# Securing Hardware Accelerators for CE Systems Using Biometric Fingerprinting

Anirban Sengupta, *Senior Member, IEEE*, and Mahendra Rathor, *Member, IEEE*

***Abstract*— This article presents a novel methodology to secure hardware accelerators (such as digital signal processing (DSP) and multimedia intellectual property (IP) cores) against ownership threats/IP piracy using biometric fingerprinting. In this approach, an IP vendor's biometric fingerprint is first converted into a corresponding digital template, followed by embedding fingerprint's digital template into the design in the form of secret biometric constraints; thereby generating a secured hardware accelerator design. The results report the following qualitative and quantitative analysis of the proposed biometric fingerprint approach: 1) impact of 11 different fingerprints on probability of coincidence (Pc) metric. As evident, the proposed approach achieves a very low Pc value in the range of 2.22E−3 to 4.35E−6. Further, the biometric fingerprint achieves total constraints size between minimum 350 bits to maximum 895 bits; 2) impact of six different resource constraints on the design cost overhead of JPEG compression hardware postembedding biometric fingerprint. As evident, for all the resource constraints implemented, the design cost overhead is 0%; and 3) comparative analysis of proposed biometric fingerprint with recent work, for five different signature strength values, in terms of Pc. As evident, the proposed approach achieves minimum 3.9E+2 times and maximum 6.9E+4 times lower Pc, when compared to recent work.**

***Index Terms*— Biometric fingerprint, consumer electronics (CEs), hardware accelerator, intellectual property (IP) protection, security. 1**

## I. Introduction

THE portable and efficient electronics systems have emerged in the consumer market due to very large-scale integration (VLSI) design process as it integrates thousands of transistors into a single chip or integrated circuit (IC). In the VLSI system, the design process of an IC involves various phases viz. electronic system-level (ESL) synthesis, logic synthesis, layout synthesis, IC fabrication, and testing. The electronic design automation (EDA) industry assists the VLSI design process by providing various EDA tools at different levels of abstraction to facilitate the design automation. Electronic products are designed as system-on-chips (SoCs) which compose of various multivendor third-party intellectual property (3PIP) cores [1]. Since the design of an IP/SoC involves lots of investment and Research and Development, protection of IP core is extremely important against threats such as false claim of ownership and piracy [2], [3]. The piracy/counterfeiting not only harm the revenue and reputation of the authentic IC/IP suppliers in the electronics industry but may also have hardware reliability and consumer safety issues [4], [5]. This is because a counterfeited IC/IP may have hidden malicious logic (Trojan) as well as it does not undergo stringent testing and reliability checks to ensure the quality and safety [6]. Moreover, false claim of IP ownership threat also causes revenue loss to a genuine owner. Therefore, security of IC/IPs is crucial during their design process. The security algorithm should be integrated within the design synthesis process to minimize design overhead and maximize security.

Hardware accelerators such as digital signal processing (DSP) cores and multimedia cores are widely used as application engine in consumer electronics (CEs) systems to perform data or computational intensive tasks to achieve high performance [7]–[9]. DSP cores such as digital filters [10], [11], discrete Fourier transform, fast Fourier transform (FFT), wavelet transformation, convolution filters, etc. find wide utility not only in the image- and signal-processing applications in electronics products such as television, camera, laptop, smart phones, etc. but also in power electronics systems [12] and medical imaging systems/modalities. From the VLSI industry perspective, the security of IP cores such as DSP and multimedia hardware accelerators is very crucial to save an industry from revenue/reputation loss resulting from IP piracy and fraud claim of ownership [13], [14]. Further, it also helps in detective control of pirated designs.

This article targets to secure hardware accelerators used in CE systems. A novel methodology of securing DSP hardware accelerators using IP vendor's biometric fingerprint is presented in this article. It enhances existing design process of designing hardware accelerators by proposing secure design methodology using biometric fingerprinting and generating a respective secured IP core architecture.

## II. Related Work

State-of-the-art approaches for securing hardware accelerators (reusable IP cores) against false claim of ownership and piracy can be broadly classified into two categories viz. hardware watermarking and hardware steganography. Watermarking approaches [14]–[18], [24], [25] that secure IP cores during ESL synthesis are available in the literature. These approaches secure the IP cores by secretly embedding vendor's signature in the design. The signature combination and size is chosen by the vendor. However, the vendor's signature is

Manuscript received February 12, 2020; revised May 4, 2020; accepted May 31, 2020. This work was supported by the Visvesvaraya Ph.D. Scheme of Ministry of Electronics and Information Technology, Government of India, being implemented by Digital India Corporation. *(Corresponding author: Anirban Sengupta.)*

The authors are with the Discipline of Computer Science and Engineering, IIT Indore, Indore 453552, India (e-mail: asengupt@iiti.ac.in; mrathor@iiti.ac.in).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2020.2999514

vulnerable as it can be compromised by an attacker [24]–[26]. This renders it difficult to prove IP ownership by a genuine owner. The signature in approaches [14]–[18], [25] is chosen by the vendor/designer based on signature variables and their encoding rules. Hence, the security of the signature depends on length of the signature, number of signature variables, and their encodings. Once these parameters contributing to the security (size, variables, and encoding) of signature variables are leaked/known to an attacker, the goal of watermarking is defeated. Furthermore, since the signature has a fixed length and it is a combination of only few digits, hence there is a possibility of tampering. Additionally, it is hard to make prior estimation as to which signature combination would result into higher security at lower design overhead. On the contrary, the proposed approach secures the IP cores by embedding biometric fingerprint-based hardware security constraints. The biometric fingerprint constraints are not vulnerable to theft or leakage by an adversary because of its natural uniqueness. Hence, for an attacker to generate the exact digital template corresponding to the original IP vendor's biometric fingerprint is impossible. This is because the digital template generated corresponding to a biometric fingerprint includes information of total number of ones and zeroes as well as the position of each bit. Although it is possible that two digital templates may include same number of ones and zeroes, however, their positions in the digital template will always be different. Thus, digital templates will always be unique. Furthermore, the correlation between a digital template to its equivalent hardware security constraint is always unique because the mapping depends on three factors: 1) vendor/designer's specified ordered list of storage variables (indicating position of storage variables in the list) of the colored interval graph (CIG) design. It is noted that since there are several possibilities of creating ordered list of storage variables (such as sorted increasing order, sorted decreasing order, sorted as per control steps (CS) in scheduling, alternate arrangements of storage variables, arrangements based on functional units (FUs), etc.), hence, designer's specified ordered list of storage variables also plays a key role in developing the correlation; 2) quantity of ones and zeroes; and 3) number of storages variables in the CIG design, which depends upon the resource constraints. On a different note, the number of security constraints embedded can be varied by an IP vendor by varying the minutiae points (using fingerprints from different fingers).

Cryptographic digital signatures-based approaches (such as proposed in [24]) are also effective. Although, cryptographic digital signatures can be considered as unique and secure as biometric fingerprints, however, following differences highlight the strength of proposed biometric fingerprint-based security when compared with cryptographic digital signatures.

1) Strength of a cryptographic digital signature [24] has too much reliance on internal auxiliary steps of signature constraint generation process—such as involving dependence on scheduling algorithm chosen, encoding algorithm devised to generate bitstream of scheduling design, secure hash algorithm (SHA) version chosen, division size of the output chunk generated, and strength of the chosen private key of Rivest-Shamir-Adleman (RSA) (which can also be compromised). Variation in any of the aforesaid internal steps will yield a uniquely different cryptographic digital signature. On the contrary, biometric fingerprint is seamless as it has very limited dependence on auxiliary factors for constraint generation process.
2) Cryptographic digital signature generation process [24] is cumbersome as it involves several steps in between which increase complexity of the process. On the other hand, biometric fingerprint is straightforward but highly secure. It does not use complex steps in between to enhance security, but rather depends on natural biometric feature to provide uniqueness.
3) Cryptographic digital signature generation process [24] is explicitly dependent on hashing algorithm which has several intermediate steps to produce the 'hash/digest.' The hashing algorithm requires knowledge and storage of several hash buffers, additive constants as well as complex computation of word computation functions, round computation functions that includes condition function, rotation function, summation function, and majority function which all add to the complexity of the process. On the contrary, biometric fingerprint produces uniquely secure constraints with minimal blocks of complexity.
4) The signature detection process for cryptographic digital signature process [24] is also arduous as it depends on several factors such as knowledge of points of concatenation of bitstreams, public key, and strength of the signature, while the detection process for biometric fingerprint is seamless as only comparison of digital templates is required to identify original owner or detect piracy.

Steganography approaches [6], [19], [20] also provide security to IP cores, against piracy, and false claim of ownership, during ESL synthesis. These approaches generate security constraints based on either entropy threshold-based key or secret stego-keys. However, the keys can be compromised by an adversary, defeating the purpose of steganography for IP ownership proof and piracy detection. In addition, the algorithmic complexity of generating secret stego-constraints is higher in these approaches. On the contrary, proposed approach does not require any secret key as well as the digital template generation corresponding to IP vendor's biometric fingerprint is simple. Hence, the proposed biometric fingerprint-based hardware security completely safeguards an authentic IP owner against aforesaid threats.

In a standard counterfeited IP detection process, the presence of authentic secret mark in the design is verified. But it is possible in some approaches that an adversary (fake IP vendor) tries to evade this detection process by implanting authentic IP vendor's secret mark. This enables a fake IP vendor to sell his/her counterfeited IP into the market with same trust. Compared to watermarking and steganography, implanting biometric fingerprint-based secret mark (of an authentic IP vendor) by a fake IP vendor into his/her counterfeited IP is impossible. This is because original biometric fingerprint-based secret mark cannot be

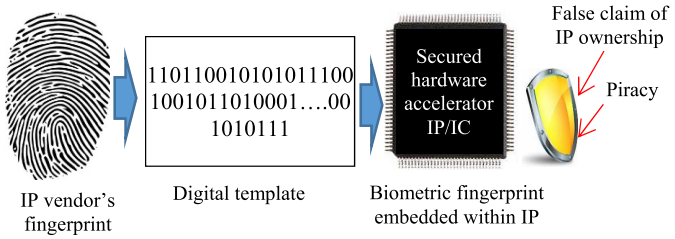


Fig. 1. Thematic representation of the proposed approach of securing hardware accelerator IPs using biometric fingerprinting.

duplicated or reproduced. Therefore, proposed biometric fingerprint-based hardware security provides stronger detective control.

## III. Proposed Work

*Threat Model:* Proposed approach targets to secure hardware accelerators used in CE systems against IP piracy and fraud IP ownership threats.

