IP CORE PROTECTION AND DETECTIVE CONTROL OF DATA-INTENSIVE HARDWARE IPs AGAINST PIRACY

Ph.D. Thesis

By
RAHUL CHAURASIA

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE
MAY 2023

IP CORE PROTECTION AND DETECTIVE CONTROL OF DATA-INTENSIVE HARDWARE IPs AGAINST PIRACY

A THESIS

Submitted in partial fulfillment of the
requirements for the award of the degree
of
DOCTOR OF PHILOSOPHY

by
RAHUL CHAURASIA

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE
MAY 2023

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **IP CORE PROTECTION AND DETECTIVE CONTROL OF DATA-INTENSIVE HARDWARE IPs AGAINST PIRACY** in the partial fulfillment of the requirements for the award of the degree of **DOCTOR OF PHILOSOPHY** and submitted in the **DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING, INDIAN INSTITUTE OF TECHNOLOGY INDORE**, is an authentic record of my own work carried out during the time period from DECEMBER, 2020 to MAY, 2023 under the supervision of Dr. ANIRBAN SENGUPTA, Associate Professor, Indian Institute of Technology, Indore.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

14-Aug.-2023
Signature of the student with date
(RAHUL CHAURASIA)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Aug 21, 2023
Signature of Thesis Supervisor with date
(ANIRBAN SENGUPTA)

RAHUL CHAURASIA has successfully given his/her Ph.D. Oral Examination held on

Aug 21, 2023		Aug 21, 2023
Signature of Chairperson (OEB)	Signature of External Examiner	Signature(s) of Thesis Supervisor(s)
Date:	Date: 31-08-2023	Date:
21/8/2023		
Signature of PSPC Member #1	Signature of PSPC Member #2	Signature of PSPC Member #3
Date:	Date:21/08/2023	Date:
	Somnath Dey	
Signature of Convener, DPGC	Signature of Head of Discipline	
Date: 23/08/2023	Date: 21/08/2023	

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank a number of persons who in one or the other way contributed by making this time as learnable and enjoyable as possible. First and foremost, I wish to express my sincere gratitude to my supervisor **Dr. Anirban Sengupta** for providing me the opportunity to do work under his supervision. I wish to thank him for his persistence and faith in me. Without his relentless effort and guidance, I would not have been to understand the importance of research and the sacrifice it requires to reach a certain level. Under his supervision, I transformed into a person with more values and learned the importance of work ethics and scientific temperament.

I also owe a mention to **Dr. Abhishek Srivastava** and **Dr. Sampak Samanta** for their valuable feedbacks on my research work throughout these years.

I am also grateful to **Dr. Somnath Dey**, Head of the Department of Computer Science & Engineering, for extending all necessary support to me. My sincere acknowledgment and respect to **Prof. Suhas Joshi**, Director, Indian Institute of Technology Indore, for providing me the opportunity to explore my research capabilities at the Indian Institute of Technology Indore.

I wish to thank all the faculty members, my lab mates Dr. Mahendra Rathor, Aditya Anshul, and my friends for their continuous support. I am grateful to the respected Mr. Utpal Ghosh (Baba), Mrs. Purnima Ghosh (Maa), and Ms. Neha Ghosh for their exceptional support, care and encouragement.

Further, I wish to express my deepest gratitude to my parents, for their strong belief in me and for their continuous support all the way. I also wish to express my gratitude to my caring sisters, Priyanka and Deeksha and my brother Rajesh for being the driving force of my career and for her moral and emotional support throughout the PhD work. I am thankful to all for being with me every single moment to keep me motivated to work for the past years.

At last, I wish to thank IIT Indore and UGC for helping me financially and providing an opportunity to present my research at international platforms.

Rahul Chaurasia

DEDICATED TO MY PARENTS

MALTI CHAURASIA &

LATE SHRI VIJAY KUMAR CHAURASIA (V.K.C)

ABSTRACT

Hardware IP core-based design paradigm has become popular for its usage in several consumer electronics and computing systems. This is because the usage of hardware IP cores enables higher performance and efficacy by accelerating the underlying process of the respective application. Further, due to their data-intensive nature and factors such as time to market pressure, process turnaround time and design complexity are some of the major reasons that have enforced or encouraged reusable IP core-based system-on-chip (SoC) designs. This scenario leads to the involvement of third-party IP vendors to match the demand and supply ratio or to accelerate the design process, thereby making it susceptible to different hardware security threats. An adversary in the untrusted offshore design house may pirate the IP core(s) for their own benefit or to satisfy malicious intentions, causing security and integrity hazards to the end consumer.

Digital signal processing (DSP), multimedia and machine learning applications are thriving in the modern consumer electronics (CE) market. These IP cores are used for facilitating several crucial applications in the domain of health care, robotics and artificial intelligence (AI) etc. Hence, they have become an important and integral part of modern electronic/automated devices. Therefore, the current generations of system-on-chip (SoC) designers amalgamate reusable IP cores imported from multiple IP vendors/manufacturers. These IP cores are mass-produced, tested and verified by various companies and this IP supply chain is distributed worldwide. Therefore, due to the involvement of multi-party vendors, their security concerns cannot be undervalued. Hence, an IP core designer needs to employ robust and seamless security measures against security threats to ensure trust in hardware IP. For DSP, multimedia and machine learning based applications which are highly complex or data-intensive in nature, their realization as reusable hardware IP cores is crucial. Further, to ensure their security against hardware threats, a high-level synthesis (HLS) framework is conducive for integrating security mechanisms. HLS offers lesser design complexity and flexibility to integrate security mechanisms. Therefore, enabling an IP designer to achieve robust security while incurring negligible or lower design

cost overhead concurrently. Towards the security of IP cores, this thesis contributes the following: (a) contact-less palmprint biometric for securing DSP coprocessors used in CE systems against IP piracy, (b) a double line of defense approach for securing DSP IP cores using structural obfuscation and chromosomal DNA impression, (c) designing secured reusable convolutional IP core in convolutional neural network (CNN) using facial biometric based hardware security approach, (d) Retinal biometric based secured JPEG-codec hardware IP core design for CE systems using HLS and (e) exploration of security-cost tradeoff for signature driven security algorithms for optimal architecture of data-intensive hardware IPs.

LIST OF PUBLICATIONS (21)

PEER-REVIEWED JOURNALS (8):

1. A. Sengupta, R. Chaurasia and T. Reddy, "Contact-Less Palmprint Biometric for Securing DSP Coprocessors Used in CE Systems," ***IEEE Trans. Consum. Electron.***, vol. 67, no. 3, pp. 202-213, Aug. 2021, doi: 10.1109/TCE.2021.3105113. **(Impact Factor: 4.414)**

2. A. Sengupta and R. Chaurasia, "Secured Convolutional Layer IP Core in Convolutional Neural Network Using Facial Biometric," ***IEEE Trans. Consum. Electron***., vol. 68, no. 3, pp. 291-306, Aug. 2022, doi: 10.1109/TCE.2022.3190069. **(Impact Factor: 4.414)**

3. R. Chaurasia and A. Sengupta, "Retinal Biometric for Securing JPEG-Codec Hardware IP core for CE systems," ***IEEE Trans. Consum. Electron***., doi: 10.1109/TCE.2023.3264669. **(Impact Factor: 4.414)**

4. 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.,*** 2023, doi: 10.1109/TVLSI.2023.3265559. **(Impact Factor: 2.775)**

5. M. Rathor**,** A. Sengupta, R. Chaurasia and A. Anshul "Exploring Handwritten Signature Image Features for Hardware Security", ***IEEE Trans. Dependable Secure Comput. (TDSC)***, 2022, doi: 10.1109/TDSC.2022.3218506. **(Impact Factor: 6.791)**

6. A. Sengupta and R. Chaurasia, "Securing IP Cores for DSP Applications Using Structural Obfuscation and Chromosomal DNA Impression," ***IEEE Access***, vol. 10, pp. 50903-50913, 2022, doi: 10.1109/ACCESS.2022.3174349. **(Impact Factor: 3.476)**.

7. R. Chaurasia, A. Anshul, A. Sengupta and S. Gupta, "Palmprint Biometric Versus Encrypted Hash Based Digital Signature for Securing DSP Cores Used in CE Systems," ***IEEE Consum. Electron. Mag. (CEM)***, vol. 11, no. 5, pp. 73-80, 1 Sept. 2022, doi: 10.1109/MCE.2022.3153276. **(Impact Factor: 4.135)**

8. M. Rathor, A. Anshul**,** K. Bharath, R. Chaurasia and A. Sengupta "Quadruple Phase Watermarking during High Level Synthesis for Securing Reusable Hardware IP Cores", "***Elsevier Journal Comput. Electr. Eng.****",* vol. 105, 2023, doi.org/10.1016/j.compeleceng.2022.108476. **(Impact Factor: 4.152)**

BOOK CHAPTERS (8):

9. **R. Chaurasia**, A. Sengupta and P. Pradeeprao "Secured Integrated Circuit (IC/IP) Design Flow", *CRC Book "**Nanoelectronics for Next-generation Integrated Circuits***", 2022, eBook ISBN9781003155751.

10. A. Sengupta and **R. Chaurasia** "Hardware IP Cores for Image Processing Functions", IOP Book "***Advances in Image and Data Processing using VLSI Design***", 2022, pp. 7.1 - 7.14, doi: 10.1088/978-0-7503-3919-3ch7.

11. A. Sengupta and **R. Chaurasia** "Integrated Defense using Structural obfuscation and Encrypted DNA based Biometric for Hardware Security**,"** ***IET Book "Physical Biometrics for Hardware Security of DSP and Machine Learning Coprocessors"***, 2023, Chap. 2, pp. 25-56, doi: 10.1049/PBCS080E_ch2.

12. A. Anshul**, R. Chaurasia** and A. Sengupta "Securing Hardware Coprocessors against Piracy using Biometrics for Secured IoT systems," ***IET Book "Artificial Intelligence for Biometrics and Cybersecurity***", 2022, Accepted.

13. A. Sengupta and **R. Chaurasia** "Facial Signature based Biometrics for Hardware Security and IP Core protection," ***IET Book "Physical Biometrics for Hardware Security of DSP and Machine Learning Coprocessors"***, 2023, Chap. 3, pp. 57-92, doi: 10.1049/PBCS080E_ch3.

14. A. Sengupta and **R. Chaurasia** "Secured Convolutional Layer Hardware Co-processor in Convolutional Neural Network (CNN) using Facial Biometric," ***IET Book "Physical Biometrics for Hardware Security of DSP and Machine Learning Coprocessors"***, 2023, Chap. 4, pp. 93-146, doi: 10.1049/PBCS080E_ch4.

15. A. Sengupta and **R. Chaurasia** "Handling Symmetrical IP Core Protection and IP Protection (IPP) of Trojan Secured Designs in HLS using Physical Biometrics," ***IET Book "Physical Biometrics for Hardware Security of DSP and Machine Learning Coprocessors"***, 2023, Chap. 5, pp. 147-198, doi: 10.1049/PBCS080E_ch5.

16. A. Sengupta and **R. Chaurasia** "Methodology for Exploration of Security-Design Cost Tradeoff for Signature-based Security Algorithms," ***IET Book "Physical Biometrics for Hardware Security of DSP and Machine Learning Coprocessors"***, 2023, Chap. 8, pp. 259-297, doi: 10.1049/PBCS080E_ch8.

PEER- REVIEWED CONFERENCE PUBLICATIONS (5):

17. **R. Chaurasia** and A. Sengupta, "Securing Reusable Hardware IP cores using Palmprint Biometric," ***2021 IEEE International Symposium on Smart Electronic Systems (iSES)***, 2021, pp. 410-413, doi: 10.1109/iSES52644.2021.00099.

18. **R. Chaurasia** and A. Sengupta, "Crypto-Genome Signature for Securing Hardware Accelerators," ***2022 IEEE 19th India Council International Conference (INDICON)***, India, 2022, pp. 1-6, doi: 10.1109/INDICON56171.2022.10039955.

19. **R. Chaurasia** and A. Sengupta, "Protecting Trojan Secured DSP cores against IP piracy using Facial Biometrics," ***2022 IEEE 19th India Council International Conference (INDICON)***, India, 2022, pp. 1-6, doi: 10.1109/INDICON56171.2022.10039864.

20. **R. Chaurasia** and A. Sengupta, "Security Vs Design Cost of Signature Driven Security Methodologies for Reusable Hardware IP Core," ***2022 IEEE International Symposium on Smart Electronic Systems (iSES)***, India, 2022, pp. 283-288, doi: 10.1109/iSES54909.2022.00064.

21. **R. Chaurasia** and A. Sengupta, "Symmetrical Protection of Ownership Right's for IP Buyer and IP Vendor using Facial Biometric Pairing," ***2022 IEEE International Symposium on Smart Electronic Systems (iSES)***, India, 2022, pp. 272-277, doi: 10.1109/iSES54909.2022.00062.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

x	Number of colors in the CIG or the number of registers into register allocation table
z	Number of constraint edges implanted into the CIG
Pc	Probability of coincidence
W	Number of types of digits in the signature
Pc	Probability of coincidence
S	Signature size
A_d/ K_d	Design area
L_d/T_d	Design latency
A_m	Maximum design area
L_m	Maximum design latency
Z	Number of encryption rounds
O_y	Output value of each element/pixel corresponding to output feature map
D	Number of padding bits
q	Stride
K	Number of filter kernels
S	The size of the filter
V	Input volume image size
TT	Tamper tolerance
Cq	Quantization coefficient
C_T	Terminating criteria
P_S	Population size
S_{Gb}	Global best resource
S_{id}	The current position of i^{th} particle in dimension 'd'
α	Minimum resource value
β	Maximum resource value
γ	Any random number value of resources between 'α' and 'β'
vid	The velocity of i^{th} particle d^{th} dimension in current or previous iteration
vid^+	The velocity of i^{th} particle d^{th} dimension in next iteration
S_{lb}	Local best resource
S_{lbi}	Local best solution of the ith particle
Z_c	Design cost

ACRONYMS

HLS	High Level Synthesis
VLSI	Very Large Scale Integration
IP	Intellectual Property
DSP	Digital Signal Processor
CE	Consumer Electronics
IC	Integrated Circuits
SoC	System on Chip
RTL	Register Transfer Level
VHDL	Very High Speed Integrated Circuit Hardware Description Language
GDS	Graphic Database System
ALU	Arithmetic Logic Unit
DFG	Data Flow Graph
CDFG	Control Data Flow Graph
SDFG	Scheduled Data Flow Graph
FSM	Finite State Machine
RE	Reverse Engineering
CS	Control Step
CIG	Colored Interval Graph
UF	Unrolling Factor
3PIP	3rd party Intellectual Property
ASIC	Application Specific Integrated Circuit
PSO	Particle Swarm Optimization
DSE	Design Space Exploration
FU	Functional Unit
HLT	High Level Transformation
opn	Operation
LU	Loop Unrolling
THT	Tree Height Transformation
ROE	Redundant Operation Elimination
LICM	Loop Invariant Code Motion
SHA	Secure Hash Algorithm
DCT	Discrete Cosine Transform

IDCT	Inverse Discrete Cosine Transform
DWT	Discrete Wavelet Transform
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
DFT	Discrete Fourier Transform
JPEG	Joint Photographic Expert Group
MPEG	Moving Picture Expert Group
CE	Consumer electronics
Mux	Multiplexer
Demux	Demultiplexer
3PIP	Third-party IP
DNA	Deoxyribonucleic acid
CNN	Convolutional neural network
MAC	Multiply accumulate
FPGA	Field programmable gate array
SE	Signature embedding

Chapter 1

Introduction

We are the most privileged human generation as we live in the era of smart technology, thanks to our scientists and researchers. In this era, the contribution of electronic systems has played a pivotal role in achieving the desired goal and fulfilling the vision of availing smart and affordable technology to everyone. In this modern era, where everyone intends to have faster and low-cost processing of their tasks either in regards of an application or a system/device, the need to develop such systems/devices is prevailing. One can easily observe several consumer electronics and computing systems such as smartphones, smart watches, tablets, digital cameras, computers and audio headsets etc. are part of our lifestyle and also have become a necessity. These computing/CE systems are ubiquitously used for performing various tasks/applications based on image processing, audio-video processing etc. However, underneath these computing/CE systems, there functions a system-on-chip (SoC). An SoC is designed using various modules such as functional blocks, memory units and memory controllers and different peripherals for wireless and wired communication etc. In deployed practice, instead of designing an SoC from scratch, its various modules/cores are purchased from third-party IP (3PIP) vendors or designers. And this kind of system design paradigm is called as core-based design paradigm [1]-[11].

In computing devices and systems, for performing data-intensive tasks, hardware accelerators are used to achieve higher performance and efficacy by accelerating the underlying process [12]. During the acceleration process of an application, certain computing tasks are offloaded into specialized hardware components, typically known as hardware accelerators or intellectual property (IP) cores. A hardware IP is a reusable unit (block of data/logic) of computational function, Boolean logic, register transfer level (RTL), or a gate structure, and is also known as the intellectual property of a designer. There are various applications for example, cryptographic applications are performed using cryptographic IP cores, while fingerprint, face recognition, and handprint biometrics require digital signal processing (DSP) and image

processing IP cores. Further Artificial Intelligence (AI) applications require AI cores, sound processing via sound card and digital signal processing via digital signal co-processor etc. Further, in computing and CE systems, different applications such as image compression-decompression, audio de-noising and video processing etc. (which are data intensive in nature) are facilitated using different IP cores with higher efficacy and at lower design cost. The IP cores employ the execution of different algorithms such as discrete cosine transformation (DCT), fast fourier transform (FFT), finite impulse response etc., used for digital signal processing (DSP), machine learning and multimedia processing etc., which are highly data-intensive in nature [13], [14].

Therefore, due to design complexity, design cost and time-to-market pressure, these application frameworks are realized as reusable IP cores. This therefore enables cost reduction and elevates design turnaround time. Therefore, current generation system-on-chip (SoC) designers amalgamate reusable IP cores imported from multiple IP vendors/manufacturers. These IP cores are mass-produced, tested and verified by various companies and the IP supply chain is distributed worldwide.

Further, from the perspective of the researcher as well as user, it becomes equally crucial to understand the process of designing and developing such systems. The design cycle of such systems involves several design phases and different entities. The different design phases may be categorized based on design complexity, designing cost and flexibility. Therefore, it becomes crucial to have an understanding of different design phases. Further, the involved entities can also be categorized in terms of their role in the design chain and trustworthiness. Different entities (third-party IP vendors, system integrators, and foundry) get involvement in the IC design chain. This helps in sustaining the IC design process at a lesser cost, lower design complexity and lower time requirement [12]. However, it also enforces to incorporate security measures to safeguard the designs against security hazards to ensure their safe usage to end consumers.

However, the involvement of distinct entities (or offshore design houses) in the design chain raises the issue of trust [15]-[26]. This is because an adversary or attacker in an untrustworthy design house may realize his/her malicious intent of IP piracy. Additionally, security against fraudulent claim of IP ownership, implantation of hidden malicious logic by reverse engineering the design, and protection of IP rights of IP buyer and seller are crucial. As the DSP, multimedia and machine learning based IP cores possess significant role in CE systems, mission-critical tasks, IoT devices and healthcare applications, therefore their security perspective cannot be overlooked. This is because integration of a pirated IP version into SoCs of such systems may lead security and integrity hazards to end consumers.

This chapter in a nutshell, discusses the background on the various key aspects that the proposed hardware security techniques are developed around. The first section provides the background on different design abstraction levels of an IP core. Further, the second section provides an overview of DSP, machine learning and multimedia-based data-intensive applications and corresponding algorithmic representations. The third section discusses the various threats to reusable data-intensive hardware IP cores. The fourth section provides a background on high-level synthesis (HLS) process and its role in designing low-cost and secured reusable hardware IPs. In the end, the fifth section presents the thesis organization.

1.1. Different design abstraction levels and corresponding form of hardware IP core

Due to the higher complexity involved, it is crucial to design an IP core from a higher abstraction level of IC design process. This is because the higher abstraction level offers lesser complexity and higher flexibility to incorporate the low-cost architecture and robust security mechanism than the lower design abstraction levels. The design abstraction levels are as follows [81], [82]: (a) system/behavioral level (b) register transfer level (c) gate level or netlist level and (d) layout or transistor level.

The top most design abstraction level is the behavioral level. At this level design/application is described based on the respected inputs, output and

transfer function or behavioral description. The behavioral/mathematical function of an application is accepted as input for transforming it into next level design. Therefore, the algorithmic description of the design/application is transformed into a register transfer level using high-level synthesis (HLS). Additionally, integrating the security mechanism during higher abstraction also is less complex as well as ensures security at subsequent lower abstraction level design versions. The design obtained post-HLS (RT level design version) is termed as soft IP. In other words, IP cores which are generally available as synthesizable register transfer level code in the form of either schematic design (.bdf file) or hardware description language (.vhd/.vhdl file), are called as soft IP. One of the advantages of the soft IP cores is that they offer a chip designer the flexibility to modify the design parameters as per the requirement.

Further, the next design abstraction level is the gate level or netlist level. IP design at this level is obtained by transforming register transfer (RT) level design into gate-level design using logic synthesis or RTL synthesis. It describes the design interconnectivity in terms of various cells that are present with in it and the output of the synthesis process at the logic level. The gate level netlist of the design is called as firm IP core. This IP version is technology dependent and is lesser modifiable than a soft IP core. RTL and gate-level netlist both allow post-synthesis processing steps such as placement, routing, and downloading into reconfigurable platforms such as (field programmable gate arrays) FPGAs.

Subsequently, the next design abstraction level is the transistor level. IP design at this level is obtained by transforming gate-level design into layout-level design using layout synthesis. The IP design version at this level is known as hard IP. Hard IP cores are generally available as a layout format (fixed masked layout) of chip designs in the graphic data system (GDS) or layout editor documentation (LEF) format. Unlike soft IP cores, hard IPs cannot be modified by chip designers or system integrators. Further, the demerit of a hard IP design is that it does not allow to be used in another foundry (for fabrication) for which it is not targeted to. This is because design at the layout level comprises of process foundries and a design rule, which incapacitates the use of layout in another foundry except to whom it was targeted. Therefore,

due to more flexibility (in terms of modifying functionality) and greater portability (can be reused), soft IPs are preferred over Hard IPs. However, soft IP cores are exposed to greater IP protection risk than hard IPs as they can be modified by system integrators. Thus, it is interpretable that an IP core is designed and sold into the market in one of its forms, such as (i) soft IP core (ii) firm IP core (iii) hard IP core.

However, based on the computational capability and design size, they are categorized into two different types: micro-IPs and macro-IPs (are essentially bigger logic). Logic gates, combinational and sequential circuits (register and memory) are some of examples of micro-IPs. On the other hand, digital signal processors (DSPs), central processing units (CPUs) and application-specific cores such as joint photographer expert group (JPEG) engines, moving picture expert group (MPEG) engines, digital filters like finite impulse response (FIR) filter and infinite impulse response (IIR) filter, falls under the category of macro-IPs. These DSP cores facilitate several applications like image compression-decompression, digital data filtration and audio processing etc., which are computationally intensive in nature.

1.2. DSP, machine learning and multimedia-based applications and their algorithmic representation

In the DSP co-processors, there functions a DSP algorithm for performing the corresponding to application/task. Some widely used DSP algorithms are discrete cosine transform (DCT), discrete Fourier transform (DFT), fast Fourier transform (FFT), Haar wavelet transform (HWT), discrete wavelet transform (DWT), inverse discrete cosine transform (IDCT). DCT is used while converting an image from a spatial domain to its frequency domain. Further, it is the basic fundamental algorithm for performing image compression-decompression in JPEG-codec co-processors. DFT, and FFT are used for representing a discrete signal from its time domain to the frequency domain. HWT is used for transforming the waveform of a signal from time domain to time-frequency. It is widely used for both lossy and lossless signal and image compression-based applications. DWT is used for performing the denoising of the real signal by decomposing it. It basically decomposes a

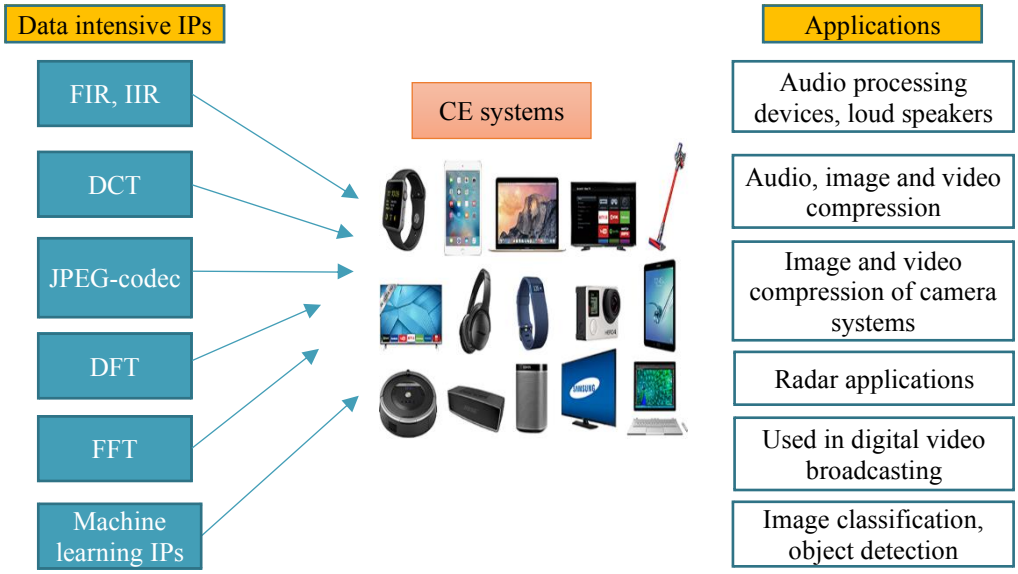


Fig. 1.1 Different data intensive IPs and their applications

digital signal to obtain finer frequency and coarser time resolution based on different sub-bands. It is the basic fundamental algorithm for performing image compression in JPEG2000. Further, digital filters like finite impulse response (FIR) filters and infinite impulse response (IIR) filters have wide utility in modern electronic systems. For example, they are used in speech processing, telecommunication, removal of attenuation of selected frequencies, etc. Different data-intensive hardware IP cores and their usages are shown in Fig. 1.1.

Further, machine learning IP cores are used for performing different tasks related to it. On the other hand, in multimedia processors, various multimedia processing algorithms work, such as joint photographic experts group compression-decompression (JPEG-codec) and moving picture experts' group (MPEG) etc. JPEG is used for performing image compression. In order to do so, it firstly converts an input image from a spatial domain to a frequency domain. Subsequently, by performing the quantization (discarding less important frequency components), it results into a compressed image. It is widely used in medical imaging, digital camera systems etc.

In order to generate an application-specific processor of data-intensive applications, its algorithmic or behavioral description is processed as input for the synthesis process [12], [91]. The algorithmic description can be of various forms, such as a C/C++ code or transfer function, or a mathematical equation representing input-output relationship) etc. For example, an algorithmic

description of FIR application in the form of a mathematical function is given as follows [81]:

$$\boldsymbol{B}[\boldsymbol{n}] = \sum_{i=0}^{N} c[i] * \boldsymbol{A}[\boldsymbol{n} - \boldsymbol{i}] \quad (1.1)$$

Where, N represents the order of the FIR filter. Further, the mathematical equation based on the order of FIR filter, can be represented as follows:

$$\boldsymbol{B}[\boldsymbol{n}] = c[0] * \boldsymbol{A}[\boldsymbol{n}] + c[1] * \boldsymbol{A}[\boldsymbol{n-1}] + c[2] * \boldsymbol{A}[\boldsymbol{n-2}] + \cdots . + c[N] * \boldsymbol{A}[\boldsymbol{n-N}] \quad (1.2)$$

Where, $\boldsymbol{A}[n]$ to $\boldsymbol{B}[n]$ represents the current input-output and **A[*n*-1]**, **A[*n*-2]** represents the previous input values and, $c[0], c[1] \ldots c[N]$ indicates input coefficients of the FIR. This mathematical description is exploited for generating the application-specific hardware co-processor design of FIR filter.

1.3. Threats to reusable data intensive hardware IP cores

As discussed earlier, in the deployed semiconductor design chain, various offshore entities such as a 3PIP vendor, a system integrator and foundry houses are involved. This is to speed up the design process for attaining the goals of low-design cost, shorter design time and time to market etc. Therefore, IP cores may be sold/supplied by different IP vendors. Based upon the design requirements, these IPs are supplied to an SoC integrator for their integration into SoC design or else they are directly supplied to foundry houses for fabrication as a standalone IC. Thus, after the integration of IPs at SoC integrator house, it is supplied to the foundry house(s) for their fabrication. In other words, the data flow in the design cycle in unidirectional e.g., from IP vendor to SoC integrator house to foundry house. More explicitly, there can be multiple IP vendors for providing the IP design and there can also be multiple foundry houses where fabrication can be done. This involvement of multiple entities in the IC design chain renders it vulnerable to different hardware security threats [15]-[26], [27]-[36], [53], [54]. The different entities involved in the design chain and possible hardware security threats are shown in Fig. 1.2.

In year 2007 and 2008, joint intellectual property rights enforcement operations (I and II respectively) was carried out by United States Customs and Border Protection (CBP) and European Union Customs. They seized lakhs of counterfeited ICs and computer network components. However, this is not the complete figure of the counterfeited parts that might have been supplied in that period. In 2010, VisionTech company owner and its administrative manager was charged for deliberately involving in trafficking of counterfeited goods [88]. They were found responsible for importing thousands of shipments of counterfeited semiconductors into the United States. They targeted the US Navy and defense contractors. When this conspiracy got detected, it was realized that how a rogue broker attempted to compromise national security and life of countless individuals on risk nearly for half a decade. It was estimated that VisionTech caused the damage to 21 semiconductor companies by supplying them the counterfeited components. In 2012, a market research firm 'iHS iSuppli' reported that the counterfeited components caused multibillion-dollar loss to the global electronics supply chain. In 2016, Dutch customs and European Union (EU) executed an operation for targeting the semiconductors supply into EU from China and Hong Kong. They seized more than one million counterfeited devices within few weeks span. Further, the report of world semiconductor council (WSC)

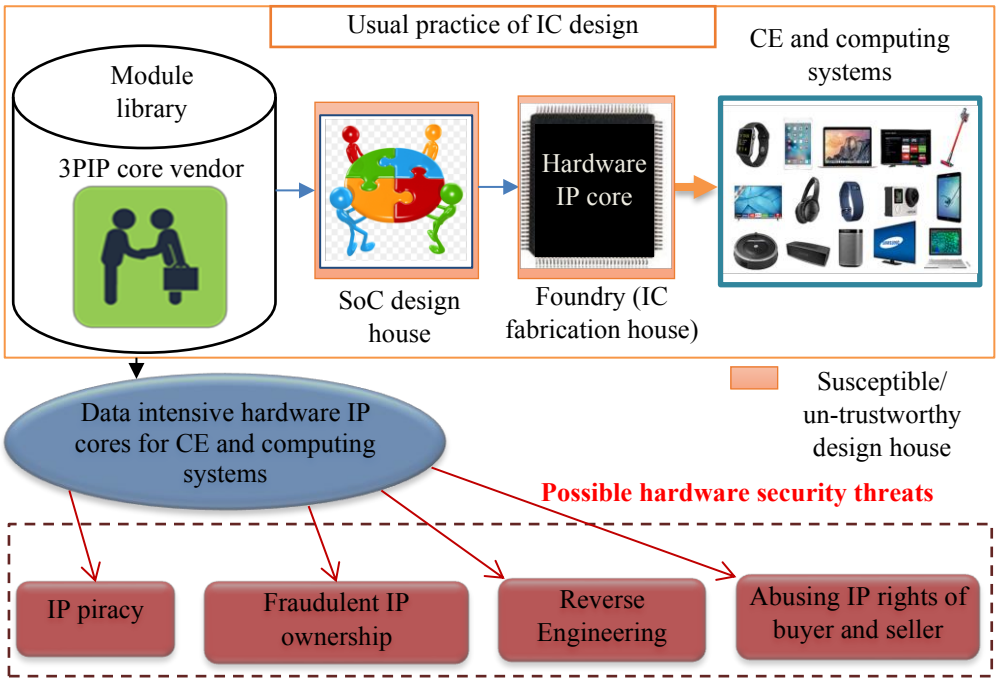


Fig. 1.2 Possible hardware security threats in IC design chain

published in 2018, states that counterfeited components significantly jeopardized security and the economy. Pirated components are responsible for the wastage of billion dollars of by semiconductor companies per year to ensure the reliable operation of customer applications [89], [90]. However, it is not possible to accurately determine the impact of semiconductor counterfeiting. But the data or reports surface the criticality of the issue. This raises serious concerns of trust in the global IC supply chain. A brief discussion on the hardware security threats is as follows:

1.3.1 IP Core Piracy

The design process of an IP core for multi-modal CE designs involves many man-hours of research, investment, validation and effort. Therefore, in the modern design cycle, multiple offshore entities are involved to cut down the overall design cost, design complexity and time-to-market. However, this involvement of offshore design houses or foundries in the design chain has posed serious hardware threats of IP piracy. A SoC integrator may purchase IP cores (to be integrated) either directly from an IP vendor or else from a broker (acting as a middleman between IP designer and the SoC integrator). However, national interest or yearning to earn illegal income may trigger a rouge IP supplier to infuse pirated or fake components (IPs) in the design supply chain. The use the fake components (pretending to be genuine) in the SoCs of CE devices may adversely impact both CE system integrator and end user. Further, ensuring security against IP piracy threat is highly important for consumers because of the following reasons: (i) counterfeited designs are not rigorously tested for ensuring reliability and security (ii) counterfeit IPs contain secret malicious logic (hardware Trojans) hidden inside. These infected IPs or ICs are unreliable and unsafe for end consumers when integrated in CE systems. Therefore, it is crucial to discern between authentic and fake IP versions for enabling the use of only authentic IPs in the CE and computing systems [31]-[41].

1.3.2. Fraudulent Ownership Claim of IP Core

A deceitful IP buyer or an adversary (may present in a foundry) present in the IC supply chain may fraudulently claim the IP ownership. This may lead to

huge financial loss for the original IP owner. Therefore, false claim of ownership is a surging security concern. The standard IP protection mechanisms such as copyright, patent, trademark, industrial design rights etc. are not applicable for reusable IP cores designs. Therefore, it is crucial to ensure the protection of the ownership rights of actual owner. In such scenarios, implanting designers' signature secretly in the IP core during its design process can be useful for proving the ownership right of an IP vendor and nullifying the fraudulent IP ownership claim by an adversary [40], [41].

1.3.3. Reverse Engineering Attack

RE of an IP core is a process of identifying its design, structure and functionality. Using RE one can identify the device technology, extract the gate-level netlist, and infer the IP functionality. Though according to the Semiconductor Chip Protection Act of 1984 (SCPA) RE is not illegal for teaching, analysis and evaluation purposes. However, an attacker can illegally use RE process for IP piracy, insertion of malicious logic etc. Since the modern design supply chain involves offshore design houses, hence they cannot be completely trustworthy. An adversary in these offshore design houses may perform the alteration of original register transfer level description or reverse-engineering the design in order to implant malicious logic into it. Therefore, the robust security against RE threat is amenable for ensuring the trust in data-intensive IPs before their integration into SoC systems, thereby ensuring the end consumer security against security hazards [25], [53], [57].

1.3.4. Infringing IP rights of Buyer and Seller

In the design chain of an IP core, two entities are involved, viz., seller and buyer. An IP seller also known as IP vendor is the creator of an IP, whereas an IP buyer also known as IP user is the purchaser of an IP. In the supply chain from the buyer's standpoint, an untrustworthy IP seller may distribute/sell illegal copies of custom IP (designed based on the IP buyer specification). This may lead to the illegal use of IPs. It must be prohibited in case if some hardware accelerator is designed for some specific purpose (mission-critical applications) corresponding to a specific IP buyer. Further, from seller's standpoint, a deceitful IP buyer may falsely claim the IP ownership rights, post

receiving the IP. Therefore, a unique one-to-one mapping between both the entities is amenable. And, a secured IP core should facilitate detection of unlawfully redistributed/resold duplicates of an IP core by a deceitful IP seller as well as protect the design in case if IP buyer falsely claims the IP ownership [79], [80].

1.4. Background on high level synthesis and its importance in designing secured and low-cost reusable hardware IPs

In the IC design cycle, synthesis process is one of the crucial steps. The synthesis process generically refers to the build-off or transforming the design from its one form to another for analysis and verification. Further, owing to higher design complexity, design cost and time constraints, it is crucial from the designer's perspective to begin with lesser complex and more flexible level of design. However, an IP designer may choose to perform design synthesis at different levels of design abstraction, depending upon the level of information that is required to analyze and represent. Depending upon the design transformation between different abstraction levels, the synthesis process is categorized as (a) high-level synthesis (b) logic synthesis (c) physical synthesis (corresponding from top-level to lower-level design, respectively). Among the other (or lower) levels of design abstraction corresponding to the synthesis process, HLS offers a designer with more flexibility while having lesser complexity [83], [87]. High-level synthesis transforms the behavioral description (mathematical equation representing the input-output relationship of the underlying functional data-intensive algorithm) of the design into register transfer level design. In order to do so, HLS process assimilates through different phase of it. An overview of different design phases of HLS, is shown in Fig. 1.3. HLS comprises of different phases like: the transformation phase, scheduling phase, binding phase and at the end, datapath and controller synthesis phase. In the transformation phase, it transforms the mathematical or behavioral description of the design in the form of data flow graph. A data flow graph is a structural representation of the design (algorithm), representing the input-output of the design and the flow of the information. The sample DFG corresponding to transfer function of FIR digital filter design (as shown in eqn. 1.2) is shown in

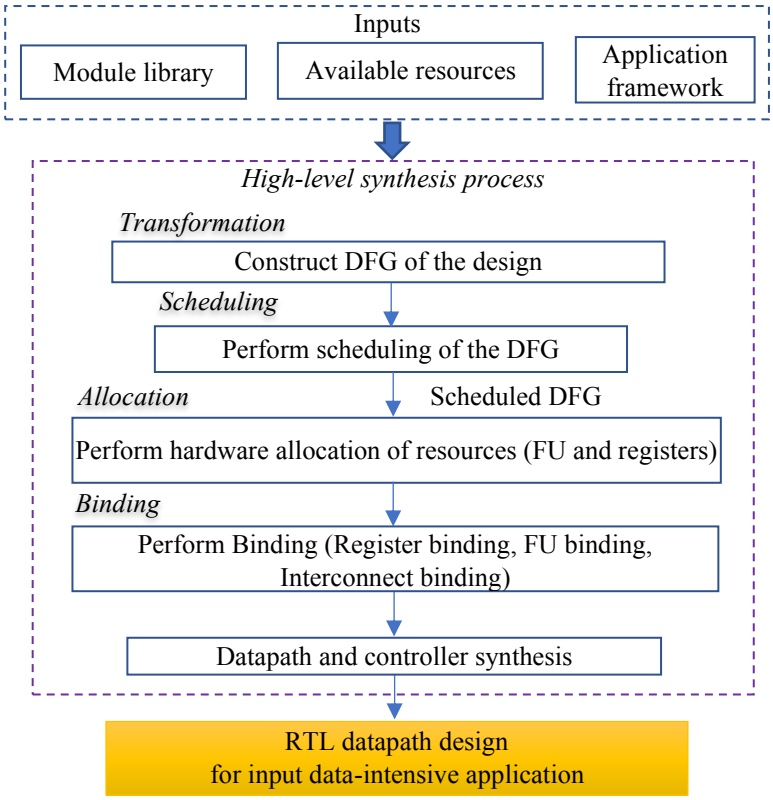


Fig. 1.3 HLS design flow overview

Fig 1.4(a), where, $\boldsymbol{A}[n]$, **A[*n*-1]**, **A[*n*-2]** and **A[*n*-3]** represent the inputs and $c[0], c[1] \ldots c[n-1]$ represent input coefficient. And $\boldsymbol{B}[n]$ represents the output and multiplication and addition operations are represented using '*'and '+' respectively. Next, is the scheduling phase. This is the crucial phase of HLS. This phase is responsible for transforming the DFG of the corresponding application into a scheduled DFG design. In order to do so, it accepts the DFG of the input application along with designer-selected resource constraints and scheduling algorithm. The scheduled DFG of FIR is shown in Fig. 1.4(b). and Fig 1.4(c). In Fig. 1.4(b) depicts the scheduled FIR design based on resource constraints one multiplier (*) and one adder (+). Further, Fig. 1.4(b) depicts the scheduled FIR design based on resource constraints two multipliers (*) and one adder (+). In order to schedule the DFG of FIR (shown in Fig. 1.4(a)), LIST scheduling algorithm has been used. LIST scheduling is a resource constraints-based algorithm. It works by trying to schedule a maximum number of operations in a control step, subject to resource constraints and data dependency. The basic idea of LIST scheduling is that it maintains a priority list of ready nodes (operations). Priority operations are those that do not depend on other operations for their execution. Further, during each iteration,

it tries to use up all resources in that state by scheduling operations in the list. However, in case of conflicts, the operator with higher priority will be scheduled first. Thus, the scheduled DFG design is obtained. However, it should be noted that the scheduled design version in Fig. 1.4(b) takes more control steps or delay (six, CS0-CS5) than the scheduled design version shown in Fig. 1.4(c). This is because of scheduling the design using a different number of resource constraints e.g., one (*) and one (+) than two (*) and one (+). However, more resources sometimes may lead to more design area. Therefore, it is important from the designer's perspective to choose such resource constraints for scheduling the design that offer lesser design latency as well as lower design area. The next phase is the hardware allocation phase. In this phase hardware resources (adder, multiplier etc.) are allocated to the operations to be executed and to the registers are assigned to the storage variables (used for accommodating the input, output and intermediate results) of the design from the HLS library. However, the allocation of the resources is based on the latency, power and area constraints of the design. As discussed earlier, more hardware resources may lead to area overhead but results shorter delay due to the parallel execution of multiple operations. On the other hand, minimum hardware resources result lesser design area but may lead to more design latency due to the serial execution of operations. Subsequently, the next phase is the binding phase. Post allocating the hardware resources to the design operations, the binding phase is performed, which decides which operation is to be associated with which instance of the respective functional unit (FU) and which variable is to which register. Fig. 1.5(a) and Fig. 1.5(b) show the allocated and binded DFG of the FIR IP core based on different resource constraints, where the storage variables of the design are represented as V_1 to V_{15} and the required registers are represented through different colors (eight registers are designated using eight diffident colors). M^1 is a multiplier resource and A^1 is an adder resource for the design version shown in Fig. 1.5(a) and M^1 and M^2 are two multiplier resources and A^1 is an adder resource for the design version shown in Fig. 1.5(b). Post scheduling, allocation and binding phases, datapath and controller synthesis phase of the HLS process is performed. This phase synthesizes the RT datapath of the design using the allocated FU resources, registers, and latches and using the Muxes and

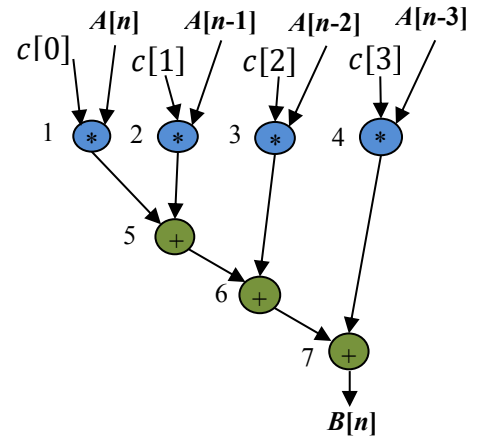


Fig. 1.4(a) DFG of FIR digital filter

1multiplier and 1 adder

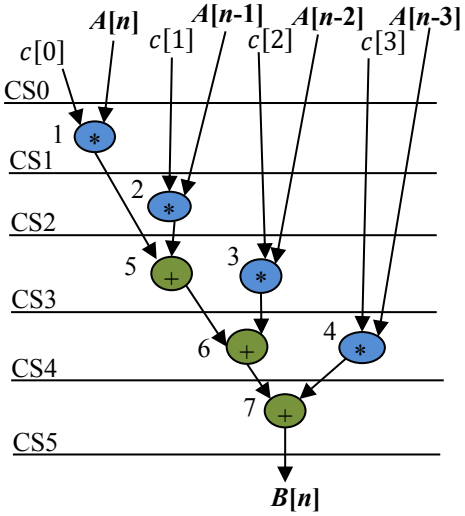


Fig. 1.4(b) Scheduled FIR filter design using resource constraints of 1(×) and 1 (+)

2 multiplier and 1 adder

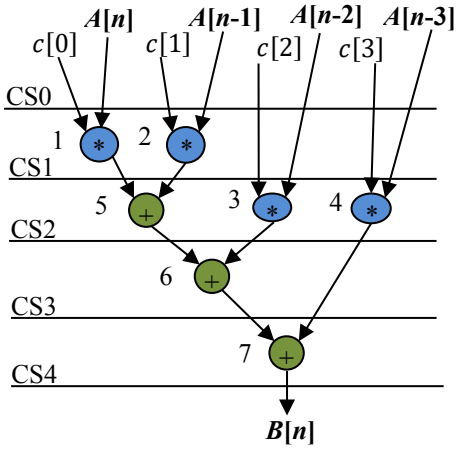


Fig. 1.4(c) Scheduled FIR filter design using resource constraints of 2(×) and 1 (+)

Fig. 1.4 Scheduled DFG of FIR based on different resource constraint

Demuxes (determined through the binding phase). Further, the controller is designed based on the scheduling and dependency information of the operations. The controller enables the control signals for different units of the datapath in the respective control steps (as per the scheduling). Thus, by using HLS, behavioral description of the input data-intensive design/application is transformed into RT-level design (also called as soft IP core). Subsequently, post obtaining the RT-level design of a sample application, it is transformed into lower design abstraction level such as logic synthesis to obtain the corresponding gate-level or netlist-level design. Gate-level design represents more complex circuitry than RT level. Subsequently, at the next design abstraction level, gate level design is transformed into a respective layout design using physical synthesis process. Post obtaining the design layout, it is

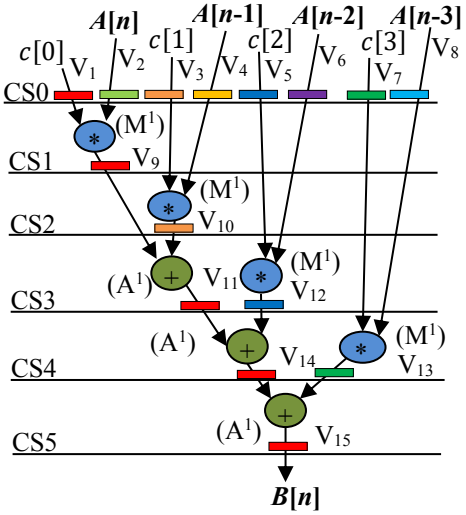


Fig. 1.5(a) Scheduled and hardware allocated FIR filter design using resource constraints of 1(×) and 1 (+)

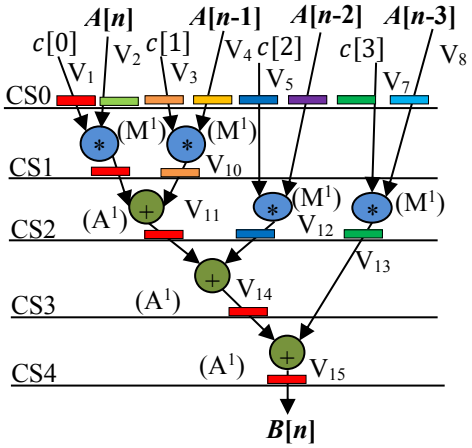


Fig. 1.5(b) Scheduled and hardware allocated FIR filter design using resource constraints of 2(×) and 1 (+)

sent to foundry house(s) for chip fabrication. It is not preferable to design an IP from lower design abstraction level of IC design process due to higher design complexity. Further, the higher design abstraction levels offer more flexibility than the lower design phases. Hence to design data-intensive co-processors (IP cores), they are meant to be synthesized from a higher abstraction level into a hardware form using high-level synthesis (HLS) framework of VLSI design process. This is because designing at lower level of abstraction such as RT-level or gate-level design involves complex design structure and huge design time, which does not remain pragmatic from a designer's perspective. The lower the design level, lower is the flexibility and harder or complex is the design process. Therefore, it is preferable to synthesize the data-intensive design from higher design abstraction level. Further, in case of other kinds of IP cores such as memory controllers, CPU, I/O module, DMA etc. are directly designed at the register transfer level (RTL) from their specifications, and hence are not targeted using the HLS-based approach.

Importance of HLS in IP core security: The IP core security for data-intensive applications during the HLS has paramount importance. This is because applying a security mechanism should not result any change in actual functionality and also should not lead to excessive area, power or delay overhead. HLS can be exploited for providing the security to the hardware IP core in terms of enabling preventive control and detective control security measures against threats. For enabling the detective pirated IP versions, different phases of HLS can be exploited for integrating the security. In order

to do so, hardware watermarking, stego-constraints and digital signature-based hardware security approaches were presented. These approaches implant the secret hardware security constraints into the design during HLS for enabling the detective control against pirated IP cores. This therefore ensure the integration of only genuine IP versions into the SoC systems. Further, for enabling preventive control during HLS, an algorithmic description of the application in the form of DFG can be transformed using different high-level transformations. These transformations obscure/obfuscate the design structure without affecting its actual functionality. Therefore, this process is also called as structural obfuscation. Through structural obfuscation, the design structure is made unobvious or hard to interpret in terms of functionality/interconnectivity for an adversary, thereby thwart reverse engineering. Various kinds of high-level transformations such as loop unrolling (LU), tree height transformation (THT) and redundant operation elimination (ROE) etc. can be applied depending on the feasibility of the application. High-level transformations enable the security against potential threat from an adversary attempting to perform RTL alteration and implant malicious logic in the safe places (not easily detectable) of the design. This hinders an adversary in reverse engineering the design by identifying its design functionality and hardware architectural details.

Moreover, integration of security mechanisms into an IP design at lower abstraction levels is arduous due to their higher design complexity. Additionally, most of time IPs are not available at the lower levels (such as gate level netlist). On the contrary, the DSP and multimedia applications are readily available in the form of their algorithmic descriptions. Additionally, they can easily be automated using commercial or non-commercial tools to generate the corresponding RTL counterparts using HLS [85]. This, therefore, enables the integration of security mechanisms with the computer-aided design (CAD) tools of HLS to generate the secured IP versions for data-intensive applications.

Importance of HLS in obtaining Low-cost IP core design [91]: As discussed earlier, different resource constraints offer different design latency and, therefore may lead to different design area. Therefore, from designer's

perspective resource selection for scheduling the design is quite crucial. In order to select optimal resource constraints, design space exploration (DSE) process can be integrated during HLS. This offers the flexibility of exploring a low-cost architectural solution that satisfies the given area and latency constraints. Further, in case of embedding the secret hardware security constraints into the design may lead to design cost overhead. Thus, the exploration of low-cost resource constraints is crucial for generating low-cost secured IP versions, which is achieved during HLS. Furthermore, employing security during HLS propagates the security at lower levels of the design. Therefore, security is ensured at the levels of firm IPs and hard IPs also. This is because the security constraints get distributed throughout the design as the subsequent level of the synthesis process is performed.

1.5. Organization of Thesis

The chapters of the thesis are organized as follows. Chapter 2 discusses the state-of-art techniques with respect to the proposed work. Chapter 3 discusses the proposed Contact-less palmprint biometric for securing DSP coprocessors used in CE systems against IP piracy. Chapter 4 discusses the proposed double line of defense approach for securing DSP IP cores using structural obfuscation and chromosomal Deoxyribonucleic acid (DNA) impression. Chapter 5 discusses the proposed methodology for designing a secured reusable convolutional IP core in CNN against piracy using facial biometric-based hardware security. Chapter 6 discusses the proposed retinal biometric approach for designing secured JPEG-codec hardware IP core for CE systems using HLS. Chapter 7 discusses the proposed methodology for performing the exploration of security-cost tradeoff for signature-driven security algorithms for optimal architecture of data-intensive hardware IPs. Chapter 8 discusses the proposed methodology for symmetrical protection of ownership right for IP buyer and seller using facial biometric pairing. Chapter 9 discusses the experimental results of the proposed techniques and compares with the state-of-the-art. Chapter 10 concludes the thesis and briefly discusses the scope for future work.

Chapter 2

State of the Art

Some hardware security techniques were proposed to counter the threats against IP core during the IC design process, for the past few years. This chapter discusses the state-of-the-art techniques along with their limitations. This therefore builds up the basis for the proposed hardware security methodologies for data-intensive IPs presented in this thesis. The first section presents the state-of-the-art on handling IP piracy threat and fraudulent ownership of IP core. The second section presents the state-of-the-art on thwarting reverse engineering attack on data-intensive IP cores. The third section presents the state-of-the-art on handling infringement of IP core buyer's and seller's right. The fourth section describes the objective of this thesis. The fifth section highlights the contributions of this thesis.

2.1. State of the Art on Handling IP Piracy and fraudulent claim of ownership Threat

The integration of pirated IP versions into SoC designs may lead to following consequences: (i) can cause security hazards to end consumer (ii) malfunctioning of the system as they might contain secret malicious logic (hardware Trojans) hidden inside. These Trojan-infected IPs or ICs are unreliable and unsafe for end consumers when integrated into CE systems (iii) may lead to security hazards by causing the malfunctioning of the device used in critical applications such as medical diagnosis, aerospace and military-based (iv) may cause revenue loss of the IP creator/designer/owner and. Therefore, the detection and isolation of pirated IP version is crucial. The IP piracy threat has been discussed in section 1.3.1 of chapter 1. The threat of IP piracy has been combated using detective control mechanisms in the literature.

