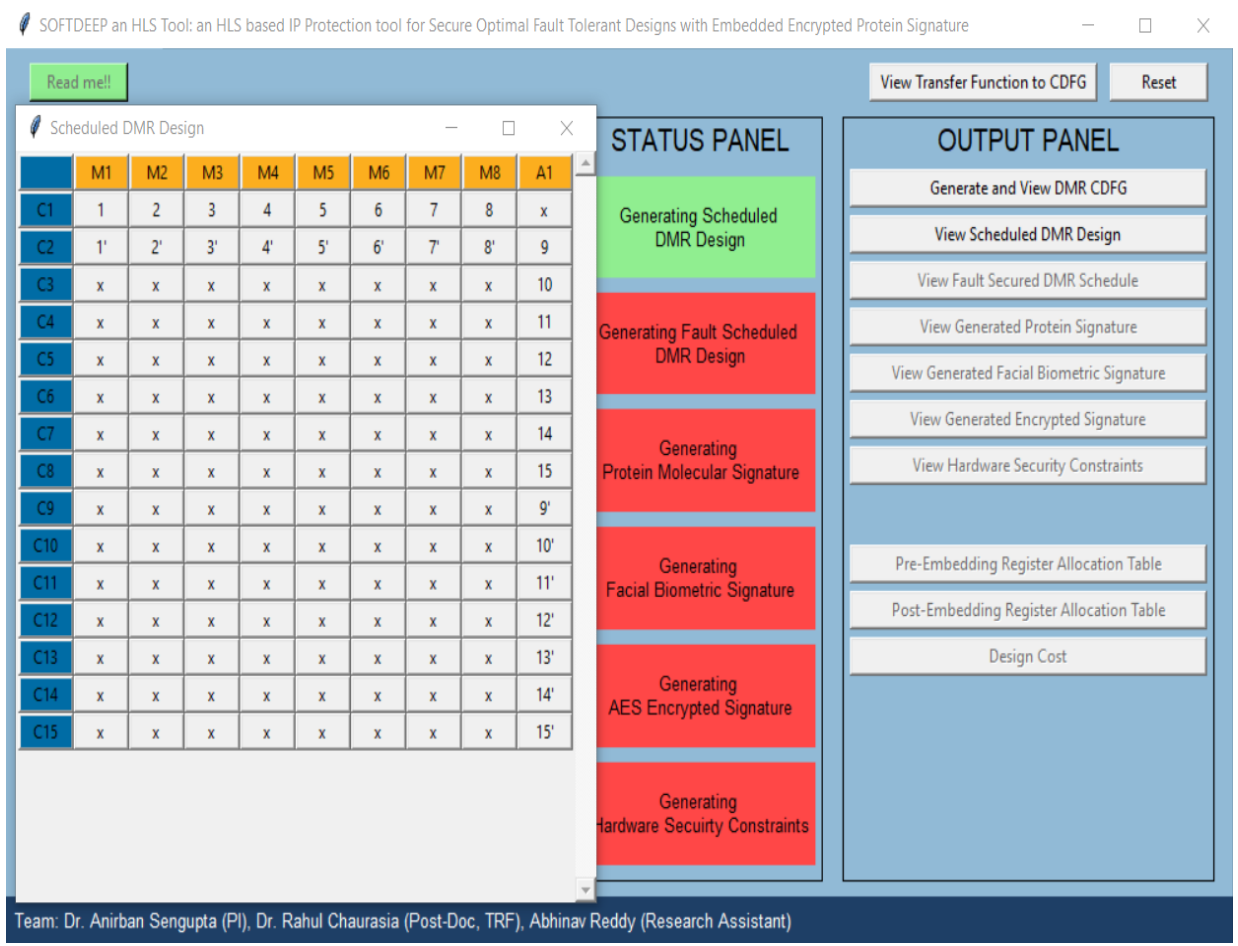




STEP-2: Next, post loading module library (15 nm open cell library) its Tab turns GREEN as shown below. Next, the user is asked to enter/specify the resources based on which design is to be scheduled/generated. *Note:* if user enters the resource configuration exceeding the limits of Min/Max available resources in the module library, then the tool throws an error saying 'Invalid Resource Constraints'.