*Overview:* This article proposes a novel methodology to secure hardware accelerators such as DSP and multimedia cores by embedding biometric fingerprint of the genuine IP vendor into the design. A thematic representation of the proposed approach is shown in Fig. 1. As shown in Fig. 1, an IP vendor's biometric fingerprint is first converted into its corresponding digital template using our algorithm. Thereafter, the fingerprint's digital template is embedded into the hardware accelerator in the form of secret biometric constraints, thus obtaining a secured hardware accelerator. A vendor's biometric (fingerprint) information embedded into the design ensures a robust way to nullify the fraudulent claim of IP ownership by an adversary. In addition, it also enables the detection of IP piracy/counterfeiting using highly authentic secret mark based on genuine biometric fingerprint.

The flow of the proposed methodology is depicted in Fig. 2. As shown in Fig. 2, inputs to the proposed methodology are as follows: 1) algorithmic- or high-level description (such as C/C++/transfer function, etc.) of hardware accelerator application; 2) module library; and 3) resource constraints. Further, an intermediate representation such as control data flow graph (CDFG) of the hardware accelerator application is generated. Further, employing biometric fingerprint security process is performed during ESL synthesis. The summary of the steps are as follows.

1) First, biometric fingerprint of the IP vendor is captured using an optical scanner device. The biometric fingerprint is a distinct pattern of ridges and valleys on the fingertip surface of an individual. In the fingerprint image, the points where the ridge lines end abruptly or fork (bifurcates into branch ridges) are called as minutiae. Ridge endings and ridge bifurcations are major minutiae features used in the verification process of fingerprints. In the proposed methodology, minutiae points (ridge endings and bifurcations) are leveraged as unique biometric fingerprint features of an IP vendor for embedding as hardware security constraints into a hardware accelerator.
2) In the next step, the input fingerprint image is preprocessed to enhance the quality and remove unnecessary noises. In general, preprocessing is performed using FFT enhancement followed by binarization and thinning.
3) Furthermore, minutiae points (ridge endings and bifurcations) are extracted out of the enhanced thinned image.
4) Next, minutiae points are converted into corresponding digital template (discussed in Section III-B).
5) Subsequently, the digital template corresponding to the vendor's biometric fingerprint is mapped into secret hardware constraints based on proposed mapping rules (discussed in Section III-C).
6) The biometric secret constraints are implanted into the hardware accelerator design during the register allocation process of ESL synthesis.
7) Postperforming ESL synthesis, register transfer level (RTL) datapath of the secured hardware accelerator is produced.

The proposed methodology of generating digital template of a vendor's biometric fingerprint and embedding it into an IP core is explained using a demonstration on JPEG coder–decoder (CODEC) hardware accelerator. The electronics industry finds wide applications of JPEG CODEC [23] in CEs systems. The proposed approach enhances the VLSI design process of JPEG CODEC hardware by generating a highly secured design. The details of the proposed methodology with demonstration are presented in Sections III-A–III-D.

### A. Minutiae Points Extraction

*1) Methodology:* In the captured vendor's fingerprint, locating correct minutiae points, which are the unique features of an individual's fingerprint, is very crucial. Therefore, before locating/extracting minutiae points, the captured fingerprint image is enhanced using following processes.

1) *FFT Enhancement:* The use of FFT on sets of pixels of the fingerprint image allows the reconnection of broken ridges, finely separates the parallel ridges, and also makes the ridges thick.
2) *Binarization:* The image with only two intensity values is called as binary image. This image usually shows only black or white, where black is represented by 0 and white is represented by 255. The elementary principle of binarization is to compare the pixel intensities with the threshold, and setting the pixels whose intensities are less than threshold, to 0 and the other to 255.
3) *Thinning:* In this process, the thickness of ridge lines is reduced to one pixel width by deleting pixels at the edge of ridge lines.

*Minutiae Extraction Process:* For extracting the minutiae, the algorithm of crossing number (CN) is employed. For a pixel $P$, the CN is defined as follows [21]:

| | | |
|---|---|---|
| $P_4$ | $P_3$ | $P_2$ |
| $P_5$ | $P$ | $P_1$ |
| $P_6$ | $P_7$ | $P_8$ |

$$\text{CN} = 0.5 * \sum_{i=1}^{8} |P_i - P_{i+1}| \qquad (1)$$

where $P_i$ is the pixel value in the neighborhood of $P$ (shown using above $3 \times 3$ block) with $P_i = 0$ or 1, and $P_1 = P_9$.

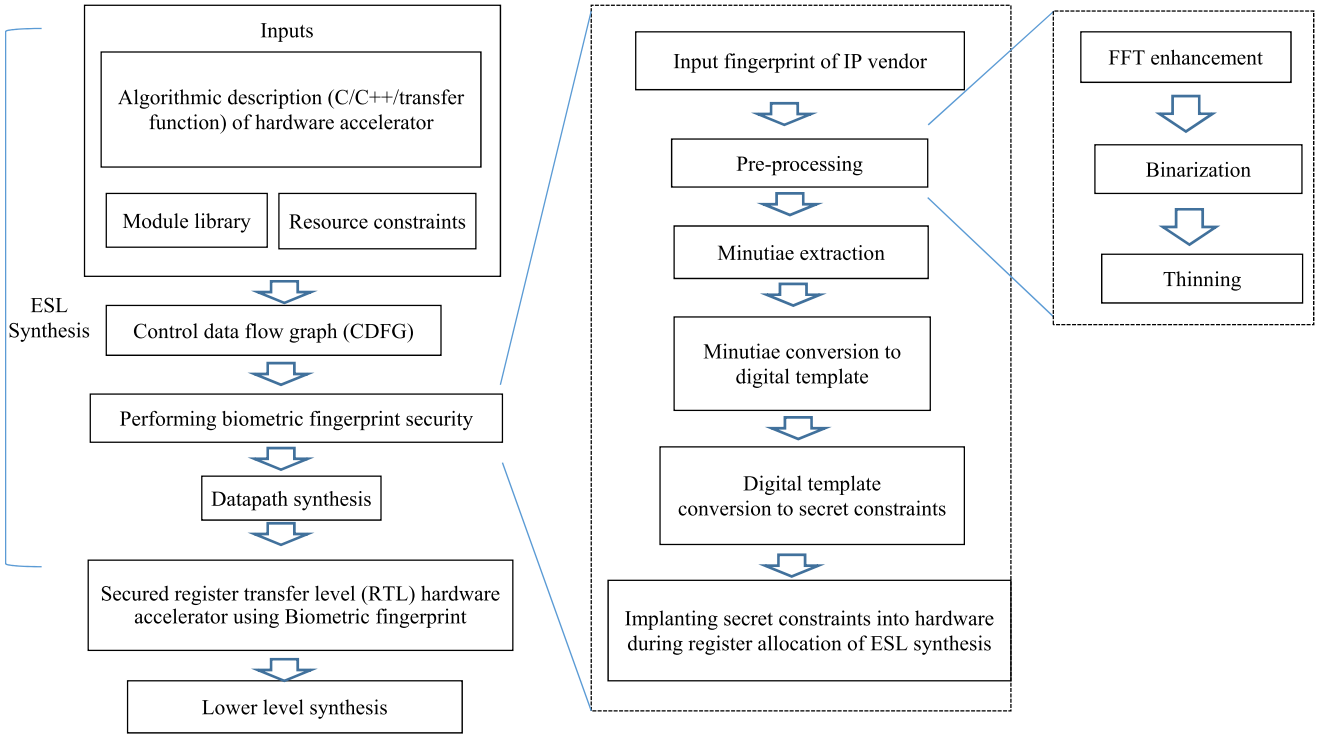


Fig. 2. Proposed methodology to secure hardware accelerators using vendor's biometric fingerprint.

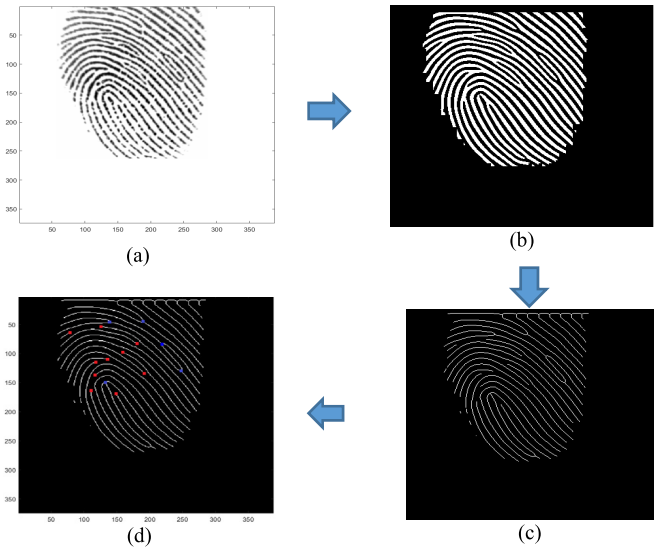


Fig. 3. (a) Original image: 101_2. (b) Binary image. (c) Thinned image. (d) Minutiae points.

The thinned image of fingerprint is scanned and all the minutiae are obtained using the values of CN. If the value of CN is "1," then it indicates ridge ending and if it is "3," then it indicates ridge bifurcation.

*2) Demonstration:* For demonstration, preprocessing followed by minutiae extraction process is performed on a fingerprint image (101_2.tif) selected from a publicly available data set [22]. The original image is shown in Fig. 3(a). Further, the FFT enhancement is performed which is followed by binarization. Binarization makes only two intensity values (0 or 255) present in the fingerprint image. The binarized image is shown in Fig. 3(b). Further, the corresponding thinned image obtained using thinning process is shown in Fig. 3(c). Subsequently, minutiae points are extracted by applying the algorithm of CN number. The fingerprint image with 15 minutiae points is shown in Fig. 3(d). As shown in Fig. 3(d), the ridge endings and bifurcations have been highlighted using red and blue color, respectively. It is worth noting that the boundary minutiae points have been discarded as they may result into false minutiae.

### *B. Digital Template Generation*

*1) Methodology:* Once all the minutiae points are extracted, the next step is to generate corresponding digital template. To do so, each minutia is first represented using following four attributes: $x$-coordinate ($x$), $y$-coordinate ($y$), minutiae type ($m$), and ridge direction/angle ($d$). The ridge angle ($d$) determination is shown in Fig. 4. As shown in Fig. 4, the ridge angle ($d = 34°$) of first minutiae point has been measured clockwise from the horizontal axis. Furthermore, the decimal value of each attribute is converted into the binary equivalent. A minutiae position ($Z$) in binary (digital template of each minutiae point representing hardware constraints) is obtained by concatenating binary equivalent of all attributes as follows:

$$Z = (x \parallel y \parallel m \parallel d). \tag{2}$$

The final digital template is generated by further concatenating individual digital template of all minutiae points.