2.1.1. Detective control mechanisms: To provide detective control against pirated IP versions, before their integration into CE and computing systems, several security mechanisms were proposed. The security mechanisms can be classified based on the security integration in different design levels, viz., higher abstraction level and lower abstraction level. At the higher abstraction

level, the security methodologies are: hardware watermarking, hardware steganography, digital signature and hardware biometrics based. Koushanfar *et al*. [31] presented watermarking approach based on binary encoding scheme to implant watermarking constraints for intellectual property protection. In this approach firstly, the vendor's signature is transformed into watermarking constraints based on binary (0 and 1) encoding. These constraints are subsequently added into the design (in the form of additional edges into the color interval graph). The added edges represent the watermark of the vendor. Sengupta and Bhadauria [32], presented hardware IP protection by inserting watermark in higher abstraction phase of HLS, which is based on the encoding of multi-variable signature. Multi-variable based signature encoding offers better robustness due to complex encoding process of four watermarking variables which results into more watermarking constraints for embedding into the design. Further, it generates a low-cost solution using particle swarm optimization (PSO) driven exploration process. PSO explores a trade-off between latency and area overhead achieved during watermarking and yields an optimal low-cost solution. These security constraints, post embedding, enables the piracy detection. Hong and Potkonjak presented IP protection mechanism using watermarking technique [33]. In this technique, the encoded vendor's secret mark or signature in the form of set of design and timing constraints is implanted into the IP core during behavioral synthesis. In this approach to detect and isolate pirated IP versions, the presence of the vendor's watermark is detected. Gal and Bossuet [34] presented an IP watermarking approach that uses mathematical relationships between numeric values as inputs and outputs at specified time. The inserted watermark protects the sellers' right while satisfying the user constraints in terms of design latency and area. R. Karmakar and S. Chattopadhyay [35] presented hardware IP protection methodology using logic encryption and watermarking. In this methodology, authors exploited the vulnerabilities of contemporary logic encryption mechanisms and how cellular automata can be employed for watermarking a finite state machine design. Sengupta *et al.* [36] presented triple phase watermarking-based hardware security approach for protecting IP core during a higher abstraction design level. This approach presented a multi-variable (seven) signature encoding approach for protecting the IP against

piracy and illegal ownership claim. In order to do so, vendor signature (comprising of 7 watermarking variables) was embedded into the DSP design during three independent phases of HLS process. In this approach signature variables were implanted during the scheduling phase, hardware allocation phase and register allocation phase. Roy and Sengupta [93] presented a multi-level watermarking approach for securing DSP IP cores against piracy. In order to secure the design, generated hardware security constraints corresponding to vendor's signature are implanted during different design abstraction levels such as high level and RT-Level. This approach firstly, accepts the DFG of the sample DSP application and performs sub-processes such as scheduling based on resource configuration, hardware allocation and binding. Subsequently, the RTL design is obtained using HLS framework (comprising of muxes, demuxes and registers). Next, based on the vendor's watermark signature is decoded to obtain the watermarking constraints. Finally, these constraints are embedded by diluting the muxes and demuxes into next hierarchy level and encoding the sharing of registers. Thus, the multilevel watermarking-based RTL design is subsequently constructed.

Further, Sengupta and Rathor [37] presented hardware steganography-based security approach for detecting the pirated DSP IP versions before being integrated into CE systems. In this approach, concealed stego-mark are implanted into the DSP design without using any external signature. Further, the amount of concealed digital evidence which is meant for embedding is fully under control of designer through a 'thresholding' parameter. In order to generate secured IP version, firstly it accepts the DFG of the design and transforms it into scheduled design version. Next, its corresponding CIG is constructed and edge set is determined for inserting into the generate CIG. Subsequently, swapping pairs for each edge are determined. Next, the maximum entropy for all edges is determined. Next, based on the designer selected threshold valve, subset of the edges is chosen. Finally, these edges are added into the CIG of the design. Thus, it generates the stego-constraints implanted secured design.

Rathor and Sengupta [38] presented hardware steganography using key-driven hash-chaining for securing such IP cores integrated into CE systems. In this

technique, secret imperceptible stego-marks are generated by performing hash-chaining process that incorporates switches, strong large stego-keys, mapping rules and hash blocks. This intricated methodology for stego-mark generation using steganography approach makes it sturdier than watermarking.

Sengupta *et al.* [39] presented digital signature-based approach for securing DSP IP cores against piracy. In order to secure the reusable IP core, encrypted-hash based digital signature approach takes DFG of the DSP application (in which the digital signature is to be embedded) and user specified resource constraints as primary input and based on which scheduling of DFG of the DSP core is performed. Subsequently, SDFG is fed as input to the phase-1 encoding based on which bitstream is generated (using the encoding rule-1). Subsequently, the generated bitstream is fed into SHA-512, which generates the bitstream digest of the corresponding DSP application as its output. The generation of bitstream digest involves word (W) computation process which employs the following functions: circular right shift of the 64-bit argument, left shift of the 64-bit argument and addition modulo 2^{64}. Next, in the post-processing, the generated binarized bitstream is bifurcated into desired blocks of equal size and has been converted into its equivalent decimal value. Subsequently, in the next phase encryption of each decimal value is performed using private key of the user (IP owner) through RSA encryption. Subsequently, encrypted data output is converted into binary bitstream during post-processing. Thereafter, the encrypted bitstream is fed as input to the encoding phase-2 in order to generate covert security constraints corresponding to the digital signature strength (chosen by IP designer by considering the security and design cost trade-off). Subsequently, the covert security constraints are implanted during register allocation phase of HLS process. Thus, the digital signature embedded secured reusable DSP IP core is obtained.

Further, Sengupta and Rathor [40] presented biometric-based hardware security methodology to secure the IP cores in terms of enabling the detective control against their pirated versions. In this methodology, fingerprint biometric of an IP vendor was exploited to generate the biometric digital template. Subsequently, the generated signature post-encoding was embedded

into the design. The embedded fingerprint biometric signature therefore provides detective control against pirated IP version before there integration into CE and computing systems. In order to generate secured IP designs using fingerprint biometric, the following process was performed: The biometric process is executed during high level synthesis phase rather than the lower phases of IP designing process, to minimize the implementation complexity. Steps towards securing hardware accelerators with biometric fingerprinting are: (a) capturing the fingerprint of IP vendor using optical scanner device (b) subsequently, pre-processing of the captured image is performed, which includes three sub-processes of (i) image enhancement using fast Fourier transform (FFT) which operates on the sets of pixels thereby magnifying and reconnecting the broken ridges (ii) binarization, represents the image with only two intensity level ('0'→low, '255'→high) by comparing with threshold intensity of pixels (iii) thinning, it reduces the thickness of the ridge lines to one-pixel width. Post-pre-processing, the thinned image of the fingerprint is operated to extract the minutiae points (points where ridge lines end abruptly or bifurcate into branches), which leverages the unique features an IP vendor (c) next, minutiae points are represented in its corresponding binary form. The signature corresponding to each minutiae point consists of the following: coordinates, crossing number value of minutiae type and ridge angle in degree. Subsequently, a final digital template is obtained by concatenating the signatures of each minutiae point. Next, digital template is converted into covert hardware security constraints depending on the encoding rule defined by the IP vendor. Subsequently, derived hardware security constraints are implanted into hardware accelerator design during the register binding phase of electronic system level (ESL) synthesis. Finally, the register transfer level (RTL) data path of biometric fingerprinting implanted- secured hardware accelerator is obtained.

Limitations: In hardware watermarking [31]-[36], the generated signature depends on factors such as number of signature variables, their combination, signature length and encoding rules. The dependency of watermarking approach on such intermediate factors renders it vulnerable, as they could be easily compromised. Further, hardware steganography approaches [37], [38]

are signature-free techniques to secure hardware IP core. Steganography results stronger security with lesser design overhead (for embedding generated stego constraints) than watermarking. Nevertheless, by exploiting secret stego-keys, stego-encoder and mapping rules, it is possible for an adversary to compromise the purpose of steganography. Further, both crypto-digital signature [39] and hardware steganography approaches [37], [38] contain encryption keys which are prone to key-based threats such as side-channel attacks. Overall, the major weakness of the aforementioned approaches [31]-[38] is that an adversary can replicate and regenerate the signature by compromising the limited number of security variables such as private key, encoding algorithm and signature combination. Therefore, these approaches do not ensure effective security of hardware IP cores against piracy.

Further, in crypto-digital signature approach [39], the generated digital signature is obtained through encoding, secure hashing algorithm (SHA-512) and RSA encryption using vendor's private key of size 1024-bit. Further, digital signature approach involves complex computation during signature generation for hindering an adversary from regenerating the digital signature. Nevertheless, its dependency on standard SHA-512 and private key only renders it vulnerable to compromise. In these hardware security approaches, if the chosen signature length, signature digit and their encodings into security constraints are compromised by the adversary, then he/she can reproduce the original vendor's security mark to evade piracy detection.

On the other hand, in the biometric-based approach [40], the generation of an accurate fingerprint signature involves image enhancement phase using the Fast Fourier Transform (that increases the complexity of the approach) and requires use of an optical scanner. Further, it is injury-prone and external factors may affect the accurate fingerprint generation; while in [41] the facial biometric approach incorporates naturally unique facial features for facial signature generation. However, several factors like aging and injury may affect authentication and verification. In both of these biometric approaches [40], [41], biometric features are exposed to the external environment.

2.2. State of the Art on Handling Reverse Engineering Threat

Security against reverse engineering threat is crucial for hindering an adversary to alter the RTL description of the design. This is because in untrustworthy design houses may attempt to perform reverse engineering the design in order to implant malicious logic [54]. In order to do so, an adversary by performing reverse engineering, exploits the design structure and tries to attempt actual functionality of the design. This therefore results into identifying the safe places (not easily detectable) for successfully inserting malicious logic. The inserted malicious logic therefore may cause security and integrity hazards to end consumer. Therefore, from the IP designer's perspective, it is crucial to integrate security mechanisms against an adversary attempting to perform RTL alteration [94]. In order to do so, a structural obfuscation mechanism was proposed to obfuscate (obscure) the design architecture (without affecting its actual functionality). This makes the design un-obvious to an attacker, thereby hindering possible Trojan insertion in an untrustworthy house [53], [92]. In general, the potential places for Trojan insertion could be a SoC design house or a foundry.

Sengupta *et al.* [53] employed compiler driven high-level transformations (HLTs) to architecturally transform DSP hardware. In this approach, authors exploited redundant operation elimination (ROE), logic transformation (LT), tree height transformation (THT), loop unrolling and loop invariant code motion-based architectural transformation [53], [57]. ROE mechanism eliminates the duplicate operational node from data flow graph of the design whose inputs and operation type match with other nodes. While, logic transformation modifies some operation types in the DFG without affecting the actual design functionality. THT mechanism attempt to perform some operations in parallel rather than sequential execution while keeping the functionality intact. However, sometimes it may also increase the tree height depending on the tree structure. On the other hand, in a loop unrolling-based transformation mechanism, loop body is being unrolled depending on the unrolling factor. The more the unrolling factor, the more the parallelism by enabling the reusability of FUs. This, therefore, offers a reduction in design latency (through parallelism). Loop invariant code motion mechanism shifts those operations out of the loop body, which are independent of the loop

iterations. The following compiler driven transformation mechanisms therefore renders significant transformation in the CDFG of DSP application without affecting actual functionality. The transformation at DFG level results into a transformed design (unobvious to an attacker) at RTL level post HLS. The same can be observed by analyzing the size and number of Muxes and Demuxes, changes in the interconnectivity of functional units with Muxes, Demuxes and changes in the number of storage elements etc. Further [53], integrated PSO-DSE framework with the HLS process. The PSO-DSE enables the generation of low-cost architectural solution which in turn leads to minimal design cost of architecturally transformed design. Furthermore, Sengupta *et al.* [55] proposed a methodology to generate a secure JPEG-codec hardware accelerator design using THT-based structural transformation. And, Sengupta *et al*. [56] proposed a methodology for providing the security of fault-secured DSP designs against reverse engineering threats through multi-phase transformations. Further, Lao and Parhi [92] presented preventive control mechanism by obfuscating DSP circuits through high-level transformations. In this approach, authors utilized hierarchical contiguous folding (HCF) for performing the architectural transformation. In this folding, all operations are performed sequentially in stages. More explicitly, Lao and Parhi [92] applied the transformation by varying the number of stages in the cascaded architecture, resulting into several variation modes. For obfuscating DSP circuits, different variation modes can be implemented to produce different outputs (meaningful and non-meaningful modes). The output of folding is exploited for performing the transformation in this approach. Configure data is used for regulating various modes of operations. A re-configurator enables the configuration of the functional mode of a DSP design through a finite state machine (FSM). Further, this FSM is controlled by a secret key. Therefore, while applying invalid key/wrong configure data, it results in either a meaningful but non-functional or non-meaningful mode. Thus, folding-based transformation results in many equivalent DSP circuits incurring obscurity in the structure. This approach mainly targets loop-based DSP applications (such as finite impulse response filters etc.) for transforming the design structure in order to hinder RE attacks.

Limitations: These hardware security methodologies presented in [53], [55]-[57], [92] are capable to be applied on particular applications. This is because the compiler-driven high-level transformations employed in existing approaches may not be directly applicable to all the different applications. Further, these approaches provide only single line of defense against reverse engineering. These security mechanisms do not integrate security measures against piracy of the target hardware designs. This therefore, demands alternative techniques which can be applied to wide range of applications and should be capable of handling reverse engineering threat along with piracy.

2.3. State of the Art on Handling Symmetrical IP Core protection

To protect IP rights of both the entities, symmetrical protection of DSP IP cores is necessary which will preserve the user right as well as invalidate the ownership abuse. Implanting buyer's signature and seller's signature into an IP core design can provide symmetrical IP core protection.

There are two approaches [79], [80] in the literature that provided symmetrical IP core protection techniques. In [79], a hidden encrypted mark is embedded into the physical layout of a digital circuit when it is placed and routed onto the FPGA. This mark not only uniquely identifies the source of the circuit but also detect the original recipient of the circuit. In [80] symmetrical IP core protection using multi-variable fingerprint encoding and hardware watermarking was presented. In this approach, along an IP seller inserting his own watermark, the multi-variable fingerprint of IP buyer is also inserted into the design using high level synthesis (HLS) to enable symmetrical security.

Limitations: The approach [79] provides protection for both entities in the lower design abstraction level, i.e., layout level, which is impractical for complex DSP IP cores. Moreover, no design optimization algorithm is used to minimize the design overhead due to the insertion of secret marks. Further, the approach [80] is not as robust as the proposed symmetrical security using facial biometrics. This is because the facial biometric-based security methodology embeds the naturally unique facial signature (yields larger security constraints than the other contemporary approaches) of IP buyer and IP seller.

2.4. Objective of the Thesis

The objective of the thesis is to develop novel hardware security methodologies/techniques for ensuring the security of data-intensive IP cores based on DSP, multimedia and machine learning applications against the foregoing hardware threats. This is achieved by setting out the following goals and objectives:

1. To develop biometric-based hardware security methodology for enabling the robust and seamless detection of pirated DSP coprocessors used in CE systems using contact-less palmprint biometrics.
2. To develop double line of defense mechanism using structural obfuscation and chromosomal DNA impression for securing DSP IP cores against the hardware threats of reverse engineering and piracy.
3. To develop secured custom reusable convolutional IP core in CNN using facial biometric approach against piracy.
4. To develop HLS-based secured JPEG-codec hardware IP core using retinal biometric-based hardware security methodology.
5. To develop a methodology for performing the exploration of security-cost tradeoffs for signature-driven security algorithms for optimal architecture of data-intensive hardware IPs.
6. To develop a methodology for ensuring the protection of IP rights of IP buyer and seller using facial biometric pairing.

2.5. Summary of the Contributions

- A novel approach for piracy detective control of IP cores used in CE systems using the proposed contact-less palmprint biometric approach. (Publications: #1, #6, #16)
  - Proposes a novel 'contact-less palmprint biometric' based hardware security approach for enabling robust and seamless detection of pirated IP versions of DSP coprocessors before being used in CE systems.
  - Exploits the naturally unique palm features of an IP vendor to generate biometric-based covert hardware security constraints. These hardware security constraints post embedding into the design enable the detective control against pirated IP version.

- Achieves higher security strength against piracy in terms of lower probability of coincidence (indicating stronger digital proof of evidence against fake IP cores) and higher tamper tolerance (indicating stronger defense against the regeneration of embedded secret signature by an adversary) at negligible design cost overhead.

- A novel double line of defense methodology for securing DSP IP cores using proposed structural obfuscation and chromosomal DNA impression. (Publications: #5, #10, #17)
  - Proposes a novel hybrid methodology to secure intellectual property (IP) cores of digital signal processing (DSP) applications against the hardware threats of reverse engineering and piracy.
  - The proposed approach exploits multilevel structural obfuscation as 1st line of defense against alteration of register transfer level (RTL) description of IP core design.
  - The proposed approach covertly implants an invisible DNA impression into the structurally obfuscated DSP design using robust encoding and encryption using multi-iteration Feistel cipher as a 2nd line of defense against IP piracy.
  - Our technique is more robust than other contemporary hardware IP security techniques in terms of yielding very low probability of coincidence (Pc) (indicating strength of digital evidence) and stronger tamper tolerance for different DSP IP cores.
  - Incurs zero design cost overhead post implanting encrypted DNA impression and post-structural obfuscation. Further, it also ensures higher strength of obfuscation in terms of number of gates obfuscated.
- A novel HLS based methodology of designing secured custom reusable convolutional IP core in CNN using facial biometric-based hardware security. (Publications: #2, #13)
  - The proposed work leverages the HLS-based methodology for designing custom reusable convolutional IP core designs.
  - Presents a methodology for securing CNN IP cores using a facial biometric signature that has a robust capability to differentiate between fake/pirated and authentic versions. This ensures the integration of only

genuine CNN IPs in computing and CE products for security of the end consumer and protecting brand value of the original vendor.

- Exploits the naturally unique facial features of an IP vendor to generate biometric-based covert hardware security constraints. These hardware security constraints are responsible for enabling security in terms of detective control against the integration of pirated convolutional IPs into computing systems.
- Yields zero design cost overhead for embedding the facial biometric of IP vendor into the convolutional IP design against piracy.

- A novel hardware security methodology for designing secure JPEG compression-decompression (CODEC) hardware IP using retinal biometric-based approach. (Publications: #3)
  - Proposes first work towards securing JPEG codec hardware using retinal biometric-based approach.
  - Presents HLS based design flow of generating a secured JPEG-codec hardware IP against IP piracy.
  - The proposed approach presents contact-less biometric process for securing JPEG-codec IP core using retinal image of the original IP vendor, where the encoded hardware security constraints corresponding to generated retinal signature are covertly implanted inside the design using HLS process.
  - The proposed approach is capable of offering higher robustness during the authentication/verification process due to the generation of the large number of secret security constraints and the highly distinctive nature of the retinal structure. It also enables sturdy isolation of pirated versions of IPs at zero design cost overhead.
- A novel approach for the exploration of security-design cost tradeoff for signature-based security methodologies used for detective control against intellectual property (IP) piracy/counterfeiting for digital signal processing (DSP) IP cores (publications: #15, #19)
  - Proposes an exploration methodology that offers low-cost hardware design architectural solution for secured IP cores using particle swarm optimization (PSO).

- Integrates three different hardware security methodologies such as IP facial biometrics, encrypted-hashing and IP watermarking the PSO framework for exploring the hardware architecture tradeoffs of security-design cost for different DSP applications.
- The results include the analysis of low-cost architectural resource configuration, impact of signature strength on security-design cost fitness value and, register count of the DSP IP core and security parameter such as the probability of co-incidence for various security methodologies for varying (scalable) signature strength.

- A novel approach for enabling symmetrical protection of ownership right's for IP buyer and IP seller using facial biometric pairing. (Publications: #14, #20)
  - Proposes HLS-based methodology for enabling symmetrical IP core protection using facial biometric pairing.
  - Integrates naturally unique facial biometric information of IP buyer and subsequently of IP seller during the register allocation phase of HLS.

The results include the analysis of ownership proof (probability of coincidence) and design cost overhead. The proposed methodology offers symmetrical protection of IP rights for both IP buyer and IP seller while incurring zero design cost overhead.

Chapter 3

Contact-less Palmprint Biometric for Securing DSP Coprocessors used in CE Systems against IP Piracy

For the past few decades, with the advancement of technology and innovations in the field of electronics and computing have led SoC-based consumer electronics systems such as smartphones, wearable gadgets, health bands, digital cameras, computing devices etc. These SoCs-based systems integrate DSP coprocessors for facilitating several crucial applications such as image, audio, and video processing, etc. In these DSP coprocessors in their backend, there function computationally intensive algorithms such as discrete cosine transforms (DCT), discrete wavelet transform (DWT), and finite impulse response (FIR) filters etc., for performing the aforementioned applications. Owing to the high computational and data intensiveness of these DSP algorithms, their realization as hardware co-processors or IP cores is very critical for high definition (HD), high performance and power-efficient CE devices. Therefore, DSP co-processors based intellectual property (IP) cores are rapidly thriving in the modern consumer electronics era. Additionally, the use of reusable IP cores includes new, high-growth markets, including healthcare, artificial intelligence (AI), Internet of Things (IoT) devices, automotive, wearables and smart cities and homes, etc. On the one hand, where its usage are increasing, its security is also becoming a big concern in terms of facilitating the creation of a root of trust in the hardware. This is because uneven supply-to-demand ratio, time to market, the intention of lower design cost and shorter design cycle are the major factors for enforcing to import of these IP cores from offshore design houses as their only practical solution. Therefore, their supply chain involves multiple offshore entities to provide the IP cores (soft IPs). This involvement of multiparty vendors renders the design chain susceptible to different hardware security threats. IP piracy is one of the dominating threats in the industry. However, there have been some security solutions based on watermarking, steganography, computer forensic engineering and hardware metering etc., to protect the IPs against piracy. Since these security methodologies involve auxiliary security parameters for providing the security of IP design against piracy. Therefore, the security

offered by these aforementioned approaches can be compromised, in case if the encoding mechanism of watermark and entropy threshold parameter of steganography approach gets compromised. Additionally, the number of generated security constraints using watermarking and steganography-based approaches are also comparatively lesser. This results into lower tamper tolerance (adversarial efforts in guessing the implanted signature) and higher probability of coincidence (false positive) which is not desirable. Further, these approaches do not uniquely associate the natural identity of an IP vendor, therefore they may not be prominent during the litigation, in case of ownership conflict. This entails developing a robust mechanism for enabling the seamless detection of pirated IP versions before their integration into SoCs of CE and computing systems. Palmprint biometric trait has shown promising results for user authentication [42]-[52]. However, it was not exploited for hardware authentication. In DSP and multimedia applications-based IP cores, the security in terms of detective control against piracy can be associated at higher design abstraction levels, followed by the synthesis process. In the case of DSP and multimedia applications, the high-level synthesis (HLS) process offers an efficient and less complex way of integrating the security mechanism. The details on IP piracy threats and the state-of-the-art security mechanisms have been discussed in chapters 1 and 2.

A novel technique for enabling the isolation followed the detection of pirated IP versions using contact-less palmprint biometrics has been presented in this chapter. The first section of the chapter describes the formulation of the problem. The second section discusses the proposed contact-less palmprint biometric-based hardware security technique under the following sub-sections: its importance for consumers and CE systems, utility of the approach, overview, motivation of proposed palmprint biometric approach, and palmprint biometric template generation. The third section discusses the process of generation of secured RT level design corresponding to target DSP application under the following sub-sections: the embedding process of palmprint biometric template into a DSP application by using FIR digital filter as a demonstrative example, the detection process of palmprint signature, the measure used for evaluating the security of palmprint biometric methodology.

Subsequently, the fourth section presents the metric for evaluating the impact of the proposed palmprint technique on enabled security strength and resulting design cost. Finally, the fifth section concludes the chapter.

3.1. Problem Formulation

Given algorithmic representation of DSP application, module library, resource constraint R_c, along with the objective of securing co-processor IP cores against piracy. To generate a secured IP by implanting the palmprint biometric driven naturally unique secret information of an IP vendor into the design that enables robust and seamless detective control against pirated IP versions.

3.2. Biometric digital template generation based on captured palmprint of an IP vendor

The proposed palmprint biometric-based hardware security methodology is discussed under the following sub-sections.

3.2.1. Importance for Consumers and CE Systems

Ensuring security against IP piracy/counterfeiting threat is highly important for consumers because of the following reasons [31], [37]: (i) pirated/counterfeited designs are not rigorously tested for ensuring reliability and security (ii) pirated IP versions contain secret malicious logic (hardware Trojans) hidden inside. These Trojan-infected IPs or ICs are unreliable and unsafe for end consumers when integrated into CE systems [25]. The piracy threat for DSP co-processors (IP cores) used in CE systems is addressed in this paper using the proposed palmprint biometric-based hardware security approach. The IP cores carrying authentic vendor palmprint signatures are genuine and, therefore, can be discerned and isolated from the pirated ones during the piracy detection process. This impedes the integration of fake or counterfeited IPs in the SoCs of CE systems and ensures the use of only authentic designs in CE and computing systems, thereby ensuring the security of end consumers also.

3.2.2. Utility of the approach

In case if an SoC integrator purchases IP cores (for integration) either directly from an IP vendor or else from a broker (acting as a middleman between IP

designer and the SoC integrator) [7]. A rouge IP supplier may attempt to infuse pirated or fake components (IPs) in the design supply chain for the purpose of national interest or yearning to earn illegal income. The use of fake components (pretending to be genuine) in the SoCs of CE devices can have an adverse impact on both the CE system integrator and the end user. Therefore, it is indispensable to address this hardware threat and enable the use of only authentic IPs in the CE systems. The proposed approach is useful to counter this threat as the vendor's palmprint signature (a highly authentic and unique mark) is used for authenticating the genuine IPs before their use in the SoCs. Additionally, the proposed approach is also useful in the following scenarios: (i) If a rouge IP supplier has already inserted the Trojan and selling such fake/compromised IPs to the system integrators, then the proposed approach helps in discerning such fake IPs as they would not contain the genuine vendor's authentic palmprint mark (ii) It may be useful in detecting ICs with poor specs when relabeled as ones with better specs also. Detection in this case, can be performed by reverse engineering the IC up to the intended level of design form (the RTL) to trace the authentic palmprint signature implanted in the genuine ICs. If the ICs with better specs are secured with vendor's palmprint, then by detecting the palmprint signature in the RTL form of ICs under test, the authentic ICs (designs with better specs) can be discerned from the undesired ICs (designs with poor specs).

3.2.3. Overview

The overview of the proposed hardware security approach using palmprint biometrics is shown in Fig. 3.1. As highlighted in the figure, firstly a palmprint signature of the original IP vendor is generated (from his/her palmprint biometric) using the proposed algorithm. Secondly, the generated palmprint signature is converted into encoded hardware security constraints and subsequently covertly implanted into the target DSP design through HLS framework. Here, the HLS framework first transforms the algorithmic representation (such as C/C++ code or computation function) of the target DSP application into its scheduled and hardware-allocated design based on module library and resource constraints. Next, the register allocation phase of the HLS framework is exploited to implant hardware security constraints

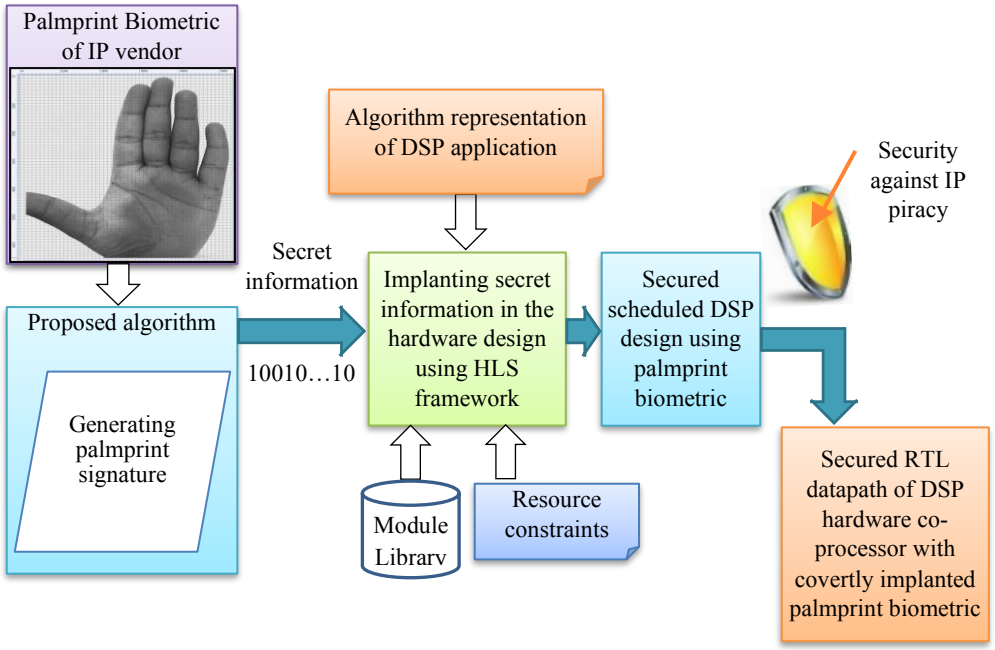


Fig. 3.1. Overview of proposed contact-less palmprint biometric based hardware security methodology

(corresponding to the secret palmprint biometric) into the scheduled design. Post HLS, a palmprint signature implanted register transfer level (RTL) design of DSP core is generated.

The proposed palmprint biometric-based approach offers security against counterfeiting threats by enabling the detection of counterfeited IP using implanted palmprint signature. The proposed methodology of hardware counterfeit detection is highly robust because: (i) the implanted palmprint signature acts as a strong, authentic secret mark since it is obtained using vendor's unique biometric information, (ii) the detection process of embedded palmprint into the design is seamless. It is noteworthy that any highly trustworthy insider in the IP vendor's firm can be selected for implanting palmprint biometric information.

3.2.4. Motivation of proposed palmprint biometric approach

The benefits offered by the proposed contact-less palmprint biometric approach are discussed in terms of the following:

a) *Injury prone*: The fingerprint biometric approach [40] is injury prone. However, in the case of the proposed palmprint biometric approach, during the validation process, recapturing the palmprint biometric information is not required. Instead, the stored palmprint image (the one used for

generating the corresponding signature and implanting into the design during the IP development process) with grid size/ spacing and nodal points are used for verification/validation purposes. This ensures that even if the insider in the IP vendor house (whose palmprint was used for embedding as a secret signature) leaves the company or unfortunately meets with an accident, it has no effect on the verification/validation process. Additionally, since a chip has a lifetime of about five years, hence the person (top-level executive or any major stakeholder in the company) selected in the vendor house for the palmprint biometric could be that one who has a bond of this period of time to work with the company.

b) *Role of external factors*: The external factors such as grease and dirt etc. don't affect the verification of IP cores using the proposed palmprint biometric approach, unlike the fingerprint biometric approach.

c) *Role of optical scanner during recapturing and verification*: The authentication of IP cores using the proposed palmprint biometric is independent of the optical scanner, unlike the biometric fingerprinting approach [40] wherein not using a good quality scanner with a similar capture area during verification would hinder the correct authentication of IP cores.

d) Contact-less authentication: The proposed palmprint biometric approach is a contact-less scheme of verifying the vendor's palmprint signature embedded into an IP core design. Therefore, the proposed approach becomes advantageous, especially in pandemic situations such as covid-19 where direct contact of external objects (scanner surfaces) is to be avoided.

e) The palmprint signature is beneficial over a random number. This is because using the palmprint signature based biometric information, the vendor's identity can uniquely be associated with his/her IPs. Hence an adversary cannot copy and misuse the genuine vendor's palmprint signature to implant it into a fake IP with the malicious intention of authenticating it as a genuine one.

f) Extracting digital templates for fingerprints is relatively more complex than palmprint biometrics. The fingerprint biometric signature generation requires pre-processing (image enhancement using FFT, binarization, and thinning) of the fingerprint image for accurate minutiae points extraction.

g) Palmprint signature, being biologically unique, is not replicable and non-vulnerable to duplication unlike digital keys/tokens (alphanumeric IDs). Further even if an attacker is able to illegally access the scanned palmprint, he/she would additionally need the following unique secret information to regenerate the palmprint signature for pirating the IP or fraudulently claim IP ownership: (a) secret naming conventions assigned to nodal points (b) covert coordinates of palm features nodal points (c) secret set of palm features chosen (d) secret sequence of concatenation of features for generating palmprint signature (e) secret constraint mapping rules. All these crucial security parameters are not known to an attacker.

h) The proposed approach provides lower probability of coincidence (indicating stronger proof of authentic digital evidence) and higher tamper tolerance (indicating higher strength of thwarting brute force attacks, ghost signature attack and unauthorized signature insertion attack) than state of the art digital signature/key based approaches [31] [37] and biometric-based approach [40].

i) Further, the proposed approach has the following advantages over state-of-the-art [37], [34]- [40]: (i) the proposed unique palmprint biometric constraints are non-replicable and non-vulnerable, unlike digital signature and stego-constraints, because of natural uniqueness of the palmprint biometric (ii) having no dependence on secret keys, the palmprint biometric constraints remain secured from key leakage, unlike steganography approaches.

3.2.5. Palmprint biometric template generation

The design flow of proposed approach for generating palmprint biometric template in order to secure DSP based hardware co-processor design has been shown in Fig. 3.2. The details of the proposed approach are discussed under following subsections:

(a) Capturing palmprint biometric with grid size and spacing

The first phase of the proposed approach accepts the palmprint of an IP vendor in order to generate its corresponding secret signature. Therefore, firstly the palmprint biometric of an IP vendor is captured with a high-quality and high-resolution digital camera. Subsequently, the captured palmprint image is

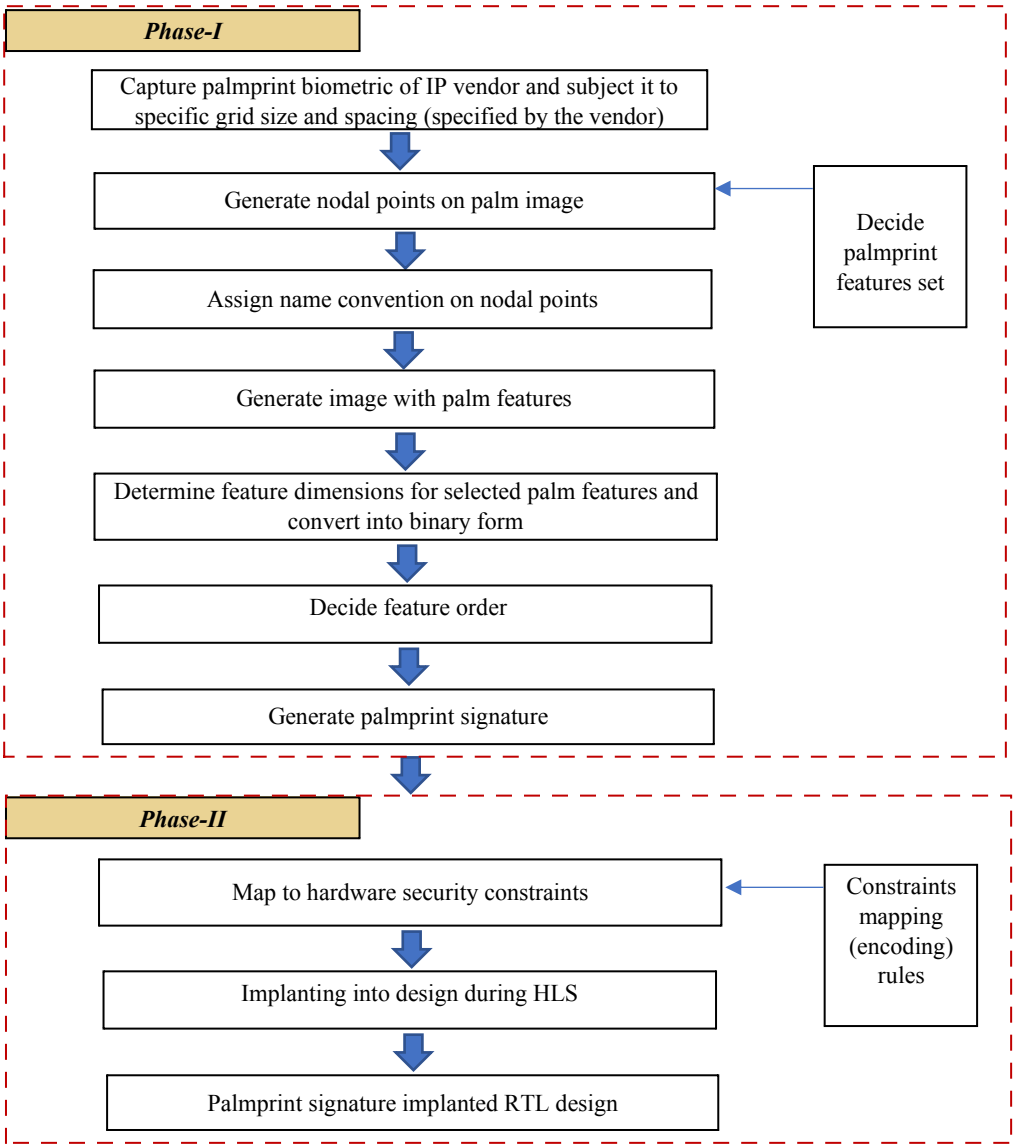


Fig.3.2. Flow of proposed palmprint biometric approach for securing DSP based co-processor designs during HLS

subjected to a specific grid size and spacing. This helps in generating precise nodal points and the coordinates of palmprint features on the palmprint image (used for palmprint signature generation). Further, this also enables the seamless verification of palmprint biometrics for hardware security, where the palmprint image with grid size and spacing would be required to reproduce palmprint features coordinates and dimensions. Fig. 3.3 depicts a sample palmprint image with a specific grid size and spacing specified by the IP vendor. This palmprint image is used to demonstrate the proposed methodology of producing a palmprint signature and implanting it into a target DSP design. Note: The capturing of palmprint (as well as the verification) in the proposed approach does not require an optical scanner because the

proposed approach is a contact-less palmprint. The captured palmprint image using a high-quality digital camera (12-megapixel camera with an f/1.8 aperture and phase detection autofocus) is capable to show the required palm features which are converted into the corresponding digital template (hash) of the palmprint signature. The stored palmprint image with grid size/ spacing and nodal points are used for verification/validation (or recognition) purposes, therefore not requiring recapturing of the palmprint image during the recognition process. This also makes the recognition process of the proposed approach independent of the different positioning of a palm image. Though different movements of a palm image may generate different hash (digital template) of the palmprint biometric, however, in the proposed approach, a pre-captured and pre-stored palm image with a specific orientation only is used to create its corresponding hash, and the same is used during the verification (recognition) process. Therefore, the recognition process is independent of any movement.

(b) Determining palmprint feature set and generating nodal points

Post subjecting the captured palmprint image to a specific grid size; nodal points are generated. These nodal points are amenable to representing the unique biometric information of an IP vendor. In order to generate nodal feature points on a palm image, firstly, the set of palmprint features are determined (used in the palmprint signature). The determined nineteen palm features are shown in Table 3.1. Further, the features listed in Table 3.1 are classified into four categories of palmprint features:

(i) Principal line feature: the feature F1 defined in Table 3.1.

(ii) Datum point feature: the feature F2.

(iii) Geometry features: the features F3 and F4.

(iv) Intersection point features: the features F5 to F19.

Every feature number is given a unique name for seamless identification/mapping process. Providing unique name to each feature enables the IP vendor to easily identify the palmprint characteristics associated with each feature number. In order to govern the size (strength) of the digital template, the number of palmprint features in the palmprint signature can also

be increased or decreased. Once the features are determined, corresponding nodal points are generated on the palmprint image. In order to generate the nodal points, the endpoints of chosen features are marked on the palmprint image. To do so, the palmprint image is scanned from *left to right* and *top to down*. Thus, generated nodal points on the palmprint image are shown in Fig. 3.3 using red color dots. As shown, there are 25 nodal points on the palmprint image. The dimensions of the selected nineteen palmprint features are computed using these nodal points.

(c) Assigning naming convention on nodal points and generating palmprint image with selected feature set

Post generating the nodal points on the palm image, their naming convention is performed. The naming convention is performed to designate each nodal point with a unique identity. A palmprint image with an assigned naming convention to the nodal points is shown in Figure 3.4. For all individual palmprint features, the corresponding naming conventions of nodal points have been shown in Table 3.1. For example, P16 and P24 are the naming conventions of two nodal points of the palm feature DL as shown in Fig. 4, and is also listed in Table 3.1. In order to highlight the selected palmprint feature on the palmprint image, the corresponding nodal points are joined together using yellow lines. The palmprint image with the IP vendor chosen features set (comprising of nineteen palm features) is shown in Fig. 3.5.

(d) Determining feature dimensions

Post generating the palmprint image with the vendor chosen feature set,

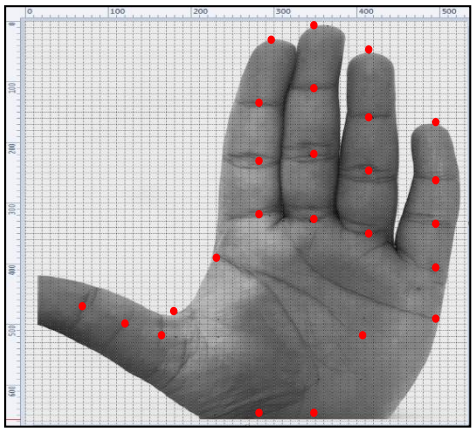


Fig.3.3. Nodal points on the sample palmprint with grid size and spacing

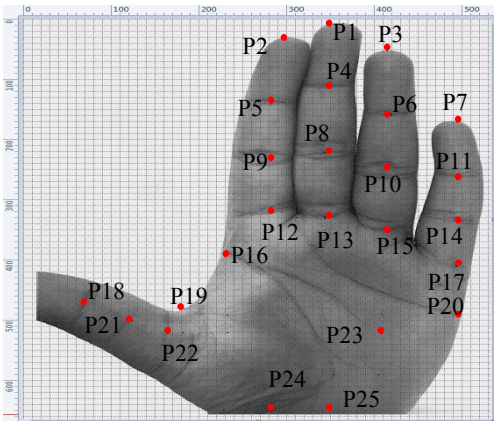


Fig.3.4. Naming convention of nodal points on the palmprint image

dimension of each feature is computed. As shown in Fig. 3.5, each feature is represented as the measure of distance between two respective nodal points. Therefore, for computing the dimension, co-ordinates (x1, y1) and (x2, y2) of two respective nodal points of each palmprint feature are obtained. The co-ordinates of each individual feature are shown in Table 3.1. Further, the dimensions of all features are computed between the coordinates of corresponding nodal points. Thus, the computed dimensions of selected palmprint features are shown in Table 3.2. It is to be noted in Fig. 3.5 that selected palm features are shown using straight and inclined yellow lines. The dimensions of straight-line features are measured between the corresponding nodal points using Manhattan distance and the dimension of inclined line features are measured using Pythagoras theorem. For example, the feature F6 has two respective nodal points as P5 and P9 whose coordinates are (285, 130) and (285, 230), respectively as shown in Table 3.1. Since the feature F6 (P5, P9) is a straight-line, as shown in Fig. 3.5, its dimension is computed using Manhattan distance. Further, the feature F2 has two respective nodal points as P23 and P24 whose coordinates are (405, 520) and (285, 650) respectively. Since the feature F2 (P23, P24) is an inclined line (as shown in Fig. 3.5), it is considered as the hypotenuse of a right-angled triangle to compute its length using the Pythagoras theorem. The values are obtained and verified using an unconstrained Cartesian coordinate system. Similarly, the dimension of other palm features is determined.

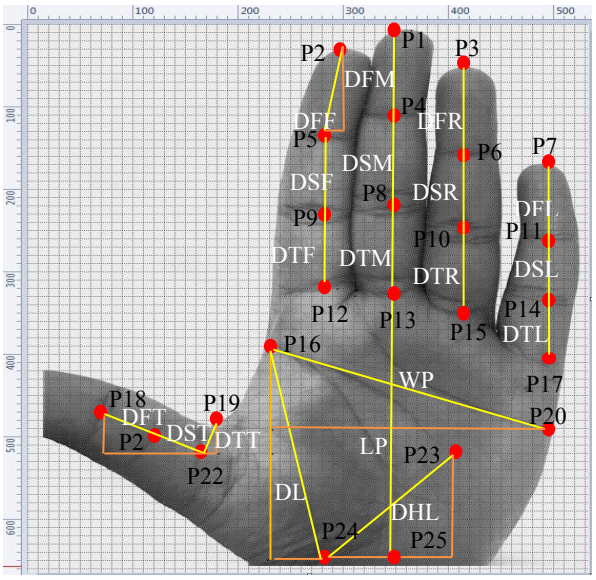


Fig.3.5. Palmprint with chosen feature set

(e) Deciding feature order and generating palmprint biometric Signature

Post determining the dimension of each feature from feature set, an IP vendor selects the order of concatenation for features, in order to generate desired palmprint template. Different possible order may lead to generate different palmprint signature. To generate the secret palmprint signature, first the order of concatenation of palmprint features is selected. Second, all feature dimensions are converted into corresponding binary equivalents (shown in Table 3.2) and then concatenated in the chosen order. Let's say the chosen order of concatenation of all selected palmprint features (nineteen) is as

Table 3.1 Selected palmprint features, corresponding nodal points and their coordinates

Feature #	Palmprint feature name	Naming conventions of nodal points	Co-ordinates (x1,y1)- (x2,y2)
F1	Distance between start of life line and end of life line (DL)	(P16) – (P24)	(230, 390)- (285, 650)
F2	Distance between datum points of head line and life line (DHL)	(P23) – (P24)	(405, 520) -(285, 650)
F3	Width of the palm (WP)	(P16) – (P20)	(230, 390)- (495, 490)
F4	Length of palm (LP)	(P13) – (P25)	(350, 325)- (350, 650)
F5	Distance between first consecutive intersection points of forefinger (DFF)	(P2) – (P5)	(300, 30)- (285, 130)
F6	Distance between second consecutive intersection points of forefinger (DSF)	(P5) – (P9)	(285, 130)- (285, 230)
F7	Distance between third consecutive intersection points of forefinger (DTF)	(P9) – (P12)	(285, 230)- (285, 320)
F8	Distance between first consecutive intersection points of middle finger (DFM)	(P1) – (P4)	(350, 5)- (350, 110)
F9	Distance between second consecutive intersection points of middle finger (DSM)	(P4) – (P8)	(350, 110)- (350, 220)
F10	Distance between third consecutive intersection points of middle finger (DTM)	(P8) – (P13)	(350, 220)- (350, 325)
F11	Distance between first consecutive intersection points of ring finger (DFR)	(P3) – (P6)	(415, 50)- (415, 160)
F12	Distance between second consecutive intersection points of ring finger (DSR)	(P6) – (P10)	(415, 160)- (415, 245)
F13	Distance between third consecutive intersection points of ring finger (DTR)	(P10) – (P15)	(415, 245)- (415, 355)
F14	Distance between first consecutive intersection points of little finger (DFL)	(P7) – (P11)	(495, 170)- (495, 265)
F15	Distance between second consecutive intersection points of little finger (DSL)	(P11) – (P14)	(495, 265)- (495, 335)
F16	Distance between third consecutive intersection points of little finger (DTL)	(P14) – (P17)	(495, 335)- (495, 405)
F17	Distance between first consecutive intersection points of thumb finger (DFT)	(P18) – (P21)	(70, 470)- (120, 495)
F18	Distance between second consecutive intersection points of thumb finger (DST)	(P21) – (P22)	(120, 495)- (165, 520)
F19	Distance between starburst point and third intersection point of thumb (DTT)	(P19) – (P22)	(180, 480) -(165, 520)

follows:

"DL╫DHL╫WP╫LP╫DFF╫DSF╫DTF╫DFM╫DSM╫DTM╫ DFR╫DSR╫ DTR╫DFL╫DSL╫DTL╫ DFT ╫ DST ╫ DTT"

Where, symbol '╫' indicates the concatenation operator. Based on the aforementioned order of palmprint features, the digital template of palmprint signature is generated by concatenating the corresponding binary equivalent of feature dimensions. Thus, obtained digital template of palmprint signature is given below:

100001001.1110110000.11101000111101011110001101 1.001111010111000010110100010111001 01.000111000010100011111100100101101011010011101111011010011101110101010111011101011111100011010001101101 11.11001100110011001111001 1.011100110011001100111010 10.101110000101000111 11

The above template of palmprint signature has size of 262 digits which include seven binary points (.). It is to be noted that, numerous possible combinations of palmprint signature of same size can be produced by using different concatenation order of same number of features. In addition, the size of digital template can be varied by selecting varying number of palm features. The scaling of palmprint signature size can be made based on the size of target DSP design and desired security strength. For example, a vendor can produce large palmprint signature (by selecting more number of palm features) using

Table 3.2 Feature dimension and corresponding binary representation of palmprint features chosen by IP vendor

Feature #	Feature name	Feature dimension	Binary representation
F1	DL	265.75	100001001.11
F2	DHL	176.91	10110000.111010001111010111
F3	WP	283.24	100011011.0011110101110000101
F4	LP	325	101000101
F5	DFF	101.11	1100101.00011100001010001111
F6	DSF	100	1100100
F7	DTF	90	1011010
F8	DFM	105	1101001
F9	DSM	110	1101110
F10	DTM	105	1101001
F11	DFR	110	1101110
F12	DSR	85	1010101
F13	DTR	110	1101110
F14	DFL	95	1011111
F15	DSL	70	1000110
F16	DTL	70	1000110
F17	DFT	55.90	110111.11100110011001100110011
F18	DST	51.45	110011.01110011001100110011
F19	DTT	42.72	101010.10111000010100011111

proposed approach to secure larger designs. Whereas, relatively lesser number of palm features can be selected to produce palmprint signature for medium size designs. Thus, based on IP vendor-specified secret security parameters such as grid size and spacing, number of palm features from feature set, feature order and final signature length, palmprint biometric digital template is generated.

3.3. Demonstration on generating palmprint embedded secured RT level design for FIR filter using HLS

So, far we discussed the process for generating the palmprint biometric signature. This generated signature is subsequently used for embedding into the design for discerning and isolating the pirated IP versions. For the sake of demonstration, FIR digital filter application has been employed for generating its corresponding secured IP design using palmprint biometrics. The details are discussed under the following sub-sections:

3.3.1. Palmprint secured RTL design generation

The details are discussed under the following sub-sections:

(a) Mapping palmprint signature into security constraints

Post obtaining a digital template of the palmprint signature, it is mapped to corresponding secret hardware security constraints based on the mapping rules. The illustration of mapping of palmprint signature bitstream into corresponding hardware security constraints is shown using finite impulse response (FIR) filter, as follows: (i) algorithmic representation of FIR filter application is transformed to corresponding data flow graph (DFG) representation (ii) scheduling of the DFG is performed based on resource constraints of 4 multipliers (M1 to M4) and 4 adders (A1 to A4), as shown in Fig. 3.6. In the scheduled DFG shown in Fig. 3.6, eight registers (named P, I, V, G, Y, O, R and B) are used to execute 31 storage variables (T0-T30), where a distinct color has been used to denote each register. The assignment of all storage variables to the registers in different control steps (C0 to C9) is shown in Table 3.3 (iii) in order to map the digits of the palmprint digital template to the hardware security constraints, a colored interval graph (CIG) framework is used. A CIG graphically shows the assignments of storage variables to the

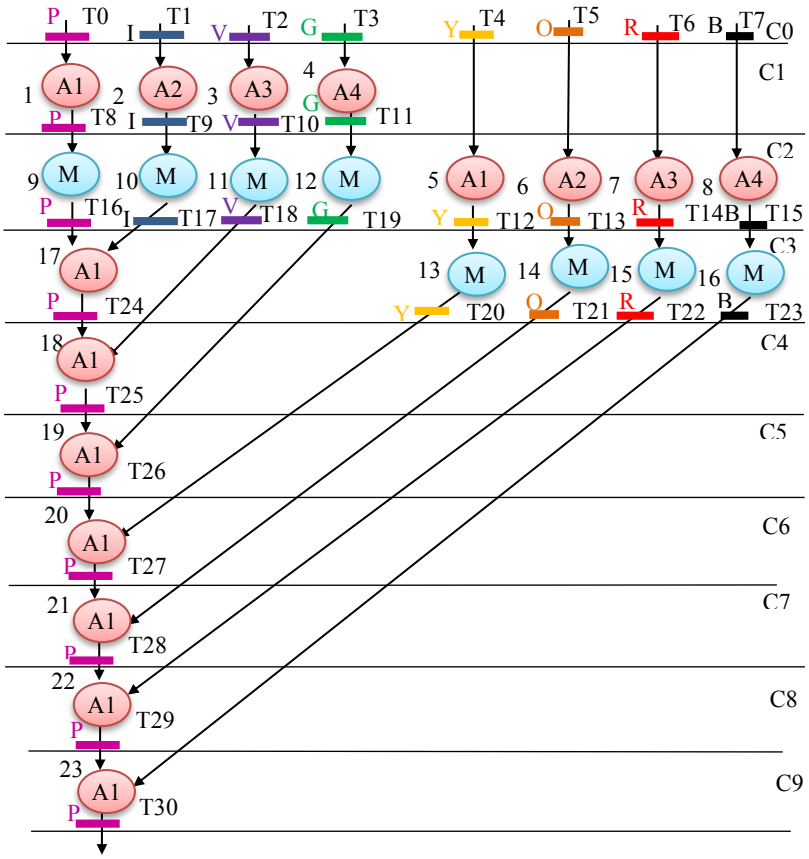


Fig. 3.6. Scheduled FIR using 4*, 4+ before implanting palmprint

distinct registers, where nodes indicate the storage variables, their color indicates the respective register assignment, and an edge between two nodes represents the overlapping of the lifetime of storage variables. The digits in the palmprint digital template are mapped to hardware security constraints in the form of additional edges (secret constraint) in the CIG. The mapping rules of bit '0', bit '1' and 'binary point' of the palmprint digital template are presented in Table 3.4. Along with the mapping rules, the chosen ordering of the storage variables also decides the hardware security constraints to be implanted into the design. It is to be noted that amongst the 31 nodes in the CIG of FIR, the maximum possible constraint edges based on the mapping rule of bit '0' and bit '1' are 120 and 105, respectively. Hence, we consider 225 bits (zeros and ones) of the palmprint signature digital template. However, there are also 6 binary points in the digital template (refer the palmprint signature obtained in previous sub-section) up 225 count of '0' and '1' bits. Hence for FIR filter application, the total size of palmprint signature is considered to be of 231 digits (225+6) for mapping into hardware security constraints based on the mapping rules. Upto 231 digits of digital template,

Table 3.3 Register assignment of storage variables (T0-T30) of FIR digital filter pre-implanting palmprint signature

	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
P	T0	T8	T16	T24	T25	T26	T27	T28	T29	T30
I	T1	T9	T17	--	--	--	--	--	--	--
V	T2	T10	T18	T18	--	--	--	--	--	--
G	T3	T11	T19	T19	T19	--	--	--	--	--
Y	T4	T4	T12	T20	T20	T20	--	--	--	--
O	T5	T5	T13	T21	T21	T21	T21	--	--	--
R	T6	T6	T14	T22	T22	T22	T22	T22	--	--
B	T7	T7	T15	T23	T23	T23	T23	T23	T23	--

there are 98 zeros, 127 ones and 6 binary points. However, as discussed earlier for FIR filter, maximum 105 ones can be converted into hardware security constraints based on the mapping rule of bit '1'. Thus, obtained hardware security constraints corresponding to 98 zeros, 105 ones and 6 binary points are given below:

Security constraints corresponding to 98 zeros of palmprint digital template: <T0,T2>,<T0,T4>,<T0,T6>,<T0,T8>,<T0,T10>,<T0,T12>,<T0,T14>,<T0,T16>, <T0,T18>,<T0,T20>,<T0,T22>, <T0,T24>, <T0,T26>, <T0,T28>, <T0,T30>, <T2,T4>, <T2,T6>, <T2,T8>, <T2,T10>, <T2,T12>, <T2,T14>, <T2,T16>, <T2,T18>,<T2,T20>,<T2,T22>, <T2,T24>, <T2,T26>, <T2,T28>, <T2,T30>, <T4,T6>, <T4,T8>, <T4,T10>, <T4,T12>, <T4,T14>, <T4,T16>, <T4,T18>, <T4,T20>, <T4,T22>, <T4,T24>, <T4,T26>, <T4,T28>, <T4,T30>, <T6,T8>, <T6,T10>,<T6,T12>,<T6,T14>,<T6,T16>,<T6,T18>,<T6,T20>,<T6,T22>, <T6,T24>,<T6,T26>,<T6,T28>, <T6,T30>, <T8,T10>, <T8,T12>, <T8,T14>, <T8,T16>,<T8,T18>,<T8,T20>, <T8,T22>, <T8,T24>, <T8,T26>, <T8,T28>, <T8,T30>,<T10,T12>, <T10,T14>, <T10,T16>, <T10,T18>, <T10,T20>, <T10,T22>, <T10,T24>, <T10,T26>, <T10,T28>, <T10,T30>, <T12,T14>, <T12,T16>, <T12,T18>, <T12,T20>, <T12,T22>, <T12,T24>, <T12,T26>, <T12,T28>, <T12,T30>,<T14,T16>, <T14,T18>, <T14,T20>, <T14,T22>, <T14,T24>, <T14,T26>, <T14,T28>, <T14,T30>, <T16,T18>, <T16,T20>, <T16,T22>, <T16,T24>, <T16,T26>, <T16,T28>

Table 3.4 Mapping rules for generating palmprint security constraints

Digits	Mapping rules
0	Implant an edge between node pair (even, even) into CIG
1	Implant an edge between node pair (odd, odd) into CIG
.	Implant an edge between node pair (0, integer) into CIG

Security constraints corresponding to 105 ones of palmprint digital template: <T1,T3>, <T1,T5>, <T1,T7>, <T1,T9>, <T1,T11>, <T1,T13>, <T1,T15>, <T1,T17>, <T1,T19>, <T1,T21>, <T1,T23>, <T1,T25>, <T1,T27>, <T1,T29>, <T3,T5>, <T3,T7>, <T3,T9>, <T3,T11>, <T3,T13>, <T3,T15>, <T3,T17>, <T3,T19>, <T3,T21>, <T3,T23>, <T3,T25>, <T3,T27>, <T3,T29>, <T5,T7>, <T5,T9>, <T5,T11>,<T5,T13>,<T5,T15>, <T5,T17>, <T5,T19>, <T5,T21>, <T5,T23>, <T5,T25>, <T5,T27>, <T5,T29>, <T7,T9>, <T7,T11>, <T7,T13>, <T7,T15>, <T7,T17>,<T7,T19>,<T7,T21>,<T7,T23>,<T7,T25>,<T7,T27>, <T7,T29>, <T9,T11>,<T9,T13>,<T9,T15>,<T9,T17>, <T9,T19>, <T9,T21>, <T9,T23>, <T9,T25>, <T9,T27>, <T9,T29>, <T11,T13>, <T11,T15>, <T11,T17>, <T11,T19>, <T11,T21>, <T11,T23>, <T11,T25>, <T11,T27>, <T11,T29>, <T13,T15>, <T13,T17>, <T13,T19>, <T13,T21>, <T13,T23>, <T13,T25>, <T13,T27>, <T13,T29>, <T15,T17>, <T15,T19>, <T15,T21>, <T15,T23>, <T15,T25>, <T15,T27>, <T15,T29>, <T17,T19>, <T17,T21>, <T17,T23>, <T17,T25>, <T17,T27>, <T17,T29>, <T19,T21>, <T19,T23>, <T19,T25>, <T19,T27>, <T19,T29>, <T21,T23>, <T21,T25>, <T21,T27>, <T21,T29>, <T23,T25>, <T23,T27>, <T23,T29>, <T25,T27>, <T25,T29>, <T27,T29>

Security constraints corresponding to 6 binary points of palmprint digital template:

<T0,T1>, <T0,T3>, <T0,T5>, <T0,T7>, <T0,T9>, <T0,T11>. Thus, palmprint biometric-based secret hardware security constraints are generated using IP vendor-specified mapping rules.