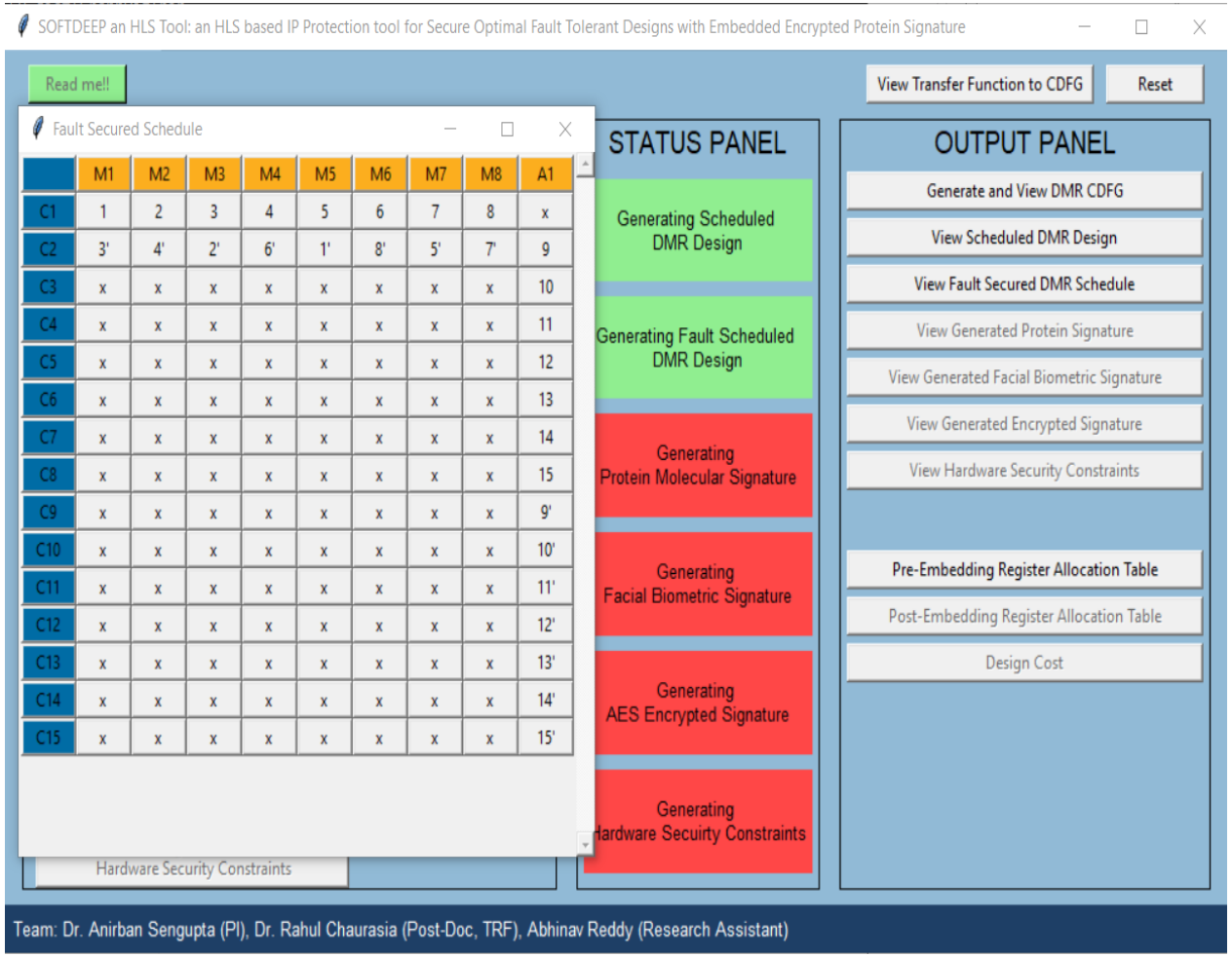
- User can view the generated 'dual modular redundant (DMR) CDFG'/scheduled DMR design (in the form of table) by clicking on it. The output screens are shown below:



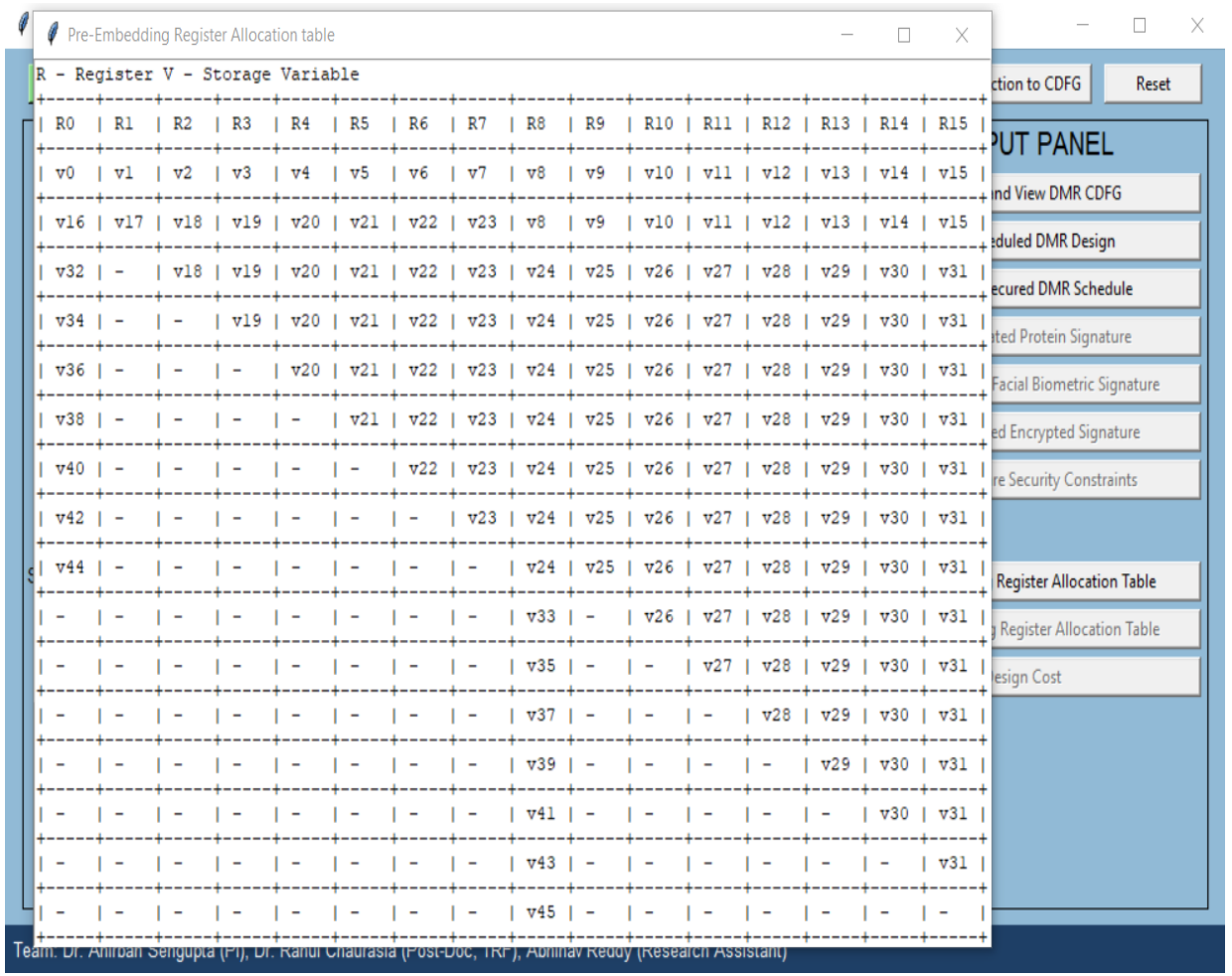


STEP-3: Next, post selecting the transient fault strength (considering in the range 1 to 3 for single/multi-cycle transient), user can generate fault secured schedule for the application. The fault secured schedule for *DCT_8pt_benchmak* is shown below.

- Further, user can also view 'Pre-Embedding Register Allocation table' corresponding to fault secured scheduled design. The output screen is shown below:

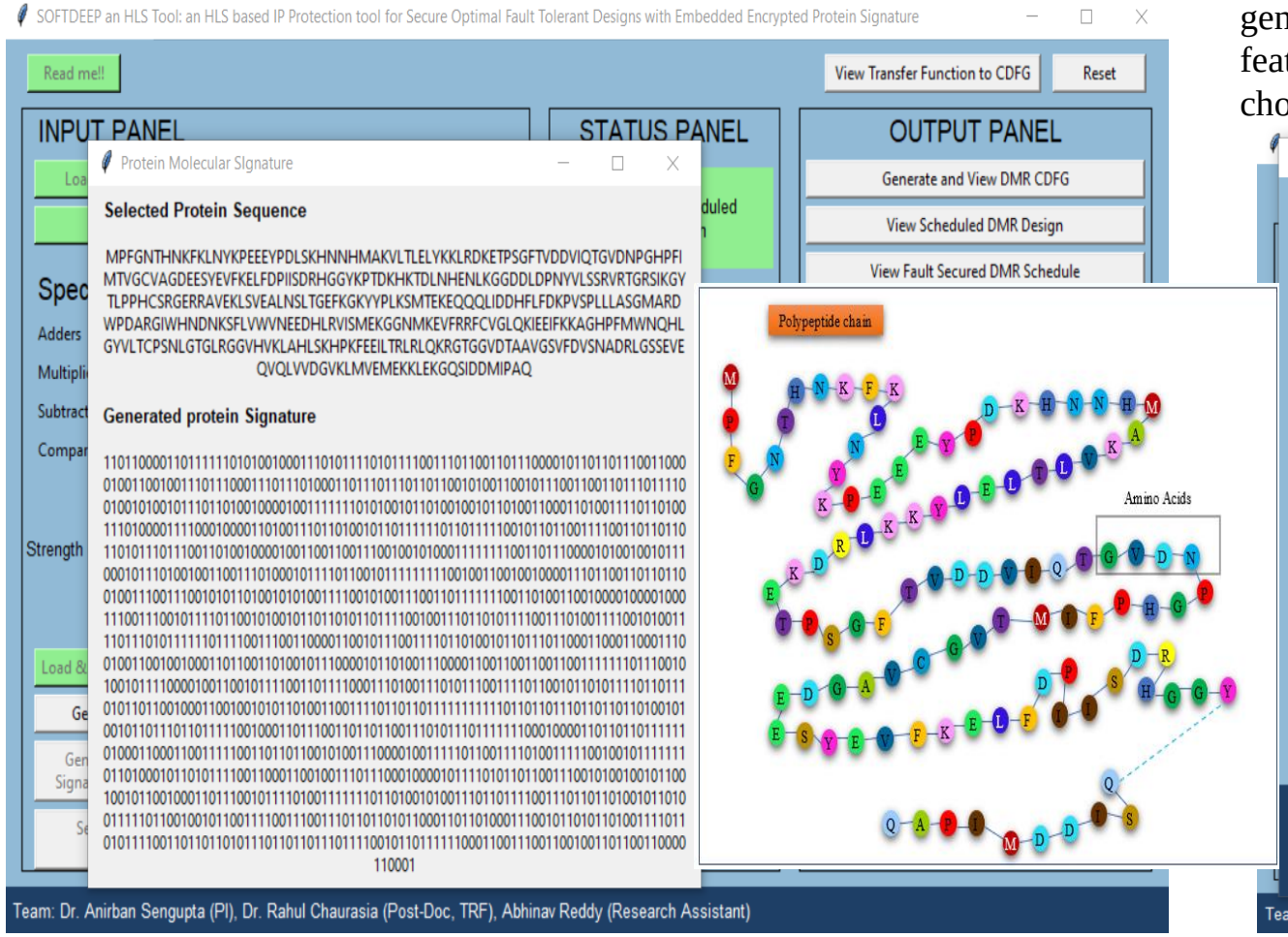


	M1	M2	M3	M4	M5	M6	M7	M8	A1
C1	1	2	3	4	5	6	7	8	x
C2	3'	4'	2'	6'	1'	8'	5'	7'	9
C3	x	x	x	x	x	x	x	x	10
C4	x	x	x	x	x	x	x	x	11
C5	x	x	x	x	x	x	x	x	12
C6	x	x	x	x	x	x	x	x	13
C7	x	x	x	x	x	x	x	x	14
C8	x	x	x	x	x	x	x	x	15
C9	x	x	x	x	x	x	x	x	9'
C10	x	x	x	x	x	x	x	x	10'
C11	x	x	x	x	x	x	x	x	11'
C12	x	x	x	x	x	x	x	x	12'
C13	x	x	x	x	x	x	x	x	13'
C14	x	x	x	x	x	x	x	x	14'
C15	x	x	x	x	x	x	x	x	15'