*2) Demonstration:* The minutiae points shown in Fig. 3(d) have been reported with their respective attributes in Table I. For each minutia point, the corresponding digital template is obtained using (2) as shown in Table I. For example, the digital template of a minutiae point (ridge bifurcation, i.e., $m = 3$) located at $x = 190$, $y = 45$ and angle $d = 34°$

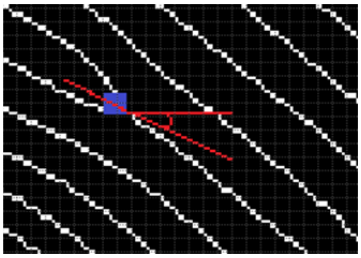

Fig. 4. Determination of ridge direction/angle of a minutiae point.

TABLE I

DIGITAL TEMPLATE OF INDIVIDUAL MINUTIAE POINTS REPRESENTING HARDWARE SECURITY CONSTRAINTS

| x, y | m | Minutiae type name | d | Minutiae position in binary (digital template representing hardware constraints) Z = (x‖ y ‖ m ‖ d) |
|---|---|---|---|---|
| 190, 45 | 3 | bifurcation | 34 | 10111110101101111100010 |
| 139, 46 | 3 | bifurcation | 9 | 10001011101110111001 |
| 126, 54 | 1 | ending | 189 | 11111101101101110111101 |
| 79, 64 | 1 | ending | 327 | 10011111000000110100 0111 |
| 181, 83 | 1 | ending | 38 | 10110101101001111100110 |
| 219, 84 | 3 | bifurcation | 225 | 11011011010100111110000 1 |
| 159, 98 | 1 | ending | 214 | 10011111110001011101011 0 |
| 136, 110 | 1 | ending | 15 | 10001000110111011111 |
| 118, 115 | 1 | ending | 334 | 11101101110011101001110 |
| 248, 130 | 3 | bifurcation | 50 | 11111000100000101111 0010 |
| 192, 134 | 1 | ending | 46 | 11000000100001101101 110 |
| 117, 137 | 1 | ending | 135 | 11101011000100111000011 1 |
| 132, 150 | 3 | bifurcation | 138 | 10000100100101101110001010 |
| 111, 164 | 1 | ending | 267 | 1101111101 0 01001100001011 |
| 149, 169 | 1 | ending | 239 | 10010101101010011111011 11 |

*Note: 'm' indicates CN value of minutiae type, 'd' indicates angle in degree and '‖' indicates concatenation operator*

is: "10111110101101111100010." The final digital template obtained by concatenating individual digital template of all 15 minutiae points is as follows: "101111101011011110001010001011101110111001111111011010110111011111011001111100000011010001111011010110100111100110111011010100111111000011001111111100010111010110110100010001101111011111111011011100111101001110111111000100000101111001010110000001000011011011101110101100010011100001111000010010010110111000101011011111010010011000001011100101011010100111111011111."

The total size of digital template of the vendor's fingerprint is 350 bits which contains 148 zeros and 202 ones.

### C. Embedding Process—Converting Fingerprint Template Into Security Constraints and Implanting Into Design

*1) Methodology:* As shown in (2), each minutia position is a combination of four unique attributes ($x$, $y$, $m$, and $d$), where the hardware security constraints generated correspond to all the minutia positions of a fingerprint. Thus, the digital template generated corresponding to a biometric fingerprint encapsulates all the minutiae points of a fingerprint as well as all the four attributes of a minutia position. The digital template in the form of 1s and 0s representing secret hardware constraints are nothing but digitized information of the fingerprint chosen for embedding. The digital template of biometric fingerprint is converted or mapped into hardware security constraints. The proposed methodology targets to embed the security constraints into the hardware during register allocation phase of ESL synthesis. Embedding constraints during register allocation phase is realized using a CIG framework. A CIG [6] graphically shows the assignment of all storage variables (primary and intermediate inputs output) of the design to minimum possible registers. In the CIG, each node indicates a storage variable, each color indicates a distinct register, and an edge between two nodes indicates overlapping of life time between two storage variables (resulting in the absence of an edge between two nodes of same color).

TABLE II

DIGITAL TEMPLATE TO HARDWARE SECURITY CONSTRAINTS MAPPING RULES

| Bit | Mapping rules |
|---|---|
| 0 | Embed an edge between node pair (even, even) into the CIG (during register allocation of ESL synthesis) |
| 1 | Embed an edge between node pair (odd, odd) into the CIG (during register allocation of ESL synthesis) |

During conversion of fingerprint digital template to hardware security constraints, three factors are important: 1) the number of 1s and 0s of the digital template; 2) number of storages variables that depends on the resource constraints; and 3) vendor/designer's specified ordered list of storage variables (indicating position of storage variables in the list) present in the CIG design. Since there are several possibilities of creating ordered list of storage variables (such as sorted increasing order, sorted decreasing order, sorted as per CS in scheduling, alternate arrangements of storage variables, arrangements based on FUs, etc.), hence designer's specified ordered list of storage variables also plays a key role in developing the correlation. Depending on this ordered list, the same number of 1s and 0s can yield different hardware security constraints. The mapping process (correlation) between hardware security constraints and the biometric fingerprint thus not only depends on the number of 1s and 0s of the digital template, but also on the designer selected ordering of the storage variables in the CIG design as well as the number of storage variables in the CIG design generated based on resource constraint. Therefore, the correlation between generated hardware security constraints and biometric fingerprint is robust, as it enfolds information of all minutia point information, designer-specified storage variable ordered list, and number of storage variables present.

The bits in the digital template are mapped to additional edges into the CIG, representing hardware security constraints of the corresponding biometric fingerprint. The rules of mapping digital template into hardware security constraints (additional edges in the CIG) are presented in Table II. The secret artificial edges representing hardware security constraints, of a vendor's biometric fingerprint, are embedded one by one into the CIG. If the edge constraint is to be embedded between two nodes of same color, then the color/register of one of the node/storage variable is essentially swapped with another node in the CIG to execute the respective nodes with different colors/registers. Therefore, embedding secret artificial edges in the form of security constraints may alter the register allocation of some storage variables. Thus, a hardware accelerator IP core is secured by implanting security constraints

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $C_4$ | $C_4$ | $C_4$ | $C_4$ | $C_4$ | $C_4$ | $C_4$ | $C_4$ |
| $C_1$ | $C_3$ | $C_5$ | $C_7$ | $-C_7$ | $-C_5$ | $-C_3$ | $-C_1$ |
| $C_2$ | $C_6$ | $-C_6$ | $-C_2$ | $-C_2$ | $-C_6$ | $-C_6$ | $C_2$ |
| $C_3$ | $-C_7$ | $-C_1$ | $-C_5$ | $C_5$ | $C_1$ | $C_7$ | $-C_3$ |
| $C_4$ | $-C_4$ | $-C_4$ | $C_4$ | $C_4$ | $-C_4$ | $-C_4$ | $C_4$ |
| $C_5$ | $-C_1$ | $C_7$ | $C_3$ | $-C_3$ | $-C_7$ | $C_1$ | $-C_5$ |
| $C_6$ | $-C_2$ | $C_2$ | $-C_6$ | $-C_6$ | $C_2$ | $-C_2$ | $C_6$ |
| $C_7$ | $-C_5$ | $C_3$ | $-C_1$ | $C_1$ | $-C_3$ | $C_5$ | $-C_7$ |

Fig. 5. 2-D-DCT coefficient matrix "$C$." Note: $c1$, $-c1$, $c2$, $-c2$, $c3$, $-c3$, $c4$, $-c4$, $c5$, $-c5$, $c6$, $-c6$, $c7$, and $-c7$ indicate elements in the eight-point DCT coefficient matrix.

corresponding to biometric fingerprint during register allocation phase of ESL synthesis.

*2) Demonstration:* To apply the proposed biometric fingerprint-based hardware security, the target designs should have following general characteristics: 1) the proposed approach is aptly suitable to such designs which are designed using ESL synthesis framework because of large size and higher complexity, for example, DSP hardware accelerators (such as finite impulse response (FIR) filters, discrete cosine transform (DCT) cores, etc.) and multimedia hardware accelerators such as JPEG CODEC and MPEG processor and 2) smaller designs can also be secured using the proposed approach by embedding that vendor's fingerprint which corresponds to less number of minutiae points (less security constraints).

The proposed methodology of embedding hardware security constraints corresponding to biometric fingerprint of an IP vendor is demonstrated on a JPEG CODEC hardware accelerator.

*a) Background on JPEG CODEC:* The first step in the JPEG compression process is to convert the input image into an $N \times N$ matrix whose values are the pixel intensities at the respective locations. Further, the $N \times N$ matrix is divided into a number of nonoverlapping $8 \times 8$ blocks of pixels. This is because the DCT function underneath the JPEG compressor uses an $8 \times 8$ block at a time. Further, each $8 \times 8$ block of pixels is transformed using 2-D-DCT transformation as follows:

$$D = (C * M) * C' \tag{3}$$

where $D$ denotes the DCT transformed matrix, $C$ denotes the 2-D-DCT coefficient matrix shown in Fig. 5, $M$ denotes the $8 \times 8$ block of pixels of input image and $C'$ denotes the transpose of $C$. The first pixel value "$D11$" of the DCT transformed matrix $D$ is computed as follows:

$$\begin{aligned} D11 = {} & (c4 * d11) + (c4 * d12) + (c4 * d13) + (c4 * d14) \\ & + (c4 * d15) + (c4 * d16) + (c4 * d17) + (c4 * d18) \end{aligned} \tag{4}$$

where in all product terms, the left side values indicate the elements of first column of matrix $C'$ and the right side values indicate the elements of first row of matrix $C * M$. Further, the first element ($d11$) of the matrix $C * M$ is computed as follows:

$$\begin{aligned} d11 = {} & (c4 * m11) + (c4 * m21) + (c4 * m31) + (c4 * m41) \\ & + (c4 * m51) + (c4 * m61) + (c4 * m71) + (c4 * m81) \end{aligned} \tag{5}$$

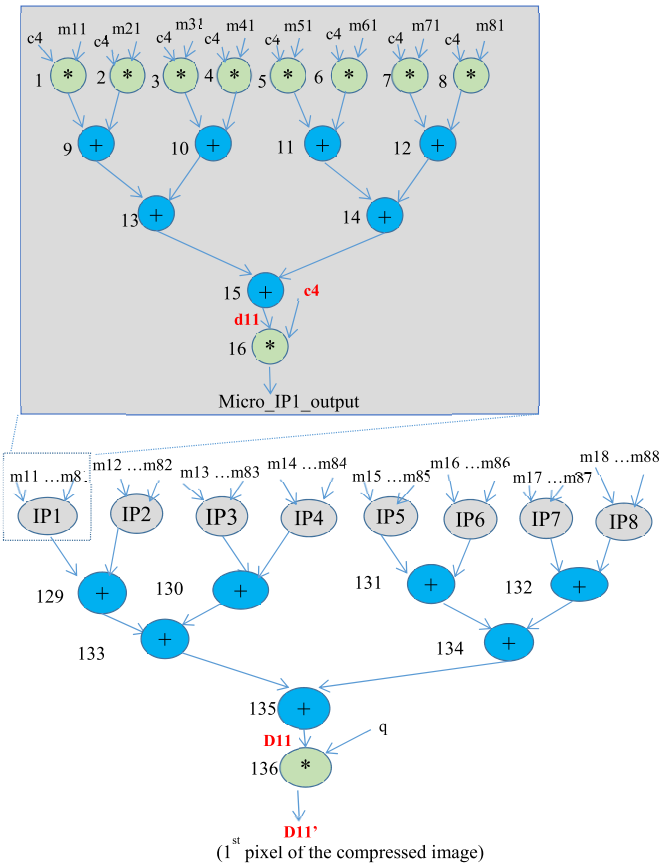


Fig. 6. DFG of JPEG CODEC hardware accelerator. Note: inputs $m11$, $m12$–$m88$ indicate elements of the first $8 \times 8$ block of pixel matrix, $c4$ indicates a coefficient in DCT matrix, $d11$ indicates the first element of the matrix $C * M$, $D11$ indicates the first element of the final output matrix $(C * M) * C'$. $D11$ is the first pixel value in the DCT transformed matrix. Furthermore, $q$ indicates a value in the quantization matrix used to generate a quantized pixel value of compressed image.