(b) Implanting palmprint signature and RTL generation

Post obtaining hardware security constraints corresponding to the palmprint signature, they are implanted into the target DSP design during HLS process. In order to do so, a CIG framework of respective design is exploited where security constraints are added as secret constraint (additional) edges into the CIG. This sub-section presents the implantation process of hardware security constraints, corresponding to the palmprint signature (obtained earlier), into FIR filter design through its CIG framework.

The number of hardware security constraints corresponding to zeros, ones, and binary points are 98, 105 and 6, respectively. These constraints are implanted into the CIG of FIR filter in the form of secret additional edges**.** During the

implantation of constraint edges, some are intended to be added between two such nodes whose colors are same. However, an edge cannot be added between two nodes of same color. This is because an edge between two nodes of same colors indicates that both storage variables (nodes in the CIG) are assigned to execute through the same register (color) in the same control step, which is not possible. Therefore, this conflict is resolved in the following two ways:

(i) Local alteration in the register allocation of storage variables: in this case, register/color of a storage variable is swapped with the register of another storage variable in the same control step. For example, storage variables T10 and T11 are swapped in control step C1 to enable the implantation of constraint edge <T2, T10>. Similarly, local alterations in the register assignment of storage variables T12, T13, T14, T15, T16 and T17 are made in control step C2. This impact on register allocation has been shown in Table 3.5.

(ii) Requirement of extra registers to satisfy the constraint edges: this situation arises when swapping of register/color of a storage variable with another register in the same control step is not possible. Therefore, extra colors are used in the CIG to enable the implantation of constraint edges. This leads to extra registers in the design. Table 3.5 shows the following extra registers/colors are required to satisfy the all-constraint edges: Br, C, L, LB, LG, T, A (highlighted in red color in the table). The overall impact of

Table 3.5 Register assignment of storage variables of FIR digital filter post implanting palmprint signature

	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
P	T0	--	T17	--	--	--	--	--	--	--
I	T1	--	T16	--	--	--	--	--	--	--
V	T2	T11	--	--	--	--	--	--	--	--
G	T3	T10	--	--	--	--	--	--	--	--
Y	T4	T4	T13	--	--	--	--	--	--	--
O	T5	T5	T12	--	--	--	--	--	--	--
R	T6	T6	T15	--	--	--	--	--	--	--
B	T7	T7	T14	--	--	--	--	--	--	--
Br	--	T8	T19	T19	T19	--	--	--	--	
C	--	T9	--	T24	--	T26	--	T28	--	T30
L	--	--	T18	T18	T25	--	--	--	--	--
LB	--	--	--	T20	T20	T20	T27	--	--	--
LG	--	--	--	T22	T22	T22	T22	T22	T29	--
T	--	--	--	T21	T21	T21	T21	--	--	--
A	--	--	--	T23	T23	T23	T23	T23	T23	--

implanting constraint edges on register allocation of the FIR filter design is shown in Table 3.5. Thus, the modified register allocation of storage variables is also shown in the scheduled DFG of FIR filter, in Fig. 3.7. Subsequently, datapath synthesis phase of HLS is performed. This therefore results into generation of the RTL design with IP vendor-selected embedded palmprint security constraints.

3.3.2. Detection of palmprint signature

The proposed palmprint biometric-based hardware security approach provides seamless and robust detection of counterfeiting of DSP coprocessors. The process of counterfeit detection using proposed approach is shown in Fig. 3.8. As shown in the figure, presence of authentic palmprint signature is verified within the design in order to discern the authentic and counterfeited ones. In order to do so, authentic palmprint signature is regenerated by a SoC integrator using the proposed algorithm during the detection process. To regenerate the palmprint signature and corresponding hardware security constraints, the following information are required: (a) original palmprint

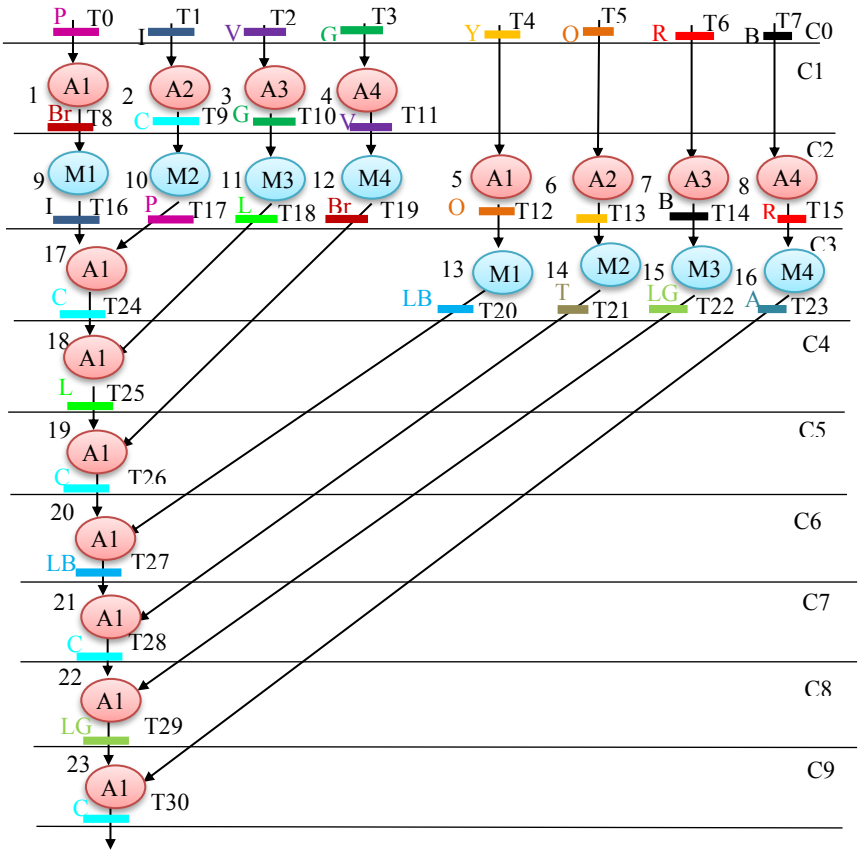


Fig. 3.7. Scheduled FIR post implanting palmprint

image with grid size/spacing/nodal points (b) naming conventions assigned to nodal points (c) coordinates of palm features nodal points (d) set of palm features chosen (e) sequence of the concatenation of features for generating palmprint signature (f) constraint mapping (encoding) rules. Based on this information, obtained hardware security constraints are detected within the RTL datapath of design under-test by inspecting the register allocation information. If the palmprint security constraints do not match with the register allocation information of the design, then the palmprint signature is absent and the design is a counterfeit.

The generated palmprint signature used proposed algorithm is robust because it acts as a highly strong secret mark that cannot be imitated by an adversary in order to evade the detection process of fake designs. This is because of the following reasons: (i) an individual always has unique palm features resulting into a unique signature (ii) the palmprint signature generation depends on several factors which an adversary is not aware of. For example: fixed grid size and spacing on the palm image, set of selected palm features among the exhaustive features, precise coordinates of the palm feature nodal points and sequence of the concatenation of the features (iii) during the detection process, positions of '0' bits, '1' bits and 'binary point' in the palmprint digital template also play a critical role. Therefore, it is not possible that an adversary could regenerate the same digital template and embed it into the counterfeit designs in order to evade the detection process.

3.4. Metrics for Evaluating Security Strength of Proposed

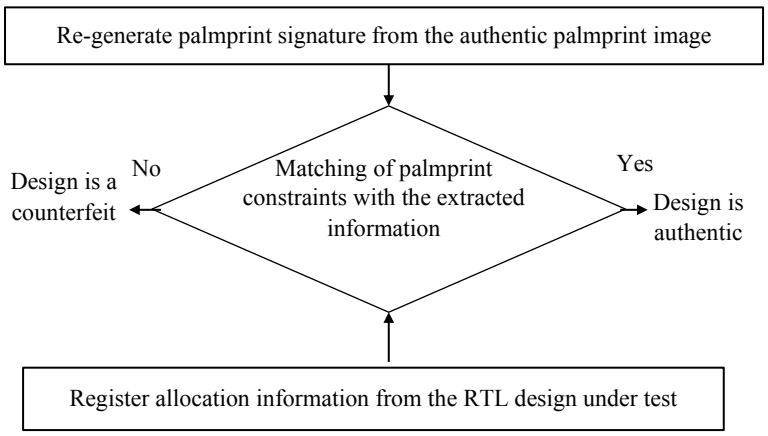


Fig.3.8. Detection of IP counterfeiting using proposed palmprint biometric

Palmprint Biometric Approach

In order to measure the effectiveness of the proposed hardware security approach in terms of achieved security strength, following security metrics are used (a) probability of coincidence (Pc) (b) tamper tolerance (TT):

a) The Pc metric is formulated as follows [31], [32], [36]-[40]:

$$\mathrm{Pc} = \left(1 - \frac{1}{\mathrm{x}}\right)^{\mathrm{z}} \tag{3.1}$$

Where, 'x' is the number of colors in the CIG or the number of registers into the register allocation table of the DSP design before implanting palmprint constraints and 'z' is the number of constraint edges implanted into the CIG of the design. Here the value of Pc signifies the probability of finding the authentic palmprint constraints in an unsecured design by coincidence by an attacker. Therefore, the lower the value of Pc, higher is the strength of the authentic palmprint signature embedded into the design. It can be observed that a very low value of Pc can be achieved if embedding of a greater number of security constraints is possible.

b) Further, the tamper tolerance (TT) metric is formulated as follows [31], [32], [36], [39], [40]:

$$\mathrm{TT} = \mathrm{W}^{\mathrm{S}} \tag{3.2}$$

Where, W is the number of types of digits in the signature and S is the signature size (or the number of corresponding hardware security constraints). As evident from (2), the tamper tolerance ability is measured in terms of total signature space. The larger the signature space, the lower is the probability that an attacker would find the exact signature and attempt tampering. Therefore, if a security methodology is able to generate a greater number of security constraints and also comprises of a greater number of signature digits, then its tamper tolerance ability will be higher. Further, because of high tamper tolerance ability, an attacker cannot find the exact palmprint signature to attempt tampering in the form of regeneration of duplicate signature. This incapacitates an attacker from duplicating the authentic palmprint signature and embedding into fake designs for evading piracy/counterfeit detection process.

3.5. Metric for Evaluating Impact of Proposed Palmprint Biometric on Design Cost

The embedding of IP vendor selected security signature may impact the design cost. This is because, implantation of secret hardware security may require additional hardware (registers) or control steps to accommodate them. Further, additional hardware may lead to extra design area and extra control steps may lead to extra design latency (in case if required). Therefore, to evaluate the feasibility of the proposed approach, the design cost post embedding palmprint biometric security constraints is required to be evaluated. A security methodology is feasible if it incurs lesser design cost overhead while offering robust security strength. The impact of embedding the proposed palmprint signature on design cost (C_d) is measured using the following function [31], [32], [36]-[40]:

$$C_d = g_1 \frac{A_d}{A_m} + g_2 \frac{L_d}{L_m} \tag{3.3}$$

Where, A_d and L_d are the design area and latency, A_m and L_m are the maximum area and latency of the design, g_1 and g_2 are the weights of area and latency in the design cost.

3.5. Results and analysis

The results of the proposed palmprint approach indicated a stronger probability of co-incidence in the range of (4.01E-14 to 4.23E-4) which is significantly lesser than related approaches (desirable) and stronger tamper tolerance in the range of (7.6E+12 to 1.6E+110) which is significantly higher than related approaches (desirable). The robust security of the different DSP benchmarks using palmprint biometric security is achieved at negligible design cost overhead (<1%). The experimental results of the contact-less palmprint biometric approach have been discussed and analyzed in chapter 9 of this thesis.

3.6. Summary

A novel hardware security approach for DSP coprocessors using palmprint biometrics has been presented in this chapter. This approach implanted

authentic palmprint signature during HLS phase of design process to enable detective control against IP piracy/counterfeiting. The security of DSP based coprocessors against piracy has been targeted in the proposed approach to disable integration of counterfeited designs in the SoCs. This ensures the security of end consumers from unreliable and unsafe components integrated into CE systems. Additionally, the proposed approach is measured in terms of its security and design cost to evaluate its effectiveness. The palmprint-based approach generates naturally unique encoded hardware security constraints for covertly implanting into the design. This, therefore, is capable of offering stronger security in terms of lower probability of coincidence (indicating stronger digital proof of evidence against fake IP cores) and higher tamper tolerance (indicating stronger defense against the regeneration of embedded secret signature by an adversary), which is the strength of the proposed palmprint approach.

Chapter 4

Double line of Defense Approach for Securing DSP IP Cores using Structural Obfuscation and Chromosomal DNA Impression

This chapter presents a novel security mechanism for enabling a double line of defense against the hardware threats of (a) reverse engineering (RE) and (b) piracy. An adversary in untrustworthy design houses may attempt to perform reverse engineering the design to obtain/analyze the internal functionality and details of the design. This is because a successful attempt of reverse engineering enables an adversary to secretly implant malicious logic into safe places (not easily detectable during normal executions) of the design. This in turn may lead to security concerns to end consumers and sabotaging the reputation of the original vendor or IP seller. Thus, to accomplish his/her malicious intention of causing security hazards to end consumers by malfunctioning the CE systems and sabotaging the reputation of the original vendor, an adversary induces the RE attack. Mainly, RE is a process by which an adversary attempts to extract the design details by back-propagating it to a desired higher level of abstraction from a given or available lower level of abstraction. Further, ensuring absolute security against RE attack may not be possible to achieve due to deployed semiconductor supply chain scenario, where IPs for different application frameworks are imported from untrustworthy multi-party vendors (causing security hazards). However, the process for RE the design can be thwarted by making it more complex and time-consuming to an adversary. In order to hinder an adversary from successfully performing RE attack, structural obfuscation technique makes the design unobvious and uninterpretable to an adversary by modifying the internal structure without causing change in its desired functionality. Further, an adversary (rogue element) in the third-party design houses may also attempt to perform piracy of IPs. The pirated IP versions may also lead to security hazards to end consumers. This is because pirated/fake components may not be thoroughly checked and verified before integration into SoCs systems. Therefore, security against pirated IP versions is also equally important for ensuring the trust into safe usage of computing and CE systems.

A novel structural obfuscation and chromosomal DNA impression-based hardware security technique have been presented in this chapter for securing the IP core design against RE attack (as the first level of security) and enabling detective control against piracy (as the second level of security). The first section formulates the problem. The second section discusses the hardware security mechanism with a double line of defense under the following sub-sections: overview, structural obfuscation mechanism against RTL alteration, encoded chromosomal DNA framework, encryption mechanism for generating the encrypted DNA signature, encoding algorithm for generating the secret hardware security constraints. The third section demonstrates the process for generating a secured 4-point DFT design (structurally obfuscated) using DNA signature under the following subsections: implanting the hardware security constraints for generating secured 4-point DFT against IP piracy and security properties of the methodology achieved through encrypted chromosomal DNA impression. Finally, the fourth section summarizes the chapter.

4.1. Problem Formulation

Given the target DSP application(s) in the form of data flow graph (DFG) representations, resource constraints, module library, secret key for different encryption rounds and DNA base pairs along with the objective of securing IP cores in terms of hindering RE attack and enabling detective control against piracy. Therefore, generating a secured (structurally obfuscated design with embedded security signature) integrated RTL design of respective DSP cores.

4.2. Security Mechanism with Double Line of Defense for Securing IP Core Design

This section discusses the proposed hardware security mechanism with double line of defense under the following sub-sections:

4.2.1. Overview

The structural obfuscation and DNA based hardware security methodology, advances CE systems security and covers consumers' security in terms of their safe usage, by protecting the underlying DSP hardware cores against the threats of counterfeiting. Furthermore, it also offers benefits from a SoC integrator's or product designer's perspective. Therefore, the proposed

methodology is a mechanism to hinder register transfer level (RTL) description alteration using structural obfuscation and a detective measure against piracy/counterfeiting threat. By detecting a designer's authentic mark in the IP cores, the SoC integrator can refrain from using fake IP components in the CE products and make sure of using only authentic designs.

The overview of the hardware security methodology is depicted in Fig. 4.1.

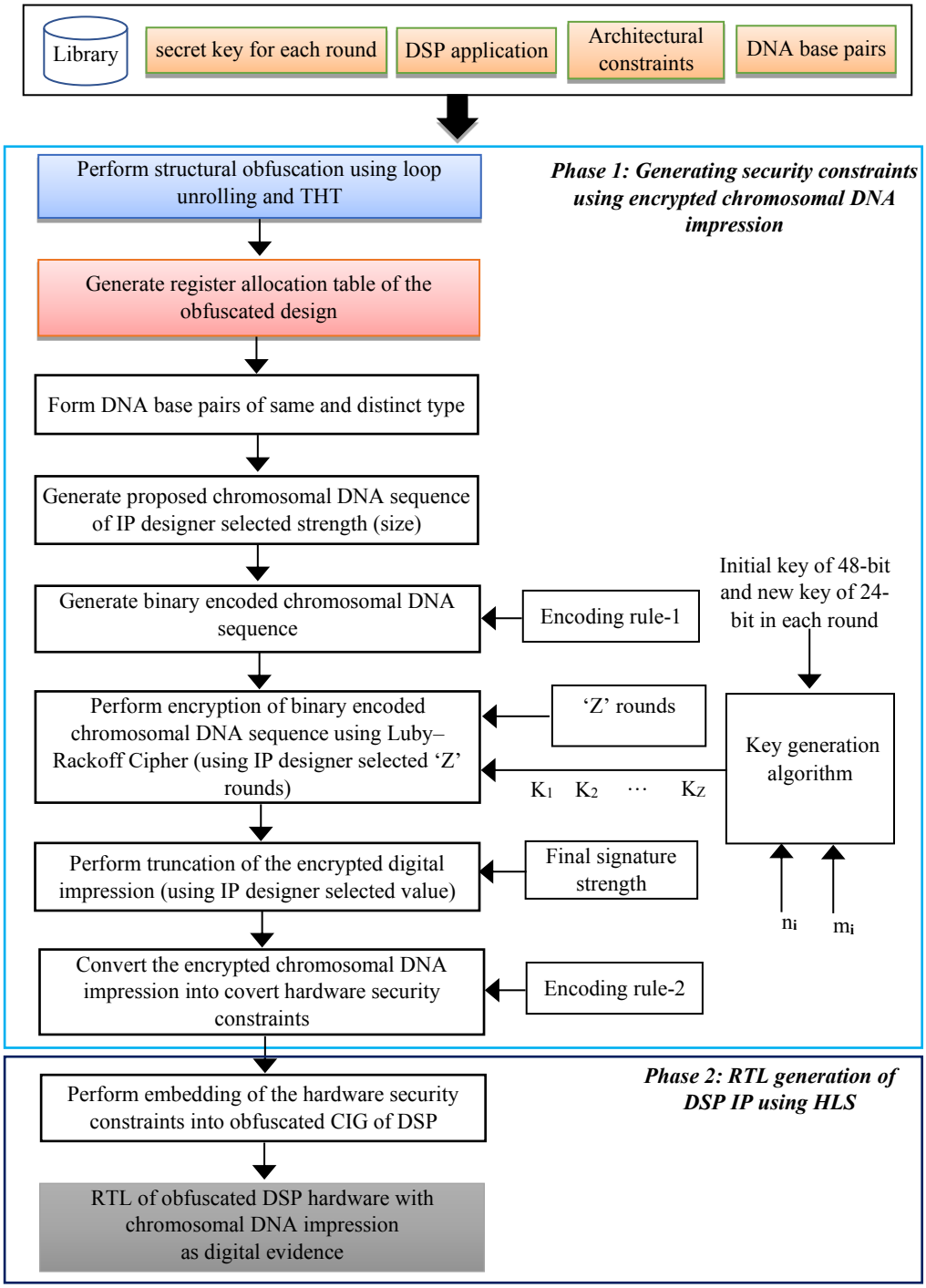


Fig.4.1. Overview details of proposed methodology based on chromosomal DNA impression

The methodology has been discussed using four major steps:

a) The first step is responsible for generating the security constraints using the chromosomal DNA impression for the structurally obfuscated hardware design. The first step considers a DSP application as its input. In this phase, initially, structural obfuscation using high-level transformation has been performed.
b) In the next step, subsequently, two DNA base pairs have been formed on the basis of four chemical elements. The two DNA base pairs which then form the chromosomal DNA sequence by taking alternative base pairs of the same as well as of distinct type as part of digital DNA impression generation process. In this phase, hardware security constraints are finally produced as an output based on the IP core designer's selected impression strength (size) of the encrypted impression.
c) In the third step, embedding of these hardware security constraints into obfuscated colored interval graph (CIG) of DSP hardware is performed.
d) In the fourth step, post embedding the obfuscated DSP hardware register transfer level (RTL) with encrypted DNA impression as digital evidence is generated.

As shown in Fig. 4.1, the input block of the proposed methodology consists of the DSP application (in the form of a control data flow graph (CDFG)), resource constraints for the structurally obfuscated design, library, secret keys for the IP designer selected rounds and DNA base pairs. The output block consists of the RTL circuit of the obfuscated DSP hardware post embedding the encrypted DNA impression as digital evidence. The overview of the functionality of each block is as follows:

- The first block is responsible for structural obfuscation of the DSP application using loop-based high-level transformation, i.e., loop unrolling.
- The second block then performs the non-loop-based high-level transformation, i.e., tree height transformation on the design architecture produced by the first block.
- The third block is responsible for generating the register allocation table of the structurally obfuscated design produced by the second block.

- The fourth block is responsible for forming the DNA base pairs of same as well as of distinct type based on the chemical elements.
- Subsequently, the next block is responsible for generating the chromosomal DNA sequence based on the strength (size) selected by the IP designer.
- The next block then generates the binary encoded chromosomal DNA sequence for the sequence produced by the previous block, based on an encoding rule-1.
- The next block is responsible for performing the encryption using Luby–Rackoff Cipher on the binary-encoded chromosomal DNA sequence (produced by the previous block). The encryption process accepts the keys generated by the proposed key generation algorithm based on the number of rounds (Z) selected by the IP designer.
- Subsequently, the next block is responsible for performing the truncation on the digital DNA impression, depending upon the final digital impression strength selected by IP designer.
- The final block of the first phase then converts the encrypted chromosomal DNA impression (selected by the IP designer as output of the previous block) into covert hardware security constraints based upon an encoding rule-2. These obtained hardware security constraints (based on the structural obfuscation of the hardware design) are given as input to the RTL generation phase, responsible for embedding the hardware security constraints into the obfuscated CIG of the DSP application. Then, the RTL circuit of the obfuscated DSP hardware with encrypted chromosomal DNA impression is generated as digital evidence.

4.2.2. Structural Obfuscation Mechanism against RTL Alteration

Hardware structural obfuscation obscures the actual hardware design architecture of the DSP IP core to protect it from an adversary attempting to alter the RTL description. Structural obfuscation is performed through several loop-based and non-loop-based high-level transformations. Structural obfuscation transforms the generic hardware design architecture into obfuscated design architecture without compromising its actual functionality. It makes it almost impossible and challenging for an adversary to alter the

original RTL description, in order to correctly interpret the functionality and hardware interconnection from the structurally obfuscated design. In the proposed methodology, DSP applications (such as FIR and DFT) are accepted as input, and then structural obfuscation is performed over them in order to make them secure against attacks from an adversary. In order to do so, structural obfuscation on DSP applications has been performed using THT and LU algorithms. The un-obfuscated CDFG of FIR filter is shown in Fig. 4.2 and the corresponding obfuscated FIR filter using structural obfuscation based on THT is shown in Fig. 4.3, respectively. THT divides the critical path computation into multiple sub-computations and then executes them in parallel. THT-based structural obfuscation results into change in the interconnectivity of the RTL datapaths of the DSP hardware in terms of multiplexer size, demultiplexer size, storage element etc., without affecting the functionality. This, therefore, produces unobvious architecture of the respective DSP hardware and thwarts alteration of the original RTL design. On the other hand, the loop transformation unrolls the loop-based application depending on the unrolling factor. LU executes the same calculation present inside the loop multiple times. Loop unrolling-based structural obfuscation also results into change in RTL datapath in terms of multiplexer size, demultiplexer size, storage element etc., without affecting the functionality. In the 4-point DFT application, tree height transformation (THT) and LU have been performed to obfuscate the structure of the application. The respective

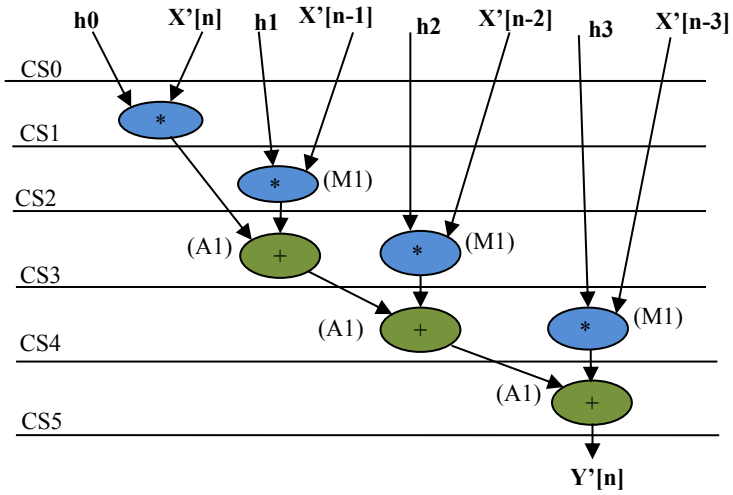


Fig. 4.2 Scheduled DFG of FIR based on resource constraint (1M,1A)

non-obfuscated and obfuscated scheduled CDGF of 4-point DFT, is shown in Fig. 4.4 and Fig. 4.5, respectively.

4.2.3. Background on DNA framework

Deoxyribonucleic acid, also known as genome is a molecule that contains the unique biological information that makes each species unique in the sense of characteristics and identification. DNA is comprised of chemical building blocks called nucleotides and each nucleotide is composed of three different components such as sugar, phosphate groups and nitrogen bases. The sugar and phosphate groups link the nucleotides together to form each strand of DNA. The four chemical elements thymine, adenine, guanine, and cytosine are the four types of nitrogen bases. The nucleotides are joined together by covalent bonds between the phosphate of one nucleotide and the sugar of the next, forming a phosphate-sugar backbone 'S' from which the nitrogenous bases protrude. The configuration of the DNA molecule is highly stable, allowing it to act as a template [97]. The genome sequence used in the proposed work is generated from real human body sample that comprises of real chemical foundations Thymine (T), Adenine (A), Guanine (G), Cytosine (C) and sugar phosphate backbone (polynucleotide) of genome. The derived genome sequence as shown in section 4.2.4, is a real human body genome sample that is used for generating the crypto genome signature, its associated security constraints, and subsequently embedded into the design during HLS. This results in a secure RTL datapath corresponding to the input DSP application framework with detective control against IP piracy. This chapter discusses how the DNA sequence of an IP vendor can be exploited to generate secure digital evidence that acts as a secret authentic mark for providing detective control against pirated IP core versions. The proposed methodology offers the following advantages in terms of using genome sequence over physical biometric techniques for hardware security:

a) sequencing does not depend on any external factor such as dirt and grease as the process of examining genome sequence and generating crypto genome signature is not affected by external environmental factors.

b) genome sequence is more unique in terms of robust security than facial biometric. Further, it is not possible for an adversary to forge or regenerate the

original genome sequence of the true IP vendor in the context of proving false IP ownership and evading piracy detection process.

4.2.4. Encoded Chromosomal DNA Framework

In the proposed chromosomal DNA model, two base pairs (BP) of chromosomal DNA have been taken. First base pair (BP-1) is formed with two chemical elements Thymine (T) and Adenine (A) whereas the second base pair (BP-2) is formed with two other chemical elements named Guanine (G) and Cytosine (C). Subsequently, from these two base pairs (BP-1 and BP-2) chromosomal DNA sequence can be formed in two ways, either by considering the alternative base pairs of same type or by considering alternative base pairs of distinct type, as presented in Fig. 4.6. The final chromosomal DNA sequence has been formed by adding the polynucleotide (Sugar phosphate backbone) represented as 'S'. Polynucleotide has been added as leading and lagging strand in the DNA sequence. The order or sequence of these chemical elements is used for determining the instructions that are contained in a strand of DNA. Thus, different sequence orders represent different and unique information. An example of IP designer created possible chromosomal DNA sequence with alternative base pairing comprising of same type of base pairs, is shown in Fig. 4.7. Similarly, chromosomal DNA sequence for alternative base pairs of distinct types can also be generated. Consequently, the chromosomal DNA sequence selected by

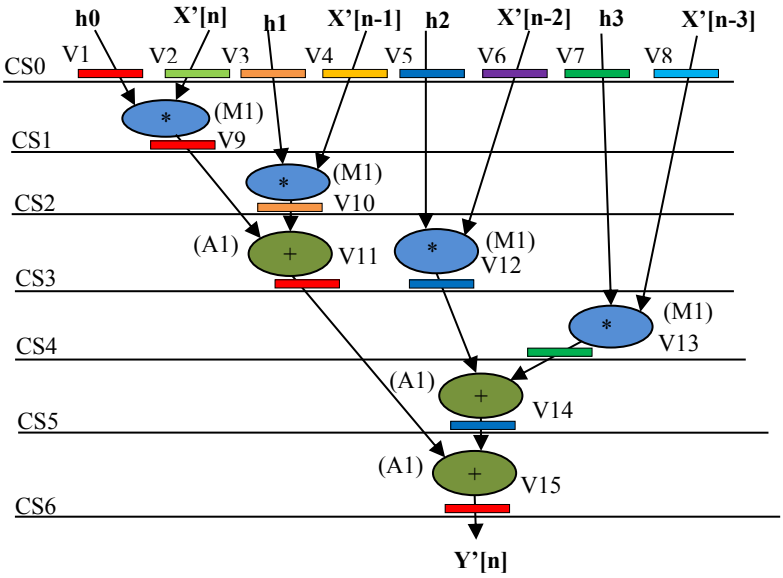


Fig. 4.3. Scheduled, hardware allocated and binded DFG of FIR based on resource constraint (1M,1A) after high level transformation

IP designer (either with alternative base pairs of same type or distinct type) can be encoded into binary digits using IP designers specified encoding rule-1. The encoding rule-1 for all the chemical elements (A, T, G, C and S) used in formation of final chromosomal DNA sequence, is shown below:

Element ‘A’(alphabet value=1) is being encoded in binary as ‘1’, ‘T’(20) as ‘10100’, element ‘G’(7) as ‘111’, ‘C’(3) as ‘11’ and ‘S’(19) as ‘10011’. An example of a final chromosomal DNA sequence with alternative base pairing of same type (as shown in Fig. 4.7) is depicted below:

STAS-SATS-SGCS-SCGS-STAS-SATS-SGCS-SCGS-STAS-SATS-SGCS-SCGS-STAS-SATS-SGCS-SCGS- - ----

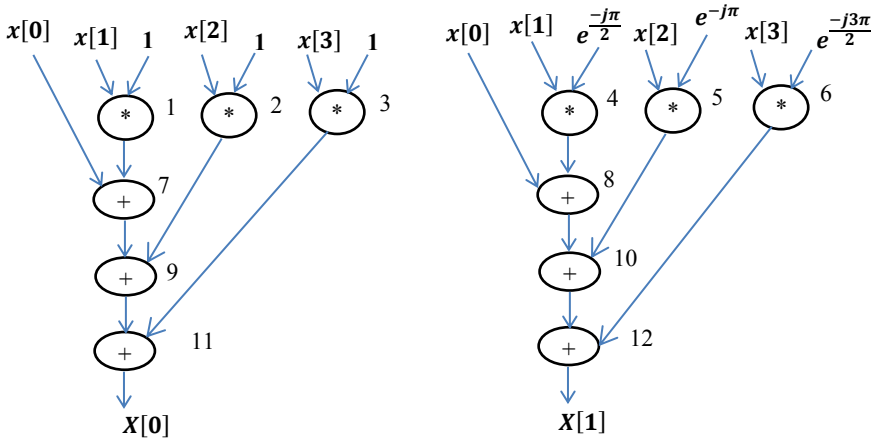


Fig. 4.4. DFG of 4-point DFT computing two samples at a time

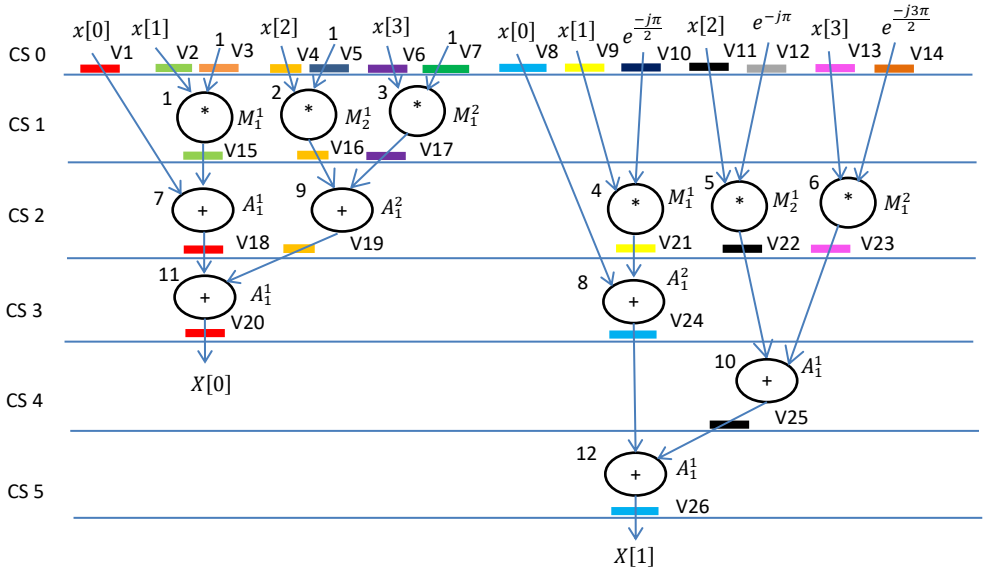


Fig. 4.5. Scheduled DFG of obfuscated 4-point DFT based on resources constraints of 3M and 2A

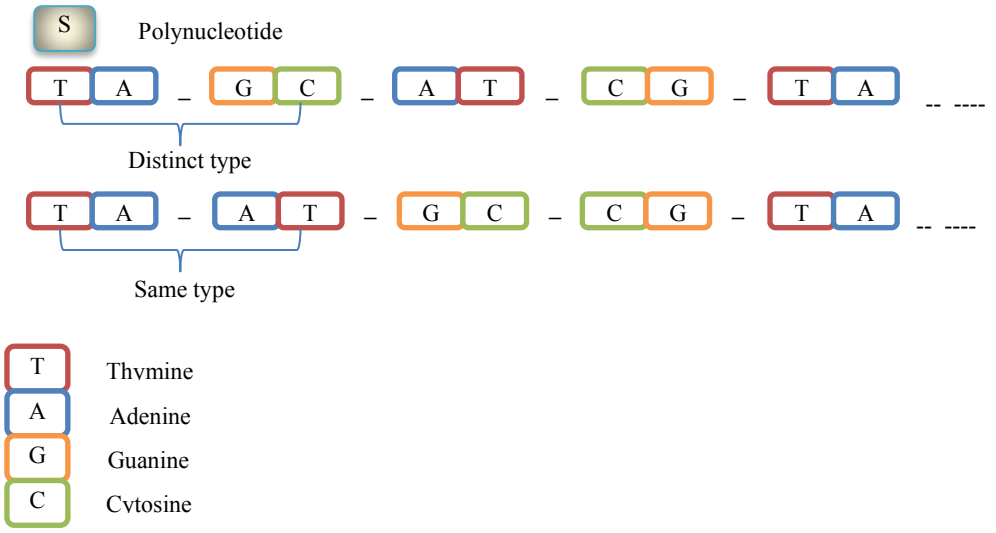


Fig. 4.6. Proposed chromosomal DNA with distinct/same type base pairs

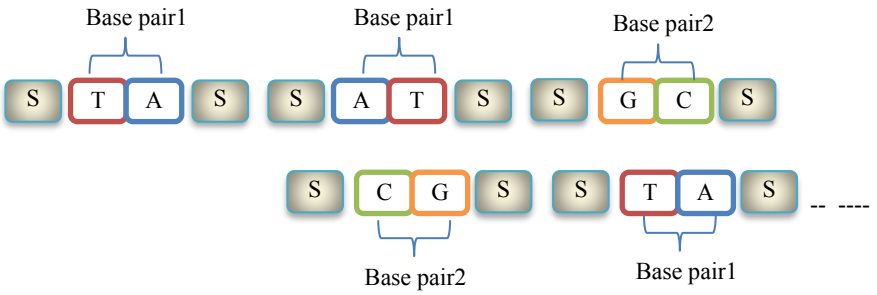


Fig. 4.7. Example of a possible chromosomal DNA sequence with base pairs and polynucleotide using proposed work

The corresponding binary encoded DNA impression, (for e.g., 128 bit), formed using encoding rule-1, is shown below:

1001110100110011-1001111010010011-100111111110011
100111111110011-1001110100110011-1001111010010011-
100111111110011-100111111110011-1001 (4.1)

Similarly, an example of a final chromosomal DNA sequence with alternative base pairing of distinct type can be created. The corresponding binary encoded digital DNA impression (for e.g., 128 bit), formed using encoding rule-1, is shown below. (*Note: The 128-bit DNA impression is also expandable upto designer specified strength such as 256-bit, 512-bit etc.*).

1001110100110011-1001111111110011-1001111010010011-
100111111110011-10011101001100 11-100111111110011-
1001111010010011-100111111110011-1001 (4.2)

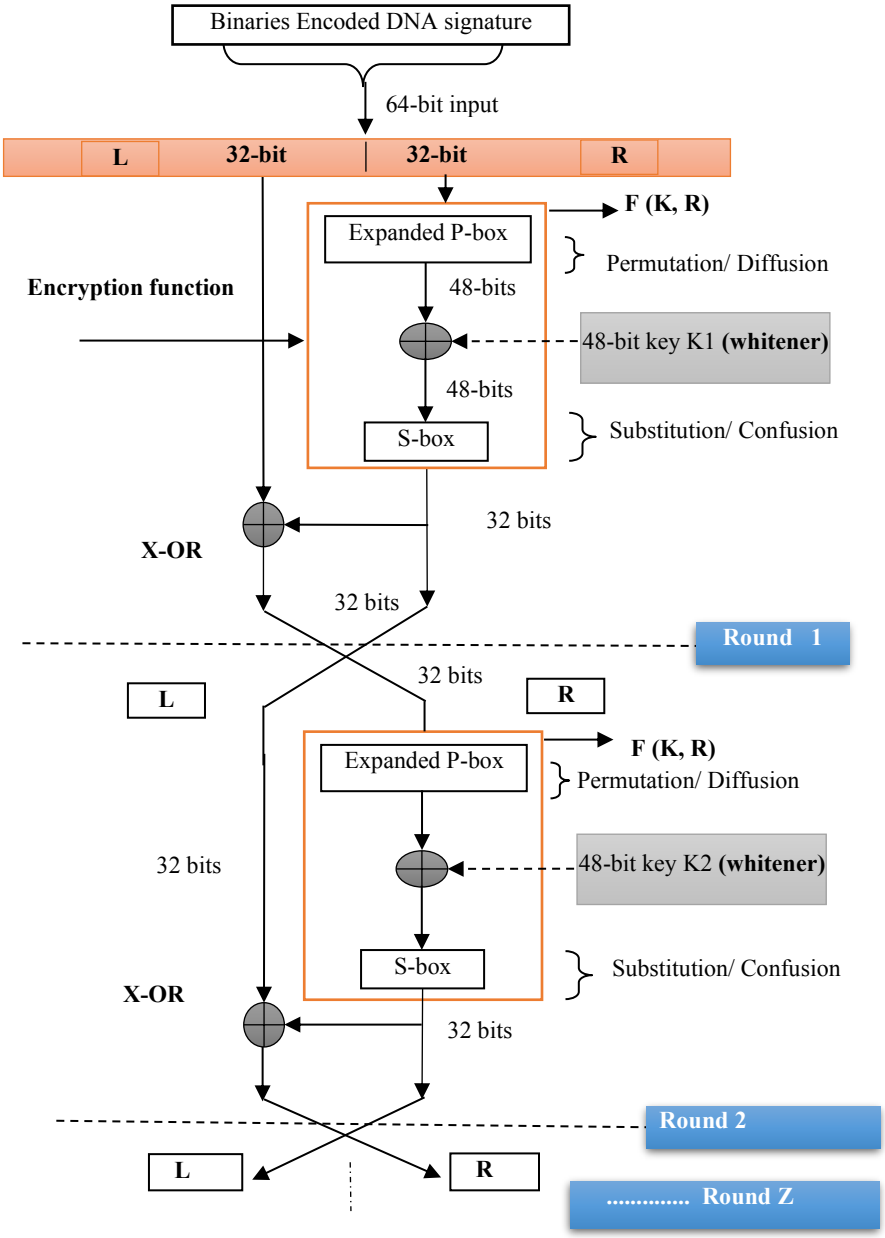


Fig. 4.8. Encryption process using Feistel cipher

4.2.5. Encryption Mechanism for Generating the Encrypted DNA Signature

The generated binary encoded chromosomal DNA impression with the alternative base pairs of the same or distinct type can be fed into the Feistel cipher for encryption purpose. For an instance, encoded chromosomal DNA impression with alternative base pairs of distinct type has been fed into the multi-round Feistel cipher process, as shown in Fig. 4.8. The 128-bit binary encoded chromosomal DNA sequence having alternative base pair of distinct type is further divided into two segments (64 bit each) and fed into the cipher process iteratively. In the first round of Feistel cipher, the initial 64-bit binary

encoded output (the first segment) of chromosomal DNA impression is bifurcated into two parts as left 'L' and right 'R' of 32 bit each. Subsequently the right part is supplied into the encryption function 'F (K, R)' which is capable of performing diffusion (permutation) and confusion (substitution) on the input value. Diffusion is performed by an expanded P-box, whereas confusion is performed by S-box mechanism. The expanded P-box proceeds with right part (R) by accepting it as its input and transforms it into 48-bit output, which then gets XORed with the 48-bit key (K1 for round 1) generated through key generation process (shown in Fig. 4.9). This 48-bit output of X-OR function is then fed into the S-box, which after the substitution, transforms it into 32-bit size. Subsequently, this 32-bit encrypted impression is XORed with the left part 'L' (32-bit) of the initial 64-bit encoded digital DNA impression. Consequently, the 64-bit encrypted chromosomal DNA impression is obtained after the first round. This process continues for the rounds 'Z' selected by the IP designer (where a separate key for each round is fed by the IP designer).

The key generation process is shown in Fig. 4.9. As can be observed from Fig. 4.9, the initial 48-bit key (K1 for round-1) is bifurcated into two parts, 24-bit each. The left 24-bit and right 24-bits are fed into the circularly left shift and circularly right shift functions, respectively. The output of both these functions is then XORed, which generates the 24-bit key value. Subsequently, this 24-bit value is concatenated with a fresh 24-bit key value selected by the

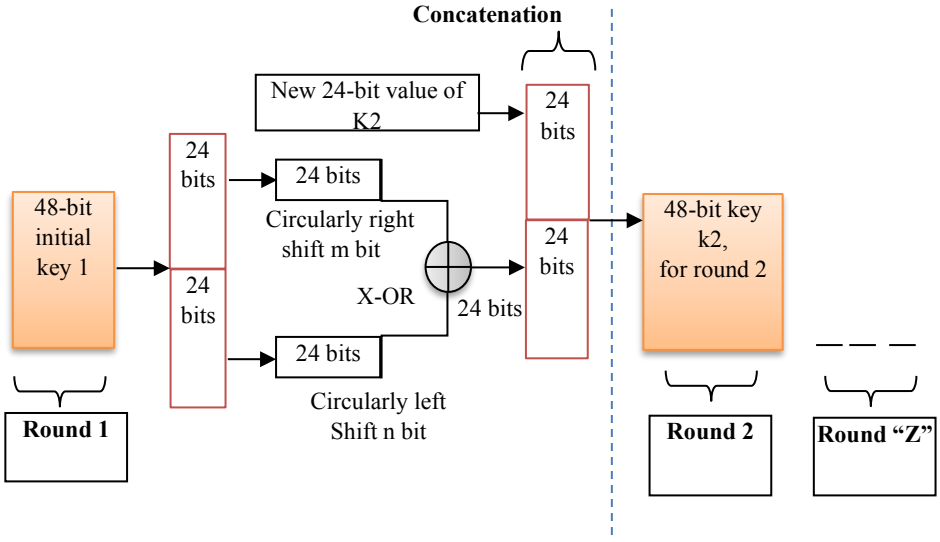


Fig. 4.9. Key generation process in Feistel encryption framework

IP designer, which generates the key K2 for the next round. Thus, by operating the keys (K1, K2….KZ) on each round of Feistel cipher process (selected by the IP designer) it produces the encrypted chromosomal DNA impression. Similarly for the second segment of encoded chromosomal DNA impression, encrypted chromosomal DNA impression is also generated.

4.2.6. Generating the Hardware Security Constraints

As observed in Fig. 4.1, truncation needs to be performed (using the IP designer selected value e.g., 32, 64, and 128 bit) on the final encrypted chromosomal DNA impression resulting from the Feistel cipher. Subsequently, the encrypted impression based on IP designer selected strength is converted into its respective hardware security constraints for embedding into the DSP design using IP designer-specified encoding rule-2. In the encoding rule-2, bit '0' signifies embedding an artificial edge between the storage variable node pair (even-even) of the obfuscated CIG of DSP application; while bit '1' signifies embedding an artificial edge between storage variable node pair (odd-odd) of the obfuscated CIG of DSP application. The 128-bit encrypted chromosomal DNA impression, generated using Feistel cipher, corresponding to the genome/DNA binary encoded digital impression (shown in sequence (4.2) in section 4.2.3) is shown below:

"00100101011000000101001111011101101011101101011111011100100000110010010001110111000001001001010111000100101001000100010010110110100".

For example, the hardware security constraints for the above 128-bit encrypted chromosomal DNA impression corresponding to 4-point DFT (comprising of storage variables (V1-V26)), using the encoding rule-2 are as follows:

<V2,V4>, <V2,V6>, <V2,V8>, <V2,V10>, <V2,V12>, <V2,V14>, <V2,V16>, <V2,V18>, <V2,V20>, <V2,V22>, ……..<V12,V18>, <V12,V20>, <V12,V22>, <V12,V24>, <V12,V26>, <V14,V16>, <V14,V18>, <V14,V20>, <V14,V22>, <V14,V24>, <V14,V26>, <V16,V18>, <V16,V20>, <V16,V22>, <V16,V24>, <V16,V26>, <V18,V20>, <V1,V3>, <V1,V5>, <V1,V7>,……..<V11,V17>, <V11,V19>, <V11,V21>, <V11,V23>, <V11,V25>, <V13,V15>,

<V13,V17>. Thus, the secret hardware security constraints corresponding to encrypted chromosomal DNA signature are generated as the outcome of the first phase of the methodology.

4.3. Demonstration on Generating Secured 4-point DFT Design

This section discusses the proposed hardware security mechanism with double line of defense under the following sub-sections:

4.3.1. Implanting Hardware Security Constraints for Generating Secured 4-point DFT Design against IP Piracy

For the sake of demonstrating the embedding the encrypted DNA signature, 4-point DFT application has been employed. Now, in order to secure the design (post performing the high-level transformation) in terms of enabling detective control against piracy, the covert implantation of the generated covert hardware security constraints into obfuscated design using HLS is performed. Register allocation tables representing the assignment of storage variables of the obfuscated 4-point DFT, before and after implantation of the constraints, is shown in Table 4.1 and Table 4.2, respectively. In Table 4.1, the assignment of storage variables (V1-V26) to fourteen distinct registers (colors) and scheduling (timing steps) represented by C0, C1….C5 are shown. The register allocation of storage variables (as shown in Table I) has been generated using

Table 4.1 Register allocation in obfuscated CIG of 4-point DFT (before implantation of chromosomal DNA)

CS	Red	L	B	O	Bl	P	Gr	C	Y	N	BLK	G	M	O
0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
1	V1	V15	--	V16	--	V17	--	V8	V9	V10	V11	V12	V13	V14
2	V18	--	--	V19	--	--	--	V8	V21	--	V22	--	V23	--
3	V20	--	--	--	--	--	--	V24	--	--	V22	--	V23	--
4	--	--	--	--	--	--	--	V24	--	--	V25	--	--	--
5	--	--	--	--	--	--	--	V26	--	--	--	--	--	--

Table 4.2 Register allocation in obfuscated CIG of 4-point DFT (after implantation of chromosomal DNA)

CS	Red	L	B	O	Bl	P	Gr	C	Y	N	BLK	G	M	O
0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
1	V1	V15	--	--	V16	V17	--	V8	V9	V10	V11	V12	V13	V14
2	V18	--	--	V19	--	--	--	V8	--	V21	V22	--	V23	--
3	--	--	V20	--	--	--	--	--	V24	--	V22	--	V23	--
4	--	--	--	--	--	--	--	--	V24	--	--	V25	--	--
5	--	--	--	--	--	--	--	--	V26	--	--	--	--	--

a scheduled graph based on designer-selected resource constraints two adder and three multipliers. The variables marked in red in Table 4.2 indicate that local transformations have been made to accommodate the above hardware security constraints. It is to be noted that the register variables required at the same time step cannot share the same register, as it results into conflict in timing overlap. However, the variables required at different time steps can share the same register (color). It can be observed (from Table 4.2) that there is no requirement of any extra color (register) for embedding all the above hardware security constraints into the structurally obfuscated CIG of 4-point DFT. Thus, by implanting the encoded hardware security constraints into the design as a second level of security (post-structural obfuscation), secured IPs are generated. These implanted encoded hardware security constraints (into structurally transformed IP) enable robust and seamless detective control against pirated IP versions. As the proposed approach incorporates several crucial security parameters which are capable of incapacitating an adversary to regenerate the exact secret hardware security constraints. This, therefore, hinders an adversary from evading the piracy detection process by performing the implantation of authentic security constraints into pirated IP versions (by regenerating them completely and correctly).

4.3.2. Security Properties of the Proposed Methodology Achieved through Encrypted Chromosomal DNA Impression

It is very challenging for an attacker to regenerate the encrypted DNA digital impression for evading IP piracy detection process because he/she needs to have the following secret information:

(a) Secret key (N): By considering the initial key size of 48 bit, the function for populating the size of the secret key in our technique is, 48*Z*I bits; where ‘Z’ and ‘I’ (both variables are unknown to an adversary) signifies the number of encryption rounds used in a single Feistel cipher and the number of Feistel cipher iterations required, respectively (depending on the strength of the binaries chromosomal DNA impression). Additionally, binaries chromosomal DNA impression strength depends on the formed chromosomal DNA

sequence, initially. For example, if the binaries chromosomal DNA impression is 128 bits, then I = 2, while if the binaries chromosomal DNA impression is 192 bits, then I = 3 and so forth. Therefore, for example, if Z= 16, C= 2, then N = 1536 bits then the key size space is 21536. Deriving an exact digital impression from this massive key space gets harder even through the brute force parser.