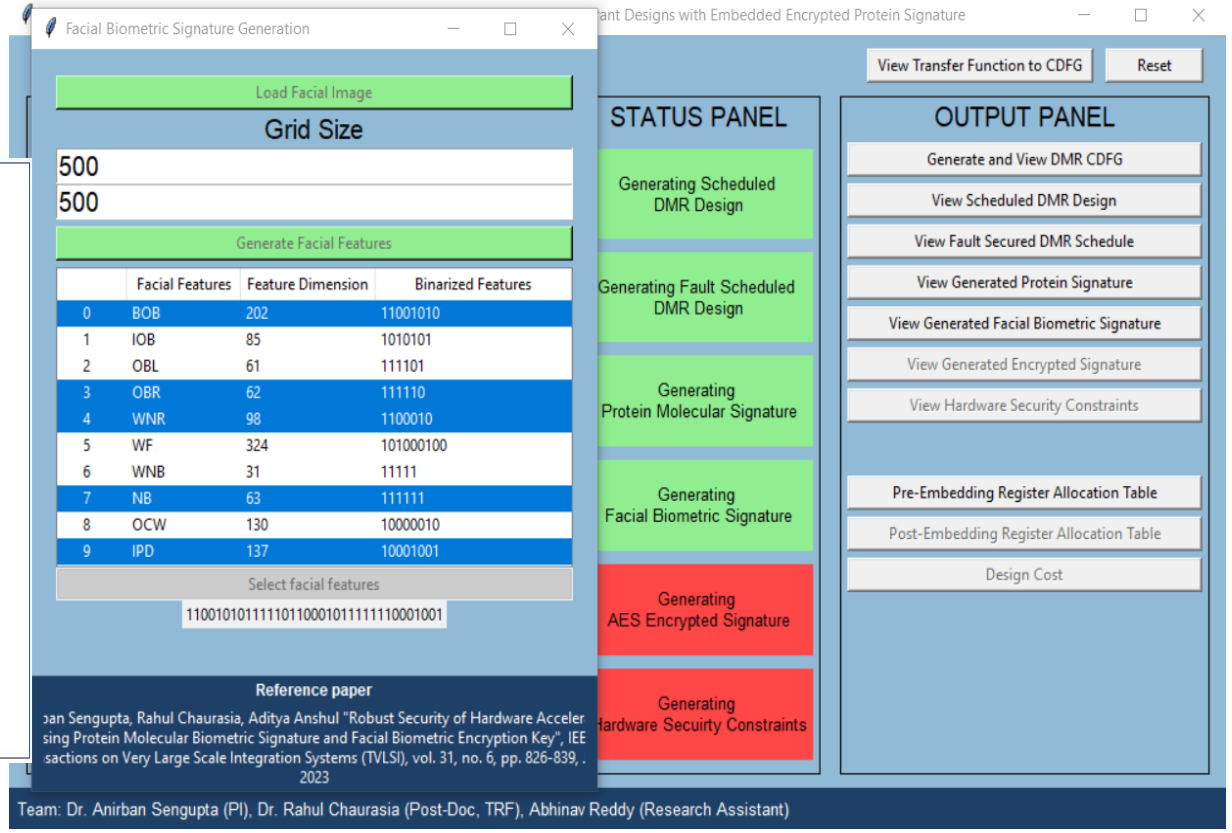


R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15
v0	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15
v16	v17	v18	v19	v20	v21	v22	v23	v8	v9	v10	v11	v12	v13	v14	v15
v32	-	v18	v19	v20	v21	v22	v23	v24	v25	v26	v27	v28	v29	v30	v31
v34	-	-	v19	v20	v21	v22	v23	v24	v25	v26	v27	v28	v29	v30	v31
v36	-	-	-	v20	v21	v22	v23	v24	v25	v26	v27	v28	v29	v30	v31
v38	-	-	-	-	v21	v22	v23	v24	v25	v26	v27	v28	v29	v30	v31
v40	-	-	-	-	-	v22	v23	v24	v25	v26	v27	v28	v29	v30	v31
v42	-	-	-	-	-	-	v23	v24	v25	v26	v27	v28	v29	v30	v31
v44	-	-	-	-	-	-	-	v24	v25	v26	v27	v28	v29	v30	v31
-	-	-	-	-	-	-	-	v33	-	v26	v27	v28	v29	v30	v31
-	-	-	-	-	-	-	-	v35	-	-	v27	v28	v29	v30	v31
-	-	-	-	-	-	-	-	v37	-	-	-	v28	v29	v30	v31
-	-	-	-	-	-	-	-	v39	-	-	-	-	v29	v30	v31
-	-	-	-	-	-	-	-	v41	-	-	-	-	-	v30	v31
-	-	-	-	-	-	-	-	v43	-	-	-	-	-	-	v31
-	-	-	-	-	-	-	-	v45	-	-	-	-	-	-	-