where in all product terms, the left side values indicate the elements of first row of matrix $C$ and the right side values indicate the elements of first column of matrix $M$. Furthermore, compression is performed on each DCT transformed $8 \times 8$ block of pixels using a quantization matrix. Postmultiplying the 1st pixel ($D11$) of DCT transformed matrix with the quantization coefficient "$q$," the 1st pixel ($D11'$) of the compressed image is generated. Postquantization, the quantized image is converted to 1-D array using zigzag scanning. Furthermore, run length encoding is applied to obtain bitstream of the compressed image for storage purpose. To reconstruct the original image from the stored compressed image data, run length decoding followed by inverse zigzag scanning, inverse quantizationm and inverse DCT transformation are performed in the JPEG decompression process. Fig. 6 shows the DFG/CDFG representing JPEG compression hardware accelerator which computes the 1st pixel ($D11'$) of the compressed image. The JPEG compression hardware comprises of eight micro-IPs underneath as shown in Fig. 6. Each micro-IP performs 16 operations and total operations in the entire JPEG CODEC IP core are 136. Further, scheduling, hardware allocation and binding of operations (opns) in the JPEG compression DFG are performed using FUs/resource constraints of three adders and three multipliers as shown

TABLE III

SCHEDULING OF JPEG COMPRESSION HARDWARE ACCELERATOR IP USING RESOURCE CONSTRAINTS OF (3A, 3M)

| CS | Opns assign to M1 | Opns assign to M2 | Opns assign to M3 | Opns assign to A1 | Opns assign to A2 | Opns assign to A3 |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | -- | -- | -- |
| 2 | 4 | 5 | 6 | 9 | -- | -- |
| 3 | 7 | 8 | 17 | 10 | 11 | -- |
| 4 | 18 | 19 | 20 | 12 | 13 | -- |
| 5 | 21 | 22 | 23 | 25 | 26 | 14 |
| 6 | 24 | 33 | 34 | 27 | 29 | 15 |
| 7 | 35 | 36 | 37 | 28 | 41 | -- |
| 8 | 38 | 39 | 40 | 42 | 30 | -- |
| 9 | 49 | 50 | 51 | 43 | 44 | 45 |
| 10 | 52 | 53 | 54 | 31 | 57 | 46 |
| 11 | 55 | 56 | 65 | 58 | 59 | 47 |
| 12 | 66 | 67 | 68 | 60 | 61 | -- |
| 13 | 69 | 70 | 71 | 73 | 74 | 62 |
| 14 | 72 | 81 | 82 | 75 | 77 | 63 |
| 15 | 83 | 84 | 85 | 76 | 89 | -- |
| 16 | 86 | 87 | 88 | 90 | 78 | -- |
| 17 | 97 | 98 | 99 | 91 | 92 | 93 |
| 18 | 100 | 101 | 102 | 79 | 105 | 94 |
| 19 | 103 | 104 | 113 | 106 | 107 | 95 |
| 20 | 114 | 115 | 116 | 108 | 109 | -- |
| 21 | 117 | 118 | 119 | 121 | 122 | 110 |
| 22 | 120 | 16 | 32 | 123 | 125 | 111 |
| 23 | 48 | 64 | 80 | 124 | 129 | -- |
| 24 | 96 | 112 | -- | 130 | 126 | -- |
| 25 | -- | -- | -- | 131 | 133 | 127 |
| 26 | 128 | -- | -- | -- | -- | -- |
| 27 | -- | -- | -- | 132 | -- | -- |
| 28 | -- | -- | -- | 134 | -- | -- |
| 29 | -- | -- | -- | 135 | -- | -- |
| 30 | 136 | -- | -- | -- | -- | -- |

*Note: A, M and CS indicate adder, multiplier and control steps respectively*

in Table III. As shown in Table III, all 136 operations are executing in 30 CS. Furthermore, there are total 209 primary and intermediate inputs and outputs which result into 209 storage variables. Assignment of 209 storage variables of the design to 73 registers can be shown using CIG comprising of 209 nodes (V0–V208) and 73 colors (registers). The JPEG compression hardware (computing pixel values of compressed image) requires 73 registers because there are maximum 73 independent primary inputs wherein 64 inputs are pixel values in the $8 \times 8$ block of input image, eight inputs are the elements of 1st row/column of $8 \times 8$ DCT coefficients matrix and one input of quantization coefficient "$q$."

*b) Embedding Biometric Fingerprint Security Constraints Into JPEG Compression Hardware:* To secure JPEG compression hardware, security constraints corresponding to biometric fingerprint of the vendor are embedded during register allocation phase of ESL synthesis while exploiting CIG framework. To do so, first the digital template of vendor's biometric fingerprint (obtained in Section III-B) is mapped to hardware security constraints based on proposed mapping rule shown in Table II. Postmapping, hardware security constraints corresponding to 148 zeros and 202 ones in the digital template are listed in Tables IV and V, respectively. Each security constraint shown in Tables IV and V is added as an additional edge in the CIG. Postembedding all security constraints, the register allocation of storage variables (0–208) is shown in Fig. 7. As discussed earlier in the methodology

TABLE IV

HARDWARE SECURITY CONSTRAINTS (ADDITIONAL EDGES IN THE CIG) CORRESPONDING TO 148 ZEROS IN THE DIGITAL TEMPLATE

| | | | | | |
|---|---|---|---|---|---|
| V0,V2 | V0,V52 | V0,V102 | V0,V152 | V0,V202 | V2,V46 |
| V0,V4 | V0,V54 | V0,V104 | V0,V154 | V0,V204 | V2,V48 |
| V0,V6 | V0,V56 | V0,V106 | V0,V156 | V0,V206 | V2,V50 |
| V0,V8 | V0,V58 | V0,V108 | V0,V158 | V0,V208 | V2,V52 |
| V0,V10 | V0,V60 | V0,V110 | V0,V160 | V2,V4 | V2,V54 |
| V0,V12 | V0,V62 | V0,V112 | V0,V162 | V2,V6 | V2,V56 |
| V0,V14 | V0,V64 | V0,V114 | V0,V164 | V2,V8 | V2,V58 |
| V0,V16 | V0,V66 | V0,V116 | V0,V166 | V2,V10 | V2,V60 |
| V0,V18 | V0,V68 | V0,V118 | V0,V168 | V2,V12 | V2,V62 |
| V0,V20 | V0,V70 | V0,V120 | V0,V170 | V2,V14 | V2,V64 |
| V0,V22 | V0,V72 | V0,V122 | V0,V172 | V2,V16 | V2,V66 |
| V0,V24 | V0,V74 | V0,V124 | V0,V174 | V2,V18 | V2,V68 |
| V0,V26 | V0,V76 | V0,V126 | V0,V176 | V2,V20 | V2,V70 |
| V0,V28 | V0,V78 | V0,V128 | V0,V178 | V2,V22 | V2,V72 |
| V0,V30 | V0,V80 | V0,V130 | V0,V180 | V2,V24 | V2,V74 |
| V0,V32 | V0,V82 | V0,V132 | V0,V182 | V2,V26 | V2,V76 |
| V0,V34 | V0,V84 | V0,V134 | V0,V184 | V2,V28 | V2,V78 |
| V0,V36 | V0,V86 | V0,V136 | V0,V186 | V2,V30 | V2,V80 |
| V0,V38 | V0,V88 | V0,V138 | V0,V188 | V2,V32 | V2,V82 |
| V0,V40 | V0,V90 | V0,V140 | V0,V190 | V2,V34 | V2,V84 |
| V0,V42 | V0,V92 | V0,V142 | V0,V192 | V2,V36 | V2,V86 |
| V0,V44 | V0,V94 | V0,V144 | V0,V194 | V2,V38 | V2,V88 |
| V0,V46 | V0,V96 | V0,V146 | V0,V196 | V2,V40 | V2,V90 |
| V0,V48 | V0,V98 | V0,V148 | V0,V198 | V2,V42 | -- |
| V0,V50 | V0,V100 | V0,V150 | V0,V200 | V2,V44 | -- |