(b) Secret key (N) also depends on the shift variables ('n' and 'm'): The derivation of the original digital impression implanted into the obfuscated design can be prevented from an adversary (due to the unknown behavior of circularly shift functions of the key generation process).

(c) Chromosomal DNA sequence and size: The sequence and size of base pairs used in forming the chromosomal DNA sequence is highly challenging to precisely estimate for an adversary. Further, the order/counting of the inserted polynucleotide as well as the number of chemical elements (A, T, C, G, S) associated in a particular DNA sequence (formed with either distinct or same type of base pairs) is highly challenging for an attacker to precisely estimate.

(d) Strength of encrypted chromosomal DNA impression: The strength of encrypted chromosomal DNA impression after performing the truncation is extremely difficult for an attacker to gauge.

(e) Dual encoding rule: intricacies of employed dual encoding rules are very complex, thereby making it extremely difficult to decode.

(f) S-Box Selection: The S-box type(s) used during the substitution phase of the encryption function is difficult to precisely estimate for an attacker. As a same S-box to convert all 6-bit to 4-bit or to convert each 6-bit to 4-bit, different S-boxes may be used.

The encrypted chromosomal DNA impression-based proposed hardware security methodology exhibits the aforesaid security properties against brute-force attacks and tamper tolerance. So, the attacker cannot extract the exact design without knowing the exact DNA impression and resource configuration (adders and multipliers used in CDFG of DSP application). Further, without the knowledge of (a) to (f), regeneration of embedded digital impression is

impossible. An adversary's extracted design cannot fully match with the original design (pre-embedding).

4.4. Results and analysis

The results report the following qualitative and quantitative analysis of the proposed structural obfuscation with encrypted chromosomal DNA impression based framework: (a) very low probability of coincidence (Pc) (indicating the strength of digital evidence) for different DSP IP cores in the range of 7.59E-5 to 1.2E-1; (b) stronger tamper tolerance for different DSP IP cores in the range of 5.62E+14 to 3.40E+38; (c) negligible design cost overhead of 0.00 % for different DSP IP cores; (d) strength of obfuscation in terms of number of gates obfuscated. The experimental results of the proposed multi-level structural obfuscation and DNA impression-based security approach have been discussed and analyzed in chapter 9 of this thesis.

4.5. Summary

This chapter presented a novel approach for ensuring the security of data-intensive DSP cores against external threats of reverse engineering and IP piracy. To safeguard the design from an adversary against interpreting the actual functionality of the design and thereby causing security hazards by performing the implantation of malicious threats, multi-level structural obfuscation has been performed. Subsequently, to detect piracy, encrypted chromosomal DNA impression of IP vendor is implanted into the design during HLS. This, therefore, ensures security against both the threats of reverse engineering and piracy. Ensuring the security of DSP based IP cores against alteration of RTL description and counterfeiting threats is crucial for both SoC designer and end consumer, as these IP cores are integral part of CE systems. The presented methodology was proven to be more robust in terms of security than recent similar works while incurring zero design cost overhead.

Chapter 5

Designing Secured Reusable Convolutional IP Core in CNN using Facial Biometric based Hardware Security Approach

This chapter presents a novel methodology to design a secured custom reusable intellectual property (IP) core for the convolutional layer of the convolutional neural network (CNN). Since the reusable IP cores used in system-on-chips (SoCs) of consumer electronics (CE) systems are susceptible to the hardware threat of IP piracy/counterfeiting. Therefore, the security of the proposed convolutional layer reusable IP core against the threat of IP piracy/counterfeiting has been ensured using facial biometrics. This enables the integration of secured reusable IP cores in the SoCs of CE systems, thereby ensuring the security of end consumers. In the proposed methodology, the convolutional layer IP core is designed through high-level synthesis (HLS) process and secured by covertly embedding secret facial biometric security information of an IP vendor into the design.

CNN finds wide utility in consumer electronics applications to facilitate tasks such as image classification, image segmentation, object/curve detection, face recognition, voice analyzing, emotion detection, and so on because of their high accuracy [58]-[61]. Further, CNNs are widely used by tech giants for photo search, for their product recommendations and for automatic tagging systems. Furthermore, the usages can be found in autonomous driving, medical diagnostics and video surveillance etc. A CNN is a highly computationally intensive framework, especially the convolutional layer among its other layers such as pooling, flattening layer and fully connected layers (memory intensive). Owning to high computational intensiveness of the convolutional layer, their realization as co-processors is very crucial for image centric applications. Further, the proposed convolutional layer reusable intellectual property (IP) core can be used in several CNN based applications and in portable or wearable devices such as mobile phones, graphics processors and Internet of Things (IOT) devices, etc. However, the security of reusable IP core (from external threats such as IP forgery and IP counterfeiting) is equally important for producing secured computing and

consumer electronics (CE) systems [4], [18] to ensure the security of end consumer.

The outline of the chapter is as follows. The first section formulates the problem. The second section discusses the HLS flow for designing a secured convolutional IP core under the following subsections: importance for consumers and CE systems, background on CNN framework, overview, and process for generating scheduled data flow graph of convolutional IP. Further, the third section demonstrates the generation of a secured convolutional IP core using facial biometrics under the following subsections: facial signature generation, secure RTL datapath generation by performing the embedding of facial biometric-based encoded hardware security constraints, and challenges of the work. The fourth section demonstrates the hardware-based convolution process using the proposed reusable convolutional IP. Finally, the fifth section summarizes the chapter.

5.1. Problem Formulation

Given the behavioral description of convolution process, module library, resource constraints, along with the objective of designing custom reusable convolutional IP core in CNN and ensuring its security by enabling the robust and seamless detective control against pirated/fake versions of proposed hardware design before their integration into computing or CE systems. The detection and isolation of pirated design versions is crucial as they may contain backdoor malicious logic causing erroneous functionality of the design. The implanted security mark, therefore can be used as secret digital evidence to discern and isolate pirated design versions.

5.2. HLS Flow for Designing Secured Convolutional IP Core

The details HLS based design flow of the proposed approach for designing secure convolutional IP is discussed under the following sub-sections:

5.2.1. Importance for Consumers and CE Systems

Integration of the proposed secured customized CNN convolutional layer reusable IP core in CE systems offers the following benefits: (i) the proposed reusable IP core is capable of parallel execution of convolution process during

pixel computation (ii) proposed reusable IP core employs facial biometric-based security thereby is capable of thwarting pirated or counterfeited IPs that ensures the security of end consumers against forged components causing possible device explosion or leakage of confidential information (iv) capable of detecting curve/object without compromising spatial information at boundary pixels thereby ensuring correctness (v) end to end demonstration for feature generation through convolution process using 2-D kernels (twice loop unrolled). Further, the proposed secured customized convolutional layer IP core is capable of performing parallel execution of six-pixel computations while offering robust security in terms of detective control against counterfeited/pirated IPs at zero design overhead.

5.2.2. Background on CNN framework

The CNN framework usually takes image data (array of pixel matrix) as input and processes it for object detection or classification. The processing of CNN framework assimilates through several layers such as the convolutional layer, pooling layer, flattening layer, and fully connected (FC) layer (as shown in Fig. 5.1). Each layer is responsible for performing some tasks and thereby cumulatively performing functionalities such as image classification and object detection. The output of one layer is fed as input to the next subsequent layer. The first layer is the convolutional layer which performs convolution operation (dot product) between the kernel and receptive field of the input

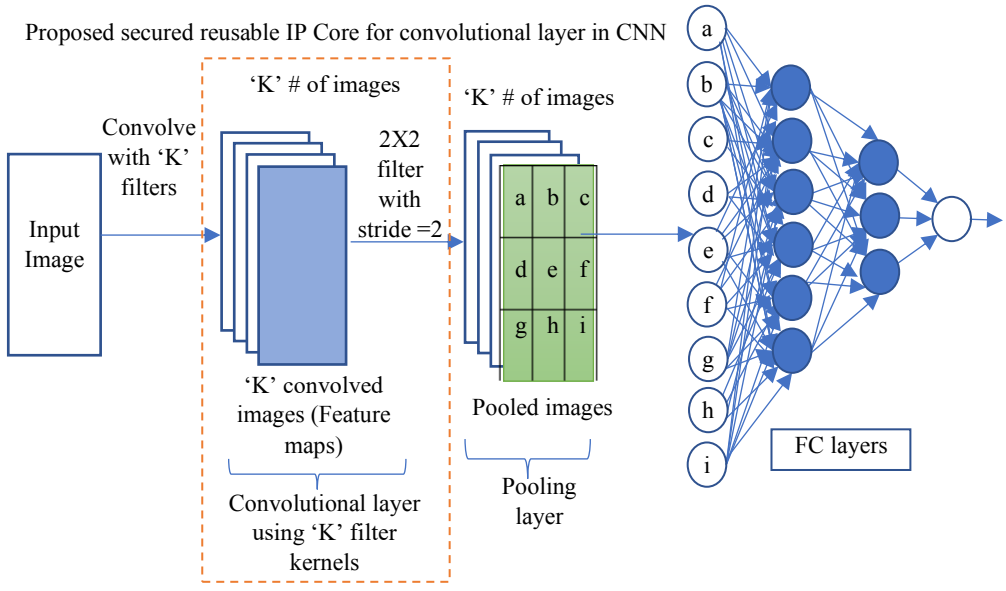


Fig. 5.1. Overview of Convolution process in CNN

image thereby generating 2-D feature maps (correspond to each kernel). Kernels/filters are used to extract the features from the input image therefore, they are also known as feature extractor. Kernel/filters (considering 2-dimensional for simplicity) are randomly generated vectors consisting of weights, which move on the input data by stride/shift value. Whereas receptive field is the portion of input image (of same dimensions to kernel) to which kernel operates. Further, the feature map (convolved image) is a response of the kernel that convolves across the input image matrix. Furthermore, features can be categorized in low level features such as edge, curves, simple colors, gradients and high-level features such as face, hand, ears as a part of a bigger object. Moreover, each filter in the convolutional layer operates on input image independently and produces the output called as feature map therefore more the kernels the better detection of visual features and patterns. Further, each conv 2-D filter is capable of identifying different features from an image depending on different weights associated with the filter. Subsequently, feature map (output matrix from the convolutional layer) is fed as input to the pooling layer. The pooling layer operates on each feature map independently. The pooling layer reduces/ down samples spatial size of representation and thus number of parameters and computation, however maintains the original shape. The pooling layer is responsible for minimizing the computation of fully connected layer. After the final convolutional layer and pooling layer, the output feature map will be converted into vectors (1-D array) called flatten layer/ feature vector. The output of the flatten layer is fed to the fully connected layer where all the features are collectively transferred into this network. Finally, predicted outputs by the network are converted into probabilistic values (corresponding to particular feature) by employing either logistic or soft max function.

In the literature, Kyriakos *et al*. [63] presented a field-programmable gate array (FPGA) based hardware accelerator for the CNN framework, which uses parallel computation at both convolutional and fully connected layer. Tsiktsiris *et al*. [64] presented an FPGA design as portable USB accelerator which implements the grayscale and Sobel edge detection. Liu *et al*. [65] presented throughput optimized FPGA accelerator for deep CNNs that maximizes

accelerator throughput by searching for optimal solution through design space exploration (DSE) algorithm. Shen *et al*. [66] presented accelerator generator which takes a CNN model and FPGA specification as input and generates optimized CNN accelerator RTL designs. All the aforesaid approaches [63]-[66] presented FPGA based solutions for mapping CNN framework on FPGA platforms. However, these approaches did not target mainly convolutional layer which is highly computationally intensive layer among other CNN layers. Further, these approaches did not ensure the security of CNN hardware designs against the threat of IP counterfeiting. On the contrary, proposed approach presents secured customized reusable IP core for the convolutional layer in CNN with robust security and zero design overhead.

Srivastava *et al*. [67] and Bai *et al*. [68] discussed a pipeline architecture for depth-wise convolution instead of standard convolution for optimizing the convolutional computation but the accuracy is not preserved. Chang *et al*. [69] presented hardware accelerator for boosting convolutional computation in image classification applications but it requires more processing resources for convolution. Ma *et al*. [70] optimized the convolutional operation based on multiple design variables to accelerate deep neural networks on FPGA. Guo *et al*. [71] presented flexible hardware architecture and network quantization methodology and a compiler program that bridges the gap between them. Kim *et al*. [72] proposed hierarchical convolution computation algorithm that is capable of reducing number of multiply-accumulate (MAC) operations but at the cost of accuracy. Further, it does not present custom hardware for the convolutional layer and only renders the efficiency for smaller feature maps.

The related approaches are either based on the FPGA based solutions [63]-[66] or targeted reducing the computational complexity of the convolutional layer in CNN [67]-[72]. However, these approaches did not provide a secured customized solution for the convolutional layer in CNN. On the contrary, the proposed approach differs from the related approaches because of presenting the following novel contributions: (i) a novel HLS design methodology is presented to design the custom reusable IP core for the convolutional layer of CNN (ii) security against the counterfeiting threat for CNN IP core is presented by enabling detection of counterfeit or pirated cores to thwart their

integration in SoCs of CE systems, thus ensuring safe usage to the consumers (iii) a facial biometric based detection is employed to detect counterfeiting of the proposed customized reusable IP core for the convolutional layer of CNN.

Further, in order to obtain the new feature map, the input is convolved with a learned kernel and then an element-wise nonlinear activation function is applied on the convolved outputs (activation maps). However, the complete feature maps are generated by convolving the 'K' kernels on the input image. The CNN takes the input in the form of image matrix (to be convolved with filter kernel for feature extraction) and perform the detection of image features as the output functionality of CNN. The input output relationship of the CNN network is discussed below:

$$O_y = \sum_{y=0}^{y=v} \left[\sum_{\substack{\mathrm{M,m} = lower\ value \\ \mathrm{M,m} = upper\ value}} \left(\sum_{\substack{\mathrm{N,n} = lower\ value \\ \mathrm{N,n} = upper\ value}} \mathrm{I_{MN}} \times \mathrm{H_{mn}} \right) \right] \tag{5.1}$$

Where, '$\mathrm{I_{MN}}$'and '$\mathrm{H_{mn}}$' represents the input image of size MxN and kernal of size mxn respectively. O_y denotes the output value of each element/pixel corresponding to output feature map; where 'v' represents the size of feature map {[(M-m+1)×(N-n+1)-1]}. Thereafter, pooling (max pooling) is performed on the convolved image (generated post performing the convolution on convolutional layer) as represented in (5.2).

$$y_{i,j,k} = pool(O_y) \tag{5.2}$$

Where, '$y_{i,j,k}$' represents the output matrix values of pooling layer. The objective of pooling is to down sample the spatial size and number of computation parameters, however maintaining the original shape. However, after multiple convolutional and pooling layers, there may be one or more fully connected layer. In order to generate global semantic information, they connect all neurons of previous layer to every single neuron of current layer. Now, the objective function at the pooling layer is derived by substituting (5.1) into (5.2), we get equation (5.3) below:

$$y_{i,j,k} = pool\left(\sum_{y=0}^{y=v} \left[\sum_{\substack{\mathrm{M,m} = lower\ value \\ \mathrm{M,m} = upper\ value}} \left(\sum_{\substack{\mathrm{N,n} = lower\ value \\ \mathrm{N,n} = upper\ value}} \mathrm{I_{MN}} \times \mathrm{H_{mn}} \right) \right] \right) \tag{5.3}$$

Next step is applying leaky ReLU activation function which accepts the output of pooling and is given below by equation (5.4).

$$a_{i,j,k} = max\ (y_{i,j,k}, 0.01 * y_{i,j,k}) \tag{5.4}$$

Where, '$a_{i,j,k}$' represents the output of ReLU. ReLU is used to improve the performance of network. The leaky ReLU activation function can be derived by substituting (5.3) in (5.4) we get equation (5.5) as shown below:

$$a_{i,j,k} = \max\ (pool\left(\begin{matrix}\mathrm{y=v}\\ \\ \mathrm{y=0}\end{matrix}\left[\begin{matrix}\mathrm{M,m} = upper\\ value\\ \mathrm{M,m} = lower\\ value\end{matrix}\left(\begin{matrix}\mathrm{N,n} = upper\\ value\\ \sum_{\mathrm{N,n}} lower\\ value\end{matrix} \mathrm{I_{MN}} \times \mathrm{H_{mn}}\right)\right]\right),$$

$$0.01 * pool\left(\begin{matrix}\mathrm{y=v}\\ \\ \mathrm{y=0}\end{matrix}\left[\begin{matrix}\mathrm{M,m} = upper\\ value\\ \mathrm{M,m} = lower\\ value\end{matrix}\left(\begin{matrix}\mathrm{N,n} = upper\\ value\\ \sum_{\mathrm{N,n}} lower\\ value\end{matrix} \mathrm{I_{MN}} \times \mathrm{H_{mn}}\right)\right]\right)) \tag{5.5}$$

Leaky ReLU function is an improved version of the ReLU activation function. As for the normal ReLU activation function, the gradient is zero for all the values of inputs that are less than zero, which would deactivate the neurons in that region and may cause 'dying ReLU' problem. A ReLU is dead if it gets stuck in the negative thereby it always results zero as its output. Once a neuron gets negative, it is unlikely for it to recover and participate in the process of discriminating the input. Leaky ReLU is defined to address this problem. Instead of defining the ReLU activation function as zero for negative values of intensities of input image, it is defined as an extremely small linear component of input values. The Leaky ReLU sacrifices hard-zero sparsity for a gradient which is potentially more robust during optimization. Leaky ReLU provides the advantage of not worrying about the initialization of neural networks. Additionally, for leaky ReLU gradient descent will be having a non-zero value always and it will continue learning without reaching dead end. Therefore, leaky ReLU performs better than ReLU [70]-[73].

The next step is to feed the output of ReLU in the input of the activation function of the fully connected layer given by equation (5.6) below:

$$Q_{i,j,k} = f(\sum w_k . a_{i,j,k} + b_k) \tag{5.6}$$

Now, by substituting (5.5) in (5.6) we get equation (5.7):

$$Q_{i,j,k} = f(\sum w_k \,.\, max\left(pool\left(\sum_{y=0}^{y=v} \left[\sum_{\substack{M,m = lower\\ value}}^{\substack{M,m = upper\\ value}} \left(\sum_{\substack{N,n = lower\\ value}}^{\substack{N,n = upper\\ value}} I_{MN} \times H_{mn} \right) \right] \right) \right),$$

$$0.01 * pool\left(\sum_{y=0}^{y=v} \left[\sum_{\substack{M,m = lower\\ value}}^{\substack{M,m = upper\\ value}} \left(\sum_{\substack{N,n = lower\\ value}}^{\substack{N,n = upper\\ value}} I_{MN} \times H_{mn} \right) \right] \right) \right) + b_k) \qquad (5.7)$$

Where, 'w_k' and 'b_k' are the weight vector and bias of the K^{th} kernel. Moreover, the weight vectors are shared such that it reduces the complexity and make the network easier to train. As evident from (5.7), it captures the input output relationship of whole CNN network.

5.2.3. Overview

The proposed methodology generates a secured reusable IP core design architecture for convolutional layer, employed in CNN. In the entire CNN framework, the most computationally intensive layer is convolutional layer. Therefore, designing the reusable IP core for convolutional layer is relevant. Further, the designed IP core may be susceptible to the hardware threat of IP piracy due to involvement of untrustworthy third-party IP vendors during the design process. Therefore, the security of IP core is also crucial to ensure the isolation of pirated design versions and allowing the integration of only authentic versions into SoCs. This ensures the correct functionality of the design. Therefore, the proposed design architecture is secured with facial biometric and is capable of computing two-pixel values (corresponding to one kernel) of the feature map (convolved image) in parallel corresponding to the input image. As shown in Fig. 5.1, in the proposed secured reusable IP core of the convolutional layer used in CNN, input matrix/image (containing complex features) is convolved with 'K=3' filters and hence generating 'K' 2-D convolved images/feature maps (one feature map corresponding to one kernel independently).

Further, a 2×2 size pooling filter with stride 2 is processed over each feature map separately to obtain pooled images (K, corresponding to each feature map). Next, the output from the pooling layer (reduced matrix based on max-

pooling) is fed to FC layer. Finally, the object containing particular curve is being detected/observed (pixel values corresponding to curve are higher/non-zero, which represents the detection of object).

5.2.4. Process for Generating Scheduled Data Flow Graph of Convolutional IP

Suppose an input image is of size P×Q (size of the input matrix is P×Q and is denoted by [I]) where each pixel value is denoted by A_{ij} (i and j varying from 0 to P-1 and Q-1 respectively).

$$[I]=\begin{pmatrix} A_{00} & A_{01} & A_{02} & \cdots & A_{0(Q-1)} \\ A_{10} & A_{11} & A_{12} & \cdots & A_{1(Q-1)} \\ A_{20} & A_{21} & A_{22} & \cdots & A_{2(Q-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{(P-1)0} & A_{(P-1)1} & A_{(P-1)2} & \cdots & A_{(P-1)(Q-1)} \end{pmatrix}_{P\times Q}$$

Where, 'A' represents the intensity value corresponding to the pixels of input image. Further, a generic kernel/filter matrix of size m×n is denoted by$[H]_{mxn}$. In case of 3×3 size filters for curve detection, three kernel matrices [H] of size 3×3 is represented as follows:

$$[H_1]=\begin{pmatrix} h^1_{00} & h^1_{01} & h^1_{02} \\ h^1_{10} & h^1_{11} & h^1_{12} \\ h^1_{20} & h^1_{21} & h^1_{22} \end{pmatrix}_{3\times3} \quad [H_2]=\begin{pmatrix} h^2_{00} & h^2_{01} & h^2_{02} \\ h^2_{10} & h^2_{11} & h^2_{12} \\ h^2_{20} & h^2_{21} & h^2_{22} \end{pmatrix}_{3\times3} \quad [H_3]=\begin{pmatrix} h^3_{00} & h^3_{01} & h^3_{02} \\ h^3_{10} & h^3_{11} & h^3_{12} \\ h^3_{20} & h^3_{21} & h^3_{22} \end{pmatrix}_{3\times3}$$

Where, $[H_1]$, $[H_2]$ and $[H_3]$ represent the curve detection kernels/filters. Further, pixel values of the kernel are represented by h^t_{pq}Where 'p, q' varies from 0 to 2 and 't' denotes kernel/filter number.

In the proposed convolutional layer IP core methodology, 'same convolution' is performed. In order to perform the same convolution', the size of the input matrix is augmented by adding zero rows and zero columns based on the following rule:

$$D=\frac{(S-1)}{2} \tag{5.8}$$

Where, 'S' is the size of the kernel, i.e., S=3 for 3×3 kernels and 'D' is the number of zero rows/columns to be added on each side of the input matrix

(top, bottom, left and right). Therefore, the post-padding size of the input matrix is increased by 2, as shown below:

$$[I]=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{00} & A_{01} & A_{02} & \cdots & A_{0(Q-1)} & 0 \\ 0 & A_{10} & A_{11} & A_{12} & \cdots & A_{1(Q-1)} & 0 \\ 0 & A_{20} & A_{21} & A_{22} & \cdots & A_{2(Q-1)} & 0 \\ 0 & \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & A_{(P-1)0} & A_{(P-1)1} & A_{(P-1)2} & \cdots & A_{(P-1)(Q-1)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}_{M\times N}$$

Where, 'M×N' is the dimension of the augmented input matrix which is equal of size (P+2)×(Q+2). Further, a generic representation of the augmented matrix post applying padding using (14) is shown below:

$$[I]=\begin{pmatrix} I_{00} & I_{01} & I_{02} & \cdots & I_{0(N-1)} \\ I_{10} & I_{11} & I_{12} & \cdots & I_{1(N-1)} \\ I_{20} & I_{21} & I_{22} & \cdots & I_{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_{(M-1)0} & I_{(M-1)1} & I_{(M-1)2} & \cdots & I_{(M-1)(N-1)} \end{pmatrix}_{M\times N}$$

Pixel values from this matrix are denoted by I_{uv}, where 'u' and 'v' vary from 0 to M-1 and N-1, respectively. For an input matrix (augmented) of size M×N, and for 'K' filters of size m×n, size of the feature map can be calculated using the following equation:

$$\left[(M-m+1)\times(N-n+1)\right]\times K \tag{5.9}$$

Further, the output matrix of the same convolution between the input matrix and kernel matrix is denoted by [O], whose dimensions are the same as that of the input matrix pre-padding (i.e., P×Q). Output pixel values of 2-D convolution are denoted by O_y^z, where 'y' varies from 0 to [(M-m+1)×(N-n+1)-1] and 'z' represents the number of output feature map corresponding to the kernel.

Output value of each element/pixel corresponding to output feature map is denoted by O_y and is evaluated as follows:

$$O_y=\sum_{M,\,m=\text{lower value}}^{M,\,m=\text{upper value}}\left(\sum_{N,\,n=\text{lower value}}^{N,\,n=\text{upper value}} I_{MN}\times H_{mn}\right) \tag{5.10}$$

In the proposed approach two sliding window of kernel matrix simultaneously convolves over input matrix to compute two-pixel outputs in parallel. Two-pixel outputs are computed as follows:

$$1^{st}\,\text{output}: O_0 = \sum_{\substack{M=0\\m=2}}^{\substack{M=2\\m=2}} \left(\sum_{\substack{N=0\\n=0}}^{\substack{N=2\\n=2}} I_{MN} \times H_{mn} \right) \qquad 2^{nd}\,\text{output}: O_1 = \sum_{\substack{M=1\\m=2}}^{\substack{M=3\\m=2}} \left(\sum_{\substack{N=0\\n=0}}^{\substack{N=2\\n=2}} I_{MN} \times H_{mn} \right) \tag{5.11}$$

By expanding the equation (5.11) to compute both output pixel values (assuming for kernel 1) will be calculated as:

$$\begin{aligned} O_0^1 = &\left[\left(I_{00} \times h_{00}^1 \right) + \left(I_{01} \times h_{01}^1 \right) + \left(I_{02} \times h_{02}^1 \right) \right] + \\ &\left[\left(I_{10} \times h_{10}^1 \right) + \left(I_{11} \times h_{11}^1 \right) + \left(I_{12} \times h_{12}^1 \right) \right] + \\ &\left[\left(I_{20} \times h_{20}^1 \right) + \left(I_{21} \times h_{21}^1 \right) + \left(I_{22} \times h_{22}^1 \right) \right] \\ O_1^1 = &\left[\left(I_{01} \times h_{00}^1 \right) + \left(I_{02} \times h_{01}^1 \right) + \left(I_{03} \times h_{02}^1 \right) \right] + \\ &\left[\left(I_{11} \times h_{10}^1 \right) + \left(I_{12} \times h_{11}^1 \right) + \left(I_{13} \times h_{12}^1 \right) \right] + \\ &\left[\left(I_{21} \times h_{20}^1 \right) + \left(I_{22} \times h_{21}^1 \right) + \left(I_{23} \times h_{22}^1 \right) \right] \end{aligned} \tag{5.12}$$

Where, each product term in (5.12) is represented as ($I_{ab} \times h_{pq}^t$); where each pixel value in the input matrix and each kernel value in kernel matrix is represented by I_{ab} and h_{pq}^t respectively. During the computation of the first two-pixel values O_0^1 and O_1^1 using 3×3 kernel, values of 'a' and 'p' varies from 0 to 2. Further, it is shown that the subscript 'b' of the value I_{ab} is varied from 0 to 2 for pixel output $O_0^1$and varies from 1 to 3 for pixel output O_1^1. Subsequently, in the remaining computations of the same row, maximum value of 'b' can go upto N-1. However, the value of 'a' varies from 0 to 2 in the first row of the output matrix. Subsequently, for computing output value of the next row of the output matrix, lower and upper values of 'a' are increased by 1. Subsequently, in the remaining computations of the output, the maximum value of 'a' can go upto M-1. Based on (5.12), data flow graph (DFG) of proposed convolutional layer IP core corresponding to each kernel is prepared that compute two output value in parallel, as shown in Fig. 5.2.

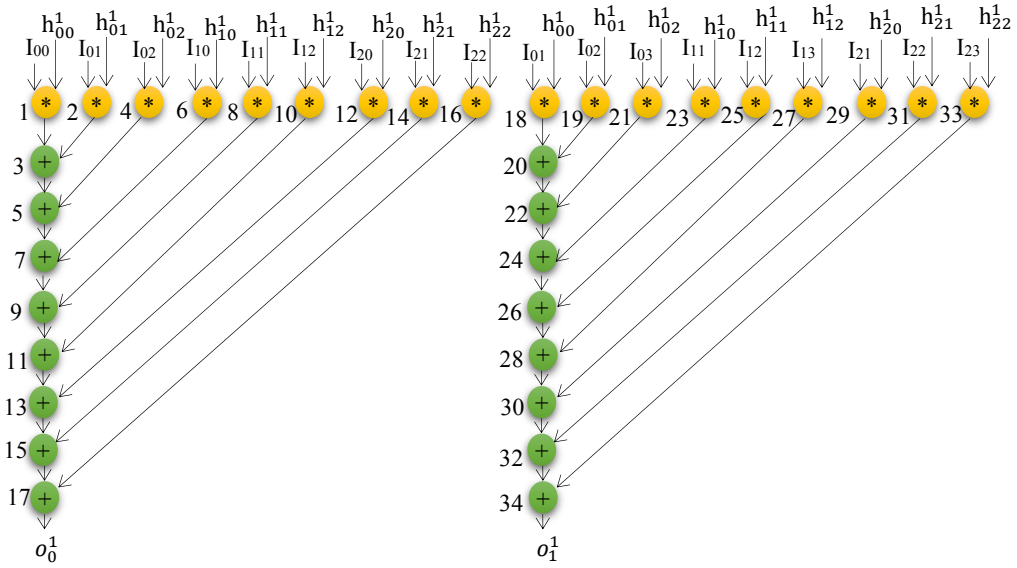


Fig. 5.2. Data flow graph (DFG) of proposed reusable IP core with filter kernel of size 3x3 and UF=2

Furthermore, total six pixels are being computed in one execution (two pixels corresponding to each of three kernels).

5.3. Demonstration on Generating Secured Convolutional IP Datapath Design using Facial Biometric

Subsequently, the DFG is fed as input to the HLS process to derive register transfer level (RTL) datapath design of proposed convolutional layer IP core. Total six data paths are prepared. In the proposed approach the basic steps of the HLS are inspired from [12], while the security-based HLS steps has been performed using our custom-designed publicly available tool called 'faciometric hardware security tool' [85]. The overall HLS flow of the proposed approach for designing secured convolutional layer in CNN is shown in Fig. 5.3. As evident in Fig. 5.3, in the proposed HLS flow following are the steps to implement the proposed methodology in HLS and design the corresponding six datapaths:

a) first, derive the DFG of the convolutional layer using the feature map generation process (by performing convolution between input image matrix and several kernel matrices), followed by a mathematical description of the convolution operation using a parallel sliding window.

b) input facial biometrics and design space exploration (DSE) parameters are fed in the proposed design flow in order to generate the facial signature and

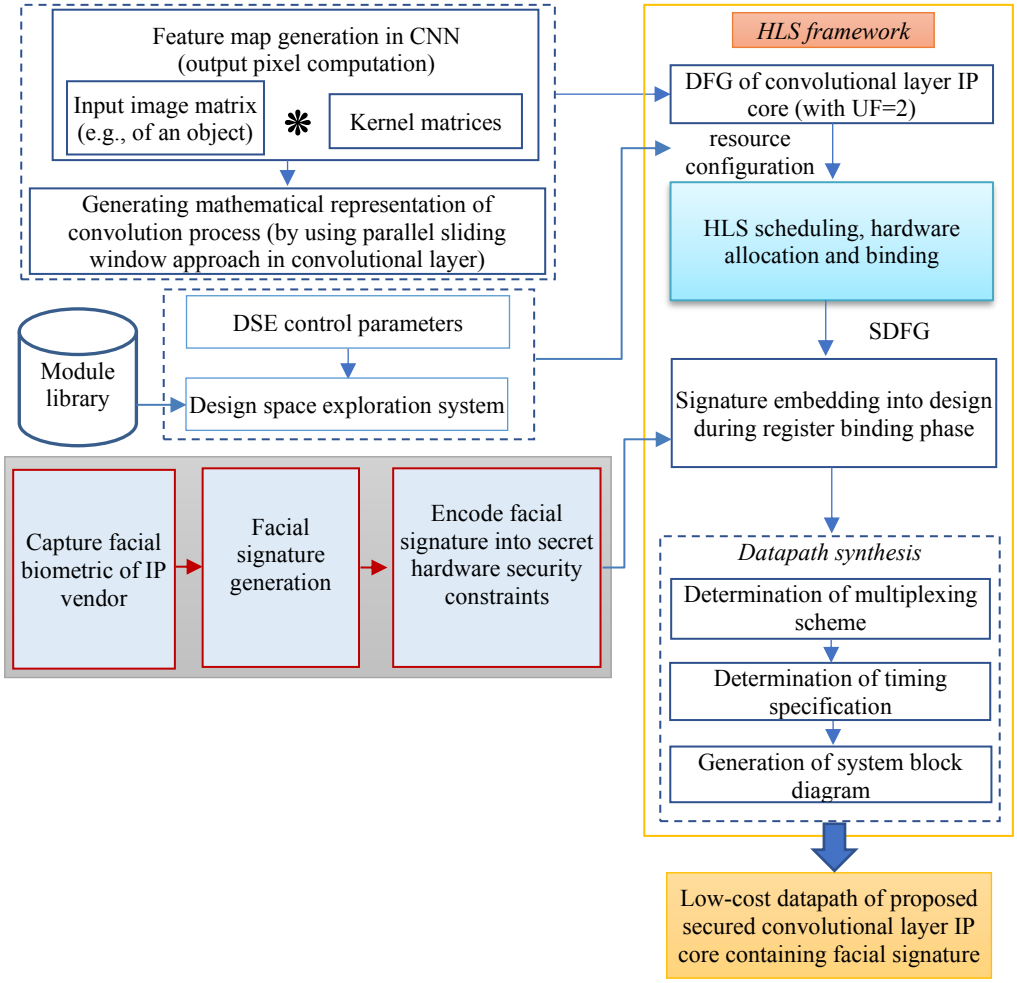


Fig. 5.3. HLS flow of the proposed approach for designing secured convolutional layer IP core in CNN

perform design space exploration respectively. *Note*: the details of the facial signature generation are shown in Fig. 5.4 and Fig. 5.5.

c) the design space exploration produces low-cost resource configuration which is used to perform scheduling of the DFG based on LIST scheduling algorithm. Subsequently, the allocation and binding of resources is performed. For example, the resource configuration used for performing scheduling is assumed as one multiplier (M) and one adder (A). The respective scheduled data flow graph (SDFG) of the convolutional layer IP core is shown in Fig. 5.6. From the scheduled data flow graph, the corresponding register allocation table is constructed that comprises of the storage variables and its respective allocation to different registers. The demonstration of constructing the register allocation table from the scheduled data flow graph is explained in section D.

d) subsequently, the facial signature is converted into respective hardware security constraints using an encoding rule. These security constraints are

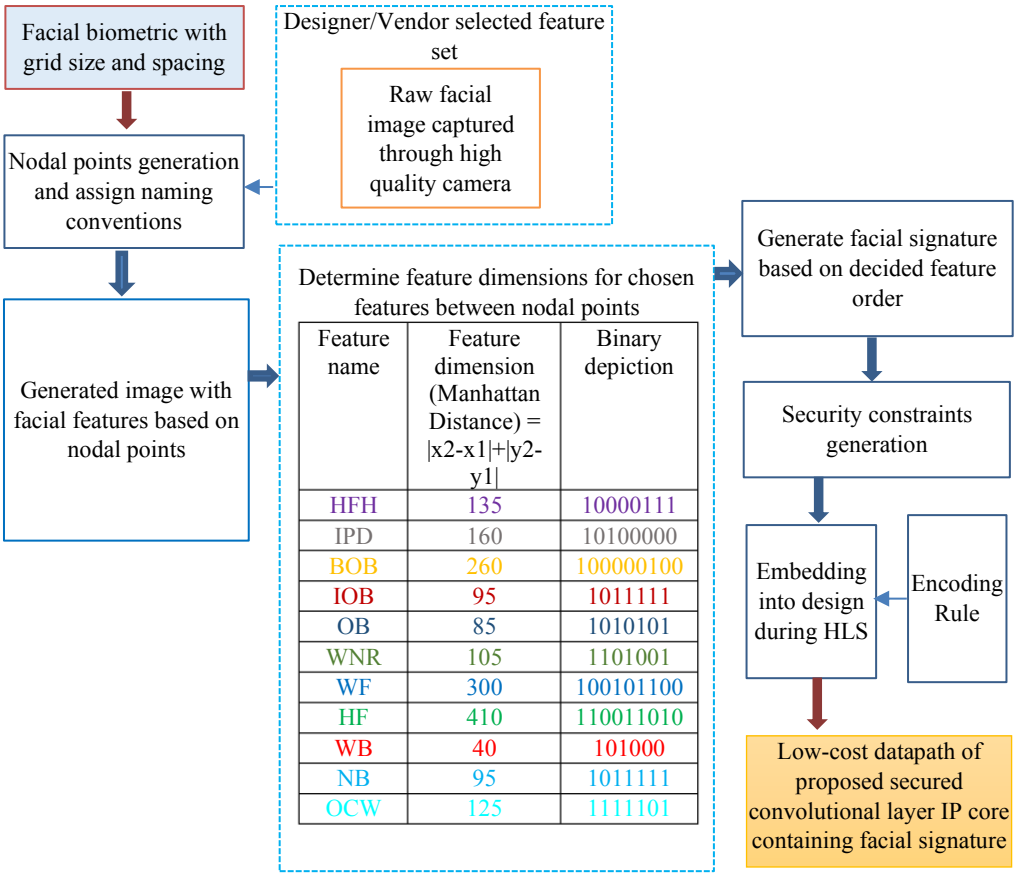


Fig. 5.4. Details of facial biometric approach for securing convolutional layer IP core

embedded in the obtained register allocation table. As a result, the security-embedded register allocation table is generated.

e) the next step is to perform datapath synthesis with the aid of determining the multiplexing scheme of each functional resource as well as the registers (obtained in the security-embedded register allocation table). Followed by determining the timing specification and development of the secured datapath of the convolutional layer IP core.

5.3.1. Facial Signature Generation

The details of the facial biometric hardware security approach integrated with the overall HLS flow is shown in Fig. 5.4. Following are the major steps:

a) capturing the facial biometric of the authentic IP vendor and representing its corresponding image with grid size and spacing.

b) designating nodal points and assigning naming conventions based on the vendor selected feature set.

c) generating an image with facial features selected by IP vendor, as shown in Fig. 5.5.

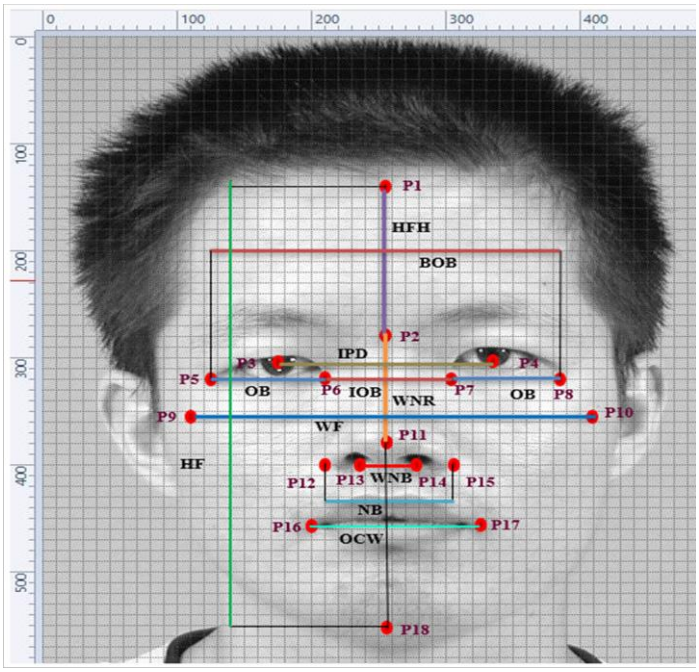
d) determining feature dimensions for the chosen features between nodal points, followed by converting the respective feature dimension into binary format.

e) generating facial signature based on the IP vendor decided feature order.

f) generating the security constraints using the encoding rule, followed by embedding the generated security constraints into register allocation table of the HLS design flow. This results into facial signature implanted RTL design.

The demonstration of securing IP core through facial biometrics is shown below:

At first, the facial image is placed on a specific grid size, thereby eliminating



HFH = Height of Forehead (P1 – P2)	WNR = Width of Nasal Ridge (P2 – P11)
IPD= Inter Pupillary Distance (P3-P4)	WF = Width of face (P9 – P10)
BOB = Bio- Ocular Breadth (P5 – P8)	HF = Height of Face (P1 – P18)
IOB = Inter – Ocular Breadth (P6 – P7)	WNB = Width of Nasal Base (P13 – P14)
OB = Ocular Breadth [(P5 – P6) or (P7 – P8)]	NB = Nasal Breadth (P12 – P15)
OCW = Oral Commissure Width (P16 – P17)	

Fig. 5.5. Generated image with facial features based on nodal points

the effect of face shift/movement. Subsequently, facial features are decided by the IP vendor to be converted into facial signature. Each feature is marked using nodal points and their naming conventions are made. Further, the co-ordinates of each feature points and their corresponding feature dimensions are obtained using Manhattan distance. Subsequently, feature dimensions are converted into binary. Finally, the facial signature is derived by concatenating the binary information of each feature (numerous ways to ordering of features are possible before concatenation). For the facial image shown in Fig. 5.5, the generated facial signature is based on a chosen concatenation order as shown below:

(HFH) &(IPD) &(BOB) &(IOB) &(OB) &(WNR) &(WF) &(HF) &(WNB) &(NB) &(OCW)

The corresponding facial signature corresponding to the above concatenation order is shown below (refer to the table in Table 5.1):

10000111101000001000001001011111101010111010011001011001100110101010001011111111101.

In the generated facial signature, total number of bits is 84 (#0s=39 and #1s =45). Further, the proposed convolutional layer kernel design is having 3 filters/kernels (K=3) and each of the kernel is unrolled twice (UF=2). Therefore, total 6 data paths are formed and correspondingly facial signature is also bifurcated into 6 parts (14 bits each). Decision rule for deciding which portion of the signature (signature bits) is to be implanted in which datapath is

Table 5.1 Decision rule (embedding of a specific 14-bit long signature part into a particular datapath of k^{th} kernel is shown using color mapping)

Signature (14 bits each)	Even-odd representation based on signature part number	Decision rule: {datapath number of k^{th} kernel + (unrolling datapath #1, unrolling datapath #2)}	Datapath number of respective kernel (obtained using corresponding SDFG) in which facial signature is implanted
10000111101000	Part(1→odd),	{1+(1,2)} = {2,3};	kernel-1 and unrolling datapath #1
00100000100101	Part(2→even)	2→even, 3→odd	kernel-1 and unrolling datapath #2
11111010101110	Part(3→odd),	{2+(1,2)} = {3,4};	kernel-2 and unrolling datapath #1
10011001011001	Part(4→even)	3→odd, 4→even	kernel-2 and unrolling datapath #2
10011010101000	Part(5→odd),	{3+(1,2)} = {4,5};	kernel-3 and unrolling datapath #1
10111111111101	Part(6→even)	4→even, 5→odd	kernel-3 and unrolling datapath #2

shown in Table 5.1. Based on the above decision rule, signature part-1 is selected to be implanted into datapath number 2 of kernel-1. Subsequently, the remaining signature parts are also implanted in the corresponding convolutional layer reusable IP core datapath.

Facial biometric has been integrated in the proposed approach for hardware security because of several advantages than the state-of-the-art hardware security approaches [12], [13] such as:

(i) Facial signature is generated based on naturally unique features (formed using nodal points) of an individual/IP vendor thus it is not possible to reuse and replicate the facial signature.

(ii) Even if an adversary accesses the facial signature s/he will not be able to regenerate exact signature because of several secret parameters which are not known to her/him such as: grid size/spacing, chosen features set by the designer, ordering of the features before concatenation, decision rule chosen by IP designer to generate security constraints and how the facial signature is implanted into convolutional layer kernel datapath are also unknown.

(iii) It is capable of detect pirated/counterfeited IP core versions.

(iv) Produces zero design overhead post-embedding facial signature.

(v) Yields a much lesser Pc value and higher tamper tolerance value as discussed in the results section, which is desirable. Thus, is capable of ensuring robust security.

5.3.2. Secure Datapath Generation by Performing the Embedding Facial Biometric based Encoded Hardware Security Constraints

In order to secure the convolutional IP core design against piracy, facial biometric-driven encoded hardware security constraints are covertly embedded into the design. For the sake of brevity demonstration of embedding of facial signature part-1 is shown below:

The signature part-1‘10000111101000’ contains six ones and eight zeroes. Security constraints for the above 14-bit signature are generated based on the proposed encoding rule as shown below:

- ‘0’signifies implanting a security constraint between even pairs of storage variables.
- ‘1’ signifies implanting a security constraint between odd pairs of storage variables.

Thus, generated security constraints are:

For ‘0’-bits→ <V0-V2>, <V0-V4>, <V0-V6>, <V0-V8>, <V0-V10>, <V0-V12>, <V0-V14>, <V0-V16>

For ‘1’-bits→ <V1-V3>, <V1-V5>, <V1-V7>, <V1-V9>, ……, <V1-V13>.

Based on the above encoding rule, security constraints for the other parts of signature (part-2,3,4,5,6) have been generated. Subsequently, from the scheduled data flow graph, shown in Fig. 5.6, of the convolutional layer IP

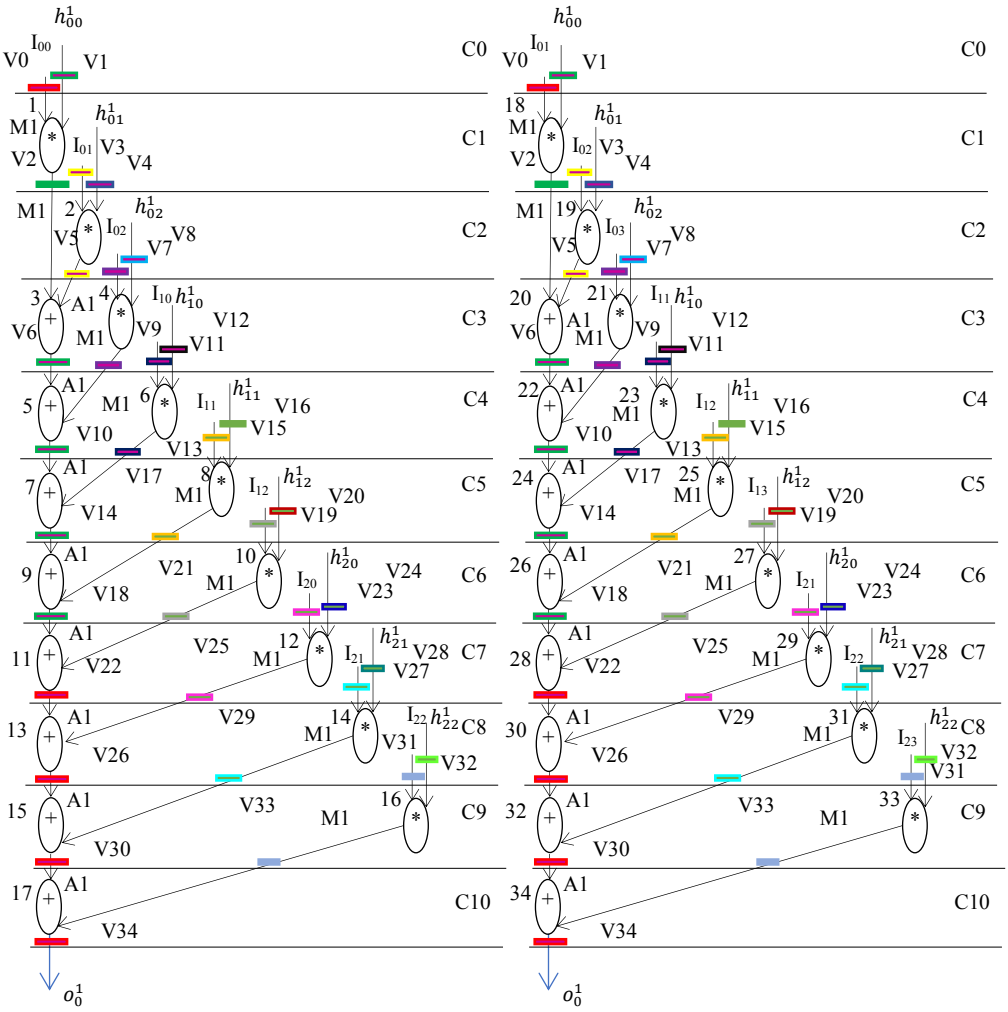


Fig. 5.6. Scheduled DFG of proposed convolutional layer IP core with kernel of size 3x3 and UF=2 based on 1M, 1A resources

core in HLS design flow, the corresponding register allocation table is constructed that comprises of the storage variables and its respective allocation to different registers. The generated security constraints are embedded in the corresponding register allocation table using the decision rule (shown in Table 5.1). The modified register allocation table containing the embedded security constraints is then constructed as shown in Table 5.2. The original assignment of storage variables into registers (pre-embedding security constraints) is highlighted in grey, while the assignment of storage variables into distinct registers (post-embedding security constraints) due to local transformations made to accommodate covert security constraints is marked in red color. Table 5.2 represents the register allocation of datapath-2 of kernel-1 where, V0 to V34 are the storage variables that are being stored in registers 'R1 to R18' during different control steps depending on their liveness (represented using different colors). Further, C0 to C10 are the number of control steps required for scheduling the secured design. Further, it is evident from the register allocation table that no extra register is required for implanting covert secret facial information. Subsequently, all datapath corresponding to each kernel are designed using the proposed HLS flow discussed earlier in section C. All the datapath circuits have been manually designed using the facial biometric based secured HLS design flow described in Fig. 5.4 and Fig. 5.3, respectively. Fig. 5.8 represents the datapath of the secured reusable IP core corresponding to signature part-1. Based on which secured convolutional layer reusable IP core data paths are designed as shown in Fig. 5.7 and Fig. 5.8 (two datapaths of first kernel for enabling the parallel computation of two output pixels).

Table 5.2 Register allocation of the proposed convolutional layer IP core (partial view post implantation)

Registers	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	-	R16	R17	R18
C0	V0	V1												
C1	V2	V2	V3	V4	--	--	--	--	--	--	-	--	--	--
C2	V2	V2	V5	--	V7	V8	--	--	--	--	-	--	--	--
C3	V6	V6	--	--	V9	--	V11	V12	--	--	-	--	--	--
C4	V10	V10	--	--	--	--	V13	--	V15	V16	-	--	--	--
C5	V14	V14	--	--	--	--	--	--	V17	--	-	--	--	--
C6	V18	--	--	--	--	--	--	--	--	--	-	--	--	--
C7	V22	--	--	--	--	--	--	--	--	--	-	V28	--	--
C8	V26	--	--	--	--	--	--	--	--	--	-	--	V31	V32
C9	V30	--	--	--	--	--	--	--	--	--	-	--	V33	--
C10	V34	--	--	--	--	--	--	--	--	--	-	--	--	--

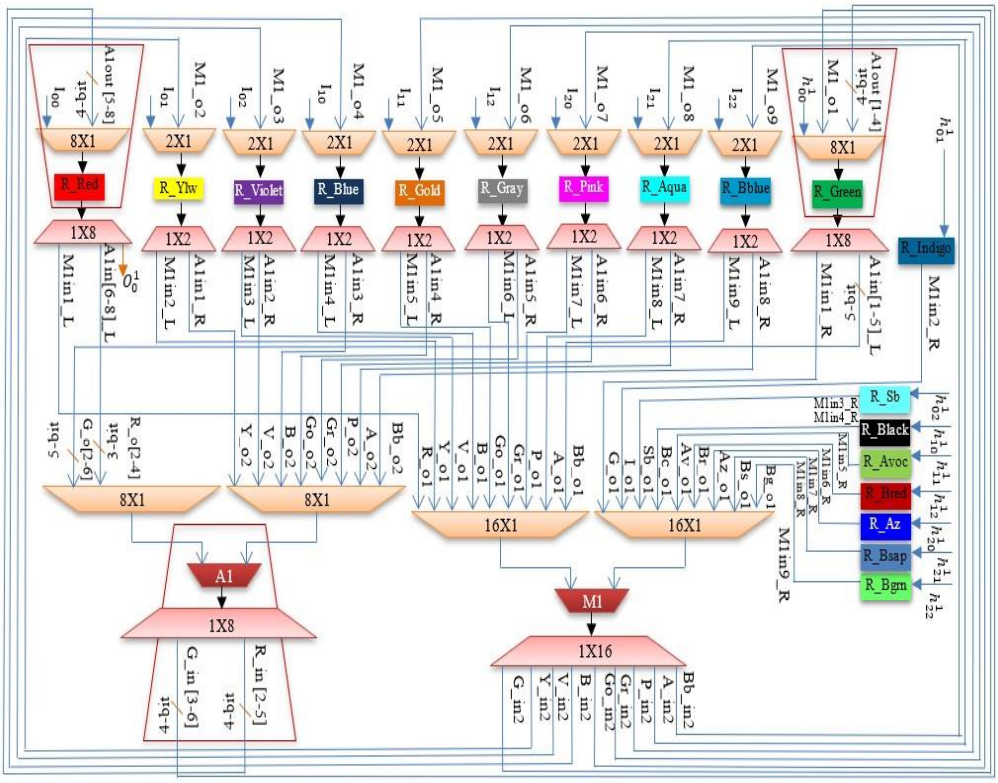


Fig. 5.7. Post-embedding facial signature, proposed secured convolution layer kernel datapath for computing first output pixel $\mathbf{O_0^1}$

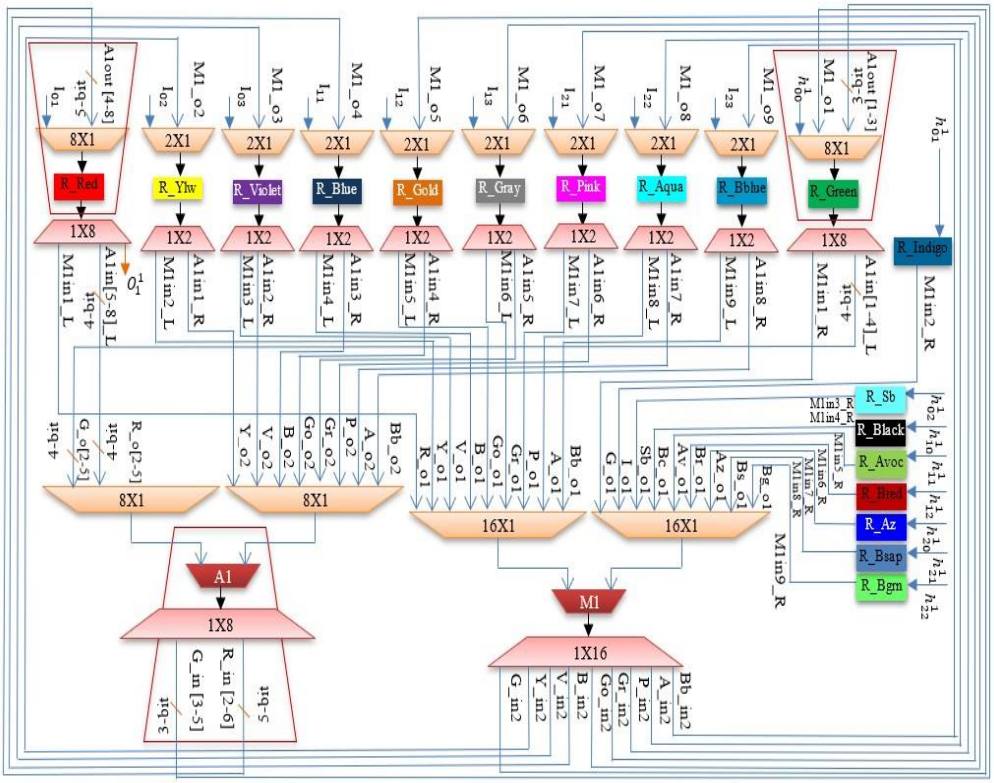


Fig. 5.8. Post-embedding facial biometric, proposed secured convolution layer kernel datapath for computing second output pixel $\mathbf{O_1^1}$

5.3.3. Challenges of the Work

Following were the challenges of the proposed approach, which were carefully considered during design and implementation:

1. The facial biometric signature generated in this approach depends on the chosen feature set by the IP vendor. Careful choice of the number of features and the type of feature selected were very important for generating robust hardware security constraints corresponding to the facial biometric signature. This is because the right choice of facial features for securing the CNN IP core impacts the robustness of the security and design cost overhead.