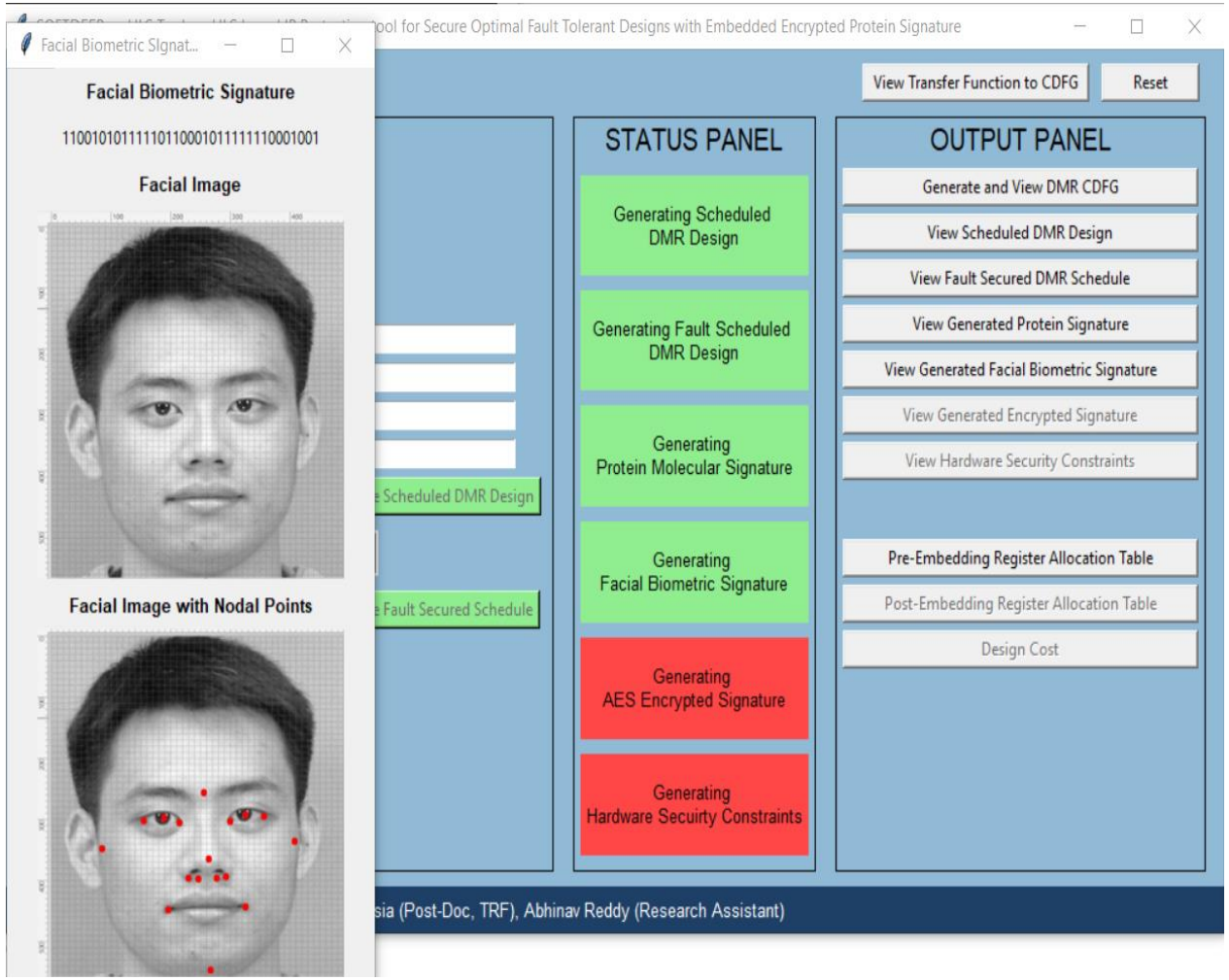
STEP-4: Next, user is asked to load protein sequence of the IP vendor to generate protein molecular signature. The output screen is shown below. Further, the sample protein molecular sequence (polypeptide chain) comprising of 20 different amino acids is also presented [6], [7].