TABLE V

HARDWARE SECURITY CONSTRAINTS (ADDITIONAL EDGES IN THE CIG) CORRESPONDING TO 202 ONES IN THE DIGITAL TEMPLATE

| | | | | | |
|---|---|---|---|---|---|
| V1, V3 | V1, V71 | V1, V139 | V1, V207 | V3, V71 | V3, V139 |
| V1, V5 | V1, V73 | V1, V141 | V3, V5 | V3, V73 | V3, V141 |
| V1, V7 | V1, V75 | V1, V143 | V3, V7 | V3, V75 | V3, V143 |
| V1, V9 | V1, V77 | V1, V145 | V3, V9 | V3, V77 | V3, V145 |
| V1, V11 | V1, V79 | V1, V147 | V3, V11 | V3, V79 | V3, V147 |
| V1, V13 | V1, V81 | V1, V149 | V3, V13 | V3, V81 | V3, V149 |
| V1, V15 | V1, V83 | V1, V151 | V3, V15 | V3, V83 | V3, V151 |
| V1, V17 | V1, V85 | V1, V153 | V3, V17 | V3, V85 | V3, V153 |
| V1, V19 | V1, V87 | V1, V155 | V3, V19 | V3, V87 | V3, V155 |
| V1, V21 | V1, V89 | V1, V157 | V3, V21 | V3, V89 | V3, V157 |
| V1, V23 | V1, V91 | V1, V159 | V3, V23 | V3, V91 | V3, V159 |
| V1, V25 | V1, V93 | V1, V161 | V3, V25 | V3, V93 | V3, V161 |
| V1, V27 | V1, V95 | V1, V163 | V3, V27 | V3, V95 | V3, V163 |
| V1, V29 | V1, V97 | V1, V165 | V3, V29 | V3, V97 | V3, V165 |
| V1, V31 | V1, V99 | V1, V167 | V3, V31 | V3, V99 | V3, V167 |
| V1, V33 | V1, V101 | V1, V169 | V3, V33 | V3, V101 | V3, V169 |
| V1, V35 | V1, V103 | V1, V171 | V3, V35 | V3, V103 | V3, V171 |
| V1, V37 | V1, V105 | V1, V173 | V3, V37 | V3, V105 | V3, V173 |
| V1, V39 | V1, V107 | V1, V175 | V3, V39 | V3, V107 | V3, V175 |
| V1, V41 | V1, V109 | V1, V177 | V3, V41 | V3, V109 | V3, V177 |
| V1, V43 | V1, V111 | V1, V179 | V3, V43 | V3, V111 | V3, V179 |
| V1, V45 | V1, V113 | V1, V181 | V3, V45 | V3, V113 | V3, V181 |
| V1, V47 | V1, V115 | V1, V183 | V3, V47 | V3, V115 | V3, V183 |
| V1, V49 | V1, V117 | V1, V185 | V3, V49 | V3, V117 | V3, V185 |
| V1, V51 | V1, V119 | V1, V187 | V3, V51 | V3, V119 | V3, V187 |
| V1, V53 | V1, V121 | V1, V189 | V3, V53 | V3, V121 | V3, V189 |
| V1, V55 | V1, V123 | V1, V191 | V3, V55 | V3, V123 | V3, V191 |
| V1, V57 | V1, V125 | V1, V193 | V3, V57 | V3, V125 | V3, V193 |
| V1, V59 | V1, V127 | V1, V195 | V3, V59 | V3, V127 | V3, V195 |
| V1, V61 | V1, V129 | V1, V197 | V3, V61 | V3, V129 | V3, V197 |
| V1, V63 | V1, V131 | V1, V199 | V3, V63 | V3, V131 | V3, V199 |
| V1, V65 | V1, V133 | V1, V201 | V3, V65 | V3, V133 | V3, V201 |
| V1, V67 | V1, V135 | V1, V203 | V3, V67 | V3, V135 | -- |
| V1, V69 | V1, V137 | V1, V205 | V3, V69 | V3, V137 | -- |

part, embedding some constraints may result into change in the register allocation of some storage variables. Postembedding, the changes in register allocation of storage variables $V196$,

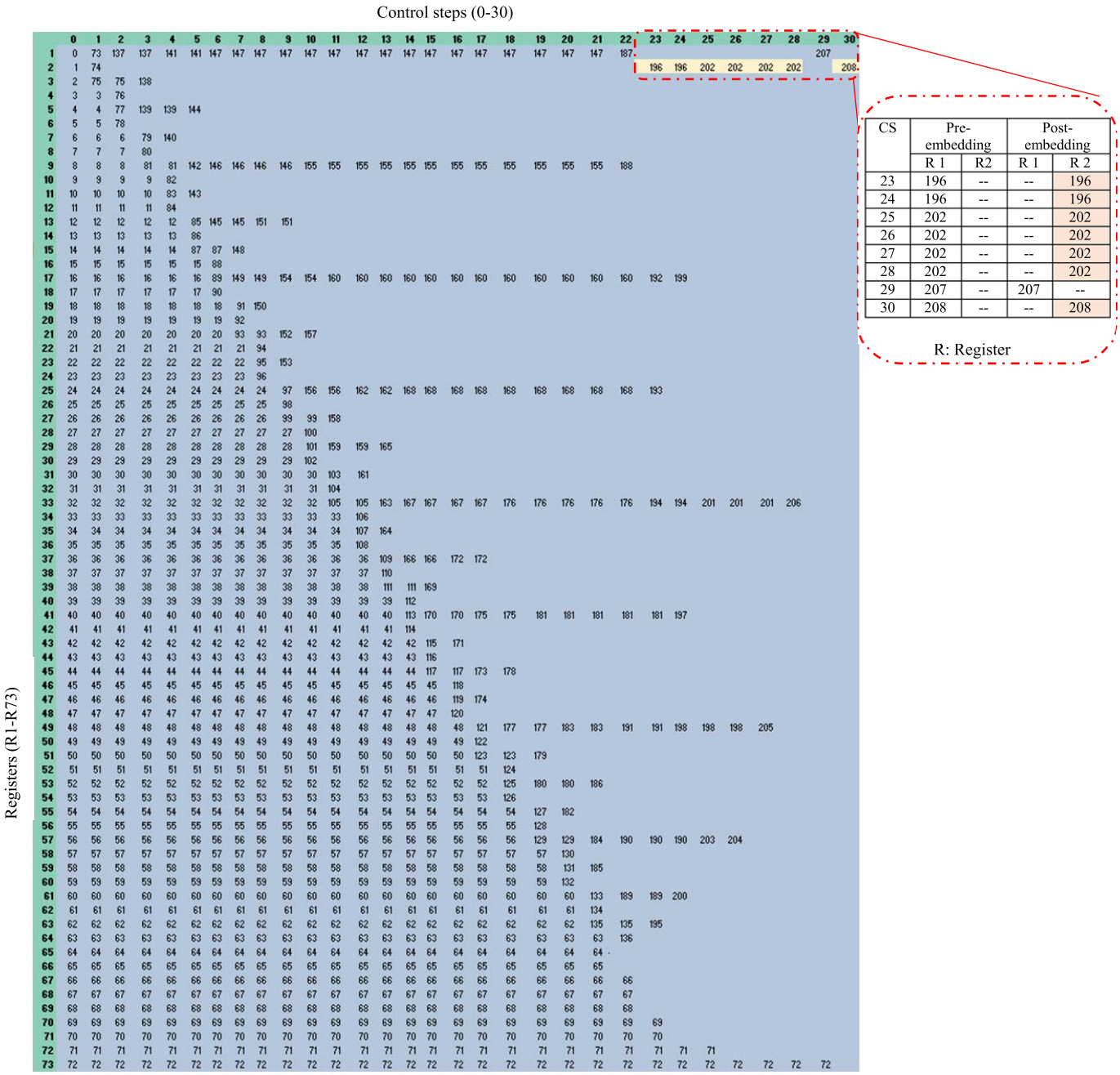


Fig. 7. Register allocation of storage variables (0–208) postembedding during ESL synthesis of JPEG compression hardware accelerator.

$V202$, and $V208$ are highlighted in Fig. 7 and also shown through a zoomed window. The change in register allocation occurs because, initially, both nodes (storage variables) of pairs $\langle V0, V196\rangle$, $\langle V0, V202\rangle$, and $\langle V0, V208\rangle$ in the CIG were assigned to same color/register ($R1$). To add additional edges between these pairs, the respective storage variables/nodes need be assigned to distinct registers/colors. Therefore, storage variables $V196$, $V202$, and $V208$ are assigned to register $R2$ postembedding security constraints as shown in Fig. 7. Furthermore, ESL synthesis is performed which generates RTL datapath of secured JPEG CODEC hardware accelerator. The secured JPEG compression hardware accelerator circuit, postembedding biometric fingerprint constraints, is shown in Fig. 8. The storage variables (primary inputs and intermediate inputs and outputs) are shown as inputs to the circuit. The red boundary indicates region where fingerprint constraints are implanted. The list of hardware components in the pre- and postsecured RTL datapath is shown in Table VI. As shown in Table VI, there is no significant change in the hardware components of datapath postembedding security constraints.

The embedded biometric fingerprint represents the authentic secret mark of an IP vendor. Therefore, embedded fingerprint of genuine IP vendor enables the verification of true IP ownership, thus nullifying the fraudulent IP ownership claim. In addition, biometric fingerprint-based authentic secret mark helps in detecting counterfeited/cloned IPs/ICs as the fingerprint of a genuine owner cannot be present in those ones.

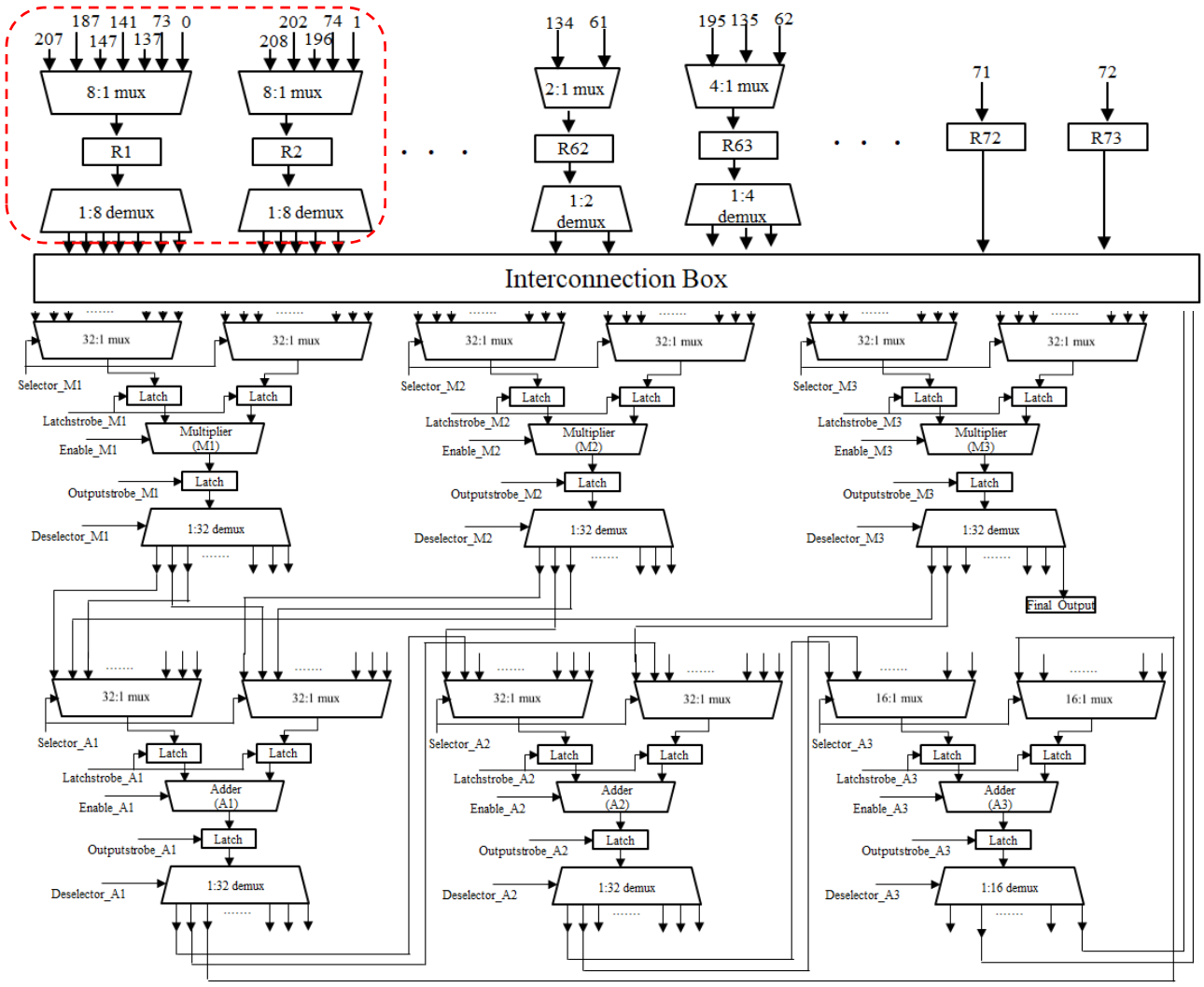


Fig. 8. Secured datapath of JPEG compression hardware accelerator implanted with biometric fingerprint.