2. Carefully designing the encoding algorithm for converting the facial biometric signature into hardware security constraints was very important for producing a large size of number of security constraints. More the number of security constraints generated, more is the amount of digital evidence embedded in the CNN IP core for securing against IP counterfeiting. Therefore, careful designing of the encoding algorithm was very important from the perspective of security.

3. Carefully choosing the scheduling algorithm for scheduling the DFG of the convolutional layer IP core was important, as scheduling affects the design latency of the IP core which in turn affects the design cost. LIST scheduling algorithm (which is resource constraints driven) was chosen amongst other scheduling algorithms (such as ASAP, ALAP etc.) in order to integrate with the design space exploration module. This enables optimization of the final design cost of the CNN IP core.

5.4. Demonstration of Hardware-based Convolution process using Proposed Convolutional IP

The proposed approach processes the input image (in form of matrix) and convolve with three (K=3) kernels/filters in parallel using customized secured reusable IP core and thereby detecting edge/curve corresponding to each filter. Proposed approach computes two pixels in parallel corresponding to each kernel, thereby is capable of enhancing the computation process of convolutional layer. For the sake of brevity entire process has been expressed in three phases such as: convolutional layer phase, pooling layer and fully connected layer phase. As shown in Fig. 5.9(a), an input image of which

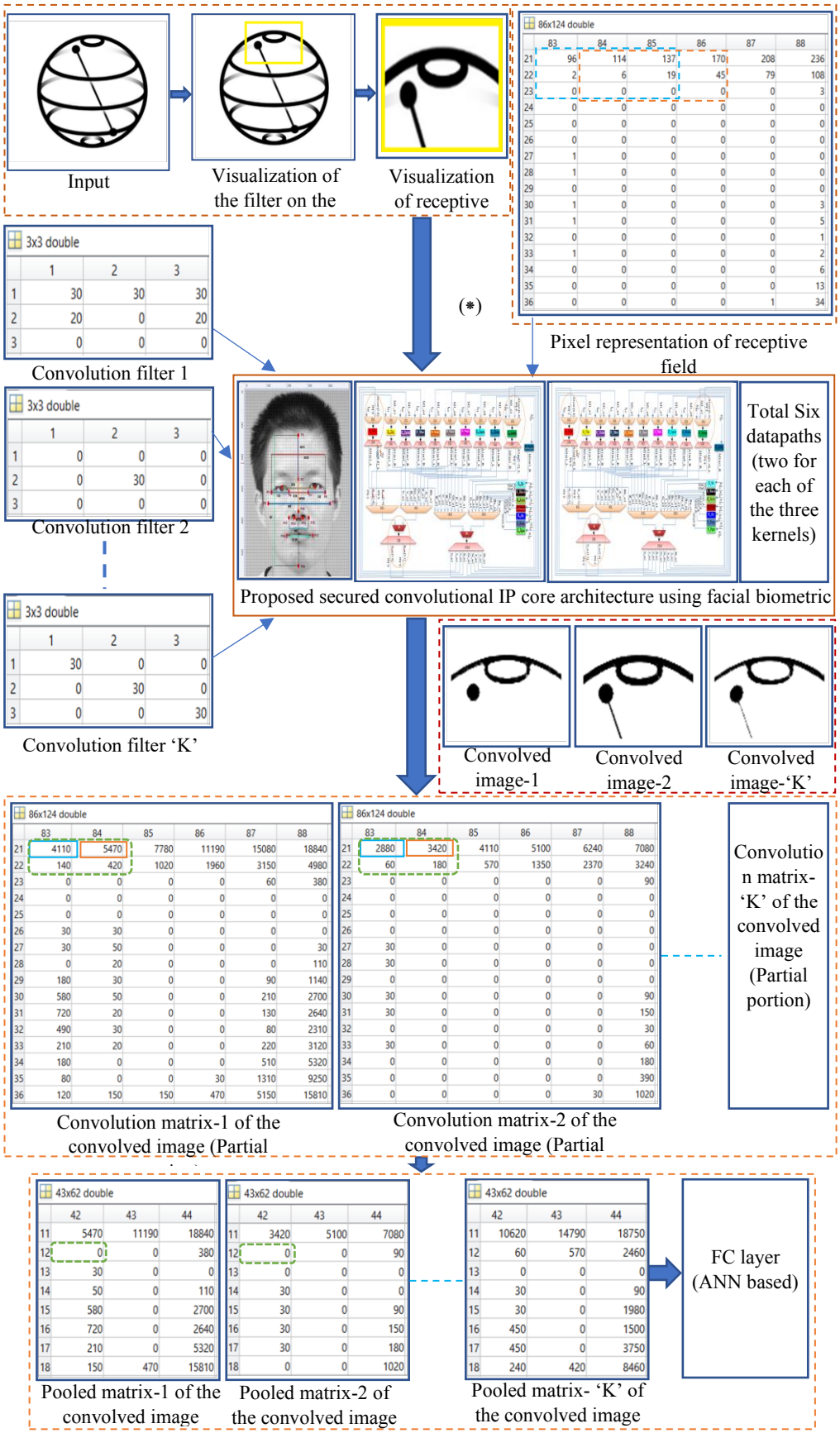


Fig. 5.9(a). Proposed approach for convolutional layer IP Core design architecture secured using facial biometric

features (specifically curve/edges) are to be detected is marked using yellow

color and is of size 86×124. The subregion on which the filter starts convolving is called as receptive field. Further, the receptive field/ subregion of input image is marked using green and brown color corresponding to the parallel computation of two output pixel values. Further, three 2-D convolutional kernels of size 3×3 are defined and weights are assigned to them. Kernel detects the shape as per the pixel orientation and their weight value. Consequently, in the input image, if there is a shape that generally resembles the curve then all the output pixel value will be non-zero/higher as a response of kernel. The output pixel value is the resultant of dot product between receptive field and the corresponding kernel. Further, it is evident from the convolution matrix of the convolved image that pixel values (represented in double datatype) are higher for the kernel responded portion. Furthermore, if the input image does not resemble or partially resembles the curve corresponding to the kernel, then the output pixel value will be zero/lesser. Therefore, if the curve lies in the input image, then the output pixel values corresponding to the output feature map will be higher. As the same convolution is applied, therefore, the size of the output matrix will also be the same as the input image matrix pre-padding. Post convolution, the convolution matrix/output matrix of the convolved image is generated corresponding to each kernel. A partial portion of the output matrix/feature map is presented in phase-2, which is also of size 86×124. Hence, feature maps corresponding to the kernel are generated independently as the output of the convolutional layer. Post-execution convolved image-1,2,3 is presented here, and the resulting images are different from each other because of different filters extracting different features.

In the next phase, each feature map is processed through the pooling layer independently. Pooling employs a 2×2 filter with stride 2 to reduce the spatial dimension of outputted feature map from the convolutional layer. Further, max-pooling is employed, which results in only maximum value from the receptive field of the convolved image corresponding to pooling. As can be observed from pooled matrix-1 of the convolved image, pooled matrix size is reduced to 43×62, and its pixel value is a maximum of 2×2 matrix marked in

green/dotted line. Subsequently, pooled matrix/images corresponding to each feature map is obtained.

Output structure of the convolved image and pooled image corresponding to different kernels (kernel-1,2,3) employed in the proposed convolutional layer reusable IP core used in CNN is presented in Fig. 5.9(b). Thus, low-level feature such as curves (if present in input image) are reported as output of feature detection process of the CNN post-processing through fully connected layer.

Furthermore, the computational complexity of CNN is not only dependent on the dimensions of the space, but also the total number of local numbers of the optimization problem such as weight vectors and bias. Further, ReLU and learning rate are additional parameters that can also impact the performance (convergence speed) of CNN network. Mathematically, the computational complexity for performing the training in CNN that is dependent on the dimensions of the space which is the total number of parameters in convolutional layer. It is the product of the number of parameters of the output activation maps (output volume) and the number of parameters of filter kernels. It is described as follows:

$$\{[((V + 2D) - S)/q + 1] * K\} * \{S * S * K\} \qquad (5.13)$$

Where, 'V' is the input volume image size, 'D' is the padding, 'S' is the size of the filter, 'q' is the stride and 'K' are the number of filter kernels. The

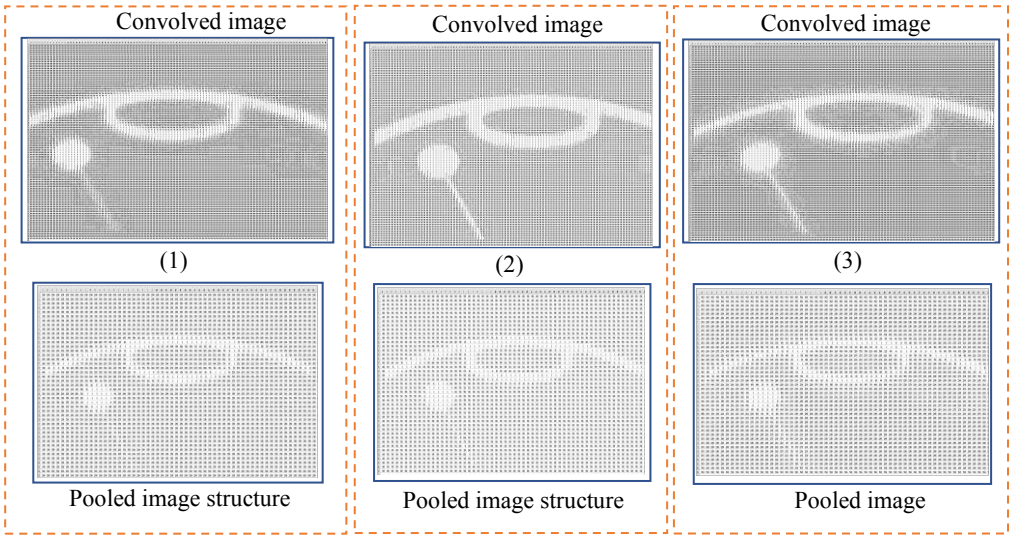


Fig. 5.9(b). Output structure (image matrix representation) of convolved image and pooled image corresponding to different filters (1,2,3) used in proposed CNN convolutional layer

approximate run time complexity for performing training varies between ~50 sec to ~148 sec.

5.5. Results and analysis

The qualitative and quantitative analysis of the proposed approach exhibits significantly lower probability of coincidence (Pc) (up to 47% less) and higher tamper tolerance (1.93E+25) than recent approaches. Further, it offers robust security with zero design overhead. The experimental results of the proposed methodology to design a secured convolutional IP core have been discussed and analyzed in detail in chapter 9 of this thesis.

5.6. Summary

This chapter presented a novel approach for designing a secured reusable CNN convolutional layer IP core using facial biometric based hardware security. The computationally intensive process of the convolutional layer has been targeted in the proposed approach. In this approach facial signature of an authentic IP vendor is implanted during HLS phase of design process to enable detective control against IP piracy/counterfeiting and minimizing the implementation complexity. This ensures the security of end consumers from unreliable and unsafe components integrated in CE systems. Therefore, the proposed approach offers both customized secured convolutional layer reusable IP core which greatly accelerates the output pixel computation process for curve detection and robust security against IP piracy/counterfeiting. The presented methodology was proven to be more robust in terms of security than recent similar works based on hardware steganography and encrypted digital signature.

Chapter 6

Retinal Biometric based Secured JPEG-Codec Hardware IP core design for CE systems using HLS

This chapter presents a novel methodology for designing secured JPEG-compression-decompression (CODEC) hardware IP core using retinal biometrics. Retinal biometric security enables robust and seamless detective control against pirated IP versions before their integration into consumer electronics (CE) systems. In order to achieve robust security, retinal biometric-driven encoded hardware security constraints are embedded into the design using high-level synthesis (HLS). These embedded retinal biometric-based hardware security constraints are responsible for discerning and isolating fake/pirated IP versions. Detective control against pirated IP versions is crucial. This is because fake/pirated hardware intellectual property (IP) cores integrated in consumer electronics systems can cause reliability hazards and jeopardize the security of end consumers. The existing approaches do not provide robust security against replication or evasion of IP detection, resulting into a higher probability of coincidence (Pc) and lesser tamper tolerance (TT) compared to the proposed approach. Further, compared to existing biometric-based hardware security methodologies, the proposed approach provides more distinctive features and does not require image enhancement compared to non-biometric-based hardware security methodologies. The proposed approach provides natural uniqueness and non-replicability of features through retinal images.

Amongst the different design objectives of hardware IP cores such as: a) optimization b) security and c) reliability, the security parameter plays a very crucial role in ensuring the authenticity of the design [16], [17]. A pirated design may render an IP core vulnerable to different security threats such as counterfeiting and cloning [19]-[23]. Therefore, in order to detect IP piracy, an embedded robust secret security mark can play an important role in detecting the pirated IP cores during detection process [24], [25]. The chapter demonstrates the retinal biometric based robust hardware security approach on JPEG-codec hardware IP core. Digital signal processing (DSP) cores such as JPEG-codec is one of the computationally-intensive hardware IP cores that are

widely used in applications such as image and video compression-decompression of camera devices. The proposed approach can also be applied to any DSP and multimedia hardware IP core designs such as finite impulse response (FIR) filter, infinite impulse response (IIR) filter, discrete cosine transformation (DCT), discrete wavelet transformation (DWT) and motion picture expert group (MPEG) etc.

The outline of the chapter is as follows. The first section formulates the problem. The second section discusses the retinal biometric-based hardware security approach under following subsections: importance for consumers and CE systems, motivation and merits of retinal biometric and overview of the approach. Further, the third section demonstrates automatic detection of retinal feature points for digital template generation under following subsections: capturing retinal biometric with IP vendor-specified grid size/spacing, automatic feature extraction from retina image and generation of nodal feature points, generating retinal image with IP vendor selected feature points and generating retinal digital template. The fourth section demonstrates the design flow for generating secured JPEG-codec IP using retinal biometric under the following subsections: generating retinal biometric-based secret hardware security constraints, generating secured RTL design, detection of retinal biometric security mark into the design and security properties/parameters of retina biometric-based security methodology. Finally, the fifth section summarizes the chapter.

6.1. Problem Formulation

Given the functional description/transfer function of JPEG-codec hardware IP, along with module library, resource constraint, and retinal biometric of genuine IP vendor along to design secured reusable hardware IP using retinal biometric security against the threats of piracy. In the case of IP piracy, an adversary designer in a third-party design house may illegally pirate the IP without the knowledge and consent of the designer (original IP vendor). Therefore, it is crucial to ensure robust security against piracy threats.

6.2. Overview of Retinal Biometric based Hardware Security Approach

The retinal biometric based hardware security is discussed under the following subsections:

6.2.1. Background on retina biometrics

Among existing biometric modalities, ocular biometric traits such as retina have received significant attention in the recent past. The retina is located towards the back of the eye. Because of its internal location within the human eye, the retina is not exposed to external environmental factors, and thus, it possesses a very secure and suitable biometric. The retina is approximately 0.5mm thick and covers the inner side at the back of the human eye. In the center of the retina is the optical nerve or optical disk (OD), a circular to oval white area measuring about 2×1.5mm across. It is the blood vessel pattern in the retina that forms the foundation for retina-based authentication. Uniqueness of retina comes from the uniqueness of blood vessels pattern distribution at the top of the retina. The landmark points in the retina are the special regions (junction)) such as vessel branching, bifurcation, crossovers and vessel ending, which are classified, based on the vessel geometry [76]. The branching and bifurcation are the points where a vessel is bifurcated or split into two vessels and a new vessel formation occurs (where a minor vessel comes out from a major vessel), respectively. On the other hand, Crossover are the points where two vessels or branches of two vessels meet at a point and vessel ending are the point where vessel terminates. These landmark features extracted from retina can identify even among genetically identical twins.

6.2.2. Importance to consumer and CE systems

The reusable IP cores are an indispensable part of consumer electronics systems. Therefore, the security of such IP cores must be ensured to secure CE systems and thereby safeguard the end consumers. The proposed retinal biometric features-driven hardware security approach ensures the same through its security features. It enables the robust detective control of counterfeited IP cores through the embedded retinal biometric signature before their integration into the system on chips (SoCs) of the CE system. Thus, ensuring the integration of secured and authentic CE systems; thereby safeguards the end consumer against the usage of fake or counterfeited

designs. Counterfeited designs may contain malicious logic, which cause unreliability in respect to their functionality and also may cause security hazards to end consumers. The IP cores integrated with authentic retinal biometric signature are genuine and therefore can be used to discern between original and counterfeited versions. Therefore, the proposed approach, by enabling the seamless detection of counterfeited IP cores, impedes the integration of fake IPs in the CE systems and assures the use of only authentic designs. Thus, the proposed security methodology ensures the security of the end consumer by providing robust security to the underlying IP cores in CE systems.

6.2.3. Motivation and Merits of retinal biometric

The proposed approach overcomes the limitations of the existing biometric and non-biometric based approaches for securing hardware IP cores. More specifically the proposed retinal biometric hardware security approach offers the following benefits over other related approaches [32], [34], [37], [39]-[41], [95]:

1) Merits of retinal biometric over other biometric approaches used for hardware security such as fingerprint, palmprint and facial biometric [40], [41], [95]:

a) retinal biometrics does not depend on any external factors such as dirt and grease as retina is not exposed to external environment (is situated at the back side of the eye) thereby offering a safe biometric; whereas in case of fingerprint and facial biometric, grease, dust and several other external factors may affect the accurate biometric feature extraction process during signature generation.

b) due to higher signature strength of retinal biometric, it results into lower probability of coincidence and higher tamper tolerance as compared to facial and fingerprint biometric-based hardware security.

c) in case of retinal biometric, it is highly impossible for an adversary to capture the retinal image without the consent of an individual. Whereas, in case fingerprint biometrics, fingerprint spoofing and in the case of facial biometrics, capturing the facial image is possible for an adversary without

consent. Therefore, retinal biometrics is aptly suitable for securing the hardware IP cores as it integrates the highly robust and secured retinal signature (of authentic IP vendor with his/her consent) as compared to fingerprint and facial biometrics.

d) retinal biometric characteristics are highly distinctive (even in case of twins) as compared to facial and finger characteristics.

e) retina scans are more accurate than fingerprint-based biometric and does not require any image enhancement using fast fourier transform (FFT).

f) moreover, in contrast to handprint biometric (palmprint), proposed retinal biometric-based methodology offers more robust security strength against IP piracy due to the following:

i) retinal biometrics comprises of highly distinctive and larger number of feature points which in turn results into retinal biometric template with higher signature strength. This therefore enables the generation of larger number of secret security constraints to be embedded into the target design (ensuring robust security in terms of lower Pc and higher TT as desirable against piracy and brute force attacks, respectively).

ii) retinal biometrics cannot be captured without the absolute consent of an individual compared to palmprint biometrics during signature generation process. Therefore, due to the inherent security of retinal biometrics in terms of distinctiveness and larger biometric template, it enables more robust and seamless detective control against IP piracy than handprint biometrics.

2) Merits of retinal biometric over digital signature-based hardware security approach [39]:

a) retinal biometrics ensures the uniqueness of generated retinal signature as it comprises naturally unique retinal features. Whereas in the case of the digital signature-based approach uniqueness of generated signature is not always guaranteed, although the algorithm is more complex compared to the proposed retinal biometric approach.

b) retinal signature generation process involves the IP vendor selected unique retinal features (which cannot be replicated), whereas digital signature generation approach depends on several factors such as encoding rule, hashing algorithm, private key for RSA encryption which can be compromised by an adversary with some efforts.

c) it is highly impossible for an adversary to replicate the retinal signature. Whereas, in case of digital signature-based approach, it may be possible by exploiting the private key through brute force attack and compromising the encoding rule with some effort.

d) retinal biometric signature is non-vulnerable as it incorporates naturally unique retinal features (decided by genuine IP vendor during signature generation). On the contrary, digital signature-based approach involves key based security technique which renders it vulnerable to theft and key based attacks.

Further, the proposed approach is also effective in the following scenarios: (i) it is effective for enabling the detection of ICs with poor specs when relabeled as ones with better specs. Detection in this scenario is performed by backpropagating the IC upto the intended level of design form to trace the implanted authentic retinal biometric signature. If the ICs with better specs are secured with vendor's retinal biometric signature, then by detecting the retinal security constraints in the register transfer-level (RTL) form of IC under-test, the genuine IC can be discerned and isolated from the fake ones. (ii) helps in isolating the designs containing malicious logic before their integration into CE systems (If a rouge IP supplier has already implanted malicious logic and selling such fake IPs to the system integrators). In this scenario, proposed approach helps in discerning such fake IPs as they would not contain the genuine vendor's authentic security signature. Thus, evasion of piracy detection process is not possible from an attacker's perspective. Further, the threat model addressed in the proposed approach is equivalent to DY adversary model where the security and robustness of the system is preserved despite adversary having the capability to intercept/access the pre-stored retinal image [96]. This is because, in that case it is not possible for him to exactly regenerate the implanted encoded hardware security constraints due to

several additional security layers of the proposed system (discussed in details in section 6.4.4).

6.2.4. Overview

The proposed retinal biometric based hardware security approach enables the robust security of IP cores against the threat of piracy. It enables sturdy isolation of pirated IPs during the piracy detection process. The proposed approach integrates retinal biometrics of genuine IP vendor in order to generate secured JPEG-codec IP core (RTL datapath or soft IP core) as shown in Fig. 6.1. The proposed approach for securing JPEG-codec IP core using retinal biometric is discussed into two modules: a) retinal digital template generation module b) secured RTL datapath generation of JPEG-codec. First module is responsible for generating the secret security constraints corresponding to the true retinal signature (post-feature extraction process). In this module binarized retinal image of a specific size (region of interest) is taken as input to the feature extraction block. This results into locating the feature nodal points corresponding to bifurcation and branching on the binarized retinal image. On a vascular structure of retinal biometric image, if a vessel is bifurcated or split into two vessels (approximately of similar pixel width) then it is known as branching and if a new vessel formation occurs where a minor (smaller pixel width) vessel grows or comes out from a major (wider pixel width) vessel, it is known as bifurcation. Subsequently, this output is fed into the retinal signature generation block, from where the retinal

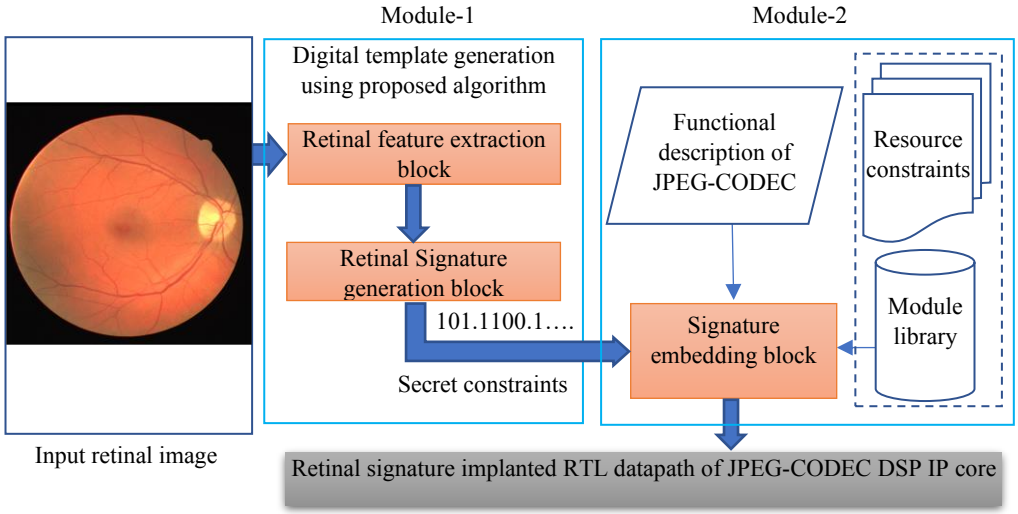


Fig. 6.1. Overview of the proposed retinal biometric based hardware security methodology

biometric based digital template is generated. Next, the corresponding covert security constraints are generated from the digital template using encoding algorithm.

The next module is responsible for embedding of the generated covert security constraints into register allocation phase of HLS framework, thereby generating the retinal signature implanted secured JPEG-codec IP core. This module accepts the following inputs: (a) library (b) resource constraints (c) functional description (high-level description or transfer function represented as DFG) of JPEG-codec (d) generated covert security constraints from the previous module. The corresponding output of this module is a retinal signature implanted robust, secured JPEG-codec IP core. The details of this module have been explained in section 6.4.2.

The flow of the proposed IP retinal biometric approach for generating a secured JPEG-codec design is shown in Fig. 6.2 and Fig. 6.3. The major steps of the proposed approach are as follows:

1) At first, in the preprocessing phase the captured retinal biometric of IP vendor/designer is used to obtain the binarized vessel structure of retina. The binarization process is used for the same.

2) The region of interest is cropped/selected from vessel structure of retinal

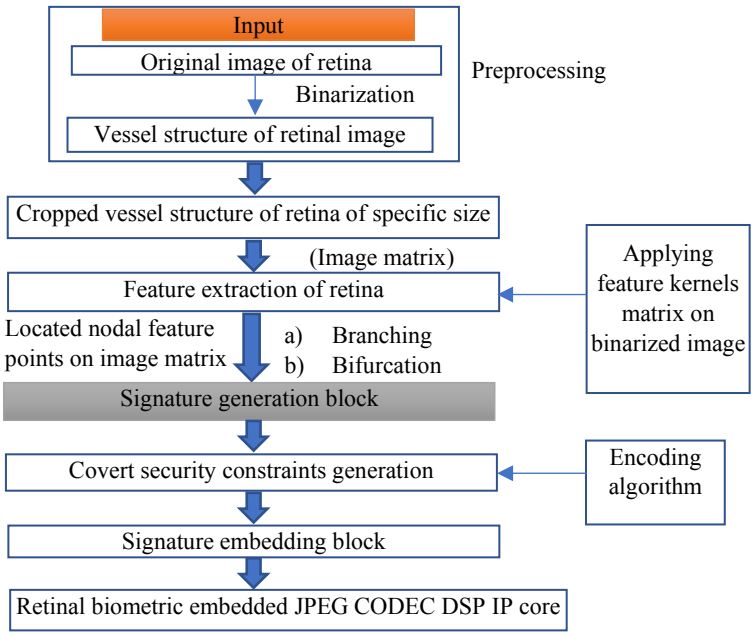


Fig. 6.2. Details of the proposed retinal biometric based security methodology

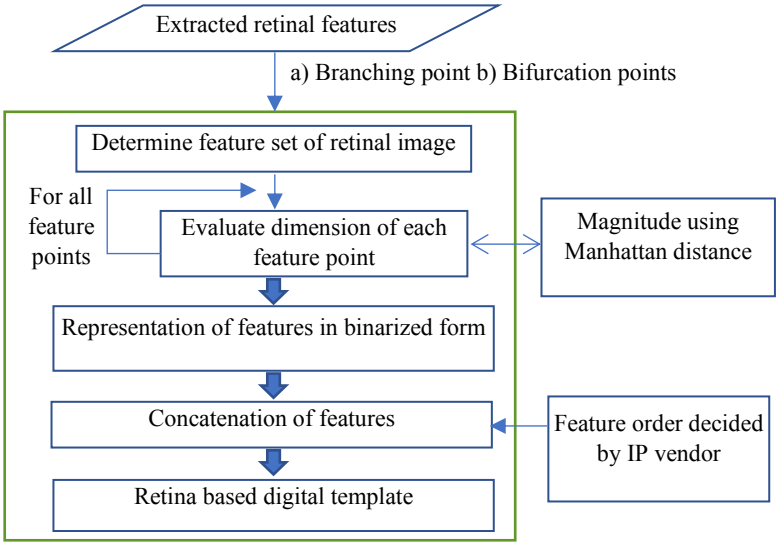


Fig. 6.3. Details of signature generation block used for retinal biometric based digital template generation

image and is subjected to designer selected specific grid size and spacing. For example, optic disc/optic nerve in the retinal image can be used as the region of interest, as it serves as the root of the retinal nerves (and blood vessels).

3) The retinal features are located/extracted by applying our kernels matrix of branching and bifurcation feature points on binarized image. Due to the uniqueness of the retinal vessel structure for each individual, the geometric properties of bifurcation and branching point can be used to generate unique retinal template.

4) The IP vendor decides the retinal features to be converted into the corresponding retinal signature. Based on the chosen features, feature points are selected on the retinal image. In the proposed approach, the IP vendor selected retinal features considered are branching and bifurcation. The vascular bifurcation and branching in the optical disk of the retina are considered landmark geometrical features that encompass vessels centerline and width information. Since they are considered special junctions in the retinal blood vessel where the vessel splits into two approximately equal-width vessels (branching) or if a new vessel formation occurs where a minor (smaller pixel width) vessel grows or comes out from a major (wider pixel width) vessel, it is known as bifurcation. Therefore, an IP vendor selects the above retinal features for securing the JPEG codec IP core. These retinal features are automatically detected using the convolution process. The convolution process automatically detects the feature points of retinal image corresponding to

feature kernel matrix by scanning the retinal image from top to bottom and left to right. Therefore, the feature extraction process accepts the kernel matrices (decided by the IP vendor corresponding to branching points or bifurcation feature points) and binarized input image matrix to generate the nodal feature points by performing the convolution operation automatedly.

5) Next, the retina biometric image with the vendor's chosen retinal features is generated.

6) For the selected retinal features, the feature dimensions are computed using the Manhattan distance metric.

7) Further, the IP vendor decides the feature concatenation order in which the retinal features are combined (concatenated) in order to generate the corresponding retinal signature.

8) Using feature dimensions and the selected order of features, the retinal signature is generated as a digital template.

9) Then, the generated retinal signature (digital template) is converted into corresponding covert hardware security constraints using an encoding algorithm specified by the IP vendor (not known to an adversary).

10) Subsequently, hardware security constraints corresponding to the retinal biometric are embedded into the design during the HLS process.

11) Finally, the retinal signature embedded secured RTL design of JPEG-codec IP is generated.

The IP vendor can vary and select the retinal signature strength accordingly (using the concatenation of all feature points' binarized Manhattan distance) by changing cropped image sizes, slight tilting of cameras, or changing in resolution. However, once the retinal signature is formed (fixed), it is final for embedding into the design and cannot be changed further as it is stored safely for the piracy detection process later. Since the retinal biometric information is not recaptured again for the piracy detection process, therefore these factors such as change in cropped image sizes, slight tilting of cameras, or change in resolution do not impact the piracy detection process. The retinal signature of an individual is always unique as the vessel structure is always unique for an

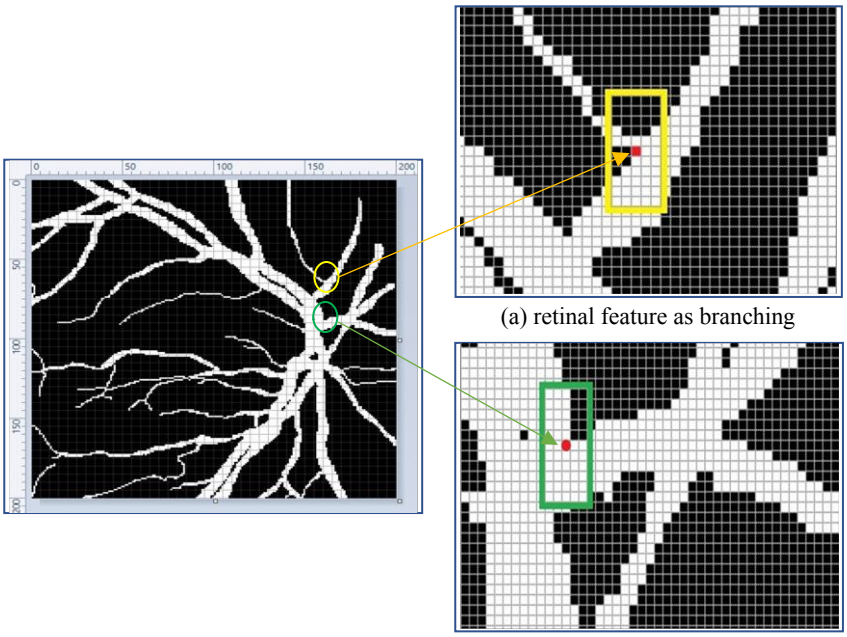
individual (even if they are twins). Furthermore, the retinal vessel structure of both eyes of an individual is also distinctive always. Hence the embedded retinal signature into a hardware design can be used as a robust unique secret mark to detect pirated design versions.

6.3. Demonstration on Automatically Detecting Retinal Feature Points for Digital Template Generation

The process for generating retinal digital template is demonstrated under the following subsections:

6.3.1. Capturing retinal biometric with IP vendor specified grid size/spacing

The retinal image is a digital image of the retina, optic nerve and blood vessels located at the back of the eye. The captured retinal biometric image (using fundus camera with the following specifications: field of view of around 45 degrees and resolution of size (565×584) of IP vendor is transformed into binarized image for accurate feature extraction [77]. The retinal images size p×q = '565×584' represents the pixel dimensions in the captured retinal image from which the ROI (comprising of the optic disc/optic nerve in the retinal image and serves as the root of the retinal blood vessels) is selected. The above retinal image size of '565×584' is taken as a sample magnitude for demonstration. However, a retinal image size with smaller or larger dimensions may also be chosen for embedding purposes. (*Note:* due to advancements in technology for capturing retinal biometrics, many easy-to-use devices with higher user acceptance rate are available for retina scanning. Retina can be captured using 20D lens and a standard quality camera also rather than uneasy exposure of eye to infrared light in conventional retinal capturing devices). The obtained binarized retinal image is subjected to vendor-specified grid size and spacing in order to generate feature nodal points (bifurcation and branching as shown in Fig. 6.4) and corresponding coordinates of retinal features accurately. This also helps during the retinal biometric verification process for hardware security, where the retinal feature coordinates and dimensions (magnitude) would easily be regenerated from the pre-stored original retinal image (with specific grid size and spacing). The

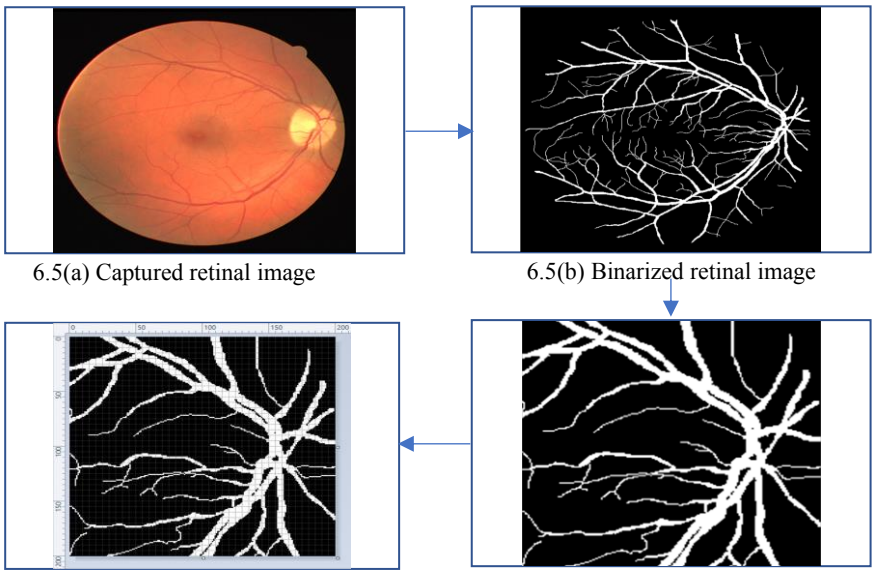


(a) retinal feature as branching

(b) retinal feature as bifurcation

Fig. 6.4. orientation of retinal features (a) representing branching nodal feature point with central pixel marked in red, is automatically detected using feature kernel matrix (as shown in Fig.6) corresponding to branching is represented in yellow (b) representing bifurcation nodal feature point with central pixel marked in red, is detected using feature kernel matrix (as shown in Fig.6) corresponding to bifurcation is represented in green.

process of generating the cropped retinal image with grid size and spacing is shown in Fig. 6.5. Where, Fig. 6.5(a) and Fig. 6.5(b) represents the captured retinal image and its binarized form respectively, while Fig. 6.5(c) and Fig. 6.5(d) represents copped binarized retinal image and its image with IP vendor specified grid size and spacing.



6.5(a) Captured retinal image

6.5(b) Binarized retinal image

6.5(d) Retinal image with specific grid size

6.5(c) Cropped retinal image of specific size (ROI)

Fig. 6.5. Placing ROI of retinal vessel structure into specific grid size (Image_1)

6.3.2. Automatic feature extraction from retina image and generation of nodal feature points

The retinal feature extraction block generates the nodal feature points such as bifurcation and branching points on retinal image matrix. The feature extraction block accepts the following inputs: generated image matrix of cropped binarized retinal image and our feature kernel matrix. The sample kernel matrices corresponding to branching and bifurcation nodal feature points are shown in Fig. 6.6, where each kernel matrix is of size m×n = 11×5. It represents the dimensions of the kernel matrix used for convolution operation in the proposed approach to automatically locate retinal features accurately. It contains binary values '0' indicating low-intensity pixels, and '255' indicates high-intensity pixels. Note: kernel matrix with lesser dimension may not be able to detect the retinal features accurately due to the wider pixel length of retinal blood vessels. The nodal feature points are generated by performing the convolution process between cropped retinal image matrix (as shown in Fig. 6.6(a)) and the kernel matrix. The convolution process automatically detects the feature points of a retinal image corresponding to the feature kernel matrix by scanning the retinal image from top to bottom and left to right. Therefore, the feature extraction process accepts the kernel matrices (decided by the IP vendor corresponding to branching points or bifurcation feature points) and binarized input image matrix to generate the nodal feature points by performing the convolution operation automatedly. As shown in Fig. 6.6(b), the nodal feature points marked in yellow indicate the branching points and feature points marked in green indicate bifurcation points. Note: the nodal point selection strategy includes 'bifurcation' and 'branching' as retinal features of the feature set. The unselected part of the retinal image (in Fig. 6.6 (b)) comprises of 'crossover' feature and does not technically fall under branching and bifurcation points. If an IP vendor wishes to expand the feature set, then he/she can include the third feature type 'crossover' into the feature set. In this proposed work we have considered 'branching' and 'bifurcation' features only, as these itself provide enough nodal points to result into adequate strength of retinal signature. Therefore, the output image matrix with located nodal feature points is

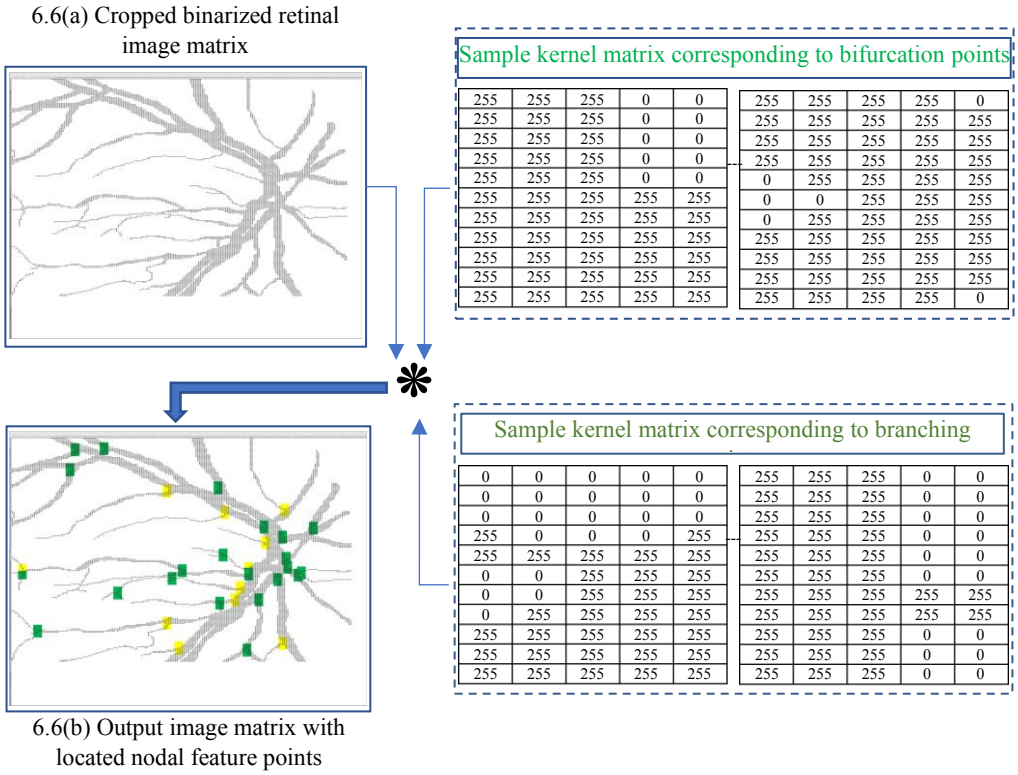


Fig. 6.6. Automatic detection of nodal feature points (bifurcation and branching) for Image_1

obtained which is used as the basis for security signature generation (explained in the subsequent sections).

6.3.3. Generating retinal image with IP vendor selected feature points

After the completion of convolution process between cropped retinal image matrix and the kernel matrix, the output image matrix with all the nodal feature points of bifurcation and branching on the retinal image is generated (as shown in Fig. 6.6(b)). Once this retinal biometric image is obtained, the vendor/designer decides the set of retinal features to be converted into corresponding retinal signature to secure JPEG-codec IP core. However, the IP vendor can generate the retinal signature by selecting the retinal features in the following ways: a) by selecting the bifurcation points only b) by selecting the branching points only c) by selecting both. Moreover, the number of features is also decided by IP vendor (the more the number of nodal features the more the signature size/strength). Therefore, depending on the target application to be secured IP vendor can choose signature of varying size/strength. Let's say the IP vendor selects both the branching nodal feature points and bifurcation nodal feature points for signature generation. Based on the selected retinal features, nodal points are generated on the retinal image. Fig. 6.6(b) shows the

nodal feature points (branching points are designated using yellow color while bifurcation points are designated with green color) on the captured image (ROI) of a retina. There are total 33 nodal feature points (22 bifurcation points and 11 branching points) on the retinal image (based on which the feature dimensions are computed as discussed in subsequent section 6.3.4). Hence, the retinal image with IP vendor-selected nodal feature points is generated.

6.3.4. Generating retinal digital template

Once the retinal biometric image with IP vendor-selected features has been obtained as discussed in the previous step, the dimension of each retinal feature point is determined. In order to do so, first the coordinates corresponding to all features (selected by IP vendor for signature generation) are determined. As each feature point is generated by applying the kernel matrix to input image, the resultant feature matrix is also of same size as kernel matrix. However, the center pixel coordinates of feature matrix are considered for determining the feature dimensions. For example, as shown earlier in Fig. 6.4 earlier, the center coordinates of feature matrix corresponding to branching and bifurcation feature is used for computing the feature dimension using Manhattan distance. Subsequently, feature dimensions (magnitude) corresponding to all the feature points are computed. The feature dimensions corresponding to IP vendor-selected retinal feature points are presented in Table 6.1. Thereafter, all feature points (22 bifurcation points and 11 branching points) are converted into their binarized form. However, in order to generate final retinal signature (digital template), features are concatenated depending on the concatenation order decided by the IP vendor. For example, the generated signature for the retinal Image_1 is obtained as follows:

a) ***Order of feature concatenation***: retinal features are concatenated as, Bi1 ╪ Br1 ╪ Bi2 ╪Br2╪Bi3…….╪Bi22). Where, bifurcation points and branching feature points are denoted by ‘Bi’ and ‘Br’ respectively and ‘╪’ represents concatenation operator. However, IP vendor can decide any of several possible concatenation orders.

Table 6.1 Determining feature dimensions and generating retinal signature

S.No.	Bifurcation feature points	Feature dimension	Binarize form	Size (bits)
Bi1	(10,55)	55.90	110111.11100110011001100110011	26
Bi2	(11,38)	39.56	100111.10001111010111000011	27
Bi3	(29,35)	45.45	101101.01110011001100110011	27
Bi4	(45,122)	130.03	10000010.00000111101011100001	29
Bi5	(80,149)	169.11	10101001.00011100001010001111	29
Bi6	(81,178)	195.56	11000011.10001111010111000011	29
Bi7	(89,160)	183.08	10110111.00010100011111010111	28
Bi8	(105,125)	163.24	10100011.00111101011100000101	28
Bi9	(108,162)	194.69	11000010.10110000101000111101	29
Bi10	(116,123)	169.07	10101001.00010001111010111	26
Bi11	(119,101)	156.08	10011100.00010100011111010111	28
Bi12	(121,7)	121.20	1111001.00110011001100110011	28
Bi13	(121,171)	209.48	11010001.01111010111000005101	27
Bi14	(122,140)	185.69	10111001.10110000101000111101	29
Bi15	(123,169)	209.02	11010001.00000101000111011	27
Bi16	(126,95)	157.80	10011101.11001100110011001101	29
Bi17	(127,157)	201.93	11001001.11101110000101001	26
Bi18	(139,63)	152.61	10011000.10011100001010001111	29
Bi19	(145,146)	205.76	11001101.11000010100011111011	28
Bi20	(148,123)	192.43	11000000.01101110000101001	26
Bi21	(173,16)	173.73	10101101.10111010111000005101	27
Bi22	(190,139)	235.41	11101011.01101000111101011	27
Branching feature points				
Br1	(48,92)	103.76	1100111.11000010100011111011	27
Br2	(65,161)	173.62	10101101.10011110101110000101	29
Br3	(67,126)	142.70	10001110.10110011001100110011	29
Br4	(94,150)	177.01	10110001.00000010100011111011	28
Br5	(117,140)	182.45	10110110.01110011001100110011	29
Br6	(119,7)	119.20	1110111.00110011001100110011	28
Br7	(134,135)	190.21	10111110.00110101110000101001	29
Br8	(145,132)	196.08	11000100.00010100011111010111	28
Br9	(166,92)	189.78	10111101.11000111101011100001	29
Br10	(184,160)	243.83	11110011.11010100011111010111	28
Br11	(188,99)	212.47	11010100.01111000010100011111	29

b) ***Number of features***: the signature security strength can be improved by selecting more number of retinal features.

The retinal signature corresponding to retinal image_1 is obtained by considering all 33 nodal feature points.

c) ***Signature generation***: the generated retinal signature is:

"110111.11100110011001100111100111.11000010100011111011100111.100
01111010111000011101011 01.10011110101110000101101101.01110011001
10011001110001110.10110011001100110011100000 10.00000111101011100
00110110001.00000010100011110 11--------11101011.01101000111101011 1"

(922bits).

Here the retinal signature strength of 922 bits represents the number of covert hardware security constraints to be embedded in the register allocation phase of HLS. The magnitude of 922 bits of retinal signature is directly obtained

from the 33 nodal points (representing bifurcation and branching) automatically detected from the retinal image using convolution operation. Therefore, the obtained retinal signature size is 922 bits in which the number of 1s is 478, the number of 0s is 411, and the number of binary points is 33. This indicates 478 security constraints are added between odd-odd storage variable pairs V<i,j> of the register allocation table of the design corresponding to bit '1'. Similarly, 411 security constraints are added between even-even storage variable pairs V<i,j> of the register allocation table of the design corresponding to bit '0'. Similarly, 33 security constraints are added between zero-integer storage variable pairs V<i,j> of the register allocation table of the design corresponding to binary point '.'. The encoding rule is shown in Table 6.3. The retinal signature strength can be varied by adding/deleting features form the feature set, as it would modify the number of nodal points on the retinal image. However, depending on the design size (to be secured using the proposed approach), IP vendor can select (truncate) the retinal signature size appropriately.

6.4. Demonstration on generating secured JPEG-codec IP using retinal biometric

In order to secure the IP design against piracy, retinal biometric signature in the form of encoded hardware security constraints is covertly embedded into the design. The details of generating secured IP design are discussed under following subsections:

6.4.1. Generating retinal biometric based secret hardware security constraints

After generating the retinal digital template using the proposed approach (as discussed in section 6.3.4), the bitstream (of 922-bit size) is converted into secret hardware security constraints. The security constraints (z), denote the number of bits of the retinal signature generated through the proposed approach for embedding. Each bit of the retina signature indicates an artificial edge inserted between two colors (registers) in the register allocation phase of the design process. These covert artificial edges enforce storage variable pairs to distinct register allocation. The secret security constraints generation

depends on the following: a) encoding rule specified by vendor/designer b) functional description of JPEG-codec (depicting the number of storage variables used to perform the operations) c) the number of signature bits (0's, 1's, binary points) and d) the ordering of storage variables. The secret security constraints generation corresponding to the retinal signature for JPEG-codec is demonstrated through the following steps:

1) Firstly, the functional description (transfer function) of JPEG-codec framework is transformed into a data flow graph. The step-by-step derivation of the JPEG codec DFG from its transfer function is discussed in section F.

2) Next, this DFG of JPEG-codec is scheduled using functional resources as shown in Table 6.2.

3) Subsequently, a register allocation table is prepared.

4) Finally, the hardware security constraints corresponding to the retinal signature are generated by using the encoding algorithm, as shown in Table 6.3.

The generated security constraints corresponding to encoding algorithm for signature bits ('411' 0's, '478' 1's and 33 binary point '.') are as follows:

The secret security constraints corresponding to the number of 0's in retinal signature bitstream is:

V<0,2>, V<0,4>, V<0,6>, V<0,8>, V<0,10>, V<0,12>, V<0,14>, V<0,16>, V<0,18>, V<0,20>, V<0,22>, V<0,24>,……, V<0,196>, V<0,198>, V<0,200>, V<0,202>, V<0,204>, V<0,206>, V<0,208>, V<2,4>, V<2,6>,……..…,<6,8>, V<6,10>, V<6,12>, V<6,14>,……..,V< 8,10>.

The secret security constraints corresponding to the number of 0's in retinal signature bitstream is:

V<1,3>, V<1,5>, V<1,7>, V<1,9>, V<1,11>, V<1,13>, V<1,15>, V<1,17>, V<1,19>, V<1,21>, V<1,23>,……, V<1,207>, V<3,5>, V<3,7>, V<3,9>, V<3,11>,……..,V<9,153>.

Further, secret security constraints corresponding to the number of binary bits '.' in retinal signature bitstream are:

Table 6.2 ASAP Scheduling (3+, 3*) of Macro IP of JPEG-codec

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					

Table 6.3 Encoding for generating the secret security constraints

Bit	Encoding rule
1	Embedding security constraints between odd-odd storage variable pair V<i, j> of the register allocation table
0	Embedding security constraints between even-even storage variable pair V<i, j> of the register allocation table
Binary point (.)	Embedding security constraints between 'zero-integer' storage variable pair V<i, j> of the register allocation table

V<0,1>, V<0,3>, V<0,5>, V<0,7>, V<0,9>, V<0,11>, V<0,13>, V<0,15>,……..,V<0,61> , V<0,63>, V<0,65>.

However, based on the different possible ordering of storage variables (sorted increasing, decreasing, sorted as per control steps ordering, alternate ordering of storage variables etc.), generating different combinations of security constraints is possible. Further, to enhance the security, an IP vendor can encode the retinal signature bits into hardware security constraints in numerous possible ways. Finally, these generated secret security constraints (as per designer selection) are embedded into the design in order to generate

retinal embedded secured JPEG-codec design as discussed in subsequent section 6.4.2.

6.4.2. Generating secured RTL design

Once the secret security constraints corresponding to the retinal signature are generated, embedding of the security constraints into the JPEG-codec design is performed in order to generate a retinal biometric implanted secured JPEG-codec design. We first discuss the general steps (1-7) of JPEG image compression and its representation as a transfer function/functional description, followed by the deduction of its respective data flow graph generation from its transfer function. Finally, the process of embedding is discussed:

6.4.2.1. Functional description of JPEG-Codec

The JPEG-codec is used to perform image compression and decompression. The process of computing the first pixel of the compressed image using JPEG compressor is discussed below:

Step1: transform the input image (to be compressed) into matrix form (square matrix form) of size M×M, where each pixel value of the matrix represents the pixel intensity value (0-255).

Step2: perform matrix slicing and generate non-overlapping matrix or block, each of size 8×8. This corresponds to the discrete cosine transform (DCT) function used in JPEG compressor, which takes 8×8 size blocks in one single control operation.

Step3: transform each 8×8 block of pixels using 2-D DCT transformation using following function:

$$T = (I*N) *I' \quad (6.1)$$

Where, 'I' denotes 2D-DCT coefficient matrix (shown in Fig. 6.7), N denotes 8×8 size block of pixels, I' represents the transpose matrix corresponding to matrix I and T denotes the transformed matrix.