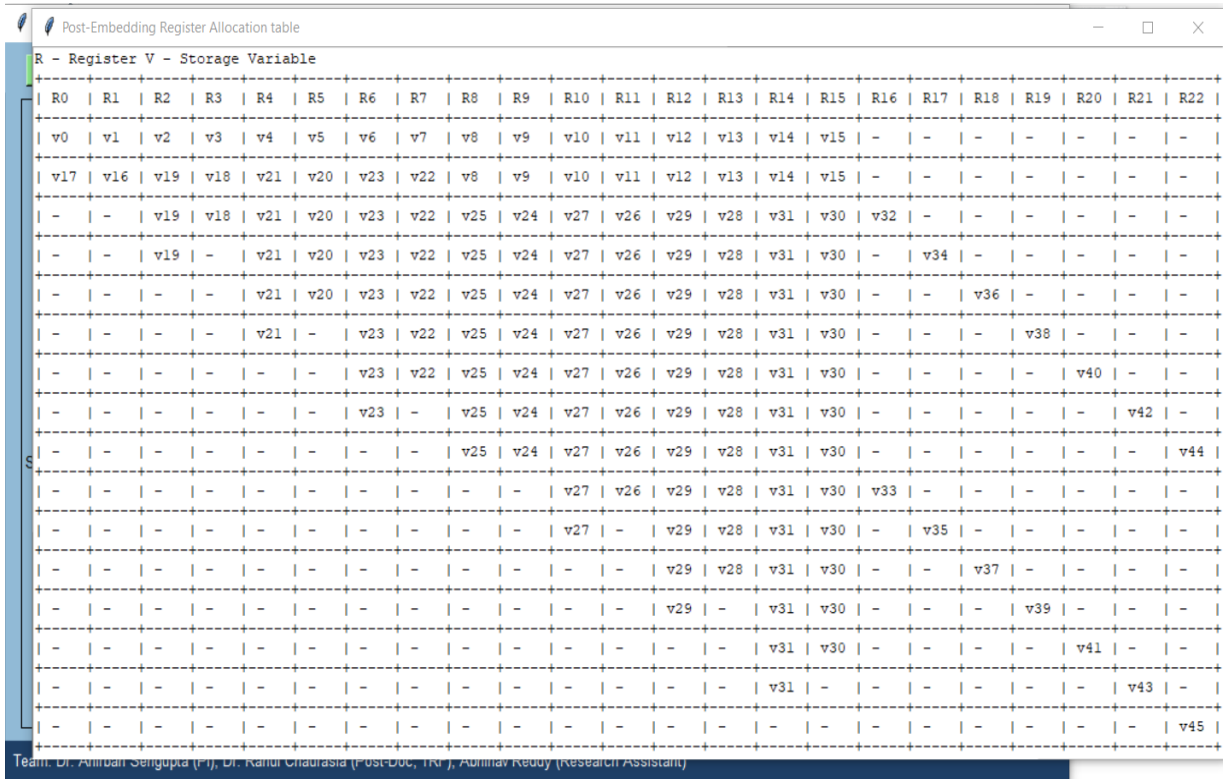
STEP-5: Next, the user is asked to load captured facial image of IP vendor to generate facial biometric signature (being used as encryption key to encrypt generated binarized protein signature in AES framework) [8]. Post loading the facial image, user is asked to select grid size (e.g., 500X500), in order to generate facial features precisely. The output screen depicting facial features, feature dimensions and binarized feature signature is shown below. The feature chosen by IP vendor to generate facial encryption key are highlighted in *BLUE*.