TABLE VI

RESOURCES IN THE RTL DATAPATH OF JPEG COMPRESSION HARDWARE (PRE- AND POSTEMBEDDING BIOMETRIC FINGERPRINT CONSTRAINTS)

| Resources pre-embedding security constraints | | | | Resources post-embedding security constraints | | | |
|---|---|---|---|---|---|---|---|
| FUs | # of registers | Multiplexers | Demultiplexers | FUs | # of registers | Multiplexers | Demultiplexers |
| 3M, 3A | 73 | # 32x1 Muxes=10<br># 16x1 Muxes=3<br># 8x1 Muxes =7<br># 4x1 Muxes =24<br># 2x1 Muxes =32 | # 1x32 Demuxes=5<br># 1x16 Demuxes=2<br># 1x8 Demuxes=7<br># 1x4 Demuxes=24<br># 1x2 Demuxes=32 | 3M, 3A | 73 | # 32x1 Muxes=10<br># 16x1 Muxes=2<br># 8x1 Muxes =9<br># 4x1 Muxes =24<br># 2x1 Muxes =31 | # 1x32 Demuxes=5<br># 1x16 Demuxes=1<br># 1x8 Demuxes=9<br># 1x4 Demuxes=24<br># 1x2 Demuxes=31 |

### *D. Detection Process of Biometric Fingerprint*

The detection process of biometric fingerprint embedded into the hardware accelerator design is highlighted in Fig. 9. As shown in Fig. 9, the digital template corresponding to the embedded fingerprint constraints in hardware accelerator design is matched with the digital template of IP vendor's biometric fingerprint. If the number of fingerprint constraints (0s and 1s) in the digital template of vendor's biometric fingerprint does not match with the embedded security constraints then the design is probably counterfeited (i.e., authentic vendor's fingerprint is not embedded into the design). However, if the number of constraints embedded matches with the two distinct templates having same number of 0s and 1s, then the ownership conflict will be resolved with matching the positions of constraints in the both digital templates and the ordered list of storage variables used to embed constraints. Here, only the digital template of true IP vendor's biometric fingerprint will match with that of embedded fingerprint constraints. Hence, the ownership is awarded to the true owner and false ownership claim by an adversary is nullified.

## IV. RESULTS AND ANALYSIS

The results of proposed biometric fingerprinting-based hardware security are analyzed in this section. The impact of embedding biometric fingerprint-based security constraints into JPEG CODEC hardware accelerator for varying

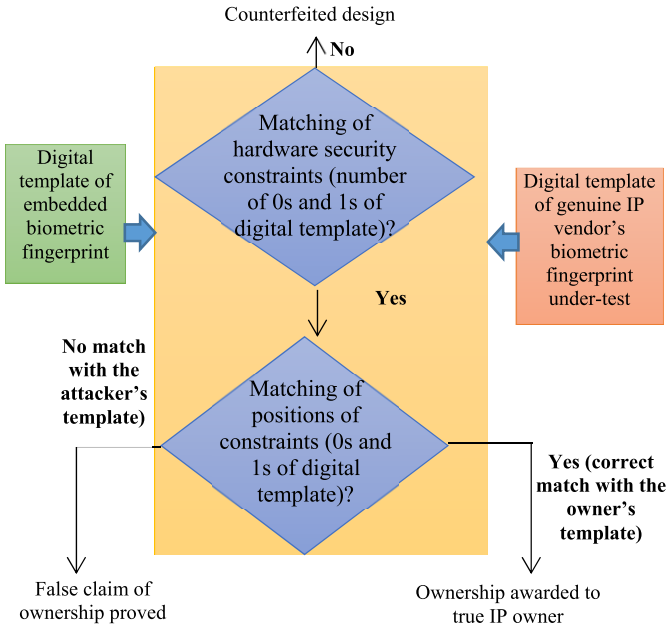


Fig. 9. Proving true IP ownership using the proposed detection approach.

fingerprints is analyzed in terms of security and design cost, as follows.

### *A. Analyzing the Impact of Varying Biometric Fingerprints on Generated Hardware Security Constraints*

The number of hardware security constraints generated depends on the number of minutiae points extracted from the biometric fingerprint. To analyze the impact of varying biometric fingerprints on hardware security constraints, different fingerprint images are selected from the data set [22]. Fig. 10 shows the impact of varying biometric fingerprints on hardware security constraints to be embedded in an IP core. As shown in Fig. 10, total 22 minutiae points are extracted from the original image: 101_1. The corresponding digital template of size 526 bits comprises of 224 zeros and 302 ones which represent the hardware security constraints to be embedded into the design. Similarly, hardware security constraints generated from different biometric fingerprints are shown in Fig. 10. Further, Fig. 11 shows the impact of varying fingerprints of different fingers of same person on the generated hardware security constraints. As shown in Fig. 11, the fingerprint of thumb results into lesser security constraints and fingerprint of ring finger results into higher security constraints. It is noteworthy that to vary embedded constraints, different fingers of an IP vendor can be used. On the contrary in the contemporary security approaches, signature size or combination is varied to increase/decrease the secret constraints to be embedded. However, it is hard to estimate the impact of varying signature combination and size on generated hardware security constraints. This is because, a larger signature size may not result into higher security constraints as some signature digits may not be embedded. In addition, the biometric fingerprint-based hardware security constraints offers following benefits.

1) *Nonvulnerable:* Unlike related approaches [6], [14]–[20], the biometric fingerprint constraints are not vulnerable to leaking or theft by an adversary. More explicitly, the vendor's signature and encoding rules in the watermarking approaches and threshold entropy or secret stego-keys in the steganography approaches may be compromised by an attacker, resulting into false claim of IP ownership and counterfeiting attack. However, biometric fingerprint of the IP vendor cannot be stolen and claimed by an adversary. This renders the vendor's fingerprint mark highly authentic and secure.
2) *Nonreplicable:* Because of uniqueness of each biometric fingerprint, the biometric security constraints are not replicable. However, in related approaches, vendor's signature may be replicated, in case it is compromised. The strength of vendor's signature depends on its secrecy. Once the factors (signature size, variables used and encoding rules) contributing to the security of the signature are known to an attacker, it becomes ineffective, as it can easily be replicated by an attacker [24]–[26]. However, since the biometric fingerprint is obtained from the vendor's finger, therefore it is unique. Hence, it rules out the possibility of replication.
3) *No Storage Required:* The biometric security constraints are not required to be stored as they can always be derived from the respective fingerprint. On the contrary, in related approaches [6], [14]–[20], vendor's signature and encoding rules or private keys are required to be stored/memorized to reproduce them during the detection process of secret constraints. In case of fingerprint biometric-based security, any person who is a highly trustworthy insider in the vendor's house can be selected for fingerprint embedding. However, the biometric fingerprint need not require storage as long as the person selected is associated with the vendor's house and is alive. Nevertheless, considering the uncertainty of a person's life, storage of fingerprint could be done to ensure the protection of previously distributed IP cores.

### *B. Security Analysis*

The strength of IP ownership proof using proposed methodology is analyzed in terms of probability of coincidence (Pc) metric given as follows [20]:

$$\text{Pc} = \left(1 - \frac{1}{R}\right)^{f1} * \left(1 - \frac{1}{\pi_{i=1}^{n} N(Fi)}\right)^{f2} \tag{6}$$

where "$R$" is the number of colors/registers in original CIG, $f1$ denotes the number of additional edges (hardware security constraints) embedded into the CIG (in register allocation phase) and $f2$ denotes the number of constraints embedded in the FU vendor allocation phase. Further, "$n$" denotes the total types of FU, "$N$" indicates the number of FUs of type $Fi$. The Pc represents the probability of coincidently detecting hardware security constraints in an unsecured design. To achieve the low Pc, higher security constraints ($f$) should be embedded which in turn results into higher digital evidence embedded into the design. This leads to higher strength of ownership proof. In case of proposed approach, security constraints [represented by $f1$ in (6)] are only embedded into the CIG during register allocation phase, therefore $f2$ remains zero. Table VII shows the Pc for varying