Step4: compute the first pixel value of the transformed matrix, 'T11'. In order to compute T11, first we compute the output of the first micro-unit (IP1 of the DCT units) t11 as follows:

t11 = (i4*p11) + (i4*p21) + (i4*p31) + (i4*p41) + (i4*p51) + (i4*p61) + (i4*p71) + (i4*p81) (6.2)

where, in all product terms, the first operand value indicates the coefficient value of the first row of the coefficient matrix I, and the second operand indicate elements of the first column of matrix N. Now the first pixel of the compressed image is computed using the following function:

T11 = (i4*t11) + (i4*t12) + (i4*t13) + (i4*t14) + (i4*t15) + (i4*t16) + (i4*t17) + (i4*t18) (6.3)

Where, in all product terms, the first operand value indicates the coefficient value of the first column of the matrix I', and the second operand indicate elements of the first row of matrix I x N.

Step5: Now compression using a quantization matrix is performed on each DCT transformed 8×8 matrix block. Finally, by multiplying the first pixel of DCT transformed matrix (T11) with the quantization coefficient "Cq", the first pixel of the compressed image is computed as X11'. Similarly, other image pixels of the compressed image are computed.

Step6: In order to store the compressed image post quantization, the following operations are performed: a) convert the quantized image into 1D array using zigzag selection of elements of image b) apply run length encoding algorithm to obtain bitstream of compressed image (to be stored).

Step7: In case if the original image is to be reconstructed from the stored bitstream of the compressed image through JPEG decompression process, the following operations are performed: a) apply run length decoding b) inverse

$$\begin{bmatrix} i4 & i4 & i4 & i4 & i4 & i4 & i4 & i4 \\ i1 & i3 & i5 & i7 & -i7 & -i5 & -i3 & -i1 \\ i2 & i6 & -i6 & -i2 & -i2 & -i6 & -i6 & i2 \\ i3 & -i7 & -i1 & -i5 & i5 & i1 & i7 & -i3 \\ i4 & -i4 & -i4 & i4 & i4 & -i4 & -i4 & i4 \\ i5 & -i1 & i7 & i3 & -i3 & -i7 & i1 & -i5 \\ i6 & -i2 & i2 & -i6 & -i6 & i2 & -i2 & i6 \\ i7 & -i5 & i3 & -i1 & i1 & -i3 & i5 & -i7 \end{bmatrix}$$

Fig. 6.7. 2-D DCT coefficient matrix "I"; Matrix elements indicate eight- point DCT coefficients.

zigzag selection of image elements c) inverse quantization d) inverse DCT transformation.

6.4.2.2. Data flow graph generation of JPEG-Codec

As discussed in earlier section that the functional description of JPEG-codec is transformed into DFG/CDFG. DFG of JPEG-codec is shown in Fig. 6.8, which computes the first pixel of the compressed image (post performing quantization). JPEG-IP core comprises of eight sub-IPs (micro-IPs, IP1 to IP8). Each micro-IP performs 16 operations (IP1 have 9 multiplications and 7 additions as shown in Fig. 6.8). Therefore, total operations are: operations of one IP* number of IPs + operations between micro-IPs. Hence total 136 operations are performed to compute the first pixel of the compressed image in the JPEG-codec design IP core.

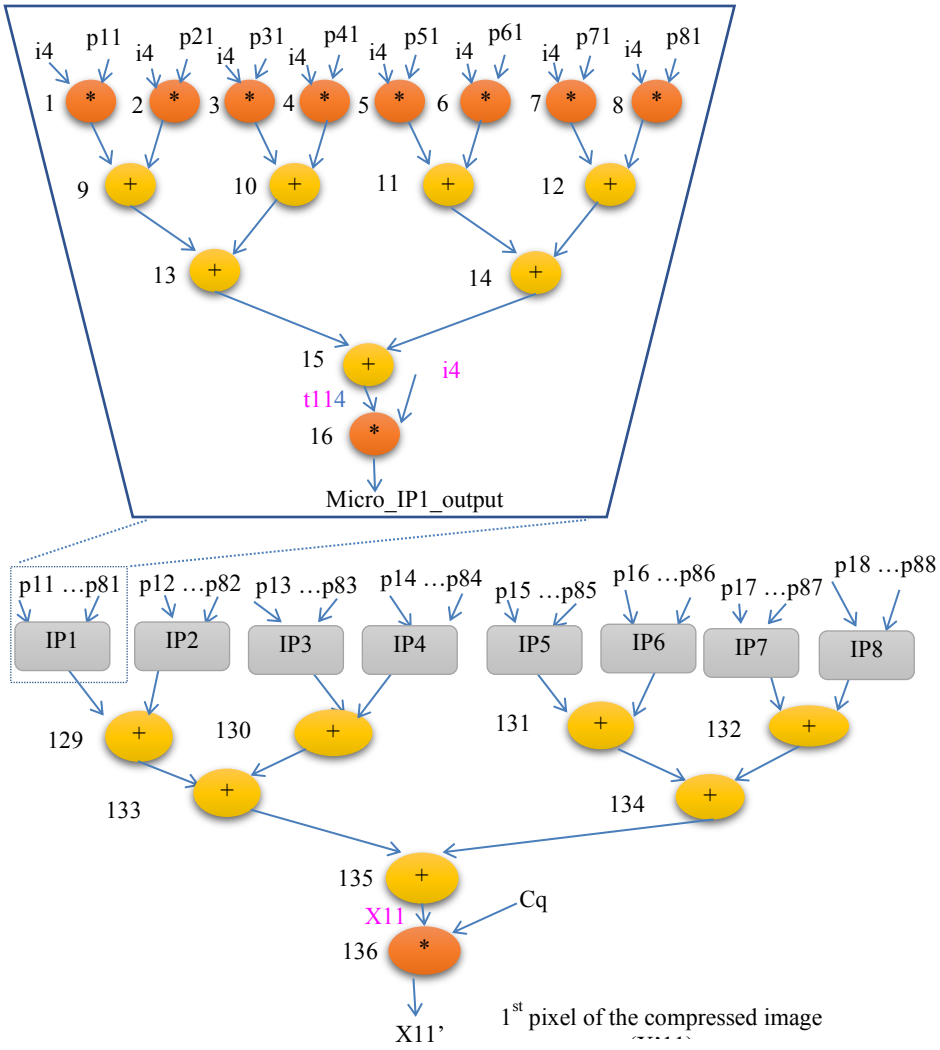


Fig. 6.8. DFG of JPEG-CODEC IP core

6.4.2.3. Scheduled Data flow graph generation of JPEG-Codec

Further, scheduling of operations, hardware allocation and bindings are performed using resource constraints. As shown in Table 6.2, three adders and three multipliers are applied for scheduling the DFG of the design. Further, there are 136 operations in the design. However, 30 control steps are required to schedule the corresponding data path using the following resource constraints. Post obtaining the scheduled DFG of JPEG-codec design register allocation table is constructed which contains the following details: a) number of storage variables (used to store primary and intermediate input/output values), b) registers required corresponding to storage variables c) control steps required to generate the first output pixel value of compressed image. As shown in Fig. 6.9, register allocation contains the 73 different registers corresponding to 209 storage variables and 30 control steps.

6.4.2.4. Retinal signature embedded register allocation framework of JPEG-codec design

After generating the secret security constraints corresponding to retinal signature using encoding algorithm (as discussed in earlier section), embedding is performed during the register allocation phase of HLS process. In order to do so, steps are as follows:

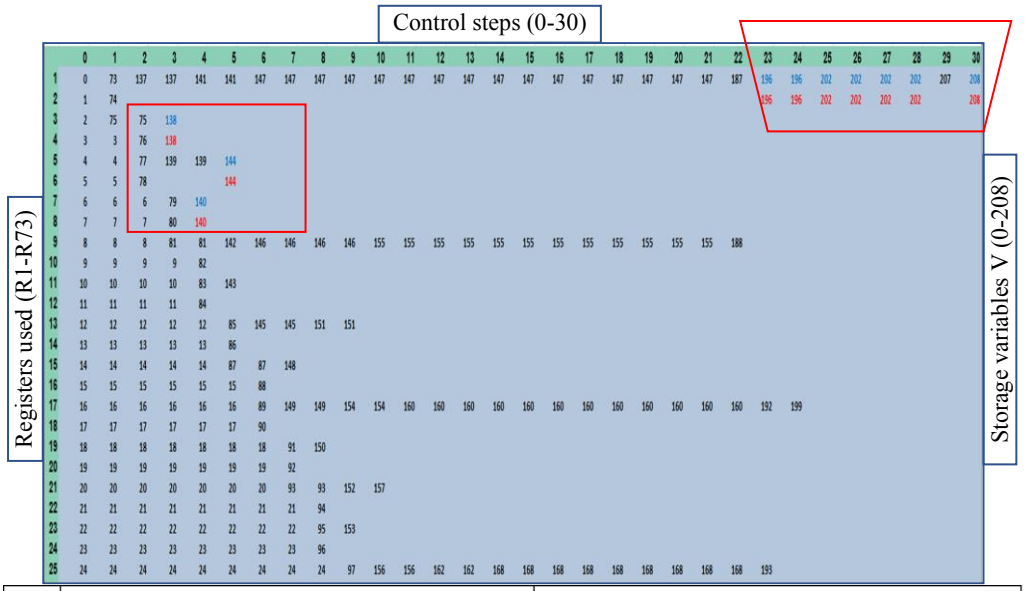
1) first perform the mapping of retinal signature template of IP vendor selected size into secret security constraints using encoding rule.

2) generate security constraints corresponding to 411 zeros, 478 ones and 33 binary points of retinal signature.

3) embed each of the security constraints into the register allocation framework.

Further, constraints embedding rules are: If any two storage variables are executing in same control step, then they cannot share the same register. Further, if any two storage variables are executing in different control steps, then they can share the registers. For example, as shown in Fig. 6.9, for the storage variable pair V(0-2), as they are already assigned in different registers, register 1 and register 3, respectively, therefore no conflict will occur. However for the storage variable pairs V(0-196), V(0-202), V(0-208), conflict

occurs. Therefore, they (V196, V202, V208) cannot be accommodated with V0 in the same register. Similarly, for the pair V(2-138), V(4-144), V(6-140), conflict occurs, and they also cannot be accommodated with storage variables V2, V4, and V6, respectively. For the sake of brevity, register allocation details, post embedding the secret security constraints is, shown in Fig. 6.9, where the storage variables in red indicate the updated positions of the storage variables corresponding to the old position of storage variables marked in blue. It is evident from the register allocation framework that no extra register is required during the embedding of all secret security constraints corresponding to retinal signature (for image_1) into the register allocation framework of JPEC-codec design. Finally, the signature embedded register allocation framework as shown in Fig. 6.9 is obtained. Further, retinal biometric implanted secured JPEG-codec datapath is subsequently designed using HLS.



CS	Pre-embedding								Post-embedding							
	R1	R2	R3	R4	R5	R6	R7	R8	R1	R2	R3	R4	R5	R6	R7	R8
3	137	-	138	-	139	-	79	80	137	-	-	138	139	-	79	80
4	141	-	-	-	139	-	140	-	141	-	-	-	139	-	-	140
5	141	-	-	-	144	-	-	-	141	-	-	-	-	144	-	-
23	196	-	-	-	-	-	-	-	-	196	-	-	-	-	-	-
24	196	-	-	-	-	-	-	-	-	196	-	-	-	-	-	-
25	202	-	-	-	-	-	-	-	-	202	-	-	-	-	-	-
26	202	-	-	-	-	-	-	-	-	202	-	-	-	-	-	-
27	202	-	-	-	-	-	-	-	-	202	-	-	-	-	-	-
28	202	-	-	-	-	-	-	-	-	202	-	-	-	-	-	-
29	207	-	-	-	-	-	-	-	207	-	-	-	-	-	-	-
30	208	-	-	-	-	-	-	-	-	208	-	-	-	-	-	-

Fig. 6.9. Register allocation framework post embedding retinal security constraints (pre and post embedding table represents changes due to security constraints). Note: For the sake of brevity, details of only 25 registers (out of 73) have been presented.

6.4.3. Detection of retinal biometric security mark into the design

In the proposed approach, the retina biometric of IP vendor is only captured once before the embedding process and the corresponding retinal biometric image (with grid size and spacing) is safely stored for IP piracy detection process later by a system integrator. There is no need to recapture the retina biometric again for the detection process. The existing pre-stored retinal biometric image is used to regenerate the retinal signature and its corresponding hardware security constraints to detect pirated designs. The same features and their dimensions of the retina biometric can be identified and computed accurately from the pre-stored retinal image. Since, the retina biometric is only captured once and hence factors such as vascular damage to the eyes, fatigueness, slight tilt of camera, variation in resolution, difference in cropping size do not have any impact on the IP piracy detection process. The detection process is independent of recapturing of the retinal biometric information. There will be no differences in extracted biometric data as second-time capturing of the biometric information is not required. The pre-stored retinal image (with grid size and spacing) is sufficient to detect IP piracy. Additionally, the embedded retinal signature acts as a strongly authentic, naturally unique secret mark that enables the detection of pirated IPs. The piracy detection process of JPEG-codec IP core design is shown in Fig. 6.10.

Note: The retinal signature of a genuine IP vendor would match with the embedded digital signature because of uniqueness of retinal vessel structure of each individual. Further, in the case of twins, the retinal structure remains distinctive. Therefore, it is impossible for an adversary to possess the same retinal characteristics as of genuine IP vendor's retina. Further involvement of several complex information during signature generation and implantation makes it almost impossible for an adversary to evade piracy detection process.

6.4.4. Security properties/parameters of retina biometric based security methodology

The proposed retinal biometrics for securing hardware IP core render several security properties as described below:

```
Input:
a) regenerated secret security constraints corresponding to authentic retinal signature
b) design under test (RTL information)
output:
IP piracy detection
Pseudo code:
While (position of retinal signature bits are matched bit by bit with embedded signature)
    {
        If (secret security constraints are matched 100%)
        {
           Design is not pirated;
        }
        Else
        {
           Design may be pirated;
        }
    }
```

Fig. 6.10. Pseudo code for isolating the pirated designs

(a) exact regeneration (replication) of retinal signature is impossible for an adversary because of several crucial security parameters required for signature generation (which are all unknown to an adversary) such as:

i) IP vendor selected region of interest in captured retinal image; ii) IP designer specified grid size and spacing of retina biometric image; iii) kernel matrix size of retinal features; iv) convolutional algorithm used for retinal nodal feature point extraction; v) orientation of kernel matrices; vi) type of nodal feature points and how many of them were used for signature generation; vii) naming convention and ordering of retinal features (may be corresponding to convolution process or IP vendor specific); viii) coordinates of retinal nodal feature points ix) truncation length of the generated retinal signature (decided by IP vendor) before generating its corresponding covert hardware security constraints for embedding.

(b) the effectiveness of the proposed retina biometric-based hardware security can also be measured using the criteria, False Accept Rate (FAR) and False Reject Rate (FRR). In the proposed approach, FAR as 0% for an adversary

and FRR is 0% for genuine IP vendor. This is because in case, even if an adversary gets access to pre-stored retinal image, he/she cannot regenerate the exact same retinal signature because an adversary is unaware of the security variables (listed above from i to ix) required, that was originally specified by genuine IP vendor.

(c) As discussed earlier, the pre-stored retina biometric image of IP vendor with specific grid size is used for piracy detection. However, in case if the stored image is leaked to an adversary, the exact regeneration of retinal signature (digital evidence) from compromised/leaked retina biometric image is not possible. This is because the security layers/parameters discussed earlier (from point (a). i to ix are all unknown to an adversary. In the proposed approach, IP vendor does not store his/her retinal signature. However, in case if an adversary even manages to derive the exact retinal signature, the generation of secret hardware security constraints is not possible because of following details unknown to an adversary:

i) truncation length of the retinal signature employed (known to the original IP vendor only) before generating the final retinal signature for extracting covert hardware security constraints.

ii) genuine IP vendor-specified encoding rule used for generating the hardware security constraints corresponding to the retinal signature strength.

iii) ordering of storage variable of the design is also unknown (either sorted in ascending order or sorted in descending order or sorted as per CS in scheduling or alternate arrangements of storage variables or arrangements based on FUs, etc.) that is responsible for creating storage variable pairs of the security constraints.

iv) retinal signature generation methodology which includes security factors such as: type of retinal features used, number of generated retinal nodal points, feature concatenation order, feature dimensions etc. These factors are only known to the original IP vendor. Therefore, an adversary cannot evade IP piracy detection process. This renders the proposed retina biometric approach for securing hardware IP core as highly robust even if retinal image is compromised/spoofed.

(d) adversary cannot evade the piracy detection process as the complete matching of secret security constraints of regenerated signature is mandatory with the extracted register allocation information of the target design under test.

(e) even in the case of two identical twins, an exact match of the retinal signature is impossible due to the highly distinctive vessel structure of retinal biometrics.

(f) information regarding the ordering of storage variables (used in the design to store intermediate, primary input and output results) based on which secret security constraints are generated for embedding into the design, is restricted to genuine IP vendor only. Further, the position of signature bits (0s, 1s and binary points) and their strength are only known to the genuine IP vendor.

(g) due to robust retinal signature, it provides higher tamper tolerance and lesser probability of coincidence.

(h) the proposed retinal biometric approach for securing JPEG-codec IP core is independent of any external key for signature generation. Therefore, it is not vulnerable to key exploitation attacks.

(i) proposed retinal biometric incorporates more robust covert security constraints generation due to more number of encoding digits of generated retinal signature than facial and fingerprint biometric (comprises two digit encoding).

(j) in case of retinal biometrics, it is not possible for an adversary to capture the retinal biometric without the consent of an individual or by using ordinary camera (through superficial imaging). Therefore, it is the safer than other biometrics for enabling robust security of hardware IP cores.

(k) moreover, in case of securing a large size hardware IP core, retinal structure of both the eyes corresponding to genuine IP vendor can be exploited to generate more robust retinal signature template.

(l) the revocability of the biometric template is also a crucial perspective for enhancing the robustness of the system. The proposed system is inherently capable of generating or reissuing another instance of retinal signature

corresponding to IP vendor in case if attacker manages to compromise biometric template (it should be noted that the proposed approach does not require storing the retinal signature). This is because different retinal signature could be generated corresponding to retinal biometrics of genuine IP vendor. In order to do so, the proposed approach offers the generation of new biometric template corresponding to an IP vendor by offering the selection of different ordering of retinal features, signature truncation length and different constraints generation encoding algorithm. This renders the formation of the different retinal template corresponding to same IP vendor. This, therefore, ensures significantly robust security of the proposed system in terms of revokable property.

(m) the proposed retinal biometric signature offers stronger security than embedding random secret key into the design. This is because in case if the random signature is leaked/compromised, then evading IP piracy detection is possible by an adversary as he/she can easily embed the information in fake IPs. However, in case of proposed retinal biometric signature, the key or retinal signature or hardware security constraints is not stored. Only the biometric retinal image is stored which on compromise does not cause security breach in terms of evasion of IP piracy detection. This is because, even if the pre-stored retinal biometric image is compromised/leaked to an adversary, regeneration of retinal signature is impossible. This is due to several security layers mentioned in points (a) i to ix earlier (which all are unknown to an adversary and is extremely difficult to guess/break). For example, in some of the security layers the conversion of a retinal image into a sequence of 0 and 1 is performed which itself is an arduous task for an adversary as there are several intricate parameters involved such as feature nodal points, feature order, feature set etc. which creates innumerable possibilities (as discussed earlier in (l)). Additionally, during the signature embedding process, an IP vendor can encode the signature into hardware security constraints in innumerous possible ways (discussed earlier in section 6.4.1). Therefore, for an adversary to evade IP piracy detection process, he/she needs to break all these security layers/parameters. Therefore, the proposed approach offers more robustness in security as compared to embedding random sequence of bit

0 and 1. Therefore, neither storing the random signature nor the retinal signature is good alternative. Hence, the proposed approach does not store retinal signature which could in turn potentially cause a security breach through leakage (therefore only retinal image is stored which on potential compromise does not allow an adversary in regenerating the retinal signature-based security constraints). In case of the proposed approach, instead of storing biometric template, only the retinal image is safely stored. Further, during IP piracy detection process, verification of the embedded encoded hardware security constraints in the register transfer level design file of the IP is performed bit-by-bit position-wise. Therefore, accessing only the biometric image will not help the adversary to spoof/compromise the security of the proposed approach in terms of evasion of IP piracy detection.

6.5. Results and analysis

The proposed result reports the following: (a) variation in probability of coincidence metric for different sizes of embedded retinal signature (ranges from 2.5E-1 to 4.0E-6) and different retinal images (ranges from 2.9E-4 to 1.4E-7) (b) variation of tamper tolerance for proposed approach for different retinal images (ranges from 1.05E+281 to 1.0E+538) (c) security comparison of proposed approach with facial, fingerprint biometric and encrypted digital signature-based hardware security approaches (d) Pc-design cost tradeoff for proposed retinal biometric approach. Results indicate enhancement in security at zero design overhead. The experimental results of the proposed methodology to design a secured JPEG-codec IP core have been discussed and analyzed in detail in chapter 9 of this thesis.

6.6. Summary

This chapter presented a novel HLS based hardware security approach for securing JPEG-coded IP core against threat of piracy using retinal biometric. Robust security against piracy (in terms of seamless detective control) is achieved by embedding unique retinal signature of authentic IP vendor into the design while incurring zero design overhead. The embedded retinal signature in the form of encoded hardware security constraints enables the detective control against pirated versions during piracy detection process. This

therefore enables to discern and isolate pirated IP versions before being integrated into SoCs of CE systems. Thus, the proposed retinal biometric approach ensures seamless and robust detection of pirated versions of design, therefore, it ensures the security and integrity of end consumer. Additionally, the proposed retinal biometric approach is also capable to ensure robust security of any DSP and multimedia hardware IP core designs. Further, ensuring security of DSP based JPEG-codec IP cores against piracy threats is crucial for both SoC designer and end consumers, as it is widely used in modern CE systems. The presented methodology was proven to be more robust in terms of security than recent similar works while incurring zero design cost overhead.

Chapter 7

Exploration of security-design cost tradeoff for signature driven security algorithms for optimal architecture of data-intensive hardware IPs

This chapter presents a novel approach for the exploration of security-design cost trade-off for signature-based hardware security algorithms for data-intensive digital signal processing (DSP) intellectual property (IP) cores. Data-intensive DSP application frameworks such as finite impulse response (FIR) filter, discrete cosine transform (DCT), discrete wavelet transform (DWT) and ARF are widely used to facilitate image compression-decompression, digital data filtering, sound processing, signal coding, gait analysis and so on [82], [83. Owing to their usages along with the rapid growth in modern technology and globalization process, the demands of optimal hardware IP core designs that are secure and low cost have become very significant and imperative. Furthermore, before integrating an IP core into system on chips (SOCs)/end systems, the following orthogonal issues need to be addressed: optimizing the design architecture (yielding lower design cost) as well as enhancing security against external hardware attacks. An IP core before its integration into an integrated circuit (ICs), may take several years of research, development and design. Exploring optimal design architecture for secured IP cores using high level synthesis (HLS) is a tiresome task [83]. Therefore, the knowledge of optimal IP design architecture can play a major role in obtaining an optimal CE system in terms of robust hardware security and lower design cost. Since, DSP applications are computationally intensive therefore their optimal hardware can be designed using HLS process integrated with design space exploration process such as particle swarm optimization (PSO) [78]. Apart from the optimality issues the security threats arising due to involvement of offshore design houses in modern design supply chain, renders a third-party IP (3PIP) core completely untrustworthy [5]-[7], [25]. Further, the involvement of the multivendor third-party IP cores (designed in a fabless center) during the process of system on chip (SoCs) integration generates possibility for an adversary to perform malicious activity [8], [11]. The major security

challenges involved during system design of an end product includes IP counterfeiting, IP cloning and false claim of IP ownership proof [31]-[41].

The proposed approach offers optimal hardware design architectural solutions using particle swarm optimization (PSO) based design space exploration (DSE) for secured IP cores that are ubiquitously used in consumer electronics (CE) systems. In the proposed methodology, three different hardware security algorithms viz. facial biometrics, encrypted-hash and watermarking, have been integrated with the PSO-DSE framework for exploring the trade-off of security-design cost. The proposed methodology enables the IP core vendor and CE integrator to decide the choice of their data-intensive hardware IP architecture such that it meets the end objective of robust security (against fake/pirated IPs) and lower design cost. The proposed approach is capable to obtain an optimal secured design solutions for DSP hardware used in electronics systems based on security-design cost tradeoff using PSO for different signature based security algorithms.

Outline of the chapter is as follows. The first section formulates the problem. The second section discusses the methodology for exploration of security-design cost for obtaining low-cost architectural solution under the following sub-sections: motivation and overview. Further, the third section discusses the process flow of different signature-driven security algorithms. The fourth section demonstrates the process flow of generating low-cost and secure architectural solution for DCT 8-point application under the following subsections: details of PSO-based design space exploration, details of scheduling, allocation and binding process, details of signature embedding process and details of security-design cost tradeoff fitness function. Finally, the fifth section summarizes the chapter.

7.1. Problem Formulation

Given the data intensive hardware IPs in the form of transfer function, module library, along with signature generation tool box comprising of different signature driven hardware security algorithms and the objective of exploration of security-design cost trade-off for obtaining low-cost architectural solution. Therefore, generating low-cost architectural solution corresponding to various

security algorithms for varying (scalable) signature strengths and different data intensive DSP based hardware IPs.

7.2. Methodology for exploration of security-design cost trade-off for obtaining low-cost architectural solution

In this section the proposed methodology has been discussed based on the motivation and overview.

7.2.1. Motivation

Ensuring optimization and robust security in parallel for the IP Core designs are the major concerns for any IP Core designer. Further, it is crucial to choose one security approach over the other in terms of generated signature strength and combination. However, selection is influenced by several crucial parameters such as: temper tolerance ability, strength of IP ownership proof by the genuine designer, vulnerability and replicability of the security mechanism, counterfeit detection control and implementation complexity. Further, there is tradeoff between design optimization and security as enhancing one may influence others. This encourages to analyze the impact of choosing a particular security approach on design optimality. Further, ensuring robust security while incurring minimal design cost is imperative for CE systems integrating the reusable hardware IPs, thereby ensuring security of end consumers against security hazards at low-design cost. Therefore, it is viable to design an optimal as well as secured IP with low design cost.

7.2.2. Overview

In this chapter, an approach for the exploration of security-design cost trade-off for signature-based security algorithms for DSP hardware used in CE systems has been presented. A stochastic multi-objective particle swarm optimization [78] algorithm has been operated for the same. The primary inputs to the proposed approach are signature generation tool box, input DSP application (in form of C-code/transfer function), library [86] and PSO input parameters such as population size, acceleration coefficient, inertia weight and terminating criteria. The output of the proposed approach is an optimal security-design cost solution for the DSP application based on the security

algorithm selected from the signature generation toolbox by the designer. The major blocks of the approach as shown in Fig. 7.1 are: PSO-based design space exploration, HLS scheduling, allocation and binding, DSP application input block, signature embedding block and security–design cost tradeoff fitness function block. PSO-based design space exploration block is responsible for performing the exploration of a low-cost resource configuration by considering the parameters of security and design cost. HLS based scheduling block is responsible for scheduling the DFG of input DSP design based on PSO-driven resource configuration. Post scheduling, hardware is allocated and their binding is performed. DSP application input block is responsible for transforming the behavioral description of the DSP application into data flow graph. Next, signature embedding block is responsible for embedding the signature corresponding to IP designer selected

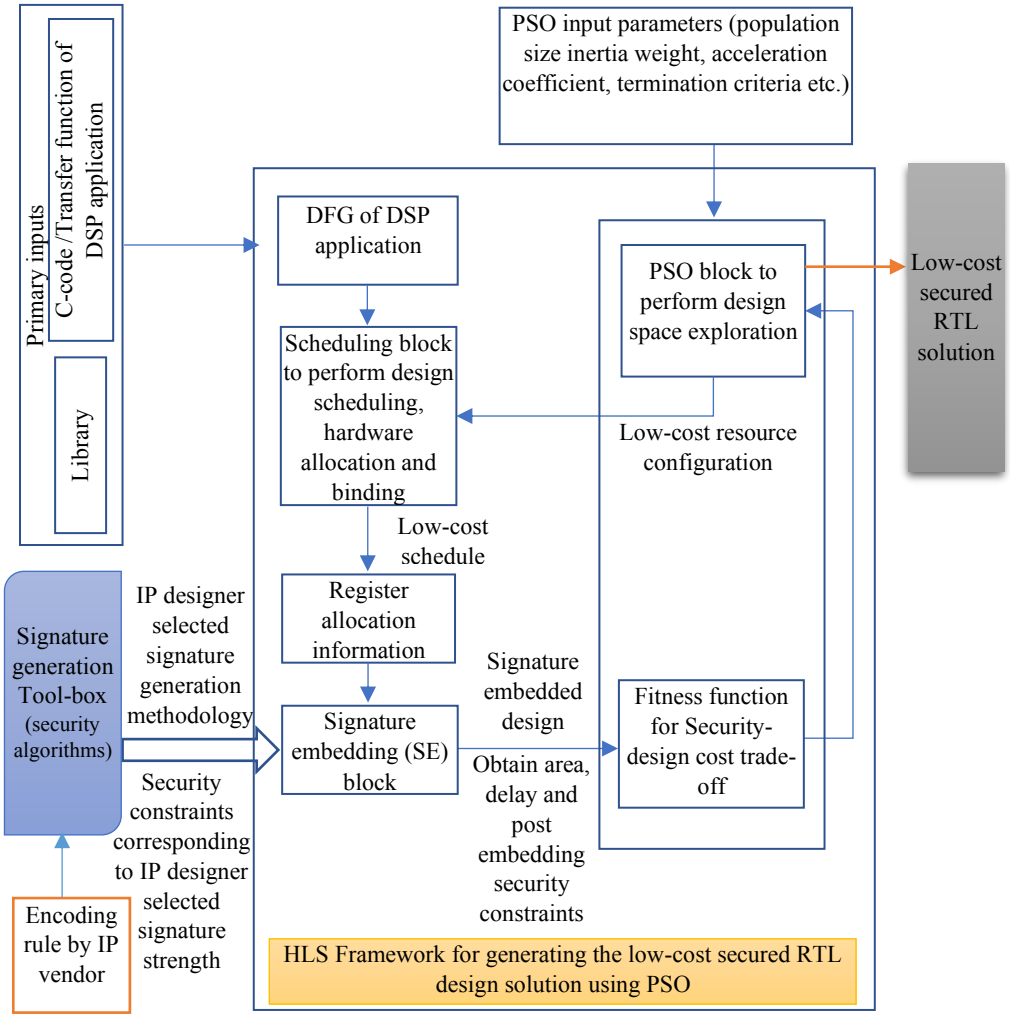


Fig. 7.1 Details of the proposed methodology for performing security-design cost trade-off

security algorithm. The security-design cost tradeoff fitness function block results into an optimal security-design cost register transfer level architectural solution as an output based on the different signature-based security algorithms used for DSP applications.

7.3. Process flow of different signature driven security algorithms

The signature (digital evidence) can be generated with respect to any DSP application and any security algorithm selected by an IP designer. As shown in Fig. 7.2, signature generation tool-box comprises of the signature-based security algorithms. An IP designer may select any of the signature-based security algorithm (with specific signature strength) for embedding into the target design. In the proposed methodology, the security algorithms that are mainly considered for analyzing the security-design cost tradeoff are IP watermarking, encrypted hashing and facial biometric based. Further, an IP designer can also select any of the DSP applications to obtain its corresponding secure and low-cost architectural solution. This therefore enables an IP designer to obtain an optimal and secured architectural solution for DSP applications corresponding to the security algorithm chosen by IP designer. To generate the signature using the watermarking-based approach [31], [32] it uses a robust multi-variable signature encoding methodology for generating the signature as secret digital evidence. An encrypted hash-based algorithm [39] it uses multi-level encoding, SHA-512, and RSA algorithms for the security of complex reusable IP cores used in CE systems. Furthermore, the signature generation process using the biometric approach [41], uses facial features (always unique in the form of nodal points) of an individual (IP vendor). Signature can be generated by the combination of different facial features (more the number of features more the signature strength) and by the different ordering possibilities. The details of precise co-ordinates of nodal points, type of feature selected, ordering of the features, position of the bits (0and1) grid size and are unknown to an adversary even if being a look alike or twin. This makes facial biometric approach more secure as compared to watermarking and encrypted hash-based approach. It is more robust to prevent

true IP designer from the fraudulent claim of IP ownership. The process flow of different signature driven security algorithms is discussed below:

7.3.1. Watermarking based hardware security

In the watermarking approach, signature is generated based on auxiliary multi-variable (i, I, T, !) combination of IP designer chosen signature length. Subsequently, multi-variable signature is encoded to generate its corresponding secret hardware security constraints using following encoding algorithm as shown below:

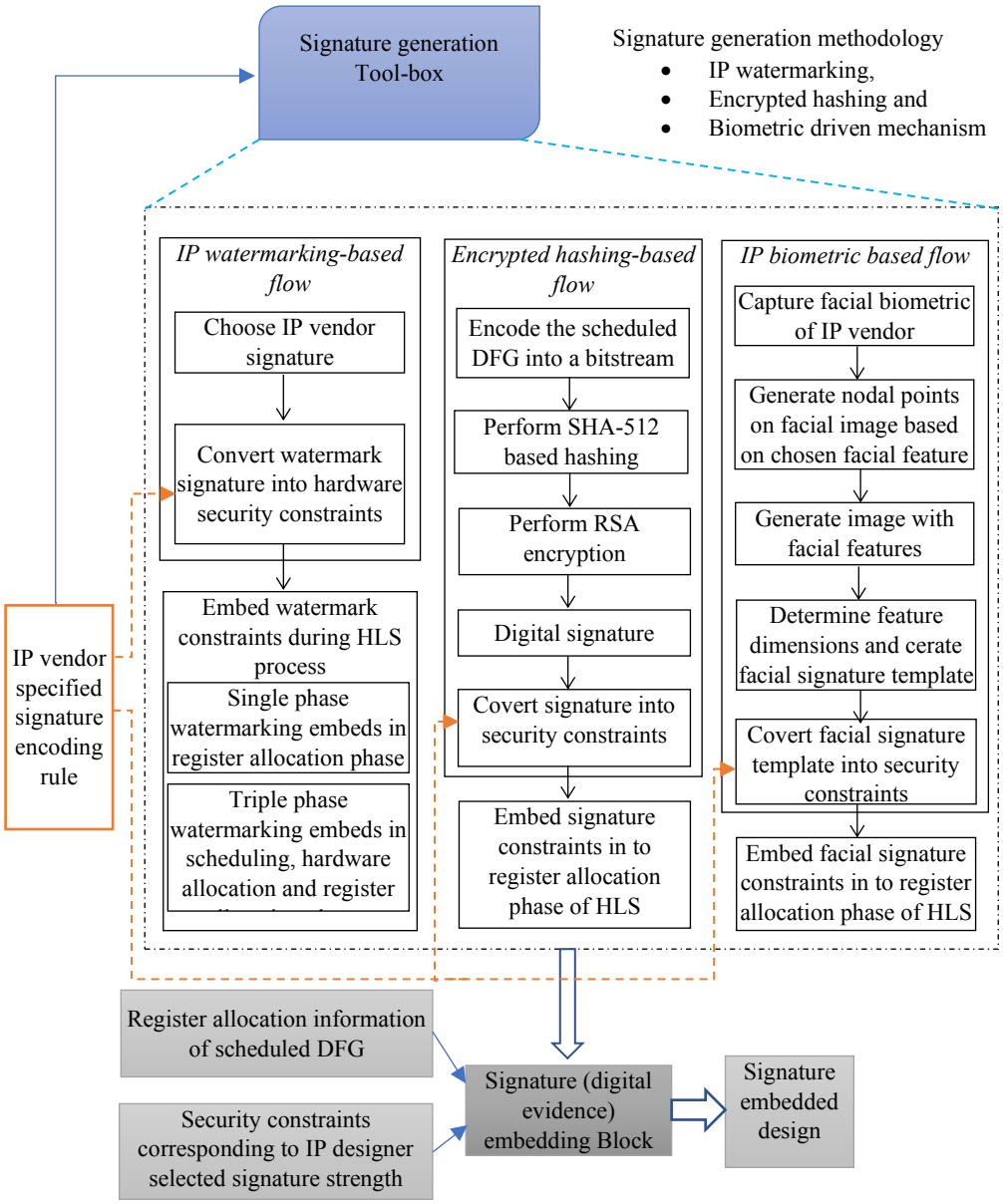


Fig. 7.2 Process flow of signature generation methodologies and embedding their corresponding generated signature during HLS process

i-embed an edge between storage variable pairs of prime-prime,

I- embed an edge between storage variable pairs of even-even,

T- embed an edge between storage variable pairs of odd-even and

!- embed an edge between storage variable pairs of zero-any integer (depending on the size of the DSP application).

The hardware security constraint pairs (S_VX to S_VY) are formed using storage variables obtained from DFG of the design, where X and Y denotes the storage variable number. Subsequently embedding of the signature constraints is performed in the last phase of watermarking scheme as shown in Fig. 7.2. On the other hand, in case of single-phase watermarking the generated security constraints are embedded in register allocation phase whereas in the triple phase watermarking approach the generated security constraints are embedded during scheduling, FU vendor and register allocation phases.

7.3.2. Encrypted-hash based hardware security

Furthermore, the encrypted hash-based algorithm [39] encodes the scheduled DFG of the DSP application into a bit stream based on the following encoding rule:

Bit='0', if the operation number and the control step number assigned to the operation are of the same parity and

Bit='1', if the operation number and the control step number assigned to the operation are of different parity.

Subsequently hashing is performed based on SHA-512 algorithm. The encrypted digital signature has been generated after performing the RSA encryption using 128-bit private key chosen by the IP vendor. The generated signature is used to obtain the security constraints using the following encoding rule:

'0'-embed an edge between storage variable pair of prime-prime and

'1'- embed an edge between storage variable pair of even-even into the register allocation information. Finally, the embedding of the signature

constraints into register allocation phase of HLS is performed to obtain the signature-embedded DSP RTL design.

7.3.3. Facial biometric based hardware security

In the facial biometric approach [40] to obtain the signature embedded design, capturing the facial biometric of the IP designer has been performed initially. Subsequently nodal points are generated on captured facial image based on the chosen facial feature set. Subsequently, in the next phase, image with the facial features has been produced and based on that feature dimensions are determined. Subsequently, facial signature template has been generated. Subsequently in the next phase converting the facial signature template into security constraints has been performed. The following encoding rule has been employed for converting the facial signature template into the respective hardware security constraints:

'0'-embed an edge between storage variable pair of even-even and

'1'- embed an edge between storage variable pair of odd-odd into the register allocation information. In the final phase, embedding of facial signature constraints into the register allocation phase of HLS has been performed to obtain the secure signature embedded DSP RTL design. Furthermore, the facial biometric based approach is more robust against forgery attacks (exact regeneration of secret mark is impossible) as the employed intricate parameters such as grid size, types of facial features chosen by IP designer, ordering of the features for deriving the signature, the position of signature bits (0s,1s) and the encoding rules, all are unknown to an adversary. Additionally, a qualitative comparison among the above security approaches in shown in Table 7.1.

7.4. Demonstration on generating low-cost and secure architectural solution for DCT 8-point application

In order to generate low-cost architectural solution, the resource constraints are obtained using PSO based design space exploration methodology. Subsequently, based on the generated optimal resource constraints, the target DSP design is scheduled. Next, the generated hardware security constraints

corresponding to different security algorithms are embedded into the design during register allocation phase of HLS framework. Thereafter, by analyzing the security-design cost tradeoff fitness function, low-cost and secured architectural solution is obtained. The details of each module have been discussed below:

7.4.1. Background on PSO

In the PSO-based design space exploration, inputs provided are CDFG, module library to fetch the details of each vendor, and PSO primary inputs such as inertia weight, acceleration coefficient, population size (number of particles) and number of iterations. Next, each particle (total number of particles' n' are user-defined) is initialized with their initial position (initial resource configuration; # adders and multipliers), and velocity is also initialized. The basic steps for performing PSO-DSE are as follows [78]:

(a) initialization of the particles corresponding to the resources used in the DSP hardware, (b) update global best and local best solution corresponding to DSP hardware, (c) determining new architectural solution (PSO based), (d) cost evaluation process considering normalized latency and design area of the DSP hardware, (e) update local best solution of the particle-based on the outcome of step number (d), (f) update global best architectural solution, and (g) repeat steps (c) to (f) until stopping criteria do not meet. The exploration process gets terminated either if there is no further updating in fitness cost value for the successive ten consecutive iterations (considering that solution got stuck or converged to local minima) or if it is iterated for the user-defined total number of iterations.

7.4.2. Details of PSO based design space exploration

To explore the optimal design space solution as an output using PSO (as shown in Fig. 7.3), the primary inputs to the PSO block are inertia weight (τ), acceleration coefficient (t1, t2), terminating criteria (C_T) and population size (P_S). Besides the secondary input is the global best resource 'S_{Gb}' (corresponding to minimum security-design cost value obtained in initial iteration). Whereas the output of the PSO block is the low-cost resource configuration as shown in Fig. 7.3.

Table 7.1 Qualitative comparison between the security approaches

S.No.	Characteristics/Parameters	Biometric [40], [41]	IP Watermarking [32]	Digital signature [39]
1.	security mechanism	Natural biometric features (minutiae points or facial nodal points)	Signature and encoding rules	RSA encryption, SHA-512
2.	Counterfeit detection control	strong	less	less
3.	Implementation complexity	less	more	More
4.	Proof of IP ownership by a genuine owner	seamless	difficult	arduous
5.	Vulnerability and replicability	Almost Impossible	yes	Yes

In the first phase of PSO number of particles are chosen and there encoding has been performed. In order to do so, first particle's position is initialized by minimum hardware resources: S_1= ($P1^{min}$, $P2^{min}$). Where P1 and P2 are the hardware resource types, adder(s) and multiplier(s) respectively (available in the library). The second particle's position is initialized by maximum resources: S_2= ($P1^{max}$, $P2^{max}$). The third particle's position (S_3) is initialized by average of maximum and minimum resource values. The rest of the particle's position (S_4…S_n) is initialized by the following equation:

$$S_{id}=(\alpha+\beta)/2 \pm \gamma \qquad (7.1)$$

Where 'S_{id}' represents the current position of i^{th} particle in dimension 'd', 'α' is the minimum resource value and 'β' is maximum resource value and 'γ' is any random number between 'α' and 'β'. For example, in 8-point DCT, particle positions are, S_1= (1,1), S_2= (1,8), S_3= (1,4) and so on.

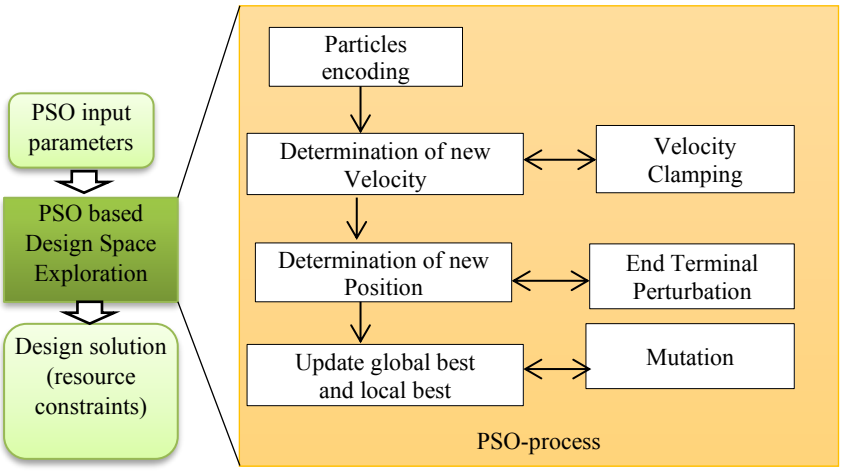


Fig.7.3. Details of the PSO based design space exploration

In the next phase, new velocity of the particles (initial velocity=0) has been determined by using the equation below:

$$v_{id}^{+} = \tau.v_{id}+t1x1(S_{lbi}-S_{id})+t2x2(S_{Gb}-S_{id}) \quad (7.2)$$

Where 'vid' and vid^{+} represents the velocity of i^{th} particle d^{th} dimension in previous and next iteration respectively and x1, x2 are random numbers between [0,1]. Further, S_{lbi} represents local best solution of the ith particle. The component '$\tau.v_{id}$' is called inertia component which prevents drastic change in the direction of particle. The other component 't1x1. (S_{lbi}-S_{id})' is called cognitive component which represents the tendency of a particle to return to its individual best resource configuration from the past. The component 't2x2. (S_{Gb}-S_{id})' is called the social component which direct the particle towards the best resource configuration found by all its neighbors, including itself. If the new velocity outreaches the boundary, then velocity clamping has been performed to control the excessive exploration drift. It helps particles to stay in the design space by taking the step size sensibly. In the next phase new position of the particles has been determined by adding the new velocity to the previous position of the resources. Furthermore, if the new position of the particle outreaches the boundary space, then the end terminal perturbation has been performed in order to keep the particle in its design space. Subsequently, in the next phase local best solution of each particle has been updated (if the solution with the minimum security-design cost is found) and based on that global best solution (S_{Gb}) is also updated in each iteration. Mutation has been performed on every local best resource (S_{lb}) and S_{Gb} is also updated. Following process continues until terminating criteria 'C_T' (if the solution executes for a certain number of times or solution converges and does not get updated for next 10 iterations) is met. The low-cost RTL solution (global best resource constrains) is explored by the PSO by converging the initial solution to the global minima.

7.4.3. Details of HLS scheduling, allocation and binding process: Demonstration on 8-point DCT

Based on the output of the PSO-DSE (resource constraints) scheduling of the DFG of the respective DSP applications has been performed in each iteration

(up to i<T). List scheduling technique has been used for scheduling. The output of the scheduling is the number of control steps (CS) using multiplier (x) and the control steps using the adder only (y). It is used for determining the design latency (L_d).

Initial allocation of the hardware resources (registers) has been performed to each operation as shown in Fig. 7.4. Based on that initial register allocation table (pre-embedding, as shown in Table 7.2) has been generated. The register allocation table comprises of the following details: number of registers required for accommodating the storage variables of the design, number of control steps required to schedule the design and the position of storage variables based on their dependency information corresponding to the functional behavior of the design. As evident from Table 7.2, the number of required registers corresponding to 8-point DCT are eight, where each register is designated using a different color name. Further, the number of required control steps are nine (C_s0-$Cs8$). Subsequently, binding has been performed to determine the multiplexer and demultiplexer information during resource sharing.

7.4.4. Details of signature embedding process

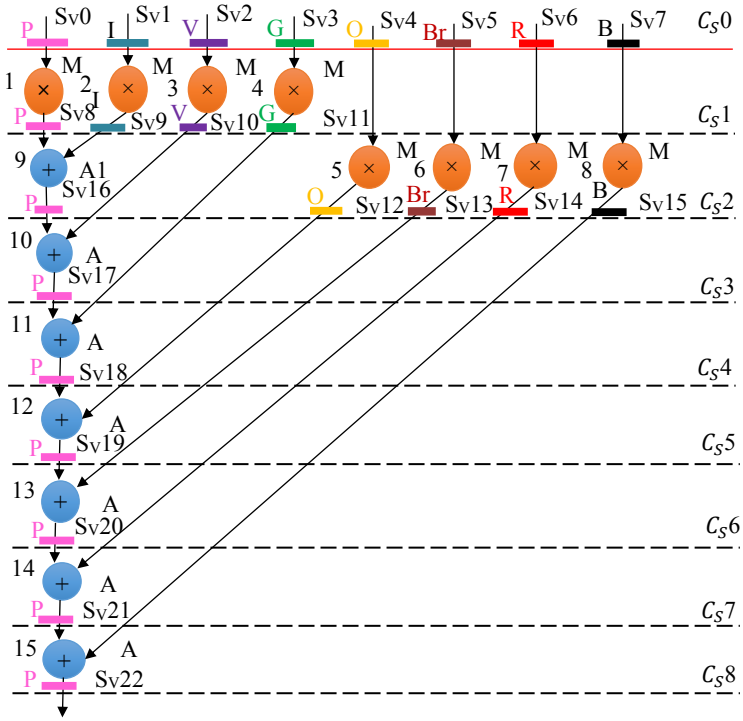


Fig. 7.4 Scheduled DFG of 8-point DCT core using one adder (A) and four multipliers (M)

As shown in Figure 7.1, the inputs of the signature (digital evidence) embedding block are the signature generation algorithm and signature strength (size) chosen by the IP designer and the scheduled and allocated/binded DFG of the DSP application. The output of the signature embedding (SE) block is the signature embedded design. The scheduled DFG design corresponding to the algorithmic description/transfer function of 8-point DCT is shown in Fig. 7.4, where, (S_V0 to S_V22) are the storage variables (comprising the inputs of the design), different colors indicate the number of registers and (C_s0-$Cs8$) are the control steps required for execution. The scheduled DFG of 8-point DCT application is based on one adder and four multiplier resources.

In case of the hardware watermarking approach, for performing the embedding of generated watermark signature into the target design, the details of embedding are discussed below:

Assuming that the IP designer chosen watermark signature based on the variables (i, I, T, !) is 16 bit long (for the sake of brevity). However, the discussed approach is easily scalable as a function of the signature and design size (IP designer can also select a signature of larger size). Let us consider the 16-bit watermark signature as follows:

!, i, I, i, !, T, i, !, i, !, I, i, !, i, I, I

The security constraints corresponding to above watermark signature are derived using designer specific encoding rule (as discussed earlier in subsection 7.3.1). Therefore, the resulting security constraints corresponding to chosen watermark signature, are shown as follows:

for signature bit '!' → (S_V0-S_V1), (S_V0- S_V2), (S_V0- S_V3), (S_V0- S_V4), (S_V0-S_V5),

for 'i→ (S_V2, S_V3), (S_V2, S_V5), (S_V2, S_V7), (S_V2, S_V11), (S_V2, S_V13), (S_V2,

Table 7.2 Register allocation of 8-point DCT (*pre-embedding*)

CS	Pink	Indigo	Violet	Green	Orange	Brown	Red	Black
C_S0	S_V0	S_V1	S_V2	S_V3	S_V4	S_V5	S_V6	S_V7
C_S1	S_V8	S_V9	S_V10	S_V11	S_V4	S_V5	S_V6	S_V7
C_S2	S_V16	--	S_V10	S_V11	S_V12	S_V13	S_V14	S_V15
C_S3	S_V17	--	--	S_V11	S_V12	S_V13	S_V14	S_V15
C_S4	S_V18		--	--	S_V12	S_V13	S_V14	S_V15
C_S5	S_V19	--	--	--	--	S_V13	S_V14	S_V15
C_S6	S_V20	--	--	--	--	--	S_V14	S_V15
C_S7	S_V21	--	--	--	--	--	--	S_V15
C_S8	S_V22	--	--	--	--	--	--	--

S_V17),

for 'I'→ (S_V2, S_V4), (S_V2, S_V6), (S_V2, S_V8), (S_V2, S_V10) and

for 'T'→ (S_V1, S_V2).

Next, these generated hardware security constraints are covertly embedded into the register allocation phase of HLS process. In order to do so, local alteration of the registers for re-allocation of the storage variables into the register allocation table, is performed based on the rule- '*both storage variables of any security constraint pair cannot be allocated into the same register*'. Thus, the register allocation table post embedding the watermark signature-driven secret hardware security constraints into 8-point DCT design is shown in Table 7.3, where the storage variables marked in red represent the embedded constraints post performing the local alteration based on the embedding algorithm. These embedded secret watermark constraints acts as secret digital evidence for enabling the detective control (security) against IP piracy and nullifying fraudulent ownership claim.

In the case of an encrypted digital signature-based approach, assuming that the IP designer's chosen encrypted digital signature is 16-bit long (for the sake of brevity). However, the discussed approach is easily scalable as a function of the signature and design size (IP designer can also select a signature of larger size). Let us consider the 16-bit encrypted digital signature as follows:

1,1,0,1,0,1,0,1,0,1,0,1,1,0,0,1

The security constraints corresponding to above encrypted digital signature are derived using designer specific encoding rule (as discussed earlier in subsection 7.3.2). Therefore, the resulting hardware security constraints corresponding to chosen digital signature, are as follows:

Table 7.3 Register allocation of 8-point DCT (*post-embedding, in case of IP watermarking approach*)

CS	pink	Indigo	violet	green	orange	brown	Red	black
C_S0	S_V 0	S_V1	S_V2	S_V3	S_V4	S_V5	S_V6	S_V7
C_S1	S_V8	S_V9	S_V11	S_V10	S_V4	S_V5	S_V6	S_V7
C_S2	S_V16	--	S_V11	S_V10	S_V12	S_V13	S_V14	S_V15
C_S3	S_V17	--	S_V11	--	S_V12	S_V13	S_V14	S_V15
C_S4	S_V18		--	--	S_V12	S_V13	S_V14	S_V15
C_S5	S_V19	--	--	--	--	S_V13	S_V14	S_V15
C_S6	S_V20	--	--	--	--	--	S_V14	S_V15
C_S7	S_V21	--	--	--	--	--	--	S_V15
C_S8	S_V22	--	--	--	--	--	--	--

For bit ‘0’- (Sv2, Sv3), (Sv2, Sv5), (Sv2, Sv7), (Sv2, Sv11), (Sv2, Sv13),…

For bit ‘1’- (Sv0, Sv2), (Sv0, SvV4), (Sv0, Sv6), (Sv0, Sv8), (Sv0, Sv10), (Sv0, Sv12), , ……, (Sv0, Sv18).

Next, these generated hardware security constraints are embedded into design during the register allocation phase of HLS. Thus, the register allocation table of 8-point DCT design post embedding the encrypted digital signature-driven secret hardware security constraints is shown in Table 7.4, where the storage variables marked in red represent the embedded constraints post performing the local alteration based on the embedding algorithm.

Further, in the case of a facial biometric-based security algorithm, for performing the embedding of generated facial signature into the target design, the details of embedding are discussed below:

Assuming that the IP designer's chosen facial biometric signature is 16-bit long (for the sake of brevity). Let us consider a 16-bit facial biometric signature as follows:

1,0,1,1,1,1,1,1,0,0,0,0,0,1,0,1

Therefore, the generated security constraints corresponding to chosen facial biometric signature, are as follows:

(Sv1,Sv3),(Sv0,Sv2),(Sv1,Sv5),(Sv1,Sv7),(Sv1,Sv9),(Sv1,Sv11),(Sv1,Sv13),(Sv1,Sv15),……,(Sv1,Sv19).

Next, these generated hardware security constraints are embedded into target design based on the same rule stated earlier. Thus, the register allocation table of 8-point DCT design post embedding the facial biometric signature driven secret hardware security constraints is shown in Table 7.5, where the storage variables marked in red represent the embedded constraints post performing the local alteration based on the embedding algorithm. These, embedded facial security constraints acts as digital evidence for enabling the detective control against IP piracy. Further, as the facial security constraints also associate the unique facial identity of IP vendor. This, therefore, enables the definitive proof of IP ownership for an original IP vendor.