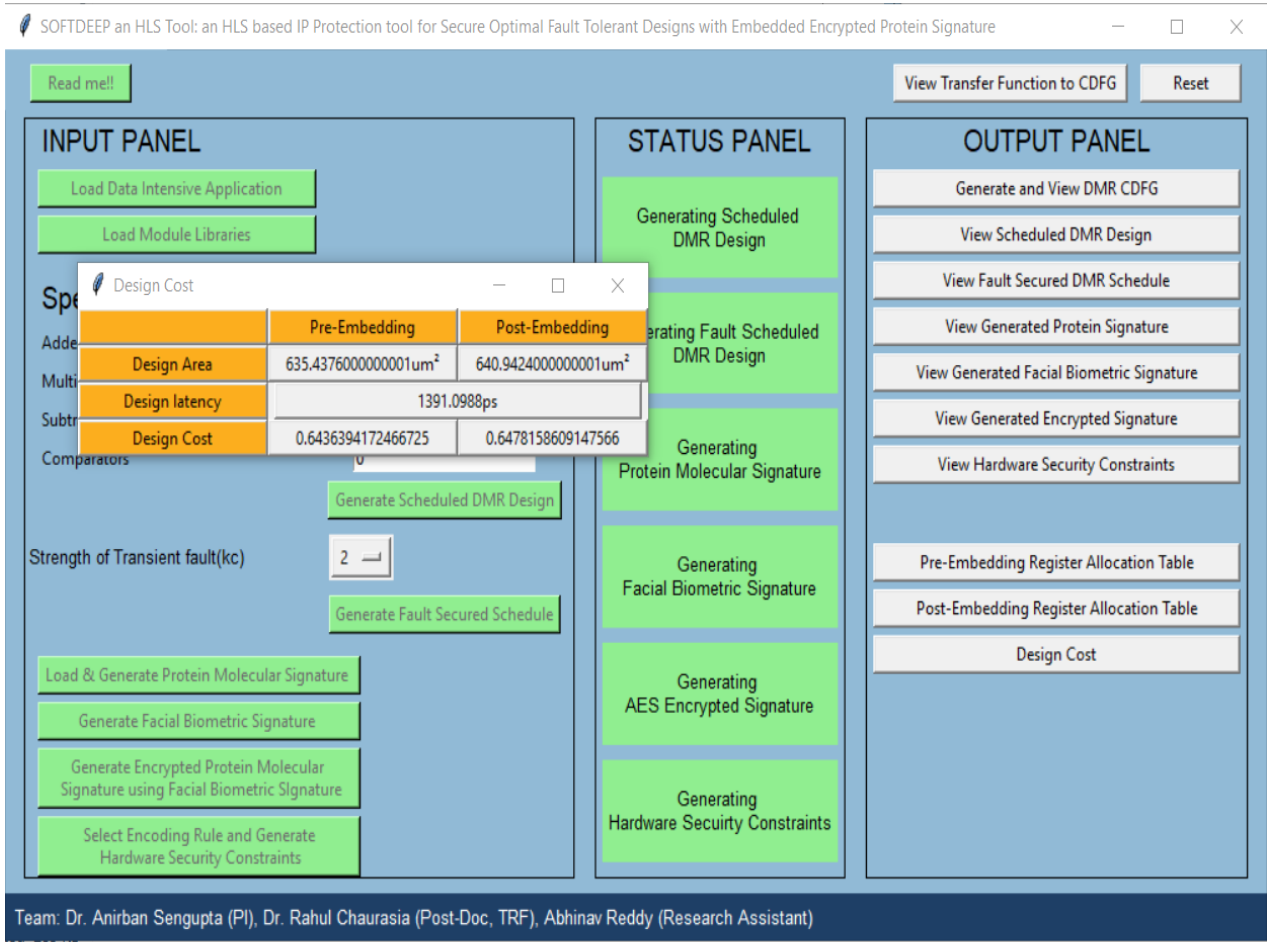
STEP-6: User can also view the facial image with facial features corresponding to captured facial image and generated facial biometric signature. The output screen is shown below:



STEP-7: Next, post using facial biometric signature as encryption key in AES framework, user can generate the encrypted protein signature. Next, the encrypted signature is being encoded into covert hardware security constraints based on the IP vendor specified encoding algorithm and final protein signature length. These generated encoded constraints are then embedded into design during register allocation phase of HLS by performing local alteration of design storage variables without affecting the design functionality. The post-embedding register allocation information is shown below.



- Further, user can view the design cost (comprising the normalized area and design latency computed using 15 nm open cell library [9]) for generating the secure optimal K-cycle fault tolerant scheduled data path processor design for sample input application with embedded encrypted protein molecular biometric of the IP vendor.



References:

1. J. J. Rajendran, O. Sinanoglu and R. Karri, "Building Trustworthy Systems Using Untrusted Components: A High-Level Synthesis Approach," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*, vol. 24, no. 9, pp. 2946-2959, Sept. 2016.
2. "Single event upsets", Intel [online]. Available: https://www.intel.com/content/www/us/en/support/programmable/support-resources/quality/seu.html, Jan. 2022.
3. C. Pilato, S. Garg, K. Wu, R. Karri and F. Regazzoni, "Securing hardware accelerators: a new challenge for high-level synthesis," *IEEE Embedded Syst. Lett.*, vol. 10, no. 3, pp. 77-80, 2018.
4. Brice Colombier, Lilian Bossuet, and David Hly. 2016. From secured logic to IP protection. *Microprocess. Microsyst.* 47, PA (November 2016), 44–54.
5. U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor and Y. Makris, "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain," in *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207-1228, Aug. 2014.
6. Duong, V.-A.; Park, J.-M.; Lim, H.-J.; Lee, H. "Proteomics in Forensic Analysis: Applications for Human Samples". *Appl. Sci.* 2021, 11, 3393.
7. A. Sengupta, R. Chaurasia and A. Anshul, "Robust Security of Hardware Accelerators Using Protein Molecular Biometric Signature and Facial Biometric Encryption Key," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 31, no. 6, pp. 826-839, June 2023.
8. A. Sengupta and M. Rathor, "Facial biometric for securing hardware accelerators," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 1, pp. 112-123, Jan. 2021.
9. 15 nm open cell library. [Online], Available: https://si2.org/open-cell-library/, last accessed on Jan. 2021.