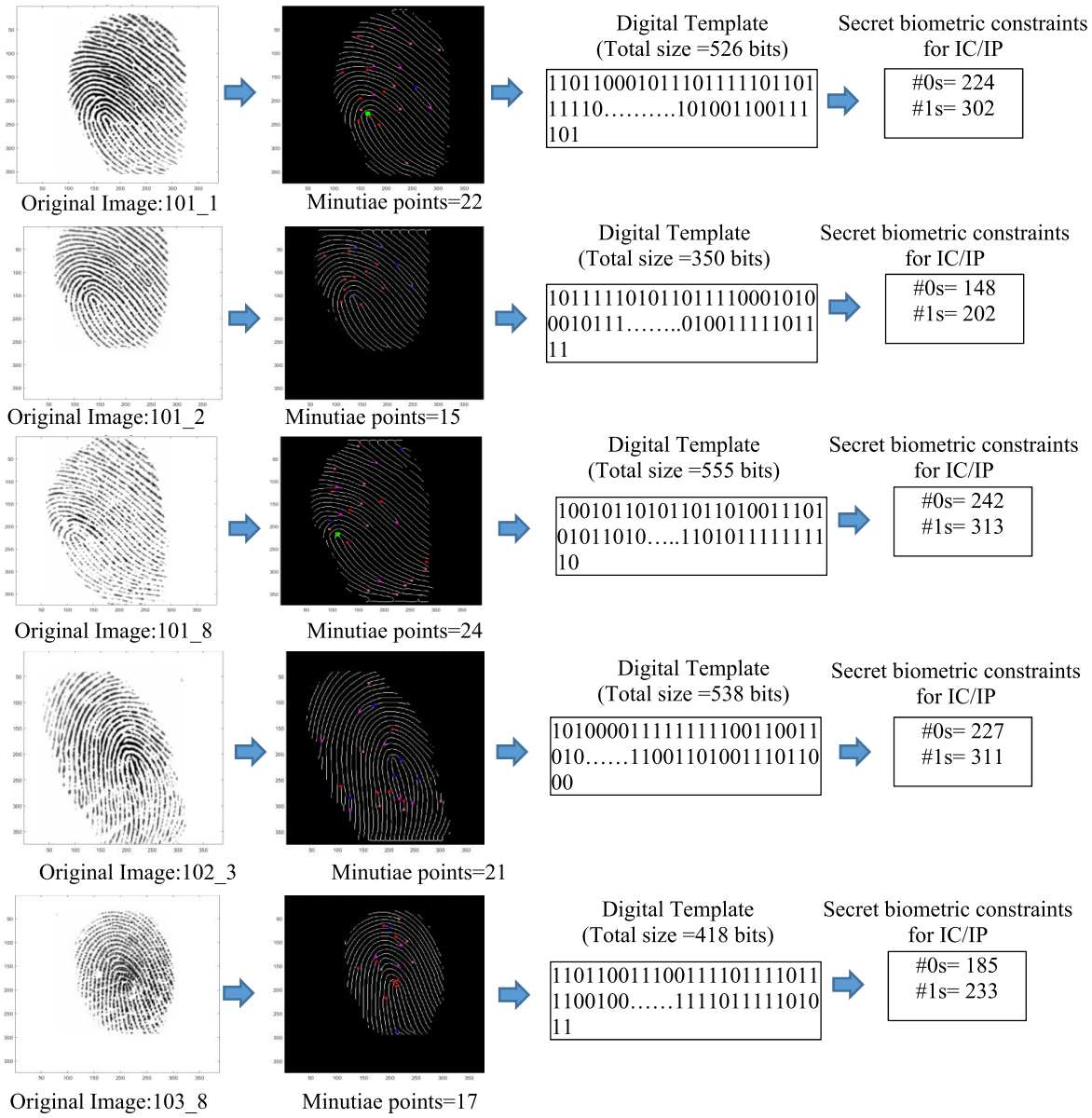


Fig. 10. Hardware security constraints generated for different fingerprints.

fingerprints selected from database [22]. As evident very low value of Pc is obtained (that is desirable for higher security). Further, Table VIII shows the impact of fingerprint of different fingers (of same person) on Pc. As shown in Table VIII, the lower Pc can be achieved by embedding the biometric fingerprint of that finger which results into higher security constraints. For example, Ring_R finger as seen in Table VIII. Further, strength of IP ownership proof in terms of Pc is compared with [20] for different size ($W$) of security-constraints. A summary of approach [20] is as follows. This approach [20] proposed security of JPEG compression hardware based on crypto-based dual phase hardware steganography, for securing against piracy threat. This approach generates stego-constraints which are further mapped to hardware security constraints based on designer's mapping rules. The process of generation of stego-constraints includes multiple steps as follows:

1) key-based state-matrix formation;
2) byte substitution using cryptographic S-box;
3) key-based row diffusion;
4) performing key-based cryptographic Trifid cipher;
5) key-based alphabet substitution;
6) matrix transformation;
7) column diffusion based on cryptographic maximum distance separable (MDS) matrix;
8) key-based byte concatenation;
9) bitstream truncation;
10) conversion of bits into stego-constraints.

The generated constraints are implanted into the register allocation and FU vendor allocation phase of ESL synthesis process. However, because of involvement of multicryptographic modules in the approach [20], both the constraint generation process and signature detection process are cumbersome and not seamless. Further, Table IX compares the complexity of [20] with respect to the proposed approach, in terms of implementation runtime (it is noted that the

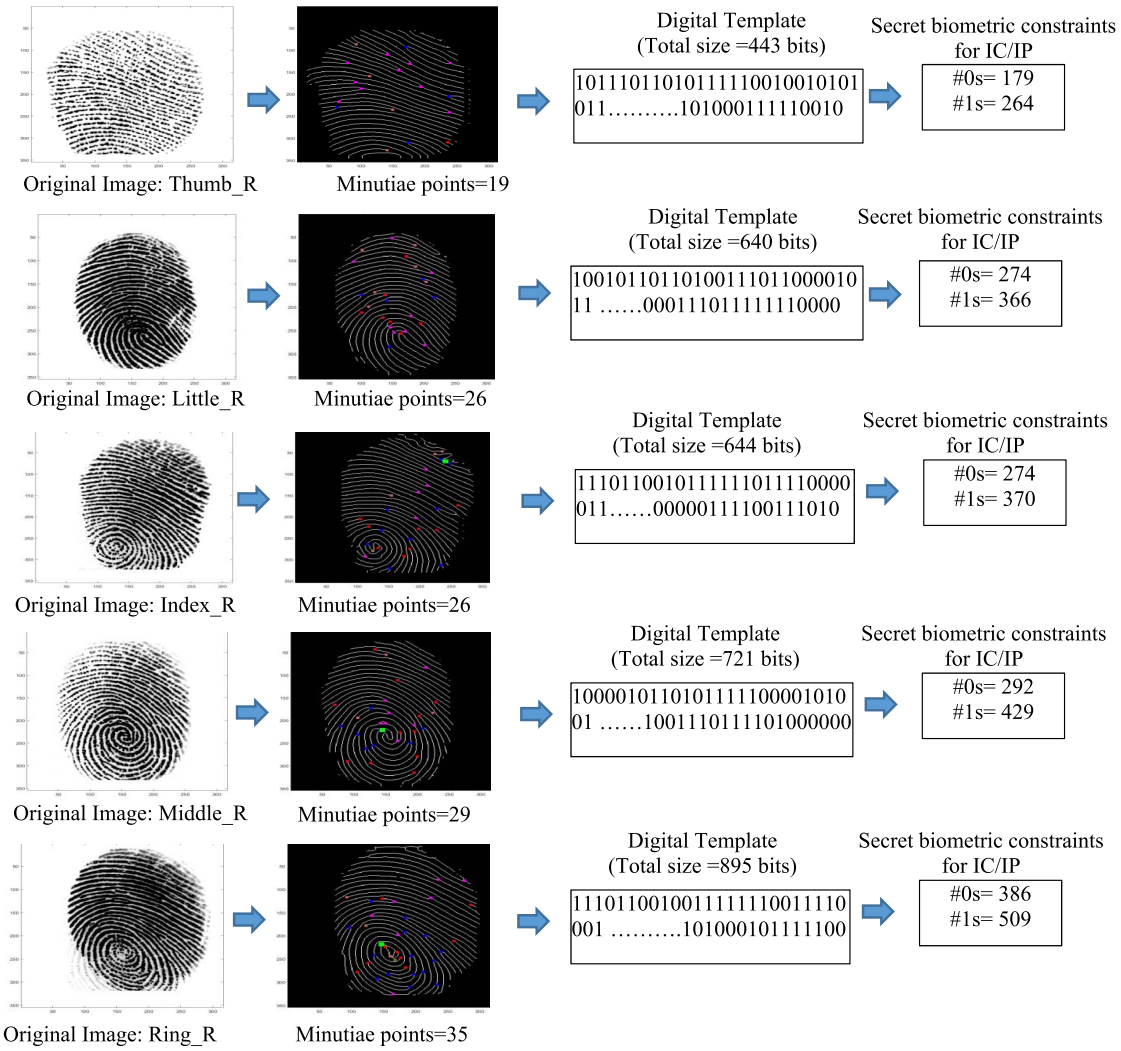


Fig. 11. Hardware security constraints generated for different fingers of the same person.

TABLE VII

VARIATION IN PC OF THE PROPOSED APPROACH FOR DIFFERENT FINGERPRINTS SELECTED FROM DATABASE

| Fingerprint image | # Minutiae points (M) | Total constraints (f1) | Probability of coincidence (Pc) |
|---|---|---|---|
| 101_1 | 22 | 526 | 7.06E-4 |
| 101_2 | 15 | 350 | 8.00E-3 |
| 101_8 | 24 | 555 | 4.73E-4 |
| 102_3 | 21 | 538 | 5.98E-4 |
| 103_8 | 17 | 418 | 3.13E-3 |
| 101_7 | 21 | 540 | 5.82E-4 |

TABLE VIII

IMPACT OF FINGERPRINT OF DIFFERENT FINGERS OF THE SAME PERSON ON PC

| Fingerprint image | # Minutiae points (M) | Total constraints (f1) | Pc |
|---|---|---|---|
| Thumb_R | 19 | 443 | 2.22E-3 |
| Little_R | 26 | 640 | 1.46E-4 |
| Index_R | 26 | 644 | 1.38E-4 |
| Middle_R | 29 | 721 | 4.79E-5 |
| Ring_R | 35 | 895 | 4.35E-6 |

system configuration for executing the proposed approach is as follows: CPU with 8-GB RAM at 2.5-GHz frequency).

TABLE IX

COMPARISON OF THE PROPOSED APPROACH WITH RELATED WORK [20]

| Approaches | Total constraints | Pc | Approximate run time |
|---|---|---|---|
| Related work [20] | W=20 | 3.0e-1 | ~400 ms |
| | W=40 | 9.8e-2 | |
| | W=60 | 1.9e-2 | |
| | W=80 | 3.7e-3 | |
| | W=100 | 1.7e-3 | |
| Proposed work | M=35 | 4.35e-6 | ~185 ms |

Further in [20], since constraints are embedded in both register allocation phase and FU vendor allocation phase of high-level synthesis, therefore, total constraints $W = f1 + f2$ is determined. The comparison of proposed approach with [20] in terms of Pc is also shown in Table IX. As shown in Table IX, the proposed approach achieves lower Pc (which is desirable) for an IP vendor fingerprint (having $M = 35$ minutiae points) than [20] due to embedding large number of security constraints ($f1 = 895$).

In addition to offering higher strength of ownership proof, the hardware security constraints corresponding to a biometric fingerprint offers higher security in following ways as well.

TABLE X
DESIGN COST OF JPEG COMPRESSION HARDWARE PRE- AND POSTEMBEDDING BIOMETRIC FINGERPRINT CONSTRAINTS

| Resource constraints | # of registers in baseline JPEG | # of registers in fingerprint implanted JPEG | Design cost of baseline JPEG | Design cost of fingerprint implanted JPEG | Estimated % overhead |
|---|---|---|---|---|---|
| 3A, 3A | 73 | 73 | 0.214 | 0.214 | 0% |
| 3A, 5M | 73 | 73 | 0.1917 | 0.1917 | 0% |
| 5A, 5M | 73 | 73 | 0.1713 | 0.1713 | 0% |
| 7A, 9M | 73 | 73 | 0.1718 | 0.1718 | 0% |
| 9A, 9M | 73 | 73 | 0.1752 | 0.1752 | 0% |
| 11A, 11M | 73 | 73 | 0.1785 | 0.1785 | 0% |

1) *Applying Brute-Force Attack to Determine Security Constraints Does Not Help to an Attacker:* This is because, the attacker cannot prove the security constraints as his/her own, even if he/she knows them. This is because the hardware security constraints always belong to the respective biometric fingerprint of the IP vendor.
2) *No Tampering of fingerprint Constraints Is Possible:* because for an attacker, generating the exact digital template corresponding to the original IP vendor's biometric fingerprint is impossible.
3) *No Possibility of Key Generation:* In the related approaches such as steganography [6], [19], [20], there exists the probability of determining the valid stego-key or entropy threshold. However, this probability is zero in the proposed approach, as it is not dependent on key-based security.
4) *Probability of Proving Fraud Claim of IP Ownership by an Attacker Is Zero:* This can be ascertained using point matching difference (MD) function given as follows:

$$\mathrm{MD} = \sum_{i,j=1}^{n} |M_i - M_j| \tag{7}$$

where $n$ denotes total number of minutiae points embedded corresponding to the original fingerprint, $M_i$ indicates decimal equivalent of a minutia point of the embedded fingerprint and $M_j$ indicates decimal equivalent of a minutia point of the fingerprint of the adversary or IP owner. In case of true IP owner, the value of MD is achieved to be zero. In case of an adversary, the value of MD is computed to be nonzero due to mismatching of minutiae points of adversary and original embedded fingerprint. The nonzero value of point MD ensures that the probability of proving fraud claim of IP ownership by an attacker is zero.