7.4.5. Security-design cost tradeoff fitness function

The primary inputs to the security–design cost fitness function are the signature embedded design (use to compile area, latency and security constraints) and the library. Based on the embedded security constraints, security metric in terms of embedded constraints size of the corresponding signature 'S_m^1' can be determined as:

$$\text{Security metric } (S_m^1) = L/M \tag{7.3}$$

Where 'L' represents number of embedded security constraints and 'M' represents total possible security constraints (corresponding to security methodology). The number of embedded security constraints 'L' is a measure of hardware security in terms of the proof of digital evidence against piracy (and IP ownership) as well as tamper tolerance ability. This is because higher the number of security constraints embedded, lower is the probability of coincidence (indicating stronger digital evidence) and higher is the tamper tolerance.

Furthermore, the design cost (Z_c) of a particular DSP application is determined using metric [31], [32], [36]-[40]:

$$Z_c(S_{id}) = W_a.(K_d/K_m) + W_l.(T_d/T_m) \tag{7.4}$$

Where, 'K_d' and T_d refers to the area and latency of the target design, 'K_m'

Table 7.4 Register allocation of 8-point DCT application (*Post embedding in case of encrypted hash-based approach*)

C_T	pink	Indigo	violet	green	orange	brown	Red	black
C_T0	V_S0	V_S1	V_S2	V_S3	V_S4	V_S5	V_S6	V_S7
C_T1	V_S9	V_S8	V_S10	V_S11	V_S4	V_S5	V_S6	V_S7
C_T2	--	V_S16	V_S10	V_S11	V_S12	V_S13	V_S14	V_S15
C_T3	V_S17	--	--	V_S11	$V_S 12$	V_S13	V_S14	V_S15
C_T4	--	V_S18	--	--	$V_S 12$	V_S13	V_S14	V_S15
C_T5	V_S19	--	--	--	--	V_S13	V_S14	V_S15
C_T6	V_S20	--	--	--	--	--	V_S14	V_S15
C_T7	V_S21	--	--	--	--	--	--	V_S15
C_T8	V_S22	--	--	--	--	--	--	--

Table 7.5 Register allocation of 8-point DCT application (*Post embedding in case of facial biometric approach*)

C_T	pink	Indigo	violet	green	orange	brown	Red	black
C_T0	V_S0	V_S1	V_S2	V_S3	V_S4	V_S5	V_S6	V_S7
C_T1	V_S9	V_S8	V_S10	V_S11	V_S4	V_S5	V_S6	V_S7
C_T2	V_S16	--	V_S10	V_S11	V_S12	V_S13	V_S14	V_S15
C_T3	V_S17	--	--	V_S11	$V_S 12$	V_S13	V_S14	V_S15
C_T4	V_S18	--	--	--	$V_S 12$	V_S13	V_S14	V_S15
C_T5	V_S19	--	--	--	--	V_S13	V_S14	V_S15
C_T6	V_S20	--	--	--	--	--	V_S14	V_S15
C_T7	V_S21	--	--	--	--	--	--	V_S15
C_T8	V_S22	--	--	--	--	--	--	--

represents maximum design area (evaluated using maximum available hardware resources ($P1^{max}$, $P2^{max}$)). 'Tm' represents the maximum latency (evaluated based on the most serial execution using minimum possible hardware resources ($P1^{min}$, $P2^{min}$)).W_a and W_l are the weighting factors for area and latency respectively. Library file (A 15nm open-cell library [86]) contains the following information such as: area of the adder, multiplier and register unit and delay (time consumed) of the adder and multiplier unit. Based on that, area of the design ('K_d') as shown in equation (5) and latency ('T_d') as shown in equation (6) has been determined [82], [83].

Design area (K_d) = n*(area of adder) + m*(area of multiplier) +P*(area of register) (7.5)

Where 'n' indicates the number of adders and 'm' indicates the number of multipliers.

Design latency (T_d) = (#CS using multiplier*delay of 1 multiplier) + (#CS using adder only*delay of 1 adder) (7.6)

Subsequently, the security-design cost tradeoff fitness value for each particle can be determined using the equation below:

$$f_{S-C} = \mathrm{W_s}(\mathrm{S_m^1}) + \mathrm{W_d}(Z_c) \quad (7.7)$$

Where, W_s and W_d indicate weight of security and design cost in security-design cost trade-off function. Based on the fitness value of each particle, the global best resource configuration is determined. The particle with the minimum fitness function value (minimum security design cost value) is declared as the fittest or global best resource configuration among all other particles in each iteration. This process is followed in each iteration until the process gets converged or the C_T is met. In the end, low-cost RTL solution for signature-based security methodologies using PSO for different DSP applications is obtained.

7.5. Results and analysis

The results of the presented approach include (i) analysis of low-cost architectural resource configuration using PSO, (ii) impact of signature strength on security-design cost fitness value, and (iii) register count of the

DSP IP core and (iv) security parameter such as probability of co-incidence for various security methodologies for varying (scalable) signature strength. The experimental results of the proposed security-design cost tradeoff methodology have been discussed and analyzed in detail in chapter 9 of this thesis.

7.6. Summary

This chapter discussed a novel approach for the exploration of security-design cost trade-off for signature based security algorithms for DSP hardware IPs. It provides optimal design architectural solutions for secured IP cores used in consumer electronics (CE) systems using PSO-based design space exploration. The proposed approach considers three different hardware security algorithms based on facial biometrics, encrypted-hash and watermarking for integration with the PSO-DSE framework for exploring the hardware architecture tradeoffs of security-design cost. Experimental results in terms of security, design cost (area, delay), exploration time and other vital parameters are obtained that offer IP designer and SOC integrator to employ optimal secured and robust IP cores for integration in modern electronic/automated system designs.

Chapter 8

Symmetrical Protection of Ownership Right's for IP Buyer and IP seller using Facial Biometric Pairing

This chapter presents a novel methodology for enabling the protection of IP rights of IP buyer and seller. In the present scenario where the development of smart cities and deployment IoT enabled devices is thriving, the demand of hardware accelerators is increasing. Therefore, in order to tradeoff the supply and demand, these hardware accelerators are developed and delivered by the third-party vendors (sellers), this scenario indeed may lead the major security concerns to end consumer along with IP buyer. Further, an untrustworthy IP buyer may also falsely claim the ownership rights (post receiving the IP). On the other hand, an IP vendor may also distribute the illegally copies of IP cores. Therefore, the security of these hardware accelerators (IP cores) along with the rights of IP vendor and buyer simultaneously, is of utmost importance before their integration into system on chips (SoCs) of consumer electronics (CE) systems.

Outline of the chapter is as follows. The first section formulates the problem. The second section discusses the process for generating the secured design through embedding facial biometric of IP buyer under following subsections: threat model and motivation, process for generating the security constraints for facial biometric of IP buyer, process for generating the signature embedded design corresponding to facial biometric of IP buyer. Further, the third section discuss the proposed approach under following subsections: process for generating the security constraints for facial biometric of IP seller, process for generating the signature embedded design corresponding to facial biometric of IP seller. The fourth section discuss the process for nullifying false claim of IP rights and detecting IP piracy. Finally, the fifth section summarizes the chapter.

8.1. Problem Formulation

Given the data-intensive hardware IP core in the form of transfer function/behavioral description, library, resource constraints and facial

biometric of IP buyer and seller along with the objective of protecting their IP rights symmetrically.

8.2. Process for generating the secured design through embedding facial biometric of IP buyer

The proposed approach presents a robust security methodology using facial biometrics-based approach for protecting ownership rights in the hardware accelerators (IPs) used in CE systems. The proposed approach ensures the protection of the rights of IP seller against the threat of fraudulent claim of ownership from IP buyer. On the other hand, it also protects the rights of IP buyer against the illegal distribution of IP cores, thereby offering symmetric security to both parties e.g., IP seller and IP buyer. As shown in Fig. 8.1, firstly, the hardware security constraints corresponding to facial biometric features of original IP buyer are generated. Subsequently, these security constraints are implanted into the baseline design during register allocation phase of behavioral synthesis. Next, the hardware security constraints corresponding to facial biometric features of original IP seller are generated. Subsequently, seller's security constraints are embedded into the design obtained post embedding buyer's security constraints. The proposed embedding process ensures the insertion of the security constraints of IP seller and IP buyer uniquely without affecting the functionality of the design. Therefore, while performing the detection of illegal IP cores and to nullify false claim of ownership, both entities (IP supplier and user) can verify their secret mark distinctly. Thus, proposed methodology offers the robust symmetrical protection of hardware accelerators by integrating the non-replicable and unique facial biometric information of IP buyer and seller. The detailed process of proposed security methodology has been presented in subsequent subsections.

8.2.1. Threat model and motivation

The proposed methodology handles the security threats from the perspective of IP seller and IP buyer (the two main entities of IP supply chain).

Threat to IP seller: an IP buyer may falsely claim the IP ownership rights, post receiving the IP. Therefore, a robust security mechanism must be integrated in order to safeguard the rights of IP seller. Additionally, the embedding of security into design should not impact its functionality and also the resulting design cost should be as minimal as possible.

Threat to IP buyer: an untrustworthy IP seller may distribute/ sell the illegal

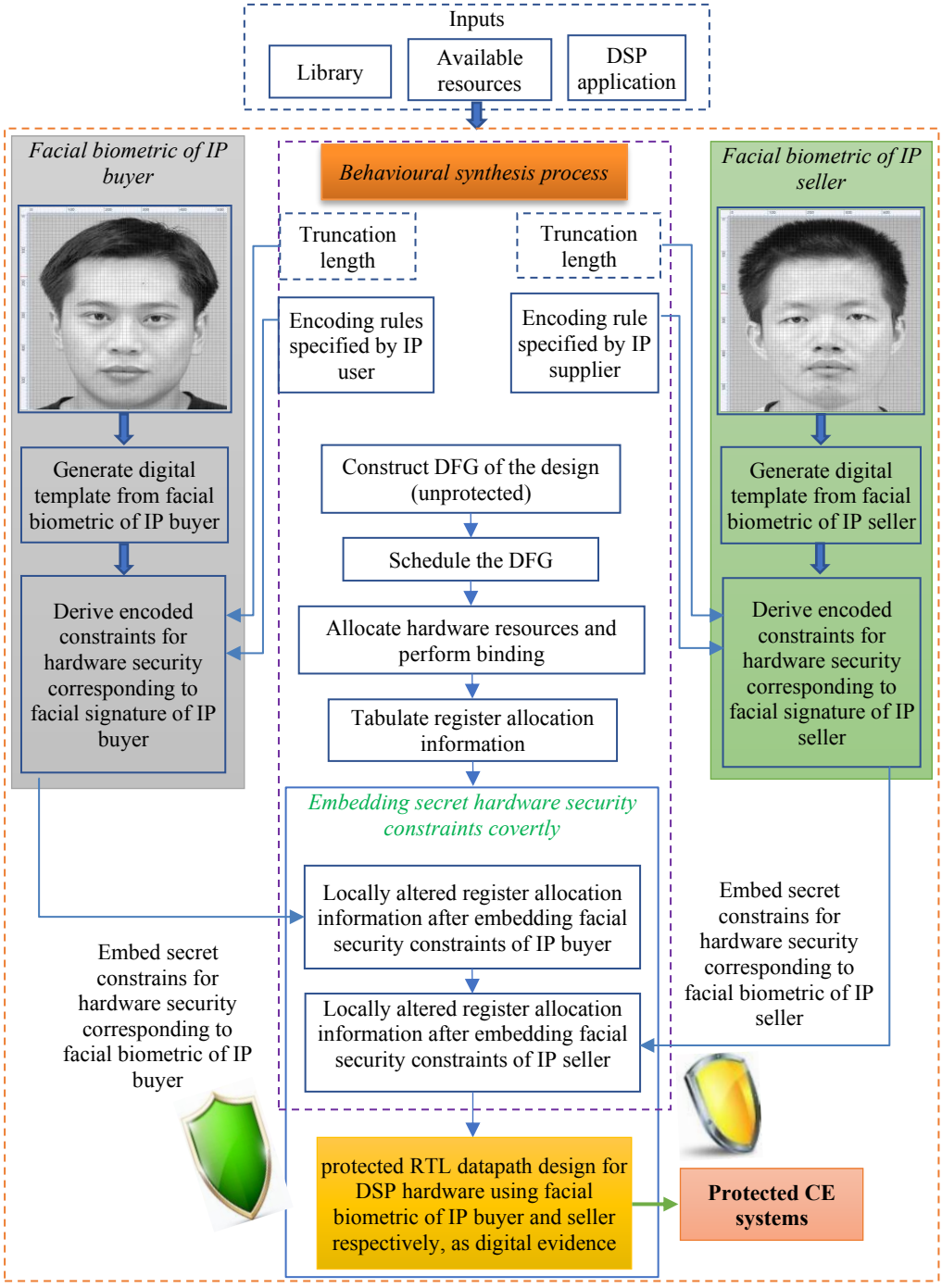


Fig. 8.1. The design flow corresponding to the proposed security approach using facial biometric

copies of custom IP (designed based on the IP buyer specification). This may lead to illegal use of IPs. It must be prohibited in case if some hardware accelerator is designed for some specific purpose (mission critical applications) corresponding to a specific IP buyer.

The proposed methodology embeds the facial biometric based digital signature driven security constraints into the design during behavioral synthesis. Biometric based protection offers the robust security in the form of embedded signature as the unique facial features driven digital signature is not replicable unlike other hardware security techniques like hardware steganography and watermarking. The embedding of security constraints at behavioral synthesis level costs lesser design overhead and results lesser implementation complexity as compared to enabling the security at lower level of the design abstraction. However, it also protects the lower-level design as the embedded signature during behavioral synthesis is distributed throughout the design or subsequent levels. Therefore, the embedded facial biometric signature (in pairing of IP buyer and IP seller) not only offers minimal design cost, low complexity, non-replicable security but also provides the protection of the rights of IP user and IP supplier simultaneously.

8.2.2. Process for generating the security constraints for facial biometric of IP buyer

The process for generating the hardware security constraints corresponding to original IP buyer, is assimilated through following steps:

a) first the facial image is captured using imaging device.

b) map the captured image of IP buyer into specific grid size and spacing which helps in obtaining the accurate facial feature dimensions.

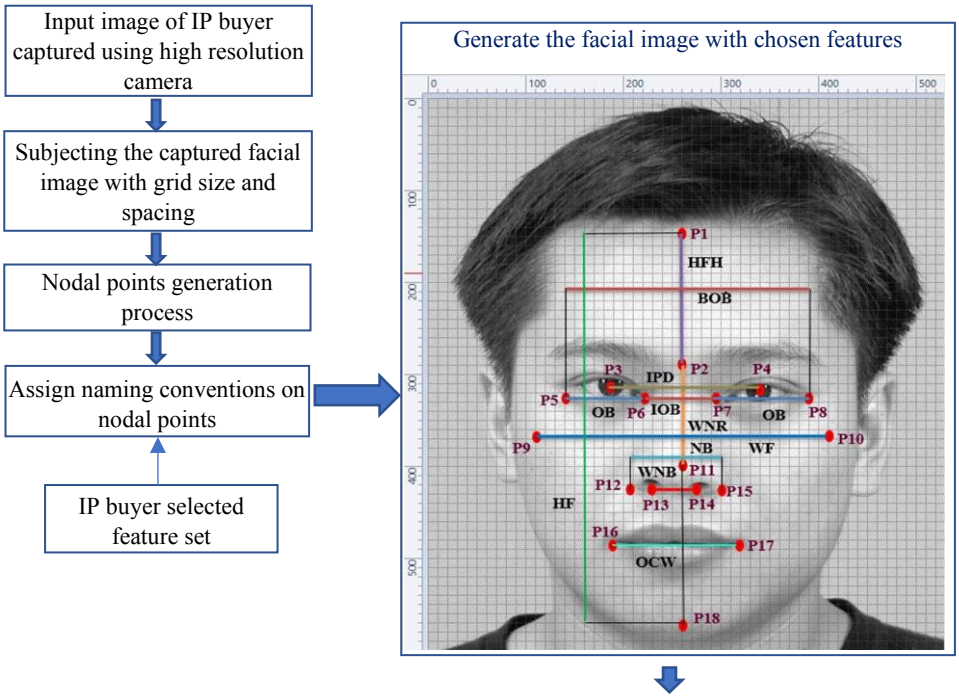
c) designate the nodal point on the resulting facial image. The nodal points are marked in red as shown in Fig. 8.2.

d) based on IP buyer selected set of facial features, perform the assignment of naming convention on the designated nodal points.

e) generate the facial image for IP buyer corresponding to chosen facial features in step d).

f) for each of the selected facial feature (comprising of two nodal points) determine the co-ordinate points.

g) subsequently, evaluate the feature dimension corresponding to each feature. The same is performed using Manhattan equation.



Calculating the feature dimensions between nodal points of chosen features

Facial features	Naming convention of points	Co-ordinates (x1, y1)- (x2, y2)	Feature dimension evaluated using Manhattan equation (|x2-x1|+|y2-y1|)	Binarize feature magnitude
Face Height (HF)	(P1) & (P18)	(260, 145) & (260,570)	425	110101001
Height of Forehead (HFH)	(P1) &(P2)	(260, 145) & (260, 290)	145	10010001
Nasal Ridge Width (WNR)	(P2) & (P11)	(260, 290) & (260, 400)	110	1101110
Inter Pupillary Distance (IPD)	(P3) & (P4)	(185, 315) & (340, 315)	155	10011011
Ocular Breadth (OB)	(P5) & (P6)	(140, 325) & (220, 325)	80	1010000
Bio- Ocular Breadth (BOB)	(P5) &(P8)	(140, 325) & (390, 325)	250	11111010
Inter – Ocular Breadth (IOB)	(P6) & (P7)	(220, 325) & (295, 325)	75	1001011
Face Width (WF)	(P9) &(P10)	(110, 370) & (410,370)	300	100101100
Nasal Breadth (NB)	(P12) & (P15)	(205, 425) & (300, 425)	95	1011111
Nasal Base Width (WNB)	(P13) & (P14)	(230, 425) & (275, 425)	45	101101
Oral Commissure Width (OCW)	(P16) & (P17)	(190, 485) & (320, 485)	130	10000010

Fig. 8.2. Process for generating biometric information corresponding to facial features of IP buyer

h) convert the magnitude of each facial feature into their binarized form.

i) Next, in order to generate the digital biometric template corresponding to authentic IP buyer, the binarize signature corresponding to each facial feature is concatenated. However, IP buyer can generate numerous signatures of particular strength and combinations, depending upon the different possible concatenation orders.

The process of generating the facial biometric signature corresponding to IP buyer is demonstrated in Fig. 8.2. Where, based on the captured input facial image of IP buyer, specific grid size, number of facial features chosen and their concatenation order and truncation length, final biometric digital template is generated. The above process has been implemented using [85].

j) Subsequently, the generated signature is converted into corresponding hardware security constraints based on the encoding rules specified by the original IP buyer.

For example, if IP buyer selects the following facial features among the total specified features, in the following concatenation order such as:

"HF→WNR→OB→IOB→NB→OCW→WNB→WF→BOB". Then, the generated facial signature will be as follows:

"11010100111011101010000100101110111111000001010110110010110011111010".

Subsequently, hardware security constraints are generated based on final truncation length, target DSP design, and by using the following encoding rule specified by the IP buyer.

- For signature bit '0', implant the security constraints between the storage variable pairs where both the variables (P) are even.
- For bit '1', implant the security constraints between the storage variable pairs where the first variable is 0 and the second variable can be of any integer value (excluding the already available pairs).

In this paper, the methodology of protection of the rights of both IP buyer and IP seller is demonstrated using IIR filter (which contains 27 storage variables

to perform the computation). The IIR DSP benchmark has been adopted from (pp.255-257) [83]. Therefore, the resulting hardware security constraints corresponding to IP buyer are:

For signature bit '0'→(P0-P2), (P0-P4), (P0-P6), (P0-P8), (P0-P10), (P0-P12), (P0-P14), (P0-P16), (P0-P18), (P0-P20), (P0-P22), (P0-P24), (P0-P26), (P2-P4), (P2-P6),..……...,(P4-P14),

For signature bit '1'→(P0-P1), (P0-P3), (P0-P5),…..(P0-P25).

Subsequently, these hardware security constraints are embedded into design by performing the local alteration of the storage variables among the available registers, during register allocation phase of behavioural synthesis. The following rule is followed while embedding the constraints into target design:

any two storage variables of the same generated pair cannot be assigned to same register available as it will result into conflict as same register cannot be assigned to two storage variables at the same time. However, in case if it is not possible to accommodate the conflict then a new register is allocated. However, it may lead to design area and delay overhead, if required.

8.2.3. Process for generating the signature embedded design corresponding to facial biometric of IP buyer

In order to generate the embedded design with IP buyer signature constraints, following steps are followed:

a) firstly, we construct the DFG of the input design by following the dependency information of operations.

b) next, schedule the target DSP design based on the resource constraints specified by IP designer. The scheduled DFG is shown in Fig. 8.3.

c) allocate the hardware resources available in the module library, to respective operations (multiplication, addition and subtraction) and perform there binding.

d) construct the register allocation table corresponding to input design. Register allocation table comprises of the details of required registers (R1 to

R14), available storage variables (P0 to P26) in the design and their assignment to particular register and required control steps (I0 to I7).

e) now, perform the embedding of the generated hardware security constraints earlier (corresponding to original IP buyer, based on the selected strength of facial biometric signature and specified encoding rules), by locally altering the register allocation information. The register allocation information post embedding the facial biometric signature of IP buyer is shown in Table 8.1 Where, the variables marked in blue color indicates the updated position of variables post embedding the security constraints based on the embedding rules. The variables marked in yellow color indicates the previous position of storage variables before embedding the IP buyer based facial signature driven security constraints.

8.3. Process for generating the secured design through embedding facial biometric of IP seller

The detailed process of the proposed security methodology corresponding to generating secured design through IP seller facial biometrics has been presented in subsequent subsections.

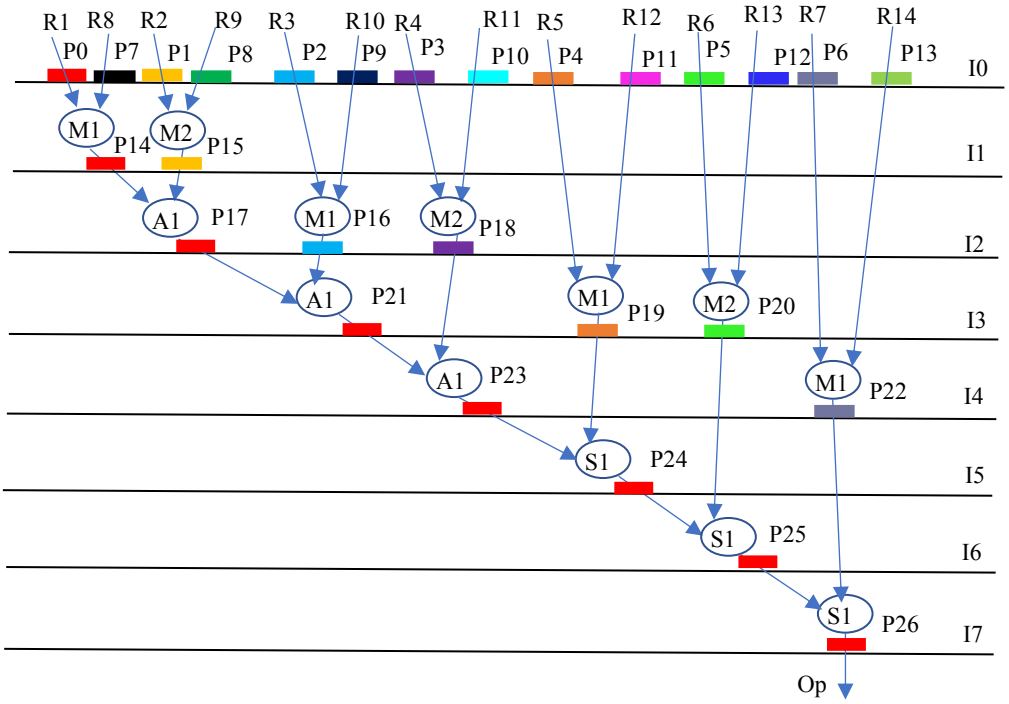


Fig. 8.3. Scheduled DFG of IIR filter corresponding to resources one adder, two multipliers and one subtractor

8.3.1. Process for generating the security constraints for facial biometric of IP seller

In order to generate the security constraints corresponding to facial biometric image of IP seller, the following steps as discussed in subsection 8.2.2 from a) to i) has been followed. The facial image of IP seller with chosen features set is shown in Fig. 8.4. However, an IP seller may choose specific value of security parameters corresponding to his facial biometric image such as specific grid size to generate facial signature, number of facial features, their concatenation order and specific truncation length. For example, if the IP seller selects following facial features among the total specified features, in the following concatenation order such as: "OB→IOB→NB→OCW→WF→BOB→HF→WNB". Then the generated facial signature will be as follows: "10101011011111101111111111101100101100100000100110011010101000" . Subsequently, in order to generate the security constraints, following encoding rule has been followed such as:

- Corresponding to facial signature of IP seller, for signature bit '0' implant the security constraints between the storage variable pairs where both the variables are odd.

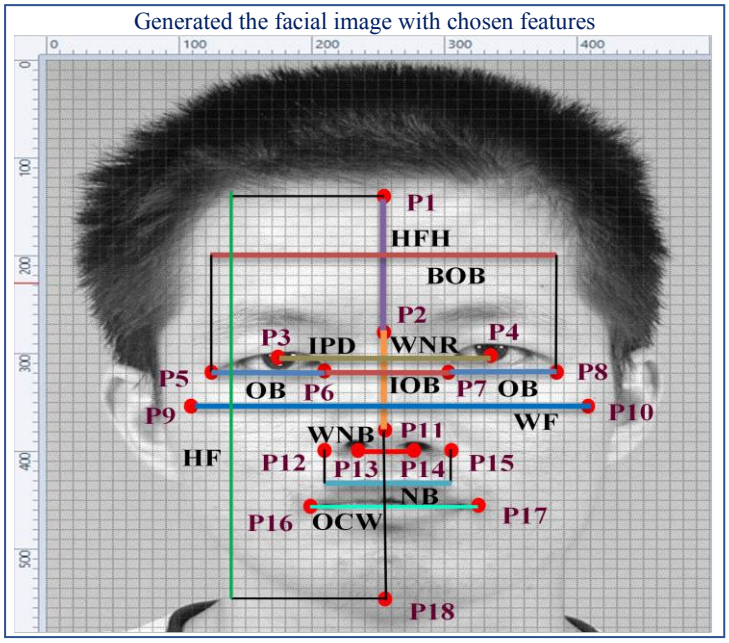


Fig. 8.4. Facial image with selected facial features corresponding to IP supplier

- For signature bit ‘1’ implant the security constraints between the storage variable pairs where both the storage variables are primes.

Therefore, the resulting hardware security constraints corresponding to IP seller are:

For signature bit ‘0’→(P1-P3), (P1-P5), (P1-P7), (P1-P9), (P1-P11), (P1-P13), (P1-P15),……..…., (P5-P11),

For signature bit ‘1’→(P2-P3), (P2-P5), ……..,.. (P17-P23).

Subsequently, these hardware security constraints are embedded into design, obtained post embedding the security constraints corresponding to the facial biometric of the original IP buyer (as discussed earlier in subsection 8.2.3).

8.3.2. Process for generating the signature embedded design corresponding to facial biometric of IP seller

The embedding process at the IP seller end takes the following inputs e.g., register allocation information post embedding IP buyer facial signature driven secret security constraints and the newly generated security constraints corresponding to the facial features of IP seller. Thereafter, the embedding process is performed. The resulting register allocation information post embedding IP seller driven facial signature is shown in Table. 8.1. Where the variables marked in red color indicate the changes due to local alteration, post embedding the security constraints based on the embedding rules (as discussed

Table 8.1 Register allocation information post embedding the facial biometric driven security constraints corresponding to IP buyer and seller

NOTE: among the total storage variables of the design, variables marked in blue color represents their new position corresponding to older position of variables (marked in yellow), post embedding the facial constraints corresponding of IP buyer. variables marked in red color represents their new position, post embedding the facial constraints of IP seller into the resulting design post embedding of the buyer constraints.

I	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14
0	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
1	P15	P14	P2	P3	P4	P5	P6	P15	--	P9	P10	P11	P12	P13
2	P17	P17	P16	P16	P4	P5	P6	P18	P17	--	--	P11	P12	P13
3	P21	P21	P21	P18	P19	P20	P6	P18	--	--	--	--	--	P13
4	P23	P23	--	--	P19	P20	P22	--	--	P23	--	--	--	--
5	P24	P24	--	--	--	P20	P22	--	--	--	--	--	--	--
6	P25	P25	P25	--	--	--	P22	--	--	--	--	--	--	--
7	P26	P26	--	--	--	--	--	--	--	--	--	--	--	--

earlier). Thus, the proposed approach ensures the protection of the rights of both the IP buyer and IP seller by embedding their facial biometric signature into the design during behavioural synthesis. Further, it is apparent form the Table 8.1 that no extra storage element was needed while performing the embedding of all the security constraints, corresponding to facial biometrics of both IP buyer and IP seller symmetrically.

8.4. Process for Nullifying false claim of IP rights and detecting IP piracy

In case if either a rogue IP buyer fraudulently claims ownership or IP seller distributes illegal copies, the proposed approach provides seamless verification of IP rights to both the parties, by detecting the unique biometric signature corresponding to each. In order to do so firstly, the signature embedded design is being inspected to reconstruct the controller and extract the embedded signature. In case if an IP buyer (or adversary) is falsely claiming the ownership, then IP seller by performing the matching of the regenerated signature (along with its corresponding security constraints from the design) with the embedded security constraints into target design, can easily prove his/her IP ownership.

Similarly, in order to trace any pirated or illegal copies made by an IP seller, the original IP buyer can prove his/her IP rights over the obtained IP core by extracting his/her biometric signature from the IP RTL design and match with the original embedded security constraints of his/her biometric signature. Only the original IP buyer will be able to match the security constraints successfully to prove his/her IP rights. Further, the covert details of security parameters used in the proposed approach to derive facial biometric security constraints is known to only genuine IP buyer and seller thereby ensuring robust security of the target design.

Further, an adversary present in untrustworthy design house may attempt to pirate the IP cores without the knowledge of IP seller (vendor). The proposed approach enables the robust detective control against pirated IP cores by the integration of non-replicable and unique facial biometric-driven secret security constraints of IP vendor. While performing the IP piracy detection, the

presence of authentic security constraints corresponding to the facial biometric signature of the original IP vendor is verified. In order to do so, the embedded constraints from the target design under test are extracted. Subsequently, if they do completely match with the security constraints of the original IP vendor, then the target IP design is considered as genuine otherwise, it is a pirated design.

8.5. Results and analysis

The proposed approach achieves the following i) symmetrical protection of IP rights of seller and buyer at zero design cost overhead ii) lesser probability of coincidence (Pc) than state-of-the-art approaches. Further, the embedded signature of the IP vendor can also be used to detect pirated IP cores. The experimental results of the approach have been discussed and analyzed in chapter 9 of this thesis.

8.6. Summary

This chapter presented a robust symmetric security methodology to enable the protection of the ownership rights of both the IP buyer and IP seller. The proposed approach exploited the unique facial biometrics of both parties for the same. Furthermore, it ensures the integration of only authentic IP cores into CE systems, thereby safeguarding the end consumers also. The proposed work presented stronger protection of hardware IPs (in terms of lower *Pc* value) while incurring zero design overhead.

Chapter 9

Experimental Results and Analysis

The experimental results and analyses of the proposed hardware security techniques for ensuring the security (in terms of IP core protection and detective control) of data intensive hardware/IP cores are presented in this chapter. The results have been calculated for various data intensive DSP and multi-media benchmarks [81]-[84].

9.1. Results and analysis: Contact-less palmprint biometric for securing DSP co-processors used in CE systems against IP piracy

The experimental results of the proposed contact-less palmprint biometric methodology for securing DSP-coprocessors discussed in section 3 are analyzed and discussed in this section. A 15 nm open cell library was used to calculate different parameters such as design area and latency [86]. The proposed method allows capturing of 'n' palmprint images where the value of 'n' depends on the IP designer's choice. However, during security constraints extraction and embedding process in an IP core, only a single palmprint image is used at a time. The choice of the palmprint image again depends on the IP vendor. The palmprint size dataset tested in our approach varies between 7 to 262 digits. The proposed approach has therefore been tested on wide variety of palm image sizes for analyzing its security and design overhead. The following subsections present the results for the palmprint biometric-based hardware security technique.

9.1.1. Analyzing the impact of varying size of palmprint features set on final palmprint signature size

The size of the palmprint signature varies in accordance with the number and type of palmprint features chosen. Fig. 9.1 shows the variation in the final palmprint signature with respect to different sizes of palmprint features set of the same palm. As shown in the figure, a larger size palmprint signature can be obtained by choosing more palmprint features for securing larger DSP designs. The palmprint signature size can be scaled down by choosing

relatively lesser number of features, according to the size of target designs to

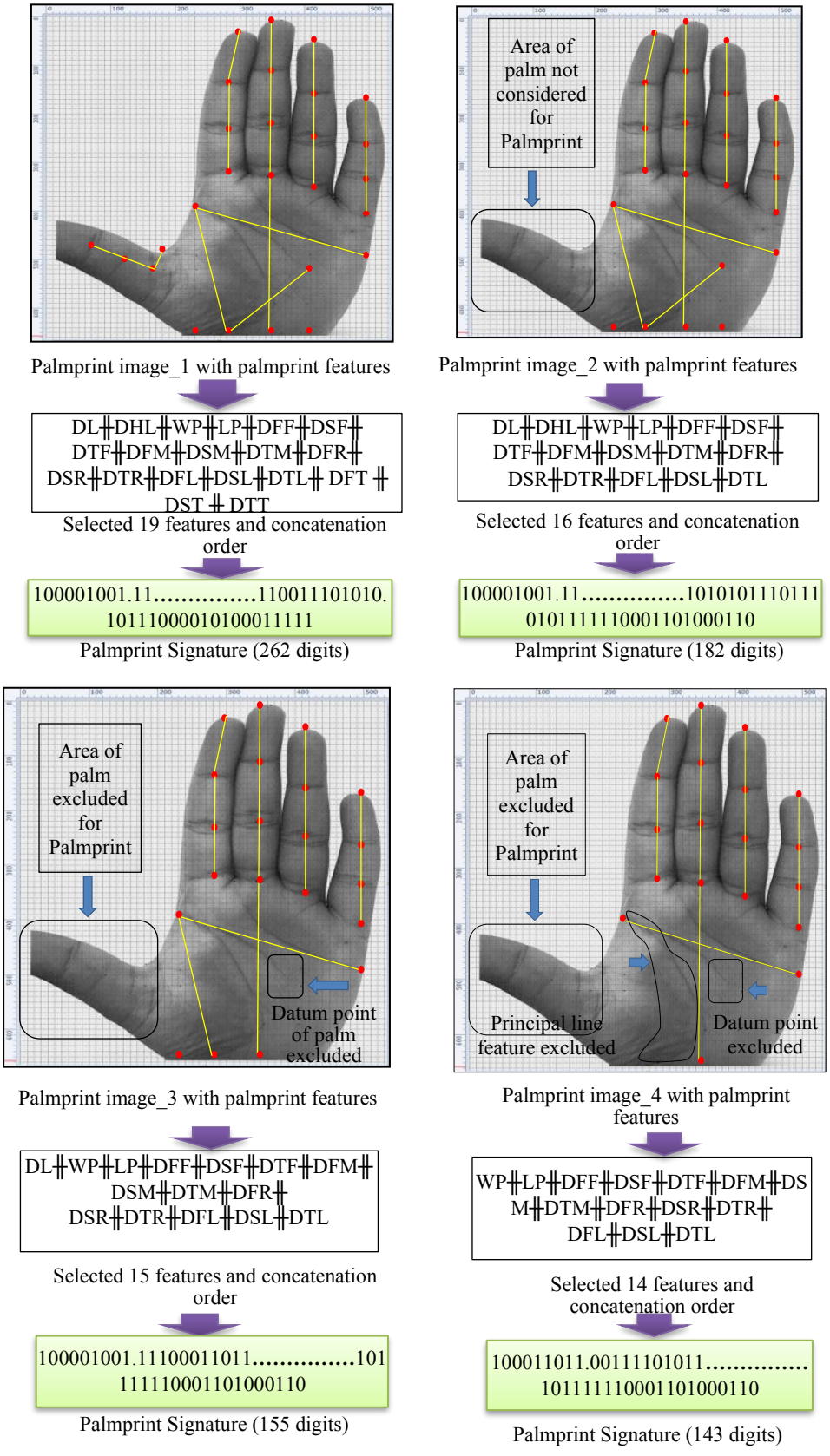


Fig. 9.1. Variation in the final palmprint signature with respect to different size of palmprint features set of the same palm

be secured.

9.1.2. Security analysis of proposed palmprint biometric based hardware security methodology

Strength of the proposed palmprint biometric based hardware security approach is analyzed in terms of probability of coincidence (Pc) and tamper tolerance (TT) metrics [31], [32], [36], [37]-[39].

Table 9.1 shows the variation in the Pc for FIR filter for varying number of palmprint features in a palmprint signature. It is observed that a very low value of Pc can be achieved by embedding a greater number of palmprint features. Further, Table 9.2 shows the Pc of different DSP designs for maximum possible number of constraints embedded and the Pc is compared with the biometric fingerprint biometric approach [40]. As shown, the proposed palmprint biometric approach achieves lower Pc than the fingerprint-based approach [40]. This is because the proposed palmprint signature comprises of three types of digits (‘0’, ‘1’ and ‘.’) in contrast to the two types of digits (‘0’ and ‘1’) in the fingerprint-based approach. Thus, the proposed approach is able to embed larger constraints (z) than the fingerprint-based approach, resulting into lower Pc and hence providing greater strength of palmprint signature. Further, the proposed palmprint approach is compared with IP steganography approach [37], IP digital signature approach [31] and IP digital signature-based watermarking approach [33] in Tables 9.3, 9.4 and 9.5, respectively. As shown in the tables, the proposed approach is capable to achieve lower Pc than hardware steganography approach [37], IP digital signature approach [31] and IP digital signature-based watermarking approach [33] because of higher yield in the number of generated security constraints. Since lower Pc is achieved for 8-point DCT, it is intuitive that smaller size DCT (e.g., 4-point) will also have lower Pc than steganography approach.

Further, the tamper tolerance ability of the proposed palmprint signature is reported in Tables 9.4, 9.5 and 9.6 and compared with IP digital signature based watermarking approach [33] and fingerprint-based approach [40] and the IP digital signature approach [31], respectively. As shown, the proposed approach has higher tamper tolerance ability (due to larger signature space)

than the IP digital signature approach [31], IP digital signature-based watermarking approach [33] and biometric fingerprint-based approach [40]. This is because of generating higher strength digits (W) in the palmprint signature than the IP digital signature approach [31], IP digital signature-based watermarking approach [33] and fingerprint-based approach [40]. Because of high tamper tolerance ability, an attacker cannot find the exact palmprint signature to attempt tampering in the form of regeneration of duplicate signature. This incapacitates an attacker from duplicating the authentic palmprint signature and embedding into fake designs for evading counterfeit detection process. Hence the proposed palmprint-based approach offers robust security against piracy/counterfeiting.

9.1.3. Design cost analysis of proposed palmprint biometric based

TABLE 9.1 Variation in Pc of FIR filter design for different size of palmprint signature of same palm

# Palmprint features	# constraints (z)	Pc
17	227	6.85E-14
16	182	2.79E-11
15	155	1.02E-9
14	143	5.09E-9
12	105	8.14E-7

Table 9.2 Comparison of Pc w.r.t related work [40]

Bench-marks	Proposed		Related work [40]	
	Maximum constraints	Pc	Maximum constraints	Pc
4-point DCT	27	4.23E-4	25	7.52E-4
4-point IDCT	27	4.23E-4	25	7.52E-4
8-point DCT	125	5.63E-8	121	9.61E-8
8-point IDCT	125	5.63E-8	121	9.61E-8
FIR	231	4.01E-14	225	8.95E-14

Table 9.3 Comparison of Pc w.r.t. related work [37]

Bench-marks	Proposed		Related work [37]	
	constraints	Pc	constraints	Pc
8-point DCT	125	5.63E-8	43	3.2E-3
8-point IDCT	125	5.63E-8	43	3.2E-3
FIR	231	4.01E-14	57	4.9E-4

Table 9.4 Comparison of proposed approach with digital signature [31]

Bench-marks	Pc		TT	
	Proposed	[31]	Proposed	[31]
4-point DCT	4.23E-4	1.00E-3	7.6E+12	1.6E+7
4-point IDCT	4.23E-4	1.00E-3	7.6E+12	1.6E+7
8-point DCT	5.63E-8	2.22E-4	4.3E+59	9.22E+18
8-point IDCT	5.63E-8	2.22E-4	4.3E+59	9.22E+18
FIR	4.01E-14	4.94E-4	1.6E+110	1.44E+17

hardware security methodology

The design cost is computed using a 15nm open-cell library [86] and has been reported in Table 9.7. As shown in the table, a very trivial overhead in the design cost (less than 0.9%) is incurred compared to the baseline counterparts (designs without embedded palmprint). The underlying reason is the incurrence of extra registers for satisfying the embedding of all palmprint biometric hardware security constraints. For example, the FIR filter requires 15 registers instead of 8 registers post embedding the signature as shown in the Table 9.7. However, the cost overhead is merely 0.8%. This is because the cost computation formula (given in 3.3) also includes the area of functional unit (FU) resources (adders, multipliers etc.) along with the area of registers. However, the area of FU resources remains unchanged and only the overall register area is increased post embedding the signature. Moreover, the design

Table 9.5 Comparison of proposed approach with digital signature based watermarking approach [33]

Bench-marks	Pc		TT	
	Proposed	[33]	Proposed	[33]
4-point DCT	4.23E-4	1.00E-3	7.6E+12	1.6E+7
4-point IDCT	4.23E-4	1.00E-3	7.6E+12	1.6E+7
8-point DCT	5.63E-8	5.63E-8	4.3E+59	4.3E+59
8-point IDCT	5.63E-8	5.63E-8	4.3E+59	4.3E+59
FIR	4.01E-14	6.46E-4	1.6E+110	3.6E+16

Table 9.6 Comparison of tamper tolerance (TT) w.r.t. related work [40]

Bench-marks	Proposed		Related work [40]	
	Signature size (S)	Tamper tolerance	Signature size (S)	Tamper tolerance
4-point DCT	27	7.6E+12	25	3.3E+7
4-point IDCT	27	7.6E+12	25	3.3E+7
8-point DCT	125	4.3E+59	121	2.6E+36
8-point IDCT	125	4.3E+59	121	2.6E+36
FIR	231	1.6E+110	225	5.4E+67

Table 9.7 Design cost pre and post embedding palmprint biometric constraints

Benchmarks	# of registers in baseline	# of registers in palmprint implanted design	Design cost of baseline	Design cost of palmprint implanted design	% cost overhead
4-point DCT	4	5	0.5611	0.5623	0.2%
4-point IDCT	4	5	0.5611	0.5623	0.2%
8-point DCT	8	11	0.4721	0.4740	0.4%
8-point IDCT	8	11	0.4721	0.4740	0.4%
FIR	8	15	0.4443	0.4479	0.8%

cost has the area weightage of only 0.5 while the other 0.5 weightage goes to latency which remains unchanged post embedding the signature. The proposed palmprint biometric based hardware security approach is therefore capable of embedding larger number of security constraints (robust security) while incurring trivial design cost overhead.

9.2. Results and analysis: Double line of defense approach for securing DSP IP cores using structural obfuscation and chromosomal DNA impression

This section analyses results of the proposed structural obfuscation and chromosomal DNA impression-based technique for securing the IP cores corresponding to the DSP applications. A 15 nm open cell library [86] was used to calculate the design cost. The experimental results have been analyzed for various DSP benchmarks. Our technique is automated using C++ language and run-on intel(R) core (TM) i5-11235G7 processor with 2.40GHz. The implementation run time of this methodology is ~2.041s.

9.2.1. Security analysis

The security of the proposed double line of defense methodology using structural obfuscation and chromosomal DNA impression is analyzed in terms of strength of obfuscation, probability of coincidence and tamper tolerance ability and design cost along with its implementation run time.

9.2.1.1. Security analysis in terms of strength of obfuscation

The obfuscation achieved is measured by the strength of obfuscation in terms of the number of gates modified in the datapath of the DSP design, as shown in Fig. 9.2. The more the number of gates affected, the more is the strength of obfuscation and the harder it is for an adversary to alter the RTL description of the DSP core. Fig. 9.2 shows the strength of obfuscation achieved using the proposed method for different DSP applications.

9.2.1.2. Security analysis in terms of probability of coincidence

Security against IP piracy is analyzed in terms of the strength of ownership proof using the probability of coincidence metric. The ‘Pc’ value specifies the

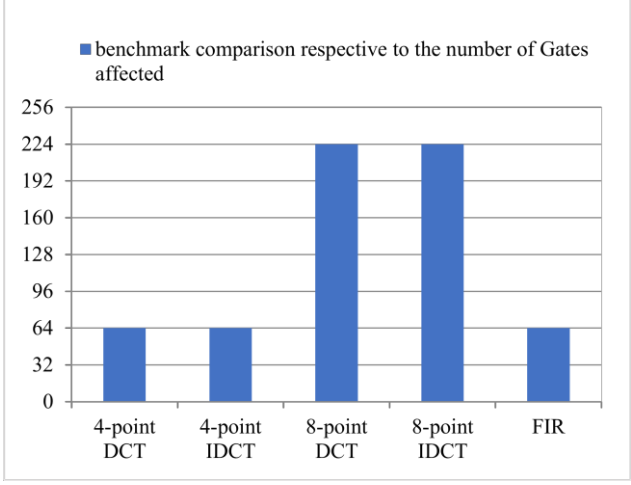


Fig. 9.2 Strength of obfuscation of proposed approach

probability of coincidently detecting security constraints in an unsecured design; hence it is desirable for it to be low as much as possible. The pc value achieved using our method for FIR, 4-point DFT, 4-point DCT design and 8-point DCT are reported in Table 9.8, for varying effective constraints size with respect to encrypted chromosomal DNA impression (corresponding to different number of base Pairs (AT/GC) in chromosomal DNA and different number of polynucleotide). As shown in the Table 9.8, a low 'Pc' is achieved for all the variations of encrypted chromosomal DNA impression sizes implanted into the obfuscated DSP designs.

The proposed encrypted digital DNA impression methodology is also compared with a recent state-of-the-art security work based on facial biometric [41] and hardware steganography [37]. The comparisons of 'Pc' of presented work with [41] and [37] are reported in Fig. 9.3. As evident, our methodology achieves much lower 'Pc' compared to both [41] and [37]. This is because the number of security constraints generated using [41] and [37] are significantly lesser compared to the proposed methodology.

Table 9.8 The Pc of the proposed approach indicating strength of digital evidence

# Base Pairs (AT/GC) in chromosomal DNA	#Polynucleostride (leading/lagging strand in DNA)	Digital DNA impression size	FIR	DFT	4point DCT	8-point DCT
			Pc			
			#E_C (Effective constraints)			
2	4	32	1.39E-2	9.3E-2	1.3E-2	1.2E-1
			32	32	32	32
4	9	64	1.4E-3	8.7E-3	1.4E-3	1.6E-2
			49	64	49	64
6	17	128	1.4E-3	7.59E-5	1.4E-3	2.5E-4
			49	128	49	128

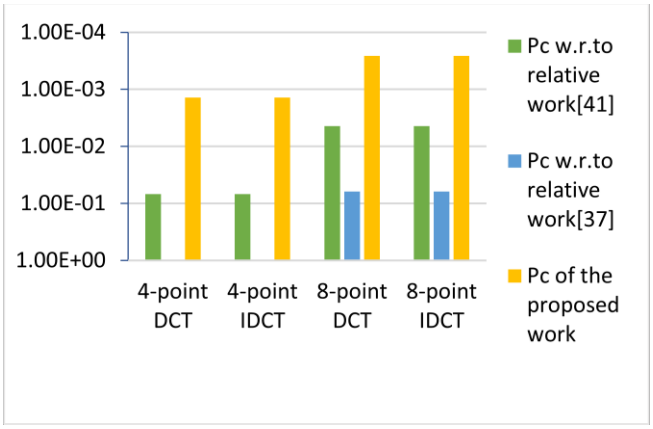


Fig. 9.3 Comparison of probability of coincidence (Pc)

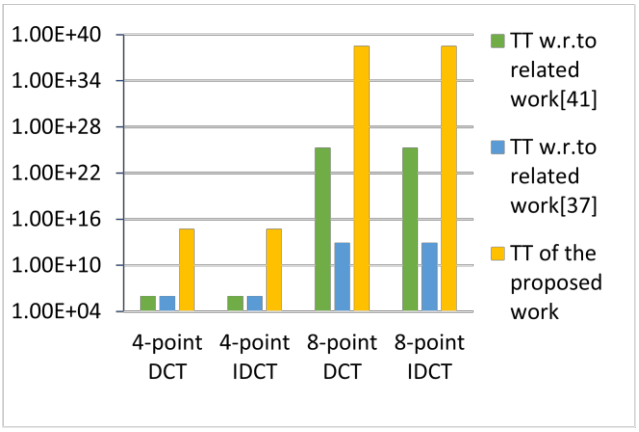


Fig. 9.4 Comparison of tamper tolerance ability (TA)

9.2.1.3. Security analysis in terms of tamper tolerance

Security against tampering vulnerability is evaluated using the tamper tolerance ability. The larger key-space proportionately increases the resistance for an attacker to find the exact encrypted digital DNA impression implanted in the design. Since the security constraints generated and embedded through our work is comparatively higher, thus the tamper tolerance ability of proposed methodology is far stronger than [41] and [37]. The comparisons of tamper tolerance ability of our work with [41] and [37] are shown in Fig. 9.4. As evident, the TA of the method that we presented is far robust than [41] and [37] due to generation of more security constraints.

9.2.2. Design cost analysis and implementation run time

A 15nm open-cell library [86] is used to calculate both the delay and area of a hardware design. Table 9.9 reports the design cost of proposed obfuscated encrypted digital DNA impression implanted design and pre-embedded

Table 9.9 Obfuscated design cost pre and post embedding encrypted chromosomal DNA impression constraints (32, 64, 128 bits)

Benchmarks [84]	# of registers in obfuscated design (pre embedding digital DNA impression)	# of registers in proposed obfuscated encrypted digital DNA impression implanted design	Design cost of baseline	Design cost of proposed obfuscated encrypted digital DNA impression implanted design	% Cost overhead in proposed obfuscated encrypted digital DNA impression implanted design
4-point DCT	8	8	0.5659	0.5659	0.00%
4-point IDCT	8	8	0.5659	0.5659	0.00%
8-point DCT	16	16	0.4771	0.4771	0.00%
8-point IDCT	16	16	0.4771	0.4771	0.00%

Table 9.10 Execution time of proposed DNA based approach

Benchmarks [84]	Execution time
4-point DCT	2.323sec
4-point IDCT	2.323sec
8-point DCT	2.491sec
8-point IDCT	2.491sec
FIR filter	2.904sec

(baseline) obfuscated version. As evident, our methodology incurs design cost overhead of 0.00 % corresponding to all DSP designs.

The implementation run time of the proposed security approach for different DSP benchmarks have been shown in Table 9.10. As evident from the table, the proposed technique is capable of embedding robust encrypted DNA impression into the DSP designs at very less implementation complexity (in terms of embedding time).

9.3. Results and analysis: Designing secured reusable convolutional IP core in CNN against piracy using facial biometric based hardware security

The proposed approach allows three curve detection kernels/filters to convolve in parallel over input image and generating feature maps corresponding to each kernel as output of convolutional layer. Further, each kernel is unrolled twice and is capable of computing two pixels in parallel. For the sake of brevity, the details of all kernel datapath could not be included. Further, CNN kernel IP core is secured with facial biometric which offers robust security

against IP piracy/counterfeiting, false claim of IP ownership proof and IP forgery attacks.

9.3.1. Analyzing the proposed reusable convolutional IP core in terms of pixel computation

Table 9.11 shows the comparison of the number of executions for the convolutional operation between a conventional hardware design and proposed reusable IP core. As evident from the table the proposed reusable IP core design offers significantly lesser number of executions of the convolution operation due to heavy parallelism involved owing to loop unrolling of the datapath. For example, for different sizes of the input image, the proposed reusable IP core produces much lower number of executions. Furthermore, Table 9.12 highlights the number of pixels computed in parallel by the proposed reusable IP core for different kernel sizes (K=3, K=4, K=5). As evident from Table 9.12, the number of pixels computed through proposed approach for respective weight loading is twice as compared to the pixels computed through conventional hardware design. This reflects the proposed approach is more efficient in terms of better performance.

9.3.2. Analyzing the impact of implanting facial biometric signature on functional units in RTL datapath of CNN convolutional layer kernels

The size of facial signature varies in accordance to the number of facial

Table 9.11 Number of executions for convolution operation

For three Kernels(K=3)	# Executions of convolution operation in conventional hardware design [73]	# Executions of convolution operation in proposed reusable IP core
For image size 128×128	16384	4096
For image size 256×256	65536	16384
For image size 512×512	262144	65536

Table 9.12 Number of pixels computed in parallel for different kernel sizes

# Kernels	Weight loading	Pixels computed through conventional hardware design [73]	Pixels computed in parallel through proposed approach
K=3	27	3	6
K=4	36	4	8
K=5	45	5	10

features chosen by IP designer/vendor. Further, different signature can be formed depending on the concatenation ordering of chosen features. The facial signature size can be scaled down by choosing relatively lesser number of features, according to the size of target IP core to be secured. The proposed approach renders zero overhead as the number of registers pre and post embedding facial signature are same. Furthermore, the impact of embedding facial signature on functional units and corresponding multiplexers and demultiplexers is shown in Table 9.13.

9.3.3. Security analysis

Strength of the secured CNN convolutional layer IP core using facial biometric approach is analyzed in terms of probability of coincidence (Pc) and tamper tolerance (TT) metrics [31], [32], [36], [37]-[39].