### C. Design Cost Analysis

The design cost ($C_d$) is calculated as follows [20]:

$$C_d(F_i) = w_1 \frac{D_\mathrm{T}}{D_\mathrm{max}} + w_2 \frac{A_\mathrm{T}}{A_\mathrm{max}} \tag{8}$$

where $D_\mathrm{T}$ and $A_\mathrm{T}$ indicates design delay and area, respectively, $A_\mathrm{max}$ and $D_\mathrm{max}$ are the maximum area and delay, $w_1$ and $w_2$ are the user defined weights (both kept at 0.5 to assign equal preference). The area and delay are computed using a 15-nm technology scale open cell library [27]. The design cost pre- and postembedding biometric hardware security constraints is reported in Table X. As shown in Table X, the design cost of proposed approach has been quantitatively assessed for various number of FU constraints. It is evident from Table X that the proposed approach secures the design at almost zero overhead for all the tested FU constraints. This is because, the security constraints are embedded during register allocation process and the proposed approach does not require any extra register to accommodate all hardware security constraints corresponding to an IP vendors biometric fingerprint. Thus, Table X also highlights that the proposed approach works efficiently with more number of FUs also. The proposed approach incurs negligible area overhead. The area overhead is not only negligible for demonstrated JPEG CODEC processor, but will also be for various real-life DSP and multimedia hardware accelerators designs. This is because these real-life DSP hardware accelerator applications usually have large register count in their design. Therefore local alterations in register reallocation required during implanting (accommodating) security constraints do not add any extra register overhead. Larger the DSP design, more hardware efficient the proposed approach is. Therefore, the resultant area overhead is almost negligible in most cases.

## V. CONCLUSION

A novel design methodology of securing hardware accelerators against IP piracy and false claim of IP ownership is presented. The proposed hardware security is driven through the vendor's biometric fingerprint. Although the proposed approach has been demonstrated for JPEG CODEC hardware accelerator, however, it is applicable to all kind DSP and multimedia hardware accelerators (data-intensive reusable IP cores) which are designed using ESL synthesis process. The proposed approach carries robust security features at zero design overhead when compared with recent similar approaches. This is because of the vulnerability of the signature (such as it can be compromised and is replicable) in the existing schemes. The signature in these approaches depends on variables and their encodings which can be compromised by an attacker. Further, the proposed biometric fingerprint mostly ensures negligible design overhead—this is because: 1) proposed approach embeds hardware security constraints during register allocation framework of ESL synthesis and 2) complex DSP/multimedia hardware accelerators (the target hardware) usually have large register count in their design,

thus local alterations in the register reallocation due to constraint implanting does not add any extra storage hardware overhead.

The future work aims to tackle other classes of minutiae points from biometric data during secret constraint extraction and embedding process in a hardware accelerator.

## REFERENCES


[1] R. S. Chakraborty and S. Bhunia, "HARPOON: An obfuscation-based SoC design methodology for hardware protection," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 10, pp. 1493–1502, Oct. 2009.

[2] E. Castillo, U. Meyer-Baese, A. Garcia, L. Parrilla, and A. Lloris, "IPP HDL: Efficient intellectual property protection scheme for IP cores," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 5, pp. 578–591, May 2007.

[3] B. Colombier and L. Bossuet, "Survey of hardware protection of design data for integrated circuits and intellectual properties," *IET Comput. Digit. Techn.*, vol. 8, no. 6, pp. 274–287, Nov. 2014.

[4] S. M. Plaza and I. L. Markov, "Solving the third-shift problem in IC piracy with test-aware logic locking," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 6, pp. 961–971, Jun. 2015.

[5] M. Yasin, J. J. Rajendran, O. Sinanoglu, and R. Karri, "On improving the security of logic locking," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 9, pp. 1411–1424, Sep. 2016.

[6] A. Sengupta and M. Rathor, "IP core steganography for protecting DSP kernels used in CE systems," *IEEE Trans. Consum. Electron.*, vol. 65, no. 4, pp. 506–515, Nov. 2019.

[7] R. Schneiderman, "DSPs evolving in consumer electronics applications special reports," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 6–10, May 2010.

[8] 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, Sep. 2018.

[9] J. Zhang, "A practical logic obfuscation technique for hardware security," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 3, pp. 1193–1197, Mar. 2016.

[10] Y. Lao and K. K. Parhi, "Obfuscating DSP circuits via high-level transformations," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 5, pp. 819–830, May 2015.

[11] B. K. Mohanty and P. K. Meher, "A high-performance FIR filter architecture for fixed and reconfigurable applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 2, pp. 444–452, Feb. 2016.

[12] J. Zhang, H. Wen, and L. Tang, "Improved smoothing frequency shifting and filtering algorithm for harmonic analysis with systematic error compensation," *IEEE Trans. Ind. Electron.*, vol. 66, no. 12, pp. 9500–9509, Dec. 2019.

[13] R. Chapman and T. S. Durrani, "IP protection of DSP algorithms for system on chip implementation," *IEEE Trans. Signal Process.*, vol. 48, no. 3, pp. 854–861, Mar. 2000.

[14] A. Sengupta and S. Bhadauria, "Exploring low cost optimal watermark for reusable IP cores during high level synthesis," *IEEE Access*, vol. 4, pp. 2198–2215, 2016.

[15] F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 10, no. 3, pp. 523–545, 2005.

[16] B. Le Gal and L. Bossuet, "Automatic low-cost IP watermarking technique based on output mark insertions," *Design Autom. Embedded Syst.*, vol. 16, no. 2, pp. 71–92, Jun. 2012.

[17] I. Hong and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," in *Proc. Design Autom. Conf.*, 1999, pp. 849–854, doi: 10.1109/DAC.1999.782161.

[18] B. Le Gal and L. Bossuet, "Automatic HLS based low-cost IP watermarking," in *Proc. IEEE 9th Int. New Circuits Syst. Conf.*, Jun. 2011, pp. 490–493.

[19] A. Sengupta and M. Rathor, "Crypto-based dual-phase hardware steganography for securing IP cores," *IEEE Lett. Comput. Soc.*, vol. 2, no. 4, pp. 32–35, Dec. 2019.

[20] A. Sengupta and M. Rathor, "Structural obfuscation and crypto-steganography-based secured JPEG compression hardware for medical imaging systems," *IEEE Access*, vol. 8, pp. 6543–6565, 2020.

[21] F. Zhao and X. Tang, "Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction," *Pattern Recognit.*, vol. 40, no. 4, pp. 1270–1281, 2007.

[22] V. K. Alilou. (Dec. 2019). *FingerPrint Matching: A simple approach MATLAB Central File Exchange*. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/44369-fingerprint-matching-a-simple-approach

[23] A. Sengupta, D. Roy, S. P. Mohanty, and P. Corcoran, "Low-cost obfuscated JPEG CODEC IP core for secure CE hardware," *IEEE Trans. Consum. Electron.*, vol. 64, no. 3, pp. 365–374, Aug. 2018.

[24] A. Sengupta, E. R. Kumar, and N. P. Chandra, "Embedding digital signature using encrypted-hashing for protection of DSP cores in CE," *IEEE Trans. Consum. Electron.*, vol. 65, no. 3, pp. 398–407, Aug. 2019.

[25] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "Embedding low cost optimal watermark during high level synthesis for reusable IP core protection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Montreal, QC, Canada, May 2016, pp. 974–977.

[26] D. Saha and S. Sur-Kolay, "SoC: A Real Platform for IP Reuse, IP Infringement, and IP Protection," *CAD Gigascale SoC Des. Verification Solutions*, vol. 2011, Apr. 2011, Art. no. 731957, doi: 10.1155/2011/731957.

[27] (Jan. 2020). *15 nm Open Cell Library*. [Online]. Available: https://si2.org/open-cell-library/


**Anirban Sengupta** (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Ryerson University, Toronto, ON, Canada, in 2012.

He is currently a Tenured Associate Professor of Computer Science and Engineering with IIT Indore, Indore, India. He has authored over 230 publications and multiple books from IET on *IP Core Protection and Hardware-Assisted Security for Consumer Electronics* published in 2019 and *Frontiers in Securing IP Cores: Forensic Detective Control and Obfuscation Techniques* published in 2020, and an edited book from Springer on *VLSI Design and Test* in 2020.

Dr. Sengupta is an Elected Fellow of IET and a fellow of the British Computer Society. He was a recipient of the 2018 IEEE Chester Sall Memorial Consumer Electronics Award. He has been awarded prestigious IEEE Distinguished Lecturer by the IEEE Consumer Electronics Society in 2017 and the IEEE Distinguished Visitor by the IEEE Computer Society in 2019. He is currently the Deputy Editor-in-Chief of the *IET Computers & Digital Techniques Journal* and the Editor-in-Chief of IEEE VLSI CIRCUITS & SYSTEMS LETTER of the IEEE Computer Society TCVLSI. He is currently the Chair of the IEEE Computer Society TCVLSI. He currently serves/served in several editorial positions as a Senior Editor, an Associate Editor, an Editor, and a Guest Editor of several IEEE Transactions/Journals, and *IET* and *Elsevier Journals*, including the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, IEEE ACCESS, the *IET Computer & Digital Techniques Journal*, the IEEE CONSUMER ELECTRONICS, the IEEE CANADIAN JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING, and the *Elsevier Microelectronics Journal*. He served as the General Chair of the 37th IEEE ICCE 2019, Las Vegas, Nevada, and the 23rd International Symposium on VLSI Design and Test (VDAT-2019), India.

**Mahendra Rathor** (Member, IEEE) received the M.E. degree in electronics engineering from Devi Ahilya Vishwavidyalaya (DAVV), Indore, India, in 2014. He is currently working toward the Ph.D. degree in computer science and engineering at IIT Indore, Indore.