9.3.3.1. Security analysis in terms of probability of coincidence

Table 9.13 Resources in the RTL datapath of CNN convolutional layer reusable IP core (pre and post embedding facial biometric constraints)

	Resources pre-embedding security constraints				Resources post-embedding security constraints			
Kernel number	FUs	# Registers (for double unrolling)	Muxes	Demuxes	FUs	# Registers (for double unrolling)	Muxes	Demuxes
Convolutional layer datapath (Kernel 1st	2M, 2A	36	#8X1 Muxes =4	#1x8 Demuxes =2	2M, 2A	36	#8X1 Muxes =8	#1x8 Demuxes =6
			#16X1 Muxes =6	#1x8 Demuxes =4			#16X1 Muxes =4	#1x8 Demuxes =2
			#2X1 Muxes =16	#1x8 Demuxes =16			#2X1 Muxes =16	#1x8 Demuxes =16
Convolutional layer datapath (Kernel 2nd	2M, 2A	36	#8X1 Muxes =4	#1x8 Demuxes =2	2M, 2A	36	#8X1 Muxes =7	#1x8 Demuxes =5
			#16X1 Muxes =6	#1x8 Demuxes =4			#16X1 Muxes =4	#1x8 Demuxes =2
			#2X1 Muxes =16	#1x8 Demuxes =16			#2X1 Muxes =16	#1x8 Demuxes =16
							#4X1 Muxes =1	
Convolutional layer datapath (Kernel 3rd	2M, 2A	36	#8X1 Muxes=4	#1x8 Demuxes =2	2M, 2A	36	#8X1 Muxes =6	#1x8 Demuxes =4
			#16X1 Muxes =6	#1x8 Demuxes =4			#16X1 Muxes =5	#1x8 Demuxes =3
			#2X1 Muxes =16	#1x8 Demuxes =16			#2X1 Muxes =17	#1x8 Demuxes =17

Pc obtained for different facial images using proposed approach is presented in Table 9.14. Where, number of security constraints are different corresponding to the facial image chosen by IP designer/vendor for generating facial signature. Pc metric shows the probability of coincident detection of covert security constraints with an unsecured design, hence low Pc is desirable. Pc value of securing CNN kernel using facial biometric is lesser than the related approaches such as digital signature [39] and steganography [37], as shown in Table 9.14. Further, percentage reduction in Pc value achieved using the proposed approach corresponding to the related approaches is shown in Table 9.15. Proposed approach renders significant reduction in Pc value therefore is capable of offering more security strength than the related approaches.

9.3.3.2. Security analysis in terms of tamper tolerance

The tamper tolerance ability of a design indicates the security in terms of rendering it difficult for an adversary to regenerate the exact signature. The tamper tolerance ability is measured in terms of total signature space. As

Table 9.14 Comparison of Pc with respect to related approach [39], [37] for CNN convolutional layer IP core

Facial images	#Security constraints	Pc of the proposed approach	Digital signature strength [39]	Pc of the related approach [39]	# Stego-constraints [37]	Pc of the related approach [37]
Image_1	81	.4707	15	.8697	13	.8860
Image_2	84	.4577	30	.7564	24	.7999
Image_3	84	.4577	60	.5722	43	.6703
Image_4	84	.4577	75	.4977	57	.5884
Image_5	83	.4620	82	.4663	59	.5776

Table 9.15 Percentage reduction in Pc value achieved using proposed approach compared to related works [39], [37]

Facial images	Reduction in Pc wrt [39]	Reduction in Pc wrt [37]
Image_1	45.87%	46.87%
Image_2	39.48%	42.78%
Image_3	20.01%	31.71%
Image_4	8.03%	22.21%
Image_5	0.92%	20.01%

Table 9.16 Comparison of tamper tolerance with respect to related approach [39] for CNN convolutional layer Reusable IP core

Facial images	#Security constraints	Tamper tolerance of the proposed approach	Digital signature strength [39]	Tamper tolerance of the related approach [39]
Image_1	81	2.417E+24	15	3.2E+4
Image_2	84	1.934E+25	30	1.07E+9
Image_3	84	1.934E+25	60	1.15E+18
Image_4	84	1.934E+25	75	3.7E+22
Image_5	83	9.67E+24	82	4.83E+24

shown in Table 9.16, the proposed approach has higher tamper tolerance ability (due to higher signature strength) than the digital signature approach [39]. In case of the facial biometric signature, total signature space is of size 2^{84} which is a huge number and can be further scaled depending on the number of facial features chosen by the IP designer/vendor. Therefore, proposed approach is capable of securing CNN convolutional layer IP core from IP piracy/ counterfeiting or IP forgery attempt by a potential adversary.

9.3.4. Design area analysis

The design area is computed using a 15nm open-cell library [86]. Further, impact of number of CNN convolutional layer kernels 'K' and their unrolling factor on design area is reported in Fig. 9.5. The more the number of kernels/filters, more is the parallel computation of pixels. Further, the value of unrolling also accelerates the pixel computation process. The proposed approach allows three CNN kernels with twice unrolling. Thus, improving the CNN performance and limiting the design area on the other hand. Further, as shown in Table 9.13, zero overhead in the design is incurred compared to the baseline counterparts (designs without embedded facial signature). The underlying reason is incurrence of no extra registers for satisfying the embedding of all facial security constraints. Therefore, the overall design area overhead is zero/trivial. Further, the proposed approach can be scaled for more number of convolutional filter kernels thereby computing more number of output pixels in one execution. Moreover, scaling can also be achieved by

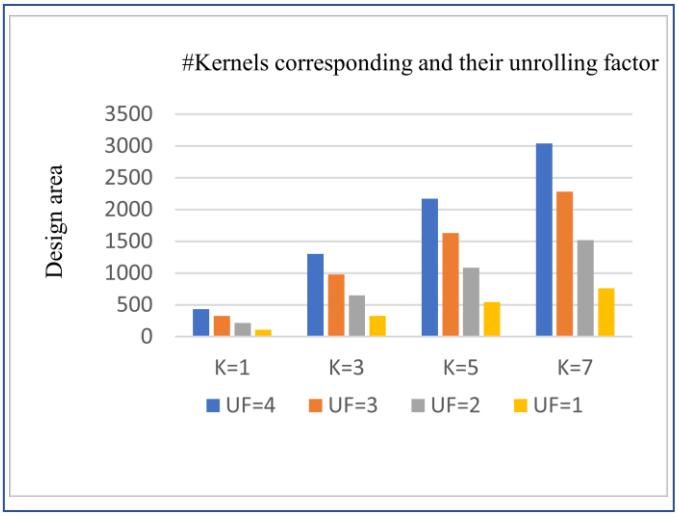


Fig. 9.5. Impact of number of CNN convolutional filter kernels 'K' and unrolling factor 'UF' on design area

increasing the number of unrolling.

9.4. Results and analysis: Retinal biometric for designing secured JPEG-codec hardware IP core for CE systems using HLS

This section analyses results of the proposed retinal biometric based hardware security approach.

9.4.1. Security analysis

Robustness of the security strength offered by the proposed retinal biometric is analyzed using probability of coincidence and tamper tolerance metrics [31], [32], [36], [37]-[39].

9.4.1.1. Security analysis in terms of probability of coincidence

The Pc value indicates probability of coincidently detecting the authentic retinal security constraints within an unsecured JPEG-codec design. Therefore, lower Pc value is desirable and it indicates higher security strength. Furthermore, lower Pc value enables the robust security in terms of strength of digital evidence (proof). The Pc value for different retinal signature size corresponding to variable number of retinal features for image_1 is shown in Table 9.17. Further, the respective Pc corresponding to five different retinal images is reported in Table 9.18. It is evident from Table 9.17 and Table 9.18 that the retinal signature of larger size (capacitates the IP vendor to generate large number of security constraints) results into lesser Pc value and vice versa.

Table 9.17 Variation in Pc for different size of retinal signature of same retina (Image_1)

# Retinal features	# Constraints (z)	Pc
33	900	4.0E-6
25	700	6.4E-5
18	500	1.0E-3
11	300	1.5E-2
4	100	2.5E-1

Table 9.18 Variation in Pc and TT for different retinal images

# Retinal images [77]	# Constraints (z)	Pc	TT
Image_1	922	2.9E-6	~1.0E+435
Image_2	589	2.9E-4	1.05E+281
Image_3	953	1.9E-6	~1.0E+449
Image_4	958	1.8E-6	~1.0E+451
Image_5	1141	1.4E-7	~1.0E+538

9.4.1.2. Security analysis in terms of tamper tolerance

Tamper tolerance is the indicative of robustness of the security strength of the design against tampering. Higher tampering tolerance indicates that adversary cannot regenerate the exact retinal signature by performing tampering. It hinders an adversary to pirate the designs by implanting the regenerated signature into pirated designs. The tamper tolerance provided by the retinal biometric corresponding to five different retinal images is shown in Table 9.18. It is evident from Table 9.18 that more tamper tolerance is achieved by embedding the retinal signature corresponding to image_5 because of more security constraints than other retinal images. The proposed retinal biometric hardware security approach is also compared with recent state-of-the art hardware security approaches such as digital signature [39], fingerprint biometric [40] and facial biometric [41] based hardware security approach. The Pc comparison with respect to fingerprint biometric, facial biometric and digital signature-based approach is shown in Table 9.19. As evident form the Pc comparison, the proposed retinal biometric approach achieves lesser Pc value than related approaches [40], [41], [39]. Therefore, lesser Pc value of the proposed approach ensures the stronger proof of digital evidence to the genuine IP design only. Further, the tamper tolerance of the proposed retinal biometric is compared to [40], [41], [39]. The proposed retinal biometric approach also attains higher tamper tolerance than related approaches. Table 9.20 shows the tamper tolerance of the proposed approach for varying retinal

Table 9.19 Comparison of Pc w.r.t related work [40], [41], [39]

Proposed		Fingerprint biometric [40]		Facial biometric [41]		Digital signature [39]	
#Security constraints (z)	**Pc**	**#Security constraints (z)**	**Pc**	**#Security constraints (z)**	**Pc**	**#Security constraints (z)**	**Pc**
922	2.9E-6	526	7.06E-4	75	3.5E-1	15	8.1E-1
589	2.9E-4	350	8.0E-3	80	3.3E-1	30	6.6E-1
953	1.9E-6	538	5.9E-4	81	3.27E-1	60	4.3E-1
958	1.8E-6	555	4.7E-4	83	3.18E-1	120	1.9E-1
1141	1.4E-7	418	3.13E-3	84	3.13E-1	240	3.6E-2

Table 9.20 Comparison of TT w.r.t related works [40], [41], [39]

Proposed		Fingerprint biometric [40]		Facial biometric [41]		Digital signature [39]	
#Security constraints	**TT**	**#Security constraints**	**TT**	**#Security constraints**	**TT**	**#Security constraints**	**TT**
922	~1.0E+435	526	9.24E+250	75	6.08E+35	15	1.43E+7
589	1.05E+281	350	9.8E+166	80	1.47E+38	30	2.05E+14
953	~1.0E+449	538	4.91E+256	81	4.43E+38	60	4.23E+28
958	~1.0E+451	555	6.34E+264	83	3.99E+39	120	1.79E+57
1141	~1.0E+538	418	2.73E+199	84	1.19E+40	240	3.22E+114

biometric signature strength; thus, making it highly improbable for an adversary to exactly regenerate the original retinal signature for evading piracy detection process.

9.4.2. Design cost analysis of proposed retinal biometric based hardware security methodology

The impact of enabling the robust security of JPEG-codec hardware IP core through proposed retinal biometric approach, on design cost, is analyzed using 15-nm NanGate library [86]. The design cost of JPEG-codec hardware IP core design, pre-embedding and post-embedding the retinal signature for different signature strength, corresponding to retinal image (Image_1), is shown in Table 9.21. As evident, no design overhead is reported for varying sizes of retinal signature strength. This is because no extra register was required during embedding all the generated retinal biometric hardware security constraints into JPEG-codec design while satisfying distinct register allocation policy. Further, the design cost corresponding to different retinal images (Image_1 to Image_5) is shown in Table 9.22. As evident, no design overhead is reported post-embedding security constraints for different retinal biometric images. Therefore, as evident, the proposed retinal biometric hardware security approach offers more robust security against IP piracy than related hardware security approaches, at zero design cost overhead.

Furthermore, Fig. 9.6 reports the variation in "Pc-design cost" tradeoff for

Table 9.21 JPEG-codec IP core design cost pre and post embedding retinal biometric constraints (Image_1)

Retinal signature size (image_1)	# of registers in baseline	# of registers in retinal signature implanted design	Design cost of baseline	Design cost of retinal signature implanted design	% Cost overhead
100bits	73	73	0.214	0.214	0.0%
300bits	73	73	0.214	0.214	0.0%
500bits	73	73	0.214	0.214	0.0%
700bits	73	73	0.214	0.214	0.0%
900bits	73	73	0.214	0.214	0.0%

Table 9.22 JPEG-codec IP core design cost pre and post embedding retinal biometric constraints for different retinal images

Retinal images	# of registers in baseline	# of registers in retinal signature implanted design	Design cost of baseline	Design cost of retinal signature implanted design	% Cost overhead
Image_1	73	73	0.214	0.214	0.0%
Image_2	73	73	0.214	0.214	0.0%
Image_3	73	73	0.214	0.214	0.0%
Image_4	73	73	0.214	0.214	0.0%
Image_5	73	73	0.214	0.214	0.0%

different number of retinal features corresponding to the same retinal image. As evident, the "Pc" value reduces with the increase in number of retinal features chosen for generating the retinal signature. This is because, increase in number of retinal features results into increased number of corresponding hardware security constraints to be embedded into the design. Hence, to achieve higher strength of digital evidence (i.e., lower "Pc"), large number of features should be used in a retinal signature.

Since the proposed approach utilizes convolution process to locate the feature points of IP vendor retinal image using branching and bifurcation kernel matrices during signature generation process, therefore the time complexity can be indicated as: O(pqmn), where p×q is the size of cropped retinal image matrix and m×n is the size of kernel matrix respectively. Further, the implementation run time of the proposed security approach has been shown in Table 9.23. As evident from the table, the proposed technique is capable of detecting and embedding robust retinal impression into the JPEG-codec design at very less implementation complexity. Since the proposed approach required 2D array for storing the kernel matrix and the retinal image matrix for locating feature points during signature generation process therefore the space complexity is given as: O(pq+mn), where p×q and m×n are the sizes of the 2-

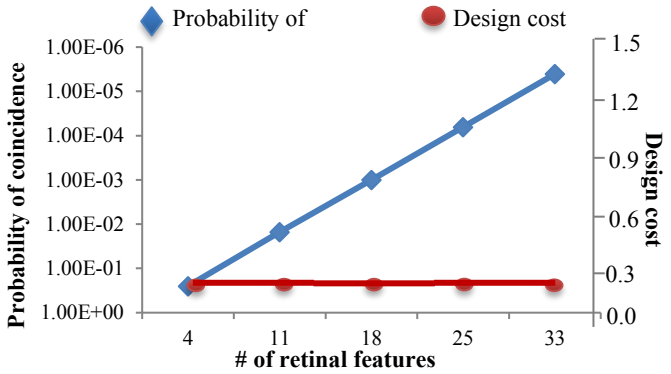


Fig. 9.6. Pc- design cost trade-off for JPEG-codec for different number of retinal features of same retinal image (Image_1)

Table 9.23 Implementation time of the proposed retinal biometric based hardware security approach

# Retinal images	# Implementation time (in msec.)
Image_1	273.677
Image_2	205.339
Image_3	272.026
Image_4	275.590
Image_5	332.695

D arrays of retinal image matrix and kernel matrix respectively.

9.5. Results and analysis: Exploration of security-cost tradeoff for signature driven security algorithms for optimal architecture of data-intensive hardware IPs

This section analyzes the proposed approach in terms of security-design cost tradeoff for the signature-based security methodologies using PSO for DSP hardware IPs. It enables the IP designer and CE integrator to choose an optimal DSP hardware solution. Furthermore, it also guides the IP designer to achieve maximum security strength and minimal design cost overhead in parallel.

9.5.1. Security analysis

9.5.1.1. Security analysis in terms of probability of coincidence

The security is analyzed in terms of strength of ownership proof (probability of coincidence) [31], [32], [36], [37]-[39]. The 'Pc' metric specifies the probability of coincidently detecting security constraints in a design; hence it is desirable for it to be low as much as possible. The Pc value achieved for the respective security algorithms (watermarking based, encrypted hash based and facial biometric based) using PSO-DSE approach for 8-point DCT and ARF are reported in Fig. 9.7 and Fig. 9.8 respectively. Similarly, Pc metric can be obtained for 4-point DCT, FIR and DWT applications. It can be observed that the pc for the facial biometric based security algorithm is lesser than the pc for watermarking and encrypted hash-based security in both 8-point DCT and ARF applications. This is because lesser Pc is encountered if more the number of constraints can be embedded using that security algorithm. Facial biometric approach [41] results into more security constraints as it generates the signature based on unique and non-replicable facial features as well as uses several features in the features set to generate large size security constraints. The number of embedded constraints has been generated based on the signature strength.

9.5.1.2. Security analysis in terms of tamper tolerance

Security against tampering vulnerability is evaluated using the tamper tolerance ability. The larger signature size proportionately increases the resistance for an attacker to find the exact security signature impression implanted in the design. Since the number of encoding variables in watermarking approach is comparatively higher (four which are more than the two for both encrypted hash [39] and facial biometric [41]), thus the TT ability of watermarking approach is stronger than [41] and [39]. The comparisons of TT ability of the watermarking [32], encrypted hash [39] and facial biometric approach [41] based on different signature sizes is shown in Fig. 9.9. As evident, the TT of the watermarking approach is far robust than [41] and [39].

9.5.2. Analysis of impact of signature strength on fitness value and register count on DSP application

The impact of signature strength on fitness value and register count based on different security algorithms [32], [39] ,[41] for varying signature strength is analyzed using the security-design cost tradeoff function (shown in equation 7.7). The corresponding results for 8-point DCT and ARF are shown in Fig. 9.10 and Fig. 9.11 respectively. The bigger signature size results into more security constraints than the smaller signature size; hence more possibility of design overhead (in form of register count on embedding all the effective security constraints). The security metric (S_m^1) as shown in equation (7.3) also affects the fitness function value.

9.5.3. Analysis of security algorithms in terms of hardware cost, embedded security constraints and exploration time

The details of the security constraints, fitness function, design area, delay, global best solution and average exploration time of the proposed PSO-DSE for the signature-based security algorithms for 8-point DCT and ARF are shown in Table 9.24 and Table 9.25 respectively. The global best resource configuration (hardware solution) reported by the proposed approach for 8-point DCT and ARF are (1A, 4M) and (2A, 4M) respectively. The PSO-DSE [78] process during security-design cost tradeoff always converges to the global best solution. Further, the details of hardware units obtained during trade-off exploration (security–design cost) are reported in Table 9.26.

9.6. Results and analysis: Symmetrical Protection of Ownership Right's for IP Buyer and IP seller using Facial Biometric Pairing

This section analyzes the proposed symmetrical security methodology for ensuring the protection of ownership Right's for IP Buyer and IP seller using

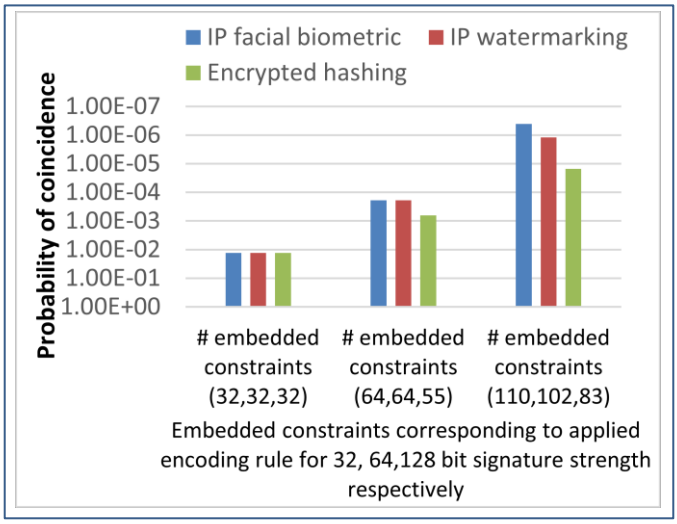


Fig. 9.7 Pc comparison of security methodologies for 8-point DCT application

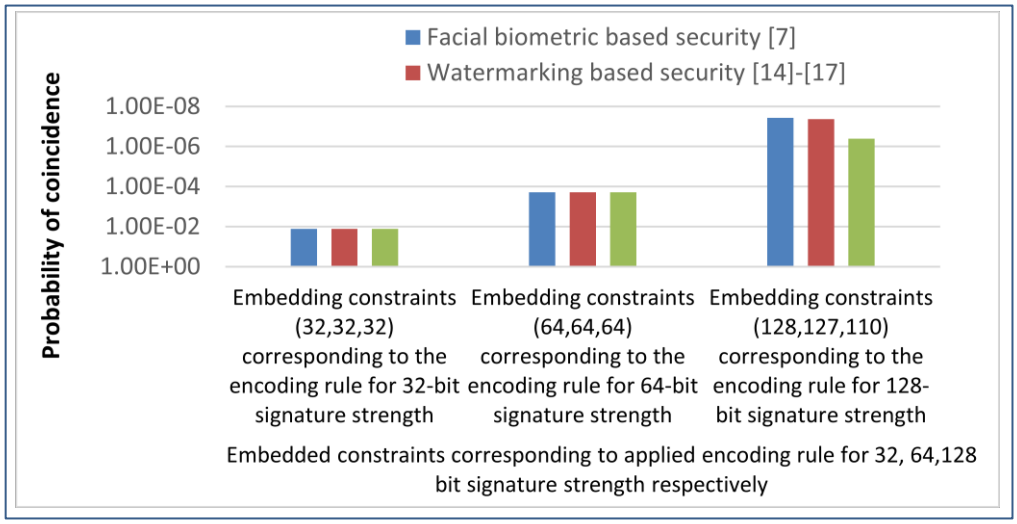


Fig. 9.8 Pc comparison of security algorithms for ARF framework

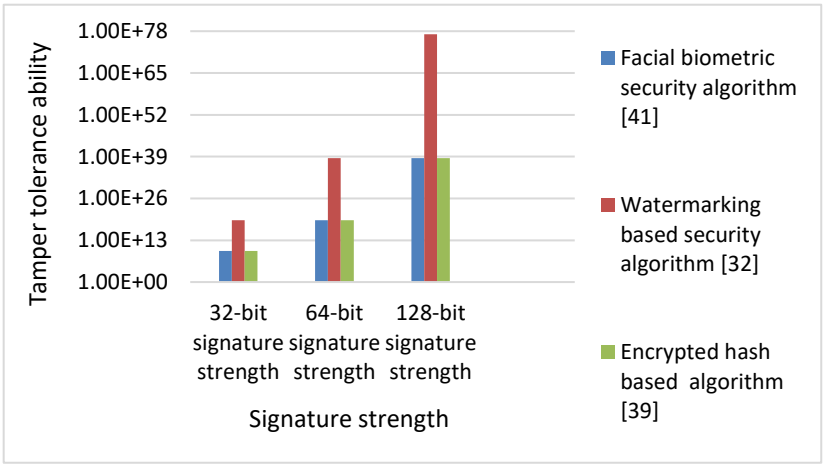


Fig. 9.9. Tamper Tolerance ability comparison of security algorithms for different signature strengths

facial biometric pairing.

9.6.1. Security analysis

The robustness of the presented security methodology is analyzed in terms of probability of coincidence (P_c). The lesser Pc value indicates the higher distinctiveness of security constraints as compared to baseline design. Therefore, lesser Pc value is the desirable. The *Pc* analyses of the proposed security methodology using the facial biometric signature corresponding to IP buyer and IP seller is depicted in Table 9.27 and Table 9.28 respectively. As

Table 9.24 Details of the security constraints, fitness function, global best solution and average exploration time of the proposed approach for 8-point DCT w.r.t. various security algorithms [84]

Application framework	Signature size (In bits)	Embedded constraints corresponding to the encoding rule of the algorithm	Security algorithm	Fitness value (Security-Design cost)	Design area 'Ad' (in um2)	Design latency 'Ld' (in ms.)	S_{Gb}	Exploration time (Avg. in µs.)
8-DCT	32	32	Facial biometric	0.36	327.15	927.39	[1, 4]	173.7
	64	64	Facial biometric	0.50	327.94	927.39	[1, 4]	
	128	110	Facial biometric	0.69	329.51	927.39	[1, 4]	
8-DCT	32	32	Watermarking	0.32	327.15	927.39	[1, 4]	164.4
	64	64	Watermarking	0.42	328.72	927.39	[1, 4]	
	128	102	Watermarking	0.53	328.72	927.39	[1, 4]	
8-DCT	32	32	Encrypted hash	0.40	327.94	927.39	[1, 4]	150
	64	55	Encrypted hash	0.52	327.94	927.39	[1, 4]	
	128	83	Encrypted hash	0.67	327.94	927.39	[1, 4]	

Table 9.25 Details of the security constraints, fitness function, global best solution and average exploration time of the proposed approach for ARF framework w.r.t. various security algorithms [84]

Application framework	Signature size (in bits)	Embedded constraints corresponding to the encoding rule of the algorithm	Security algorithm	Fitness value (Security-Design cost)	Design area 'Ad' (in um2)	Design latency 'Ld' (in ms.)	S_{Gb}	Exploration time (Avg. in µs.)
ARF	32	32	Facial biometric	0.2512	346.03	1391.09	[2, 4]	168.8
	64	64	Facial biometric	0.2980	346.03	1391.09	[2, 4]	
	128	128	Facial biometric	0.3916	346.03	1391.09	[2, 4]	
ARF	32	32	Watermarking	0.2466	346.03	1391.09	[2, 4]	157.2
	64	64	Watermarking	0.2890	346.81	1391.09	[2, 4]	
	128	127	Watermarking	0.3721	347.60	1391.09	[2, 4]	
ARF	32	32	Encrypted hash	0.2814	346.03	1391.09	[2, 4]	153.4
	64	64	Encrypted hash	0.3583	346.03	1391.09	[2, 4]	
	128	110	Encrypted hash	0.4697	348.38	1391.09	[2, 4]	

evident from Table 9.27 and 9.28 the proposed approach results into lesser Pc value.

9.6.2. Design cost analysis

The design cost, post embedding the facial biometric signature of IP user and IP supplier is shown in Table 9.29. As evident from Table 9.29, the proposed approach incurs zero design overhead while embedding the security constraints corresponding to IP buyer and IP seller. Thus, the proposed

Table 9.26 The details of DSP hardware units obtained during trade-off exploration (security–design cost)

Application framework [84]	Security algorithm	Post embedding register count based on signature size(bits)			#adder unit(s)	#multiplier unit(s)	#Mux units	#Demux units
		32	64	128				
4-point DCT	Facial biometric	5	6	6	1	2	6	3
	Watermarking	7	8	9	1	2	6	3
	Encrypted hash	6	6	6	1	2	6	3
8-point DCT	Facial biometric	8	9	11	1	4	10	5
	Watermarking	8	10	10	1	4	10	5
	Encrypted hash	9	9	9	1	4	10	5
FIR	Facial biometric	8	8	10	4	4	16	8
	Watermarking	9	10	11	4	4	16	8
	Encrypted hash	8	9	10	4	4	16	8
DWT	Facial biometric	5	7	11	1	1	4	2
	Watermarking	6	8	11	1	1	4	2
	Encrypted hash	7	8	11	1	1	4	2
ARF	Facial biometric	8	8	8	2	4	12	6
	Watermarking	8	9	10	2	4	12	6
	Encrypted hash	8	8	11	2	4	12	6

Table 9.27 PC analysis corresponding to facial signature of IP buyer w.r.t. [80]

Bench-marks	Proposed		Related work [80]	
	Max. security constraints (*h*)	*Pc*	Max. security constraints	*Pc*
DCT-8point	84	4.4E-3	30	1.4E-1
FIR	84	4.4E-3	30	1.4E-1
JPEG-codec	84	5.2E-1	30	7.9E-1
ARF	84	4.4E-3	30	1.4E-1
IIR	84	1.9E-3	30	1.0E-1

Table 9.28 PC analysis corresponding to facial signature of IP seller w.r.t. [80]

Bench-marks	Proposed		Related work [80]	
	Max. security constraints (*h*)	*Pc*	Max. security constraints	*Pc*
DCT-8point	84	4.4E-3	30	1.4E-1
FIR	84	4.4E-3	30	1.4E-1
JPEG-codec	84	5.2E-1	30	7.9E-1
ARF	84	4.4E-3	30	1.4E-1
IIR	84	1.9E-3	30	1.0E-1

security methodology ensures the protection of the rights of both the parties, IP buyer and vendor with zero design overhead.

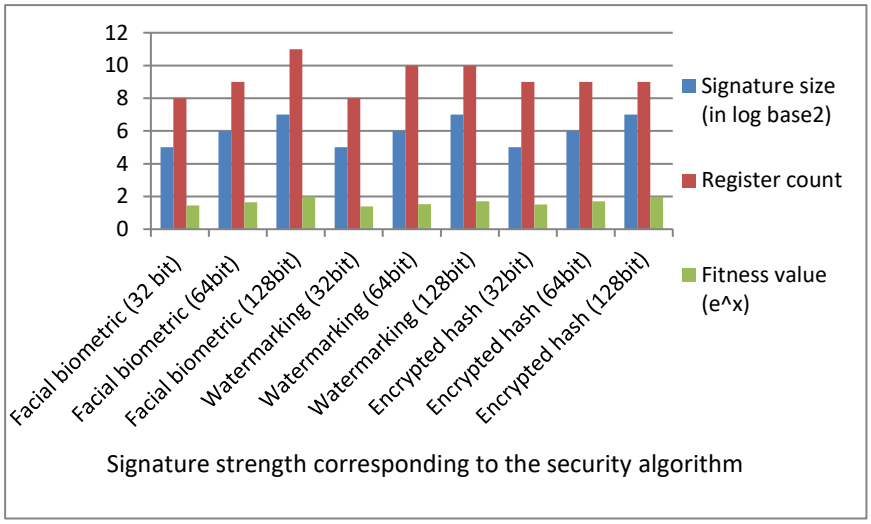


Figure 9.10 Impact of signature strength on fitness value and register count in 8-point DCT application [84]

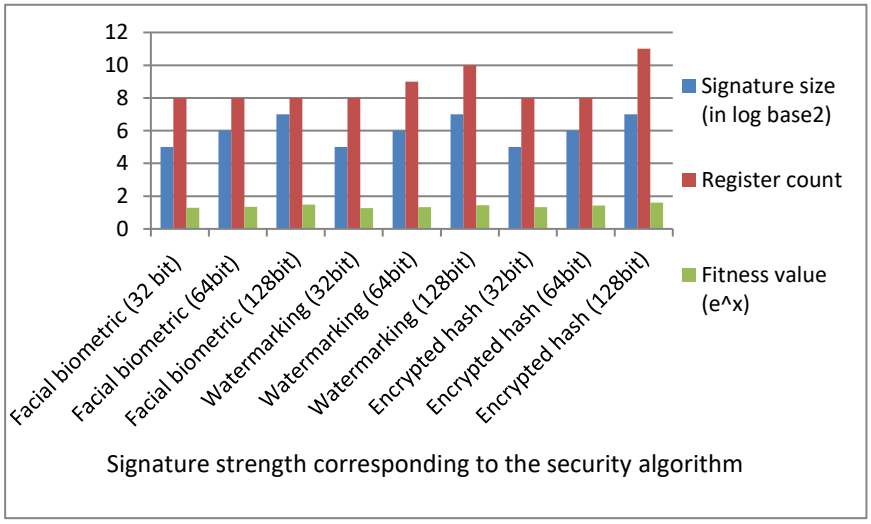


Figure 9.11 Impact of signature strength on fitness value and register count in ARF application [84]

Table 9.29 Design cost of the proposed approach post embedding facial biometric signature of IP buyer and then of IP seller into the design

DSP benchmarks [84]	No. of registers (**δ**)	Resource configuration	Design cost of baseline design	Design cost after embedding facial biometric of IP buyer	Design cost after embedding facial biometric of IP seller	% Design cost overhead
DCT-8point	16	1(+), 2(*)	0.447	0.447	0.447	0.00%
FIR	16	1(+), 3(*)	0.5697	0.5697	0.5697	0.00%
JPEG-codec	129	3(+), 3(*)	0.2178	0.2178	0.2178	0.00%
ARF	16	2(+), 4(*)	0.4121	0.4121	0.4121	0.00%
IIR	14	1(+),2(*),1(-)	0.5247	0.5247	0.5247	0.00%

Chapter 10

Conclusion and Future work

10.1 Conclusion

The DSP, multimedia and machine learning based applications are prevailing in the modern consumer electronics systems. Therefore, to design the secure hardware IP cores is crucial for modern SoC based designs. However, different entities or design houses involved in the SoC design process are situated globally. This enforces to ensure the trust in hardware before integration of imported IPs into target systems. Therefore, it is crucial to devise robust security measures against external hardware security threats. These threats may pose substantial impact on end consumer, system and as well as on IP vendor/designer itself. This thesis presented novel hardware security techniques for generating secure IP cores to produce secure CE or computing systems, thereby ensure the trust in hardware. The following objectives were accomplished:

- Proposed a 'contact-less palmprint biometric' based hardware security approach for enabling robust and seamless detection of pirated IP versions of DSP designs before being used in CE systems. The proposed approach exploits the naturally unique palm features of an IP vendor to generate biometric signature. The implanted palmprint signature in the form of encoded hardware security constraints is then covertly implanted into design during register allocation phase of HLS process. These covertly implanted hardware security constraints enable seamless detective control against pirated IP versions while incurring negligible design overhead. This produced robust security at lower design cost compared to non-biometric-based IP core protection techniques.
- Proposed a hybrid methodology to secure intellectual property (IP) cores of data intensive DSP applications against the hardware threats of reverse engineering and piracy. The proposed approach exploits multilevel structural obfuscation as 1st line of defense against alteration

of register transfer level (RTL) description of IP core design, ensuring preventive control for hindering RE attack. Additionally, the proposed approach covertly implants an invisible DNA impression into the structurally obfuscated DSP design using robust encoding and encryption using multi-iteration Feistel cipher as a 2nd line of defense, ensuring detective control against piracy. The proposed technique renders more robust security than other contemporary techniques while incurring zero design cost overhead.

- Proposed approach leverages the HLS based methodology for designing secured custom reusable convolutional IP core in CNN. Further, in order to ensure the security of reusable IP core, facial biometric based hardware security has been employed. The proposed methodology exploits the naturally unique facial features of an IP vendor to generate biometric based covert hardware security constraints. These hardware security constraints are responsible for enabling the security in terms of detective control against the integration of pirated convolutional IPs into computing systems. The integrated facial biometric based digital evidence therefore enables to discern and isolate fake/pirated IP versions. This ensures the integration of only genuine CNN IPs in computing and CE products for security of the end consumer and protecting brand value of the original vendor. The facial biometric based security offers seamless detective control against pirated IP versions while incurring zero design cost overhead.
- Proposed HLS based hardware security methodology for designing secure JPEG compression-decompression (CODEC) hardware IP using retinal biometric. The proposed approach exploits naturally unique features of retinal biometric of original IP vendor for securing JPEG-codec IP core, where the covert security constraints corresponding to generated retinal signature are implanted inside the design during higher abstraction level. The proposed approach is capable of offering higher robustness during authentication/verification process due to generation of large number of secret security constraints and highly

distinctive nature of retinal structure. It also enables sturdy isolation of pirated versions of IPs at zero design cost overhead.

- Proposed an exploration methodology that offers low-cost hardware design architectural solution for secured IP cores using particle swarm optimization (PSO). The proposed approach integrates three different hardware security methodologies such as IP facial biometrics, encrypted-hashing and IP watermarking the PSO framework for exploring the hardware architecture tradeoffs of security-design cost for different DSP applications. Further, proposed approach is scalable to perform security design cost tradeoff corresponding to any signature-based security algorithm. The proposed methodology offers the analysis of low-cost architectural resource configuration, impact of signature strength on security-design cost fitness value and, register count of the DSP IP core and security parameter such as probability of co-incidence for various security methodologies for varying (scalable) signature strength.

10.2 Future work

This thesis has presented various hardware security techniques for generating secured IP cores corresponding to different data intensive applications from the various domains such as DSP, multimedia and machine learning etc. In future works, we target the following aspects of designing secure hardware IP cores:

- To design more HLS based low-cost secured IP core solutions for different data intensive applications in the field of medical and Internet of Things (IOT).
- To devise more robust hybrid security solutions for IP cores for handling multiple hardware security threats by providing preventive as well as detective security control. Data intensive frameworks in the domain of machine learning and medical applications are to be exploited for the same.

- To explore more robust security mechanisms using multi modal biometrics as well as 3D biometrics etc., to offer robust and seamless detective control on IP piracy.
- To explore the solution for handling transient fault security of IP cores with integrated piracy detective control mechanism.
- To explore security mechanism comprising of high-level as well as physical level security of the design.

REFERENCES

[1] Mahdiany H. R., Hormati A. and Fakhraie S. M. (2001). A hardware accelerator for DSP system design. in *Proc. ICM*, pp. 141-144.

[2] Schneiderman R. (2010), "DSPs evolving in consumer electronics applications," *IEEE Signal Process. Mag*., vol. 27(3), pp. 6–10.

[3] Castillo E., Meyer-Baese U., Garcia A., Parilla L., Lloris A. (2007). IPP@HDL: Efficient intellectual property protection scheme for IP cores. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,* vol. 15, no. 5, pp. 578–590.

[4] J. A. Roy, F. Koushanfar and I. L. Markov (2008), "EPIC: ending piracy of integrated circuits," in *Proc. DATE*, Munich, pp. 1069-1074.

[5] F. Koushanfar *et al*. (2012), "Can EDA combat the rise of electronic counterfeiting?," in *Proc. DAC*, San Francisco, CA, pp. 133-138.

[6] B. Colombier and L. Bossuet (2015), "Survey of hardware protection of design data for integrated circuits and intellectual properties," *IET Computers & Digital Techniques*, vol. 8, no. 6, pp. 274-287.

[7] B. Colombier (2017), "Methods for protecting intellectual property of IP cores designers," *Micro and nanotechnologies/Microelectronics*, Université de Lyon, NNT : 2017LYSES038.

[8] C. Pilato, S. Garg, K. Wu, R. Karri and F. Regazzoni (2018), "Securing hardware accelerators: a new challenge for high-level synthesis," *IEEE Embedded Syst. Lett.*, vol. 10, no. 3, pp. 77-80.

[9] G. He, C. Dong, Y. Liu and X. Fan (2020), "IPlock: An Effective Hybrid Encryption for Neuromorphic Systems IP Core Protection," *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 612-616.

[10] Y. Xie, C. Bao and A. Srivastava (2017), "Security-Aware 2.5D Integrated Circuit Design Flow Against Hardware IP Piracy," *in Computer*, vol. 50, no. 5, pp. 62-71.

[11] X. Wang, Y. Zheng, A. Basak and S. Bhunia (2015), "IIPS: Infrastructure IP for Secure SoC Design," *IEEE Trans.Comput*., vol. 64, no. 8, pp. 2226-2238.

[12] A. Sengupta, S. P.Mohanty (2016), “High-Level Synthesis of Digital Circuits in the Nanoscale Mobile Electronics Era”, *IET Book: Nano-CMOS and Post-CMOS Electronics: Circuits and Design*, pp: 219 - 261.

[13] B. K. Mohanty and P. K. Meher (2016), “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.

[14] S. Sitjongsataporn, A. Thitinaruemit and S. Prongnuch (2021), “Implementation of High Level Synthesis for Adaptive FIR Filtering on Embedded System,” *2021 7th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, pp. 257-260.

[15] S. Chen, J. Jung, P. Song, K. Chakrabarty and G. -J. Nam (2020), “BISTLock: Efficient IP Piracy Protection using BIST,” *2020 IEEE International Test Conference (ITC)*, pp. 1-5.

[16] M. T. Arafin, A. Stanley and P. Sharma (2017), “Hardware-based anti-counterfeiting techniques for safeguarding supply chain integrity,” *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-4.

[17] M. Yasin, J. J. Rajendran, O. Sinanoglu and R. Karri (2016), “On improving the security of logic locking,” *IEEE Trans. on CAD of Integr. Circuits Syst.*, vol. 35, no. 9, pp. 1411-1424.

[18] S. M. Plaza, I. L. Markov (2015), “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.

[19] S. M. Saeed, A. Zulehner, R. Wille, R. Drechsler and R. Karri (2019), “Reversible Circuits: IC/IP Piracy Attacks and Countermeasures,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 11, pp. 2523-2535.

[20] A. R. D. Rizo, J. Leonhard, H. Aboushady and H. -G. Stratigopoulos, (2022), “RF Transceiver Security Against Piracy Attacks,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 7, pp. 3169-3173.

[21] Potluri, A. Aysu and A. Kumar (2020), “SeqL: Secure Scan-Locking for IP Protection,” *2020 21st International Symposium on Quality Electronic Design (ISQED)*, pp. 7-13.

[22] D. Mouris, C. Gouert and N. G. Tsoutsos (2022), “Privacy-Preserving IP Verification,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 7, pp. 2010-2023.

[23] A. Hroub and M. E. S. Elrabaa (2022), “SecSoC: A Secure System on Chip Architecture for IoT Devices,” *2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2022, pp. 41-44.

[24] M. Rathor and A. Sengupta (2021), “Signature Biometric based Authentication of IP Cores for Secure Electronic Systems,” *2021 IEEE International Symposium on Smart Electronic Systems (iSES)*, 2021, pp. 384-388.

[25] W. Hu, C. -H. Chang, A. Sengupta, S. Bhunia, R. Kastner and H. Li (2021), “An Overview of Hardware Security and Trust: Threats, Countermeasures, and Design Tools,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 6, pp. 1010-1038.

[26] F. Koushanfar and G. Qu (2001), “Hardware metering,” *in Proc. DAC*, pp. 490-493.

[27] D. Ziener and J. Teich (2008), “Power signature watermarking of IP cores for FPGAs,” *J. Signal Process. Syst*., vol. 51, no. 1, pp. 123–136.

[28] A. Cui and C. Chang (2007), “Watermarking for IP protection through template substitution at logic synthesis level,” *Proc. ISCAS*, New Orleans, LA, pp. 3687-3690.

[29] M. Ni and Z. Gao (2005), “Detector-based watermarking technique for soft IP core protection in high synthesis design level,” *Proc. CCS*, Hong Kong, pp. 1348–1352.

[30] S. Rai, A. Rupani, P. Nath and A. Kumar (2019), “Hardware Watermarking Using Polymorphic Inverter Designs Based On Reconfigurable Nanotechnologies,” *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 663-669.

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

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

[33] M. Potkonjak and I. Hong (1999), "Behavioral Synthesis Techniques for Intellectual Property Protection," *IEEE/ACM Design Automation Conference*, New Orleans, Louisiana, United States, pp. 849-854.

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

[35] R. Karmakar and S. Chattopadhyay (2020), “Hardware IP Protection Using Logic Encryption and Watermarking,” *2020 IEEE International Test Conference (ITC)*, 2020, pp. 1-10.

[36] A. Sengupta, D. Roy and S. P. Mohanty (2018), “Triple-Phase Watermarking for Reusable IP Core Protection During Architecture Synthesis,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 4, pp. 742-755.

[37] A. Sengupta and M. Rathor (2019), “IP core steganography for protecting DSP kernels used in CE systems,” *IEEE Trans. Consum. Electron.*, vol. 65, no. 4, pp. 506-515.

[38] M. Rathor and A. Sengupta (2020), “IP Core Steganography Using Switch Based Key-Driven Hash-Chaining and Encoding for Securing DSP Kernels Used in CE Systems,” *IEEE Trans. Consum. Electron*., vol. 66, no. 3, pp. 251-260.

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

[40] A. Sengupta and M. Rathor (2020), “Securing hardware accelerators for CE systems using biometric fingerprinting,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 9, pp. 1979-1992.

[41] A. Sengupta and M. Rathor (2021), “Facial Biometric for Securing Hardware Accelerators,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 1, pp. 112 – 123.

[42] S. Chen, Z. Guo, J. Feng and J. Zhou (2020), "An improved contact-based high-resolution palmprint image acquisition system," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 9, pp. 6816-6827.

[43] Q. Zhao, X. Wu and W. Bu (2013), "Contactless palmprint verification based on SIFT and iterative RANSAC," *in Proc. ICIP*, Melbourne, pp. 4186-4189.

[44] J. P. Patil, C. Nayak and M. Jain (2015), "Palmprint recognition using DWT, DCT and PCA techniques," *in Proc. ICCIC*, Madurai, pp. 1-5.

[45] M. Pudzs, R. Fuksis, R. Ruskuls, T. Eglitis, A. Kadikis and M. Greitans (2013), "FPGA based palmprint and palm vein biometric system," *in Proc. BIOSIG*, Darmstadt, pp. 1-4, eISBN: 978-3-88579-606-0.

[46] R. Shi and D. Sun, (2007), "A New Security Scheme based on Palmprint Biometrics for Signature," *IEEE International Conference on Biometrics: Theory, Applications, and Systems*, Crystal City, VA, pp. 1-6.

[47] I. Bouraoui and F. Merazka (2015), "Performance and security evaluation of palmprint biometric templates," *4th International Conference on Electrical Engineering (ICEE)*, Boumerdes, pp. 1-4.

[48] J. I. Agbinya (2019), "Human Palm Geometry Modelling for Biometric Security Systems," *Cybersecurity and Cyberforensics Conference (CCC)*, Melbourne, Australia, pp. 160-164.

[49] B. Arslan, E. Yorulmaz, B. Akca and S. Sagiroglu,(2016), "Security Perspective of Biometric Recognition and Machine Learning Techniques," *15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Anaheim, CA, pp. 492-497.

[50] B. Ríos-Sánchez, M. Viana-Matesanz and C. Sánchez-Ávila (2017), "A comparative study of palmprint feature extraction methods for contact-less biometrics under different environmental conditions," *IEEE International Carnahan Conference on Security Technology (ICCST)*, Madrid, pp. 1-6.

[51] J. Qiu, H. Li, J. Dong and G. Feng (2017), "Biometrics Encryption Based on Palmprint and Convolutional Code," *IEEE International Conference on Multimedia and Image Processing (ICMIP)*, Wuhan, pp. 187-190.

[52] Y. Xu, L. Fei and D. Zhang (2015), "Combining Left and Right Palmprint Images for More Accurate Personal Identification," *IEEE Trans. Image Process.*, vol. 24, no. 2, pp. 549-559.

[53] A. Sengupta, D. Roy, S. P. Mohanty and P. Corcoran (2017), "DSP design protection in CE through algorithmic transformation based structural obfuscation," *IEEE Trans. Consum. Electron*, vol. 63, no. 4, pp. 467-476.

[54] Xue, M., Gu, C., Liu, W., Yu, S. and O'Neill, M. (2020), "Ten years of hardware Trojans: a survey from the attacker's perspective," *IET Comput. Digit. Tech*., 14: 231-246.

[55] Sengupta A., Roy D., Mohanty S.P., and Corcoran P. (2018). Low-cost obfuscated JPEG CODEC IP core for secure CE hardware. I*EEE Trans. Consum. Electron*., vol. 64(3), pp. 365–374.

[56] Sengupta A., Mohanty S. P., Pescador F., Corcoran P. (2018). Multi-Phase Obfuscation of Fault Secured DSP Designs with Enhanced Security Feature. *IEEE Transactions on Consumer Electronics*, vol. 64(3), pp: 356-364.

[57] Sengupta A. and Roy D. (2017). Protecting an intellectual property core during architectural synthesis using high-level transformation based obfuscation. *IET Electronics Letters*, vol: 53(13), pp. 849 – 851.

[58] Zeiler, Matthew & Fergus, Rob. (2013), "Visualizing and understanding convolutional neural networks," in *Proc. ECCV*, 2014, Part I, LNCS 8689.

[59] D. Davalle, B. Carnevale, S. Saponara, L. Fanucci and P. Terreni (2014), "Hardware accelerator for fast image/video thinning," in *Proc. IST*, pp. 64-67.

[60] J. Lemley, S. Bazrafkan and P. Corcoran (2017), "Deep learning for consumer devices and services: pushing the limits for machine learning, artificial intelligence, and computer vision," *IEEE Consum. Electron. Mag.*, vol. 6, no. 2, pp. 48-56.

[61] Z. Zhao, P. Zheng, S. Xu and X. Wu (2019), "Object detection with deep learning: a review," *IEEE Trans. Neural Netw. Learn. Syst*., vol. 30, no. 11, pp. 3212-3232.

[62] A. Sengupta and M. Rathor (2020), "Obfuscated hardware accelerators for image processing filters—application specific and functionally reconfigurable processors," *IEEE Trans. Consum. Electron*., vol. 66, no. 4, pp. 386-395.

[63] A. Kyriakos, V. Kitsakis, A. Louropoulos, E. -A. Papatheofanous, I. Patronas and D. Reisis (2019), "High Performance accelerator for CNN applications," in *Proc. PATMOS*, pp. 135-140.

[64] D. Tsiktsiris, D. Ziouzios and M. Dasygenis (2018), "A portable image processing accelerator using FPGA," in *Proc. MOCAST*, pp. 1-4.

[65] Liu, Z., Dou, Y., Jiang, J., Xu, J., Li, S., Zhou, Y., & Xu, Y. (2017), "Throughput-optimized FPGA accelerator for deep convolutional neural networks," *ACM Trans. Reconfigurable Technol. Syst*., 10, 1 – 23.

[66] Y. Shen, T. Ji, M. Ferdman and P. Milder (2019), "Argus: An end-to-end framework for accelerating CNNs on FPGAs," *IEEE Micro*, vol. 39, no. 5, pp. 17-25.

[67] H. Srivastava and K. Sarawadekar (2020), "A depthwise separable convolution architecture for CNN accelerator," *Proc. ASPCON*, pp. 1-5.

[68] L. Bai, Y. Zhao and X. Huang (2018), "A CNN accelerator on FPGA using depthwise separable convolution," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 65, no. 10, pp. 1415-1419.

[69] M. Chang, Z. Pan and J. Chen (2017), "Hardware accelerator for boosting convolution computation in image classification applications," in *Proc. GCCE*, pp. 1-2.

[70] Y. Ma, Y. Cao, S. Vrudhula and J. Seo (2018), "Optimizing the convolution operation to accelerate deep neural networks on FPGA," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*., vol. 26, no. 7, pp. 1354-1367.

[71] K. Guo *et al.* (2016), "Angel-Eye: A Complete design flow for mapping CNN onto customized hardware," in *Proc. ISVLSI*, pp. 24-29.

[72] T. S. Kim, J. Bae and M. H. Sunwoo (2019), "Fast convolution algorithm for convolutional neural networks," in *Proc. AICAS*, pp. 258-261.

[73] S. Albawi, T. A. Mohammed and S. Al-Zawi (2017), "Understanding of a convolutional neural network," in *Proc. ICET*, pp. 1-6.

[74] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen. (2018), "Recent advances in convolutional neural networks," *Pattern Recogn*." 77, C, 354–377.

[75] B. D. Haeffele and R. Vidal (2017), "Global Optimality in Neural Network Training," 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4390-4398.

[76] S. Aleem, B. Sheng, P. Li, P. Yang and D. D. Feng (2019), "Fast and Accurate Retinal Identification System: Using Retinal Blood Vasculature Landmarks," *IEEE Trans Ind. Informat.*, vol. 15, no. 7, pp. 4099-4110.

[77] Multimedia Laboratory datasets, Available: https://www.medicmind.tech/retinal-image-databases, accessed in 2022.

[78] V. Mishra and A. Sengupta (2014), "MO-PSE: Adaptive Multi Objective Particle Swarm Optimization Based Design Space Exploration in Architectural Synthesis for Application Specific Processor Design", *Elsevier Journal on Adv. in Eng. Softw.*, Vol. 67, pp. 111-124.

[79] J. Lach, W.H. Mangione-Smith, M. Potkonjak (2001), Fingerprinting techniques for field-programmable gate array intellectual property protection, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*. 20 (10) 1253–1261.

[80] D. Roy and A. Sengupta (2017), "Low Overhead Symmetrical Protection of Reusable IP Core using Robust Fingerprinting and Watermarking during High Level Synthesis," *Future Gener. Comput. Syst.*, Volume 71, June 2017, pp. 89–101.

[81] Sait, S. M., & Youssef, H. (1999). *VLSI physical design automation: theory and practice* (Vol. 6). World Scientific Publishing Company.

[82] Sengupta A. (2020). Secured hardware accelerators for DSP and image processing applications. *The Institute of Engineering and Technology (IET) Book*, e-ISBN: 9781839533075.

[83] Sengupta A. (2020). Frontiers in securing IP cores - Forensic detective control and obfuscation techniques. *The Institute of Engineering and Technology (IET) Book*, ISBN-10: 1-83953-031-6, ISBN-13: 978-1-83953-031-9.

[84] Express benchmark suite, University of California San Diego, 2016, https://www.ece.ucsb.edu/EXPRESS/benchmark/.

[85] CAD for Assurance, IEEE Hardware Security and Trust Technical Committee, https://cadforassurance.org/tools/ip-ic-protection/faciometric-hardware-security-tool/, accessed on Jan 2022.

[86] 15 nm open cell library. [Online]. Available: https://si2.org/open-cell-library/, last accessed on Jan. 2020.

[87] G. Martin and G. Smith (2009), "High-Level Synthesis: Past, Present, and Future," *IEEE Design & Test of Computers*, vol. 26, no. 4, pp. 18-25.

[88] Gorman, C. (2012). Counterfeit Chips on the Rise. *IEEE Spectrum*. 49. 16-17. 10.1109/MSPEC.2012.6203952.

[89] Guin U., Huang K., DiMase D., Carulli J. M., Tehranipoor M. and Makris Y. (2014). Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain. *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207-1228.

[90] Mitra S., Wong H.P. and Wong S. (2015). The Trojan-proof chip. *IEEE Spectrum*, vol. 52, no. 2, pp. 46-51.

[91] Rajendran, J., Zhang, H., Sinanoglu, O., & Karri, R. (2013). High-level synthesis for security and trust. In *On-Line Testing Symposium (IOLTS), 2013 IEEE 19th International,* pp. 232-233.

[92] Y. Lao and K. K. Parhi (2015), "Obfuscating DSP Circuits via High-Level Transformations," *IEEE Trans. Very Large Scale Integr. Syst.,* vol. 23 (5), pp. 819–830.

[93] D. Roy and A. Sengupta (2019), "Multilevel Watermark for Protecting DSP Kernel in CE Systems, "*IEEE Consum. Electron. Mag.*, vol. 8, no. 2, pp. 100-102.

[94] Torrance, R., & James, D. (2009). The state-of-the-art in IC reverse engineering. In *Cryptographic Hardware and Embedded Systems-CHES 2009* (pp. 363-381). Springer, Berlin, Heidelberg.

[95] R. Chaurasia, A. Anshul, A. Sengupta and S. Gupta (2022), "Palmprint Biometric Versus Encrypted Hash Based Digital Signature for Securing DSP Cores Used in CE Systems," *IEEE Consum. Electron. Mag*., vol. 11, no. 5, pp. 73-80.

[96] D. Dolev and A. Yao (1983), "On the security of public key protocols," *IEEE Trans.Inf. Theory*, vol. 29, no. 2, pp. 198-208.

[97] J. Dong, X. Meng, M. Chen and Z. Wang (2017), "Template protection based on DNA coding for multimodal biometric recognition," *4th International Conference on Systems and Informatics (ICSAI)*, Hangzhou, China, 2017, pp. 1738-1742.